Guest Editors' Introduction: Configurable Computing

Fadi J. Kurdahi and Nader Bagherzadeh
University of California, Irvine

Peter Athanas
Virginia Tech

José L. Muñoz
Defense Advanced Research Projects Agency

■ Configurable computing machines (CCMs) are emerging as a technology capable of providing high computational performance on a diversity of applications, including 1D and 2D signal processing, image processing, simulation acceleration, and scientific computing. In many cases, applications solved with configurable computing techniques supplant the equivalent of tens or hundreds of contemporary microprocessors or digital signal processors (DSPs). High performance is achieved by (dynamically) building custom computational operators, pathways, and pipelines suited to specific properties of the task at hand. With this approach, characteristics of a particular application (e.g., parallelism, locality, and data resolution) can be fully exploited. Hence, CCMs provide computational performance benefits close to application-specific integrated circuits (ASICs) at a lower cost, yet retain the flexibility and rapid reconfigurability of general-purpose microprocessors. Contemporary CCMs rely on RAM-based field-programmable gate arrays (FPGAs) as the mechanism for achieving reconfigurability. Although these devices were not originally intended for computing tasks, they have proven quite effective in demonstrating the potential for high-performance computing.

Runtime reconfiguration (known as RTR) is an implementation approach that divides an application into a series of sequentially executed stages, with each stage implemented as a separate execution module. Partial RTR extends this approach by partitioning these stages into finer-grain submodules constructed so that they can be swapped into the platform as needed to contribute to a given computation, much like paging in virtual memory systems. With this "virtual hardware" model, one can essentially emulate a much larger computing platform by rapidly changing the hardware functionality to suit the needs of the application at hand. Because of its unique computing paradigm, RTR offers an opportunity to reexamine many of the organizational assumptions of programming and computation. Two articles in this special issue address different aspects of RTR. Bazargan et al. look at online placement algorithms to page-in hardware modules during runtime. Rakhmatov et al. explore issues related to compilation and operating system technologies to support RTR on a system of FPGAs and multiprocessors.

Generally speaking, configurable computing techniques may not be suitable for all computationally intensive applications. Applications have a number of factors that are good candidates for CCM implementation. Three of these

0740-7475/00/$10.00 © 2000 IEEE

factors are extraordinarily high I/O data rate requirements, repetitive operations on huge (or virtually infinite) data sets, and a large number of computational operations per input data point—far beyond the capabilities of general-purpose processors and DSPs. Many multimedia applications and embedded computing applications fall into these categories. Possible product areas include real-time image processing, wireless digital products, simulation accelerators, visual inspection systems, automated target-tracking systems, and avionics. These are areas that traditionally use DSPs.

In addition, applications that need to adapt quickly to environmental conditions may benefit from configurable computing. These applications can be characterized by the need to change computational pipelines or data flows in real-time applications without the threat of data loss. The field of communications offers many challenging problems. Some hot areas in communications that appear to be a good match for configurable computing are high-speed adaptive (IP6) routers, encryption and decryption, and cellular base station management.

Some configurable computing solutions can offer power savings over general-purpose alternatives. These solutions can offer potential advantages in wireless and battery-powered devices. Power consumption savings in CCMs come in four forms. First, configurable computing solutions can exploit specific numerical properties of a given application by varying computational precision. If an algorithm requires only, say, 19 bits of precision for a particular multiplication, then a 19-bit multiplier can be instantiated, saving energy over a fixed 32-bit multiplier. Second, deep computational pipelines can be built within configurable computing designs. Since perhaps tens or hundreds of operations can be performed in a deep pipeline every clock cycle, the overall system clock frequency can be orders of magnitude lower than general-purpose alternatives. Third, configurable computing embedded solutions can fold in many of the sensor/actuator preconditioning and "glue logic" functions on a chip. This provides the distinct advantage of reducing the amount of board-level interchip signaling, thus reducing the work required by large chip-level output drivers. The fourth source of savings comes from the availability of reconfigurable memories on-chip. By dynamically selecting a memory hierarchy that is most suitable for a given application, one can selectively disable parts of memory that are not needed and achieve power savings. Xu and Albonesi's article explores such an approach and demonstrates power reductions on standard benchmarks.

