Guest Editorial: Application Specific Processors and Architectures

Melissa C. Smith · Kubilay Atasu

Published online: 7 August 2014

© Springer Science+Business Media New York 2014

Application specific processors and architectures are becoming increasingly important across all fields of computing from embedded to high-performance systems. These architectures and systems require efficient arithmetic algorithms, compilers, operating systems, and specialized applications to achieve optimum performance under strict constraints including size, power, throughput and operating frequency. This special double issue focuses on the latest developments and solutions concerning application specific processors and architectures from the hardware and software perspectives.

This special issue consists of eleven papers related to the area of application-specific processors and architectures that are divided into three categories: compilers and operating systems, arithmetic algorithms, and application/algorithm specialization. The first paper in the compilers and operating systems category “Compact Code Generation for Tightly-Coupled Processor Arrays”, by Boppu et al., presents methods for code compaction and generation for programmable tightly-coupled processor arrays consisting of interconnected small light-weight VLIW cores. The authors integrate these methods into a design tool and evaluate the results with benchmarks and compare the results to other existing compiler frameworks. The methods exploit compute-intensive nested loops, providing design entry in the form of a functional programming language and loop parallelization in the polyhedron model. The methods also support zero-overhead looping not only for the innermost loops but also for arbitrarily nested loops.

The next paper, “Symbolic Mapping of Loop Programs onto Processor Arrays”, by Teich, et al., presents a solution to

the problem of joint tiling and scheduling of a loop nest with uniform data dependencies *symbolically*. This challenge arises when the size and number of available processors for parallel loop execution is not known at compile time. The paper discusses a solution for deriving *parameterized latency-optimal schedules* statically by proposing a two step approach that determines two potential schedules. Once the size of the processor array becomes known at run time, simple comparisons of latency-determining expressions finally steer which of these schedules will be dynamically selected and the corresponding program configuration executed on the resulting processor array so to avoid any further run-time optimization or expensive recompilation.

In the paper “Virtualized Execution and Management of Hardware Tasks on a Hybrid ARM-FPGA Platform”, by Jain et al., the authors focus on managing the execution of hardware tasks within a processor-based system, and in doing so, how to virtualize the resources to ensure isolation and predictability. The authors use a microkernel-based hypervisor running on a commercial hybrid (FPGA-based) computing platform. The hypervisor leverages the capabilities of the FPGA fabric, with support for discrete hardware accelerators, dynamically reconfigurable regions, and regions of virtual fabric. The authors study the communication overheads, quantify the context switch overhead of the hypervisor approach and compare with the idle time for a standard Linux implementation, showing two orders of magnitude improvement.

The final compilers and operating systems paper, “A Novel Object-Oriented Software Cache for Scratchpad-Based Multi-Core Clusters”, by Pinto et al., presents a software cache implementation for an accelerator fabric, with special focus on object-oriented caching techniques aimed at reducing the global overhead introduced by the proposed software cache. The authors validate the software cache approach with a set of experiments, and three case studies of Object-Oriented software cache for computer vision applications.

There are four papers related to application implementations in this special issue. The first paper, “Hardware Acceleration of

M. C. Smith (✉)
Holcombe, Department of Electrical and Computer Engineering,
Clemson University, Clemson, SC, USA
e-mail: smithmc@clemson.edu

K. Atasu
IBM Research, Zürich, Switzerland
e-mail: KAT@zurich.ibm.com

Red-Black Tree Management and Application to Just-In-Time Compilation”, authored by Carbon et al., investigates opportunities for improving the performance of Just-in-time (JIT) compilation. The authors present a performance analysis of different JIT compilation technologies and identify hardware and software optimization opportunities. They propose a solution based on a dedicated processor with specialized instructions for critical functions of JIT compilers.

For the paper “Pipelined HAC Estimation Engines for Multivariate Time Series”, by Guo, et al., discusses a hardware implementation of the Heteroskedasticity and autocorrelation consistent (HAC) covariance matrix estimation. The authors introduce a pipeline-friendly HAC estimation algorithm that eliminates conditionals, parallelizes computation, and promotes data reuse. The initial hardware architecture is discussed along with two performance-optimized architectures.

In “Large-Scale Pairwise Alignments on GPU Clusters: Exploring the Implementation Space”, by Truong et al., the authors present four GPU implementations for large-scale pairwise sequence alignment: *TiledDScan-mNW*, *DScan-mNW*, *RScan-mNW* and *LazyRScan-mNW*. The implementations are evaluated across a spectrum of GPUs with different compute capabilities. The proposed GPU kernels are integrated into a hybrid MPI-CUDA framework for deployment on both homogeneous and heterogeneous CPU-GPU clusters.

