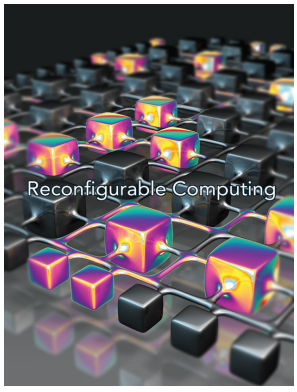
RECONFIGURABLE COMPUTING



Walid A. Najjar
University of California, Riverside

Paolo Ienne
École Polytechnique Fédérale de Lausanne

Reconfigurable computing is rapidly emerging as a third computing paradigm, along with hard-coded designs, often in the form of application specific integrated circuits (ASICs), and programmable systems, such as classic processors (CPUs), digital signal processors (DSPs), and GPUs. In hard-coded designs, the hardware datapaths and the computing algorithm are both fixed at production. In programmable systems, the hardware datapaths are fixed before production but implement generic primitives with some regular interconnection; algorithms are implemented post-production by freely scheduling operations on the datapaths through software. In the new paradigm of reconfigurable computing, both datapaths and algorithms are customized to the application only once the device is used.

This new computing paradigm was made possible by the rapid growth in size, speed, and complexity of field-programmable gate array (FPGA) devices. The combination of hardware efficiency and reconfigurability is what makes this paradigm attractive. Furthermore, the high device bandwidth of FPGAs makes them ideally suited for streaming applications, obviating the need for memory off-loading. An alternative to FPGAs are coarse-grained reconfigurable arrays (CGRAs), which comprise a collection of processing elements whose complexity varies from expanded arithmetic logic units (ALUs) to small CPUs with a rich and high-bandwidth interconnect. Their main attractiveness is in eliminating the overhead due to the fine-grained configuration of FPGAs but, at least thus far, they have not come even close the market penetration of FPGAs.

Evolution of FPGAs

The first generation of FPGAs was designed as glue logic, where a single device replaced multiple TTL devices in connecting a microcontroller or microprocessor to multiple peripherals, interfacing devices such as I/O and interrupt controllers, DMAs, and memory banks. Because of their low-cost objectives, these devices were lagging in technology well behind CPUs. Also, the additional cost for configurability was prohibitive for other, more complex commercial applications. It took many years of exponential fall in transistor cost, and FPGAs finally gained a strong place in functional verification, rapid prototyping, and low-volume products. Ramping mask and design costs are making FPGAs increasingly competitive to dedicated circuits, and they are slowly eroding some ASIC markets. Today's high-end FPGAs are at the cutting edge of CMOS technology, offering up to five orders of magnitude more logic than the first generation and four orders of magnitude higher bandwidth per device.

Early reconfigurable computing

Only three years after the introduction of the first FPGA device, the Xilinx XC2064 (1985), and a quarter of a century after the introduction of the idea of the Variable Structure Computer,[1] the first practical reconfigurable computing platform was developed at the DEC Paris Research Laboratory (PReL): the Programmable Active Memory (PAM),[2] a "universal hardware co-processor closely coupled to a standard host computer." Ten benchmark codes were implemented and evaluated on the PAM.[3] The authors' conclusions were that the PAM delivered a performance comparable to that of ASIC chips or supercomputers of the time, and was one to two orders of magnitude faster than software. Furthermore, its large off-chip bandwidth

Published by the IEEE Computer Society

(6.4 Gbits/s) made it ideally suited for "on-the-fly data acquisition and filtering."

Getting there

Over the past decade, multiple implementations of a wide variety of applications on reconfigurable platforms have demonstrated substantial speed-ups over both CPUs and GPUs, which is due to the elimination of memory offloading, a decoupled execution model where support operations (such as memory addressing and control) are not on the critical path, multiple deep pipelines customized to the application, and large degrees of parallelism supported by the high data bandwidth and large area of FPGAs. Initially, reconfigurable computing focused, for the most part, on signal, image, and video processing. However, very rapidly the spectrum of applications was broadened to include deep packet inspection, bioinformatics, linear algebra, string manipulation, data mining, and molecular dynamics, as well as many others.

