# Applied Formal Verification For Digital Circuit Design 1st Edition

The Practical, Start-to-Finish Guide to Modern Digital Design Verification As digital logic designs grow larger and more complex, functional verification has become the number one bottleneck in the design process. Reducing verification time is crucial to project success, yet many practicing engineers have had little formal training in verification, and little exposure to the newest solutions.Hardware Design Verificationsystematically presents today's most valuable simulation-based and formal verification techniques, helping test and design engineers choose the best approach for each project, quickly gain confidence in their designs, and move into fabrication far more rapidly. College students will find that coverage of verification principles and common industry practices will help them prepare for jobs as future verification engineers. Author William K. Lam, one of the world's leading experts in design verification, is a recent winner of the Chairman's Award for Innovation, Sun Microsystems' most prestigious technical achievement award. Drawing on his wide-ranging experience, he introduces the foundational principles of verification, presents traditional techniques that have survived the test of time, and introduces emerging techniques for today's most challenging designs. Throughout, Lam emphasizes practical examples rather than mathematical proofs; wherever advanced math is essential, he explains it clearly and accessibly. Coverage includes Simulation-based versus formal verification: advantages, disadvantages, and tradeoffs Coding for verification: functional and timing correctness, syntactical and structure checks, simulation

performance, and more Simulator architectures and operations, including event-driven, cycle-based, hybrid, and hardware-based simulators Testbench organization, design, and tools: creating a fast, efficient test environment Test scenarios and assertion: planning, test cases, test generators, commercial and Verilog assertions, and more Ensuring complete coverage, including code, parameters, functions, items, and cross-coverage The verification cycle: failure capture, scope reduction, bug tracking, simulation data dumping, isolation of underlying causes, revision control, regression, release mechanisms, and tape-out criteria An accessible introduction to the mathematics and algorithms of formal verification, from Boolean functions to state-machine equivalence and graph algorithms Decision diagrams, equivalence checking, and symbolic simulation Model checking and symbolic computation Simply put,Hardware Design Verificationwill help you improve and accelerate your entire verification process--from planning through tape-out--so you can get to market faster with higher quality designs.
Software engineering requires specialized knowledge of a broad spectrum of topics, including the construction of software and the platforms, applications, and environments in which the software operates as well as an understanding of the people who build and use the software. Offering an authoritative perspective, the two volumes of the Encyclopedia of Software Engineering cover the entire multidisciplinary scope of this important field. More than 200 expert contributors and reviewers from industry and academia across 21 countries provide easy-to-read entries that cover software requirements, design, construction, testing, maintenance, configuration management, quality control, and software engineering management tools and methods. Editor Phillip A. Laplante uses the most universally recognized definition of the areas of relevance to software engineering, the Software

Engineering Body of Knowledge (SWEBOK®), as a template for organizing the material. Also available in an electronic format, this encyclopedia supplies software engineering students, IT professionals, researchers, managers, and scholars with unrivaled coverage of the topics that encompass this ever-changing field. Also Available Online This Taylor & Francis encyclopedia is also available through online subscription, offering a variety of extra benefits for researchers, students, and librarians, including: Citation tracking and alerts Active reference linking Saved searches and marked lists HTML and PDF format options Contact Taylor and Francis for more information or to inquire about subscription options and print/online combination packages. US: (Tel) 1.888.318.2367; (E-mail) e-reference@taylorandfrancis.com International: (Tel) +44 (0) 20 7017 6062; (E-mail) online.sales@tandf.co.uk

This book constitutes the refereed proceedings of the 4th International Conference on Formal Engineering methods, ICFEM 2002, held in Shanghai, China, in October 2002. The 43 revised full papers and 16 revised short papers presented together with 5 invited contributions were carefully reviewed and selected from a total of 108 submissions. The papers are organized in topical sections on component engineering and software architecture, method integration, specification techniques and languages, tools and environments, refinement, applications, validation and verification, UML, and semantics.

CHARM '97 is the ninth in a series of working conferences devoted to the development and use of formal techniques in digital hardware design and verification. This series is held in collaboration with IFIP WG 10.5. Previous meetings were held in Europe every other year.

