Software Co-Design of Multimedia SOC Platform is one of the first of its kinds to provide a comprehensive overview of the design and implementation of the hardware and software of an SoC platform for multimedia applications. Topics covered in this book range from system level design methodology, multimedia algorithm implementation, a sub-word parallel, single-instruction-multiple data (SIMD) processor design, and its virtual platform implementation, to the development of an SIMD parallel compiler as well as a real-time operating system (RTOS). Hardware Software Co-Design of a Multimedia SOC Platform is written for practitioner engineers and technical managers who want to gain first hand knowledge about the hardware/software design process of an SoC platform. It offers both tutorial-like details to help readers become familiar with a diverse range of subjects, and in-depth analysis for advanced readers to pursue further. Presenting a comprehensive overview of the design automation algorithms, tools, and methodologies used to design integrated circuits, the Electronic Design Automation for Integrated Circuits Handbook is available in two volumes. The first volume, EDA for IC System Design, Verification, and Testing, thoroughly examines system-level design, microarchitectural design, logical verification, and testing. Chapters contributed by leading experts authoritatively discuss processor modeling and design tools, using performance metrics to select microprocessor cores for IC designs, design and verification languages, digital simulation, hardware acceleration and emulation, and much more. Save on the complete set.

The two-volume set LNCS 2686 and LNCS 2687 constitute the refereed proceedings of the 7th International Work-Conference on Artificial and Natural Neural Networks, IWANN 2003, held in Maó, Menorca, Spain in June 2003. The 197 revised papers presented were carefully reviewed and selected for inclusion in the book and address the following topics: mathematical and computational methods in neural modelling, neurophysiological data analysis and modelling, structural and functional models of neurons, learning and other plasticity phenomena, complex systems dynamics, cognitive processes and artificial intelligence, methodologies for net design, bio-inspired systems and engineering, and applications in a broad variety of fields.

Hardware/software co-verification is how to make sure that embedded system software works correctly with the hardware, and that the hardware has been properly designed to run the software successfully - before large sums are spent on prototypes or manufacturing. This is the first book to apply this verification technique to the rapidly growing field of embedded systems-on-a-chip (SoC). As traditional embedded system design evolves into single-chip design, embedded engineers must be armed with the necessary information to make educated decisions about which tools and methodology to deploy. SoC verification requires a mix of expertise from the disciplines of microprocessor and computer architecture, logic design and simulation, and C and Assembly language embedded software. Until now, the relevant information on how it all fits together has not been available. Andrews, a
recognized expert, provides in-depth information about how co-verification really works, how to be successful using it, and pitfalls to avoid. He illustrates these concepts using concrete examples with the ARM core - a technology that has the dominant market share in embedded system product design. The companion CD-ROM contains all source code used in the design examples, a searchable e-book version, and useful design tools. * The only book on verification for systems-on-a-chip (SoC) on the market * Will save engineers and their companies time and money by showing them how to speed up the testing process, while still avoiding costly mistakes * Design examples use the ARM core, the dominant technology in SoC, and all the source code is included on the accompanying CD-Rom, so engineers can easily use it in their own designs

Embedded systems are becoming one of the major driving forces in computer science. Furthermore, it is the impact of embedded information technology that dictates the pace in most engineering domains. Nearly all technical products above a certain level of complexity are not only controlled but increasingly even dominated by their embedded computer systems. Traditionally, such embedded control systems have been implemented in a monolithic, centralized way. Recently, distributed solutions are gaining increasing importance. In this approach, the control task is carried out by a number of controllers distributed over the entire system and connected by some interconnect network, like fieldbuses. Such a distributed embedded system may consist of a few controllers up to several hundred, as in today's top-range automobiles. Distribution and parallelism in embedded systems design increase the engineering challenges and require new development methods and tools. This book is the result of the International Workshop on Distributed and Parallel Embedded Systems (DIPES'98), organized by the International Federation for Information Processing (IFIP) Working Groups 10.3 (Concurrent Systems) and 10.5 (Design and Engineering of Electronic Systems). The workshop took place in October 1998 in Schloss Eringerfeld, near Paderborn, Germany, and the resulting book reflects the most recent points of view of experts from Brazil, Finland, France, Germany, Italy, Portugal, and the USA. The book is organized in six chapters: 'Formalisms for Embedded System Design': IP-based system design and various approaches to multi-language formalisms. 'Synthesis from Synchronous/Asynchronous Specification': Synthesis techniques based on Message Sequence Charts (MSC), StateCharts, and Predicate/Transition Nets. 'Partitioning and Load-Balancing': Application in simulation models and target systems. 'Verification and Validation': Formal techniques for precise verification and more pragmatic approaches to validation. 'Design Environments' for distributed embedded systems and their impact on the industrial state of the art. 'Object Oriented Approaches': Impact of OO-techniques on distributed embedded systems. This volume will be essential reading for computer science researchers and application developers.

