Advanced Design Practical Examples Verilog

This book is designed specifically to make the cutting-edge techniques of digital hardware design more accessible to those just entering the field. The text uses a simpler language (Verilog) and standardizes the methodology to the point where even novices can get medium complex designs through to gate-level simulation in a short period of time. Requires a working knowledge of computer organization, Unix, and X windows. Some knowledge of a programming language such as C or Java is desirable, but not necessary. Features a large number of worked examples and problems--from 100 to 100k gate equivalents--all synthesized and successfully verified by simulation at gate level using the VCS compiled simulator, the FPGA Compiler and Behavioral Compiler available from Synopsys, and the FPGA tool suites from Altera and Xilinx. Basic Language Constructs. Structural and Behavioral Specification. Simulation. Procedural Specification. Design Approaches for Single Modules. Validation of Single Modules. Finite State Machine Styles. Control-Point Writing Style. Managing Complexity--Large Designs. Improving Timing, Area, and Power. Design Compiler. Synthesis to Standard Cells. Synthesis to FPGA. Gate Level Simulation and Testing. Alternative Writing Styles. Mixed Technology Design. For anyone wanting an accessible, accelerated introduction to the cutting-edge tools for Digital Hardware Design.

This book describes RTL design using Verilog, synthesis and timing closure for System On Chip (SOC) design blocks. It covers the complex RTL design scenarios and challenges for SOC designs and provides practical information on performance improvements in SOC, as well as Application Specific Integrated Circuit (ASIC) designs. Prototyping using modern high density Field Programmable Gate Arrays (FPGAs) is discussed in this book with the practical examples and case studies. The book discusses SOC design, performance improvement techniques, testing and system level verification, while also describing the modern Intel FPGA/XILINX FPGA architectures and their use in SOC prototyping. Further, the book covers the Synopsys Design Compiler (DC) and Prime Time (PT) commands, and how they can be used to optimize complex ASIC/SOC designs. The contents of this book will be useful to students and professionals alike.

This book provides the most up-to-date coverage using the Synopsys program in the design of integrated circuits. The incorporation of "synthesis tools" is the most popular new method of designing integrated circuits for higher speeds covering smaller surface areas. Synopsys is the dominant computer-aided circuit design program in the world. All of the major circuit manufacturers and ASIC design firms use Synopsys. In addition, Synopsys is used in teaching and laboratories at over 600 universities. First practical guide to using synthesis with Synopsys Synopsys is the #1 design program for IC design

Starts with an overview of today's FPGA technology, devices, and tools for designing state-of-the-art DSP systems. A case study in the first chapter is the basis for more than 30 design examples throughout. The following chapters deal with computer arithmetic concepts, theory and the implementation of FIR and IIR filters, multirate digital signal processing systems, DFT and FFT algorithms, and advanced algorithms with high future potential. Each chapter contains exercises. The VERILOG source code and a glossary are given in the appendices, while the accompanying CD-ROM contains the examples in VHDL and Verilog code as well as the newest Altera "Baseline" software. This edition has a new chapter on adaptive filters, new sections on division and floating point arithmetics, an up-date to the current Altera software, and some new exercises.

Practical Examples in Verilog

Architecture, Implementation, and Optimization

HDL Chip Design

Advanced FPGA Design

Digital Systems Design and Practice

Using Verilog HDL and FPGAs

Explores the unique hardware programmability of FPGA-based embedded systems, using a learn-by-doing approach to introduce the concepts and techniques for embedded SoPC design with Verilog An SoPC (system on a programmable chip) integrates a processor, memory modules, I/O peripherals, and custom hardware accelerators into a single FPGA (field-programmable gate array) device. In addition to the customized software, customized hardware can be developed and incorporated into the embedded system as well—allowing us to configure the soft-core processor, create tailored I/O interfaces, and develop specialized hardware accelerators for computation-intensive tasks. Utilizing an Altera FPGA prototyping board and its Nios II soft-core processor, Embedded SoPC Design with Nios II Processor and Verilog Examples takes a "learn by doing" approach to illustrate the hardware and software design and development process by including realistic projects that can be implemented and tested on the board. Emphasizing hardware design and integration throughout, the book is divided into four major parts: Part I covers HDL and synthesis of custom hardware Part II introduces the Nios II processor and provides an overview of embedded software development Part III demonstrates the design and development of hardware and software of several complex I/O peripherals, including a PS2 keyboard and mouse, a graphic video controller, an audio codec, and an SD (secure digital) card Part IV provides several case studies of the integration of hardware accelerators, including a custom GCD (greatest common divisor) circuit, a Mandelbrot set fractal circuit, and an audio synthesizer based on DDFS (direct digital frequency synthesis) methodology While designing and developing an embedded SoPC can be rewarding, the learning can be a long and winding journey. This book shows the trail ahead and guides readers through the initial steps to exploit the full potential of this emerging methodology.

