System Level Modeling And Design Space Exploration For Multiprocessor Embedded System On Chip Architectures Aup Dissertation Series

The editors and authors present a wealth of knowledge regarding the most relevant aspects in the field of MOS transistor modeling. The variety of subjects and the high quality of content of this volume make it a reference document for researchers and users of MOSFET devices and models. The book can be recommended to everyone who is involved in compact model developments, numerical TCAD modeling, parameter extraction, space-level simulation or model standardization. The book will appeal equally to PhD students who want to understand the ins and outs of MOSFETs as well as to modeling designers working in the analog and high-frequency areas.

Visit the authors' companion site! http://www.electronicsystemlevel.com/ - Includes interactive forum with the authors! Electronic System Level (ESL) design has mainstreamed – it is now an established approach at most of the world’s leading system-on-chip (SoC) design companies and is being used increasingly in system design. From its genesis as an algorithm modeling methodology with ‘no links to implementation’, ESL is evolving into a set of complementary methodologies that enable embedded system design, verification and debug through to the hardware and software implementation of custom SoC, system-on-FPGA, system-on-board, and entire multi-board systems. This book arises from experience the authors have gained from years of work as industry practitioners in the Electronic System Level design area; they have seen "SLD" or "ESL" go through many stages and false starts, and have observed that the shift in design methodologies to ESL is finally occurring. This is partly because of ESL technologies themselves are stabilizing on a useful set of languages being standardized (SystemC is the most notable), and use models are being identified that are beginning to get real adoption. ESL DESIGN & VERIFICATION offers a true prescriptive guide to ESL that reviews its past and outlines the best practices of today. Table of Contents CHAPTER 1: WHAT IS ESL? CHAPTER 2: TAXONOMY AND DEFINITIONS FOR THE ELECTRONIC SYSTEM LEVEL CHAPTER 3: EVOLUTION OF ESL DEVELOPMENT CHAPTER 4: WHAT ARE THE ENABLERS OF ESL? CHAPTER 5: ESL FLOW CHAPTER 6: SPECIFICATIONS AND MODELING CHAPTER 7: PRE-PARTITIONING ANALYSIS CHAPTER 8: PARTITIONING CHAPTER 9: POST-PARTITIONING ANALYSIS AND DEBUG CHAPTER 10: POST-PARTITIONING VERIFICATION CHAPTER 11: HARDWARE IMPLEMENTATION CHAPTER 12: SOFTWARE IMPLEMENTATION CHAPTER 13: USE OF ESL FOR IMPLEMENTATION VERIFICATION CHAPTER 14: RESEARCH, EMERGING AND FUTURE PROSPECTS APPENDIX: LIST OF ACRONYMS * Provides broad, comprehensive coverage not available in any other such book * Massive global appeal with an internationally recognised author team * Crammed full of state of the art content from notable industry experts

This book is devoted to modeling of multi-level complex systems, a challenging domain for engineers, researchers and entrepreneurs, confronted with the transition from learning and adaptability to evolvability and autonomy for technologies, devices and problem solving methods. Chapter 1 introduces the multi-scale and multi-level systems and highlights their presence in different domains of science and technology. Methodologies as, random systems, non-Archimedean analysis, category theory and specific techniques as model categorification and integrative closure, are presented in chapter 2. Chapters 3 and 4 describe polystochastic models, PSM, and their developments. Categorical formulation of integrative closure offers the general PSM framework which serves as a flexible guideline for a large variety of multi-level modeling problems. Focusing on chemical engineering, pharmaceutical and environmental case studies, the chapters 5 to 8 analyze mixing, turbulent dispersion and entropy production for multi-scale systems. Taking inspiration from systems sciences, chapters 9 to 11 highlight multi-level modeling potentialities in formal concept analysis, existential graphs and evolvable designs of experiments. Case studies refer to separation flow-sheets, pharmaceutical pipeline, drug design and development, reliability management systems, security and failure analysis. Perspectives and integrative points of view are discussed in chapter 12. Autonomous and viable systems, multi-agents, organic and autonomic computing, multi-level informational systems, are revealed as promising domains for future applications. Written for: engineers, researchers, entrepreneurs and students in chemical, pharmaceutical, environmental and systems sciences engineering, and for applied mathematicians.

