Raspberry Pi Cookbook

Use Arrow's affordable and breadboard-friendly FPGA development board (BeMicro MAX 10) to create a light sensor, temperature sensor, motion sensor, and the KITT car display from Knight Rider. You don't need an electronics engineering degree or even any programming experience to get the most out of Beginning FPGA: Programming Metal. Just bring your curiosity and your Field-Programmable Gate Array. This book is for those who have tinkered with Arduino or Raspberry Pi, and want to get more hands-on experience with hardware or...
for those new to electronics who just want to dive in. You'll learn the theory behind FPGAs and electronics, including the math and logic you need to understand what's happening—all explained in a fun, friendly, and accessible way. It also doesn't hurt that you'll be learning VHDL, a hardware description language that is also an extremely marketable skill. What You'll Learn: Learn what an FPGA is and how it's different from a microcontroller or ASIC. Set up your toolchain. Use VHDL, a popular hardware description language, to tell your FPGA what to be. Explore the theory behind FPGA and electronics. Use your FPGA with a variety of sensors and to talk to a Raspberry Pi. Who This Book is For: Arduino, Raspberry Pi, and other electronics enthusiasts who want a clear and practical introduction to FPGA.

Practical Image and Video Processing Using MATLAB

“This book gives thorough, scholarly coverage of an area of growing importance in computer security and is a ‘must have’ for every researcher, student, and practicing professional in software protection.” —Mikhail Atallah, Distinguished Professor of Computer Science at Purdue University Theory, Techniques, and Tools for Fighting Software Piracy, Tampering, and Malicious Reverse Engineering The last decade has seen significant progress in the development of techniques for resisting software piracy and tampering. These techniques are indispensable for software developers seeking to protect vital intellectual property. Surreptitious Software is the first authoritative, comprehensive resource for researchers, developers, and students who want to understand these approaches, the level of security they afford, and the performance penalty they incur. Christian Collberg and Jasvir Nagra bring together techniques drawn from related areas of computer science, including cryptography, steganography, watermarking, software metrics, reverse engineering, and compiler optimization. Using extensive sample code, they show readers how to implement protection schemes ranging from code obfuscation and software fingerprinting to tamperproofing and birthmarking, and discuss the theoretical and practical limitations of these techniques. Coverage includes Mastering techniques that both attackers and defenders use to analyze programs. Using code obfuscation to make software harder to analyze and understand. Fingerprinting software to identify its author and to trace software pirates.
software using guards that detect and respond to illegal modifications of code and data. Strengthening content protection through dynamic watermarking and dynamic obfuscation. Detecting code theft via software similarity analysis and birthmarking algorithms. Using hardware techniques to defend software and media against piracy and tampering. Detecting software tampering in distributed systems. Understanding the theoretical limits of code obfuscation.

**RTL Hardware Design Using VHDL**

The book presents laboratory experiments concerning ARM microcontrollers, and discusses the architecture of the Tiva Cortex-M4 ARM microcontrollers from Texas Instruments, describing various ways of programming them. Given the meager peripherals and sensors available on the kit, the authors describe the design of Padma - a circuit board with a large set of peripherals and sensors that connects to the Tiva Launchpad and exploits the Tiva microcontroller family's on-chip features. ARM microcontrollers, which are classified as 32-bit devices, are currently the most popular of all microcontrollers. They cover a wide range of applications that extend from traditional 8-bit devices to 32-bit devices. Of the various ARM subfamilies, Cortex-M4 is a middle-level microcontroller that lends itself well to data acquisition and control as well as digital signal manipulation applications. Given the prominence of ARM microcontrollers, it is important that they should be incorporated in academic curriculums. However, there is a lack of up-to-date teaching material - textbooks and comprehensive laboratory manuals. In this book, each of the microcontroller's resources - digital input and output, timers and counters, serial communication channels, analog-to-digital conversion, interrupt structure and power management features - are addressed in a set of more than 70 experiments to help teach a full semester course on these microcontrollers. Beyond these physical interfacing exercises, it describes an inexpensive BoB (break out board) that allows students to learn how to design and build standalone projects, as well as a number of illustrative projects.

