Memory Issues in Embedded Systems-on-Chip

Multiprocessor Systems-on-Chip

Introduction to Reconfigurable Computing

A Primer on Memory Consistency and Cache Coherence

Embedded Microprocessor Systems

Multicore Systems On-Chip: Practical Software/Hardware Design

The Future of Computing Performance

Designing Embedded Hardware and Implementation of Instruction Set Extension Identification for a Multiprocessor System-on-chip

Hardware/software Co-design Toolchain

System-level Modelling and Design Space Exploration for Multiprocessor Architectures

Readings in Hardware/software Co-design

Embedded Software for SoC Processor Design

Reconfigurable and Adaptive Computing

Electronic System-Level Hw/Sw Co-Design of Heterogeneous Multi-Processor Embedded Systems

Understanding the Linux Kernel

Dependable Embedded Systems

MicroC/OS-II Arm System-On-Chip Architecture, 2/E

Computer System Design

Multi-Processor Systems on Chip

Parallel Computer Architecture

Embedded Multiprocessors

On-Chip Instrumentation

Embedded Design and Programming of Multiprocessor System-on-Chip

Multi-Core Embedded Systems

Processor and System-on-Chip
Multi-Processor System-on-Chip 2

System on chips designs have evolved from fairly simple unicore, single memory designs to complex heterogeneous multicore SoC architectures consisting of a large number of IP blocks on the same silicon. To meet high computational demands posed by latest consumer electronic devices, most current systems are based on such paradigm, which represents a real revolution in many aspects in computing. The attraction of multicore processing for power reduction is compelling. By splitting a set of tasks among multiple processor cores, the operating frequency necessary for each core can be reduced, allowing to reduce the voltage on each core. Because dynamic power is proportional to the frequency and to the square of the voltage, we get a big gain, even though we may have more cores running. As more and more cores are integrated into these designs to share the ever increasing processing load, the main challenges lie in efficient memory hierarchy, scalable system interconnect, new programming paradigms, and efficient integration methodology for connecting such heterogeneous cores into a single system capable of leveraging their individual flexibility. Current design methods tend toward mixed HW/SW co-designs targeting multicore systems on-chip for specific applications. To decide on the lowest cost mix of cores, designers must iteratively map the device’s functionality to a particular HW/SW partition and target architectures. In addition, to connect the
heterogeneous cores, the architecture requires high performance complex communication architectures and efficient communication protocols, such as hierarchical bus, point-to-point connection, or Network-on-Chip. Software development also becomes far more complex due to the difficulties in breaking a single processing task into multiple parts that can be processed separately and then reassembled later. This reflects the fact that certain processor jobs cannot be easily parallelized to run concurrently on multiple processing cores and that load balancing between processing cores - especially heterogeneous cores - is very difficult.

Introduction to Reconfigurable Computing This Open Access book introduces readers to many new techniques for enhancing and optimizing reliability in embedded systems, which have emerged particularly within the last five years. This book introduces the most prominent reliability concerns from today’s points of view and roughly recapitulates the progress in the community so far. Unlike other books that focus on a single abstraction level such circuit level or system level alone, the focus of this book is to deal with the different reliability challenges across different levels starting from the physical level all the way to the system level (cross-layer approaches). The book aims at demonstrating how new hardware/software co-design solution can be proposed to effectively mitigate reliability degradation such as transistor aging, processor variation, temperature effects, soft errors, etc. Provides readers with latest insights into novel, cross-layer methods and models with respect to dependability of embedded systems; Describes cross-layer approaches that can leverage reliability through techniques that are pro-actively designed with respect to techniques at other layers; Explains run-time adaptation and concepts/means of self-
organization, in order to achieve error resiliency in complex, future many core systems.

A Primer on Memory Consistency and Cache Coherence This work is a comprehensive study of the field. It provides an entry point to the novice willing to move in the research field reconfigurable computing, FPGA and system on programmable chip design. The book can also be used as teaching reference for a graduate course in computer engineering, or as reference to advance electrical and computer engineers. It provides a very strong theoretical and practical background to the field, from the early Estrin's machine to the very modern architecture such as embedded logic devices.

