**IMORC: An Infrastructure for Performance Monitoring and Optimization of Reconfigurable Computers**

Tobias Schumacher, Christian Plessl and Marco Platzner  
University of Paderborn

---

**Motivation**

**Model Driven Design**
- Model driven approach for implementation of FPGA accelerators
- Performance estimation before real implementation
- Different cores communicate to each other
- Data flow graphs for performance estimation

**Open Questions**
- How should cores communicate to each other?
- How to verify and optimize the parameters of the model?
- Portability to new architectures?

---

**Introduction to our Modeling Approach**

**Architecture Model**
- Represents all cores in the system
- Cores specified by different parameters:
  - Memory latency / bandwidth
  - Operations available
  - General purpose cores (CPU) and special purpose cores available
- Performance counters help identifying performance bottlenecks
- Easily exchangeable
- Designed to be easily portable between different target architectures

**Execution Model**
- Task graph represents the algorithm
- Blocks of operations connected by communication points
- Graph gets mapped to the architecture model

---

**THE IMORC Infrastructure**

**Request Arbiter**
- One request FIFO per master
- PORTSEL: scheduler for selecting source port
- Can utilize performance counters for scheduling strategy
- Performance counters:
  - Number of times FIFOs are full
  - Same counters on data path

**Work Flow using IMORC**

**Create Task Graph For Target Algorithm**
- Additional cores
- Different memory locations

**Create Initial Architecture Model**
- Map task graph to architecture model
- Graph gets mapped to the architecture
- Synthesis and place & route using vendor tools

**Estimate Performance, Identify Bottlenecks**
- FPGA implementation
- Synthesis and place & route using vendor tools
- Estimate area and performance

**Implement System Using IMORC**
- FPGA implementation
- Synthesis and place & route using vendor tools
- Synthesize and implement system
- Estimate area and performance

**FPGA Implementation**
- Verification and optimization
- Port host bridge to other architectures
- Off-chip interconnects between FPGAs
- Arithmetic units (eg. mul/add)
- Threads (eg. matrix-vector product)
- Sparse matrix solver
- Cube cut
- Finger print and iris code matching

---

**Example: k-th Nearest Neighbor Thinning**

**Algorithm outline**
1. Input data: list of vectors v, goal k
2. Calculate all vector's distances
3. Sort the resulting distance table
4. Remove nearest neighbor
5. If |v| > k, goto (1) else return

---

**Conclusion & Future Work**

**IMORC**
- Easy to use core interconnect infrastructure
- Performance counters help identifying performance bottlenecks
- Seamless integration into our modeling approach
- VHDL, avoiding instantiation of vendor specific blocks
- Easily portable between different target architectures

**Future Work**
- Port host bridge to other architectures
- Off-chip interconnects between FPGAs
- Investigation of suitable IMORC core granularity
- Arithmetic units (eg. mul/add)
- Threads (eg. matrix-vector product)
- Optimizing performance of further accelerators
- Sparse matrix solver
- Cube cut
- Finger print and iris code matching

---

**Contact:**  
Tobias Schumacher  
tobe@uni-paderborn.de  
http://www.upb.de/pc2