**The Design Warrior's Guide to FPGAs**
Learn how to design digital circuits with FPGAs (field-programmable gate arrays), the devices that reconfigure themselves to become the very hardware circuits you set out to program. With this practical guide, author Justin Rajewski shows you hands-on how to create FPGA projects, whether you’re a programmer, engineer, product designer, or maker. You’ll quickly go from the basics to designing your own processor. Designing digital circuits used to be a long and costly endeavor that only big companies could pursue. FPGAs make the process much easier, and now they’re affordable enough even for hobbyists. If you’re familiar with electricity and basic electrical components, this book starts simply and progresses through increasingly complex projects. Set up your environment by installing Xilinx ISE and the author’s Mojo IDE. Learn how hardware designs are broken into modules, comparable to functions in a software program. Create digital hardware designs and learn the basics on how they’ll be implemented by the FPGA. Build your projects with Lucid, a beginner-friendly hardware description language, based on Verilog, with syntax similar to C/C++ and Java.

FPGAs 101

FPGA Architecture: Survey and Challenges reviews the historical development of programmable logic devices, the fundamental programming technologies that the programmability is built on, and then describes the basic understandings gleaned from research on architectures. It is an invaluable reference for engineers and computer scientists. It is also an excellent primer for senior or graduate-level students in electrical engineering or computer science.

Make: FPGAs

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
Read Online Programming Fpgas Getting Started With Verilog

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.

Designing with Xilinx® FPGAs

Learn how to accelerate C++ programs using data parallelism. This open access book enables C++ programmers to be at the forefront of this exciting and important new development that is helping to push computing to new levels. It is full of practical advice, detailed explanations, and code examples to illustrate key topics. Data parallelism in C++ enables access to parallel resources in a modern heterogeneous system, freeing you from being locked into any particular computing device. Now a single C++ application can use any combination of devices—including GPUs, CPUs, FPGAs and AI ASICs—that are suitable to the problems at hand. This book begins by introducing data parallelism and foundational topics for effective use of the SYCL standard from the Khronos Group and Data Parallel C++ (DPC++), the open source compiler used in this book. Later chapters cover advanced topics including error handling, hardware-specific programming, communication and synchronization, and memory model considerations. Data Parallel C++ provides you with everything needed to use SYCL for programming heterogeneous systems. What You'll Learn Accelerate C++ programs using data-parallel programming Target multiple device types (e.g. CPU, GPU, FPGA) Use SYCL and SYCL compilers Connect with computing’s heterogeneous future via Intel’s oneAPI initiative Who This Book Is For Those new data-parallel programming and computer programmers interested in data-parallel programming and computer programmers interested in data-parallel programming using C++.

Programming Arduino Getting Started with Sketches

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.

Hardware Description Language Demystified

LabVIEW programming techniques, tips, and practices Learn to build effective LabVIEW programs using the detailed information contained in this thoroughly revised resource. This edition updates all content to align with the latest version and adds new chapters that clearly explain object-oriented programming methods, and programming in teams using the cloud. LabVIEW Graphical Programming, Fifth Edition begins with basics for beginners and quickly progresses to intermediate and advanced programming techniques. Written by a pair of LabVIEW experts, this hands-on guide shows how to work with data types, start building your own applications, handle I/O, and use the DAQmix library. You will also find out how to build applications that communicate with enterprise message brokers and with Amazon Web Services’ Internet of Things (IoT) message broker. Coverage includes: The origin and evolution of LabVIEW LabVIEW programming fundamentals Data acquisition Object-oriented programming in LabVIEW Frameworks, including the Delacor Queued Message Handler (DQMH®) and Actor Framework Unit testing Enterprise and IoT messaging Programming in teams using the cloud