Embedded Microprocessor Systems The end of dramatic exponential growth in single-processor performance marks the end of the dominance of the single microprocessor in computing. The era of sequential computing must give way to a new era in which parallelism is at the forefront. Although important scientific and engineering challenges lie ahead, this is an opportune time for innovation in programming systems and computing architectures. We have already begun to see diversity in computer designs to optimize for such considerations as power and throughput. The next generation of discoveries is likely to require advances at both the hardware and software levels of computing systems. There is no guarantee that we can make parallel computing as common and easy to use as yesterday's sequential single-processor computer systems, but unless we aggressively pursue efforts suggested by the recommendations in this book, it will be "game over" for growth in computing performance. If parallel programming and related software efforts fail to become widespread, the development of
exciting new applications that drive the computer industry will stall; if such innovation stalls, many other parts of the economy will follow suit. The Future of Computing Performance describes the factors that have led to the future limitations on growth for single processors that are based on complementary metal oxide semiconductor (CMOS) technology. It explores challenges inherent in parallel computing and architecture, including ever-increasing power consumption and the escalated requirements for heat dissipation. The book delineates a research, practice, and education agenda to help overcome these challenges. The Future of Computing Performance will guide researchers, manufacturers, and information technology professionals in the right direction for sustainable growth in computer performance, so that we may all enjoy the next level of benefits to society.

Multicore Systems On-Chip: Practical Software/Hardware Design The less-experienced engineer will be able to apply Ball's advice to everyday projects and challenges immediately with amazing results. In this new edition, the author has expanded the section on debug to include avoiding common hardware, software and interrupt problems. Other new features include an expanded section on system integration and debug to address the capabilities of more recent emulators and debuggers, a section about combination microcontroller/PLD devices, and expanded information on industry standard embedded platforms. * Covers all 'species' of embedded system chips rather than specific hardware * Learn how to cope with 'real world' problems * Design embedded systems products that are reliable and work in real applications
System-on-Chip (MPSoC) is the key component for complex applications. These applications put huge pressure on memory, communication devices and computing units. This book, presented in two volumes – Architectures and Applications – therefore celebrates the 20th anniversary of MPSoC, an interdisciplinary forum that focuses on multi-core and multi-processor hardware and software systems. It is this interdisciplinarity which has led to MPSoC bringing together experts in these fields from around the world, over the last two decades. Multi-Processor System-on-Chip 1 covers the key components of MPSoC: processors, memory, interconnect and interfaces. It describes advance features of these components and technologies to build efficient MPSoC architectures. All the main components are detailed: use of memory and their technology, communication support and consistency, and specific processor architectures for general purposes or for dedicated applications.

Designing Embedded Hardware A Multi-Processor System-on-Chip (MPSoC) is the key component for complex applications. These applications put huge pressure on memory, communication devices and computing units. This book, presented in two volumes – Architectures and Applications – therefore celebrates the 20th anniversary of MPSoC, an interdisciplinary forum that focuses on multi-core and multi-processor hardware and software systems. It is this interdisciplinarity which has led to MPSoC bringing together experts in these fields from around the world, over the last two decades. Multi-Processor System-on-Chip 2 covers application-specific MPSoC design, including compilers and architecture exploration. This second volume describes optimization methods, tools to optimize and port specific applications on MPSoC architectures. Details on compilation, power consumption and wireless communication are also
presented, as well as examples of modeling frameworks and CAD tools. Explanations of specific platforms for automotive and real-time computing are also included.