This book constitutes the refereed proceedings of the Second International Conference on Formal Methods in Computer-Aided Design, FMCAD '98, held in Palo Alto, California, USA, in

November 1998. The 27 revised full papers presented were carefully reviewed and selected from a total of 55 submissions. Also included are four tools papers and four invited contributions. The papers present the state of the art in formal verification methods for digital circuits and systems, including processors, custom VLSI circuits, microcode, and reactive software. From the methodological point of view, binary decision diagrams, model checking, symbolic reasoning, symbolic simulation, and abstraction methods are covered.

This book presents the technical program of the International Embedded Systems Symposium (IESS) 2009. Timely topics, techniques and trends in embedded system design are covered by the chapters in this volume, including modelling, simulation, verification, test, scheduling, platforms and processors. Particular emphasis is paid to automotive systems and wireless sensor networks. Sets of actual case studies in the area of embedded system design are also included. Over recent years, embedded systems have gained an enormous amount of proce-ing power and functionality and now enter numerous application areas, due to the fact that many of the formerly external components can now be integrated into a single System-on-Chip. This tendency has resulted in a dramatic reduction in the size and cost of embedded systems. As a unique technology, the design of embedded systems is an essential element of many innovations. Embedded systems meet their performance goals, including real-time constraints, through a combination of special-purpose hardware and software components tailored to the system requirements. Both the development of new features and the reuse of existing intellectual property components are essential to keeping up with ever more demanding customer requirements. Furthermore, design complexities are steadily growing with an increasing number of components that have to cooperate properly. Embedded system

designers have to cope with multiple goals and constraints simul- neously, including timing, power, reliability, dependability, maintenance, packaging and, last but not least, price.

This book constitutes the proceedings of the 11th International Symposium on NASA Formal Methods, NFM 2019, held in Houston, TX, USA, in May 2019. The 20 full and 8 short papers presented in this volume were carefully reviewed and selected from 102 submissions. The papers focus on formal verification, including theorem proving, model checking, and static analysis; advances in automated theorem proving including SAT and SMT solving; use of formal methods in software and system testing; run-time verification; techniques and algorithms for scaling formal methods, such as abstraction and symbolic methods, compositional techniques, as well as parallel and/or distributed techniques; code generation from formally verified models; safety cases and system safety; formal approaches to fault tolerance; theoretical advances and empirical evaluations of formal methods techniques for safety-critical systems, including hybrid and embedded systems; formal methods in systems engineering and model-based development; correct-by-design controller synthesis; formal assurance methods to handle adaptive systems.

The Handbook of Logic in Artificial Intelligence and Logic Programming is a multi-volume work covering all major areas of the application of logic to artificial intelligence and logic programming. The authors are chosen on an international basis and are leaders in the fields covered. Volume 5 is the last in this well-regarded series. Logic is now widely recognized as one of the foundational disciplines of computing. It has found applications in virtually all aspects of the subject, from software and hardware engineering to programming languages and artificial intelligence. In response to the growing need for an in-depth survey of these

applications the Handbook of Logic in Artificial Intelligence and its companion, the Handbook of Logic in Computer Science have been created. The Handbooks are a combination of authoritative exposition, comprehensive survey, and fundamental research exploring the underlying themes in the various areas. Some mathematical background is assumed, and much of the material will be of interest to logicians and mathematicians. Volume 5 focuses particularly on logic programming. The chapters, which in many cases are of monograph length and scope, emphasize possible unifying themes.

