# Towards an Automated Design of Application-specific Reconfigurable Logic

Peter Zipf and Manfred Glesner

Institute of Microelectronic Systems,
Darmstadt University of Technology,
D-64283 Karlstr. 15, Darmstadt, Germany.
`zipf@mes.tu-darmstadt.de`

Reconfigurable architectures can bridge the gap between programmable solutions (processors, DSPs) and highly optimised ASICs if they are structured in an application-specific way. Two of the main disadvantages of reconfigurable logic are its low functional density, caused by the area overhead introduced by configuration logic and memory, but also reduced clock frequencies, caused by extended signal paths and sub-optimal placement and routing possibilities. In particular, fine-grained generic structures like FPGAs suffer from this deficiency and some coarse-grained architectures were developed to correct this situation. Coarse-grained reconfigurable architectures expose a high performance but they are designed more specific to their application domain in terms of the hardware functions they directly provide inside their cells. A second way to increase functional density in reconfigurable architectures is to apply run-time reconfiguration to reuse the resources in time. In [1], Wirthlin et al. define a functional density measure. Based on this measure, it can be shown that the functional density of reconfigurable architectures can be increased by run-time reconfiguration. On-the-fly changes of structural properties of a circuit must be directly supported by architectural features of the hardware and dynamic reconfiguration is therefore an intensively investigated area.

Our previous work [2] has shown, that area and energy efficiency can only be achieved by highly adapted circuits which can be characterised as application-specific reconfigurable architectures, a result which is supported also by related work [3]. Dynamic reconfiguration is an inherent feature of such circuits.

The main challenges for application-specific reconfigurable architectures are their design and the organisation of the reconfiguration sequencing. Associated with this are open issues on how to integrate them into an embedded system environment, or, more specific, into a real-time processing system. Currently, application-specific reconfigurable structures must be designed manually for each situation, resulting in high NRE costs and a complete lack of programming tools. An integration must also be done manually and no general programming model is supplied.

The objective of our current work is to develop the methodologies and supporting algorithms to advance an automated design of the reconfigurable fabric itself and its integration into a processor environment and tool flow. The integration strategy we want to support is the systematic extension of a processor's instruction set architecture (ISA). Looking at the micro-architecture, the fabric shall be attached directly inside the processor data path as an additional function unit. It has been shown in [4] that such a combination has the capability of exploiting theoretical speed-ups to a high extend.

The closest connection of a processor and a reconfigurable unit is an integration into the processor data path and an according extension of the ISA of the processor to include instructions to configure the extra unit and execute commands on it. Because the CPU register file is accessed directly, no additional data transfer between the CPU and the reconfigurable unit is required. As hardware accelerators usually process large sets of data, having only access to the register file creates a bottleneck and can lead to an under-utilisation of the accelerator. This situation can be improved if the functional unit is given a memory interface of its own while still remaining an element of the processor data path. Based on our experience with earlier designs, such a solution can be identified as the most general one. Some of our work is currently directed towards a deeper investigation of such a coupling concept, including a possible definition of a general interface for the functional units.

In the targeted design methodology for the reconfigurable fabric, the data flow graphs of the required application-specific operations will be used to infer a general structure providing the overall functionality. Dynamic reconfiguration is used to optimise the area/speed trade-off for the resulting structure. One part of the optimisation consists in the rearrangement of the execution times of the involved functional components at different abstraction levels.

To apply scheduling algorithms to DFGs describing a partial or complete functional unit, an integrated design flow must be defined, comprising also a conditioning of the initial functions, inferring selected parts of the graphs to construct new elementary flows, a reconfiguration-aware retiming, and an efficient structure generation based on the resulting schedule. The backbone of this flow must be the notion of dynamic reconfiguration as an optimisation keystone. Our future work will be focused on the integration of this concept and the exploration of different possibilities of reconfiguration optimisation.

## References

1. Wirthlin, M., Hutchings, B.: Improving functional density using run-time circuit reconfiguration. IEEE Transactions on VLSI Systems **6** (1998) 247–256
2. Pionteck, T., Kabulepa, L.D., Glesner, M.: Exploring the Capabilities of Reconfigurable Hardware for OFDM-based WLANs. In: International Conference on Very Large Scale Integration of System-on-Chip. (2003) 161–166
3. Blume, H., Huebert, H., Feldkämper, H., Noll, T.: Model based exploration of the design space for heterogeneous Systems on Chip. In: Proceedings of the IEEE ASAP Conference, San Jose, USA (2002) 29–40
4. Vassiliadis, S., Wong, S., Gaydadjiev, G., Bertels, K., Kuzmanov, G., Panainte, E.: The molen polymorphic processor. IEEE Transactions on Computers **53** (2004) 1363–1375