Intel Fpga Sdk For Opencl Altera

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

This book constitutes the proceedings of the 14th International Workshop on Open MP, IWOMP 2018, held in Barcelona, Spain, in September 2018. The 16 full papers presented in this volume were carefully reviewed and selected for inclusion in this book. The papers are organized in topical sections named: best paper; loops and OpenMP; OpenMP in heterogeneous systems; OpenMP improvements and innovations; OpenMP user experiences: applications and tools; and tasking evaluations.

In this new edition of the Handbook of Signal Processing Systems, many of the chapters from the previous editions have been updated, and several new chapters have been added. The new contributions include chapters on signal processing methods for light field displays, throughput analysis of dataflow graphs, modeling for reconfigurable signal processing systems, fast Fourier transform architectures, deep neural networks, programmable architectures for histogram of oriented gradients processing, high dynamic range video coding, system-on-chip architectures for data analytics, analysis of finite word-length effects in fixed-point systems, and models of architecture. There are more than 700 tables and illustrations; in this edition over 300 are in color. This new edition of the handbook is organized in three parts. Part I motivates representative applications that drive and apply state-of-the art methods for design and implementation of signal processing systems; Part II discusses architectures for implementing these applications; and Part III focuses on

compilers, as well as models of computation and their associated design tools and methodologies.
The demand for scalable, high-performance computing has increased as the size of datasets has grown in recent years. However, the breakdown of Dennard's scaling has led to energy efficiency becoming an important concern in datacenters, and spawned exploration into using power-efficient processors such as GPUs (Graphic Processing Units) and FPGAs (Field-Programmable Gate Arrays) as accelerators in datacenters. In particular, the FPGA's low power consumption and the re-programmability allow datacenters to use FPGAs as highly energy-efficient accelerators for a variety of application. On the other hand, FPGA has poor programmability compared to instructions-based architectures like CPU and GPU. To facilitate the process of implementing and deploying FPGA accelerators, High-Level Synthesis (HLS) that generates functional-equivalent RTL from C-based programming languages attracts more and more attention since past decades. Nowadays, both FPGA vendors have their commercial HLS products -- Xilinx SDx and Intel FPGA SDK for OpenCL. However, modern HLS is still not friendly for software designers who have limited FPGA domain knowledge. Since the hardware architecture inferred from a syntactic C implementation could be ambiguous, current commercial HLS tools usually generate architecture structures according to specific HLS C code patterns. As a result, even though the authors have illustrated that the HLS tool is capable of generating FPGA designs with competitive performance as the one in RTL, designers must manually reconstruct the HLS C kernel with specific code patterns to achieve high performance. This problem becomes one of the main impediments to consolidating the FPGA community on cooperation and developments. In this dissertation, we first present an automated framework that frees human efforts from code reconstruction and design space exploration (DSE). The framework creates a more comprehensive micro-architecture design space from user-written C-based kernel with the Merlin compiler, so the design point should cover the design point with better performance when compared to the HLS-pragma-based design space. To efficiently identify the best design configuration in the tremendous design space, we first propose efficient design space pruning processes that reduce the design space by 24.65x. Accordingly, we develop and evaluate several approaches, including multi-armed bandit hyper heuristic approach, gradient-based approach, and design bottleneck optimization approach. The evaluation result shows that our DSE framework is able to identify the design point that achieves on average (using geometric mean) 93.78% QoR compared to the corresponding manual design. Based on the proposed DSE framework, we further support automated design optimization for high level domain specific languages (DSLs). Since DSLs might not explicitly provide interfaces for users to specify design configurations, automatic DSE becomes even more important when supporting DSLs for FPGAs. Specifically, we adopt Merlin C, an OpenMP-like C-based programming model, as the intermediate representation (IR) and implement DSL-to-Merlin front-end compilers while preserving the semantic and