Introduction to Digital Design Using Digilent FPGA Boards

Advanced FPGA Design

Explore the complete process of developing systems based on field-programmable gate arrays (FPGAs), including the design of electronic circuits and the construction and debugging of prototype embedded devices Key Features Learn the basics of embedded systems and real-time operating systems Understand how FPGAs implement processing algorithms in hardware Design, construct, and debug custom digital systems from scratch using KiCad Book Description Modern digital devices
used in homes, cars, and wearables contain highly sophisticated computing capabilities composed of embedded systems that generate, receive, and process digital data streams at rates up to multiple gigabits per second. This book will show you how to use Field Programmable Gate Arrays (FPGAs) and high-speed digital circuit design to create your own cutting-edge digital systems. Architecting High-Performance Embedded Systems takes you through the fundamental concepts of embedded systems, including real-time operation and the Internet of Things (IoT), and the architecture and capabilities of the latest generation of FPGAs. Using powerful free tools for FPGA design and electronic circuit design, you'll learn how to design, build, test, and debug high-performance FPGA-based IoT devices. The book will also help you get up to speed with embedded system design, circuit design, hardware construction, firmware development, and debugging to produce a high-performance embedded device - a network-based digital oscilloscope. You'll explore techniques such as designing four-layer printed circuit boards with high-speed differential signal pairs and assembling the board using surface-mount components. By the end of the book, you'll have a solid understanding of the concepts underlying embedded systems and FPGAs and will be able to design and construct your own sophisticated digital devices. What you will learn Understand the fundamentals of real-time embedded systems and sensors Discover the capabilities of FPGAs and how to use FPGA development tools Learn the principles of digital circuit design and PCB layout with KiCad Construct high-speed circuit board prototypes at low cost Design and develop high-performance algorithms for FPGAs Develop robust, reliable, and efficient firmware in C Thoroughly test and debug embedded device hardware and firmware Who this book is for This book is for software developers, IoT engineers, and anyone who wants to understand the process of developing high-performance embedded systems. You'll also find this book useful if you want to learn about the fundamentals of FPGA development and all aspects of firmware development in C and C++. Familiarity with the C language, digital circuits, and electronic soldering is necessary to get started.

Gettting Started with Tiva ARM Cortex M4 Microcontrollers
FPGAs (Field-Programmable Gate Arrays) can be found in applications such as smart phones, mp3 players, medical imaging devices, and for aerospace and defense technology. FPGAs consist of logic blocks and programmable interconnects. This allows an engineer to start with a blank slate and program the FPGA for a specific task, for instance, digital signal processing, or a specific device, for example, a software-defined radio. Due to the short time to market and ability to reprogram to fix bugs without having to respin FPGAs are in increasingly high demand. This book is for the engineer that has not yet had any experience with this electrifying and growing field. The complex issue of FPGA design is broken down into four distinct phases - Design / Synthesis / Simulation / Place & Route. Numerous step-by-step examples along with source code accompany the discussion. A brief primer of one of the popular FPGA and hardware languages, VHDL, is incorporated for a simple yet comprehensive learning tool. While a general technology background is assumed, no direct hardware development understanding is needed. Also, included are details on tool-set up, verification techniques, and test benches. Reference material consists of a quick reference guide, reserved words, and common VHDL/FPGA terms. Learn how to design and develop FPGAs -- no prior experience necessary! Breaks down the complex design and development of FPGAs into easy-to-learn building blocks Contains examples, helpful tips, and step-by-step tutorials for synthesis, implementation, simulation, and programming phases

Beginning FPGA: Programming Metal

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

Design Recipes for FPGAs: Using Verilog and VHDL

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.
Read Online Programming Fpgas Getting Started With Verilog

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.

Data Parallel C++

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 FPGA design techniques at an engineer's fingertips Easy-to-find structure that allows the engineer to quickly locate the information to solve their FPGA design problem, and obtain the level of detail and understanding needed

Architecting High-Performance Embedded Systems
In this TAB book, bestselling electronics author Simon Monk shows maker-entrepreneurs how to use Fritzing’s open-source software and services to create electronics prototypes, design and manufacture printed circuit boards (PCBs), and bring professional-quality electronic products to market. Fritzing for Inventors: Take Your Electronics Project from Prototype to Product explains how to use this set of free, open-source electronics prototyping tools to lay out breadboards, create schematics, and design professional-quality printed circuit boards (PCBs). No engineering skills needed! Whether you’re a hobbyist, artist, inventor, or student, you’ll be able to develop a product from schematic to prototype to professional-quality printed circuit board, all from one easy-to-use software package. Fritzing works well with prototyping boards such as Arduino, Raspberry Pi, and BeagleBone. This DIY guide covers the whole lifecycle of product development for a hobbyist entrepreneur. It takes you from initial concept, to prototyping, to PCB production, to distribution. Along the way, it examines the sourcing of components, product testing, and even how to price products for wholesale and retail. Simon Monk is a bestselling TAB electronics author and popular presenter at MakerFaires. Well-illustrated tutorial with screen captures, easy-to-follow instructions, and step-by-step projects Describes an up-to-date contemporary approach to PCB design, including surface-mount designs Explains how to become a maker entrepreneur by using crowdfunding and indie marketplaces for technical products

