Mapping of Applications to MPSoCs

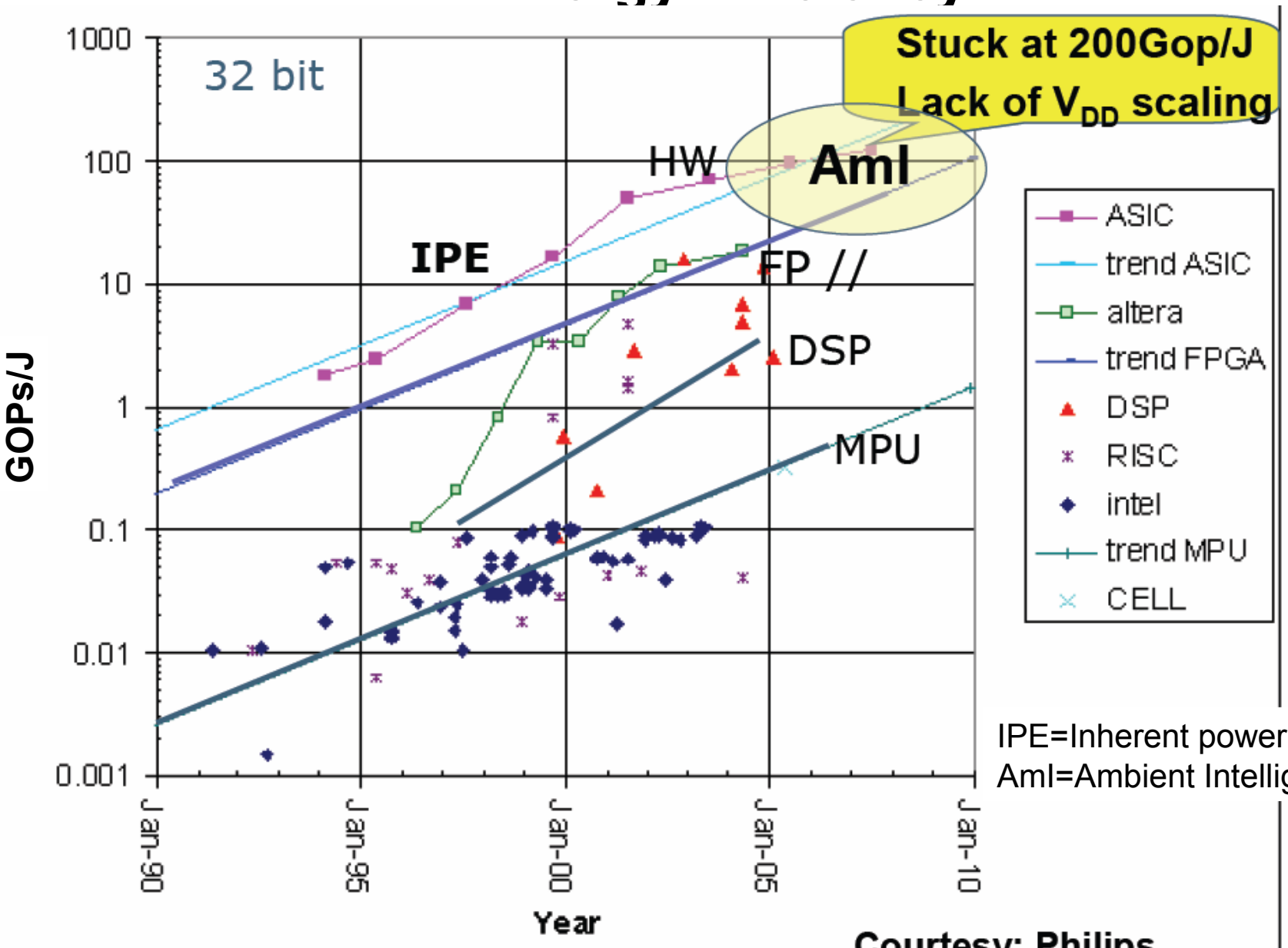
- Survey -

Peter Marwedel
TU Dortmund/Informatik 12
& ICD/ES
Germany

Outline

- Architectures of MPSoCs
- heterogeneous
- homogeneous
- Problem description
- Related work
- Mapping Techniques
- from task graphs/from sequential programs
- with design space exploration (DSE)/no DSE
- Future work
- Summary

