

---

DFG Schwerpunktprogramm 1148  
Zwischenkolloquium der dritten Förderphase 2008

# Entwurf von On-Chip Multiprozessorsystemen

Christophe Bobda

Harold Ishebabi

Philipp Mahr

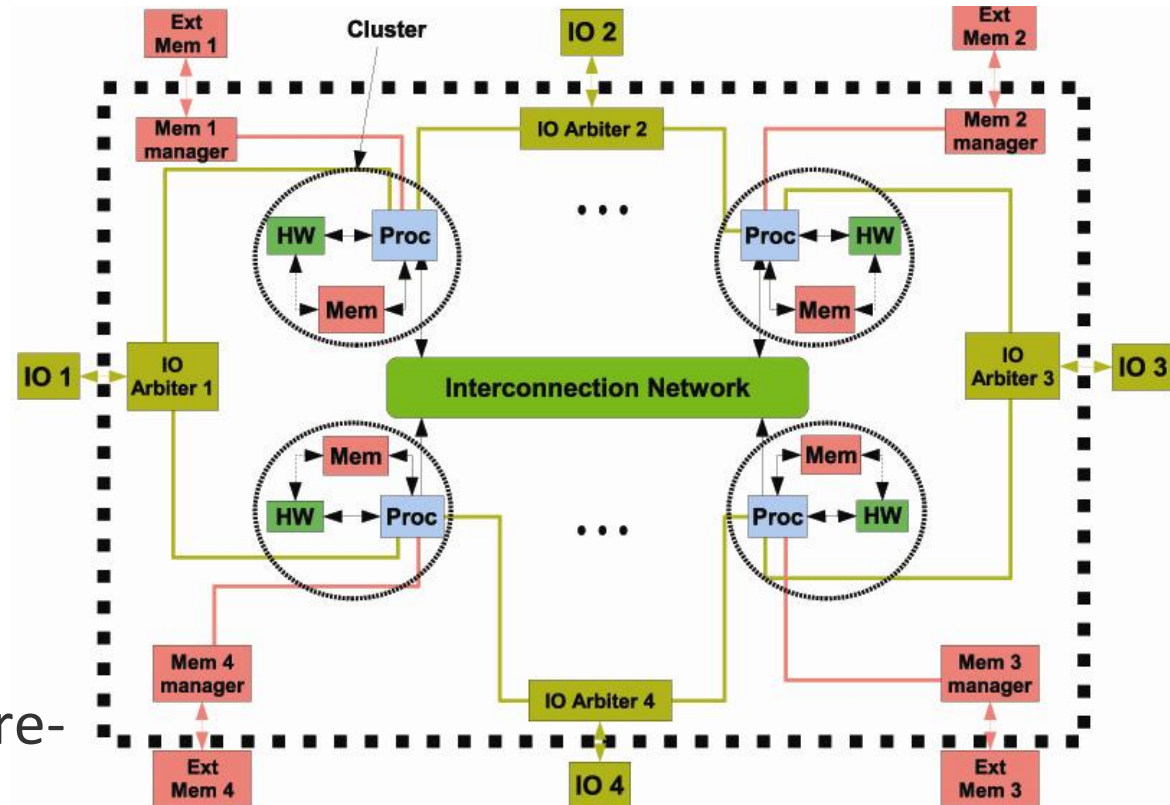
# Das AMoC-Projekt

## Adaptive Multiprocessor on Chip

„Das oberste Ziel soll es sein, einen durchgängigen Entwurfsfluss zu ermöglichen und ein Werkzeug bereitzustellen ...“

### Agenda:

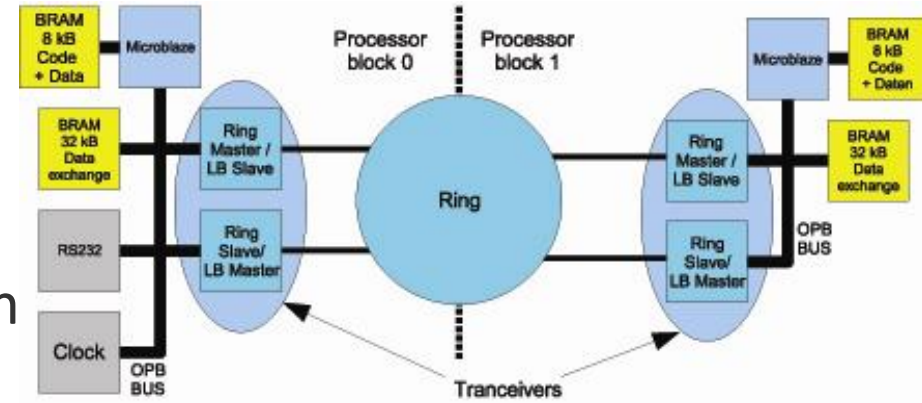
- Integrierte on-Chip-Kommunikation
- Adaptivität der Rechnerressourcen
- Werkzeugunterstützung
- Anwendungen
- Modellierung der Hardwareinfrastruktur



## Kommunikationsnetzwerke

### ■ Ring-Cluster

- Ringtranceiver
- bis zu vier Ring-Cluster können an einen Router angeschlossen werden

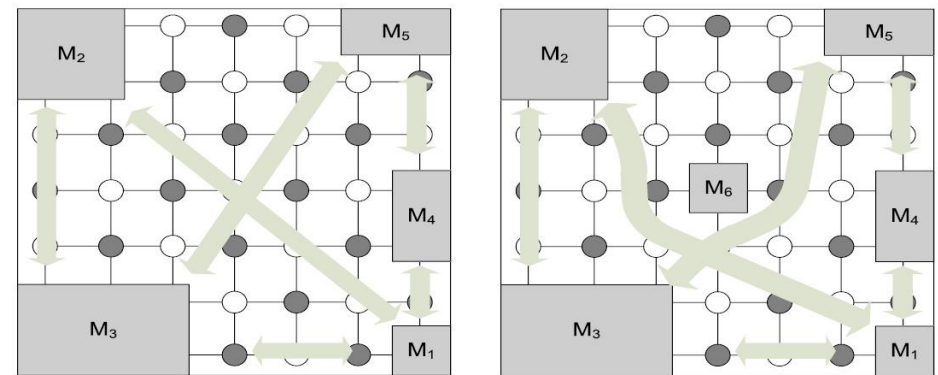


### ■ Sternnetzwerk

- bis zu fünf Komponenten über eine Punkt-zu-Punkt Verbindung an den Router angebunden
- Xilinx MicroBlaze soft-core Prozessoren und FSL

### ■ CuNoC

- Communication Unit Network on-Chip
- Dynamische Platzierung von Module



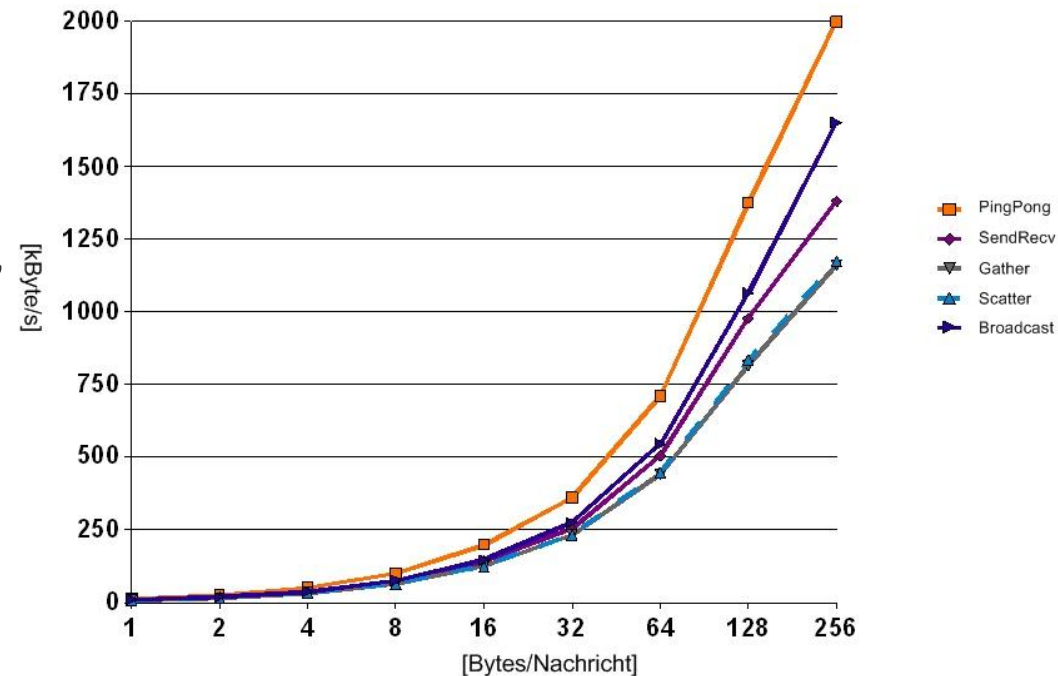
**Publikation:** S. Jovanovic, C. Tanougast, C. Bobda und S. Weber. *CuNoC: A Dynamical Scalable Communication Structure for Dynamically Reconfigurable FPGAs* Elsevier, 10.03.2008 (submitted)

## SocMPI Kommunikationsbibliothek



- On-Chip Message Passing
- kompatibel zum MPI-Standard

MPI\_Barrier, MPI\_BCast,  
MPI\_Bsend, MPI\_Comm\_Rank,  
MPI\_Comm\_Size, MPI\_Initialized,  
MPI\_Init, MPI\_Gather, MPI\_Recv,  
MPI\_Scatter, MPI\_Send,  
MPI\_SendRecv, MPI\_SSend,  
MPI\_Wtime, MPI\_Wtick

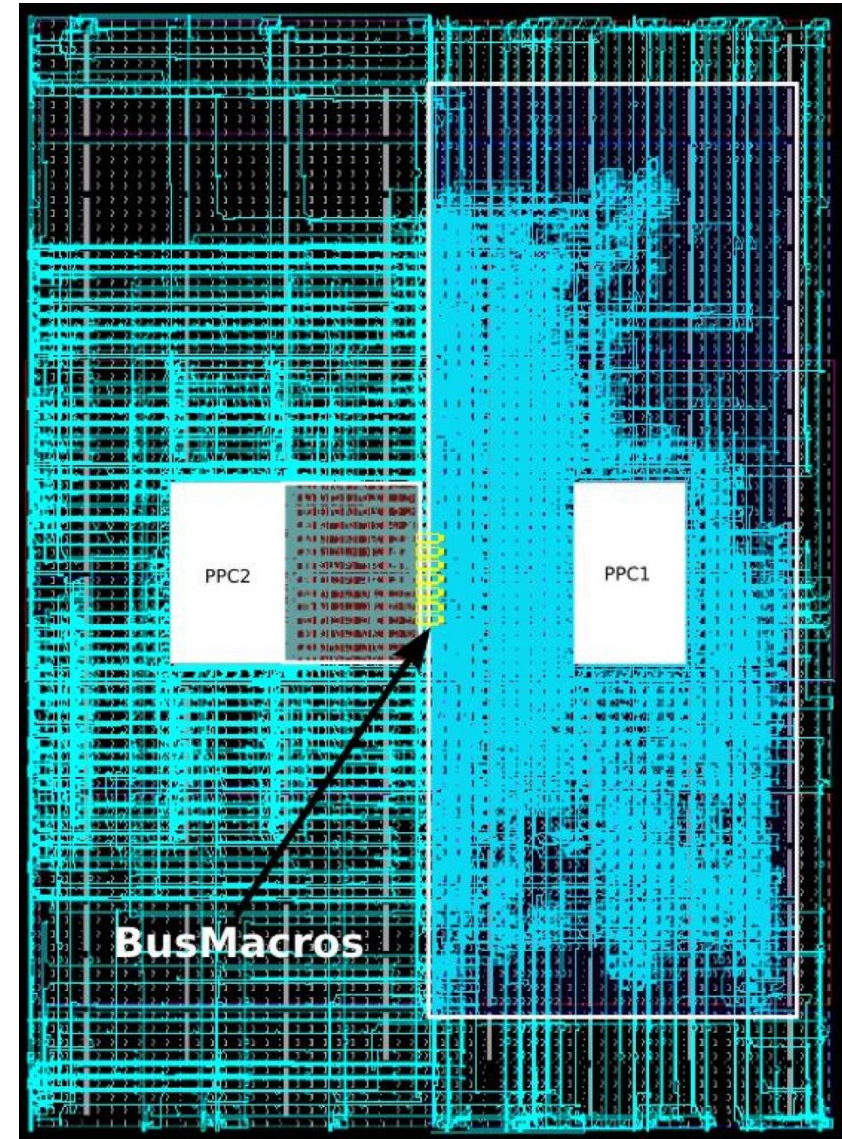
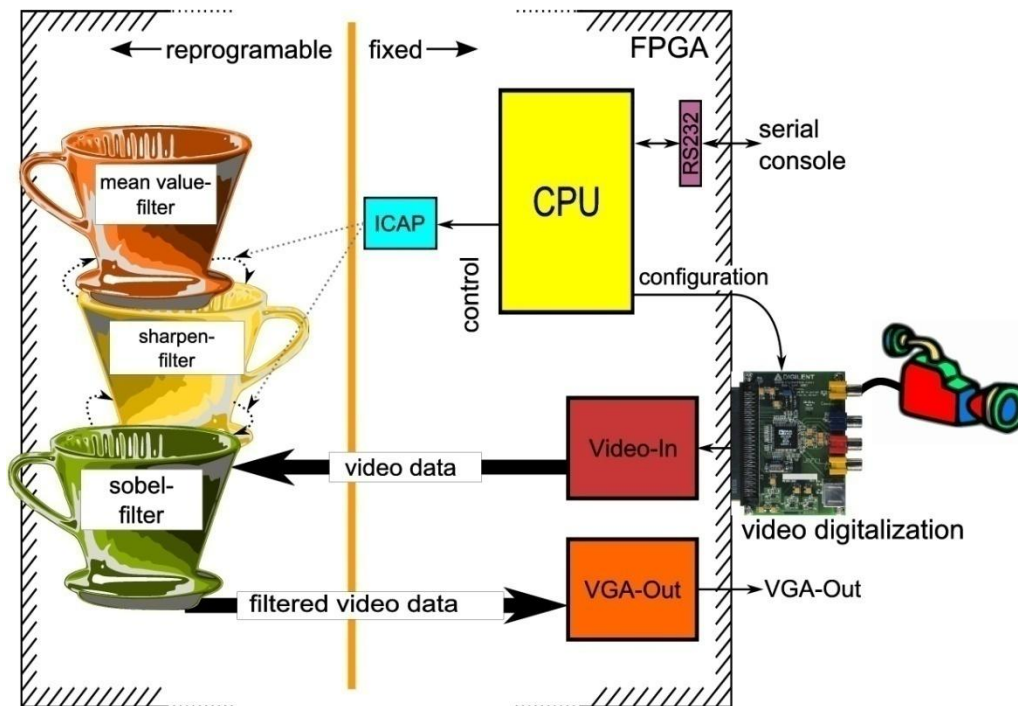


➔ max. 25 kByte Größe

➔ ca. 4 MByte/s (bei 128 Byte/Nachricht)

# Adaptivität der Rechnerressourcen

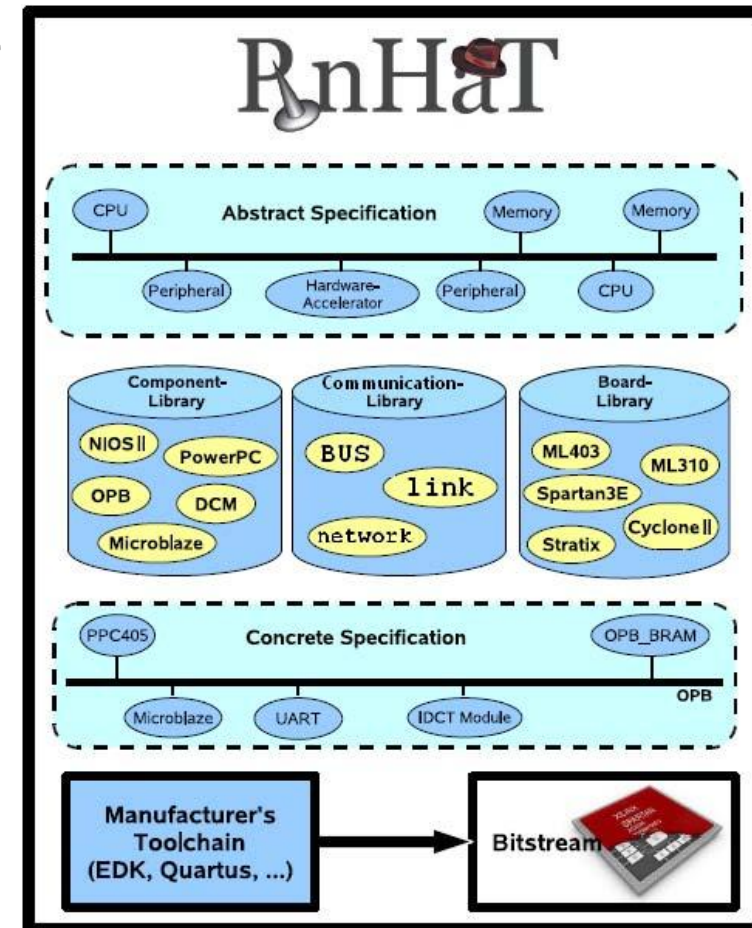
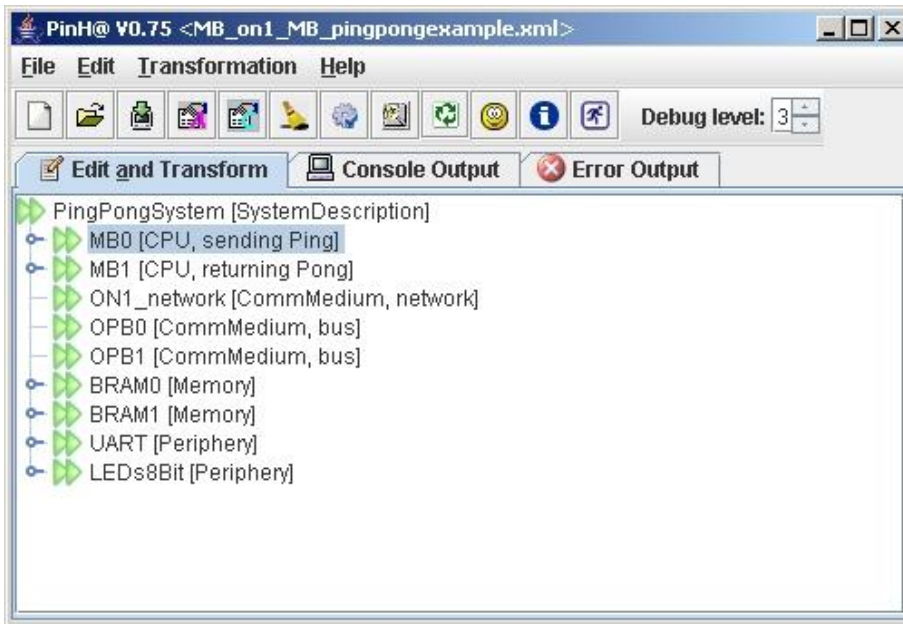
- Vorlage für partielle Rekonfiguration auf FPGAs
- Testsystem: Videofilter



Publikation: D. Murr, F. Mühlbauer, F. Dressler und C. Bobda. *Utilizing Reconfigurable Hardware To Optimize Workflows In Networked Nodes*, IESS 2007

# Werkzeugunterstützung

- **PinHaT** (Plattform-independent Hardware Generation Tool)
  - *Eingabe*: abstrakte Spezifikation des Systems
  - abstrakten Spezifikation wird auf konkrete Komponenten abgebildet
  - *Ausgabe*: Daten, die von der Toolchain des Herstellers weiterverarbeitet werden

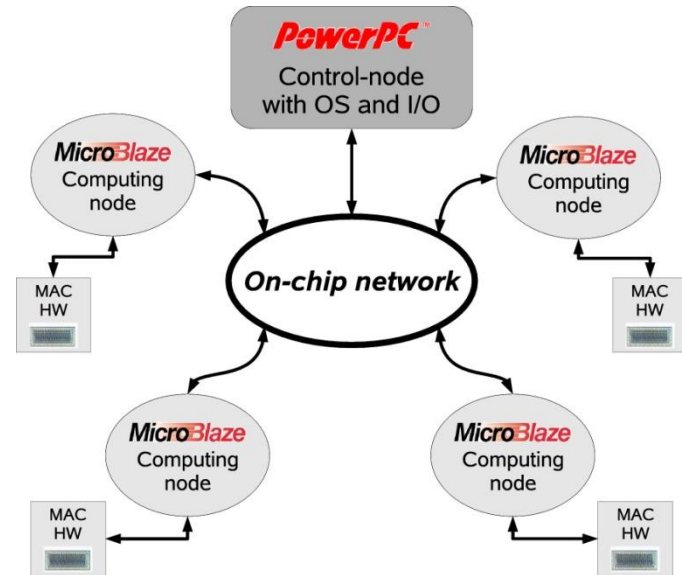


Publikation: C. Bobda, T. Haller und F. Mühlbauer. *Design of Adaptive Multiprocessor on Chip Systems*, SBCCI 2007

# Anwendungen

## ■ Singulärwertzerlegung

- Mit PinHaT realisiert
- Ring-Cluster auf ML310 (XC2VP30) FPGA-Board mit 60MHz
- Multiply-Accumulate (MAC) Hardwarebeschleuniger
- 4 - 32 Spalten pro Prozessor



	1 Prozessor	2 Prozessoren	4 Prozessoren	8 Prozessoren
Flächenbedarf [Slices]	1712 (12%)	3286 (23%)	6391 (47%)	12654 (92%)
16 x 200	13270 $\mu$ s	8071 $\mu$ s	6682 $\mu$ s	-
32 x 200	54584 $\mu$ s	30502 $\mu$ s	20997 $\mu$ s	17378 $\mu$ s
64 x 200	-	117834 $\mu$ s	70797 $\mu$ s	50240 $\mu$ s

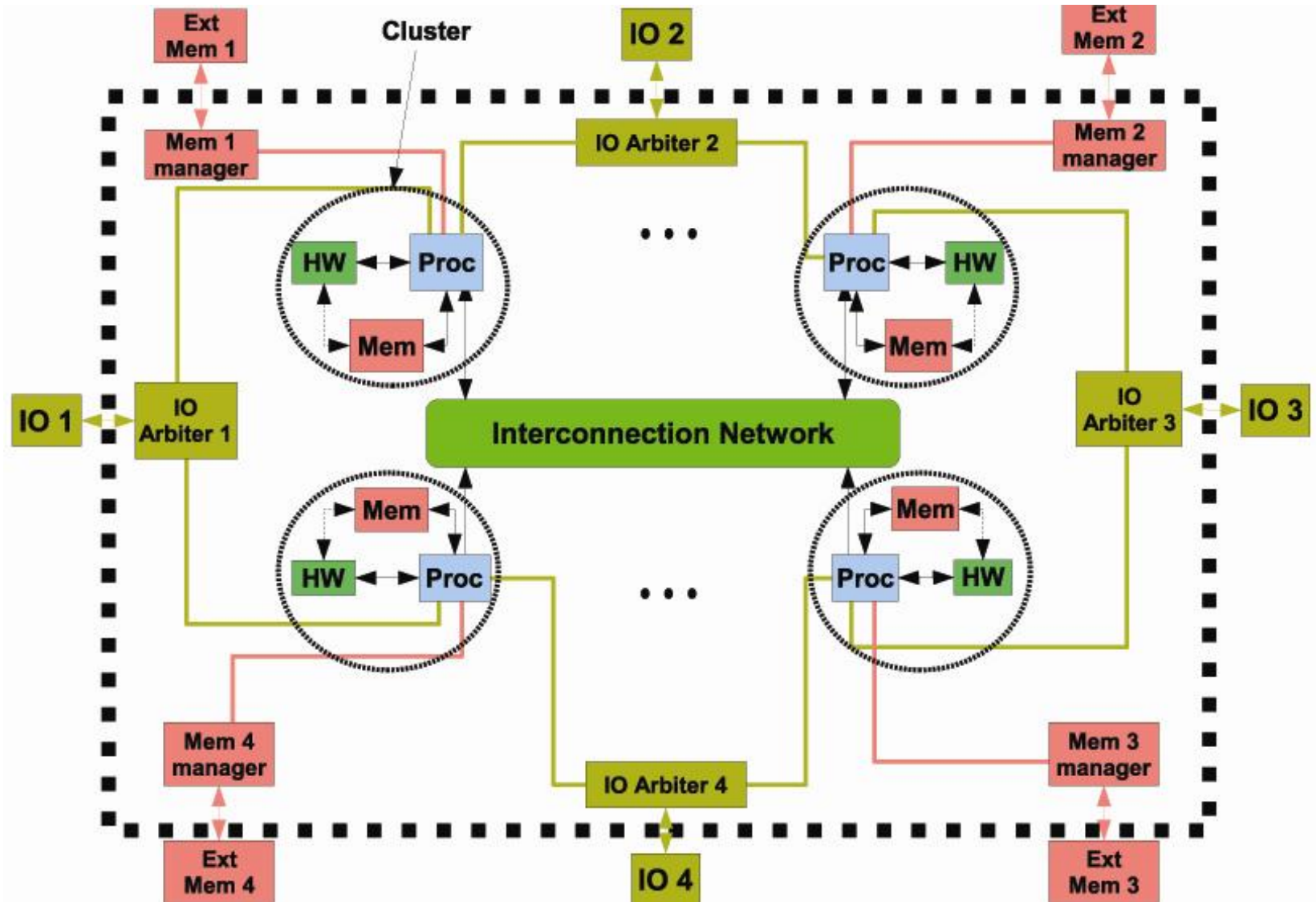
- Parallelisierung von H. 264 unter Verwendung der SocMPI Kommunikationsbibliothek

# Modeling : Scope

- Modeling goal (AMoC):
  - Allow selection of the best configuration (application-specific)
    - ➔ Enable (runtime) optimization
  - Lead to (automatic) generation of HW infrastructure
    - ➔ Cost models of HW components
    - ➔ System model/description (PEs, networks, mapping,...)
    - ➔ SW configuration
    - ➔ Synthesis and device configuration
- Approach:
  - Synthesize an application-specific optimum system from parallel programs (high-level synthesis)

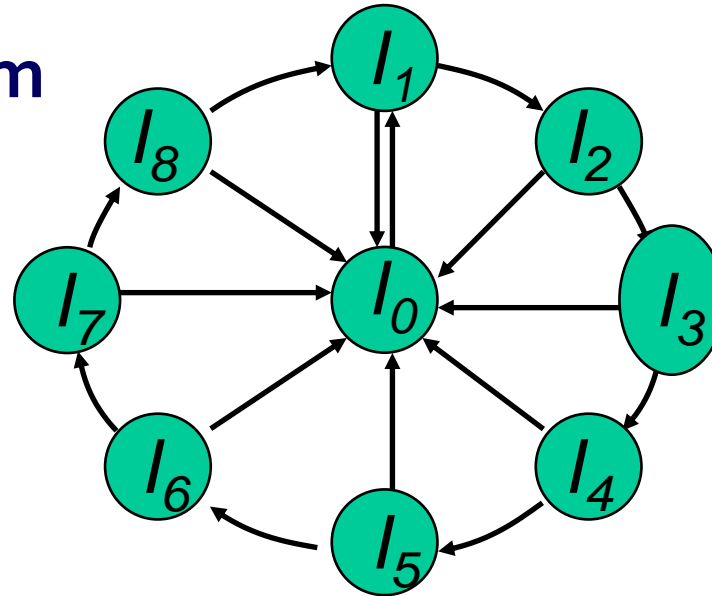
} **PinHaT**

# Target: Application-Specific Optimum Multiprocessor System

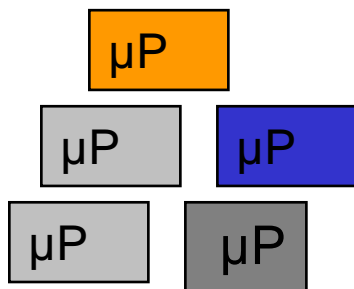


# Simultaneous Task-Mapping and Selection of Resources

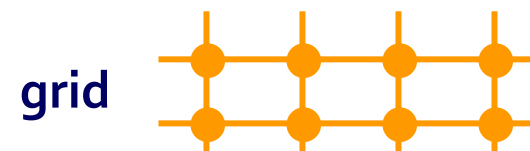
## Parallel Program



## Processors

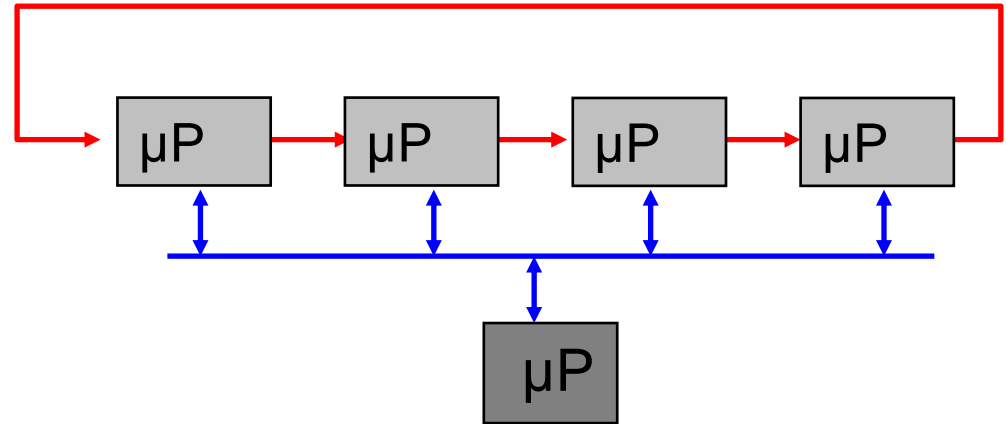
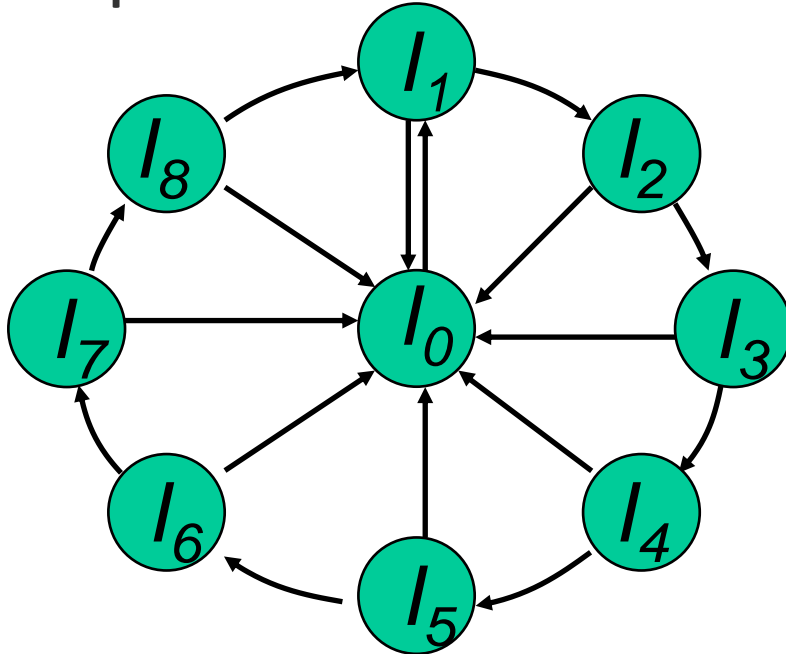


## Comm. networks



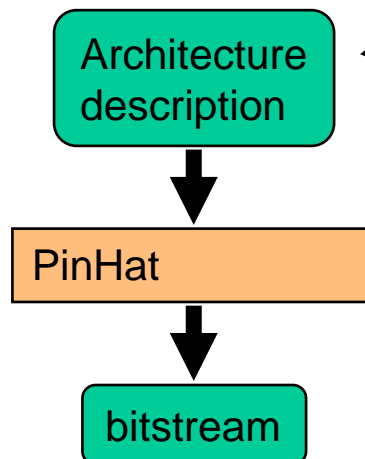
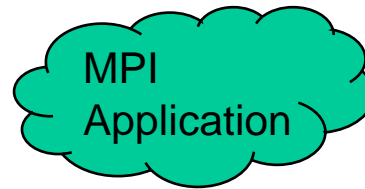
# Motivational Example

- Optimum interconnect depends on data traffic



# Methodologies

- How to synthesize an efficient architecture?



## Specifies

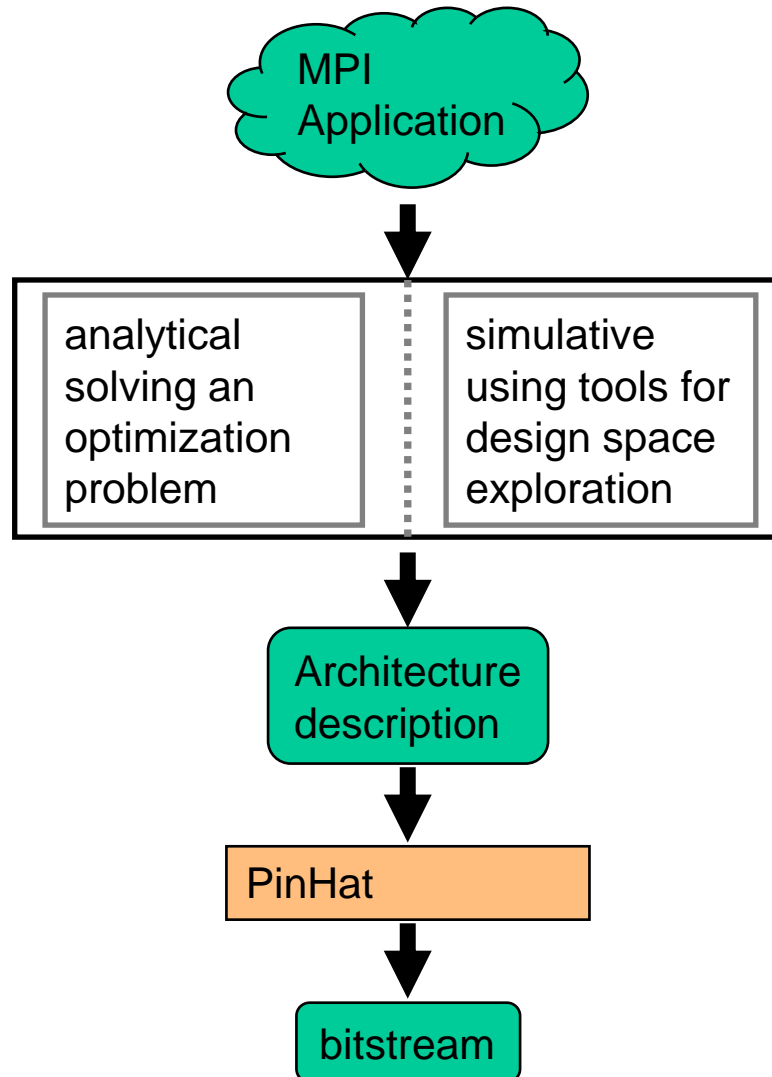
- number of processors
- type of processors
- network
- task mapping
- scheduling

## Specification

- meets area constraints
- minimizes comp. time

# Methodologies

- Two possible approaches (or their hybrid):



+ formal auto-  
mated method

+ solvers exist

- capturing all  
parameters?

- tractable?

+ incorporate  
designer's  
expertise

+ similar  
methodologies/  
tools exist

- manual process

# ILP Model (1/3)

## Map all tasks

$$\sum_{j=0}^m x_{ij} = 1, \forall I_i$$

## Max # tasks per PE

$$\sum_{i=0}^n x_{ij} \cdot s_{ij} \leq s_j, \forall J_j$$

## Max usable FPGA area for PEs

$$\frac{1}{n+1} \sum_{i=0}^n x_{ij} \leq v_j \quad \forall J_j$$

$$v_j \leq \sum_{i=0}^n x_{ij} \quad \forall J_j$$

$$\sum_{j=0}^m v_j \cdot a_j \leq A_{PE}$$

# ILP Model (2/3)

All comm. tasks on different PEs must use a network

$$\alpha_{i_1 i_2 j_1 j_2} \leq \frac{x_{i_1 j_1} + x_{i_2 j_2}}{2}$$

$$\alpha_{i_1 i_2 j_1 j_2} \geq x_{i_1 j_1} + x_{i_2 j_2} - 1$$

$$\sum_{k=0}^K z_{k i_1 i_2} \lambda_{i_1 i_2} \quad \forall I_{i_1}, I_{i_2} \mid I_{i_1} \triangleleft I_{i_2}$$

$$y_k \geq z_{k i_1 i_2} \quad \forall C_k \text{ and } I_{i_1}, I_{i_2} \mid I_{i_1} \triangleleft I_{i_2}$$

