Introduction

Jürgen Becker

Institut für Technik der Informationsverarbeitung (ITIV),
Universität Karlsruhe (TH), Karlsruhe, Germany
E-mail: becker@itiv.uni-karlsruhe.de
WWW: http://www.itiv.uni-karlsruhe.de

Serge Vernalde

Interuniversity MicroElectronics Center (IMEC), Leuven, Belgium
E-mail: Serge.Vernalde@imec.be
WWW: http://www.imec.be

Biographical notes: Jürgen Becker is full Professor for embedded electronic systems and head of the Institute for Information Processing (ITIV) at Universität Karlsruhe (TH). His actual research is focused on industrial-driven System-on-Chip (SoC) integration with emphasis on adaptivity and reconfigurability in hardware/software development for automotive and communication systems. He is Vice-President ('Prorektor') for Studies and Teaching at Universität Karlsruhe (TH). He is author and co-author of more than 150 scientific papers, and active as Chairman of international conferences, Chair of GI/ITG Technical Committee of Architectures for VLSI Circuits, Associate Editor of IEEE Transactions on Computers, and Senior Member of the IEEE.

Serge Vernalde received the Electrical Engineering Degree in 1990 at the University of Leuven, Belgium. In 1990 he joined the IMEC-laboratory, focusing on design technology for wireless and multimedia systems. Between 1995 and 2004, he has been heading multiple research groups in the fields of digital broadband transceivers, reconfigurable systems and design technology for heterogeneous multiprocessor platforms. He is currently technical business director at IMEC, responsible for the partner relations in the context of IMEC's Multi-Mode Multi-Media (M4) program. He is author or co-author of over 60 scientific publications in international conferences and journals and was the general chair of the FPL 2004 conference.

In the last decade, it has become obvious that embedded systems are an essential part of our every day lives. Additionally, with the advent of VLSI system level integration and system-on-chip, the centre of gravity of the computer industry is now moving from personal computing into embedded computing. FPGAs have emerged as a critical component in many of today's most demanding embedded system designs. Since FPGA's manufacturers have come closer to filling the performance gap between FPGAs and ASICs, Field Programmable Logic devices are no longer just a prototyping vehicle for Application Specific Integrated Circuits (ASICs), but are increasingly found in computing systems where the user configurable logic and interconnects offer unique advantages, enabling them to become active players as components in embedded systems. As characteristic of an emerging technology, current FPGA research is taking diverse directions. At the lowest level, chip design and manufacturing issues such as interconnect technology and testing continue to demand attention. Fundamental questions about the internal organisation and the relative use of logic and routing resources in a chip remain only partially addressed. FPGA research plays a substantial role at system level too. The rapid development of these chips has provided the technological basis for attaining configurable computing solutions with relative ease. As evidence of this technology's implications for the future of computing, FPGAs have led to remarkable execution improvements over more traditional techniques in a large number of applications. This issue is the first of two special issues, and features 12 papers out of 23 papers in total, selected from 38 submissions that represent the diverse problems being addressed today by the FPGA research community. The published papers are extended special editions of excellent papers, which have been presented at the *11th Reconfigurable Architectures Workshop (RAW)* as part of the *18th Annual International Parallel and Distributed Processing Symposium (IPDPS 2004)*. With authors from around the world, these papers bring us an international sampling of significant work.

In the first contribution on 'Multi task hyperreconfigurable architectures: models and reconfiguration problems' by S. Lange and M. Middendorf, the authors present their concept of 'hyperreconfigurable' architectures which can adapt their reconfiguration abilities during run time for increasing the speed of dynamic reconfiguration. These types of architectures use two ways of dynamic reconfiguration steps: In hyperreconfiguration steps they change their ability for reconfiguration and in ordinary reconfiguration steps they reconfigure the actual contexts for a computation within the limits that have been set by the last hyperreconfiguration step. In their paper they study the concept of partial hyperreconfiguration for multitask environments where they propose several models for partially hyperreconfigurable architectures and study corresponding reconfiguration problems for finding optimal (hyper)reconfigurations. Although the problem is known to be NP-complete even for a single task, they identified an interesting special case that can be solved by a polynomial time algorithm even for multiple tasks. Their introduced concept and results are illustrated with a partially hyperreconfigurable example architecture.

