

Fpga Prototyping By Verilog Examples Xilinx Spartan 3 Version By Chu Pong P Published By Wiley Blackwell 2008

FPGAs have almost entirely replaced the traditional Application Specific Standard Parts (ASSP) such as the 74xx logic chip families because of their superior size, versatility, and speed. For example, FPGAs provide over a million fold increase in gates compared to ASSP parts. The traditional approach for hands-on exercises has relied on ASSP parts, primarily because of their simplicity and ease of use for the novice. Not only is this approach technically outdated, but it also severely limits the complexity of the designs that can be implemented. By introducing the readers to FPGAs, they are being familiarized with current digital technology and the skills to implement complex, sophisticated designs. However, working with FPGAs comes at a cost of increased complexity, notably the mastering of an HDL language, such as Verilog. Therefore, this book accomplishes the following: first, it teaches basic digital design concepts and then applies them through exercises; second, it implements these digital designs by teaching the user the syntax of the Verilog language while implementing the exercises. Finally, it employs contemporary digital hardware, such as the FPGA, to build a simple calculator, a basic music player, a frequency and period counter and it ends with a microprocessor being embedded in the fabric of the FPGA to communicate with the PC. In the process, readers learn about digital mathematics and digital-to-analog converter concepts through pulse width modulation.

Rapid Prototyping of Digital Systems, Second Edition provides an exciting and challenging laboratory component for an undergraduate digital logic design class. The more advanced topics and exercises are also appropriate for consideration at schools that have an upper level course in digital logic or programmable logic. Design engineers working in industry will also want to consider this book for a rapid introduction to FPLD technology and logic synthesis using commercial CAD tools, especially if they have not had previous experience with the new and rapidly evolving technology. Two tutorials on the Altera CAD tool environment, an overview of programmable logic, and a design library with several easy-to-use input and output functions were developed for this book to help the reader get started quickly. Early design examples use schematic capture and library components. VHDL is used for more complex designs after a short introduction to VHDL-based synthesis. A coupon is included with the text for purchase of the new UP 1X board. The additional logic and memory in the UP 1X's FLEX 10K70 is useful on larger design projects such as computers and video games. The second edition includes an update chapter on programmable logic, new robot sensors and projects, optional Verilog examples, and a meta assembler which can be used to develop assemble language programs for the computer designs in Chapters 8 and 13.

This book constitutes the refereed proceedings of the 11th International Symposium on Applied Reconfigurable Computing, ARC 2015, held in Bochum, Germany, in April 2015. The 23 full papers and 20 short papers presented in this volume were carefully reviewed and selected from 85 submissions. They are organized in topical headings named: architecture and modeling; tools and compilers; systems and applications; network-on-a-chip; cryptography applications; extended abstracts of posters. In addition, the book contains invited papers on funded R&D - running and completed projects and Horizon 2020 funded projects.

Take your creations to the next level with FPGAs and Verilog This fun guide shows how to get started with FPGA technology using the popular Mojo, Papilio One, and Elbert 2 boards. Written by electronics guru Simon Monk, Programming FPGAs: Getting Started with Verilog features clear explanations, easy-to-follow examples, and downloadable sample programs. You'll get start-to-finish assembly and programming instructions for numerous projects, including an LED decoder, a timer, a tone generator—even a memory-mapped video display! The book serves both as a hobbyists' guide and as an introduction for professional developers.

- Explore the basics of digital electronics and digital logic
- Examine the features of the Mojo, Papilio One, and Elbert 2 boards
- Set up your computer and dive in to Verilog programming
- Work with the ISE Design Suite and user constraints files
- Understand and apply modular Verilog programming methods
- Generate electrical pulses through your board's GPIO ports
- Control servomotors and create your own sounds
- Attach a VGA TV or computer monitor and generate video
- All source code and finished bit files available for download

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

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 multiple 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.

A hands-on introduction to FPGA prototyping and SoC design This Second Edition of the popular book follows the same "learning-by-doing" approach to teach the fundamentals and practices

of VHDL synthesis and FPGA prototyping. It 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 strict design guidelines and coding practices used for large, complex digital systems. The new edition is completely updated. 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 revised 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) envelop 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. Introduces basic concepts of software-hardware co-design with Xilinx MicroBlaze MCS soft-core processor. Provides an overview of bus interconnect and interface circuit. Introduces basic embedded system software development. Suggests additional modules and peripherals for interesting and challenging projects. The FPGA Prototyping by VHDL Examples, Second Edition makes a natural companion text for introductory and advanced digital design courses and embedded system course. It also serves as an ideal self-teaching guide for practicing engineers who wish to learn more about this emerging area of interest.