Design and Implementation of Instruction Set Extension Identification for a Multiprocessor System-on-chip Hardware/software Co-design Toolchain Modern embedded systems come with contradictory design constraints. On one hand, these systems often target mass production and battery-based devices, and therefore should be cheap and power efficient. On the other hand, they still need to show high (sometimes real-time) performance, and often support multiple applications and standards which requires high programmability. This wide spectrum of design requirements leads to complex heterogeneous System-on-Chip (SoC) architectures -- consisting of several types of processors from fully programmable microprocessors to configurable processing cores and customized hardware components, integrated on a single chip. This study targets such multiprocessor embedded systems and strives to develop algorithms, methods, and tools to deal with a number of fundamental problems which are encountered by the system designers during the early design stages.

System-level Modelling and Design Space Exploration for Multiprocessor Embedded System-on-chip Architectures This book gives a comprehensive introduction to the design challenges of MPSoC platforms, focusing on early design space exploration. It defines an iterative methodology to increase the abstraction level so that evaluation of design decisions can be performed earlier in the design process. These techniques enable exploration on the system level before undertaking time- and cost-intensive development.
Readings in Hardware/software Co-design This book outlines a set of issues that are critical to all of parallel architecture—communication latency, communication bandwidth, and coordination of cooperative work (across modern designs). It describes the set of techniques available in hardware and in software to address each issue and explore how the various techniques interact.

Modern VLSI Design Many modern computer systems, including homogeneous and heterogeneous architectures, support shared memory in hardware. In a shared memory system, each of the processor cores may read and write to a single shared address space. For a shared memory machine, the memory consistency model defines the architecturally visible behavior of its memory system. Consistency definitions provide rules about loads and stores (or memory reads and writes) and how they act upon memory. As part of supporting a memory consistency model, many machines also provide cache coherence protocols that ensure that multiple cached copies of data are kept up-to-date. The goal of this primer is to provide readers with a basic understanding of consistency and coherence. This understanding includes both the issues that must be solved as well as a variety of solutions. We present both high-level concepts as well as specific, concrete examples from real-world systems. This second edition reflects a decade of advancements since the first edition and includes, among other more modest changes, two new chapters: one on consistency and coherence for non-CPU accelerators (with a focus on GPUs) and one that points to formal work and tools on consistency and coherence.

Embedded Multiprocessor System-on-Chip for Access Network Processing This book describes analytical models and estimation methods to enhance performance estimation
of pipelined multiprocessor systems-on-chip (MPSoCs). A framework is introduced for both design-time and run-time optimizations. For design space exploration, several algorithms are presented to minimize the area footprint of a pipelined MPSoC under a latency or a throughput constraint. A novel adaptive pipelined MPSoC architecture is described, where idle processors are transitioned into low-power states at run-time to reduce energy consumption. Multi-mode pipelined MPSoCs are introduced, where multiple pipelined MPSoCs optimized separately are merged into a single pipelined MPSoC, enabling further reduction of the area footprint by sharing the processors and communication buffers. Readers will benefit from the authors’ combined use of analytical models, estimation methods and exploration algorithms and will be enabled to explore billions of design points in a few minutes.