This book provides the advanced issues of FPGA design as the underlying theme of the work. In practice, an engineer typically needs to be mentored for several years before these principles are appropriately utilized. The topics that will be discussed in this book are essential to designing FPGA's beyond moderate complexity. The goal of the book is to present practical design techniques that are otherwise only available through mentorship and real-world experience.

System-on-a-chip (SoC) has become an essential technique to lower product costs and maximize power efficiency, particularly as the mobility and size requirements of electronics continues to grow. It has therefore become increasingly important for electrical engineers to develop a strong understanding of the key stages of hardware description language (HDL) design flow based on cell-based libraries or field-programmable gate array (FPGA) devices. Honed and revised through years of classroom use, Lin focuses on developing, verifying, and synthesizing designs of practical digital systems using the most widely used hardware description Language: Verilog HDL. Explains how to perform synthesis and verification to achieve optimized synthesis results and compiler times Offers complete coverage of Verilog syntax Illustrates the entire design and verification flow using an FPGA case study Presents real-world design examples such as LED and LCD displays, GPIO, UART, timers, and CPUs Emphasizes design/implementation tradeoff options, with coverage of ASICs and FPGAs Provides an introduction to design for testability Gives readers deeper understanding by using problems and review questions in each chapter Comes with downloadable Verilog HDL source code for most examples in the text Includes presentation slides of all book figures for student reference Digital System Designs and Practices Using Verilog HDL and FPGAs is an ideal textbook for either fundamental or advanced digital design courses beyond the digital logic design level. Design engineers who want to become more proficient users of Verilog HDL as well as design FPGAs with greater speed and accuracy will find this book indispensable.

DIGITAL SYSTEMS DESIGN USING VERILOG integrates coverage of logic design principles, Verilog as a hardware design language, and FPGA implementation to help electrical and computer engineering students master the process of designing and testing new hardware configurations. A Verilog equivalent of authors Roth and John's previous successful text using VHDL, this practical book presents Verilog constructs side-by-side with hardware, encouraging students to think in terms of desired hardware while writing synthesizable Verilog. Following a review of the basic concepts of logic design, the authors introduce the basics of Verilog using simple combinational circuit examples, followed by models for simple sequential circuits. Subsequent chapters ask readers to tackle more and more complex designs. Important Notice: Media content referenced within the product description or the product text may not be available in the ebook version.

A Concise Introduction for FPGA Design

Using Verilog, State Machines, and Synthesis for FPGAs

Verilog Styles for Synthesis of Digital Systems

Design Recipes for FPGAs: Using Verilog and VHDL

Advanced Digital Design with the Verilog HDL

Digital System Design with FPGA: Implementation Using Verilog and VHDL

With the advance of semiconductor and communication technologies, the use of systemon-a-chip (SoC) has become an essential technique to decrease product costs. To design and implement an SoC-based product, it proves necessary to totally or partly rely on the hardware description language (HDL) synthesis flow and field programmable gata array (FPGA) devices or cell libraries. As a consequence, it has become an important attainment for electrical engineers to develop a good understanding of the key issues of HDL design flows based on FPGA devices or cell libraries. To achieve this, this book addresses the need for teaching such a topic based on Verilog HDL and FPGAs. This book, Digital System Designs and Practices: Using Verilog HDL and FPGAs, aim to be used as a text for students and as a reference book for professionals or a self-study book for readers. For classroom use, each chapter includes many worked examples and review questions for helping readers test their understanding of the contents. In addition, throughout the book, an abundance of worked examples are provided for helping readers realize the basic features of Verilog HDL and grasp the essentials of digital system designs as well. The contents of this book largely stem from the course FPGA System Designs and Practices, offered at our campus over the past decade. This course is an undergraduate elective and the first-year graduate course. This book is so structured that it can be used as a sequence of courses, including Hardware Description Language, FPGA System Designs and Practices, Digital System Designs, Advanced Digital System Designs, and others. HDL-based design has become an essential technique for modern digital systems. This book focuses on developing, verifying, and synthesizing designs of practical digital systems using the most widely used hardware description Language: Verilog HDL and FPGAs. The main features of this book are: -- Explains how to perform synthesis and verification to achieve optimized synthesis results and compiler times -- Offers complete coverage of Verilog HDL syntax -- Illustrates the entire design and verification flow using an FPGA case study -- Presents many real-world worked design examples -- Gives readers deeper understanding with review questions in each section and end-of-chapter problems -- Emphasizes design/implementation tradeoff options, with coverage of ASICs and FPGAs

Start by walking a typical Verilog design all the way through to silicon; then, review basic Verilog syntax, design, simulation and testing, advanced simulation, and more."--BOOK JACKET.