Applied Formal VerificationFor Digital Circuit DesignMcGraw Hill Professional

One of the biggest challenges in chip and system design is determining whether the hardware works correctly. That is the job of functional verification engineers and they are the audience for this comprehensive text from three top industry professionals. As designs increase in complexity, so has the value of verification engineers within the hardware design team. In fact, the need for skilled verification engineers has grown dramatically--functional verification now consumes between 40 and 70% of a project's labor, and about half its cost. Currently there are very few books on verification for engineers, and none that cover the subject as comprehensively as this text. A key strength of this book is that it describes the entire verification cycle and details each stage. The organization of the book follows the cycle, demonstrating how functional verification engages all aspects of the overall design effort and how individual cycle stages relate to the larger design process. Throughout the text, the authors leverage their 35 plus years experience in functional verification, providing examples and case

studies, and focusing on the skills, methods, and tools needed to complete each verification task. Comprehensive overview of the complete verification cycle Combines industry experience with a strong emphasis on functional verification fundamentals Includes real-world case studies

Advanced Formal Verification shows the latest developments in the verification domain from the perspectives of the user and the developer. World leading experts describe the underlying methods of today's verification tools and describe various scenarios from industrial practice. In the first part of the book the core techniques of today's formal verification tools, such as SAT and BDDs are addressed. In addition, multipliers, which are known to be difficult, are studied. The second part gives insight in professional tools and the underlying methodology, such as property checking and assertion based verification. Finally, analog components have to be considered to cope with complete system on chip designs.

Circuits and architectures have become more complex in terms of structure, interconnection topology, and data flow. Design correctness has become increasingly significant, as errors in design may result in strenuous debugging, or even in the repetition of a costly manufacturing process. Although circuit simulation has been used traditionally and widely as the technique for checking hardware and architectural designs, it does not guarantee the conformity of designs to specifications. Formal methods therefore become vital in guaranteeing the correctness of designs and have

thus received a significant amount of attention in the CAD industry today.This book presents a formal method for specifying and verifying the correctness of systolic array designs. Such architectures are commonly found in the form of accelerators for digital signal, image, and video processing. These arrays can be quite complicated in topology and data flow. In the book, a formalism called STA is defined for these kinds of dynamic environments, with a survey of related techniques. A framework for specification and verification is established. Formal verification techniques to check the correctness of the systolic networks with respect to the algorithmic level specifications are explained. The book also presents a Prolog-based formal design verifier (named VSTA), developed to automate the verification process, as using a general purpose theorem prover is usually extremely time-consuming. Several application examples are included in the book to illustrate how formal techniques and the verifier can be used to automate proofs.

Functional Design Errors in Digital Circuits Diagnosis covers a wide spectrum of innovative methods to automate the debugging process throughout the design flow: from Register-Transfer Level (RTL) all the way to the silicon die. In particular, this book describes: (1) techniques for bug trace minimization that simplify debugging; (2) an RTL error diagnosis method that identifies the root cause of errors directly; (3) a counterexample-guided error-repair framework to automatically fix errors in gate-level and RTL designs; (4) a symmetry-based rewiring technology for fixing electrical errors;

(5) an incremental verification system for physical synthesis; and (6) an integrated framework for post-silicon debugging and layout repair. The solutions provided in this book can greatly reduce debugging effort, enhance design quality, and ultimately enable the design and manufacture of more reliable electronic devices.
CHARME'99 is the tenth in a series of working conferences devoted to the dev- opment and use of leading-edge formal techniques and tools for the design and veri?cation of hardware and systems. Previous conferences have been held in Darmstadt (1984), Edinburgh (1985), Grenoble (1986), Glasgow (1988), Leuven (1989), Torino (1991), Arles (1993), Frankfurt (1995) and Montreal (1997). This workshop and conference series has been organized in cooperation with IFIP WG 10. 5. It is now the biannual counterpart of FMCAD, which takes place every even-numbered year in the USA. The 1999 event took place in Bad Her- nalb, a resort village located in the Black Forest close to the city of Karlsruhe. The validation of functional and timing behavior is a major bottleneck in current VLSI design systems. A predominantly academic area of study until a few years ago, formal design and veri?cation techniques are now migrating into industrial use. The aim of CHARME'99 is to bring together researchers and users from academia and industry working in this active area of research. Two invited talks illustrate major current trends: the presentation by G ?erard Berry (Ecole des Mines de Paris, Sophia-Antipolis, France) is concerned with the use of synchronous languages in circuit design, and the talk given by Peter Jansen (BMW, Munich, Germany)

demonstrates an application of formal methods in an industrial environment. The program also includes 20 regular presentations and 12 short presentations/poster exhibitions that have been selected from the 48 submitted papers.