Multiprocessor Systems-on-chips For Electrical Engineering and Computer Engineering courses that cover the design and technology of very large scale integrated (VLSI) circuits and systems. May also be used as a VLSI reference for professional VLSI design engineers, VLSI design managers, and VLSI CAD engineers. Modern VSLI Design provides a comprehensive “bottom-up” guide to the design of VSLI systems, from the physical design of circuits through system architecture with focus on the latest solution for system-on-chip (SOC) design. Because VSLI system designers face a variety of challenges that include high performance, interconnect delays, low power, low cost, and fast design turnaround time, successful designers must understand the entire design process. The Third Edition also provides a much more thorough discussion of hardware description languages, with introduction to both Verilog and VHDL. For that reason, this book presents the entire VSLI design process in a single
Multi-Processor System-on-Chip 1 To thoroughly understand what makes Linux tick and why it's so efficient, you need to delve deep into the heart of the operating system--into the Linux kernel itself. The kernel is Linux--in the case of the Linux operating system, it's the only bit of software to which the term "Linux" applies. The kernel handles all the requests or completed I/O operations and determines which programs will share its processing time, and in what order. Responsible for the sophisticated memory management of the whole system, the Linux kernel is the force behind the legendary Linux efficiency. The new edition of Understanding the Linux Kernel takes you on a guided tour through the most significant data structures, many algorithms, and programming tricks used in the kernel. Probing beyond the superficial features, the authors offer valuable insights to people who want to know how things really work inside their machine. Relevant segments of code are dissected and discussed line by line. The book covers more than just the functioning of the code, it explains the theoretical underpinnings for why Linux does things the way it does. The new edition of the book has been updated to cover version 2.4 of the kernel, which is quite different from version 2.2: the virtual memory system is entirely new, support for multiprocessor systems is improved, and whole new classes of hardware devices have been added. The authors explore each new feature in detail. Other topics in the book include: Memory management including file buffering, process swapping, and Direct memory Access (DMA) The Virtual Filesystem and the Second Extended Filesystem Process creation and scheduling Signals, interrupts, and the essential interfaces to device drivers Timing Synchronization in the kernel Interprocess Communication (IPC) Program execution
Understanding the Linux Kernel, Second Edition will acquaint you with all the inner workings of Linux, but is more than just an academic exercise. You'll learn what conditions bring out Linux's best performance, and you'll see how it meets the challenge of providing good system response during process scheduling, file access, and memory management in a wide variety of environments. If knowledge is power, then this book will help you make the most of your Linux system.

Chip Multiprocessor Architecture MicroC/OS II Second Edition describes the design and implementation of the MicroC/OS-II real-time operating system (RTOS). In addition to its value as a reference to the kernel, it is an extremely detailed and highly readable design study particularly useful to the embedded systems student. While documenting the design and implementation of the ker

Multiprocessor System-on-Chip Techniques for Optimizing Multiprocessor Implementations of Signal Processing Applications An indispensable component of the information age, signal processing is embedded in a variety of consumer devices, including cell phones and digital television, as well as in communication infrastructure, such as media servers and cellular base stations. Multiple programmable processors, along with custom hardware running in parallel, are needed to achieve the computation throughput required of such applications. Reviews important research in key areas related to the multiprocessor implementation of multimedia systems Embedded Multiprocessors: Scheduling and Synchronization, Second Edition presents architectures and design methodologies for parallel systems in embedded digital signal processing (DSP) applications. It discusses application modeling techniques for multimedia systems, the incorporation of interprocessor communication costs into
multiprocessor scheduling decisions, and a modeling methodology (the synchronization graph) for multiprocessor system performance analysis. The book also applies the synchronization graph model to develop hardware and software optimizations that can significantly reduce the interprocessor communication overhead of a given schedule. Chronicles recent activity dealing with single-chip multiprocessors and dataflow models This edition updates the background material on existing embedded multiprocessors, including single-chip multiprocessors. It also summarizes the new research on dataflow models for signal processing that has been carried out since the publication of the first edition. Harness the power of multiprocessors This book explores the optimization of interprocessor communication and synchronization in embedded multiprocessor systems. It shows you how to design multiprocessor computer systems that are streamlined for multimedia applications.

Embedded Software for SoC Here is an extremely useful book that provides insight into a number of different flavors of processor architectures and their design, software tool generation, implementation, and verification. After a brief introduction to processor architectures and how processor designers have sometimes failed to deliver what was expected, the authors introduce a generic flow for embedded on-chip processor design and start to explore the vast design space of on-chip processing. The authors cover a number of different types of processor core.