domain-specific information such as parallel patterns, systolic patterns, and scheduling functions. We first implement Spark-to-Merlin front-end compiler that translates Spark applications in Scala to Merlin C for FPGA acceleration. By leveraging parallel patterns as scheduling hints, the generated accelerators are able to achieve 50x speedup on geometric mean for a set of machine learning kernels. In addition, we also demonstrate that our DSE framework can be even more practical for the DSLs with plenty scheduling functions. Specifically, we implement HeteroCL-to-Merlin front-end that takes HeteroCL programming model embedded in Python. Our DSE framework is capable of exploring a subset of HeteroCL scheduling primitives and let users focus on the platform independent loop transformations. With the help from the DSE framework, we achieve 27.62x speedup on geometric mean over a CPU core for a variety of compute-intensive kernels (chapter 3). On the other hand, a main challenge of performing design space exploration for a design with arbitrary functionality is the lack of the assumption to underlying micro-architectures. As we will illustrate in the dissertation, the cost of evaluating the quality of a design point is extremely expensive (15-60 minutes) so only a limited number of design points can be explored. In addition, due to the uncertainty of vendor tool behaviors, the development of performance and resource modeling is also unrealistic. As a result, we propose composable, parallel and pipeline (CPP) architecture template to limit the design space to a certain region that is more practical and has less exceptions (chapter 4). With the CPP architecture, we are able to derive an incremental analytical model, which only requires a few HLS run to be initialized, to facilitate the DSE process. In the last part of this dissertation, we use convolutional neural network (CNN) to demonstrate that the HLS runtime cost can be totally saved with the use of a more domain specific architecture (chapter 5). Specifically, we leverage a systolic array architecture template for CNN accelerator generation. By mapping a CNN model to the pre-defined systolic array template, we can guarantee the model accuracy and DSE efficiency. The experimental result shows that our analytical model for the architecture template achieves 96% accuracy, and the mapped CNN model achieves up to 1.2 Tops throughput on Intel Arria 10 FPGA.

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.

The book presents the proceedings of four conferences: The 26th International Conference on Parallel and Distributed

Processing Techniques and Applications (PDPTA'20), The 18th International Conference on Scientific Computing (CSC'20); The 17th International Conference on Modeling, Simulation and Visualization Methods (MSV'20); and The 16th International Conference on Grid, Cloud, and Cluster Computing (GCC'20). The conferences took place in Las Vegas, NV, USA, July 27-30, 2020. The conferences are part of the larger 2020 World Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE'20), which features 20 major tracks. Authors include academics, researchers, professionals, and students. Presents the proceedings of four conferences as part of the 2020 World Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE'20); Includes the research tracks Parallel and Distributed Processing, Scientific Computing, Modeling, Simulation and Visualization, and Grid, Cloud, and Cluster Computing; Features papers from PDPTA'20, CSC'20, MSV'20, and GCC'20.

High-performance computing (HPC) has become an essential tool in the modern world. However, systems frequently run well below theoretical peak performance, with only 5% being reached in many cases. In addition, costly components often remain idle when not required for specific programs, as parts of the HPC systems are reserved and used exclusively for applications. A project was started in 2013, funded by the German Ministry of Education and Research (BMBF), to find ways of improving system utilization by compromising on dedicated reservations for HPC codes and applying co-scheduling of applications instead. The need was recognized for international discussion to find the best solutions to this HPC utilization issue, and a workshop on co-scheduling in HPC, open to international participants – the COSH workshop – was held for the first time at the European HiPEAC conference, in Prague, Czech Republic, in January 2016. This book presents extended versions of papers submitted to the workshop, reviewed for the second time to ensure scientific quality. It also includes an introduction to the main challenges of co-scheduling and a foreword by Arndt Bode, head of LRZ, one of Europe's leading computer centers, as well as a chapter corresponding to the invited keynote speech by Intel, whose recent extensions to their processors allow for better control of co-scheduling.

This book constitutes the refereed proceedings of the 9th International Conference on Data-Driven Innovation, ICT Innovations 2017, held in Skopje, Macedonia, in September 2017. The 26 full papers presented were carefully reviewed and selected from 90 submissions. They cover the following topics: big data analytics, cloud computing, data mining, digital signal processing, e-health, embedded systems, emerging mobile technologies, multimedia, Internet of Things (IoT), machine learning, software engineering, security and cryptography, coding theory, wearable technologies, wireless communication, and sensor networks.

This book constitutes the proceedings of the 15th International Symposium on Applied Reconfigurable Computing, ARC 2019, held in Darmstadt, Germany, in April 2019. The 20 full papers and 7 short papers presented in this volume were carefully reviewed

and selected from 52 submissions. In addition, the volume contains 1 invited paper. The papers were organized in topical sections named: Applications; partial reconfiguration and security; image/video processing; high-level synthesis; CGRAs and vector processing; architectures; design frameworks and methodology; convolutional neural networks.