This textbook is intended to serve as a practical guide for the design of complex digital logic circuits such as digital control circuits, network interface circuits, pipelined arithmetic units, and RISC microprocessors. It is an advanced digital logic design textbook that emphasizes the use of synthesizable Verilog code and provides numerous fully worked-out practical design examples including a Universal Serial Bus interface, a pipelined multiply-accumulate unit, and a pipelined microprocessor for the ARM THUMB architecture.

The book is divided into four major parts. Part I covers HDL constructs and synthesis of basic digital circuits. Part II provides an overview of embedded software development with the emphasis on low–level I/O access and drivers. Part III demonstrates the design and development of hardware and software for several complex I/O peripherals, including PS2 keyboard and mouse, a graphic video controller, an audio codec, and an SD (secure digital) card. Part IV provides three case studies of the integration of hardware accelerators, including a custom GCD (greatest common divisor) circuit, a Mandelbrot set fractal circuit, and an audio synthesizer based on DDFS (direct digital frequency synthesis) methodology. The book utilizes FPGA devices, Nios II soft–core processor, and development platform from Altera Co., which is one of the two main FPGA manufactures. Altera has a generous university program that provides free software and discounted prototyping boards for educational institutions (details at http://www.altera.com/university). The two main educational

prototyping boards are known as DE1 ($99) and DE2 ($269). All experiments can be implemented and tested with these boards. A board combined with this book becomes a "turn–key" solution for the SoPC design experiments and projects. Most HDL and C codes in the book are device independent and can be adapted by other prototyping boards as long as a board has similar I/O configuration.

A Guide to Digital Design and Synthesis
VHDL Coding and Logic Synthesis with Synopsys
Digital Design (Verilog)
Design Through Verilog HDL
Xilinx MicroBlaze MCS SoC Edition
Real World FPGA Design with Verilog

Digital Design: An Embedded Systems Approach Using Verilog provides a foundation in digital design for students in computer engineering, electrical engineering and computer science courses. It takes an up-to-date and modern approach of presenting digital logic design as an activity in a larger systems design context. Rather than focus on aspects of digital design that have little relevance in a realistic design context, this book concentrates on modern and evolving knowledge and design skills. Hardware description language (HDL)-based design and verification is emphasized--Verilog examples are used extensively throughout. By treating digital logic as part of embedded systems design, this book provides an understanding of the hardware needed in the analysis and design of systems comprising both hardware and software components. Includes a Web site with links to vendor tools, labs and tutorials. Presents digital logic design as an activity in a larger systems design context Features extensive use of Verilog examples to demonstrate HDL (hardware description language) usage at the abstract behavioural level and register transfer level, as well as for low-level verification and verification environments Includes worked examples throughout to enhance the reader's understanding and retention of the material Companion Web site includes links to tools for FPGA design from Synplicity, Mentor Graphics, and Xilinx, Verilog source code for all the examples in the book, lecture slides, laboratory projects, and solutions to exercises
Master FPGA digital system design and implementation with Verilog and VHDL This practical guide explores the development and deployment of FPGA-based digital systems using the two most popular hardware description languages, Verilog and VHDL. Written by a pair of digital circuit design experts, the book offers a solid grounding in FPGA principles, practices, and applications and provides an overview of more complex topics. Important concepts are demonstrated through real-world examples, ready-to-run code, and inexpensive start-to-finish projects for both the Basys and Arty boards. Digital System Design with FPGA: Implementation Using Verilog and VHDL covers: • Field programmable gate array fundamentals • Basys and Arty FPGA boards • The Vivado design suite • Verilog and VHDL • Data types and operators • Combinational circuits and circuit blocks • Data storage elements and sequential circuits • Soft-core microcontroller and digital interfacing • Advanced FPGA applications • The future of FPGA
This book describes simple to complex ASIC design practical scenarios using Verilog. It builds a story from the basic fundamentals of ASIC designs to advanced RTL design concepts using Verilog. Looking at current trends of miniaturization, the contents provide practical information on the issues in ASIC design and synthesis using Synopsys DC and their solution. The book explains how to write efficient RTL using Verilog and how to improve design performance. It also covers architecture design strategies, multiple clock domain designs, low-power design techniques, DFT, pre-layout STA and the overall ASIC design flow with case studies. The contents of this book will be useful to practicing hardware engineers, students, and hobbyists looking to learn about ASIC design and synthesis.
SystemVerilog is a Hardware Description Language that enables designers to work at the higher levels of logic design abstractions that match the increased complexity of current day integrated circuit and field-programmable gate array (FPGA) designs. The majority of the book assumes a basic background in logic design and software programming concepts. It is directed at: * students currently in an introductory logic design course that also teaches SystemVerilog, * designers who want to update their skills from Verilog or VHDL, and * students in VLSI design and advanced logic design courses that include verification as well as design topics. The book starts with a tutorial introduction on hardware description languages and simulation. It proceeds to the register-transfer design topics of combinational and finite state machine (FSM) design - these mirror the topics of introductory logic design courses. The book covers the design of FSM-datapath designs and their interfaces, including SystemVerilog interfaces. Then it covers the more advanced topics of writing testbenches including using assertions and functional coverage. A comprehensive index provides easy access to the book's topics.The goal of the book is to introduce the broad spectrum of features in the language in a way that complements introductory and advanced logic design and verification courses, and then provides a basis for further learning.Solutions to problems at the end of

