Fpgas For Software Programmers

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

This textbook for courses in Digital Systems Design introduces students to the fundamental hardware used in modern computers. Coverage includes both the classical approach to digital system design (i.e., pen and paper) in addition to the modern hardware description language (HDL) design approach (computer-based). Using this textbook enables readers to design digital systems using the modern HDL approach, but they have a broad foundation of knowledge of the underlying hardware and theory of their designs. This book is designed to match the way the material is actually taught in the classroom. Topics are presented in a manner which builds foundational knowledge before moving onto advanced topics. The author has designed the presentation with learning Goals and assessment at its core. Each section addresses a specific learning outcome that the student should be able to “do” after its completion. The concept checks and exercise problems provide a rich set of assessment tools to measure student performance on each outcome.

This book provides readers with an overview of the architectures, programming frameworks, and hardware accelerators for typical cloud computing applications in data centers. The authors present the most recent and promising solutions, using hardware accelerators to provide high throughput, reduced latency and higher energy efficiency compared to current servers based on commodity processors. Readers will benefit from state-of-the-art information regarding application requirements in contemporary data centers, computational complexity of typical tasks in cloud computing, and a programming framework for the efficient utilization of the hardware accelerators.

This book constitutes the proceedings of the 14th International Conference on Applied Reconfigurable Computing, ARC 2018, held in Santorini, Greece, in May 2018. The 29 full papers and 22 short presented in this volume were carefully reviewed and selected from 78 submissions. In addition, the volume contains 9 contributions from research projects. The papers were organized in topical sections named: machine learning and neural networks; FPGA-based design and CGRA optimizations; applications and surveys; fault-tolerance, security and communication architectures; reconfigurable and adaptive architectures; design methods and fast prototyping; FPGA-based design and applications; and special session: research projects.

Dijkstra once wrote that computer science is no more about computers than astronomy is about telescopes. Despite the many incredible advances in c- puter science from times that predate practical mechanical computing, there is still a myriad of fundamental questions in understanding the interface between computers and the rest of the world. Why is it still hard to mechanize many tasks that seem to be fundamentally routine, even as we see ever-increasing - pacity for raw mechanical computing? The disciplined study of domain-speci?c languages (DSLs) is an emerging area in computer science, and is one which has the potential to revolutionize the ?eld, and bring us closer to answering this question. DSLs are formalisms that have four general characteristics. – They relate to a well-de?ned domain of discourse, be it controlling tra?c lights or space ships. – They have well-de?ned notation, such as the ones that exist for prescribing music, dance routines, or strategy in a football game. – The informal or intuitive meaning of the notation is clear. This can easily be overlooked, especially since intuitive meaning can be expressed by many di?erent notations that may be received very di?erently by users. – The formal meaning is clear and mechanizable, as is, hopefully, the case for the instructions we give to our bank or to a merchant online.

Tactile Internet with Human-in-the-Loop describes the change from the current Internet, which focuses on the democratization of information independent of location or time, to the Tactile Internet, which democratizes skills to promote equity that is independent of age, gender, sociocultural background or physical limitations. The book promotes the concept of the Tactile Internet for remote closed-loop human-machine interaction and describes the main challenges and key technologies. Current standardization activities in the field for IEEE and IETF are also described, making this book an ideal resource for researchers, graduate students, and industry R&D engineers in communications engineering, electronic engineering, and computer engineering. Provides a comprehensive reference that addresses all aspects of the Tactile Internet – technologies, engineering challenges, use cases and standards Written by leading researchers in the field Presents current standardizations surrounding the IETF and the IEEE Contains use cases that illustrate practical applications

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 hardward/software and DSP-based design flows, along with innovative new devices such as field programmable node arrays (FPNAs). Clive "Max" Maxfield is a bestselling author and engineer with a large following in the electronic design automation (EDA)and embedded systems industry. In this comprehensive book, he covers all the issues of interest to designers working with, or contemplating a move to, FPGAs in their product designs. While other books cover fragments of FPGA technology or applications this is the first to focus exclusively and comprehensively on FPGA use for embedded systems. First book to focus exclusively and comprehensively on FPGA use in embedded designs World-renowned best-selling author Will help engineers get familiar and succeed with this new technology by providing much-needed advice on choosing the right FPGA for any design project