Embedded computer systems are now everywhere: from alarm clocks to PDAs, from mobile phones to cars, almost all the devices we use are controlled by embedded computers. An important class of embedded computer systems is that of hard real-time systems, which have to fulfill strict timing requirements. As real-time systems become more complex, they are often implemented using distributed heterogeneous architectures. Analysis and Synthesis of Distributed Real-Time Embedded Systems addresses the design of real-time applications implemented using distributed heterogeneous architectures. The systems are heterogeneous not only in terms of hardware components, but also in terms of communication protocols and scheduling policies. Regarding this last aspect, time-driven and event-driven systems, as well as a combination of the two, are considered. Such systems are used in many application areas like automotive electronics, real-time multimedia, avionics, medical equipment, and factory systems. The proposed analysis and synthesis techniques derive optimized implementations that fulfill the imposed design constraints. An important part of the implementation process is the synthesis of the communication infrastructure, which has a

significant impact on the overall system performance and cost. Analysis and Synthesis of Distributed Real-Time Embedded Systems considers the mapping and scheduling tasks within an incremental design process. To reduce the time-to-market of products, the design of real-time systems seldom starts from scratch. Typically, designers start from an already existing system, running certain applications, and the design problem is to implement new functionality on top of this system. Supporting such an incremental design process provides a high degree of flexibility, and can result in important reductions of design costs. STRONGAnalysis and Synthesis of Distributed Real-Time Embedded Systems will be of interest to advanced undergraduates, graduate students, researchers and designers involved in the field of embedded systems.

High-Level Modeling and Directed Test Generation Techniques

System Design with SystemC™

Device-Level Modeling and Synthesis of High-Performance Pipeline ADCs

System Level Design with .Net Technology

Quality-Driven SystemC Design

System-level Modeling of MEMS

The emergence of the system-on-chip (SoC) era is creating many new challenges at all stages of the design process. Engineers are reconsidering how designs are specified, partitioned and verified. With systems and software engineers programming in C/C++ and their hardware counterparts working in hardware description languages such as VHDL and Verilog, problems arise from the use of different design languages, incompatible tools and fragmented tool flows. Momentum is building behind the SystemC language and modeling platform as the best solution for representing functionality, communication, and software and hardware implementations at various levels of abstraction. The reason is clear: increasing design complexity demands very fast executable specifications to validate system concepts, and only C/C++ delivers adequate levels of abstraction, hardware-software integration, and performance. System design today also demands a single common language and modeling foundation in order to make interoperable system--level design tools, services and intellectual property a reality. SystemC is entirely based on C/C++ and the complete source code for the SystemC reference simulator can be freely downloaded from www.systemc.org and executed on both PCs and workstations. System Design and SystemC provides a comprehensive introduction to the powerful modeling capabilities of the SystemC language, and also provides a large and valuable set of system level modeling examples and techniques. Written by experts from Cadence Design Systems, Inc. and Synopsys, Inc. who were deeply involved in the definition and implementation of the SystemC language and reference simulator, this book will provide you with the key concepts you need to be successful with SystemC. System Design with SystemC thoroughly covers the new system level modeling capabilities available in SystemC 2.0 as well as the hardware modeling capabilities available in earlier versions of SystemC. designed and implemented the SystemC language and reference simulator, this book will provide you with the key concepts you need to be successful with SystemC. System Design with SystemC will be of interest to designers in industry working on complex system designs, as well as students and researchers within academia. All of the examples and techniques described within this book can be used with freely available compilers and debuggers – no commercial software is needed. Instructions for obtaining the free source code for the examples obtained within this book are included in the first chapter.

