**PARO – A Design Tool for the Automatic Generation of Hardware Accelerators**

Frank Hannig, Hritam Dutta, and Jürgen Teich  
Hardware/Software Co-Design, Department of Computer Science  
University of Erlangen-Nuremberg, Germany  
(hannig, dutta, teich)@cs.fau.de  
URL: http://www12.cs.fau.de/research/paro

---

**Overview**

- Tool for the automated hardware synthesis of massively parallel embedded architectures  
- Application domains: Video, audio, image, and other digital signal processing, scientific computing, ...  
- Design entry of dataflow-intensive algorithms in form of a compact and intuitive language  
- Advanced partitioning and scheduling techniques in order to balance trade-offs in cost and performance

---

**PARO Design Flow**

- **High-Level Transformations**  
  - Affine transformations  
  - Dead-code elimination  
  - Constant/variable propagation  
  - Loop perfectionization  
  - Loop unrolling  
  - Strength reduction of operators  
  - Reductions: \( \Sigma, \Pi, \max, \min \)

- **Space-Time Mapping**  
  - Allocation  
  - Resource binding  
  - Scheduling

- **Hardware Synthesis**  
  - Processor element  
  - Controller  
  - Processor array  
  - I/O interface  
  - Functional resources, memory, I/O bandwidth

- **Simulation (PARO)**

- **Hardware Description (VHDL)**

- **Simulation (ModelSim)**

---

**Space-Time Mapping**

- Assignment of iteration points to processors (allocation) and start times (scheduling)

- **Scheduling**  
  - Resource constraints  
  - Module selection  
  - Functional and software pipelining  
  - Run-time dependent conditional and hierarchical partitioned algorithms  
  - Mixed integer programming (MIP)

---

**Hardware Synthesis**

- Generation of platform and language independent RTL description  
- Synthesis steps:  
  1. Processor elements  
  2. Array interconnection structure  
  3. Controller  
- Backend for VHDL

---

**Case Studies**

- **PARO Design Flow**

---

**Conclusions**

- Good scalability of the PARO methodology  
- Processor array approach outperforms conventional loop unrolling (10-61% higher throughput)

- Generation of highly parallelized hardware accelerators