This book covers the latest approaches and results from reconfigurable computing architectures employed in the finance domain. So-called field-programmable gate arrays (FPGAs) have already shown to outperform standard CPU- and GPU-based computing architectures by far, saving up to 99% of energy depending on the compute tasks. Renowned authors from financial mathematics, computer architecture and finance business introduce the readers into today’s challenges in finance IT, illustrate the most advanced approaches and use cases and present

currently known methodologies for integrating FPGAs in finance systems together with latest results. The complete algorithm-to-hardware flow is covered holistically, so this book serves as a hands-on guide for IT managers, researchers and quants/programmers who think about integrating FPGAs into their current IT systems.

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.

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

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, verifaction 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

This is the first book to focus on designing run-time reconfigurable systems on FPGAs, in order to gain resource and power efficiency, as well as to improve speed. Case studies in partial reconfiguration guide readers through the FPGA jungle, straight toward a working system. The discussion of partial reconfiguration is comprehensive and practical, with models introduced together with methods to implement efficiently the corresponding systems. Coverage includes concepts for partial module integration and corresponding communication architectures, floorplanning of the on-FPGA resources, physical implementation aspects starting from constraining primitive placement and routing all the way down to the bitstream required to configure the FPGA, and verification of reconfigurable systems.

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

Embedded Anwendungen gewinnen immer mehr an Bedeutung. Solche Anwendungen bestehen aus der eigentlichen Software und der Hardware, die von der Software zum Funktionieren benötigt wird. Der Softwareentwickler muss also nicht nur wissen, wie die Software programmiert werden muss, er muss auch wissen, wie er die Hardwareplattform für die Software erstellen kann. Üblicherweise werden FPGAs zum Entwickeln und Testen von Mikroprozessoranwendungen verwendet, weil mit ihnen die benötigte Hardware leicht erstellt werden kann. Diese Arbeit stellt ein Programm vor, dass die benötigten Hardwarekomponenten aus dem Quelltext der Mikroprozessoranwendung aus speziellen Kommentaren ausliest. Das Programm erzeugt daraus

eine Konfiguration für eine FPGA-Toolchain, welche die von der Mikroprozessoranwendung benötigten Komponenten enthält. Wenn die Konfigurationsdatenbank und das Programm für die jeweilige FPGA-Toolchain angepasst sind, benötigt der Mikroprozessorentwickler kein spezielles Wissen zum Erstellen der Hardwareplattform.*****Embedded applications are becoming ever increasingly important. These applications consist of the software and the hardware required for the software to run. Therefore, the software developer is required to know how to program the software and how to build a hardware platform that is suitable for the application. Usually, FPGAs are used to develop and test embedded applications as they allow the easy creation of different hardware. This work presents software that extracts the required components from the microprocessor application's source code via special comments. The software creates a configuration that can be used as input to the FPGA's toolchain so that the microprocessor application can run on the programmed FPGA. Therefore, if the configuration database and the program are configured for the specific toolchain of the FPGA, the software developer does not need any special knowledge of FPGA design.

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. 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.

Reconfigurable Computing marks a revolutionary and hot topic that bridges the gap between the separate worlds of hardware and software design— the key feature of reconfigurable computing is its groundbreaking ability to perform computations in hardware to increase performance while retaining the flexibility of a software solution. Reconfigurable computers serve as affordable, fast, and accurate tools for developing designs ranging from single chip architectures to multi-chip and embedded systems. Scott Hauck and Andre DeHon have assembled a group of the key experts in the fields of both hardware and software computing to provide an introduction to the entire range of issues relating to reconfigurable computing. FPGAs (field programmable gate arrays) act as the “computing vehicles to implement this powerful technology. Readers will be guided into adopting a completely new way of handling existing design concerns and be able to make use of the vast opportunities possible with reconfigurable logic in this rapidly evolving field. Designed for both hardware and software programmers Views of reconfigurable programming beyond standard programming languages Broad set of case studies demonstrating how to use FPGAs in novel and efficient ways

