10. Conclusions

The wide availability of reconfigurable architectures has opened new perspectives to the area of high performance data processing. Providing spatial structures, which can be altered using structural code, those architectures are situated between hardwired ASICs and microprocessor solutions, providing both flexibility and performance, as needed in demanding application areas like mobile communication or networking.

While the classical fine grained FPGAs are still the dominating devices in the reconfigurable area, coarse grained reconfigurable architectures have been developed as an alternative approach, which promise better suitability for the area of reconfigurable computing than fine grained FPGAs. As has been shown in chapter 2 of this thesis, a wide range of such architectures based on quite different approaches and ideas has been developed in the recent years.

A common issue of all reconfigurable architectures is the presence of structural code, which describes the spatial arrangement of resources on the device. The classic area, which features such code, has been VLSI design. Thus, many algorithms from design for VLSI circuits have found their way into development systems for reconfigurable architectures. In fact, due to their fine granularity, the design flow for FPGAs resembles mostly the one known from VLSI design. Basic steps like technology mapping and placement and routing have also moved into design environments for coarse grained architectures. The general approach for this design flow and common algorithms have been reviewed in chapter 3.

In chapter 4, actual existing development frameworks for the architectures discussed in chapter 2 have been presented, identifying and classifying the basic design steps. Advantages of coarse grained architectures are the lower complexity of the design problem as well as the higher level of input languages. These advantages show up in the presented frameworks, with some architectures allowing for greedy algorithms in important design steps, or for input specifications in standard high level programming languages like C or FORTRAN.

As an observation from the range of existing coarse grained reconfigurable systems and their frameworks, a general problem of these architectures can be noted. While bit-level FPGAs can be seen as universal, being capable of implementing arbitrary circuits, coarse grained architectures are often suitable only for a specific range of applications, while performing poorly for others. If such an application domain is given, the problem arises, how a reconfigurable architecture capable of processing the given applications should look like. Thus, an according design space exploration has to take place. In chapter 5, the general

problem of this exploration process has been described, identifying the two basic operations of status evaluation and status generation (see section 5.1). In addition, a selection of existing design space exploration systems is presented, showing different approaches for the realization of the two basic operations and the degree of user involvement.

Based on the previous work discussed in chapters 2 to 5, a novel design space exploration framework called KressArray Xplorer is presented in this thesis, which allows to find a suitable coarse grained architecture for a given set of applications. The design space used for the exploration resembles the KressArray architecture family, which has shown promising results in an earlier prototype. The exploration process includes also a synthesis step, which is done by an according mapping subsystem. Thus, it is made sure, that the resulting architecture is in fact suitable for the given application domain. The goals achieved by the framework can be divided into three major parts, which are described in the following:

(1) KressArray Design Space

As a basis for the exploration process, the KressArray design space defining the KressArray architecture family has been presented in chapter 6. Being an extension of the historic KressArray-I, this family features a large variety of architectures, which can be used for different application ranges of reconfigurable computing. The following results and goals can be identified for this design space:

- The KressArray family is backward-compatible to the historic KressArray-I, allowing reuse of surrounding hardware structures, like those based on the Xputer paradigm.
- The envisioned architectures allow for an efficient implementation of datapaths, including support of if-else constructs and loops. To enable recursive datapaths and loop implementation, cycles in datapaths are allowed using pseudo operators (see section 6.1.3).
- For the implementation of loops, a novel approach for KressArrays using two specialized operators for data stream synchronization has been introduced in section 6.1.4. This implementation does not require any further external control.
- The design space supports processing elements featuring two ALUs (see section 6.2.3), which may be used either independently or together, providing a more complex function. It is possible to define for each single operator, if it uses up a whole processing element or if it may be paired with another function.

- Arithmetic operators with two outputs are allowed (see section 6.2.4). This is useful as sometimes, two results are automatically generated by the implementation of the operator, e.g. division with remainder.
- A variety of three different basic communication resources is available, which can be combined arbitrarily. Thus, a wide range of possible combinations is possible, allowing for a specific choice for a given application set.
    - Nearest neighbor connections can be defined in four directions, with unidirectional or bidirectional ports.
    - Backbuses spanning several processing elements in a row or column may have a specific length of segments and a specific offset for the first segment, thus allowing a large variety of backbus structures.
    - A serial global bus being adopted from the KressArray-I prototype may be used for connections between arbitrary PEs.
    - External extensions of the communication can be considered by specifying a torus structure, which can be set for each individual nearest neighbor connection.
- For the backbuses, different implementations are possible. A distributed static control structure for backbuses with multiple writers using moderate hardware effort has been introduced in section 6.3.3.2.
- Heterogeneous structures (see section 6.4.2) allow for the definition of KressArrays with embedded special resources, like e.g. memory blocks. Furthermore, a more efficient design of KressArrays is possible, as complex operators need not be available in every processing element.
- Three different types of I/O ports for data streams can be defined, extending the application of KressArrays to a wide range of embedding structures (see section 6.4.3):
    - Ports over the global bus retain compatibility with the historic KressArray-I architecture.
    - Edge ports allow I/O over array borders. The location of virtual ports can be defined freely, so a large variety of surrounding architecture scenarios can be considered.
    - Internal ports can be used to implement a library concept. Furthermore, such ports allow the connection of the datapath to embedded special resources.