Learning FPGAs

Field Programmable Gate Arrays (FPGAs) are currently recognized as the most suitable platform for the implementation of complex digital systems targeting an increasing number of industrial electronics applications. They cover a huge variety of application areas, such as: aerospace, food industry, art, industrial automation, automotive, biomedicine, process control, military, logistics, power electronics, chemistry, sensor networks, robotics, ultrasound, security, and artificial vision. This book first presents the basic architectures of the devices...
to familiarize the reader with the fundamentals of FPGAs before identifying and discussing new resources that extend the ability of the devices to solve problems in new application domains. Design methodologies are discussed and application examples are included for some of these domains, e.g., mechatronics, robotics, and power systems.

Designing Video Game Hardware in Verilog

FPGA brings high performance applications to market quickly – this book covers the many emerging platforms in a proven, effective manner.

Surreptitious Software

Market: electronics hobbyists and Tesla societies and websites Features 76 worksheets to simplify design The only book available to cover the Tesla coil in so much detail

Practical FPGA Programming in C

Hundreds of well-illustrated articles explore the most important fields of science. Based on content from the McGraw-Hill Concise Encyclopedia of Science & Technology, Fifth Edition, the most widely used and respected science reference of its kind in print, each of these subject-specific quick-reference guides features: * Detailed, well-illustrated explanations, not just definitions * Hundreds of concise yet authoritative articles in each volume * An easy-to-understand presentation, accessible and interesting to non-specialists * A portable, convenient format * Bibliographies, appendices, and other information supplement the articles

Embedded Systems Design with Platform FPGAs

Program Arduino with ease! Using clear, easy-to-follow examples, Programming Arduino: Getting Started with Sketches reveals the software side of Arduino and explains how to write well-crafted sketches using the modified C language of Arduino. No prior programming experience is required! The downloadable sample programs featured in the book can be used as-is or modified to suit your purposes. Understand Arduino hardware fundamentals Install the software, power it
up, and upload your first sketch

Learn C language basics
Write functions in Arduino sketches
Structure data using arrays and strings
Use Arduino's digital and analog inputs and outputs in your programs
Work with the Standard Arduino Library
Write sketches that can store data
Program LCD displays
Use an Ethernet shield to enable Arduino to function as a web server
Write your own Arduino libraries
In December 2011, Arduino 1.0 was released. This changed a few things that have caused two of the sketches in this book to break. The change that has caused trouble is that the classes 'Server' and 'Client' have been renamed to 'EthernetServer' and 'EthernetClient' respectively. To fix this: Edit sketches 10-01 and 10-02 to replace all occurrences of the word 'Server' with 'EthernetServer' and all occurrences of 'Client' with 'EthernetClient'. Alternatively, you can download the modified sketches for 10-01 and 10-02 from here: http://www.arduino-book.com/arduino-1-0

Make Great Stuff!

TAB, an imprint of McGraw-Hill Professional, is a leading publisher of DIY technology books for makers, hackers, and electronics hobbyists.

Fritzing for Inventors: Take Your Electronics Project from Prototype to Product

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.

Design for Embedded Image Processing on FPGAs

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

**FPGAs: Instant Access**

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 VHDL 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 core features of VHDL 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," VHDL BY EXAMPLE does for FPGA design.

**Verilog by Example**

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.

Real World FPGA Design with Verilog

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!
FPGAs for Software Programmers

UP-TO-DATE, TECHNICALLY ACCURATE COVERAGE OF ESSENTIAL TOPICS IN IMAGE AND VIDEO PROCESSING This is the first book to combine image and video processing with a practical MATLAB®-oriented approach in order to demonstrate the most important image and video techniques and algorithms. Utilizing minimal math, the contents are presented in a clear, objective manner, emphasizing and encouraging experimentation. The book has been organized into two parts. Part I: Image Processing begins with an overview of the field, then introduces the fundamental concepts, notation, and terminology associated with image representation and basic image processing operations. Next, it discusses MATLAB® and its Image Processing Toolbox with the start of a series of chapters with hands-on activities and step-by-step tutorials. These chapters cover image acquisition and
digitization; arithmetic, logic, and geometric operations; point-based, histogram-based, and neighborhood-based image enhancement techniques; the Fourier Transform and relevant frequency-domain image filtering techniques; image restoration; mathematical morphology; edge detection techniques; image segmentation; image compression and coding; and feature extraction and representation. Part II: Video Processing presents the main concepts and terminology associated with analog video signals and systems, as well as digital video formats and standards. It then describes the technically involved problem of standards conversion, discusses motion estimation and compensation techniques, shows how video sequences can be filtered, and concludes with an example of a solution to object detection and tracking in video sequences using MATLAB®. Extra features of this book include: More than 30 MATLAB® tutorials, which consist of step-by-step guides to exploring image and video processing techniques using MATLAB® Chapters supported by figures, examples, illustrative problems, and exercises Use of websites and an extensive list of bibliographical references This accessible text is ideal for upper-level undergraduate and graduate students in digital image and video processing courses, as well as for engineers, researchers, software developers, practitioners, and anyone who wishes to learn about these increasingly popular topics on their own.