In this special issue

We open this special issue with three quite different applications. Reprogrammability and high device bandwidth make FPGAs natural choices for high-speed network processing, and the increasing complexity of packet processing operations calls for more sophisticated languages to express them. In "High-Speed Packet Processing using Reconfigurable Computing" Gordon Brebner and Weirong Jiang describe PX, a declarative language for high-speed packet processing in FPGAs. Data analytics, requiring the rapid processing of massive streaming data, is rapidly becoming an important application for reconfigurable computing, as described in "Database Analytics: A Reconfigurable-Computing Approach" by Bharat Sukhwani et al. Finally, high-performance seismic data analysis requires both high bandwidth and computational density, which can be delivered by advanced reconfigurable computing systems, as shown by Haohuan Fu et al. in "Scaling Reverse Time Migration Performance through Reconfigurable Dataflow Engines."

One of the major impediments to a wider adoption of reconfigurable computing as a new paradigm is the complexity of programming FPGAs and the need for some hardware design expertise to tame them. A high-level language compilation of programs for reconfigurable hardware would make these platforms accessible to a large community of traditionally trained software application developers. In "Fast, Flexible High-Level Synthesis from OpenCL using Reconfiguration Contexts," James Coole and Greg Stitt describe a compilation framework from OpenCL to CGRAs.

FPGAs in computing are certainly not set to replace processors, but they are one of the heterogeneous system components available for solving problems efficiently. Effective integration of reconfigurable resources with more traditional computing components requires new features from operating systems. David Andrews gives a concise overview of the nascent field of operating systems for reconfigurable systems in "Operating Systems Research for Reconfigurable Computing." In "ReconOS: An Operating System Approach for Reconfigurable Computing," Andreas Agne et al. present a system that semantically integrates hardware accelerators and software processes in a multithreaded programming environment.

The performance and costs of reconfigurable systems are closely tied to the technology underlying FPGAs. However, such systems are by no means the drivers of FPGA evolution. In "Prospects for Reconfigurable Systems," Nick Tredennick and Brion Shimamoto revisit the predictions they expressed 10 years ago in an article in *ACM Queue*.[4] Finally, in "The Case for Embedded Networks on Chip on Field-Programmable Gate Arrays," Mohamed S. Abdelfattah and Vaughn Betz propose modifications to the FPGA architecture to make more efficient system-level integration possible and ultimately benefit large computing applications.

We hope that our selections for this special issue will both interest the computer architect practitioner and entice new researchers to this thriving area. Challenges are all but solved, included many of those already addressed here: The future will exact a much more seamless integration of heterogeneous computing paradigms and some

fundamentally simpler programming abstractions, as well as, perhaps, more dedicated and less inefficient FPGA architectures targeting reconfigurable computation. New challenges are also looming on the horizon, with, for instance, physical device sizes starting to strain our current capability to map and place and route circuits in reasonable time, hugely deviating from common runtimes of software tool chains, or our limited ability to help systematically debug efficiently large designs. Plenty of exciting work lies ahead of us all! MICRO

REFERENCES

1. G. Estrin, "Reconfigurable Computer Origins: The UCLA Fixed-Plus-Variable (F+V) Structure Computer," *IEEE Annals of the History of Computing,* vol. 24, no. 4, 2002, pp. 3-9.
2. P. Bertin, D. Roncin, and J. Vuillemin, *Introduction to Programmable Active Memories,* Prentice Hall, 1989, pp. 300-309.
3. P. Bertin, D. Roncin, and J. Vuillemin, "Programmable Active Memories: A Performance Assessment," *Parallel Architectures and Their Efficient Use,* 1992, pp. 119-130.
4. N. Tredennick and B. Shimamoto, "The Inevitability of Reconfigurable Systems," *ACM Queue,* Oct. 2003, pp. 34-43.

Walid A. Najjar is a professor in the Department of Computer Science and Engineering at the University of California, Riverside. His research interests include computer architectures and compilers for parallel and high-performance computing, embedded systems, FPGA-based code acceleration, and reconfigurable computing. Najjar has a PhD in computer engineering from the University of Southern California. He is a fellow of IEEE and the AAAS.

Paolo Ienne is a professor in the School of Computer and Communication Sciences at École Polytechnique Fédérale de Lausanne. His research interests include computer and processor architecture, electronic design automation, computer arithmetic, FPGAs and reconfigurable computing, and multiprocessor systems-on-chip. Ienne has a PhD in computer science from École Polytechnique Fédérale de Lausanne. He is a senior member of IEEE.

Direct questions and comments about the articles in this special section to Paolo Ienne at paolo.ienne@epfl.ch or to Walid Najjar at najjar@cs.ucr.edu.