Processor Design Master's Thesis from the year 2007 in the subject Computer Science - Applied, grade: 1.0, Technical University of Munich (Institute for Informatics), 82 entries in the bibliography, language: English, abstract: Multicore systems are dominating the processor market; they enable
the increase in computing power of a single chip in proportion to the Moore's law-driven increase in number of transistors. A similar evolution is observed in the system-on-chip (SoC) market through the emergence of multi-processor SoC (MPSoC) designs. Nevertheless, MPSoCs introduce some challenges to the system architects concerning the efficient design of memory hierarchies and system interconnects while maintaining the low power and cost constraints. In this master thesis, I try to address some of these challenges: namely, non-cache coherent DMA transfers in MPSoCs, low instruction cache utilization by OS codes, and factors governing the system throughput in MPSoC designs. These issues are investigated using the empirical and simulation approaches. Empirical studies are conducted on the Danube platform. Danube is a commercial MPSoC platform that is based on two 32-bit MIPS cores and developed by Infineon Technologies AG for deployment in access network processing equipments such as integrated access devices, customer premises equipments, and home gateways. Simulation-based studies are conducted on a system based on the ARM MPCore architecture. Achievements include the successful implementation and testing of novel hardware and software solutions for improving the performance of non-cache coherent DMA transfers in MPSoCs. Several techniques for reducing the instruction cache miss rate are investigated and applied. Finally, a qualitative analysis of the impact of instruction reuse, number of cores, and memory bandwidth on the system throughput in MPSoC systems is presented.

Reconfigurable and Adaptive Computing The purpose of this book is to evaluate strategies for future system design in multiprocessor system-on-chip (MPSoC) architectures. Both hardware design and integration of new development tools
will be discussed. Novel trends in MPSoC design, combined with reconfigurable architectures are a main topic of concern. The main emphasis is on architectures, design-flow, tool-development, applications and system design.

Electronic System-Level Hw/SW Co-Design of Heterogeneous Multi-Processor Embedded Systems A Multi-Processor System-on-Chip (MPSoC) is the key component for complex applications. These applications put huge pressure on memory, communication devices and computing units. This book, presented in two volumes – Architectures and Applications – therefore celebrates the 20th anniversary of MPSoC, an interdisciplinary forum that focuses on multi-core and multi-processor hardware and software systems. It is this interdisciplinarity which has led to MPSoC bringing together experts in these fields from around the world, over the last two decades. Multi-Processor System-on-Chip 2 covers application-specific MPSoC design, including compilers and architecture exploration. This second volume describes optimization methods, tools to optimize and port specific applications on MPSoC architectures. Details on compilation, power consumption and wireless communication are also presented, as well as examples of modeling frameworks and CAD tools. Explanations of specific platforms for automotive and real-time computing are also included.

Understanding the Linux Kernel Reconfigurable computing techniques and adaptive systems are some of the most promising architectures for microprocessors. Reconfigurable and Adaptive Computing: Theory and Applications explores the latest research activities on hardware architecture for reconfigurable and adaptive computing systems. The first section of the book covers reconfigurable systems. The book presents a software and hardware codesign flow for coarse-
grained systems-on-chip, a video watermarking algorithm for the H.264 standard, a solution for regular expressions matching systems, and a novel field programmable gate array (FPGA)-based acceleration solution with MapReduce framework on multiple hardware accelerators. The second section discusses network-on-chip, including an implementation of a multiprocessor system-on-chip platform with shared memory access, end-to-end quality-of-service metrics modeling based on a multi-application environment in network-on-chip, and a 3D ant colony routing (3D-ACR) for network-on-chip with three different 3D topologies. The final section addresses the methodology of system codesign. The book introduces a new software–hardware codesign flow for embedded systems that models both processors and intellectual property cores as services. It also proposes an efficient algorithm for dependent task software–hardware codesign with the greedy partitioning and insert scheduling method (GPISM) by task graph.

Dependable Embedded Systems Authored by two of the leading authorities in the field, this guide offers readers the knowledge and skills needed to achieve proficiency with embedded software.