McGraw-Hill Concise Encyclopedia of Engineering

With millions of new users and several new models, the Raspberry Pi ecosystem continues to expand—along with a lot of new questions about the Pi’s capabilities. The second edition of this popular cookbook provides more than 240 hands-on recipes for running this tiny low-cost computer with Linux, programming it with Python, and hooking up sensors, motors, and other hardware—including Arduino and the Internet of Things. Prolific hacker and author Simon Monk also teaches basic principles to help you use new technologies with Raspberry Pi as its ecosystem continues to develop. This cookbook is ideal for programmers and hobbyists familiar with the Pi through resources, including Getting Started with Raspberry Pi (O’Reilly). Python and other code examples from the book are available on GitHub. Set up your Raspberry Pi and connect to a network Work with its Linux-based operating system Program Raspberry Pi with
Python Give your Pi "eyes" with computer vision Control hardware through the GPIO connector Use Raspberry Pi to run different types of motors Work with switches, keypads, and other digital inputs Use sensors to measure temperature, light, and distance Connect to IoT devices in various ways Create dynamic projects with Arduino

Computer Architecture Tutorial Using an FPGA

What if you could use software to design hardware? Not just any hardware--imagine specifying the behavior of a complex parallel computer, sending it to a chip, and having it run on that chip--all without any manufacturing? With Field-Programmable Gate Arrays (FPGAs), you can design such a machine with your mouse and keyboard. When you deploy it to the FPGA, it immediately takes on the behavior that you defined. Want to create something that behaves like a display driver integrated circuit? How about a CPU with an instruction set you dreamed up? Or your very own Bitcoin miner You can do all this with FPGAs. Because you're not writing programs--rather, you're designing a chip whose sole purpose is to do what you tell it--it's faster than anything you can do in code. With Make: FPGAs, you'll learn how to break down problems into something that can be solved on an FPGA, design the logic that will run on your FPGA, and hook up electronic components to create finished projects.

FPGAs

This book makes powerful Field Programmable Gate Array (FPGA) and reconfigurable technology accessible to software engineers by covering different state-of-the-art high-level synthesis approaches (e.g., OpenCL and several C-to-gates compilers). It introduces FPGA technology, its programming model, and how various applications can be implemented on FPGAs without going through low-level hardware design phases. Readers will get a realistic sense for problems that are suited for FPGAs and how to implement them from a software designer’s point of view. The authors demonstrate that FPGAs and their programming model reflect the needs of stream processing problems much better than traditional CPU or GPU architectures, making them well-suited for a wide variety of systems, from embedded systems performing sensor processing to large setups for Big Data number crunching.
Read Online Programming Fpgas Getting Started With Verilog

This book serves as an invaluable tool for software designers and FPGA design engineers who are interested in high design productivity through behavioural synthesis, domain-specific compilation, and FPGA overlays. Introduces FPGA technology to software developers by giving an overview of FPGA programming models and design tools, as well as various application examples; Provides a holistic analysis of the topic and enables developers to tackle the architectural needs for Big Data processing with FPGAs; Explains the reasons for the energy efficiency and performance benefits of FPGA processing; Provides a user-oriented approach and a sense for where and how to apply FPGA technology.