A different approach is presented in the second paper 'Online placement for dynamic reconfigurable devices' by A. Ahmadinia, C. Bobda, and J. Teich. In order to use efficiently, the dynamic and partial reconfiguration possibility on FPGAs, one needs a support in the form of operating systems for managing both software and reconfigurable hardware processes. Online placement is one of these management issues that are investigated in this paper. There they present a new approach for online placement of modules on reconfigurable devices, by managing the occupied space rather the free space on the device. Also they propose an optimisation of communication between running modules themselves and outside of the chip. Their experimental results show a considerable decrease in communication and routing costs. Also task scheduling between hardware and software is investigated, and a communication infrastructure for the online placement has been proposed and implemented. As a contrast to digital FPGA based approaches the third contribution deals with fieldprogrammable analog arrays (FPAAs). In their paper 'Developing large-scale field-programmable analog arrays for rapid prototyping' T.S. Hall, C.M. Twigg, P. Hasler, and D.V. Anderson give an overview of the many past and present FPAA designs and introduce a new, large scale FPAA architecture that offers increased functionality and flexibility in realising analog signal processing systems. In addition, experimental data from a testbed FPAA are shown.

An actual topic of reconfigurable hardware/software systems is addressed by the fourth paper 'On-demand FPGA run-time system for flexible and dynamical reconfiguration' by M. Ullmann, M. Hübner, B. Grimm, and J. Becker. They point out the increasing problems of today's automobile manufacturers where an increasing number of built in microcontroller and ASIC based automotive control devices cause a growing design complexity of automotive applications, which has additional impact on communication, power consumption, available space and cost. They present an approach for a flexible versatile FPGA based MicroBlaze-softcore processor controlled runtime system supporting a resource saving function multiplex by using a locally controlled partial runtime reconfiguration of slotted hardware ECU modules, which can be a possible partial solution for the arising problems of embedded automotive as described above.

Another interesting topic is described in the fifth paper 'Symmetric encryption in reconfigurable and custom hardware' by E. Swankoski, N. Vijaykrishnan, R. Brooks, M. Kandemir, and M.J. Irwin. They address the growing needs for security in today's informational society. Fast implementations of symmetric key crypto algorithms are needed, which satisfy the demands of commerce, government, and private persons. In their paper they study the Advanced Encryption Standard (AES) algorithm and its optimised implementations in both software and hardware. Also, they study the performance of optimised AES implementations in Xilinx Spartan-3 and Virtex-II field programmable gate arrays as well as in standard cell ASICs.

The sixth contribution 'Integrated modelling and generation of a reconfigurable network-on-chip' by D. Ching, P. Schaumont, and I. Verbauwhede presents an integrated modelling, simulation and implementation tool for reconfigurable network-on-chip (NoC). A key contribution is that the network architecture can be cosimulated with the embedded application software in an early design stage. This allows an energy and performance tuning of the NoC early on. The described system simulation supports multiple instructionset simulators together with the NoC, and obtains cycle accurate performance metrics. A high level description of the network-on-chip is next converted into VHDL. This way, an optimal network configuration can be determined easily. They discuss their approach by designing a flexible network-on-chip and present implementation results after mapping into FPGA.

The next paper 'Improving Java performance using dynamic method migration on FPGAs' by E. Lattanzi, A. Gayasen, M. Kandemir, N. Vijaykrishnan, L. Benini, and A. Bogliolo addresses the issue of reconfigurable Java coprocessors. In their work they propose and analyse a microprocessor with FPGA coprocessor architecture with efficient shared-memory communication support. Furthermore, they describe a complete runtime environment that supports dynamic migration of Java methods to the coprocessor, and quantitatively analyse speedups achievable under a number of system configurations using an accurate complete system simulator. The eighth contributed paper by N.A. Saqib, F. Rodíguez-Henriquez, and A. Díaz-Pérez deals with 'A reconfigurable processor for high speed point multiplication in elliptic curves'. In this paper they present a generic architecture for the computation of elliptic curve scalar multiplication over binary extension fields. In order to optimise the performance as much as possible they designed a parallelised version of the wellknown Montgomery point multiplication algorithm implemented on a reconfigurable hardware platform (Xilinx XCV3200). The proposed architecture allows the computation of the main building blocks required by the Montgomery algorithm in an efficient manner.