MicroC/OS-II

Arm System-On-Chip Architecture, 2/E Memory Issues in Embedded Systems-On-Chip: Optimizations and Explorations is designed for different groups in the embedded systems-on-chip arena. First, it is designed for researchers and graduate students who wish to understand the research issues involved in memory system optimization and exploration for embedded systems-on-chip. Second, it is intended for designers of embedded systems who are migrating from a
traditional micro-controllers centered, board-based design methodology to newer design methodologies using IP blocks for processor-core-based embedded systems-on-chip. Also, since Memory Issues in Embedded Systems-on-Chip: Optimization and Explorations illustrates a methodology for optimizing and exploring the memory configuration of embedded systems-on-chip, it is intended for managers and system designers who may be interested in the emerging capabilities of embedded systems-on-chip design methodologies for memory-intensive applications.

Computer System Design Modern electronic systems consist of a fairly heterogeneous set of components. Today, a single system can be constituted by a hardware platform, frequently composed of a mix of analog and digital components, and by several software application layers. The hardware can include several heterogeneous microprocessors (e.g. GPP, DSP, GPU, etc.), dedicated ICs (ASICs and/or FPGA s), memories, a set of local connections between the system components, and some interfaces between the system and the environment (sensors, actuators, etc.). Therefore, on the one hand, multi-processor embedded systems are capable of meeting the demand of processing power and flexibility of complex applications. On the other hand, such systems are very complex to design and optimize, so that the design methodology plays a major role in determining the success of the products. For these reasons, to cope with the increasing system complexity, the approaches typically used today are oriented towards co-design methodologies working at the higher levels of abstraction. Unfortunately, such methodologies are typically customized for the specific application, suffer of a lack of generality and still need a considerable effort when real-size project are envisioned. Therefore, there is still the need for a general methodology
Read Book Multiprocessor System On Chip Hardware Design And Tool Integration

able to support the designer during the high-level steps of a co-design flow, enabling an effective design space exploration before tackling the low-level steps and thus committing to the final technology. This should prevent costly redesign loops. In such a context, the work described in this book, composed of two parts, aims at providing models, methodologies and tools to support each step of the co-design flow of embedded systems implemented by exploiting heterogeneous multi-processor architectures mapped on distributed systems, as well as fully integrated onto a single chip. The first part focuses on issues like the analysis of system specification languages, and the analysis of existing system-level HW/SW co-simulation methodologies to support heterogeneous multi-processor architectures. The second part focuses mainly on Design Space Exploration, and it presents both some theoretical advancements with respect to the first part, and the development of a prototypal framework that provides practical exploitation of the proposed concepts.

Multi-Processor System-on-Chip 1 A Multi-Processor System-on-Chip (MPSoC) is the key component for complex applications. These applications put huge pressure on memory, communication devices and computing units. This book, presented in two volumes – Architectures and Applications – therefore celebrates the 20th anniversary of MPSoC, an interdisciplinary forum that focuses on multi-core and multi-processor hardware and software systems. It is this interdisciplinarity which has led to MPSoC bringing together experts in these fields from around the world, over the last two decades. Multi-Processor System-on-Chip 1 covers the key components of MPSoC: processors, memory, interconnect and interfaces. It describes advance features of these components and technologies to build efficient MPSoC architectures. All the main components are detailed: use of
memory and their technology, communication support and consistency, and specific processor architectures for general purposes or for dedicated applications.

Embedded Systems The next generation of computer system designers will be less concerned about details of processors and memories, and more concerned about the elements of a system tailored to particular applications. These designers will have a fundamental knowledge of processors and other elements in the system, but the success of their design will depend on the skills in making system-level tradeoffs that optimize the cost, performance and other attributes to meet application requirements. This book provides a new treatment of computer system design, particularly for System-on-Chip (SOC), which addresses the issues mentioned above. It begins with a global introduction, from the high-level view to the lowest common denominator (the chip itself), then moves on to the three main building blocks of an SOC (processor, memory, and interconnect). Next is an overview of what makes SOC unique (its customization ability and the applications that drive it). The final chapter presents future challenges for system design and SOC possibilities.

