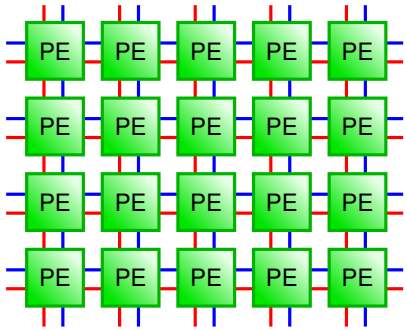
Run-time Requirement Enforcement for Processor Arrays

Tightly coupled processor arrays (TCPAs) are massively parallel loop accelerators that consist of a large grid of processors. Our compiler maps loop programs to TCPAs *symbolically*—the number of processors to allocate must only be known at run-time. That way, application can dynamically adapt the allocation size to its computational needs.



In many cases, loops are not required to be executed as fast as possible but instead to satisfy certain run-time bounds. This so-called *predictability* is crucial especially in real-time systems (e. g., automotives, avionics, robotics).

Given a runtime bound in seconds and a symbolic formula for calculating a loop's latency in cycles, how many processors need to be allocated and what frequency is required to satisfy the bound? Is it better to increase the frequency and allocate fewer processors, or vice versa? If there are multiple possible allocations, which one is the most sensible?

These questions and many more are the topic of this thesis. In addition, a code template needs to be created that implements the described *run-time requirement enforcement*.

Prerequisites: Experience in in C
Type of work: Theory (60%), Design (30%), Implementation (10%)
Supervisors: Michael Witterauf (witterauf@cs.fau.de), Marcel Brand (marcel.brand@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT