



# Algorithmic Skeletons for the Design of Partially Reconfigurable Systems



Florian Dittmann  
Stefan Frank  
Franz Rammig

Universität Paderborn

## Motivation

### § Dynamically Reconfigurable Systems

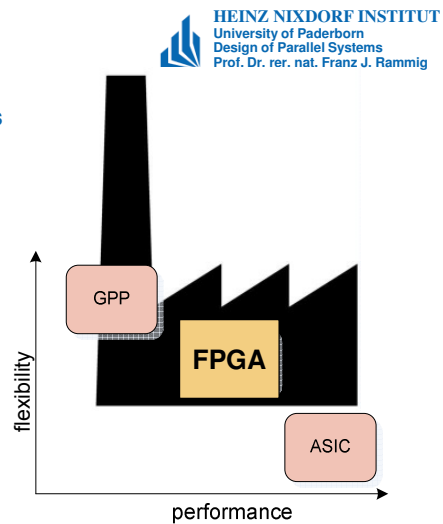
- Promising benefits
  - § Processing in parallel
  - § Runtime reconfiguration
  - Space and time

### § Towards Industrialization

### § Major Requirement

- Comprehensive Design Methods

Development of new Design Techniques  
Making reconfigurable computing mature



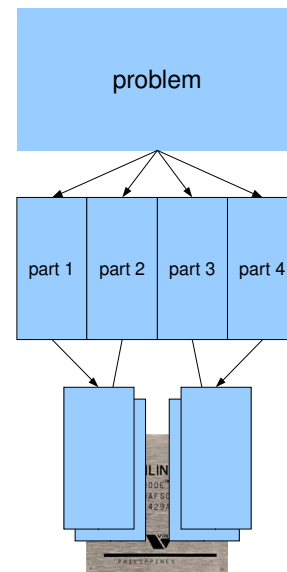
## Content

- FPGAs and partially dynamic reconfiguration
- Algorithmic Skeletons
- The use of skeletons for the partial reconfiguration
  - § Scheduling and Mapping
  - § Design of hardware
- Prototypical implementation
- Conclusion

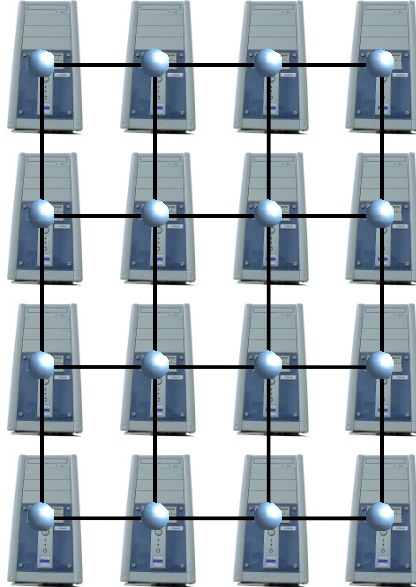
## Partial and dynamic reconfiguration

- § **Partially dynamical reconfiguration for**
  - Mapping of big circuits on small FPGAs
  - Replacement of circuits at run time
- § **Which problems are to be overcome?**
  - Fragmentation
  - Challenging design,
  - Etc.
- § **Problems more abstractly**
  - Decomposition
  - Distribution
  - Communication

Search for related known approaches



## Similar to Partial Run-time Reconfiguration:



§ **Computing cluster of the High Performance Computing domain**

§ **Tasks require**

- Decomposition
- Distribution
- Communication

§ **Which promising design method fits?**

**consider organization of parallel programs**

## Algorithmic Skeletons Background

§ **Objectives**

- Separate structure of a computation from the computation itself
- Free programmer from the implementation details of the structure
- Implementation guidelines for activities and their interactions

§ **Related: design patterns**

- Differences
  - § Design level
  - § Final implementation left to the freedom of the designer

§ **Algorithmic skeletons force the applications to be well-formed**

- Enable to extract valuable information
- Design space exploration on a high level of abstraction
- Static and dynamic optimization of implementations

## Algorithmic Skeletons Overview

### § Tasks

- Programmability and portability
- Structure and behavior of the tasks  
Application level

### § Use of algorithmic skeletons

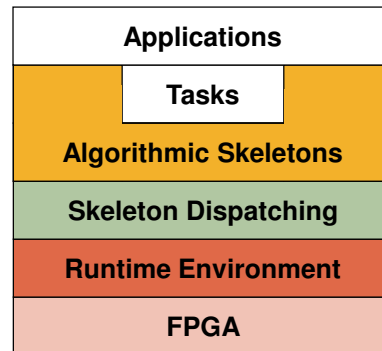
- Wrapping of tasks
- Programming templates

### § Runtime environment

- Partial reconfiguration capabilities

### § Dispatcher

Dynamic Reconfiguration



## Problems of the reconfiguration

### § Problems

- Decomposition
- Distribution
- Communication

### § Moreover

- Portability
- Performance

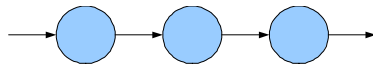
algorithmic skeletons offer a suitable concept for abstraction  
that allows us to reach the goals above

## Algorithmic Skeletons

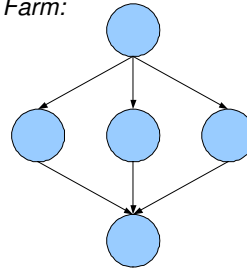
### § Important algorithmic skeletons

- Pipe
- Farm

Pipeline:



Farm:

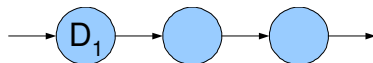


## Algorithmic Skeletons

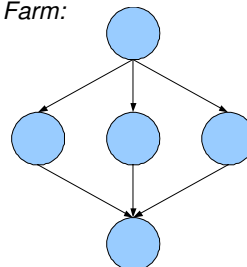
### § Important algorithmic skeletons

- Pipe
- Farm

Pipeline:



Farm:

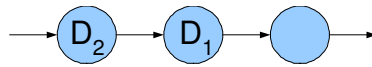


## Algorithmic Skeletons

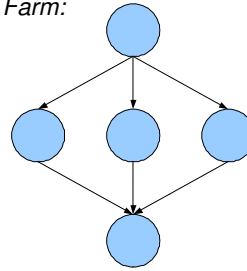
### § Important algorithmic skeletons

- Pipe
- Farm

Pipeline:



Farm:

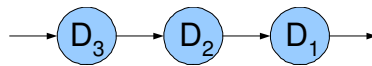


## Algorithmic Skeletons

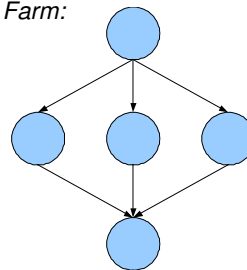
### § Important algorithmic skeletons

- Pipe
- Farm

Pipeline:



Farm:

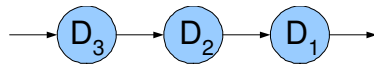


## Algorithmic Skeletons

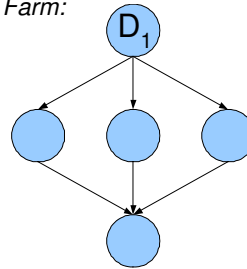
### § Important algorithmic skeletons

- Pipe
- Farm

Pipeline:



Farm:

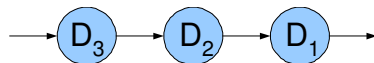


## Algorithmic Skeletons

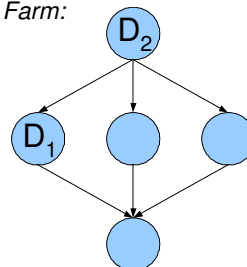
### § Important algorithmic skeletons

- Pipe
- Farm

Pipeline:



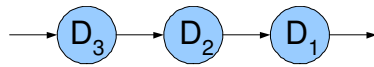
Farm:



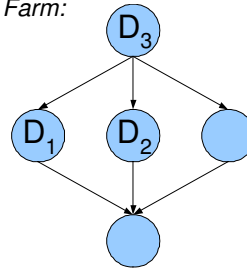
## § Important algorithmic skeletons

- Pipe
- Farm

Pipeline:



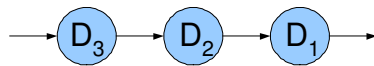
Farm:



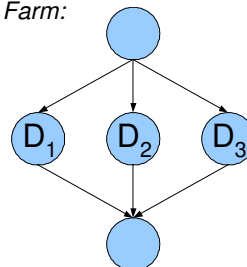
## § Important algorithmic skeletons

- Pipe
- Farm

Pipeline:



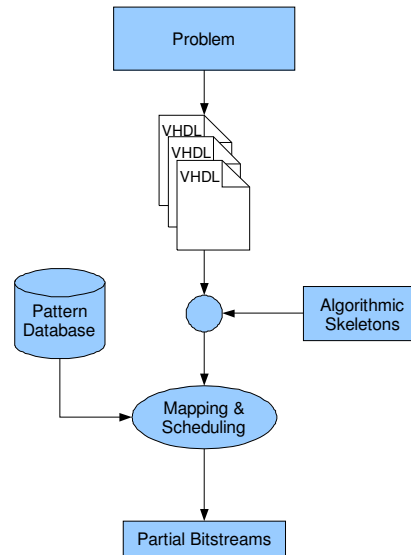
Farm:



**Assumption:** all pipeline stages resp. farm workers running in parallel  
**Needs:** re-thinking to include partial reconfiguration

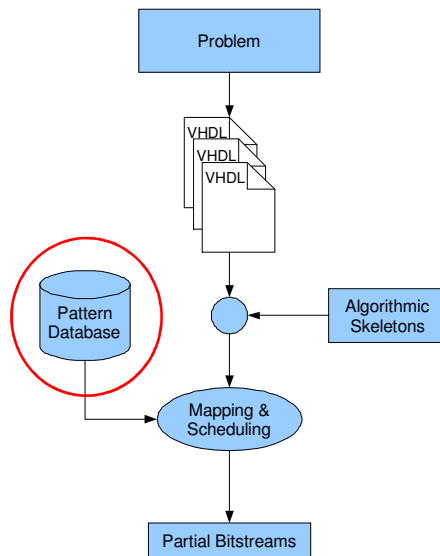
## New concept

- § Problem: large circuits for small FPGAs
- § Two possibilities for the placement of reconfigurable slots:
  - Usage of a pattern
  - individual placement and calculation of the slots



17

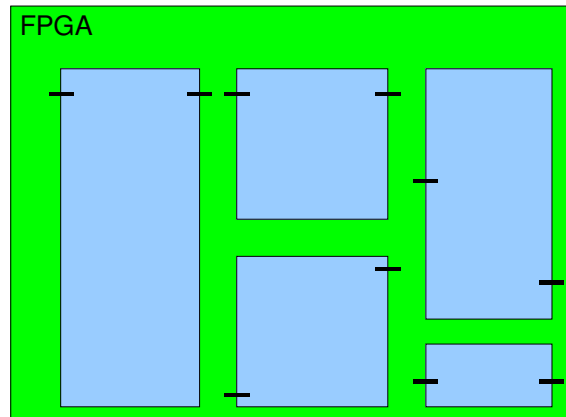
## Pattern Database



18

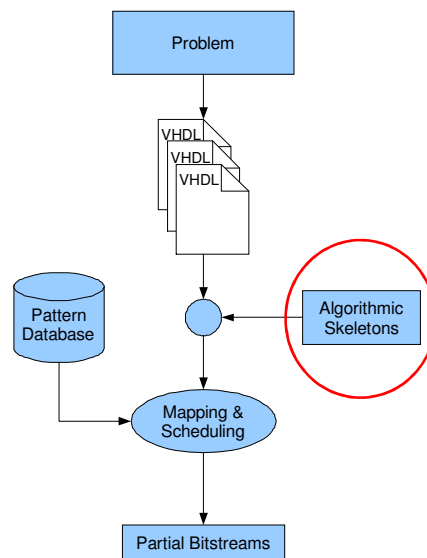
## Pattern Database

- Pattern consists of:
  - § Description of the slot placement
  - § Description of the Bus Macro placement



19

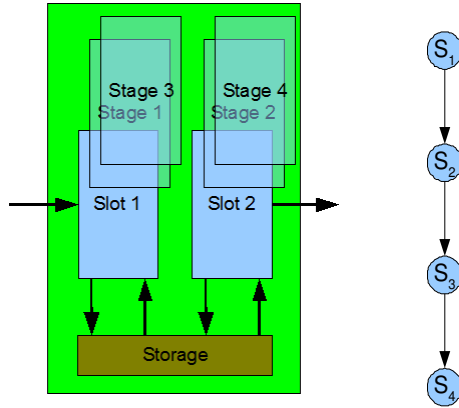
## Pipe and Farm



20

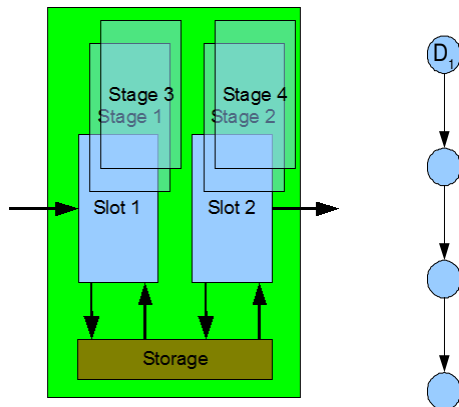
## Pipe and Farm

- The meaning of a pipe in our concept



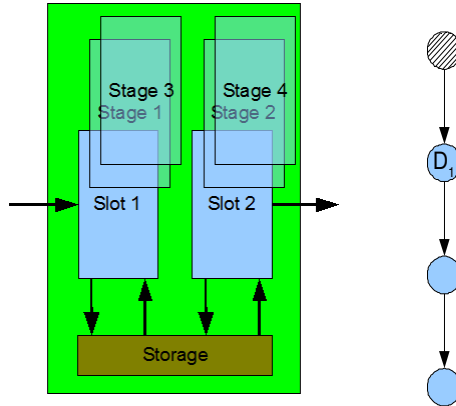
## Pipe and Farm

- The meaning of a pipe in our concept



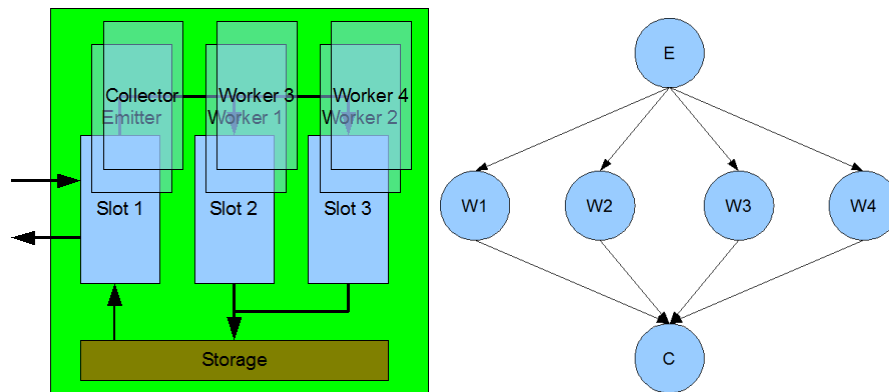
## Pipe and Farm

- The meaning of a pipe in our concept



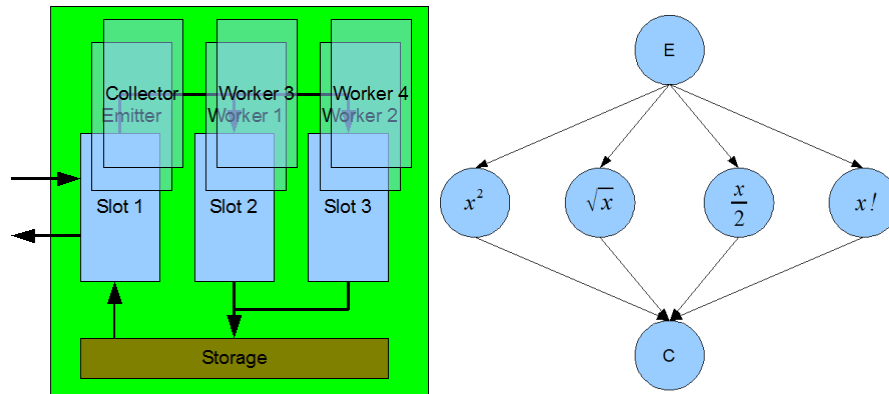
## Pipe and Farm

- The meaning of a farm in our concept

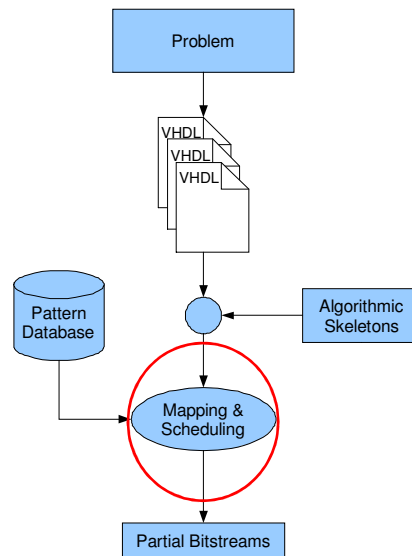


## Pipe and Farm

- The meaning of a farm in our concept

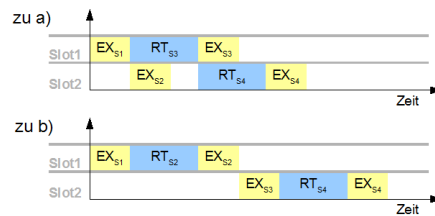
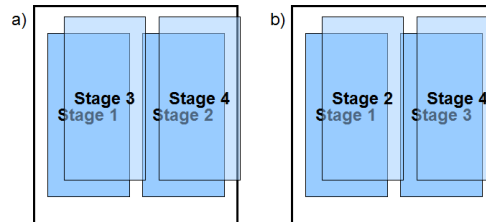


## Mapping and Scheduling



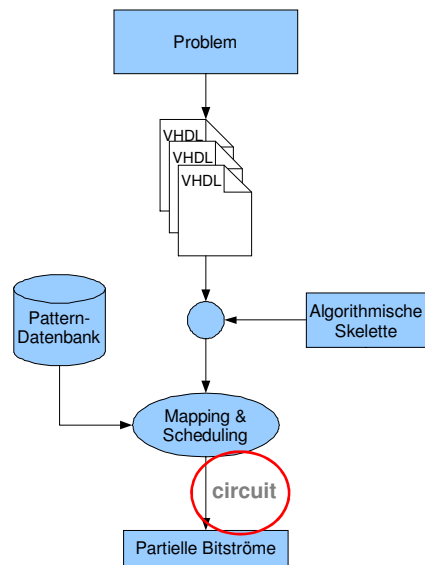
## Algorithmic Skeletons for partial reconfiguration?