This book serves as a reference for researchers and designers in Embedded Systems who need to explore design alternatives. It provides a design space exploration methodology for the analysis of system characteristics and the selection of the most appropriate architectural solution to satisfy requirements in terms of performance, power consumption, number of required resources, etc. Coverage focuses on the design of complex multimedia applications, where the choice of the optimal design alternative in terms of application/architecture pair is too complex to be pursued through a full search comparison, especially because of the multi-objective nature of the designer's goal, the simulation time required and the number of parameters of the multi-core architecture to be optimized concurrently.

The cumulative impact of co-channel interferers, commonly referred to as aggregate network interference, is one of the main performance limiting factors in today's mobile cellular networks. Thus, its careful statistical description is decisive for system analysis and design. A system model for interference analysis is required to capture essential network variation effects, such as base station deployment- and signal propagation characteristics. Furthermore it should be simple and tractable so as to enable first-order insights on design fundamentals and rapid exchange of new ideas. Interference modeling has posed a challenge ever since the establishment of traditional macro-cellular deployments. The recent emergence of heterogeneous network topologies complicates matters by contesting many established aspects of time-honored approaches. This book presents user-centric system models that enable to investigate scenarios with an anisotropic interference impact.

System-level modeling of MEMS - microelectromechanical systems - comprises integrated approaches to simulate, understand, and optimize the performance of sensors, actuators, and microsystems, taking into account the intricacies of the interplay between mechanical and electrical properties, circuitry, packaging, and design considerations. Thereby, system-level modeling overcomes the limitations inherent to methods that focus only on one of these aspects and do not incorporate their mutual dependencies. The book addresses the two most important approaches of system-level modeling, namely physics-based modeling with lumped elements and mathematical modeling employing model order reduction methods, with an emphasis on combining single device models to entire systems. At a clearly understandable and sufficiently detailed level the readers are made familiar with the physical and mathematical underpinnings of MEMS modeling. This enables them to choose the adequate methods for the respective application needs. This work is an invaluable resource for all materials scientists, electrical engineers, scientists working in the semiconductor and/or sensor industry, physicists, and physical chemists.

SystemC Kernel Extensions for Heterogeneous System Modeling

A Method for High-Level System Design and Analysis

Modeling, Synthesis and Verification

Combining Formal Model-based System-level Design with SystemC Transaction Level Modeling

Multi-objective Design Space Exploration of Multiprocessor SoC Architectures

Embedded Software for SoC

Various approaches for finding optimal values for the parameters of analog cells have made their entrance in commercial applications. However, a larger impact on the performance is expected if tools are developed which operate on a higher abstraction level and consider multiple architectural choices to realize a particular functionality. This book examines the opportunities, conditions, problems, solutions and systematic methodologies for this new generation of analog CAD tools.

In system design, generation of high-level abstract models that can be closely associated with evolving lower-level models provides designers with the ability to incrementally `test' an evolving design against a model of a specification. Such high-level models may deal with areas such as performance, reliability, availability, maintainability, and system safety. Abstract models also allow exploration of the hardware versus software design space in an incremental fashion as a fuller, detailed design unfolds, leaving behind the old practice of hardware-software binding too early in the design process. Such models may also allow the inclusion of non-functional aspects of design (e.g. space, power, heat) in a simulatable information model dealing with the system's operation. This book addresses Model Generation and Application specifically in the following domains: Specification modeling (linking object/data modeling, behavior modeling, and activity modeling). Operational specification modeling (modeling the way the system is supposed to operate - from a user's viewpoint). Linking non-functional parameters with specification models. Hybrid modeling (linking performance and functional elements). Application of high-level modeling to hardware/software approaches. Mathematical analysis techniques related to the modeling approaches. Reliability modeling. Applications of High Level Modeling. Reducing High Level Modeling to Practice. High-Level System Modeling: Specification and Design Methodologies describes the latest research and practice in the modeling of electronic systems and as such is an important update for all researchers, design engineers and technical managers working in design automation and circuit design.