chapters, and text copies of the SystemVerilog examples are available from the author as described in the Preface.

Embedded SoPC Design with Nios II Processor and Verilog Examples

Digital Systems Design Using Verilog

ASIC Design and Synthesis

Using Verilog and VHDL

Logic Design and Verification Using SystemVerilog (Revised)

Printed Circuit Board Designer's Reference

A hands-on introduction to FPGA prototyping and SoC design This is the successor edition of the popular FPGA Prototyping by Verilog Examples text. It follows the same "learning-by-doing" approach to teach the fundamentals and practices of HDL synthesis and FPGA prototyping. The new edition uses a coherent series of examples to demonstrate the process to develop sophisticated digital circuits and IP (intellectual property) cores, integrate them into an SoC (system on a chip) framework, realize the system on an FPGA prototyping board, and verify the hardware and software operation. The examples start with simple gate-level circuits, progress gradually through the RT (register transfer) level modules, and lead to a functional embedded system with custom I/O peripherals and hardware accelerators. Although it is an introductory text, the examples are developed in a rigorous manner, and the derivations follow the strict design guidelines and coding practices used for large, complex digital systems. The book is completely updated and uses the SystemVerilog language, which "absorbs" the Verilog language. It presents the hardware design in the SoC context and introduces the hardware-software co-design concept. Instead of treating examples as isolated entities, the book integrates them into a single coherent SoC platform that allows readers to explore both hardware and software "programmability" and develop complex and interesting embedded system projects. The new edition: Adds four general-purpose IP cores, which are multi-channel PWM (pulse width modulation) controller, I2C controller, SPI controller, and XADC (Xilinx analog-to-digital converter) controller. Introduces a music synthesizer constructed with a DDFS (direct digital frequency synthesis) module and an ADSR (attack-decay-sustain-release) envelope generator. Expands the original video controller into a complete stream based video subsystem that incorporates a video synchronization circuit, a test-pattern generator, an OSD (on-screen display) controller, a sprite generator, and a frame buffer. Provides a detailed discussion on blocking and nonblocking statements and coding styles. Describes basic concepts of software-hardware co-design with Xilinx MicroBlaze MCS soft-core processor. Provides an overview of bus interconnect and interface circuit. Presents basic embedded system software development. Suggests additional modules and peripherals for interesting and challenging projects. FPGA Prototyping by SystemVerilog Examples makes a natural companion text for introductory and advanced digital design courses and embedded system courses. It also serves as an ideal self-teaching guide for practicing engineers who wish to learn more about this emerging area of interest.

This book is designed to serve as a hands-on professional reference with additional utility as a textbook for upper undergraduate and some graduate courses in digital logic design. This book is organized in such a way that that it can describe a number of RTL design scenarios, from simple to complex. The book constructs the logic design story from the fundamentals of logic design to advanced RTL design concepts. Keeping in view the importance of miniaturization today, the book gives practical information on the issues with ASIC RTL design and how to overcome these concerns. It clearly explains how to write an efficient RTL code and how to improve design performance. The book also describes advanced RTL design concepts such as low-power design, multiple clock-domain design, and SOC-based design. The practical orientation of the book makes it ideal for training programs for practicing design engineers and for short-term vocational programs. The contents of the book will also make it a useful read for students and hobbyists.

Verilog is a Hardware Description Language (HDL) used to design and document electronic systems. Verilog HDL allows designers to virtually design systems without expending time or resources on physical models. It is the most widely used HDL with a user community of more than 50,000 active designers.

A practical primer for the student and practicing engineer already familiar with the basics of digital design, the reference develops a working grasp of the verilog hardware description language step-by-step using easy-to-understand examples. Starting with a simple but workable design sample, increasingly more complex fundamentals of the language are introduced until all major features of verilog are brought to light. Included in the coverage are state machines, modular design, FPGA-based memories, clock management, specialized I/O, and an introduction to techniques of simulation. The goal is to prepare the reader to design real-world FPGA solutions. All the sample code used in the book is available online. What Strunk and White did for the English language with "The Elements of Style," VERILOG BY EXAMPLE does for FPGA design.