Embedded and Networking Systems: Design, Software, and Implementation explores issues related to the design and synthesis of high-performance embedded computer systems and networks. The emphasis is on the fundamental concepts and analytical techniques that are applicable to a range of embedded and networking applications, rather than on specific embedded architectures, software development, or system-level integration. This system point of view guides designers in dealing with the trade-offs to optimize performance, power, cost, and other system-level non-functional requirements. The book brings together contributions by researchers and experts from around the world, offering a global view of the latest research and development in embedded and networking systems. Chapters highlight the evolution and trends in the field and supply a fundamental and analytical understanding of some underlying technologies. Topics include the co-design of embedded systems, code optimization for a variety of applications, power and performance trade-offs, benchmarks for evaluating embedded systems and their components, and mobile sensor network systems. The book also looks at novel applications such as mobile sensor systems and video networks. A comprehensive review of groundbreaking technology and applications, this book is a timely resource for system designers, researchers, and students interested in the possibilities of embedded and networking systems. It gives readers a better understanding of an emerging technology evolution that is helping drive telecommunications into the next decade.

This book contains a selection of refereed and revised papers of Intelligent Techniques and Applications track, and the Special Track on Intelligent Image Processing and Artificial Vision track originally presented at the International Symposium on Intelligent Systems Technologies and Applications (ISTA), August 10-13, 2015, Kochi, India. 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 FPGAs), 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 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 prototypical framework that provides practical exploitation of the proposed concepts. This book presents a number of issues of fundamental importance for the design of integrated hardware software products such as embedded, communication, and multimedia systems. This book is a comprehensive
introduction to the fundamentals of hardware/software co-design. Co-design is still a new field but one which has substantially matured over the past few years. This book, written by leading international experts, covers all the major topics including: fundamental issues in co-design; hardware/software co-synthesis algorithms; prototyping and emulation; target architectures; compiler techniques; specification and verification; system-level specification. Special chapters describe in detail several leading-edge co-design systems including Cosyma, LYCOS, and Cosmos. Introduction to Hardware-Software Co-Design contains sufficient material for use by teachers and students in an advanced course of hardware/software co-design. It also contains extensive explanation of the fundamental concepts of the subject and the necessary background to bring practitioners up-to-date on this increasingly important topic.

The new RISC-V Edition of Computer Organization and Design features the RISC-V open source instruction set architecture, the first open source architecture designed to be used in modern computing environments such as cloud computing, mobile devices, and other embedded systems. With the post-PC era now upon us, Computer Organization and Design moves forward to explore this generational change with examples, exercises, and material highlighting the emergence of mobile computing and the Cloud. Updated content featuring tablet computers, Cloud infrastructure, and the x86 (cloud computing) and ARM (mobile computing devices) architectures is included. An online companion Web site provides advanced content for further study, appendices, glossary, references, and recommended reading. Features RISC-V, the first such architecture designed to be used in modern computing environments, such as cloud computing, mobile devices, and other embedded systems Includes relevant examples, exercises, and material highlighting the emergence of mobile computing and the cloud HW/SW Co-Design for Heterogeneous Multi-Core Platforms describes the results and outcome of the FP6 project which focuses on the development of an integrated tool chain targeting a heterogeneous multi core platform comprising of a general purpose processor (ARM or powerPC), a DSP (the diopsis) and an FPGA. The tool chain takes existing source code and proposes transformations and mappings such that legacy code can easily be ported to a modern, multi-core platform. Downloadable software will be provided for simulation purposes.

This book describes how evolutionary algorithms (EA), including genetic algorithms (GA) and particle swarm optimization (PSO) can be utilized for solving multi-objective optimization problems in the area of embedded and VLSI system design. Many complex engineering optimization problems can be modelled as multi-objective formulations. This book provides an introduction to multi-objective optimization using meta-heuristic algorithms, GA and PSO and how they can be applied to problems like hardware/software partitioning in embedded systems, circuit partitioning in VLSI, design of operational amplifiers in analog VLSI, design space exploration in high-level synthesis, delay fault testing in VLSI testing and scheduling in heterogeneous distributed systems. It is shown how, in each case, the various aspects of the EA, namely its representation and operators like crossover, mutation, etc, can be separately formulated to solve these problems. This book is intended for design engineers and researchers in the field of VLSI and embedded system design. The book introduces the multi-objective GA and PSO in a simple and easily understandable way that will appeal to introductory readers.

This book is concerned with studying the co-design methodology in general, and how to determine the more suitable interface mechanism in a co-design system in particular. This is based on the characteristics of the application and those of the target architecture of the system. Guidelines are provided to support the designer's choice of the interface mechanism. Some new trends in co-design and system acceleration are also introduced.