Formal Verification: An Essential Toolkit for Modern VLSI Design presents practical approaches for design and validation, with hands-on advice to help working engineers integrate these techniques into their work. Formal Verification (FV) enables a designer to directly analyze and mathematically explore the quality or other aspects of a Register Transfer Level (RTL) design without using simulations. This can reduce time spent validating designs and more quickly reach a final design for manufacturing. Building on a basic knowledge of SystemVerilog, this book demystifies FV and presents the practical applications that are bringing it into mainstream design and validation processes at Intel and other companies. After reading this book, readers will be prepared to introduce FV in their organization and effectively deploy FV techniques to increase design and validation productivity. Learn formal verification algorithms to gain full coverage without exhaustive simulation Understand formal verification tools and how they differ from simulation tools Create instant test benches to gain insight into how models work and find initial bugs Learn from Intel insiders sharing their hard-won knowledge and solutions to complex design problems

Now available in a three-volume set, this updated and expanded edition of the bestselling Digital Signal Processing Handbook continues to provide the engineering

community with authoritative coverage of the fundamental and specialized aspects of information-bearing signals in digital form. Encompassing essential background material, technical details, standards, and software, The Digital Signal Processing Handbook, Second Edition reflects cutting-edge information on signal processing algorithms and protocols related to speech, audio, multimedia, and video processing technology associated with standards ranging from WiMax to MP3 audio, low-power/high-performance DSPs, color image processing, and chips on video. The three-volume set draws on the experience of leading engineers, researchers, and scholars and includes 29 new chapters that address multimedia and Internet technologies, tomography, radar systems, architecture, standards, and future applications in speech, acoustics, video, radar, and telecommunications. Each volume in the set is also available individually ... Emphasizing theoretical concepts, Digital Signal Processing Fundamentals (Catalog no. 46063) provides comprehensive coverage of the basic foundations of DSP. Coverage includes: Signals and Systems, Signal Representation and Quantization, Fourier Transforms, Digital Filtering, Statistical Signal Processing, Adaptive Filtering, Inverse Problems and Signal Reconstruction, and Time–Frequency and Multirate Signal Processing. Wireless, Networking, Radar, Sensor Array Processing, and Nonlinear Signal Processing (Catalog no. 46047) thoroughly covers the foundations of signal processing related to wireless, radar, space–time coding, and mobile communications together with associated applications to networking, storage,

and communications. Video, Speech, and Audio Signal Processing and Associated Standards, (Catalog no. 4608X) details the basic foundations of speech, audio, image, and video processing and associated applications to broadcast, storage, search and retrieval, and communications.

Avionics provide crews and passengers with an array of capabilities. Cockpit crews can operate with fewer pilots, greater efficiency, and immediate critical information. Passengers can enjoy the ultimate in inflight entertainment: live television and audio broadcasts and access to the Internet and e-mail. Since avionics are the among most ex

Formal verification is a powerful new digital design method. In this cutting-edge tutorial, two of the field's best known authors team up to show designers how to efficiently apply Formal Verification, along with hardware description languages like Verilog and VHDL, to more efficiently solve real-world design problems. Contents: Simulation-Based Verification * Introduction to Formal Techniques * Contrasting Simulation vs. Formal Techniques * Developing a Formal Test Plan * Writing High-Level Requirements * Proving High-Level Requirements * System Level Simulation * Design Example * Formal Test Plan * Final System Simulation

This book grew out of a NATO Advanced Study Institute summer school that was held in Antalya, TUrkey from 26 May to 6 June 1997. The purpose of the summer school was to expose recent advances in the formal verification of systems composed of both

logical and continuous time components. The course was structured in two parts. The first part covered theorem-proving, system automaton models, logics, tools, and complexity of verification. The second part covered modeling and verification of hybrid systems, i. e. , systems composed of a discrete event part and a continuous time part that interact with each other in novel ways. Along with advances in microelectronics, methods to design and build logical systems have grown progressively complex. One way to tackle the problem of ensuring the error-free operation of digital or hybrid systems is through the use of formal techniques. The exercise of comparing the formal specification of a logical system namely, what it is supposed to do to its formal operational description-what it actually does!-in an automated or semi-automated manner is called verification. Verification can be performed in an after-the-fact manner, meaning that after a system is already designed, its specification and operational description are regenerated or modified, if necessary, to match the verification tool at hand and the consistency check is carried out.