This book presents the latest knowledge of the newly discovered Earth-like exoplanets and reviews improvements in both radio and optical SETI. A key aim is to stimulate fresh discussion on algorithms that will be of high value in this extremely complicated search. Exoplanets resembling Earth could well be able to sustain life and support the evolution of technological civilizations, but to date, all searches for such life forms have proved fruitless. The failings of SETI observations are well recognized, and a new search approach is necessary. In this book, different detection algorithms that exploit state-of-the-art, low-cost, and extremely fast multiprocessors are examined and compared. Novel methods such as the agnostic entropy and high-sensitivity blind signal extraction algorithms should represent a quantum leap forward in SETI. The book is of interest to all researchers in the field and hopefully stimulates significant progress in the search for extraterrestrial intelligence.

This book constitutes the proceedings of the 17th International Conference on Algorithms and Architectures for Parallel Processing, ICA3PP 2017, held in Helsinki, Finland, in August 2017. The 25 full papers presented were carefully reviewed and selected from 117 submissions. They cover topics such as parallel and distributed architectures; software systems and programming models; distributed and network-based computing; big data and its applications; parallel and distributed algorithms; applications of parallel and distributed computing; service dependability and security in distributed and parallel systems; service dependability and security in distributed and parallel systems; performance modeling and evaluation.This volume also includes 41 papers of four workshops, namely: the 4th International Workshop on Data, Text, Web, and Social Network Mining (DTWSM 2017), the 5th International Workshop on Parallelism in Bioinformatics (PBio 2017), the First International Workshop on Distributed Autonomous Computing in Smart City (DACSC 2017), and the Second International Workshop on Ultrascale Computing for Early Researchers (UCER 2017).

This book constitutes revised selected papers from the workshops held at 24th International Conference on Parallel and Distributed Computing, Euro-Par 2018, which took place in Turin, Italy, in August 2018. The 64 full papers presented in this volume were carefully reviewed and selected from 109 submissions. Euro-Par is an annual, international conference in Europe, covering all aspects of parallel and distributed processing. These range from theory to practice, from small to the largest parallel and distributed systems and infrastructures, from fundamental computational problems to full-edged applications, from architecture, compiler, language and interface design and implementation to tools, support infrastructures, and application performance aspects.

Most emerging applications in imaging and machine learning must perform immense amounts of computation while holding to strict limits on energy and power. To meet these goals, architects are building increasingly specialized compute engines tailored for these specific tasks. The resulting computer systems are heterogeneous, containing multiple processing cores with wildly

different execution models. Unfortunately, the cost of producing this specialized hardware—and the software to control it—is astronomical. Moreover, the task of porting algorithms to these heterogeneous machines typically requires that the algorithm be partitioned across the machine and rewritten for each specific architecture, which is time consuming and prone to error. Over the last several years, the authors have approached this problem using domain-specific languages (DSLs): high-level programming languages customized for specific domains, such as database manipulation, machine learning, or image processing. By giving up generality, these languages are able to provide high-level abstractions to the developer while producing high performance output. The purpose of this book is to spur the adoption and the creation of domain-specific languages, especially for the task of creating hardware designs. In the first chapter, a short historical journey explains the forces driving computer architecture today. Chapter 2 describes the various methods for producing designs for accelerators, outlining the push for more abstraction and the tools that enable designers to work at a higher conceptual level. From there, Chapter 3 provides a brief introduction to image processing algorithms and hardware design patterns for implementing them. Chapters 4 and 5 describe and compare Darkroom and Halide, two domain-specific languages created for image processing that produce high-performance designs for both FPGAs and CPUs from the same source code, enabling rapid design cycles and quick porting of algorithms. The final section describes how the DSL approach also simplifies the problem of interfacing between application code and the accelerator by generating the driver stack in addition to the accelerator configuration. This book should serve as a useful introduction to domain-specialized computing for computer architecture students and as a primer on domain-specific languages and image processing hardware for those with more experience in the field.