The fourth applications paper in this special issue, “Cryptographic Algorithms on the GA144 Asynchronous Multi-Core Processor”, by Schneider, et al., proposes an implementation of a standardized and well-established cryptography on an alternative, lightweight platform, namely an asynchronous GA144 ultra-low-powered multi-core processor with 144 tiny cores. The authors demonstrate that symmetric and asymmetric cryptography such as AES and RSA can be realized on this low-end device with low energy consumption while maintaining high performance. The authors also evaluate the side-channel resistance of the proposed design.

The final three papers included in this special issue are related to specific arithmetic algorithms. The first, “A Highly Efficient Multicore Floating-Point FFT Architecture Based on Hybrid Linear Algebra/FFT Cores”, by Pedram, et al., presents a power and area efficient multicore FFT processor that exhibits parallel scaling. Using a highly efficient hybrid linear algebra/FFT core, the authors co-design the on-chip memory hierarchy, on-chip interconnect, and FFT algorithms for the multicore FFT processor. The resulting architecture effectively scales up to 16 hybrid cores for transform sizes that can be contained in on-chip SRAM.

The second arithmetic paper, “An Efficient Scalable RNS Architecture for Large Dynamic Ranges”, by Matutino, et al., proposes an efficient scalable Residue Number System (RNS) architecture supporting moduli sets with an arbitrary number

of channels, supporting larger dynamic range and a higher level of parallelism. The proposed architecture supports forward and reverse RNS conversion, via reuse of the arithmetic channel units. Experimental results discussed in the paper suggest significant gains in arithmetic operation delay, with a similar area reduction compared to the state of the art RNS.

The final paper in this special issue and the arithmetic focus, “Options for Denormal Representation in Logarithmic Arithmetic”, by Arnold, et al., proposes three variations of a new number system that are all hybrids of the properties of Fixed-point Number System (FXNS) and Signed Logarithmic Number System (SLNS). The novel Denormal LNS (DLNS) circuit introduced allows arithmetic to be performed directly on such data encoded for gradual underflow (denormalized numbers). Further, the proposed approach allows for customization of the range in which the gradual underflow occurs. Other variations introduced in the paper utilize the well-known Mitchell’s method to make the cost of general multiplication, division and roots closer to that of SLNS.

The eleven papers included in this special double issue cover a spectrum of important developments pertaining to application specific processors and architectures. These papers provide new developments related to the hardware architecture and software components including compilers and operating systems, arithmetic algorithms, and application/algorithm specialization.

Melissa C. Smith received her B.S. (1993) and M.S. (1994) degrees in Electrical Engineering from Florida State University and a Ph.D. (2003) in Electrical Engineering from the University of Tennessee. Dr. Smith was a Research Associate at the Oak Ridge National Laboratory (ORNL) from 1994 to 2006. While at ORNL, she designed ASICs and FPGAs for various high-speed signal-processing applications. She joined the ECE faculty at Clemson in the Fall of 2006 where her current research

focuses on the performance analysis and optimization of emerging heterogeneous computing architectures (GPGPU- and FPGA-based systems) for various application domains including scientific applications, high-performance or real-time embedded applications, and medical and image processing. She is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE) since 2005. She received the Award of Distinction from the Clemson National Scholars Program in 2009 and was recognized by the YWCA in 2006 with the Tribute to Women Award in Science and Technology. Dr. Smith currently serves as the Faculty Advisor for the Clemson Student chapter of the Society of Women Engineers among other service duties including the Doctoral Showcase Committee for the Supercomputing conference.

Kubilay Atasu received his B.Sc. and Ph.D. degrees, both in computer engineering, from Bogazici University, Istanbul, respectively in 2000 and 2007. He also has a M.Eng. Degree from ALaRI, University of Lugano. From 2002 to 2003, he was a research assistant at the Swiss Federal Institute of Technology Lausanne. Between 2005 and 2008 he was a research associate at the Department of Computing, Imperial College London. In summer 2006, he was a visiting researcher at

Stanford University. Since May 2008, Kubilay is with IBM Research - Zurich. His research interests include design of hardware accelerators and use of hardware/software co-design and reconfigurable computing techniques to improve the computational efficiency, especially, of Big Data applications.

Kubilay Atasu was the recipient of a best paper award at the Design Automation Conference (DAC) in 2003, and at the IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP) in 2008. He was a program co-chair of ASAP 2013 and the general chair of ASAP 2014. He has also served in the program committees of DATE, FPL, and FPT conferences.