Design Verification with E

Verilog Designer's Library

Using Verilog Hdl and FPGAs

Embedded SoPC Design with Nios II Processor and VHDL Examples

Digital Logic Design Using Verilog

Basics

This title builds on the student's background from a first course in logic design and focuses on developing, verifying, and synthesizing designs of digital circuits. The Verilog language is introduced in an integrated, but selective manner, only as needed to support design examples.

Designing a complex ASIC/SoC is similar to learning a new language to start with and ultimately creating a masterpiece using experience, imagination, and creativity. Digital design starts with RTL such as Verilog or VHDL, but it is only the beginning. A complete designer needs to have a good understanding of the Verilog language, digital design techniques, system architecture, IO protocols, and hardware-software interaction. Some of it will come from experience, and some will come with concerted effort. Graduating from college and entering into the world of digital system design becomes an overwhelming task, as not all the information is readily available. In this book, we have made an effort to explain the concepts in a simple way with real-world examples in Verilog. The book is intended for digital and system design engineers with emphasis on design and system architecture. The book is broadly divided into two sections - chapters 1 through 10, focusing on the digital design aspects and chapters 11 through 20, focusing on the system aspects of chip design. This book can be used by students taking digital design and chip design courses in college and availing it as a

guide in their professional careers. Chapter 3 focuses on the synthesizable Verilog constructs, with examples on reusable design (parameterized design, functions, and generate structure). Chapter 5 describes the basic concepts in digital design - logic gates, truth table, De Morgan's theorem, set-up and hold time, edge detection, and number system. Chapter 6 goes into details of digital design explaining larger building blocks such as LFSR, scrambler/descramblers, error detection and correction, parity, CRC, Gray encoding/decoding, priority encoders, 8b/10b encoding, data converters, and synchronization techniques. Chapter 7 and 8 bring in advanced concepts in chip design and architecture - clocking and reset strategy, methods to increase throughput and reduce latency, flow-control mechanisms, pipeline operation, out-of-order execution, FIFO design, state machine design, arbitration, bus interfaces, linked list structure, and LRU usage and implementation. Chapter 9 and 10 describe how to build and design ASIC/SoC. It talks about chip micro-architecture, portioning, datapath, control logic design, and other aspects of chip design such as clock tree, reset tree, and EEPROM. It also covers good design practices, things to avoid and adopt, and best practices for high-speed design. The second part of the book is devoted to System architecture, design, and IO protocols. Chapter 11 talks about memory, memory hierarchy, cache, interrupt, types of DMA and DMA operation. There is Verilog RTL for a typical DMA controller design that explains the scatter-gather DMA concept. Chapter12 describes hard drive, solid-state drive, DDR operation, and other parts of a system such as BIOS, OS, drivers, and their interaction with hardware. Chapter 13 describes embedded systems and internal buses such as AHB, AXI used in embedded design. It describes the concept of transparent and non-transparent bridging. Chapter 14 and chapter 15 bring in practical aspects of chip development - testing, DFT, scan, ATPG, and detailed flow of the chip development cycle (Synthesis, Static timing, and ECO). Chapter 16 and chapter 17 are on power saving and power management protocols. Chapter 16 has a detailed description of various power savings techniques (frequency variation, clock gating, and power well isolation). Chapter 17 talks about Power Management protocols such as system S states, CPU C states, and device D states. Chapter 18 explains the architecture behind serial-bus technology, PCS, and PMA layer. It describes clocking architecture and advanced concepts such as elasticity FIFO, channel bonding (deskewing), link aggregation, and lane reversal. Chapter 19 and 20 are devoted to serial bus protocols (PCI Express, Serial ATA, USB, Thunderbolt, and Ethernet) and their operation.

This book introduces the reader to FPGA based design for RTL synthesis. It describes simple to complex RTL design scenarios using SystemVerilog. The book builds the story from basic fundamentals of FPGA based designs to advance RTL design and verification concepts using SystemVerilog. It provides practical information on the issues in the RTL design and verification and how to overcome these. It focuses on writing efficient RTL codes using SystemVerilog, covers design for the Xilinx FPGAs and also includes implementable code examples. The contents of this book cover improvement of design performance, assertion based verification, verification planning, and architecture and system testing using FPGAs. The book can be used for classroom teaching or as a supplement in lab work for undergraduate and graduate coursework as well as for professional development and training programs. It will also be of interest to researchers and professionals interested in the RTL design for FPGA and ASIC.