Rapid energy estimation for energy efficient applications using field-programmable gate arrays (FPGAs) remains a challenging research topic. Energy dissipation and efficiency have prevented the widespread use of FPGA devices in embedded systems, where energy efficiency is a key performance metric. Helping overcome these challenges, Energy Efficient Hardware-Software Co-Synthesis Using Reconfigurable Hardware offers solutions for the development of energy efficient applications using FPGAs. The book integrates various high-level abstractions for describing hardware and software platforms into a single, consistent application development framework, enabling users to construct, simulate, and debug systems. Based on these high-level concepts, it proposes an energy performance modeling technique to capture the energy dissipation behavior of both the reconfigurable hardware platform and the target applications running on it. The authors also present a dynamic programming-based algorithm to optimize the energy performance of an application running on a reconfigurable hardware platform. They then discuss an instruction-level energy estimation technique and a domain-specific modeling technique to provide rapid and fairly accurate energy estimation for hardware-software co-designs using reconfigurable hardware. The text concludes with example designs and illustrative examples that show how the proposed co-synthesis techniques lead to a significant amount of energy reduction. This book explores the advantages of using reconfigurable hardware for application development and looks ahead to future research directions in the field. It outlines the range of aspects and steps that lead to an energy efficient hardware-software application synthesis using FPGAs.

The proceedings of the September 1994 workshop comprise 28 technical papers that represent several important trends in co-design research: use of design case studies to drive research; algorithms for hardware- software partitioning; algorithms for system verification and validation; and a continuing interest in design representations. No index. Annotation copyright by Book News, Inc., Portland, OR.

SC15: The International Conference for High Performance Computing, Networking, Storage and Analysis Nov 15, 2015-Nov 20, 2015 Austin, USA. You can view more information about this proceeding and all of ACMs other published conference proceedings from the ACM Digital Library: http://www.acm.org/dl.

Current practice dictates the separation of the hardware and software development paths early in the design cycle. These paths remain independent with very little interaction occurring between them until system integration. In particular, hardware is often specified without fully appreciating the computational requirements of the software. Also, software development does not influence hardware development and does not track changes made during the hardware design phase. Thus, the ability to explore hardware/software tradeoffs is restricted, such as the movement of functionality from the software domain to the hardware domain (and vice-versa) or the modification of the hardware/software interface. As a result, problems that are encountered during system integration may require modification of the software and/or hardware, resulting in potentially significant cost increases and schedule overruns. To address the problems described above, a cooperative design approach, one that utilizes a unified view of hardware and software, is described. This approach is called hardware/software co-design. The Codeign of Embedded Systems develops several fundamental hardware/software codeign concepts and a methodology that supports them.
representation, referred to as a decomposition graph, is presented which can be used to describe hardware or software using either functional abstractions or data abstractions. Using a unified representation based on functional abstractions, an abstract hardware/software model has been implemented in a common simulation environment named ADEPT (Advanced Design Environment Prototyping Tool). This model permits early hardware/software evaluation and tradeoff exploration. Techniques have been developed which support the identification of software bottlenecks and the evaluation of design alternatives with respect to multiple metrics. The application of the model is demonstrated on several examples. A unified representation based on data abstractions is also explored. This work leads to investigations regarding the application of object-oriented techniques to hardware design. The Codesign of Embedded Systems: A Unified Hardware/Software Representation describes a novel approach to a topic of immense importance to CAD researchers and designers alike.

This text on hardware and software co-design covers such topics as: system-level modelling; partitioning; communication and interface synthesis; co-simulation; scheduling; case studies; system on chip; and system level modelling.

Concurrent design, or co-design of hardware and software is extremely important for meeting design goals, such as high performance, that are the key to commercial competitiveness. Hardware/Software Co-Design covers many aspects of the subject, including methods and examples for designing: (1) general purpose and embedded computing systems based on instruction set processors; (2) telecommunication systems using general purpose digital signal processors as well as application specific instruction set processors; (3) embedded control systems and applications to automotive electronics. The book also surveys the areas of emulation and prototyping systems with field programmable gate array technologies, hardware/software synthesis and verification, and industrial design trends. Most contributions emphasize the design methodology, the requirements and state of the art of computer aided co-design tools, together with current design examples.