FPGA Prototyping By Verilog Examples Xilinx Spartan-3 Version Wiley-Interscience

This book presents a selection of papers representing current research on using field programmable gate arrays (FPGAs) for realising image processing algorithms. These papers are reprints of papers selected for a Special Issue of the Journal of Imaging on image processing using FPGAs. A diverse range of topics is covered, including parallel soft processors, memory management, image filters, segmentation, clustering, image analysis, and image compression. Applications include traffic sign recognition for autonomous driving, cell detection for histopathology, and video compression. Collectively, they represent the current state-of-the-art on image processing using FPGAs.

This book provides step-by-step guidance on how to design VLSI systems using Verilog. It shows the way to design systems that are device, vendor and technology independent. Coverage presents new material and theory as well as synthesis of recent work with complete Project Designs using industry standard CAD tools and FPGA boards. The reader is taken step by step through different designs, from implementing a single digital gate to a massive design consuming well over 100,000 gates. All the design codes developed in this book are Register Transfer Level (RTL) compliant and can be readily used or amended to suit new projects.

by Phil Moorby The Verilog Hardware Description Language has had an amazing impact on the modern electronics industry, considering that the essential composition of the language was developed in a surprisingly short period of time, early in 1984. Since its introduction, Verilog has changed very little. Over time, users have requested many improvements to meet new methodology needs. But, it is a complex and time consuming process to add features to a language without ambiguity, and maintaining consistency. A group of Verilog enthusiasts, the IEEE 1364 Verilog committee, have broken the Verilog feature doldrums. These individuals should be applauded. They invested the time and energy, often their personal time, to understand and resolve an extensive wish-list of language enhancements. They took on the task of choosing a feature set that would stand up to the scrutiny of the standardization process. I would like to personally thank this group. They have shown that it is possible to evolve Verilog, rather than having to completely start over with some revolutionary new language. The Verilog 1364-2001 standard provides many of the advanced building blocks that users have requested. The enhancements include key components for verification, abstract design, and other new methodology capabilities. As designers tackle advanced issues such as automated verification, system partitioning, etc., the Verilog standard will rise to meet the continuing challenge of electronics design. The skills and guidance needed to master RTL hardware design This book teaches readers how to systematically design efficient, portable, and scalable Register Transfer Level (RTL) digital circuits using the VHDL hardware description language and synthesis software. Focusing on the module-level design, which is composed of functional units, routing circuit, and storage, the book illustrates the relationship between the VHDL constructs and the underlying hardware components, and shows how to develop codes that faithfully reflect the module-level design and can be synthesized into efficient gate-level implementation. Several unique features distinguish the book: * Coding style that shows a clear relationship between VHDL constructs and hardware components * Conceptual diagrams that illustrate the realization of VHDL codes * Emphasis on the code reuse * Practical examples that demonstrate and reinforce design concepts, procedures, and techniques * Two chapters on realizing sequential algorithms in hardware * Two chapters on scalable and parameterized designs and coding * One chapter covering the synchronization and interface between multiple clock domains Although the focus of the book is RTL synthesis, it also examines the synthesis task from the perspective of the overall development process. Readers learn good design practices and guidelines to ensure that an RTL design can accommodate future simulation, verification, and testing needs, and can be easily incorporated into a larger system or reused. Discussion is independent of technology and can be applied to both ASIC and FPGA devices. With a balanced presentation of fundamentals and practical examples, this is an excellent textbook for upper-level undergraduate or graduate courses in advanced digital logic. Engineers who need to make effective use of today's synthesis software and FPGA devices should also refer to this book.

Field Programmable Gate Arrays (FPGAs) are devices that provide a fast, low-cost way for embedded system designers to customize products and deliver new versions with upgraded features, because they can handle very complicated functions, and be reconfigured an infinite number of times. In addition to introducing the various architectural features available in the latest generation of FPGAs, The Design Warrior's Guide to FPGAs also covers different design tools and flows. This book covers information ranging from schematic-driven entry, through traditional HDL/RTL-based simulation and logic synthesis, all the way up to the current state-of-the-art in pure C/C++ design capture and synthesis technology. Also discussed are specialist areas such as mixed hardware/software and DSP-based design flows, along with innovative new devices such as field programmable node arrays (FPNAs). Clive "Max" Maxfield is a

bestselling author and engineer with a large following in the electronic design automation (EDA) and embedded systems industry. In this comprehensive book, he covers all the issues of interest to designers working with, or contemplating a move to, FPGAs in their product designs. While other books cover fragments of FPGA technology or applications this is the first to focus exclusively and comprehensively on FPGA use for embedded systems. First book to focus exclusively and comprehensively on FPGA use in embedded designs World-renowned best-selling author Will help engineers get familiar and succeed with this new technology by providing much-needed advice on choosing the right FPGA for any design project