SystemVerilog provides abundant features that could overwhelm a SystemVerilog beginner. Fortunately, for a decent RTL design, only a small subset of SystemVerilog is needed. The purpose of this book is to carefully choose the right subset of SystemVerilog so that the digital designer can comfortably start their SystemVerilog design project. In this book, FPGA application is chosen not only for its easy and quick practice but also for its wider adoption. SystemVerilog examples will be deployed broadly throughout this book for reference. For those who want to learn HDL design, this book will help them ramp up their HDL design skill quickly while avoiding the pitfalls. For those who have experience in Verilog but want to advance their knowledge to SystemVerilog, this book can be a good reference. For the VHDL designers who want to explore the features in SystemVerilog, this book can serve as a bridge since it is written in a way that the common and different concepts between VHDL and SystemVerilog are emphasized.The following are the specialties of this book:1. It provides a carefully chosen subset of SystemVerilog language for FPGA design.2. It provides a great number of examples for easier learning and practice3. It shows using SystemVerilog as an efficient way for a productive verification4. It emphasizes on the FPGA application but the presented RTL design is also applicable to ASIC.This book is organized as follows: Chapter 1 first briefly describes the HDL digital design methodology. Then it describes SystemVerilog language and its syntax. The basic topics include lexical convention, data type, operators, and expressions. It also explains various programming statements such as assignment statements, if-else statements, case statements and loop statements.Chapter 2 shows how to use SystemVerilog to describe the basic digital gates and digital hardware circuits as well as to model their behavior. It explains SytemVerilog modelling constructs. The constructs are modules, procedures, interfaces, functions and packages. This chapter also covers advanced topics such as compiler directives, digital arithmetic operation and design optimization.Chapter 3 introduces the synchronous sequential digital design. It gives some example designs such as flip-flop registers, shift registers, counters and adders. The design of finite-state machine (FSM) is discussed in depth for control circuit in digital systems. The algorithmic state machine (ASM) with data path is described for data-processing digital system. It also addresses other advanced topics of timing analysis, design performance and clock-domain crossing.Chapter 4 focuses on the functional simulation of digital design. It describes the general construction of test bench using SystemVerilog. It introduces the initial procedure for pre-simulation initialization, the final procedure for post-simulation processing and the task procedure for repetitive operations. It explains how to control the simulation proceeding with procedure timing control. It presents some useful system functions and tasks for math functions, file I/O and etc.. Chapter 5 addresses the FPGA design methodology. The topics covers design flow, design environment, intellectual property (IP) core usage, simulation and constraints. The FPGA design for system-on-chip (SOC) is emphasized as this type of FPGA becomes popular. The FPGA configuration options are discussed. Last but not least, it introduces helpful FPGA design practices for a successful design.

Advanced Digital Logic Design

SystemVerilog Assertions Handbook

Verilog by Example

Verilog HDL
FPGA Prototyping by SystemVerilog Examples
Advanced Digital System Design

VERILOG HDL, Second Editionby Samir PalnitkarWith a Foreword by Prabhu GoelWritten forboth experienced and new users, this book gives you broad coverage of VerilogHDL. The book stresses the practical design and verification perspective ofVerilog rather than emphasizing only the language aspects. The informationpresented is fully compliant with the IEEE 1364-2001 Verilog HDL standard. Among its many features, this edition- bull; bull;Describes state-of-the-art verification methodologies bull;Provides full coverage of gate, dataflow (RTL), behavioral and switch modeling bull;Introduces you to the Programming Language Interface (PLI) bull;Describes logic synthesis methodologies bull;Explains timing and delay simulation bull;Discusses user-defined primitives bull;Offers many practical modeling tips Includes over 300 illustrations, examples, and exercises, and a Verilog resource list.Learning objectives and summaries are provided for each chapter. About the CD-ROMThe CD-ROM contains a Verilog simulator with agraphical user interface and the source code for the examples in the book. Whatpeople are saying about Verilog HDL- "Mr.Palnitkar illustrates how and why Verilog HDL is used to develop today'smost complex digital designs. This book is valuable to both the novice and theexperienced Verilog user. I highly recommend it to anyone exploring Verilogbased design." -RajeevMadhavan, Chairman and CEO, Magma Design Automation "Thisbook is unique in its breadth of information on Verilog and Verilog-relatedtopics. It is fully compliant with the IEEE 1364-2001 standard, contains allthe information that you need on the basics, and devotes several chapters toadvanced topics such as verification, PLI, synthesis and modelingtechniques." -MichaelMcNamara, Chair, IEEE 1364-2001 Verilog Standards Organization Thishas been my favorite Verilog book since I picked it up in college. It is theonly book that covers practical Verilog. A must have for beginners andexperts." -BerendOzceri, Design Engineer, Cisco Systems, Inc. "Simple,logical and well-organized material with plenty of illustrations, makes this anideal textbook." -Arun K. Somani, Jerry R. Junkins Chair Professor,Department of Electrical and Computer Engineering, Iowa State University, Ames PRENTICE HALL Professional Technical Reference Upper Saddle River, NJ 07458 www.phptr.com ISBN: 0-13-044911-3