(2) Multi Architecture Datapath Synthesis System

In order to execute the synthesis step during the exploration process, and for the generation of configuration data, a multi architecture datapath synthesis system (MA-DPSS) has been developed. The following features of this system can be identified:

- Being backwards compatible to the historic DPSS, the MA-DPSS can be integrated into the CoDe-X framework for Xputer-based accelerators. This includes the use of the high-level ALE-X language for input specification (see section 7.1.1), which allows for a convenient way of application development.
- Like the name suggests, the MA-DPSS is capable of synthesizing applications on all architectures of the KressArray design space, thus considering the different types of communication resources, heterogeneous structures, double-output operators, double ALU PEs and the other features described above.
- Besides the mapping for the reconfigurable architecture itself, the MA-DPSS generates additional configuration code for optional parts requiring schedules for data transfers, including a central control unit and backbuses with multiple writers.
- An intermediate file format used by the system allows the iterative refinement of a mapping, as the output of a mapping can be used directly as input again.
- The MA-DPSS allows the fixing of PE positions, thus enabling a library concept on mapping level (see section 7.3.4).
- The synthesis process itself is based on a heuristic and can be steered by a large variety of parameters, including the possibility to specify duration of the synthesis and the separate definition of preferences for each communication resource.
- In addition to the possible fine-tuning of the synthesis, also an automatic mechanism for an adaptive cooling schedule is provided for inexperienced users (see section 7.3.1.3).
- The MA-DPSS generates statistic data, which is used in the exploration process (see section 7.4.1.6).

(3) Design Space Exploration

In chapter 8, a novel design space exploration approach for coarse grained architectures has been introduced. The approach is influenced by two research areas, design guidance systems and approximate reasoning. The features of the Xplorer design space exploration approach are summarized in the following:

- The approach is based on user interaction, retaining the total control of the process at the designer. Thus, the designer acts as a corrective element in the process, which allows for better exploration results.
- Although the core of the exploration process is based on a single application, the framework allows for the consideration of an application set. The designer selects one of these applications, being supported by an architecture estimation tool (see section 7.2).
- During the core exploration, the designer is supported by an approximate reasoning system based on fuzzy logic (see section 8.4.3). Furthermore, the use of fuzzy logic allows for tolerance in observed statistics, which are used to generate the suggestions.
- The knowledge base for the generation of suggestions can be easily extended by the user. The designer can easily incorporate his or her experience into the system, thus extending the knowledge base, which is then available to other designers.
- To generate the design suggestions, an analyzer tool collects data from the mapping result. The plug-in based approach provides for easy extensibility for different optimization objectives (see sectio n8.3).
- The user can control the exploration process by a convenient graphical user interface, which supports remote operation and offers an effective project management (see section 8.5).
- The user interface offers also a mapping editor for fine-tuning of mapping results, thus enhancing the synthesis system (see section 8.5.4), and an architecture editor (see section 8.5.3), which provides an easy way to alter the current architecture in the exploration process.

In summary, this thesis presents a design space exploration and synthesis framework for the KressArray architecture family. The KressArray family provides a large variety of architectures for different application requirements. The KressArray Xplorer framework provides a means for finding an optimized coarse grained reconfigurable architecture for a given set of target applications. The general exploration approach is based on user interaction with design suggestions by the system. Thus, the user retains the control of the system and can consider factors and preferences beyond the scope of the current implementation.

The Xplorer framework can easily be extended for new optimization objectives by adding new analyzer plug-ins, which provide according statistic data. Furthermore, the knowledge base of the suggestion generation subsystem can be enhanced easily by adding more rules. Thus, the knowledge of the user and the expert knowledge inherent in the system complement each other. The KressArray Xplorer is therefore a flexible, powerful framework for design space exploration.