A presentation of real examples of industrial uses for formalmethods such as SCADE, the B-Method, ControlBuild, Matelo, etc. invarious fields, such as railways, aeronautics, and the automotiveindustry, the purpose of this book is to present a summary ofexperience on the use of these “formal methods” (suchas proof and model-checking) in industrial examples of complexsystems. It is based on the experience of people who are currently involvedin the creation and evaluation of safety critical system

software.The involvement of people from within the industry allows us toavoid the usual problems of confidentiality which could arise andthus enables us to supply new useful information (photos,architecture plans, real examples, etc.).

This book is a solid foundation of the most important formalisms used for specification and verification of reactive systems. In particular, the text presents all important results on m-calculus, w-automata, and temporal logics, shows the relationships between these formalisms and describes state-of-the-art verification procedures for them. It also discusses advantages and disadvantages of these formalisms, and shows up their strengths and weaknesses. Most results are given with detailed proofs, so that the presentation is almost self-contained. Includes all definitions without relying on other material Proves all theorems in detail Presents detailed algorithms in pseudo-code for verification as well as translations to other formalisms

This volume gives the proceedings of the Fourth Workshop on Computer-Aided Verification (CAV '92), held in Montreal, June 29 - July 1, 1992. The objective of this series of workshops is to bring together researchers and practitioners interested in the development and use of methods, tools and theories for the computer-aided verification of concurrent systems. The workshops provide an opportunity for comparing various verification methods and practical tools that can be used to assist the applications designer. Emphasis is placed on new research results and the application of existing results to real verification problems. The volume contains 31 papers selected from 75

submissions. These are organized into parts on reduction techniques, proof checking, symbolic verification, timing verification, partial-order approaches, case studies, model and proof checking, and other approaches. The volume starts with an invited lecture by Leslie Lamport entitled "Computer-hindered verification (humans can do it too)".

This book constitutes the refereed proceedings of the 19th International Conference on Computer Safety, Reliability, and Security, SAFECOMP 2000, held in Rotterdam, The Netherlands in October 2000.The 33 revised full papers presented together with three invited papers were carefully reviewed and selected for inclusion in the book. The papers are organized in topical sections on verification and validation; software process improvement; formal methods; safety guidelines, standards and certification; hardware aspects; safety assessment; design for safety; and transport and infrastructure.

A handbook to the Coq software for writing and checking mathematical proofs, with a practical engineering focus. The technology of mechanized program verification can play a supporting role in many kinds of research projects in computer science, and related tools for formal proof-checking are seeing increasing adoption in mathematics and engineering. This book provides an introduction to the Coq software for writing and checking mathematical proofs. It takes a practical engineering focus throughout, emphasizing techniques that will help users to build, understand, and maintain large Coq developments and minimize the cost of code change over time. Two topics, rarely discussed elsewhere, are covered in detail: effective dependently typed programming

(making productive use of a feature at the heart of the Coq system) and construction of domain-specific proof tactics. Almost every subject covered is also relevant to interactive computer theorem proving in general, not just program verification, demonstrated through examples of verified programs applied in many different sorts of formalizations. The book develops a unique automated proof style and applies it throughout; even experienced Coq users may benefit from reading about basic Coq concepts from this novel perspective. The book also offers a library of tactics, or programs that find proofs, designed for use with examples in the book. Readers will acquire the necessary skills to reimplement these tactics in other settings by the end of the book. All of the code appearing in the book is freely available online.

This book provides readers with a comprehensive introduction to the formal verification of hardware and software. World-leading experts from the domain of formal proof techniques show the latest developments starting from electronic system level (ESL) descriptions down to the register transfer level (RTL). The authors demonstrate at different abstraction layers how formal methods can help to ensure functional correctness. Coverage includes the latest academic research results, as well as descriptions of industrial tools and case studies.