A comprehensive resource on Verilog HDL for beginners and experts Large and complicated digital circuits can be incorporated into hardware by using Verilog, a hardware description language (HDL). A designer aspiring to master this versatile language must first become familiar with its constructs, practice their use in real applications, and apply them in combinations in order to be successful. Design Through Verilog HDL affords novices the opportunity to perform all of these tasks, while also offering seasoned professionals a comprehensive resource on this dynamic tool. Describing a design using Verilog is only half the story: writing test-benches, testing a design for all its desired functions, and how identifying and removing the faults remain significant challenges. Design Through Verilog HDL addresses each of these issues concisely and effectively. The authors discuss constructs through illustrative examples that are tested with popular simulation packages, ensuring the subject matter remains practically relevant. Other important topics covered include: Primitives Gate and Net delays Buffers CMOS switches State machine design Further, the authors focus on illuminating the differences between gate level, data flow, and behavioral styles of Verilog, a critical distinction for designers. The book's final chapters deal with advanced topics such as timescales, parameters and related constructs, queues, and switch level design. Each chapter concludes with exercises that both ensure readers have mastered the present material and stimulate readers to explore avenues of their own choosing. Written and assembled in a paced, logical manner, Design Through Verilog HDL provides professionals, graduate students, and advanced undergraduates with a one-of-a-kind resource.

FPGA Prototyping Using Verilog Examples will provide you with a hands-on introduction to Verilog synthesis and FPGA programming through a “learn by doing” approach. By following the clear, easy-to-understand templates for code development and the numerous practical examples, you can quickly develop and simulate a sophisticated digital circuit, realize it on a prototyping device, and verify the operation of its physical implementation. This introductory text that will provide you with a solid foundation, instill confidence with rigorous examples for complex systems and prepare you for future development tasks.

With this book, you can: - Start writing synthesizable Verilog models quickly. - See what constructs are supported for synthesis and how these map to hardware so that you can get the desired logic. - Learn techniques to help avoid having functional mismatches. - Immediately start using many of the models for commonly used hardware elements described for your own use or modify these for your own application.

Xilinx Spartan-3 Version

Design Recipes for FPGAs
Real Chip Design and Verification Using Verilog and VHDL
An Embedded Systems Approach Using Verilog
A Practical Guide for Designing, Synthesizing, and Simulating ASICs and FPGAs Using VHDL Or Verilog
RTL Design Using Verilog

This book concentrates on common classes of hardware architectures and design problems, and focuses on the process of transitioning design requirements into synthesizable HDL code. Using his extensive, wide-ranging experience in computer architecture and hardware design, as well as in his training and consulting work, Ben provides numerous examples of real-life designs illustrated with VHDL and Verilog code. This code is shown in a way that makes it easy for the reader to gain a greater understanding of the languages and how they compare. All code presented in the book is included on the companion CD, along with other information, such as application notes.

This book provides a rich toolbox of design techniques and templates to solve practical, every-day problems using FPGAs. Using a modular structure, it provides design techniques and templates at all levels, together with functional code, which you can easily match and apply to your application. Written in an informal and easy to grasp style, this invaluable resource goes beyond the principles of FPGAs and hardware description languages to demonstrate how specific designs can be synthesized, simulated and downloaded onto an FPGA. In addition, the book provides advanced techniques to create ‘real world’ designs that fit the device required and which are fast and reliable to implement. Examples are rewritten and tested in Verilog and VHDL Describes high-level applications as examples and provides the building blocks to implement them, enabling the student to start practical work straight away Singles out the most important parts of the language that are needed for design, giving the student the information needed to get up and running quickly

As part of the Modern Semiconductor Design series, this book details a broad range of e-based topics including modelling, constraint-driven test generation, functional coverage and assertion checking.

This book was written for new designers looking for a solid foundation in PCB design although designers with more experience will find the reference material, software, and explanations of the values that manufacturers use invaluable as well.

