# UC Riverside
## UC Riverside Previously Published Works

**Title**

Introduction to the special issue on application-specific processors

**Permalink**

**Journal**

**ISSN**

**Authors**

Brisk, Philip
Mitra, Tulika

**Publication Date**

**DOI**

Peer reviewed

# Introduction to the Special Issue on Application-Specific Processors

Application-specific processors offer performance, energy, and cost benefits compared to their general-purpose counterparts for a wide variety of market segments, ranging from low-cost micro-controllers to high-end supercomputers. The design and usage of application-specific processors are strikingly different from general-purpose computers as the architecture is tuned to accelerate only a specific application or a class of applications.

Application-specific architectures are often derived from a high-level specification of an application through a hardware/software co-design process, which can be highly automated; such a process would be inappropriate for general-purpose processor design. Moreover, code compilation targeting application-specific processors is often intertwined with the co-design process. Lastly, the system software may provide reduced and targeted functionality and interfaces at a much lower cost than a traditional embedded or real-time operating system such as iOS or Android.

Historically, application-specific processors have been part of low-cost embedded devices; however, the landscape is rapidly changing. One important factor is the emergence of graphics processing units (GPUs) as general-purpose high-performance computing devices, which utilize graphics-specific hardware, but can also provide significant acceleration boosts for vectorizable applications. Similarly, reconfigurable computing devices, such as FPGAs, provide a flexible hardware fabric that can be used to implement a wide variety of application-specific functionality and as such are increasingly used for code acceleration in different domains.

This special issue of *ACM Transactions on Embedded Computing Systems* is dedicated to all aspects of application-specific processors. Part of this special issue presents extended versions of some of the best papers that were presented at the IEEE Symposium on Application-Specific Processors (SASP) in 2009 and 2010. Altogether, we received 66 submitted manuscripts for the special issue, 10 of which were accepted for inclusion in this special issue. It is our great honor to introduce the articles.

Our first article, "*Hardware Architectural Support for Control Systems and Sensor Processing*" by Sundhanshu Vyas, Adwait Gupta, Christopher Gill, Ron K. Cytron, Joseph Zambreno and Philip Jones, describes a microcontroller that has been customized to control thousands of PID controllers concurrently. Most application-specific processors in the past have extended RISCs or VLIWs. This article represents a landmark effort in the design of customizable application-specific microcontrollers.

This special issue features three articles that focus on the architecture of application-specific processors and the design of application-specific accelerators, namely, "*Multicore-Based Vector Coprocessor Sharing for Performance and Energy Gains*" by Spiridon F. Beldianu and Sotirios G. Ziavras, "*A Systematic Approach for Optimized Bypass Configurations for Application-Specific Embedded Processors*" by Thorsten Jungeblut, Boris Hubener, Mario Porrmann, and Ulrich Rückert, and "*Custom Architecture for Multicore Audio Beamforming Systems*" by Dimitris Theodoropoulos and Georgi Kuzmanov. They all represent interesting design points for different application-specific architectural styles. This is a large design space, and these three articles provide excellent insights about how one can customize a hardware platform.

Our next article addresses the ever important issue of design tools for application-specific processors. "*Design Space Exploration and Runtime Resource Management for Multicores*" by Giovanni Mariani, Gianluca Palermo, Vittorio Zaccaria and Cristina Silvano provides an efficient approach for design space exploration. When generating an application-specific hardware platform, there exist a vast number of design decisions about how to optimize both the hardware and/or software; hence exhaustive enumeration of every possible design point is simply infeasible. This article describes a systematic way to efficiently explore the design space and ensure a high-quality design, while avoiding the unacceptable overhead of exhaustive methods.

The article "*Memory Performance Estimation of CUDA Programs*" by Yooseong Kim and Aviral Shrivastava focuses on compilation and performance tuning issues for GPUs. GPUs employ a vector-threaded architecture that exploits a large number of on-chip threads to (partially) mask long memory latencies. Understanding the sources of these latencies and addressing them by rewriting the software is of particular importance to GPU programmers. The design tool that is described and developed in this article is expected to have a significant impact in the GPU programming community.

The next two articles explore the usage of FPGAs in novel and emerging application domains. "*Parallel Architectures for the kNN Classifier—Design of Soft IP Cores and FPGA Implementations*" by Ioannis Stamoulias and Elias S. Manolakos is an interesting case study in which an FPGA can be customized to prototype hardware accelerators for machine learning algorithms. The article "*Automatic Synthesis of Physical System Differential Equation Models to a Custom Network of General Processing Elements on FPGAs*" by Chen Huang, Frank Vahid, and Tony Givargis describes a tool that compiles systems of partial differential equations onto an FPGA, effectively transforming the device into a custom-computing machine. The objective of this particular research effort is to allow FPGAs to replace physical mockup devices that are used for medical testing, which is a cutting-edge application of great importance to society.

Finally we have two articles that support high-level synthesis of FPGA-based accelerators but are diametrically different in their approach. "*LegUp: An Open-Source High-Level Synthesis Tool for FPGA-Based Processor/Accelerator Systems*" by Andrew Canis, Jongsok Choi, Mark Aldham, Victor Zhang, Ahmed Kammoona, Tomasz Czajkowski, Stephen D. Brown, and Jason H. Anderson describes an open-source high-level synthesis framework targeting FPGAs. We strongly believe that open-source research tools provide great value to the community; so we are proud to publish this article and expect that LegUp will be widely used by the FPGA research community in the years to come. "*Efficient Compilation of CUDA Kernels for High-Performance Computing on FPGAs*" by Alexandros Papakonstantinou, Karthik Gururaj, John Stratton, Deming Chen, Wen and Mei Hwu is an extension of the article that won the Best Paper Award at SASP 2009. This article describes an FPGA compiler for programs written using the CUDA API, which was originally developed for NVIDIA GPUs. This work is the first to demonstrate the portability between FPGAs and GPUs, providing a significant design productivity advantage for researchers and practitioners working with both devices. We are confident that this article will positively influence future efforts to port emerging GPU-oriented APIs, such as OpenCL and OpenACC, to FPGAs.

It is with great pleasure that we offer this selection of excellent work to you. We trust you will like reading it, and we hope these research efforts will inspire your future work and thoughts. Enjoy!

Philip Brisk
University of California, Riverside

Tulika Mitra
National University of Singapore

*Guest Editors*