Multi-Processor System-on-Chip 2 Covers the significant embedded computing technologies—highlighting their applications in wireless communication and computing power. An embedded system is a computer system designed for specific control functions within a larger system—often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. Presented in three parts, Embedded Systems: Hardware, Design, and Implementation provides readers with an immersive introduction to this rapidly growing segment of the computer industry. Acknowledging the fact that
embedded systems control many of today's most common devices such as smart phones, PC tablets, as well as hardware embedded in cars, TVs, and even refrigerators and heating systems, the book starts with a basic introduction to embedded computing systems. It hones in on system-on-a-chip (SoC), multiprocessor system-on-chip (MPSoC), and network-on-chip (NoC). It then covers on-chip integration of software and custom hardware accelerators, as well as fabric flexibility, custom architectures, and the multiple I/O standards that facilitate PCB integration. Next, it focuses on the technologies associated with embedded computing systems, going over the basics of field-programmable gatearray (FPGA), digital signal processing (DSP) and application-specific integrated circuit (ASIC) technology, architectural support for on-chip integration of custom accelerators with processors, and O/S support for these systems. Finally, it offers full details on architecture, testability, and computer-aided design (CAD) support for embedded systems, soft processors, heterogeneous resources, and on-chip storage before concluding with coverage of software support—in particular, O/S Linux. Embedded Systems: Hardware, Design, and Implementation is an ideal book for design engineers looking to optimize and reduce the size and cost of embedded system products and increase their reliability and performance.

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

Multiprocessor Systems on Chip This book provides an in-
depth overview of on chip instrumentation technologies and various approaches taken in adding instrumentation to System on Chip (ASIC, ASSP, FPGA, etc.) design that are collectively becoming known as Design for Debug (DfD). On chip instruments are hardware based blocks that are added to a design for the specific purpose and improving the visibility of internal or embedded portions of the design (specific instruction flow in a processor, bus transaction in an on chip bus as examples) to improve the analysis or optimization capabilities for a SoC. DfD is the methodology and infrastructure that surrounds the instrumentation. Coverage includes specific design examples and discussion of implementations and DfD tradeoffs in a decision to design or select instrumentation or SoC that include instrumentation. Although the focus will be on hardware implementations, software and tools will be discussed in some detail.

Parallel Computer Architecture Simulation of computer architectures has made rapid progress recently. The primary application areas are hardware/software performance estimation and optimization as well as functional and timing verification. Recent, innovative technologies such as retargetable simulator generation, dynamic binary translation, or sampling simulation have enabled widespread use of processor and system-on-chip (SoC) simulation tools in the semiconductor and embedded system industries. Simultaneously, processor and SoC simulation is still a very active research area, e.g. what amounts to higher simulation speed, flexibility, and accuracy/speed trade-offs. This book presents and discusses the principle technologies and state-of-the-art in high-level hardware architecture simulation, both at the processor and the system-on-chip level.

Embedded Multiprocessors Millimeter-Wave Integrated
Circuits delivers a detailed overview of MMIC design, specifically focusing on designs for the millimeter-wave (mm-wave) frequency range. The scope of the book is broad, spanning detailed discussions of high-frequency materials and technologies, high-frequency devices, and the design of high-frequency circuits. The design material is supplemented as appropriate by theoretical analyses. The broad scope of the book gives the reader a good theoretical and practical understanding of mm-wave circuit design. It is best-suited for both undergraduate students who are reading or studying high frequency circuit design and postgraduate students who are specializing in the mm-wave field.

On-Chip Instrumentation Intelligent readers who want to build their own embedded computer systems--installed in everything from cell phones to cars to handheld organizers to refrigerators--will find this book to be the most in-depth, practical, and up-to-date guide on the market. Designing Embedded Hardware carefully steers between the practical and philosophical aspects, so developers can both create their own devices and gadgets and customize and extend off-the-shelf systems. There are hundreds of books to choose from if you need to learn programming, but only a few are available if you want to learn to create hardware. Designing Embedded Hardware provides software and hardware engineers with no prior experience in embedded systems with the necessary conceptual and design building blocks to understand the architectures of embedded systems. Written to provide the depth of coverage and real-world examples developers need, Designing Embedded Hardware also provides a road-map to the pitfalls and traps to avoid in designing embedded systems. Designing Embedded Hardware covers such essential topics as: The principles of developing computer hardware Core hardware designs
Assembly language concepts Parallel I/O Analog-digital conversion Timers (internal and external) UART Serial Peripheral Interface Inter-Integrated Circuit Bus Controller Area Network (CAN) Data Converter Interface (DCI) Low-power operation This invaluable and eminently useful book gives you the practical tools and skills to develop, build, and program your own application-specific computers.