The Complete Guide to OpenACC for Massively Parallel Programming Scientists and technical professionals can use OpenACC to leverage the immense power of modern GPUs without the complexity traditionally associated with programming them. OpenACC™ for Programmers is one of the first comprehensive and practical overviews of OpenACC for massively parallel programming. This book integrates contributions from 19 leading parallel-programming experts from academia, public research organizations, and industry. The authors and editors explain each key concept behind OpenACC, demonstrate how to use essential OpenACC development tools, and thoroughly explore each OpenACC feature set. Throughout, you'll find realistic examples, hands-on exercises, and case studies showcasing the efficient use of OpenACC language constructs. You'll discover how OpenACC's language constructs can be translated to maximize application performance, and how its standard interface can target multiple platforms via widely used programming languages. Each chapter builds on what you've already learned, helping you build practical mastery one step at a time, whether you're a GPU programmer, scientist, engineer, or student. All example code and exercise solutions are available for download at GitHub. Discover how OpenACC makes scalable parallel programming easier and more practical Walk through the OpenACC spec and learn how OpenACC directive syntax is structured Get productive with OpenACC code editors, compilers, debuggers, and performance analysis tools Build your first real-world OpenACC programs Exploit loop-level parallelism in OpenACC, understand the levels of parallelism available, and maximize accuracy or performance

Learn how OpenACC programs are compiled Master OpenACC programming best practices Overcome common performance, portability, and interoperability challenges Efficiently distribute tasks across multiple processors Register your product at informit.com/register for convenient access to downloads, updates, and/or corrections as they become available.

Design of FPGA-Based Computing Systems with OpenCLSpringer

This book constitutes the refereed proceedings of the 35th International Conference on High Performance Computing, ISC High Performance 2020, held in Frankfurt/Main, Germany, in June 2020.* The 27 revised full papers presented were carefully reviewed and selected from 87 submissions. The papers cover a broad range of topics such as architectures, networks & infrastructure; artificial intelligence and machine learning; data, storage & visualization; emerging technologies; HPC algorithms; HPC applications; performance modeling & measurement; programming models & systems software. *The conference was held virtually due to the COVID-19 pandemic. Chapters "Scalable Hierarchical Aggregation and Reduction Protocol (SHARP) Streaming-Aggregation Hardware Design and Evaluation", "Solving Acoustic Boundary Integral Equations Using High Performance Tile Low-Rank LU Factorization", "Scaling Genomics Data Processing with Memory-Driven Computing to Accelerate Computational Biology", "Footprint-Aware Power Capping for Hybrid Memory Based Systems", and "Pattern-Aware Staging for Hybrid Memory Systems" are available open access under a Creative Commons Attribution 4.0 International License via link.springer.com.

This book follows an example-driven, simplified, and practical approach to using OpenCL for general purpose GPU programming. If you are a beginner in parallel programming and would like to quickly accelerate your algorithms using OpenCL, this book is perfect for you! You will find the diverse topics and case studies in this book interesting and informative. You will only require a good knowledge of C programming for this book, and an understanding of parallel implementations will be useful, but not necessary.

Using the new OpenCL (Open Computing Language) standard, you can write applications that access all available programming resources: CPUs, GPUs, and other processors such as DSPs and the Cell/B.E. processor. Already implemented by Apple, AMD, Intel, IBM, NVIDIA, and other leaders, OpenCL has outstanding potential for PCs, servers, handheld/embedded devices, high performance computing, and even cloud systems. This is the first comprehensive, authoritative, and practical guide to OpenCL 1.1 specifically for working developers and software architects. Written by five leading OpenCL authorities, OpenCL Programming Guide covers the entire specification. It reviews key use cases, shows how OpenCL can express a wide range of parallel algorithms, and offers complete reference material on both the API and OpenCL C programming language. Through complete case studies and downloadable code examples, the authors show how to write complex parallel programs that decompose workloads across many different devices. They also present all the essentials of OpenCL software performance optimization, including probing and adapting to hardware. Coverage includes Understanding OpenCL’s architecture, concepts, terminology, goals, and rationale Programming with OpenCL C and the runtime API Using buffers, sub-buffers, images, samplers, and events Sharing and synchronizing data with OpenGL and Microsoft’s Direct3D Simplifying development with the C++ Wrapper API Using OpenCL Embedded Profiles to support devices ranging from cellphones to supercomputer nodes Case studies dealing with physics simulation; image and signal processing, such as image histograms, edge detection filters, Fast Fourier Transforms, and optical flow; math

libraries, such as matrix multiplication and high-performance sparse matrix multiplication; and more Source code for this book is available at https://code.google.com/p/opencl-book-samples/

