Parallel Computing: From Multicores and GPU's to Petascale
B. Chapman et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-530-3-581

ParaFPGA: Parallel Computing with Flexible Hardware

Erik H. D'HOLLANDER [a], Dirk STROOBANDT [a] and Abdellah TOUHAFI [b]

[a] *ELIS Department, Ghent University, Belgium*
[b] *IWT Department, Brussels University Association, Belgium*

Abstract. ParaFPGA 2009 is a Mini-Symposium on parallel computing with field programmable gate arrays (FPGAs), held in conjunction with the ParCo conference on parallel computing. FPGAs allow to map an algorithm directly onto the hardware, optimize the architecture for parallel execution, and dynamically reconfigure the system in between different phases of the computation. Compared to e.g. Cell processors, GPGPU's (general-purpose GPU's) and other high-performance devices, FPGAs are considered as flexible hardware in the sense that the building blocks of one or more single or multiple FPGAs can be interconnected freely to build a highly parallel system. In this Mini-Symposium the following topics are addressed: clustering FPGAs, evolvable hardware using FPGAs and fast dynamic reconfiguration.

Keywords. FPGAs, flexible hardware, dynamic reconfiguration, parallel processing, high performance computing

Introduction

Within the world of parallel computing, FPGAs constitute an attractive option for specialized applications, because of their flexibility, massively parallel framework, relatively low energy consumption and continually expanding features. Due to their inherently parallel electronic components and computation elements, FPGAs can be synthesized into a highly parallel computing architecture. This allows one to construct dataflow architectures and to map an algorithm onto the hardware. FPGAs have been used as accelerators in supercomputers, number crunching for multimedia applications, simulators for new computer architectures and standalone computing platforms. It is clear that no single computing paradigm fits all these applications, and this is reflected in the different topics of this Mini-Symposium: clustering FPGAs, developing evolvable hardware and new techniques for fast dynamic reconfiguration.

The selection and review of the number of papers presented in this Mini-Symposium could not be possible without the extensive help of the program committee. We thank these members for reviewing the submitted papers and selecting the best ones to be presented here. The members of the program committee are listed below.

We also thank Prof. Jonathan Rose of the University of Toronto for giving a keynote on *Experience, Frustration and Hope with Acceleration of Computation on FPGAs*.

1. Contributions

The first two papers address clustering and communication between FPGAs. In *Software versus Message Passing Implementations for FPGA Clusters* by Creedon and Manzke, a speed-area trade-off was made between the hardware and software implementation of a fast interconnection between FPGAs in a cluster. Interestingly, the authors developed a message passing interface which mimics the well-known MPI standard.

RAPTOR, a Scalable Platform for Rapid Prototyping and FPGA-based Cluster Computing by Porrmann et al., presents the development of a Gigabit-interconnected FPGA cluster used to emulate new multiprocessor hardware architectures. The RAPTOR hardware/software environment has been used to emulate large multiprocessor systems on a chip (MPSoC) and Kohonen self-organizing maps (SOM). In addition, the RAPTOR platform enables the development of powerful hardware accelerators, e.g., to speed-up large scientific simulations.

The next two papers address speeding up the development of evolvable hardware. Evolvable hardware consists of using genetic techniques to develop the optimal or near-optimal implementation of a combinational circuit such as a digital filter, or a complex controller. In *Speeding up Combinational Synthesis in an FPGA Cluster* by Pedraza et al., the computational intensive calculation is delegated to an FPGA in the so-called Fitness Calculation Unit. The speedup over a conventional HPC cluster depends on the search space, and is typically one or more orders of magnitude.

In *A Highly Parallel FPGA-based Evolvable Hardware Architecture* by Cancare et al., another approach is taken, in which the evolution of the candidate solutions is programmed at runtime using partial reconfiguration. This means that in the evolvable region, the quality of the solutions is calculated, while in another part of the evolvable region, new candidates are generated using partial reconfiguration. In this way it is possible to partially or completely hide the reconfiguration time.

Dynamic reconfiguration is the topic of the last three papers. In *Applying Parameterizable Dynamic Configurations to Sequence Alignment* by Davidson et al., an existing algorithm for DNA alignment is reorganized for execution on a smaller FPGA using runtime reconfiguration. In this way, the same application can run on a smaller and cheaper FPGA.

The paper *Towards a More Efficient Runtime FPGA Configuration Generation* Abouelella et al., gives a new insight into generating and minimizing the configuration time for a family of parameterizable bitstreams. A stack machine is presented which allows to compile a parameterizable configuration into configuration bits which are injected into the configuration memory. Parameterizable bitstreams allow a smaller and faster reconfiguration.

Finally, the paper *ACCFS – Virtual Filesystem Support for Host Coupled Run-Time Reconfigurable FPGAs* by Strunk et al., represents a file system for storing runtime reconfigurable modules, which can be loaded on demand by the user. This allows a multitude of compute kernels to be stored in a virtual file system for easy access by a multiple of users in multiple contexts.

2. Conclusion

The contributions presented in this Mini-Symposium show that there exist many opportunities for research and development in the area of parallel computing with FPGAs. Despite the existence of specialized architectures, it is expected that FPGAs will grow in flexibility and performance, and contribute to the parallel computing community in the future.