Energy Efficiency

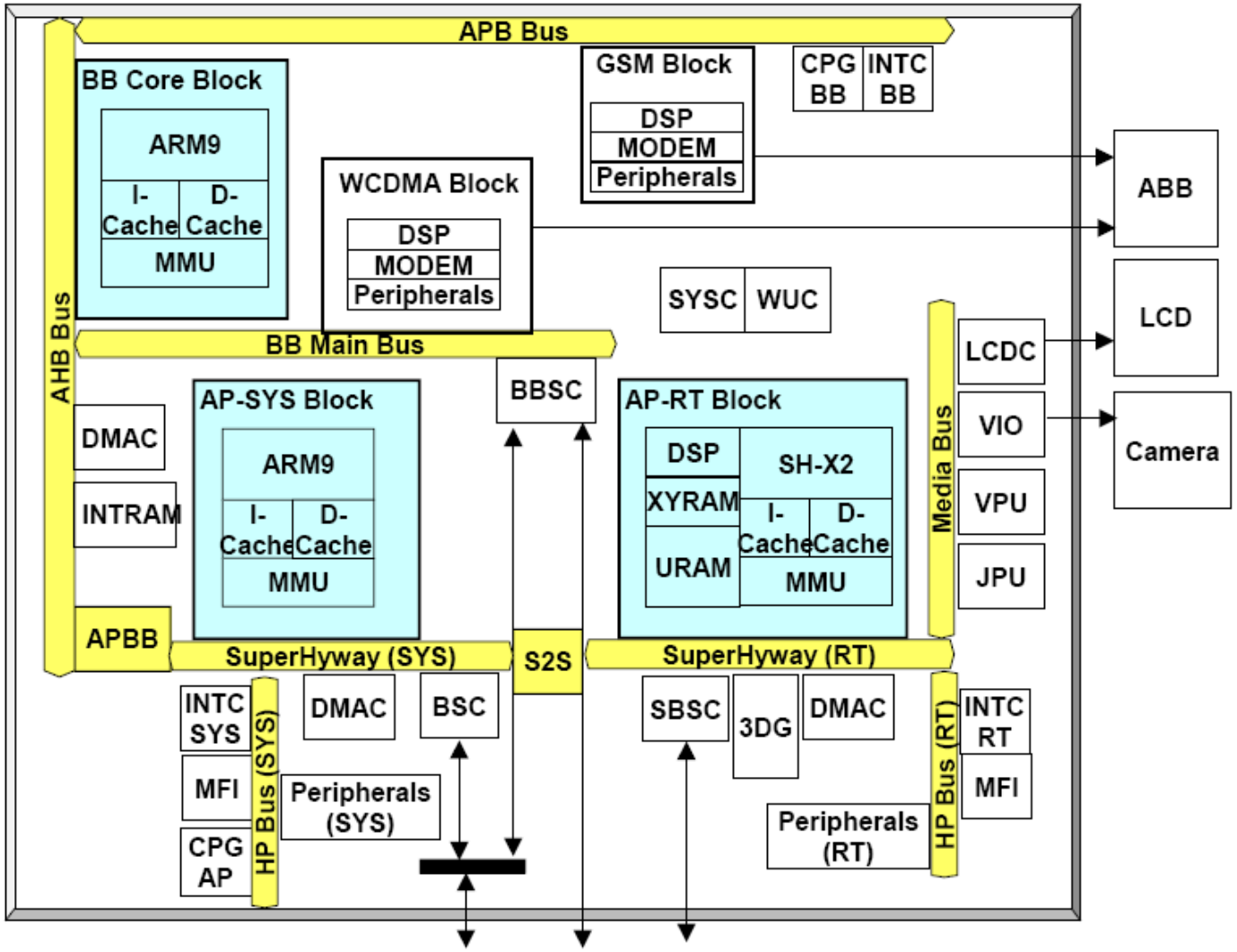


IPE=Inherent power efficiency
AmI=Ambient Intelligence

Courtesy: Philips

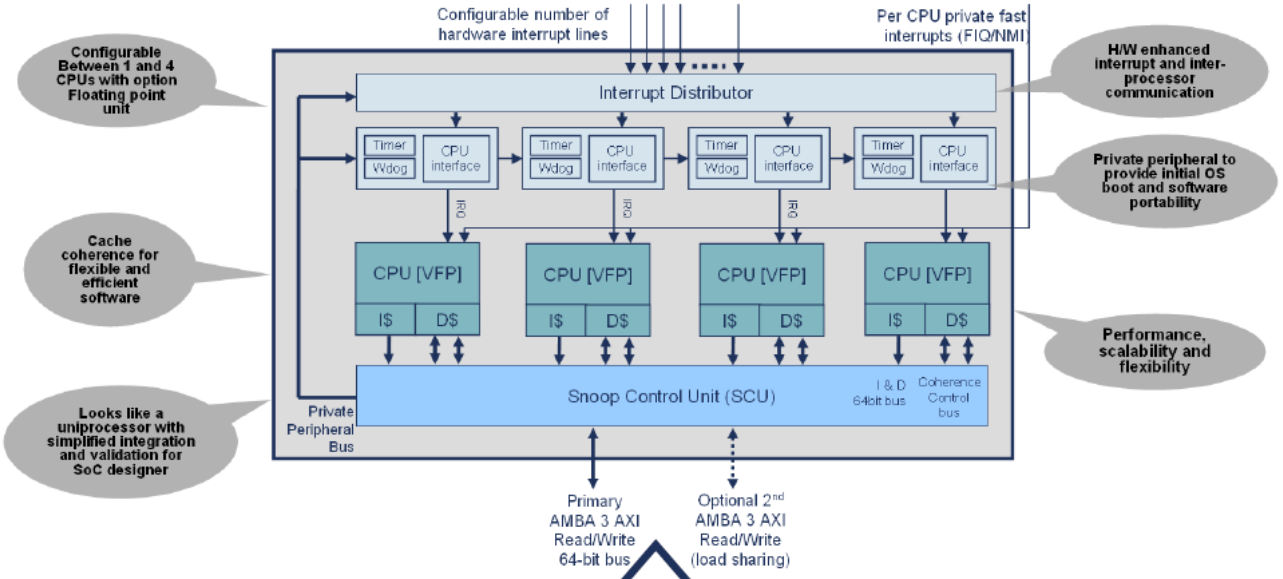
Heterogeneous Architectures

G1 Module Diagram

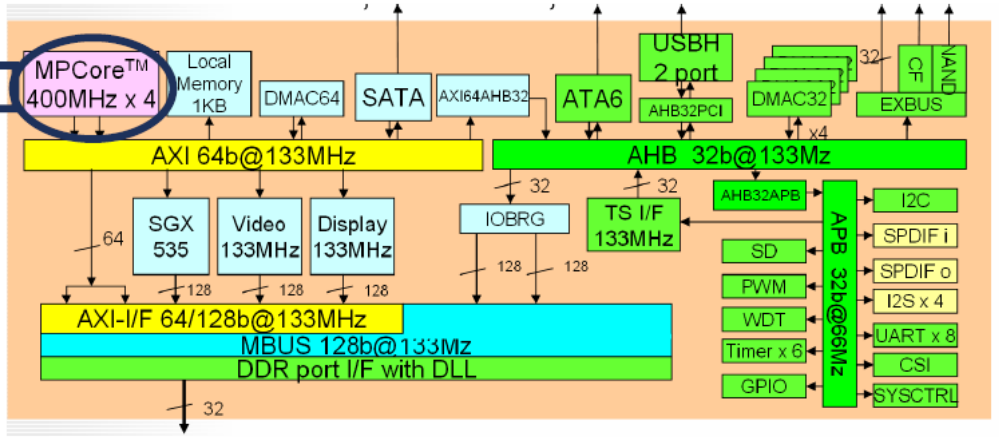


http://www.mpsoc-forum.org/2007/slides/Hattori.pdf

Homogeneous Architectures



- Utilizing the ARM11 MPCore multicore processor to extend the host processor node (application software domain) to execute across multiple homogeneous cores



© J. Goodacre, ARM, 2008

NaviEngine®の特徴 NEC

Outline

- Architectures of MPSoCs
- heterogeneous
- homogeneous
- Problem description
- Related work
- Mapping Techniques
- from task graphs/from sequential programs
- with design space exploration (DSE)/no DSE
- Future work
- Summary

Problem Description

Tools urgently needed!

Not many contributions yet!