This book presents the combined proceedings of the 10th International Conference on Computer Science and its Applications (CSA 2018) and the 13th KIPS International Conference on Ubiquitous Information Technologies and Applications (CUTE 2018), both held in Kuala Lumpur, Malaysia, Dec 17 - 19, 2018. The aim of these two meetings was to promote discussion and interaction among academics, researchers and professionals in the field of ubiquitous computing technologies. These proceedings reflect the state of the art in the development of computational methods, involving theory, algorithms, numerical simulation, error and uncertainty analysis and novel applications of new processing techniques in engineering, science, and other disciplines related to ubiquitous computing.

This book discusses and compares several new trends that can be used to overcome Moore’s law limitations, including Neuromorphic, Approximate, Parallel, In Memory, and Quantum Computing. The author shows how these paradigms are used to enhance computing capability as developers face the practical and physical limitations of scaling, while the demand for computing power keeps increasing. The discussion includes a state-of-the-art overview and the essential details of each of these paradigms.

This proceedings book presents the latest research in the fields of information theory, communication system, computer science and signal processing, as well as other related technologies. Collecting selected papers from the 3rd Conference on Signal and Information Processing, Networking and Computers (ICSINC), held in Chongqing, China on September 13-15, 2017, it is of interest to professionals from academia and industry alike.

This book gathers 14 of the most promising papers presented at the 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2017), which was held on June 26–28, 2017 in Kanazawa, Japan. The aim of this conference was to bring together researchers and scientists, businessmen and entrepreneurs, teachers, engineers, computer users, and students to discuss the various fields of computer science and to share their experiences and exchange new ideas and information in a meaningful way. The book presents research findings concerning all aspects (theory, applications and tools) of computer and information science, and discusses the practical challenges encountered along the way, as well as the solutions adopted to solve them.

??????????????????Intel FPGA/CPLD????????????FPGA/CPLD?????????????Intel FPGA??SoC FPGA?OpenCL????????????????????????Intel????????????????????

This textbook for courses in Embedded Systems introduces students to necessary concepts, through a hands-on approach. It gives a great introduction to FPGA-based microprocessor system design using state-of-the-art boards, tools, and microprocessors from Altera/Intel® and Xilinx®. HDL-based designs (soft-core), parameterized cores (Nios II and MicroBlaze), and ARM Cortex-A9 design are discussed, compared and explored using many hand-on designs projects. Custom IP for HDMI coder, Floating-point operations, and FFT bit-swap are developed, implemented, tested and speed-up is measured. Downloadable files include all design examples such as basic processor synthesizable code for Xilinx and Altera tools for PicoBlaze, MicroBlaze, Nios II and ARMv7 architectures in VHDL and Verilog code, as well as the custom IP projects. Each Chapter has a substantial number of short quiz questions, exercises, and challenging projects. Explains soft, parameterized, and hard core systems design tradeoffs; Demonstrates design of popular KCPSM6 8 Bit microprocessor step-by-step; Discusses the 32 Bit ARM Cortex-A9 and a basic processor is synthesized; Covers design flows for both FPGA Market leaders Nios II Altera/Intel and MicroBlaze

Xilinx system; Describes Compiler-Compiler Tool development; Includes a substantial number of Homework's and FPGA exercises and design projects in each chapter.

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.

Hardware Accelerator Systems for Artificial Intelligence and Machine Learning, Volume 122 delves into artificial Intelligence and the growth it has seen with the advent of Deep Neural Networks (DNNs) and Machine Learning. Updates in this release include chapters on Hardware accelerator systems for artificial intelligence and machine learning, Introduction to Hardware Accelerator Systems for Artificial Intelligence and Machine Learning, Deep Learning with GPUs, Edge Computing Optimization of Deep Learning Models for Specialized Tensor Processing Architectures, Architecture of NPU for DNN, Hardware Architecture for Convolutional Neural Network for Image Processing, FPGA based Neural Network Accelerators, and much more. Updates on new information on the architecture of GPU, NPU and DNN Discusses In-memory computing, Machine intelligence and Quantum computing Includes sections on Hardware Accelerator Systems to improve processing efficiency and performance

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.