# ILP Model (3/3)

## Max network capacity

$$y_k + \sum_{I_{i_1}, I_{i_2} | I_{i_1} \triangleleft I_{i_2}} z_{ki_1 i_2} \leq M_k \quad \forall C_k$$

## Max FPGA area for networks

$$\sum_{k=0}^K A_k y_k \leq A_{net}$$

## Network communication cost

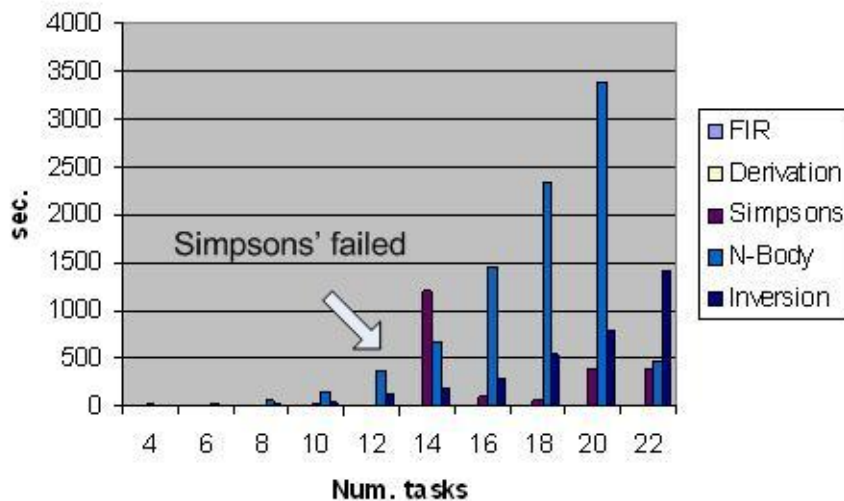
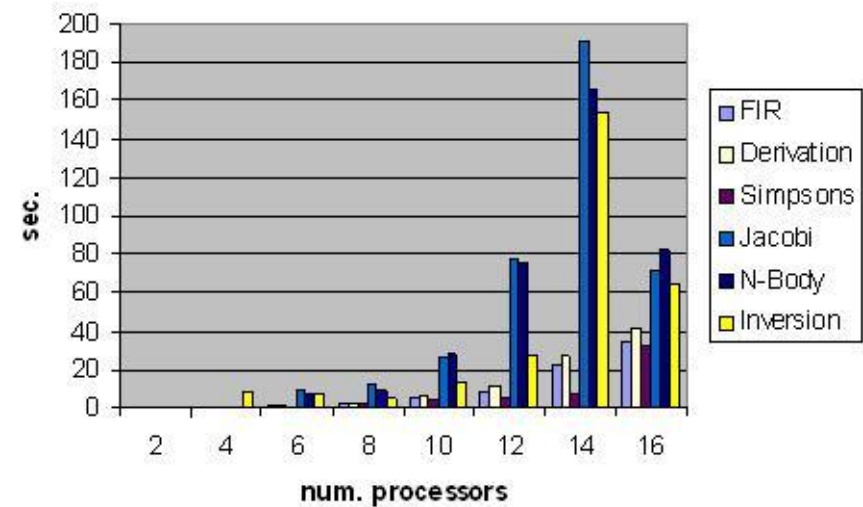
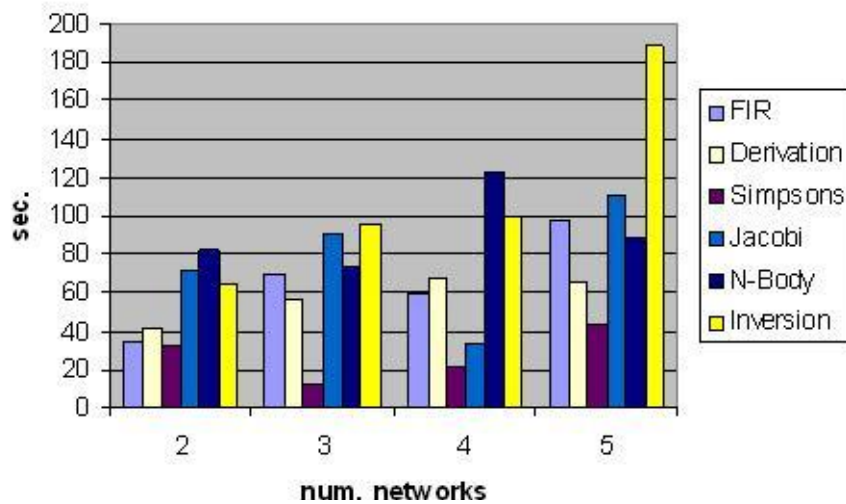
$$T_{net} = \sum_{I_{i_1}, I_{i_2} | I_{i_1} \triangleleft I_{i_2}} \left( \sum_{k=0}^K (L_k \cdot D_{i_1 i_2} + \tau_k \cdot p_k \cdot B_{i_1, i_2}) \cdot z_{ki_1 i_2} \right)$$

## Total max FPGA area

$$A_{PE} + A_{net} \leq A$$

# Feasibility?

- High-level synthesis run time



- Results promising!

# Summary and Outlook

---

- On-Chip-Communication
- PinHaT
- Adaptivity + Applications
- Modelling

## Outlook:

- Extension of SocMPI library
- Scheduling Constraints
- Memory management
- Extension of PinHaT

# Fragen ?