A quality-driven design and verification flow for digital systems is developed and presented in Quality-Driven SystemC Design. Two major enhancements characterize the new flow: First, dedicated verification techniques are integrated which target the different levels of abstraction. Second, each verification technique is complemented by an approach to measure the achieved verification quality. The new flow distinguishes three levels of abstraction (namely system level, top level and block level) and can be incorporated in existing approaches. After reviewing the preliminary concepts, in the following chapters the three levels for modeling and verification are considered in detail. At each level the verification quality is measured. In summary, following the new design and verification flow a high overall quality results.

System Level Modeling and Design Using SysML and SystemCSystem-Level ValidationHigh-Level Modeling and Directed Test Generation TechniquesSpringer Science & Business Media

The Symbiosis of Theory and Simulations

Transistor Level Modeling for Analog/RF IC Design

Chip-level Modeling with VHDL

Electronic Design Automation for IC System Design, Verification, and Testing

ESL Design and Verification

ESL Models and their Application

Integrated System-Level Modeling of Network-on-Chip Enabled Multi-Processor Platforms first gives a comprehensive update on recent developments in the area of SoC platforms and ESL design methodologies. The main contribution is the rigorous definition of a framework for modeling at the timing approximate level of abstraction. Subsequently this book presents a set of tools for the creation and exploration of timing approximate SoC platform models.

Current multimedia and telecom applications require complex, heterogeneous multiprocessor system on chip (MPSoC) architectures with specific communication infrastructure in order to achieve the required performance. Heterogeneous MPSoC includes different types of processing units (DSP, microcontroller, ASIP) and different communication schemes (fast links, non standard memory organization and access). Programming an MPSoC requires the generation of efficient software running on MPSoC from a high level environment, by using the characteristics of the architecture. This task is known to be tedious and error prone, because it requires a combination of high level programming environments with low level software design. This book gives an overview of concepts related to embedded software design for MPSoC. It details a full software design approach, allowing systematic, high-level mapping of software applications on heterogeneous MPSoC. This approach is based on gradual refinement of hardware/software interfaces and simulation models allowing to validate the software at different abstraction levels. This book combines Simulink for high level programming and SystemC for the low level software development. This approach is illustrated with multiple examples of application software and MPSoC architectures that can be used for deep understanding of software design for MPSoC.

Perhaps nothing characterizes the inherent heterogeneity in embedded sys tems than the ability to choose between hardware and software implementations of a given system function. Indeed, most embedded systems at their core repre sent a careful division and design of hardware and software parts of the system To do this task effectively, models and methods are necessary functionality. to capture application behavior, needs and system implementation constraints. Formal modeling can be valuable in addressing these tasks. As with most engineering domains, co-design practice defines the state of the it seeks to add new capabilities in system conceptualization, mod art, though eling, optimization and implementation. These advances -particularly those related to synthesis and verification tasks -direct1y depend upon formal under standing of system behavior and performance measures. Current practice in system modeling relies upon exploiting high-level programming frameworks, such as SystemC, EstereI, to capture design at increasingly higher levels of ab straction and attempts to reduce the system implementation task. While raising the abstraction levels for design and verification tasks, to be really useful, these approaches must also provide for reuse, adaptation of the existing intellectual property (IP) blocks.

Presenting a comprehensive overview of the design automation algorithms, tools, and methodologies used to design integrated circuits, the Electronic Design Automation for Integrated Circuits Handbook is available in two volumes. The first volume, EDA for IC System Design, Verification, and Testing, thoroughly examines system-level design, microarchitectural design, logical verification, and testing. Chapters contributed by leading experts authoritatively discuss processor modeling and design tools, using performance metrics to select microprocessor cores for IC designs, design and verification languages, digital simulation, hardware acceleration and emulation, and much more. Save on the complete set.