Given

- A set of applications
- Scenarios on how these applications will be used
- A set of candidate architectures comprising
  - (Possibly heterogeneous) processors
  - (Possibly heterogeneous) communication architectures
  - Possible scheduling policies

Find

- A mapping of applications to processors
- Appropriate scheduling techniques (if not fixed)
- A target architecture (if DSE is included)

Objectives

- Keeping deadlines and/or maximizing performance
- Minimizing cost, energy consumption

Related Work

- Mapping to EXUs in automotive design
- Scheduling theory:
  Provides insight for the mapping *task* → *start times*
- Hardware/software partitioning:
  Can be applied if it supports multiple processors
- High performance computing (HPC)
  Automatic parallelization, but only for
  - single applications,
  - fixed architectures,
  - no support for scheduling,
  - memory and communication model usually different
- High-level synthesis
  Provides useful terms like scheduling, allocation, assignment
- Optimization theory

Focus of the ArtistDesign Network

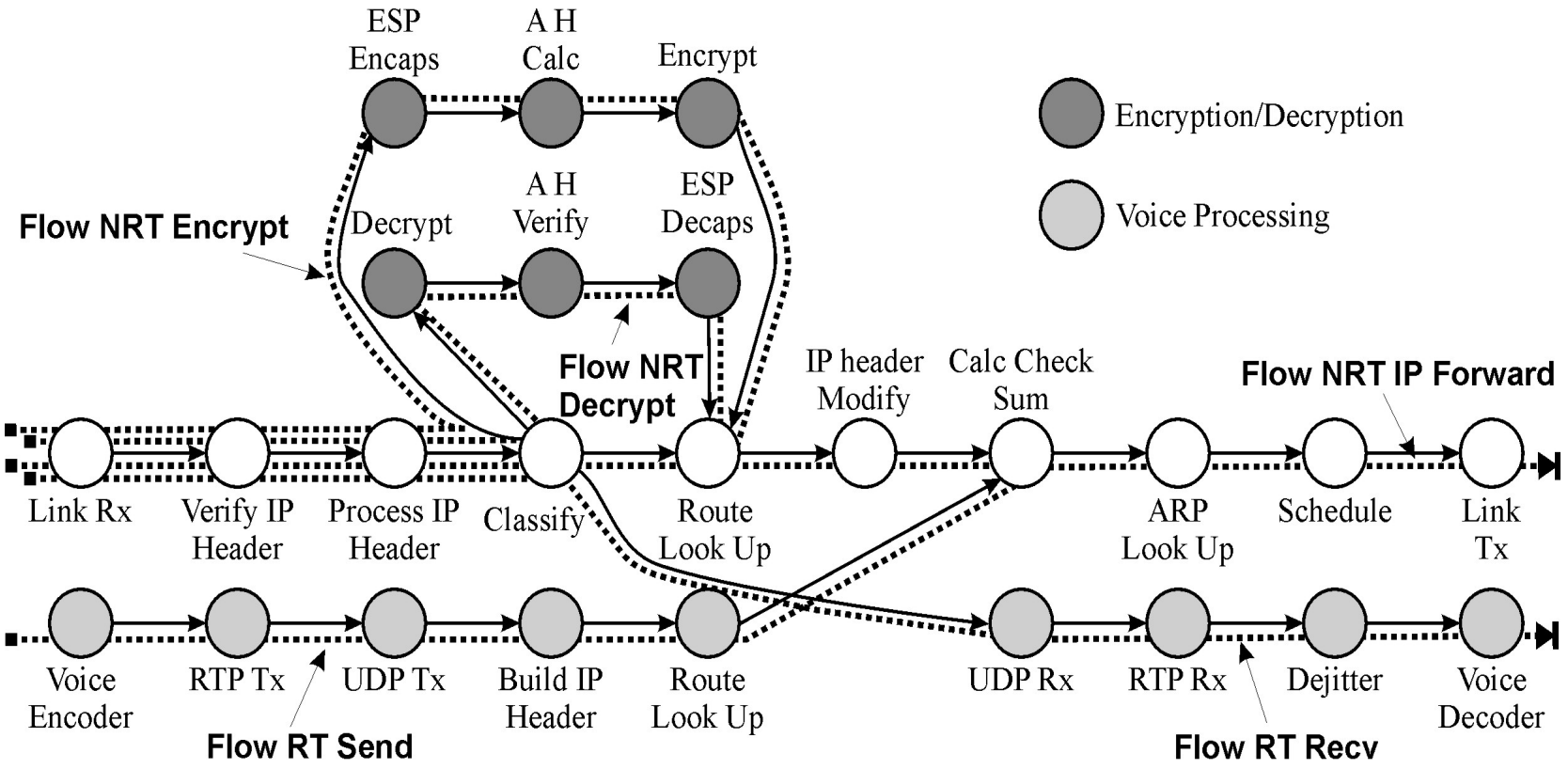
- 1st Workshop on Mapping Applications To MPSoCs, Rheinfels castle, June, 2008
  http://www.artist-embedded.org/artist/-map2mpsoc-2008-.html
- 2nd Workshop on Mapping Applications To MPSoCs, Rheinfels castle, June, 2009
  http://www.artist-embedded.org/artist/-map2mpsoc-2009-.html