Integrated circuit capacity follows Moore's law, and chips are commonly produced at the time of this writing with over 70 million gates per device. Ensuring correct functional behavior of such large designs before fabrication poses an extremely challenging

problem. Formal verification validates the correctness of the implementation of a design with respect to its specification through mathematical proof techniques. Formal techniques have been emerging as commercialized EDA tools in the past decade. Simulation remains a predominantly used tool to validate a design in industry. After more than 50 years of development, simulation methods have reached a degree of maturity, however, new advances continue to be developed in the area. A simulation approach for functional verification can theoretically validate all possible behaviors of a design but requires excessive computational resources. Rapidly evolving markets demand short design cycles while the increasing complexity of a design causes simulation approaches to provide less and less coverage. Formal verification is an attractive alternative since 100% coverage can be achieved; however, large designs impose unrealistic computational requirements. Combining formal verification and simulation into a single integrated circuit validation framework is an attractive alternative. This book focuses on an Integrated Design Validation (IDV) system that provides a framework for design validation and takes advantage of current technology in the areas of simulation and formal verification resulting in a practical validation engine with reasonable runtime. After surveying the basic principles of formal verification and simulation, this book describes the IDV approach to integrated circuit functional validation. Table of Contents: Introduction / Formal Methods Background / Simulation Approaches / Integrated Design Validation System / Conclusion and

Summary

This state-of-the-art monograph presents a coherent survey of a variety of methods and systems for formal hardware verification. It emphasizes the presentation of approaches that have matured into tools and systems usable for the actual verification of nontrivial circuits. All in all, the book is a representative and well-structured survey on the success and future potential of formal methods in proving the correctness of circuits. The various chapters describe the respective approaches supplying theoretical foundations as well as taking into account the application viewpoint. By applying all methods and systems presented to the same set of IFIP WG10.5 hardware verification examples, a valuable and fair analysis of the strenghts and weaknesses of the various approaches is given.

Formal verification is a powerful new digital design method In this cutting-edge tutorial, two of the field's best known authors team up to show designers how to efficiently apply Formal Verification, along with hardware description languages like Verilog and VHDL, to more efficiently solve real-world design problems.

Given the growing size and heterogeneity of Systems on Chip (SOC), the design process from initial specification to chip fabrication has become increasingly complex. This growing complexity provides incentive for designers to use high-level languages such as C, SystemC, and SystemVerilog for system-level design. While a major goal of these high-level languages is to enable verification at a higher level of abstraction,

allowing early exploration of system-level designs, the focus so far for validation purposes has been on traditional testing techniques such as random testing and scenario-based testing. This book focuses on high-level verification, presenting a design methodology that relies upon advances in synthesis techniques as well as on incremental refinement of the design process. These refinements can be done manually or through elaboration tools. This book discusses verification of specific properties in designs written using high-level languages, as well as checking that the refined implementations are equivalent to their high-level specifications. The novelty of each of these techniques is that they use a combination of formal techniques to do scalable verification of system designs completely automatically. The verification techniques presented in this book include methods for verifying properties of high-level designs and methods for verifying that the translation from high-level design to a low-level Register Transfer Language (RTL) design preserves semantics. Used together, these techniques guarantee that properties verified in the high-level design are preserved through the translation to low-level RTL.

This book constitutes the proceedings of the 26th International Workshop on Formal Methods for Industrial Critical Systems, FMICS 2021, which was held during August 24-26, 2021. The conference was planned to take place in Pairs, France. Due to the COVID-19 pandemic it changed to a virtual event. The 10 full papers and 6 short papers presented in this volume were carefully reviewed and selected from 31

submissions. The papers are organized in topical sections as follows: Verification, Program Safety and Education, (Event-)B Modeling and Validation, Formal Analysis, Tools, Test Generation and Probabilistic Verification.

This book constitutes the refereed proceedings of the Second International Conference on E-learning and Games, Edutainment 2007, held in Hong Kong, China, in June 2007. It covers virtual and augmented reality in game and education, virtual characters in games and education, e-learning platforms and tools, geometry in games and virtual reality, vision, imaging and video technology, as well as collaborative and distributed environments.