The state of the art in design and development flows for FPGAs are not sufficiently mature to allow programmers to implement their applications through traditional software development flows. The stipulation of synthesis as well as the requirement of background knowledge on the FPGAs' low-level physical hardware structure are major challenges that prevent programmers from using FPGAs. The reconfigurable computing community is seeking solutions to raise the level of design abstraction at which programmers must operate, and move the synthesis process out of the programmers' path through the use of overlays. A recent approach, Just-In-Time Assembly (JITA), was proposed that enables hardware accelerators to be assembled at runtime, all from within a traditional software compilation flow. The JITA approach presents a promising path to constructing hardware designs on FPGAs using pre-synthesized parallel programming patterns, but suffers from two major limitations. First, all variant programming patterns must be pre-synthesized. Second, conditional operations are not supported. In this thesis, I present a new reconfigurable overlay, URUK, that overcomes the two limitations imposed by the JITA approach. Similar to the original JITA approach, the proposed URUK overlay allows hardware accelerators to be constructed on FPGAs through software compilation flows. To this basic capability, URUK adds additional support to enable the assembly of presynthesized fine-grained computational operators to be assembled within the FPGA. This thesis provides analysis of URUK from three different perspectives; utilization, performance, and productivity. The analysis includes comparisons against High-Level Synthesis (HLS) and the state of the art approach to creating static overlays. The tradeoffs conclude that URUK can achieve approximately equivalent performance for algebra operations compared to HLS custom accelerators, which are designed with simple experience on FPGAs. Further, URUK shows a high degree of flexibility for runtime placement and routing of the primitive operations. The analysis shows how this flexibility can be leveraged to reduce communication overhead among tiles, compared to traditional static overlays. The results also show URUK can enable software programmers without any hardware skills to create hardware accelerators at productivity levels consistent with software development and compilation.

Real-time testing and simulation of open- and closed-loop radio frequency (RF) systems for signal generation, signal analysis and digital signal processing require deterministic, low-latency, high-throughput capabilities afforded by user reconfigurable field programmable gate arrays (FPGAs). This comprehensive book introduces LabVIEW FPGA, provides best practices for multi-FPGA solutions, and guidance for developing high-throughput, low-latency FPGA based RF systems. Written by a recognized expert with a wealth of real-world experience in the field, this is the first book written on the subject of FPGAs for radar and other RF applications.

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.

Concurrent and parallel systems are intrinsic to the technology which underpins almost every aspect of our lives today. This book presents the combined post-proceedings for two important conferences on concurrent and parallel systems: Communicating Process Architectures 2017, held in Sliema, Malta, in August 2017, and Communicating Process Architectures 2018, held in

Dresden, Germany, in August 2018. CPA 2017: Fifteen papers were accepted for presentation and publication, they cover topics including mathematical theory, programming languages, design and support tools, verification, and multicore infrastructure and applications ranging from supercomputing to embedded. A workshop on domain-specific concurrency skeletons and the abstracts of eight fringe presentations reporting on new ideas, work in progress or interesting thoughts associated with concurrency are also included in these proceedings. CPA 2018: Eighteen papers were accepted for presentation and publication, they cover topics including mathematical theory, design and programming language and support tools, verification, multicore run-time infrastructure, and applications at all levels from supercomputing to embedded. A workshop on translating CSP-based languages to common programming languages and the abstracts of four fringe presentations on work in progress, new ideas, as well as demonstrations and concerns that certain common practices in concurrency are harmful are also included in these proceedings. The book will be of interest to all those whose work involves concurrent and parallel systems.

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

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 Book 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 serial interfaces 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 serial interfaces 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.

Programmers Guide for FPGA and Verilog is specifically written with a software developer in mind. The book is an invaluable resource for understanding the power and applicability of FPGAs and how to utilize the Verilog language to develop fast, efficient, parallel designs for real world applications. Using examples of functional code, it provides the building blocks, and discusses the pitfalls of FPGA development; enabling the developer to quickly become proficient and bypass many of the common FPGA mistakes.This book is written to help a software developer with the following:* Understand differences inherent in a FPGA * Understand Verilog's simulation and synthesis constructs* Point out pitfalls that make the transition to FPGA development difficult* Design parallel applications that utilize the power of the FPGA* Provide Verilog coding examples for commonly used programming concepts* Describe best practices for improving readability and maintainability

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

This book includes high impact papers presented at the International Conference on Communication, Computing and Electronics Systems 2019, held at the PPG Institute of Technology, Coimbatore, India, on 15-16 November, 2019. Discussing recent trends in cloud computing, mobile computing, and advancements of electronics systems, the book covers topics such as automation, VLSI, embedded systems, integrated device technology, satellite communication, optical communication, RF communication, microwave engineering, artificial intelligence, deep learning, pattern recognition, Internet of Things, precision models, bioinformatics, and healthcare informatics.