FPGA Programming for Beginners

Embedded Systems Design with Platform FPGAs introduces professional engineers and students alike to system development using Platform FPGAs. The focus is on embedded systems but it also serves as a general guide to building custom computing systems. The text describes the fundamental technology in terms of hardware, software, and a set of principles to guide the development of Platform FPGA systems. The goal is to show how to systematically and creatively apply these principles to the construction of application-specific embedded system architectures. There is a strong focus on using free and open source software to increase productivity. Each chapter is organized into two parts. The white pages describe concepts, principles, and general knowledge. The gray pages provide a technical rendition of the main issues of the chapter and show the concepts applied in practice. This includes step-by-step details for a specific development board and tool chain so that the reader can carry out the same steps on their own. Rather than try to demonstrate the concepts on a broad set of tools and boards, the text uses a single set of tools (Xilinx Platform Studio, Linux, and GNU) throughout and uses a single developer board (Xilinx ML-510) for the examples. Explains how to use the Platform FPGA to meet complex design requirements and improve product performance. Presents both fundamental concepts together with pragmatic, step-by-step instructions for building a system on a Platform FPGA Includes detailed case studies, extended real-world examples, and lab exercises.
FPGA Architecture

Get started with FPGA programming using SystemVerilog, and develop real-world skills by building projects, including a calculator and a keyboard. Key Features: Explore different FPGA usage methods and the FPGA tool flow. Learn how to design, test, and implement hardware circuits using SystemVerilog. Build real-world FPGA projects such as a calculator and a keyboard using FPGA resources.

Description: Field Programmable Gate Arrays (FPGAs) have now become a core part of most modern electronic and computer systems. However, to implement your ideas in the real world, you need to get your head around the FPGA architecture, its toolset, and critical design considerations. FPGA Programming for Beginners will help you bring your ideas to life by guiding you through the entire process of programming FPGAs and designing hardware circuits using SystemVerilog. The book will introduce you to the FPGA and Xilinx architectures and show you how to work on your first project, which includes toggling an LED. You'll then cover SystemVerilog RTL designs and their implementations. Next, you'll get to grips with using the combinational Boolean logic design and work on several projects, such as creating a calculator and updating it using FPGA resources. Later, the book will take you through the advanced concepts of AXI and show you how to create a keyboard using PS/2. Finally, you'll be able to consolidate all the projects in the book to create a unified output using a Video Graphics Array (VGA) controller that you'll design. By the end of this SystemVerilog FPGA book, you'll have learned how to work with FPGA systems and be able to design hardware circuits and boards using SystemVerilog programming. What you will learn: Understand the FPGA architecture and its implementation. Get to grips with writing SystemVerilog RTL. Make FPGA projects using SystemVerilog programming. Work with computer math basics, parallelism, and pipelining. Explore the advanced topics of AXI and keyboard interfacing with PS/2. Discover how you can implement a VGA interface in your projects.

Who this book is for: This FPGA design book is for embedded system developers, engineers, and programmers who want to learn FPGA and SystemVerilog programming from scratch. FPGA designers looking to gain hands-on experience in working on real-world projects will also find this book useful.
Dr Donald Bailey starts with introductory material considering the problem of embedded image processing, and how some of the issues may be solved using parallel hardware solutions. Field programmable gate arrays (FPGAs) are introduced as a technology that provides flexible, fine-grained hardware that can readily exploit parallelism within many image processing algorithms. A brief review of FPGA programming languages provides the link between a software mindset normally associated with image processing algorithms, and the hardware mindset required for efficient utilization of a parallel hardware design. The design process for implementing an image processing algorithm on an FPGA is compared with that for a conventional software implementation, with the key differences highlighted. Particular attention is given to the techniques for mapping an algorithm onto an FPGA implementation, considering timing, memory bandwidth and resource constraints, and efficient hardware computational techniques. Extensive coverage is given of a range of low and intermediate level image processing operations, discussing efficient implementations and how these may vary according to the application. The techniques are illustrated with several example applications or case studies from projects or applications he has been involved with. Issues such as interfacing between the FPGA and peripheral devices are covered briefly, as is designing the system in such a way that it can be more readily debugged and tuned. Provides a bridge between algorithms and hardware Demonstrates how to avoid many of the potential pitfalls Offers practical recommendations and solutions Illustrates several real-world applications and case studies Allows those with software backgrounds to understand efficient hardware implementation Design for Embedded Image Processing on FPGAs is ideal for researchers and engineers in the vision or image processing industry, who are looking at smart sensors, machine vision, and robotic vision, as well as FPGA developers and application engineers. The book can also be used by graduate students studying imaging systems, computer engineering, digital design, circuit design, or computer science. It can also be used as supplementary text for courses in advanced digital design, algorithm and hardware implementation, and
FPGA Prototyping by Verilog Examples

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.