Formal Methods and Models for System Design

A Framework for Multi-MoC Modeling & Simulation

System Level Design Model with Reuse of System IP

System Level Modeling and Design Using SysML and SystemC

System-Level Validation

Transaction-Level Modeling with SystemC

Abstract: Since their introduction, modern computer systems have been increasing in complexity. System designers have been dealing with ever larger designs by moving to higher abstraction level system descriptions. The existing register transfer level of abstraction has become unable to handle modern designs, requiring a move to high level modeling. The most popular of the current approaches for high level design is using SystemC, a set of libraries built in C++, to model hardware using software concepts. However, software based approaches suffer from a major drawback--the lack of a formal definition for both communication and computation. Basic hardware primitives such as concurrency and multiparty communication cannot be easily expressed in software and the translation of these models into hardware equivalents is difficult. As a result, most designers choose to represent systems as a high level SystemC model for simulation, and a set of register transfer level designs for implementation. This gap presents challenges in the design and verification of the system. This work proposes a novel high level modeling methodology, called Lyra, which uses the well studied concepts of finite state machines for computation and of rendezvous for communication. A rendezvous is a bidirectional, atomic, synchronous communication construct that supports a wide variety of communication patterns such as multiparty and variable party communication. The presence of a novel mechanism to handle nondeterminism from the use of rendezvous allows Lyra to model designs that existing rendezvous based approaches cannot. Finite state machine based modeling makes Lyra amenable to hardware implementation and easily understandable by hardware engineers. The formal foundation of Lyra and the ability to implement models as hardware are advantages compared to other high level modeling approaches. This thesis presents Lyra and the novel rendezvous nondeterminism resolution mechanism, called the communication scheduler. It develops a graph based method to analyze and compare different rendezvous based approaches. This Work demonstrates the implementation of Lyra into a simulator and a synthesis tool, creating a practical design flow. This work examines some system models that demonstrate the benefits of using Lyra.

Suitable for bookstore catalogue

This title covers all software-related aspects of SoC design, from embedded and application-domain specific operating systems to system architecture for future SoC. It will give embedded software designers invaluable insights into the constraints imposed by the use of embedded software in an SoC context.

This book addresses system design, providing a framework for assessing and developing system design practices that observe and utilise reuse of system design know-how. The know-how accumulated in the companies represents an intellectual asset, or property ('IP').

Out-of-order Parallel Discrete Event Simulation for Electronic System-level Design

The MULTICUBE Approach

High-Level Modeling and Synthesis of Analog Integrated Systems

Specification and Design Methodologies

Embedded Software Design and Programming of Multiprocessor System-on-Chip

An Open-Source Approach

This book offers readers a set of new approaches and tools a set of tools and techniques for facing challenges in parallelization with design of embedded systems. It provides an advanced parallel simulation infrastructure for efficient and effective system-level model validation and development so as to build better products in less time. Since parallel discrete event simulation (PDES) has the potential to exploit the underlying parallel computational capability in today's multi-core simulation hosts, the author begins by reviewing the parallelization of discrete event simulation, identifying problems and solutions. She then describes out-of-order parallel discrete event simulation (OoO PDES), a novel approach for efficient validation of system-level designs by aggressively exploiting the parallel capabilities of todays' multi-core PCs. This approach enables readers to design simulators that can fully exploit the parallel processing capability of the multi-core system to achieve fast speed simulation, without loss of simulation and timing accuracy. Based on this parallel simulation infrastructure, the author further describes automatic approaches that help the designer quickly to narrow down the debugging targets in faulty ESL models with parallelism.

The complexity of modern embedded systems has increased rapidly in the recent past. Introducing models of computation into the design flow has significantly raised the abstraction in system level design of embedded systems. Establishing such high abstraction levels in common hardware /software co-design flows is still in its infancy. H. Gregor Molter develops a hardware / software co-design flow based on the Discrete Event System Specification model of computation. He advocates that such a system level design flow should exploit a timed model of computation to allow a broad application field. The presented design flow will transform timed DEVS models to both synthesizable VHDL source code and embeddable C++ source code.

Embedded System Design: Modeling, Synthesis and Verification introduces a model-based approach to system level design. It presents modeling techniques for both computation and communication at different levels of abstraction, such as specification, transaction level and cycle-accurate level. It discusses synthesis methods for system level architectures, embedded software and hardware components. Using these methods, designers can develop applications with high level models, which are automatically translatable to low level implementations. This book, furthermore, describes simulation-based and formal verification methods that are essential for achieving design confidence. The book concludes with an overview of existing tools along with a design case study outlining the practice of embedded system design. Specifically, this book addresses the following topics in detail: . System modeling at different abstraction levels . Model-based system design . Hardware/Software codesign . Software and Hardware component synthesis . System verification This book is for groups within the embedded system community: students in courses on embedded systems, embedded application developers, system designers and managers, CAD tool developers, design automation, and system engineering.