Publisher's Note: Products purchased from Third Party sellers are not guaranteed by the publisher for quality, authenticity, or access to any online entitlements included with the

product. Take your creations to the next level with FPGAs and Verilog

It is a pleasure to present the proceedings of the 22nd European Conference on Object-Oriented Programming (ECOOP 2008) held in Paphos, Cyprus. The conference continues to serve a broad object-oriented community with a tech- cal program spanning theory and practice and a healthy mix of industrial and academic participants. This year a strong workshop and tutorial program c- plementedthemaintechnicaltrack.Wehad13workshopsand8tutorials,aswell as the co-located Dynamic Language Symposium (DLS). Finally, the program was rounded out with a keynote by Rachid Guerraoui and a banquet speech by James Noble. As in previous years, two Dahl-Nygaard awards were selected by AITO, and for the ?rst time, the ECOOP Program Committee gave a best paper award. Theproceedingsinclude27papersselectedfrom138submissions.Thepapers werereviewed in a single-blind process with three to ?ve reviews per paper. P- liminaryversionsofthereviewsweremadeavailabletotheauthorsaweekbefore the PC meeting to allow for short (500 words or less) author responses. The - sponses were discussed at the PC meeting and were instrumental in reaching decisions. The PC discussions followed Oscar Nierstrasz'Champion pattern. PC papers had ?ve reviews and were held at a higher standard.

The year 2019 marked four decades of cluster computing, a history that began in 1979 when the first cluster systems using Components Off The Shelf (COTS) became operational. This achievement resulted in a rapidly growing interest in affordable parallel computing for solving compute intensive and large scale problems. It also directly lead to the founding of the Parco conference series. Starting in 1983, the International Conference on Parallel Computing, ParCo, has long been a leading venue for discussions of important developments, applications, and future trends in cluster computing, parallel computing, and high-performance computing. ParCo2019, held in Prague, Czech Republic, from 10 – 13 September 2019, was no exception. Its papers, invited talks, and specialized mini-symposia addressed cutting-edge topics in computer architectures, programming methods for specialized devices such as field programmable gate arrays (FPGAs) and graphical processing units (GPUs), innovative applications of parallel computers, approaches to reproducibility in parallel computations, and other relevant areas. This book presents the proceedings of ParCo2019, with the goal of making the many fascinating topics discussed at the meeting accessible to a broader audience. The proceedings contains 57 contributions in total, all of which have been peer-reviewed after their presentation. These papers give a wide ranging overview of the current status of research, developments, and applications in parallel computing.

The impending advent of GSM in the early 1990s triggered massive investment that revolutionised the capability of DSP technology. A decade later, the vastly increased processing requirements and potential market of 3G has triggered a similar revolution, with a host of start-up companies claiming revolutionary technologies hoping to challenge and displace incumbent suppliers. This book, with contributions from today's major players and leading start-ups, comprehensively describes both the new approaches and the responses of the incumbents, with detailed descriptions of the design philosophy, architecture, technology maturity and software support. Analysis of SDR baseband processing requirements of cellular handsets and basestations 3G handset baseband - ASIC, DSP, parallel processing, ACM and customised programmable architectures 3G basestation baseband - DSP (including co-processors), FPGA-based approaches, reconfigurable and parallel architectures Architecture optimisation to match 3G air interface and application algorithms Evolution of existing DSP, ASIC & FPGA solutions Assessment of the architectural approaches and the implications of the trends. An essential resource for the 3G product designer, who needs to understand immediate design options within a wider context of future product roadmaps, the book will also benefit researchers and commercial managers who need to understand this rapid evolution of baseband signal processing and its industry impact.