Embedded Software Design and Programming of Multiprocessor System-on-Chip Chip multiprocessors - also called multi-core microprocessors or CMPs for short - are now the only way to build high-performance microprocessors, for a variety of reasons. Large uniprocessors are no longer scaling in performance, because it is only possible to extract a limited amount of parallelism from a typical instruction stream using conventional superscalar instruction issue techniques. In addition, one cannot simply ratchet up the clock speed on today's processors, or the power dissipation will become prohibitive in all but water-cooled systems. After a discussion of the basic pros and cons of CMPs when they are compared with conventional uniprocessors, this book examines how CMPs can best be designed to handle two radically different kinds of workloads that are likely to be used with a CMP: highly parallel, throughput-sensitive applications at one end of the spectrum, and less parallel, latency-sensitive applications at the other. Throughput-sensitive applications, such as server workloads that handle many independent transactions at once, require careful balancing of all parts of a CMP that can limit throughput, such as the individual cores, on-chip cache memory, and off-chip memory interfaces. Several studies and example systems, such as the Sun Niagara, that examine the necessary tradeoffs are presented here. In contrast, latency-sensitive applications - many desktop applications fall into this category - require a
focus on reducing inter-core communication latency and applying techniques to help programmers divide their programs into multiple threads as easily as possible. This book discusses many techniques that can be used in CMPs to simplify parallel programming, with an emphasis on research directions proposed at Stanford University. To illustrate the advantages possible with a CMP using a couple of solid examples, extra focus is given to thread-level speculation (TLS), a way to automatically break up nominally sequential applications into parallel threads on a CMP, and transactional memory. This model can greatly simplify manual parallel programming by using hardware - instead of conventional software locks - to enforce atomic code execution of blocks of instructions, a technique that makes parallel coding much less error-prone. Book jacket.

Multi-Core Embedded Systems

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

Pipelined Multiprocessor System-on-Chip for Multimedia Details a real-world product that applies a cutting-edge multi-core architecture. Increasingly demanding modern applications—such as those used in telecommunications networking and real-time processing of audio, video, and multimedia streams—require multiple processors to achieve computational performance at the rate of a few giga-operations per second. This necessity for speed and manageable power consumption makes it likely that the next generation of embedded processing systems will include hundreds of cores, while being increasingly programmable, blending processors and configurable hardware in a power-efficient manner. Multi-Core Embedded Systems presents a variety of perspectives that elucidate the technical challenges associated with such increased integration of homogeneous (processors) and heterogeneous multiple cores. It offers an analysis that industry engineers and professionals will need to understand the physical details of both software and hardware in embedded architectures, as well as their limitations and potential for future growth. Discusses the available programming models spread across different abstraction levels. The book begins with an overview of the evolution of multiprocessor architectures for embedded
applications and discusses techniques for autonomous power management of system-level parameters. It addresses the use of existing open-source (and free) tools originating from several application domains—such as traffic modeling, graph theory, parallel computing and network simulation. In addition, the authors cover other important topics associated with multi-core embedded systems, such as: Architectures and interconnects Embedded design methodologies Mapping of applications

Millimeter-Wave Integrated Circuits This title serves as an introduction and reference for the field, with the papers that have shaped the hardware/software co-design since its inception in the early 90s.

Programming Embedded Systems This book describes the architecture of microprocessors from simple in-order short pipeline designs to out-of-order superscalars.

Copyright code:  
f52c89c47c45dc11467a12b5af873772