- Skeletons contains information about:
  - § the necessary communication between the components
  - § the placement of the components



27

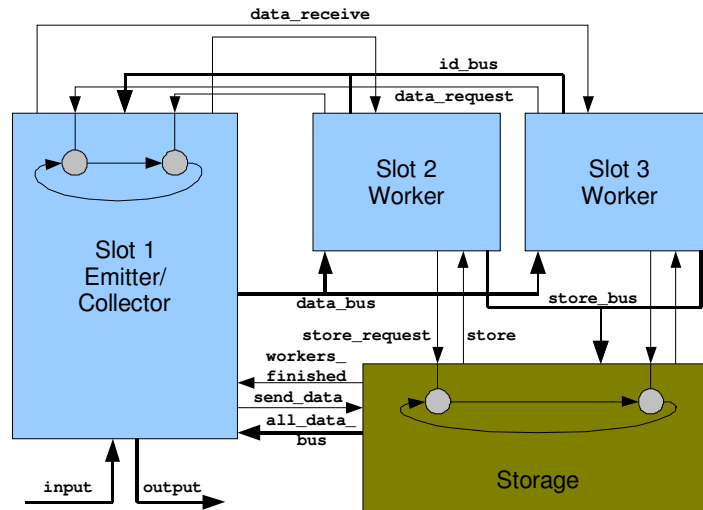
## Technical realization



28

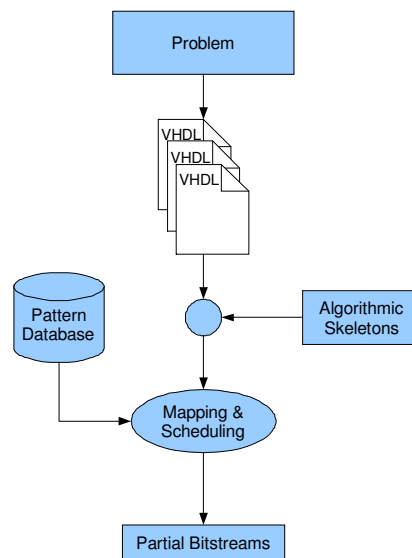
## Technical realization

- Technical realization of a farm:



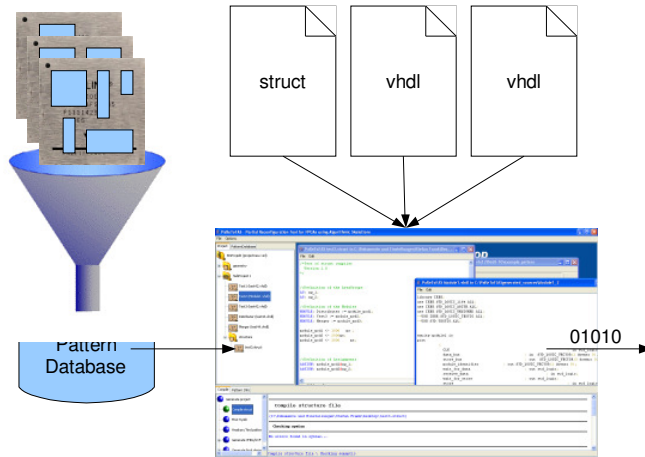
29

## Generating of partial bitstreams



30

- **Partial Reconfiguration Tool for FPGAs using Algorithmic Skeletons**
- The application developer writes a VHDL-file (modules)
- The application developer writes a struct file