The first book to harness the power of .NET for system design, System Level Design with .NET Technology constitutes a software-based approach to design modeling verification and simulation. World class developers, who have been at the forefront of system design for decades, explain how to tap into the power of this dynamic programming environment for more effective and efficient management of metadata—and introspection and interoperability between tools. Using readily available technology, the text details how to capture constraints and requirements at high levels and describes how to percolate them during the refinement process. Departing from proprietary environments built around System Verilog and VHDL, this cutting-edge reference includes an open source environment (ESys.NET) that readers can use to experiment with new ideas, algorithms, and design methods; and to expand the capabilities of their current tools. It also covers: Modeling and simulation—including requirements specification, IP reuse, and applications of design patterns to hardware/software systems Simulation and validation—including transaction-based models, accurate simulation at cycle and transaction levels, cosimulation and acceleration technique, as well as timing specification and validation Practical use of the ESys.NET environment Worked examples, end of chapter references, and the ESys.NET implementation test bed make this the ideal resource for system engineers and students looking to maximize their embedded system designs.

TLM Concepts and Applications for Embedded Systems

Automatic Test Program Generation Workshop

Modeling Embedded Systems and SoC's

Concurrency and Time in Models of Computation

High-Level System Modeling

System-Level Modeling and Evaluation of Heterogeneous Mobile Networks

The systems engineering method proposed in this book, which is based on Abstract State Machines (ASMs), guides the development of software and embedded hardware-software systems seamlessly from requirements capture to actual implementation and documentation. The method bridges the gap between the human understanding and formulation of real-world problems and the deployment of their algorithmic solutions by code-executing machines. Within a single conceptual framework it covers design, verification by reasoning techniques, and validation by simulation and testing. ASMs improve current industrial practice by using accurate high-level modeling and by linking the descriptions at the successive stages of system development in an organic and efficiently maintainable chain of