When I attended college we studied vacuum tubes in our junior year. At that time an average radio had ?ve vacuum tubes and better ones even seven. Then transistors appeared in 1960s. A good radio was judged to be one with more pententtransistors. Latergoodradioshad15–20transistors and after that everyone stopped counting transistors. Today modern processors runing personal computers have over 1milliontransistorsandmoremillionswillbeadddeeveryear. The difference between 20 and 20M is in complexity, methodology and business models. Designs with 20 tr- sistors are easily generated by design engineers without any tools, whilst designs with 20M transistors can not be done by humans in reasonable time without the help of Prof. Dr. Gajski demonstrates the Y-chart automation. This difference in complexity introduced a paradigm shift which required sophisticated methods and tools, and introduced design automation into design practice. By the decomposition of the design process into many tasks and abstraction levels the methodology of designing chips or systems has also evolved. Similarly, the business model has changed from vertical integration, in which one company did all the tasks from product specification to manufacturing, to globally distributed, client server production in which most of the design and manufacturing tasks are outsourced. This handbook presents fundamental knowledge on the hardware/software (HW/SW) codesign methodology. Contributing expert authors look at key techniques in the design flow as well as selected codesign tools and design environments, building on basic knowledge to consider the latest techniques. The book enables readers to gain real benefits from the HW/SW codesign methodology through explanations and case studies which demonstrate its usefulness. Readers are invited to follow the progress of design techniques through this work, which assists readers in following current research directions and learning about state-of-the-art techniques. Students and researchers will appreciate the wide spectrum of subjects that belong to the design methodology from this handbook.

Hardware/software co-design requires the smooth interaction of both disciplines and the 16 selected papers from the March 1996 workshop summarize some of the most important aspects of this growing area of computer science. The sessions focus on issues in transformation based co-design and communicat

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 issues and explore how the various techniques interact.

Hardware/Software Co-DesignPrinciples and PracticeSpringer Science & Business Media

The design process of embedded systems has changed substantially in recent years. One of the main reasons for this change is the pressure to shorten time-to-market when designing digital systems. To shorten the product cycles, programmable processes are used to implement more and more functionality of the embedded system. Therefore, nowadays, embedded systems are very often implemented by heterogeneous systems consisting of ASICs, processors, memories and peripherals. As a consequence, the research topic of hardware/software co-design, dealing with the problems of designing these heterogeneous systems, has gained great importance. Hardware/Software Co-design for Data Flow Dominated Embedded Systems introduces the different tasks of hardware/software co-design including system specification, hardware/software partitioning, co-synthesis and co-simulation. The book summarizes and classifies state-of-the-art co-design tools and methods for these tasks. In addition, the co-design tool COOL is presented which solves the co-design tasks for the class of data-flow dominated embedded systems. In Hardware/Software Co-design for Data Flow Dominated Embedded Systems the primary emphasis has been put on the hardware/software partitioning and the co-synthesis phase and their coupling. In contrast to many other publications in this area, a mathematical formulation of the hardware/software partitioning problem is given. This problem formulation supports target architectures consisting of multiple processors and multiple ASICs. Several novel approaches are presented and compared for solving the partitioning problem, including an MILP approach, a heuristic solution and an approach based on genetic algorithms. The co-synthesis phase is based on the idea of controlling the system by means of a static run-time scheduler implemented in hardware. New algorithms are introduced which generate a complete set of hardware and software specifications required to implement heterogeneous systems. All of these techniques are described in detail and exemplified. Hardware/Software Co-design for Data Flow Dominated Embedded Systems is intended to serve students and researchers working on hardware/software co-design. At the same time the variety of presented techniques automating the design tasks of hardware/software systems will be of interest to industrial engineers and designers of digital systems. From the foreword by Peter Marwedel: Niemann's method should be known by all persons working in the field. Hence, I recommend this book for everyone who is interested in hardware/software co-design.

Hardware/Software Co-Design for Data Flow Dominated Embedded Systems introduces the set of emerging techniques which allows for the simultaneous design of Hardware and Software. In many cases where the application is very demanding in terms of various performances (time, surface, power consumption), trade-offs between dedicated hardware and dedicated software are becoming increasingly difficult to decide upon in the early stages of a design. Verification techniques - such as simulation or proof techniques - that have proven necessary in the hardware design must be dramatically adapted to the simultaneous verification of Software and Hardware. Describing the latest tools available for both Co-Design and Co-Verification of systems, Hardware/Software Co-Design and Co-Verification offers a complete look at this evolving set of procedures for CAD environments. The book
considers all trade-offs that have to be made when co-designing a system. Several models are presented for determining the optimum solution to any co-design problem, including partitioning, architecture synthesis and code generation. When deciding on trade-offs, one of the main factors to be considered is the flow of communication, especially to and from the outside world. This involves the modeling of communication protocols. An approach to the synthesis of interface circuits in the context of co-design is presented. Other chapters present a co-design oriented flexible component data-base and retrieval methods; a case study of an ethernet bridge, designed using LOTOS and co-design methodologies and finally a programmable user interface based on monitors. Hardware/Software Co-Design and Co-Verification will help designers and researchers to understand these latest techniques in system design and as such will be of interest to all involved in embedded system design.

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.

Copyright: c44b95387a4722d950a81b0be97db17e