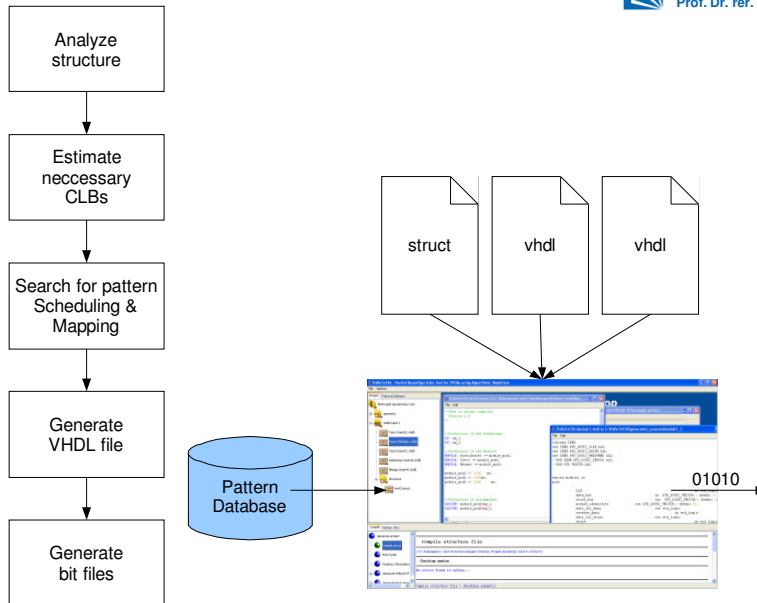
The screenshot shows the PaReToFAS FPGA-Editor software interface. The main window displays a grid of FPGA resources with a highlighted area. A table at the top lists bus macros with columns for Name, AreaGroups, BusMacros, Others, Notes, X1, Y1, Type, In/Out, and Belongs to AG... A file explorer on the right shows the directory structure for the Pattern Database.

Name	AreaGroups	BusMacros	Others	Notes	X1	Y1	Type	In/Out	Belongs to AG...
bm_0		32			30		busmacro_xc4v_l2r_async_narrow	In	ag_0
bm_1		32			28		busmacro_xc4v_l2r_async_narrow	In	ag_0
bm_2		32			26		busmacro_xc4v_l2r_async_narrow	In	ag_0

File Explorer Structure:

- PatternDatabase
  - xilinx
    - Vertex4
      - xc4vx12ff668-10
        - 2slot.pattern
        - 3slot.pattern
        - 4slot.pattern
        - 5slot.pattern
        - 1slot.pattern
        - inhomogeneous
      - xc4vx25ff668-10

## PaReToFAS – Steps of the tool



33

## Conclusion

- § Algorithmic skeletons prove to be applicable for the development of partially reconfigurable circuits
- § Experiments showed us that the idea works
- § Algorithmic skeletons allow for evaluating partially reconfigurable designs on a very high level of abstraction
- § A new interpretation of the skeletons is required
- § Scheduling and mapping can be time-consuming  
Design database

34

## Conclusion (continued)

### § Publications concerning this approach up to now:

- § Götz, Marcelo; Dittmann, Florian; Xie, Tao: Dynamic Relocation of Hybrid Tasks: A Complete Design Flow. In: Proceedings of Reconfigurable Communication-centric SoCs (ReCoSoc'07), Montpellier, France, (2007)
- § F. Dittmann: Algorithmic Skeletons for the Programming of Reconfigurable Systems. In *Proceedings of IFIP SEUS 2007*, Springer LNCS (2007)
- § S. Frank: Einsatz algorithmischer Skelette zur Generierung partiell dynamisch rekonfigurierbarer Schaltungen. *Diplomarbeit Universität Paderborn* (2007)
- § F. Dittmann, S. Frank, S. Oberthür: Algorithmic Skeletons for the Design of Partially Reconfigurable Systems. In *Proceedings of 15th Reconfigurable Architectures Workshop -- RAW 2008*, IEEE (2008)

# Thank you for your attention.

*Florian Dittmann*  
*Stefan Frank*  
*Simon Oberthür*



Heinz Nixdorf Institute  
University of Paderborn  
Fuerstenallee 11  
33102 Paderborn  
Germany

Phone: +49 (0) 52 51/60 65 15  
Fax: +49 (0) 52 51/62 65 02  
Email: oberthuer@upb.de  
<http://www.heinz-nixdorf-institut.de>