It constitutes the refereed proceedings of the 4th Asian Supercomputing Conference, SCFA 2018, held in Singapore in March 2018. Supercomputing Frontiers will be rebranded as Supercomputing Frontiers Asia (SCFA), which serves as the technical programme for SCA18. The technical programme for SCA18 consists of four tracks: Application, Algorithms & Libraries Programming System Software Architecture, Network/Communications & Management Data, Storage & Visualisation The 20 papers presented in this volume were carefully reviewed nd selected from 60 submissions.

P2P, Grid, Cloud and Internet computing technologies have been very fast established as breakthrough paradigms for solving complex problems by enabling aggregation and sharing of an increasing variety of distributed computational resources at large scale. The aim of this volume is to provide latest research findings, innovative research results, methods and development techniques from both theoretical and practical perspectives related to P2P, Grid, Cloud and Internet computing as well as to reveal synergies among such large scale computing paradigms. This proceedings volume presents the results of the 11th International Conference on P2P, Parallel, Grid, Cloud And Internet Computing (3PGCIC-2016), held November 5-7, 2016, at Soonchunhyang University, Asan, Korea

As predicted by Gordon E. Moore in 1965, the performance of computer processors increased at an exponential rate. Nevertheless, the increases in computing speeds of single processor machines were eventually curtailed by physical constraints. This led to the development of parallel computing, and whilst progress has been made in this field, the complexities of parallel algorithm design, the deficiencies of the available software development tools and the complexity of scheduling tasks over thousands and even millions of processing nodes represent a major challenge to the construction and use of more powerful parallel systems. This book presents the proceedings of the biennial International Conference on Parallel Computing (ParCo2015), held in Edinburgh, Scotland, in September 2015. Topics covered include computer architecture and performance, programming models and methods, as well as applications. The book also includes two invited talks and a number of mini-symposia. Exascale computing holds enormous promise in terms of increasing scientific knowledge acquisition and thus contributing to the future well-being and prosperity of mankind. A number of innovative approaches to the development and use of future high-performance and high-throughput systems are to be found in this book, which will be of interest to all those whose work involves the handling and processing of large amounts of data.

This book constitutes the proceedings of the 15th International Workshop on Open MP, IWOMP 2019, held in Auckland, New Zealand, in September 2019. The 22 full papers presented in this volume were carefully reviewed and selected for inclusion in this book. The papers are organized in topical sections named: best paper; tools, accelerators, compilation, extensions, tasking, and using OpenMP.

This book constitutes the proceedings of the 16th International Symposium on Applied Reconfigurable Computing, ARC 2020, held in Toledo, Spain, in April 2020. The 18 full papers and 11 poster presentations presented in this volume were carefully reviewed and selected from 40 submissions. The papers are organized in the following topical sections: design methods & tools; design space exploration & estimation techniques; high-level synthesis; architectures; applications.

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.

This book gathers selected research papers presented at the First International Conference on Digital Technologies and Applications (ICDTA 21), held at Sidi Mohamed Ben Abdellah University, Fez, Morocco, on 29-30 January 2021. highlighting the latest innovations in digital technologies as: artificial intelligence, Internet of things, embedded systems, network technology, information processing, and their applications in several areas such as hybrid vehicles, renewable energy, robotic, and COVID-19. The respective papers encourage and inspire researchers, industry professionals, and policymakers to put these methods into practice.

Heterogeneous Computing Architectures: Challenges and Vision provides an updated vision of the state-of-the-art of heterogeneous computing systems, covering all the aspects related to their design: from the architecture and programming models to hardware/software integration and orchestration to real-time and security requirements. The transitions from multicore

processors, GPU computing, and Cloud computing are not separate trends, but aspects of a single trend-mainstream; computers from desktop to smartphones are being permanently transformed into heterogeneous supercomputer clusters. The reader will get an organic perspective of modern heterogeneous systems and their future evolution.

