Motivation
Task scheduling on Field Programmable Gate Arrays (FPGAs) is primarily implemented using Partial Reconfiguration (PR). The scheduling approaches work on PR has a high development overhead, is hardly portable and occupies FPGA resources [4].

Inputs
Each PE has at least one task and has a set of PE properties.

Machine models
Machine models build around Processing Elements (PEs). Each PE can execute a set of tasks and has a property set.

Predicates
First-order predicates restrict the set of valid schedules. Predicates can be trivially overridden to program environments.

Schedules
Schedules specify when and where a task is executed.

Task graphs
Task graphs are generated from trace data and annotated with FPGA-specific information.

Simulation
We introduce two reconfiguration-aware polynomial time scheduling algorithms without PR.

Constraint programming
The predicates generated from the machine model can be trivially converted to constraint programming (CP) inputs. A software generates valid schedules for a given set of predicates.

Key Contributions
1. Flexible and arbitrarily accurate machine models for FPGA-based accelerators.
2. Automated derivation of CP programs from machine model.
3. Three heuristic-based polynomial-time scheduling algorithms.
4. Automated recommendations for HLS code.
5. Traces of OpenDwarf executions on FPGAs.

References
[1] Nethercote, Nicholas, Peter J. Stuckey, Ralph Becket, Sebastian Brand, Gregory J. Duck, and Martin所引用的文献

Tools
- OpenDwarf
- CP
- HLS
- CP-CP

Figure:
- Task scheduling on FPGA-Based Accelerators without Partial Reconfiguration
- Machine models are build around Processing Elements (PEs).
- Task graphs are generated from trace data and annotated with FPGA-specific information.
- Simulation: Two types of results from our approach: statistical analysis of valid schedules and inductive proofs for lower bounds, achievable parallelism.
- Constraint programming: The predicates generated from the machine model can be trivially converted to constraint programming (CP) inputs. A software generates valid schedules for a given set of predicates.

Outputs and Results
- Two types of results from our approach: statistical analysis of valid schedules and inductive proofs for lower bounds, achievable parallelism.
- The hardware-agnostic model allows straightforward comparison of distinguished scheduling algorithms and FPGAs [2].

Contact
Pascal Jungblut
pascal.jungblut@mm-team.org
+49 89 2180 8985
LMU Munich
Oettingerstr. 67
81369 Munich, Germany