This book attempts to capture the spirit of the "Bronze Age" of video games, when video games were designed as circuits, not as software. We'll delve into these circuits as they morph from Pong into programmable personal computers and game consoles. Instead of wire-wrap and breadboards, we'll use modern tools to approximate these old designs in a simulated environment from the comfort of our keyboards. At the end of this adventure, you should be well-equipped to begin exploring the world of FPGAs, and maybe even design your own game console. You'll use the 8bitworkshop.com IDE to write Verilog programs that represent digital circuits, and see your code run instantly in the browser.

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.

This book is both a tutorial and a reference for engineers who use the SystemVerilog Hardware Description Language (HDL) to design ASICs and FPGAs. The book shows how to write SystemVerilog models at the Register Transfer Level (RTL) that simulate and synthesize correctly, with a focus on proper coding styles and best practices. SystemVerilog is the latest generation of the original Verilog language, and adds many important capabilities to efficiently and more accurately model increasingly complex designs. This book reflects the SystemVerilog-2012/2017 standards. This book is for engineers who already know, or who are learning, digital design engineering. The book does not present digital design theory; it shows how to apply that theory to write RTL models that simulate and synthesize correctly. The creator of the original Verilog Language, Phil Moorby says about this book (an excerpt from the book's Foreword): "Many published textbooks on the design side of SystemVerilog assume that the reader is familiar with Verilog, and simply explain the new extensions. It is time to leave behind the stepping-stones and to teach a single consistent and concise language in a single book, and maybe not even refer to the old ways at all! If you are a designer of digital systems, or a verification engineer searching for bugs in these designs, then SystemVerilog will provide you with significant benefits, and this book is a great place to learn the design aspects of SystemVerilog."

A no-nonsense, practical guide to current and future processor and computer architectures, enabling you to design computer systems and develop better software applications across a variety of domains Key Features Understand digital circuitry with the help of transistors, logic gates, and sequential logic Examine the architecture and instruction sets of x86, x64, ARM, and RISC-V processors Explore the architecture of modern devices such as the iPhone X and high-performance gaming PCs Book Description Are you a software developer, systems designer, or computer architecture student looking for a methodical introduction to digital device architectures but overwhelmed by their complexity? This book will help you to learn how modern computer systems work, from the lowest level of transistor switching to the macro view of collaborating multiprocessor servers. You'll gain unique insights into the internal behavior of processors that execute the code developed in high-level languages and enable you to design more efficient and scalable software systems. The book will teach you the fundamentals of computer systems including transistors, logic gates, sequential logic, and instruction operations. You will learn details of modern processor architectures and instruction sets including x86, x64, ARM, and RISC-V. You will see how to implement a RISC-V processor in a low-cost FPGA board and how to write a quantum computing program and run it on an actual quantum computer. By the end of this book, you will have a thorough understanding of modern processor and computer architectures and the future directions these architectures are likely to take. What you will learn Get to grips with transistor technology and digital circuit principles Discover the functional elements of computer processors Understand pipelining and superscalar execution Work with floating-point data formats Understand the purpose and operation of the supervisor mode Implement a complete RISC-V processor in a low-cost FPGA Explore the techniques used in virtual machine implementation Write a quantum computing program and run it on a quantum computer Who this book is for This book is for software developers, computer engineering students, system designers, reverse engineers, and anyone looking to understand the architecture and design principles underlying modern computer systems from tiny embedded devices to warehouse-size cloud server farms. A general understanding of computer processors is helpful but not required.