This book addresses the question how run-time reconfigurable constant multipliers (RCMs) can be efficiently implemented on field programmable gate arrays (FPGAs). RCMs calculate the multiplication of an input number by one out of several constants which can be selected during run-time. This is important as constant multiplication is an essential operation in digital signalprocessing (DSP) applications. The evaluation of RCMs is done by considering reconfiguration using reconfigurable look-up tables (LUTs),reconfiguration using multiplexers (MUXs) and Partial Reconfiguration (PR). This book contributes two new methods to generate RCMs using the first two reconfiguration principles. First, a LUT-based constant multiplier is extended to be reconfigurable. Second, optimized constant multipliers without reconfiguration are fused using MUXs. Moreover, a general post-optimization for MUX-based RCMs is proposed. Finally, the design space produced in this way is analyzed using synthesis experiments. The contributed methods provide important trade-off points in the design space of RCMs on FPGAs.

This open access book summarizes the research done and results obtained in the second funding phase of the Priority Program 1648 "Software for Exascale Computing" (SPPEXA) of the German Research Foundation (DFG) presented at the SPPEXA Symposium in Dresden during October 21-23, 2019. In that respect, it both represents a continuation of Vol. 113 in Springer's series Lecture Notes in Computational Science and Engineering, the corresponding report of SPPEXA's first funding phase, and provides an overview of SPPEXA's contributions towards exascale computing in today's sumpercomputer technology. The individual chapters address one or more of the research directions (1) computational algorithms, (2) system software, (3) application software, (4) data management and exploration, (5) programming, and (6) software tools. The book has an interdisciplinary appeal: scholars from computational sub-fields in computer science, mathematics, physics, or engineering will find it of particular interest.

This dissertation, "A Run-time Hardware Task Execution Framework for FPGA-accelerated Heterogeneous Cluster" by Yuk-ming, Choi, ???, was obtained from The University of Hong Kong (Pokfulam, Hong Kong) and is being sold pursuant to Creative Commons: Attribution 3.0 Hong Kong License. The content of this dissertation has not been altered in any way. We have altered the formatting in order to facilitate the ease of printing and reading of the dissertation. All rights not granted by the above license are retained by the

author. Abstract: The era of big data has led to problems of unprecedented scale and complexity that are challenging the computing capability of conventional computer systems. One way to address the computational and communication challenges of such demanding applications is to incorporate the use of non-conventional hardware accelerators such as FPGAs into existing systems. By providing a mix of FPGAs and conventional CPUs as computing resources in a heterogeneous cluster, a distributed computing environment can be achieved to address the need of both compute-intensive and data-intensive applications. However, utilizing heterogeneous clusters requires application developers' comprehensive knowledge on both hardware and software. In order to assist programmers to take advantage of the synergy between hardware and software easily, an easy-to-use framework for virtualizing the underlying FPGA computing resources of the heterogeneous cluster is motivated. In this work, a heterogeneous cluster consisting of both FPGAs and CPUs was built and a framework for managing multiple FPGAs across the cluster was designed. The major contribution of the framework is to provide an abstraction layer between the application developer and the underlying FPGA computing resources, so as to improve the overall design productivity. An inter-FPGA communication system was implemented such that gateware executing on FPGAs can communicate with each other autonomously to the CPU. Furthermore, to demonstrate a real-life application on the heterogeneous cluster, a generic k-means clustering application was implemented, using the MapReduce programming model. The implementation of the k-means application on multiple FPGAs was compared with a software-only version that was run on a Hadoop multi-core computer cluster. The performance results show that the FPGA version outperforms the Hadoop version across various parameters. An in-depth study on the communication bottleneck presented in the system was also carried out. A number of experiments were specifically designed to benchmark the performance of each I/O channel. The study shows that the major source of I/O bottleneck lies at the communication between the host system and the FPGA. This gives insight into programming considerations of potential applications on the cluster as well as improvement to the framework. Moreover, the benefit of multiple FPGAs was investigated through a series of experiments. Compared with putting all mappers on a single FPGA, it was found that distributing the same amount of mappers across more FPGAs can provide a tradeoff between FPGA resources and I/O performance. DOI: 10.5353/th_b5270558 Subjects: High performance computing Field programmable gate arrays