Programming FPGAs: Getting Started with Verilog

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.

Vhdl by Example

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.

FPGA Prototyping by SystemVerilog Examples

Get familiar and work with the basic and advanced Modeling types in Verilog HDL. Key Features: Learn about the step-wise process to use Verilog design tools such as Xilinx, Vivado, Cadence NC-SIM. Explore the various types of HDL.
and its need ● Learn Verilog HDL modeling types using examples ● Learn advanced concept such as UDP, Switch level modeling ● Learn about FPGA based prototyping of the digital system Description Hardware Description Language (HDL) allows analysis and simulation of digital logic and circuits. The HDL is an integral part of the EDA (electronic design automation) tool for PLDs, microprocessors, and ASICs. So, HDL is used to describe a Digital System. The combinational and sequential logic circuits can be described easily using HDL. Verilog HDL, standardized as IEEE 1364, is a hardware description language used to model electronic systems. This book is a comprehensive guide about the digital system and its design using various VLSI design tools as well as Verilog HDL. The step-wise procedure to use various VLSI tools such as Xilinx, Vivado, Cadence NC-SIM, is covered in this book. It also explains the advanced concept such as User Define Primitives (UDP), switch level modeling, reconfigurable computing, etc. Finally, this book ends with FPGA based prototyping of the digital system. By the end of this book, you will understand everything related to digital system design. What will you learn ● Implement Adder, Subtractor, Adder-Cum-Subtractor using Verilog HDL ● Explore the various Modeling styles in Verilog HDL ● Implement Switch level modeling using Verilog HDL ● Get familiar with advanced modeling techniques in Verilog HDL ● Get to know more about FPGA based prototyping using Verilog HDL Who this book is for Anyone interested in Electronics and VLSI design and want to learn Digital System Design with Verilog HDL will find this book useful. IC developers can also use this book as a quick reference for Verilog HDL fundamentals & features. Table of Contents 1. An Introduction to VLSI Design Tools 2. Need of Hardware Description Language (HDL) 3. Logic Gate Implementation in Verilog HDL 4. Adder–Subtractor Implementation Using Verilog HDL 5. Multiplexer/Demultiplexer Implementation in Verilog HDL 6. Encoder/Decoder Implementation Using Verilog HDL 7. Magnitude Comparator Implementation Using Verilog HDL 8. Flip-Flop Implementation Using Verilog HDL 9. Shift Registers Implementation Using Verilog HDL 10. Counter Implementation Using Verilog HDL 11. Shift Register Counter Implementation Using Verilog HDL 12. Advanced Modeling Techniques 13. Switch Level Modeling 14. FPGA Prototyping in Verilog HDL

FPGAs: World Class Designs

Page 24/25
All the design and development inspiration and direction a
hardware engineer needs in one blockbuster book! Clive "Max" Maxfield renowned author, columnist, and editor of PL DesignLine has selected the very best FPGA design material from the Newnes portfolio and has compiled it into this volume. The result is a book covering the gamut of FPGA design from design fundamentals to optimized layout techniques with a strong pragmatic emphasis. In addition to specific design techniques and practices, this book also discusses various approaches to solving FPGA design problems and how to successfully apply theory to actual design tasks. The material has been selected for its timelessness as well as for its relevance to contemporary FPGA design issues.

Contents

Chapter 1 Alternative FPGA Architectures
Chapter 2 Design Techniques, Rules, and Guidelines
Chapter 3 A VHDL Primer: The Essentials
Chapter 4 Modeling Memories
Chapter 5 Introduction to Synchronous State Machine Design and Analysis
Chapter 6 Embedded Processors
Chapter 7 Digital Signal Processing
Chapter 8 Basics of Embedded Audio Processing
Chapter 9 Basics of Embedded Video and Image Processing
Chapter 10 Programming Streaming FPGA Applications Using Block Diagrams In Simulink
Chapter 11 Ladder and functional block programming
Chapter 12 Timers

"Hand-picked content selected by Clive "Max" Maxfield, character, luminary, columnist, and author "Proven best design practices for FPGA development, verification, and low-power "Case histories and design examples get you off and running on your current project"