A completely updated and expanded comprehensive treatment of VHDL and its applications to the design and simulation of real, industry-standard circuits. This comprehensive treatment of VHDL and its applications to the design and simulation of real, industry-standard circuits has been completely updated and expanded for the third edition. New features include all VHDL-2008 constructs, an extensive review of digital circuits, RTL analysis, and an unequalled collection of VHDL examples and exercises. The book focuses on the use of VHDL rather than solely on the language, with an emphasis on design examples and laboratory exercises. The third edition begins with a detailed review of digital circuits (combinatorial, sequential, state machines, and FPGAs), thus providing a self-contained single reference for the teaching of digital circuit design with VHDL. In its coverage of VHDL-2008, it makes a clear distinction between VHDL for synthesis and VHDL for simulation. The text offers complete VHDL codes in examples as well as simulation results and comments. The significantly expanded examples and exercises include many not previously published, with multiple physical demonstrations meant to inspire and motivate students. The book is suitable for undergraduate and graduate students in VHDL and digital circuit design, and can be used as a professional reference for VHDL practitioners. It can also serve as a text for digital VLSI in-house or academic courses.

Verilog aims to introduce new users to the language of Verilog with instruction on how to write hardware descriptions in Verilog in a style that can be synthesized by readily available synthesis tools. Offers clear exposition of the Verilog hardware description language. This book is written in a style that allows the user who has no previous background with hardware description languages (HDLs) to become skillful with the language. Features treatment of synthesis-friendly descriptive styles. An excellent book for self-study, reference, seminars, and workshops on the subject.

Project Oberon contains a definition of the Oberon Language and describes its relation to Modula-2 and the software tools developed with the system. This definitive, first-hand account of the design, development, and implementation of Oberon completes the Oberon trilogy.

• • Learn the 'whys and hows' of digital system design with FPGAs from this thorough treatment. • Up-to-date information and comparison of different modern FPGA devices. • IEEE Fellow Wayne Wolf brings all related aspects of VLSI to FPGA system design in this thorough introduction.

"Introduction to Embedded System Design Using Field Programmable Gate Arrays" provides a starting point for the use of field programmable gate arrays in the design of embedded systems. The text considers a hypothetical robot controller as an embedded application and weaves around it related concepts of FPGA-based digital design. The book details: use of FPGA vis-à-vis general purpose processor

and microcontroller; design using Verilog hardware description language; digital design synthesis using Verilog and Xilinx® Spartan™ 3 FPGA; FPGA-based embedded processors and peripherals; overview of serial data communications and signal conditioning using FPGA; FPGA-based motor drive controllers; and prototyping digital systems using FPGA. The book is a good introductory text for FPGA-based design for both students and digital systems designers. Its end-of-chapter exercises and frequent use of example can be used for teaching or for self-study.

This book describes RTL design, synthesis, and timing closure strategies for SOC blocks. It covers high-level RTL design scenarios and challenges for SOC design. The book gives practical information on the issues in SOC and ASIC prototyping using modern high-density FPGAs. The book covers SOC performance improvement techniques, testing, and system-level verification. The book also describes the modern Xilinx FPGA architecture and their use in SOC prototyping. The book covers the Synopsys DC, PT commands, and use of them to constraint and to optimize SOC design. The contents of this book will be of use to students, professionals, and hobbyists alike.

The push to move products to market as quickly and cheaply as possible is fiercer than ever, and accordingly, engineers are always looking for new ways to provide their companies with the edge over the competition. Field-Programmable Gate Arrays (FPGAs), which are faster, denser, and more cost-effective than traditional programmable logic devices (PLDs), are quickly becoming one of the most widespread tools that embedded engineers can utilize in order to gain that needed edge. FPGAs are especially popular for prototyping designs, due to their superior speed and efficiency. This book hones in on that rapid prototyping aspect of FPGA use, showing designers exactly how they can cut time off production cycles and save their companies money drained by costly mistakes, via prototyping designs with FPGAs first. Reading it will take a designer with a basic knowledge of implementing FPGAs to the "next-level of FPGA use because unlike broad beginner books on FPGAs, this book presents the required design skills in a focused, practical, example-oriented manner. In-the-trenches expert authors assure the most applicable advice to practicing engineers. Dual focus on successfully making critical decisions and avoiding common pitfalls appeals to engineers pressured for speed and perfection. Hardware and software are both covered, in order to address the growing trend toward "cross-pollination" of engineering expertise.