rigorous and coherent system models at stepwise-refined abstraction levels. In several industrial projects the ASM method has proven its superiority compared to the popular UML methodology when designing complex parallel or dynamic systems. This book combines the features of a textbook and a handbook: the reader will find detailed explanations, proofs, and exercises as well as numerous examples and real-world case studies. Researchers will find here the most comprehensive description of ASMs available today and professionals will use it as a "modeling handbook for the working software engineer." As a textbook it supports self-study or it can form the basis of a lecture course. Even more information can be found on the related website maintained by the authors: http://www.di.unipi.it/AsmBook/
SystemC Kernel Extensions for Heterogeneous System Modeling is a result of an almost two year endeavour on our part to understand how SystemC can be made useful for system level modeling at higher levels of abstraction. Making it a truly heterogeneous modeling language and platform, for hardware/software co-design as well as complex embedded hardware designs has been our focus in the work reported in this book.
This book arises from experience the authors have gained from years of work as industry practitioners in the field of Electronic System Level design (ESL). At the heart of all things related to Electronic Design Automation (EDA), the core issue is one of models: what are the models used for, what should the models contain, and how should they be written and distributed. Issues such as interoperability and tool transportability become central factors that may decide which ones are successful and those that cannot get sufficient traction in the industry to survive. Through a set of real examples taken from recent industry experience, this book will distill the state of the art in terms of System-Level Design models and provide practical guidance to readers that can be put into use. This book is an invaluable tool that will aid readers in their own designs, reduce risk in development projects, expand the scope of design projects, and improve developmental processes and project planning.
This book presents models and procedures to design pipeline analog-to-digital converters, compensating for device inaccuracies, so that high-performance specs can be met within short design cycles. These models are capable of capturing and predicting the behavior of pipeline data converters within less than half-a-bit deviation, versus transistor-level simulations. As a result, far fewer model iterations are required across the design cycle. Models described in this book accurately predict transient behaviors, which are key to the performance of discrete-time systems and hence to the performance of pipeline data converters.
An Image Compression System Design Using Transaction Level Modeling with SystemC
Analysis and Synthesis of Distributed Real-Time Embedded Systems
A Prescription for Electronic System Level Methodology
A System Level Perspective
SynDEVS Co-Design Flow
Modeling Multi-Level Systems
This book describes for readers a methodology for dynamic power estimation, using Transaction Level Modeling (TLM). The methodology exploits the existing tools for RTL simulation, design synthesis and SystemC prototyping to provide fast and accurate power estimation using Transaction Level Power Modeling (TLPM). Readers will benefit from this innovative way of evaluating power on a high level of abstraction, at an early stage of the product life cycle, decreasing the number of the expensive design iterations.
This book covers state-of-the art techniques for high-level modeling and validation of complex hardware/software systems, including those with multicore architectures. Readers will learn to avoid time-consuming and error-prone validation from the comprehensive coverage of system-level validation, including high-level modeling of designs and faults, automated generation of directed tests, and efficient validation methodology using directed tests and assertions. The methodologies described in this book will help designers to improve the quality of their validation, performing as much validation as possible in the early stages of the design, while reducing the overall validation effort and cost.
Electronic System Level Design: an Open-Source Approach is based on the successful experience acquired with the conception of the ADL ArchC, the development of its underlying tool suite, and the building of its platform modeling infrastructure. With more than 10000 accesses per year since 2004, the dissemination of ArchC models reached not only students in quest of proper infrastructure to develop their research projects but also some companies in need of processor models to build virtual platforms using SystemC. The need to anticipate the development of hardware-dependent software and to build virtual prototypes gave rise to Transaction Level Modeling (TLM). Since SystemC provided the elements and the adequate abstraction level for supporting TLM, their relation has grown so strong that OSCI created a TLM Working Group whose effort resulted in the recently released TLM 2.0 standard, which is also covered in this book.
System level design is a critical component for the methods to develop designs more productively. But there are a number of challenges in implementing system level modeling. This book addresses that need by developing organizing principles for understanding, assessing, and comparing the different models of computation in system level modeling.
A High Level Modeling and Synthesis Methodology for Concurrent Systems Using Rendezvous
Integrated System-Level Modeling of Network-on-Chip enabled Multi-Processor Platforms
A Hardware / Software Co-Design Flow Based on the Discrete Event System Specification Model of Computation

EDA for IC System Design, Verification, and Testing
Transaction-Level Power Modeling
1983 IEEE ATPG Workshop Proceedings, March 15-16, 1983, San Francisco, California

The first of two volumes in the Electronic Design Automation for Integrated Circuits Handbook, Second Edition, Electronic Design Automation for IC System Design, Verification, and Testing thoroughly examines system-level design, microarchitectural design, logic verification, and testing. Chapters contributed by leading experts authoritatively discuss processor modeling and design tools, using performance metrics to select microprocessor cores for integrated circuit (IC) designs, design and verification languages, digital simulation, hardware acceleration and emulation, and much more. New to This Edition: Major updates appearing in the initial phases of the design flow, where the level of abstraction keeps rising to support more functionality with lower non-recurring engineering (NRE) costs Significant revisions reflected in the final phases of the design flow, where the complexity due to smaller and smaller geometries is compounded by the slow progress of shorter wavelength lithography New coverage of cutting-edge applications and approaches realized in the decade since publication of the previous edition—these are illustrated by new chapters on high-level synthesis, system-on-chip (SoC) block-based design, and back-annotating system-level models Offering improved depth and modernity, Electronic Design Automation for IC System Design, Verification, and Testing provides a valuable, state-of-the-art reference for electronic design automation (EDA) students, researchers, and professionals.

Lyra
System-level Modeling and Design with the SpecC Language
Simulink and System C Case Studies
Electronic System Level Design
Abstract State Machines
Electronic System Level Design and Verification in Practice