Over the past eight years, the CCM research community has built and tested many prototype configurable computing platforms. Furthermore, there have been a significant number of new commercial ventures focused on the production of configurable computing platforms. Typically, these machines take the form of an FPGA processing array connected to a host general-purpose processor-based system. The host provides system-level control by configuring the FPGA array, setting up input data for processing, and collecting the computed results. When the size of the problem exceeds these resources, the problem is unsolvable using a single configuration, so multiple-configuration "overlays" are required.

Significant challenges exist in the realm of software for CCMs. One challenge is in CCM application development tools. To date, the most successful CCM development environments are hardware description language-based environments (either VHDL or Verilog). This design flow mimics the successful and proven ASIC design flow, but also restricts access to those developers who are well-versed in hardware design. This weakness often leads to much longer application development and debugging time with respect to traditional microprocessor-based environments. Another challenge is in CCM runtime environments, including the management and allocation of reconfigurable resources, RTR scheduling, and hardware–software interaction. These challenges are topics of intense research. There is a need to "lower the bar" required to program and effectively utilize CCM devices. New experimental languages and environments are emerging for CCMs that are more closely linked to the traditional software design flow. The article by Hutchings et al. presents such a framework for designing and debugging CCM applications using the JHDL language.

One inherent advantage of CCMs, especially RTR-capable systems, is their fault tolerance. By adding structures to detect faults, one can reconfigure the hardware to bypass such faults without having to shut down the operation. The article by Saxena et al. presents an architecture (called Reliability Obtained by Adaptive Reconfiguration) that combines standard processor architecture with reconfigurable logic to provide enhanced performance as well as fault tolerance. Another aspect of reliability measurement using FPGAs is explored in the article by Lopez-Buedo et al. that describes a method for online thermal testing by monitoring the output frequency of the on-chip oscillators. By being able to continually monitor the die temperature, one can detect short circuits and other problems that affect reliability.

IN CONCLUSION, CCMs have been coming of age during the past few years. These systems combine the flexibility of programmable systems with the performance of ASIC-like hardware. The articles appearing in this special issue present work exploring some issues related to CCM design, testing, and programming. We hope the reader will find these articles useful and informative. ■

Fadi J. Kurdahi is with the Department of Electrical and Computer Engineering at the University of California, Irvine, where he is currently a professor. His areas of interest are reconfigurable computing and VLSI Design Automation. He received the BE degree in EE from the American University of Beirut, Lebanon, in 1981, and the MS degree and PhD degrees in computer engineering from the University of Southern California in 1982 and 1987, respectively. Dr. Kurdahi was Associate Editor of *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* from 1993 to 1995.

For further research

1. J. Villasenor and B. Hutchings, "The Flexibility of Configurable Computing," *IEEE Signal Processing Magazine*, Sept. 1999.
2. D. Buell, J. Arnold, and W. Kleinfelder, *Splash 2: FPGAs in a Custom Computing Machine*. Los Alamitos, Calif.: IEEE CS Press, 1995.
3. T. Kean and J. Gray, "Configurable Hardware: Two Case Studies of Micro-Grain Computation," *J. VLSI Signal Processing*, vol. 2, no. 1, pp. 9-16, 1990.

Nader Bagherzadeh is with the Department of Electrical and Computer Engineering at University of California, Irvine. Currently, he is serving as the chair of the department. His research interests are parallel processing, reconfigurable computing, computer architecture, and VLSI design. He received a PhD degree in computer engineering from the University of Texas at Austin in 1987.

Peter Athanas is an associate professor in the Bradley Department of Electrical and Computer Engineering at Virginia Tech. His research interests focus on configurable computing and microelectronics. He received his PhD in electrical engineering and his ScM in applied math from Brown University, his MSEE degree from Rensselaer Polytechnic Institute, and his BSEE from the University of Toledo.

José L. Muñoz is at the Defense Advanced Research Projects Agency's Information Technology Office, where he is the program manger for both the Adaptable Computing Systems and the Data Intensive Systems programs.