With the advance of semiconductor and communication industry, the use of system-on-chip (SoC) has become an essential technique to reduce product costs. The development of a good understanding of the key stages of the hardware description language (HDL) design flow based on cell-based libraries or field-programmable gate array (FPGA) devices becomes essential. This book addresses the needs for such a topic based on Verilog HDL and FPGAs. The most important features of this book include 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 include: - Explaining how to perform synthesis and verification to achieve optimized synthesis results and compiler times - Illustrating the entire design and verification flow using an FPGA case study - Emphasizing design/implementation trade-off options, with coverage of ASICs and FPGAs - Providing plentiful worked examples and review questions in each section for readers to test their understanding of the related topics - Giving readers deeper understanding with plentiful review questions in each section and end-of-chapter problems - Incorporating many case studies to help the reader grasp the essentials of practical digital systems to be designed using Verilog HDL and FPGAs - Highlighting Verilog HDL syntax throughout the book to facilitate readers to refer the desired syntax as they need - Printing all keywords in boldface throughout the book to emphasize the language structures and improve the readability of Verilog HDL modules. This book is the ideal textbook for the following courses: Digital System Design, FPGA System Designs and Practices, Advanced Digital Systems Design, and the like. In addition, it can be used as a self-studying or professional reference book in this field.

The Definitive, Up-to-Date Guide to Digital Design with SystemVerilog: Concepts, Techniques, and Code To design state-of-the-art digital hardware, engineers first specify functionality in a high-level Hardware Description Language (HDL)—and today's most powerful, useful HDL is SystemVerilog, now an IEEE standard. Digital System Design with SystemVerilog is the first comprehensive introduction to both SystemVerilog and the contemporary digital hardware design techniques used with it. Building on the proven approach of his bestselling Digital System Design with VHDL, Mark Zwolinski covers everything engineers need to know to automate the entire design process with SystemVerilog—from modeling through functional simulation, synthesis, timing simulation, and verification. Zwolinski teaches through about a hundred and fifty practical examples, each with carefully detailed syntax and enough in-depth information to enable rapid hardware design and verification. All examples are available for download from the book's companion Web site, zwolinski.org. Coverage includes Using electronic design automation tools with programmable logic and ASIC technologies Essential principles of Boolean algebra and combinational logic design, with discussions of timing and hazards Core modeling techniques: combinational building blocks, buffers, decoders, encoders, multiplexers, adders, and parity checkers Sequential building blocks: latches, flip-flops, registers, counters, memory, and sequential multipliers Designing finite state machines: from ASM chart to D flip-flops, next state, and output logic Modeling interfaces and packages with SystemVerilog Designing testbenches: architecture, constrained random test generation, and assertion-based verification Describing RTL and FPGA synthesis models Understanding and implementing Design-for-Test Exploring anomalous behavior in asynchronous sequential circuits Performing Verilog-AMS and mixed-signal modeling Whatever your experience with digital design, older versions of Verilog, or VHDL, this book will help you discover SystemVerilog's full power and use it to the fullest.

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 helps readers to implement their designs on Xilinx® FPGAs. The authors demonstrate how to get the greatest impact from using the Vivado® Design Suite, which delivers a SoC-strength, IP-centric and system-centric, next generation development environment that has been built from the ground up to address the productivity bottlenecks in system-level integration and implementation. This book is a hands-on guide for both users who are new to FPGA designs, as well as those currently using the legacy Xilinx tool set (ISE) but are now moving to Vivado. Throughout the presentation, the authors focus on key concepts, major mechanisms for design entry, and methods to realize the most efficient implementation of the target design, with the least number of iterations.

This book uses a "learn by doing" approach to introduce the concepts and techniques of VHDL and FPGA to designers through a series of hands-on experiments. FPGA Prototyping by VHDL Examples provides a collection of clear, easy-to-follow templates for quick code development; a large number of practical examples to illustrate and reinforce the concepts and design techniques; realistic projects that can be implemented and tested on a Xilinx prototyping board; and a thorough exploration of the Xilinx PicoBlaze soft-core microcontroller.

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 practice. 3. It shows using SystemVerilog as an efficient way for a productive verification. 4. 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 SystemVerilog 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.

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.

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

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.

This second edition focuses on the thought process of digital design and implementation in the context of VLSI and system design. It covers the Verilog 2001 and Verilog 2005 RTL design styles, constructs and the optimization at the RTL and synthesis level. The book also covers the logic synthesis, low power, multiple clock domain design concepts and design performance improvement techniques. The book includes 250 design examples/illustrations and 100 exercise questions. This volume can be used as a core or supplementary text in undergraduate courses on logic design and as a text for professional and vocational coursework. In addition, it will be a hands-on professional reference and a self-study aid for hobbyists.

[Copyright: ef49e29bafb75cb163b0634594402a87](https://www.wiley.com/9781119999999)