There is much excitement in the design and verification community about assertion-based design. The question is, who should study assertion-based design? The emphatic answer is, both design and verification engineers. What may be unintuitive to many design engineers is that adding assertions to RTL code will actually reduce design time, while better documenting design intent. Every design engineer should read this book! Design engineers that add assertions to their design will not only reduce the time needed to complete a design, they will also reduce the number of interruptions from verification engineers to answer questions about design intent and to address verification suite mistakes. With design assertions in place, the majority of the interruptions from verification engineers will be related to actual design problems and the error feedback provided will be more useful to help identify design flaws. A design

engineer who does not add assertions to the RTL code will spend more time with verification engineers explaining the design functionality and intended interface requirements, knowledge that is needed by the verification engineer to complete the job of testing the design.

This volume contains the proceedings of the second workshop on Computer Aided Verification, held at DIMACS, Rutgers University, June 18-21, 1990. Itfeatures theoretical results that lead to new or more powerful verification methods. Among these are advances in the use of binary decision diagrams, dense time, reductions based upon partial order representations and proof-checking in controller verification. The motivation for holding a workshop on computer aided verification was to bring together work on effective algorithms or methodologies for formal verification - as distinguished, say,from attributes of logics or formal languages. The considerable interest generated by the first workshop, held in Grenoble, June 1989 (see LNCS 407), prompted this second meeting. The general focus of this volume is on the problem of making formal verification feasible for various models of computation. Specific emphasis is on models associated with distributed programs, protocols, and digital circuits. The general test of algorithm feasibility is to embed it into a verification tool, and exercise that tool on realistic examples: the workshop included sessionsfor the demonstration of new verification tools.

The second of two volumes in the Electronic Design Automation for Integrated Circuits Handbook, Second Edition, Electronic Design Automation for IC Implementation, Circuit Design, and Process Technology thoroughly examines real-time logic (RTL) to GDSII (a file format used to transfer data of semiconductor physical layout) design flow, analog/mixed signal

design, physical verification, and technology computer-aided design (TCAD). Chapters contributed by leading experts authoritatively discuss design for manufacturability (DFM) at the nanoscale, power supply network design and analysis, design modeling, 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 3D circuit integration and clock design Offering improved depth and modernity, Electronic Design Automation for IC Implementation, Circuit Design, and Process Technology provides a valuable, state-of-the-art reference for electronic design automation (EDA) students, researchers, and professionals.

The HOL system is a higher order logic theorem proving system implemented at Edinburgh University, Cambridge University and INRIA. Its many applications, from the verification of hardware designs at all levels to the verification of programs and communication protocols are considered in depth in this volume. Other systems based on higher order logic, namely Nuprl and LAMBDA are also discussed. Features given particular consideration are: novel developments in higher order logic and its implementations in HOL; formal design and verification methodologies for hardware and software; public domain availability of the HOL system. Papers addressing these issues have been divided as follows: Mathematical Logic; Induction; General Modelling and Proofs; Formalizing and Modelling of Automata; Program

Verification; Hardware Description Language Semantics; Hardware Verification Methodologies; Simulation in Higher Order Logic; Extended Uses of Higher Order Logic. Academic and industrial researchers involved in formal hardware and software design and verification methods should find the publication especially interesting and it is hoped it will also provide a useful reference tool for those working at software institutes and within the electronics industries.

"This book explores different applications in V & V that spawn many areas of software development -including real time applications- where V & V techniques are required, providing in all cases examples of the applications"--Provided by publisher.

This book constitutes the refereed proceedings of the 11th International Conference on Theorem Proving in Higher Order Logics, TPHOLs '98, held in Canberra, Australia, in September/October 1998. The 26 revised full papers presented were carefully reviewed and selected from a total of 52 submissions. Also included are two invited papers. The papers address all current aspects of theorem proving in higher order logics and formal verification and program analysis. Besides the HOL system, the theorem provers Coq, Isabelle, LAMBDA, LEGO, NuPrl, and PVS are discussed.