The paper 'A probabilistic analysis of fault tolerance for switch block array in FPGAs' by J. Huang, M.B. Tahoori, and F. Lombardi presents a new approach for the evaluation of FPGA routing resources in the presence of faulty switches and wires. Switch stuck-open (switch permanently off) and stuck-closed faults (switch permanently on) as well as wire faults are addressed. This study is directly related to the fault tolerance interconnect problem domain for testing and reconfiguration at manufacturing and runtime application. Signal routing in the presence of faulty resources is analysed at switch block and switch block array levels. Probabilistic routing (routability) is used as figure of merit for evaluating the programmable interconnect resources of FPGA architectures. The proposed approach is based on finding a permutation (one to one mapping) between the input and output endpoints. A probabilistic approach is also presented to evaluate fault tolerant routing for the entire FPGA by connecting switch blocks in chains as required for testing and to account for the I/O pin restrictions of an FPGA chip. The results are reported for various commercial and academic FPGA architectures.

Although FPGAs are a convenient platform for development and prototyping, the large size of configuration bitstreams of several megabytes is an important reason why FPGA implementations of control dominated functions still have problems competing with slim microcontroller based software solutions, since the configuration code has to be stored on board in a large local persistent memory, which becomes an important cost factor. The paper ‘Real-time configuration code decompression for dynamic FPGA self-reconfiguration: evaluation and implementation’ by M. Huebner, M. Ullmann, and J. Becker addresses this topic. Since Xilinx Virtex FPGAs have the possibility of dynamical partial run-time reconfiguration this feature can be used with many different partial configuration bitstreams for substitution of parts in reconfiguration memory. In this context the authors present an interesting approach where precompressed bitstream data which are stored locally on rather small Flash memories, are decompressed at runtime by a fast pipelined decompressor mechanism on FPGA before being fed into the device’s internal configuration access port (ICAP). It should be annotated that the described decompressor module is used as an integrated component of the FPGA runtime system, which was introduced as fourth paper in this issue.

The 11th contribution, which addresses partial reconfiguration by H. Kalte, B. Kettelhoit, M. Koester, M. Porrmann, and U. Rückert presents ‘A system approach for partially reconfigurable architectures’ on FPGAs.

In this paper they show a new, realisable approach for dynamic system integration on Xilinx Virtex FPGAs. In contrast to existing approaches that consider fixed slots for module placement, their approach allows finegrained placement of modules with variable width along a horizontal communication infrastructure. Hardware designs can be constrained for an optimal integration into their system approach without reasonable cost. By means of a new simulation framework they show the predominance of the proposed 1D-approach over 2D-approaches.

The last contribution in this issue ‘Hardware assisted two dimensional ultra fast online placement’ by M. Handa and R. Vemuri addresses the problem that time taken to find a placement location for a dynamic task is an overhead on the application execution time in a partially reconfigurable system. Thus, inherent parallelism of the reconfigurable hardware can be used for speeding up the placement process. In this paper, they propose the use of the reconfigurable hardware resources on the target platform for speeding up an online placement algorithm. Three different architectures are presented for two dimensional online placement. Each architecture makes different tradeoffs between area usage, memory requirement and execution time. These architectures are capable of achieving very fast placement while using a very small number of hardware resources.

Acknowledgements

We would like to thank all the people who contributed to make this special issue possible. These include the authors who submitted their papers, the reviewers, the editor in chief, and the editorial and production staff at the IJES. We hope the readers will enjoy reading this special issue as much as we enjoyed compiling it.