FPGA Prototyping by Verilog Examples

RTL Design and Verification
Digital Signal Processing with Field Programmable Gate Arrays
Verilog Coding for Logic Synthesis
A Practical Guide to Verilog Based FPGA and ASIC Implementation, Numerous Examples of Digital System Design Along with Brief Discussion on Verilog HDL, Digital Logic Design, Computer Arithmetic, Implementation Platforms , Timing Analysis and Low Power Design Techniques
Using Systemverilog and Fpga

Ready-to-use building blocks for integrated circuit design. Why start coding from scratch when you can work from this library of pre-tested routines, created by an HDL expert? There are plenty of introductory texts to describe the basics of Verilog, but Verilog Designer's Library is the only book that offers real, reusable routines that you can put to work right away. Verilog Designer's Library organizes Verilog routines according to functionality, making it easy to locate the material you need. Each function is described by a behavioral model to use for simulation, followed by the RTL code you'll use to synthesize the gate-level implementation. Extensive test code is included for each function, to assist you with your own verification efforts. Coverage includes: Essential Verilog coding techniques Basic building blocks of successful routines State machines and memories Practical debugging guidelines Although Verilog Designer's Library assumes a basic familiarity with Verilog structure and syntax, it does not require a background in programming. Beginners can work through the book in sequence to develop their skills, while experienced Verilog users can go directly to the routines they need. Hardware designers, systems analysts, VARs, OEMs, software developers, and system integrators will find it an ideal sourcebook on all aspects of Verilog development.

The practical guide for every circuit designer creating FPGA designs with Verilog! Walk through design step-by-step-from coding through silicon. Partitioning, synthesis, simulation, test benches, combinatorial and sequential designs, and more. Real World FPGA Design with Verilog guides you through every key challenge associated with designing FPGAs and ASICs using Verilog, one of the world's leading hardware design languages. You'll find irreverent, yet rigorous coverage of what it really takes to translate HDL code into hardware-and how to avoid the pitfalls that can occur along the way. Ken Coffman presents no-frills, real-world design techniques that can improve the stability and reliability of virtually any design. Start by walking a typical Verilog design all the way through to silicon; then, review basic Verilog syntax, design; simulation and testing, advanced simulation, and more. Coverage includes: Essential digital design strategies: recognizing the underlying analog building blocks used to create digital primitives; implementing logic with LUTs; clocking strategies, logic minimization, and more Key engineering tradeoffs, including operating speed vs. latency Combinatorial and sequential designs Verilog test fixtures: compiler directives and automated testing A detailed comparison of alternative architectures and software-including a never-before-published FPGA technology selection checklist Real World FPGA Design with Verilog introduces libraries and reusable modules, points out opportunities to reuse your own code, and helps you decide when to purchase existing IP designs instead of building from scratch. Essential rules for designing with ASIC conversion in mind are presented. If you're involved with digital hardware design with Verilog, Ken Coffman is a welcome voice of experience-showing you the shortcuts, helping you over the rough spots, and helping you achieve competence faster than you ever expected!

SystemVerilog language consists of three categories of features -- Design, Assertions and Testbench. Assertions add a whole new dimension to the ASIC verification process. Engineers are used to writing testbenches in verilog that help verify their design. Verilog is a procedural language and is very limited in capabilities to handle the complex ASICs built today. SystemVerilog assertions (SVA) is a declarative language. The temporal nature of the language provides excellent control over time and allows mulitple processes to execute simultaneously. This provides the engineers a very strong tool to solve their verification problems. The language is still new and the thinking is very different from the user's perspective when compared to standard verilog language. There is not enough expertise or intellectual property available as of today in the field. While the language has been defined very well, there is no practical guide that shows how to use the language to solve real verification problems. This book is a practical guide that will help people to understand this new language and adopt assertion based verification methodology quickly.

Design Recipes for FPGAs: Using Verilog and VHDL provides a rich toolbox of design techniques and templates to solve practical, every-day problems using FPGAs. Using a modular structure, the book gives ‘easy-to-find’ design techniques and templates at all levels, together with functional code. Written in an informal and ‘easy-to-grasp’ style, it goes beyond the principles of FPGA s and hardware description languages to actually demonstrate how specific designs can be synthesized, simulated and downloaded onto an FPGA. This book's ‘easy-to-find’ structure begins with a design application to demonstrate the key building blocks of FPGA design and how to connect them, enabling the experienced FPGA designer to quickly select the right design for their application, while providing the less experienced a ‘road map’ to solving their specific design problem. The book also provides advanced techniques to create ‘real world’ designs that fit the device required and which are fast and reliable to implement. This text will appeal to FPGA designers of all levels of experience. It is also an ideal resource for embedded system development engineers, hardware and software engineers, and undergraduates and postgraduates studying an embedded system which focuses on FPGA design. A rich toolbox of practical FGPA design techniques at an engineer's finger tips Easy-to-find structure that allows the engineer to quickly locate the information to solve their FGPA design problem, and obtain the level of detail and understanding needed

Advanced Chip Design
Advanced HDL Synthesis and SOC Prototyping

SystemVerilog for Hardware Description
Verilog Hdl Synthesis, a Practical Primer
Coding and RTL Synthesis
Digital System Designs and Practices