Autonomous vehicles need new and diverse computing workloads, which demand stringent performance requirements at real-time. Such high efficiency can be achieved by heterogeneous and parallel processing systems. Then it is critical to assess the state of the art in terms of current heterogeneous hardware platforms for autonomous driving tasks. This thesis thus focuses on finding the suitable hardware platform for autonomous vehicle workloads. Specifically, this study compares system implementation and platform performance of FPGAs and GPUs running a traffic sign recognition task. We first implement a speed-limit-sign recognition task using a template-based approach on the FPGA with the Intel FPGA SDK for OpenCL. Then we evaluate its throughput, power consumption, accuracy, and development effort against those of the GPU implementation that is based on our previous study. During the implementation and evaluation process,we make the following contributions: Develop a general design flow for translating/converting a GPU software system to an FPGA counterpart; Design various optimization techniques in the FPGA version; Invent a novel and highly efficient real-to-complex FFT engine for image processing on FPGAs. This FFT engine can be utilized by other developers to implement related tasks; Provide insights from our development and evaluation experience. The major insight reveals that FPGA implementation can provide significantly better power consumption for the same detection accuracy while the GPU supports better programmer efficiency; Deliver our recommendations on how to improve Intel FPGA SDK for OpenCL.These contributions can be useful for designing upcoming versions of FPGA-focused OpenCL development environments. Besides, this thesis also presents a study of an implementation of Convolutional Neural Network (CNN) models on FPGA along with our enhancement effort to add new operations, such as dilated convolution.

This book constitutes the proceedings of the workshops of the 23rd International Conference on Parallel and Distributed Computing, Euro-Par 2017, held in Santiago de Compostela. Spain in August 2017. The 59 full papers presented were carefully reviewed and selected from 119 submissions. Euro-Par is an annual, international conference in Europe, covering all aspects of parallel and distributed processing. These range from theory to practice, from small to the largest parallel and distributed systems and infrastructures, from fundamental computational problems to full-edged applications, from architecture, compiler, language and interface design and implementation to tools, support infrastructures, and application performance aspects.

Heterogeneous Computing with OpenCL 2.0 teaches OpenCL and parallel programming for complex systems that may include a variety of device architectures: multi-core CPUs, GPUs, and fully-integrated Accelerated Processing Units (APUs). This fully-revised edition includes the latest enhancements in OpenCL 2.0 including: • Shared virtual memory to increase programming flexibility and reduce data transfers that consume resources • Dynamic parallelism which reduces processor load and avoids bottlenecks • Improved imaging support and integration with OpenGL Designed to work on multiple platforms, OpenCL will help you more effectively program for a heterogeneous future. Written by leaders in the parallel computing and OpenCL communities,

this book explores memory spaces, optimization techniques, extensions, debugging and profiling. Multiple case studies and examples illustrate high-performance algorithms, distributing work across heterogeneous systems, embedded domain-specific languages, and will give you hands-on OpenCL experience to address a range of fundamental parallel algorithms. Updated content to cover the latest developments in OpenCL 2.0, including improvements in memory handling, parallelism, and imaging support Explanations of principles and strategies to learn parallel programming with OpenCL, from understanding the abstraction models to thoroughly testing and debugging complete applications Example code covering image analytics, web plugins, particle simulations, video editing, performance optimization, and more

This book constitutes the proceedings of the 25th International Conference on Parallel and Distributed Computing, Euro-Par 2019, held in Göttingen, Germany, in August 2019. The 36 full papers presented in this volume were carefully reviewed and selected from 142 submissions. They deal with parallel and distributed computing in general, focusing on support tools and environments; performance and power modeling, prediction and evaluation; scheduling and load balancing; high performance architectures and compilers; data management, analytics and deep learning; cluster and cloud computing; distributed systems and algorithms; parallel and distributed programming, interfaces, and languages; multicore and manycore parallelism; theory and algorithms for parallel computation and networking; parallel numerical methods and applications; accelerator computing; algorithms and systems for bioinformatics; and algorithms and systems for digital humanities.

This book provides wide knowledge about designing FPGA-based heterogeneous computing systems, using a high-level design environment based on OpenCL (Open Computing language), which is called OpenCL for FPGA. The OpenCL-based design methodology will be the key technology to exploit the potential of FPGAs in various applications such as low-power embedded applications and high-performance computing. By understanding the OpenCL-based design methodology, readers can design an entire FPGA-based computing system more easily compared to the conventional HDL-based design, because OpenCL for FPGA takes care of computation on a host, data transfer between a host and an FPGA, computation on an FPGA with a capable of accessing external DDR memories. In the step-by-step way, readers can understand followings: how to set up the design environment how to write better codes systematically considering architectural constraints how to design practical applications