A Simple Classification

Architecture fixed/ Auto-parallelizing	Fixed Architecture	Architecture to be designed
Starting from given task graph	Map to CELL, Hopes, Qiang XU (HK) Simunic (UCSD)	COOL codesign tool; EXPO/SPEA2 SystemCodesigner
Auto-parallelizing	Mnemee (Dortmund) Franke (Edinburgh) MAPS	Daedalus

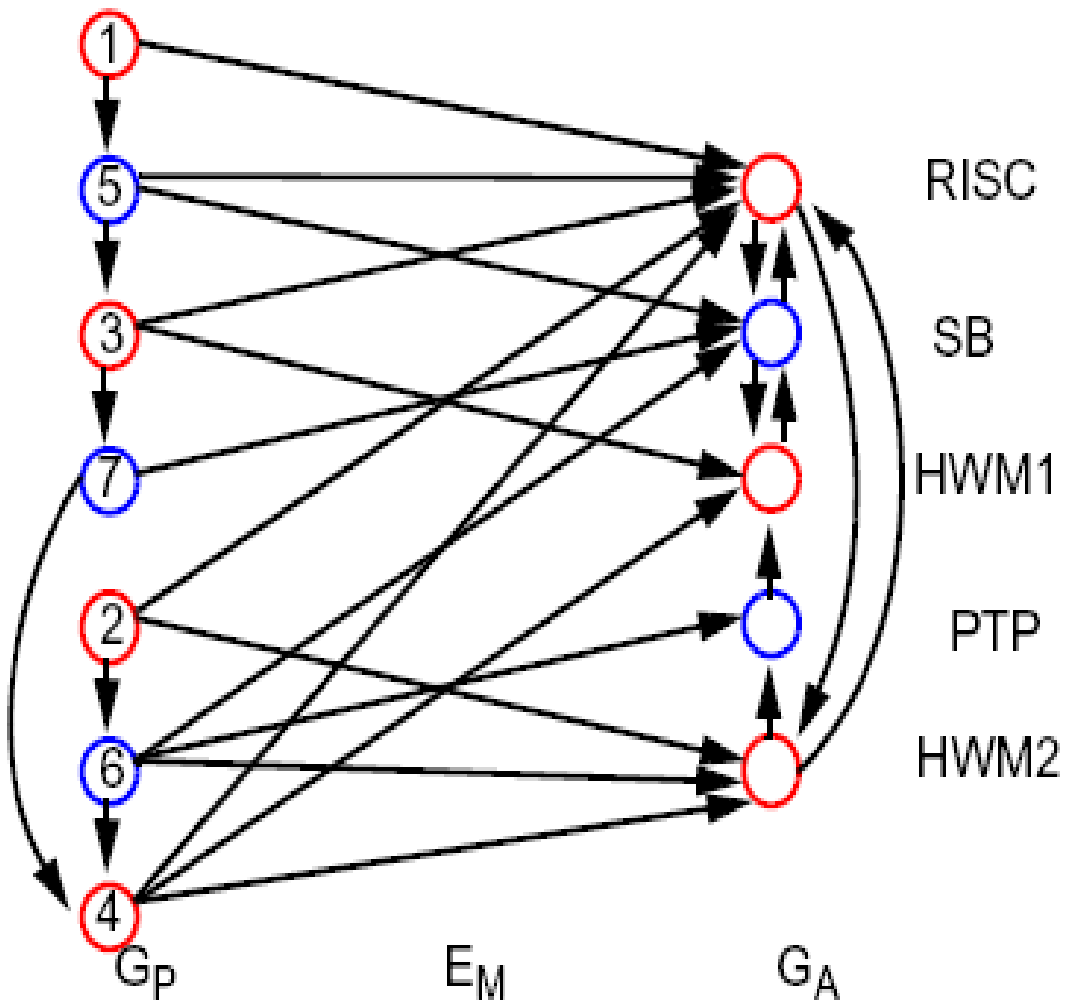
Starting from task graphs (ETH Zürich)

Example of a simple stream processing task structure:

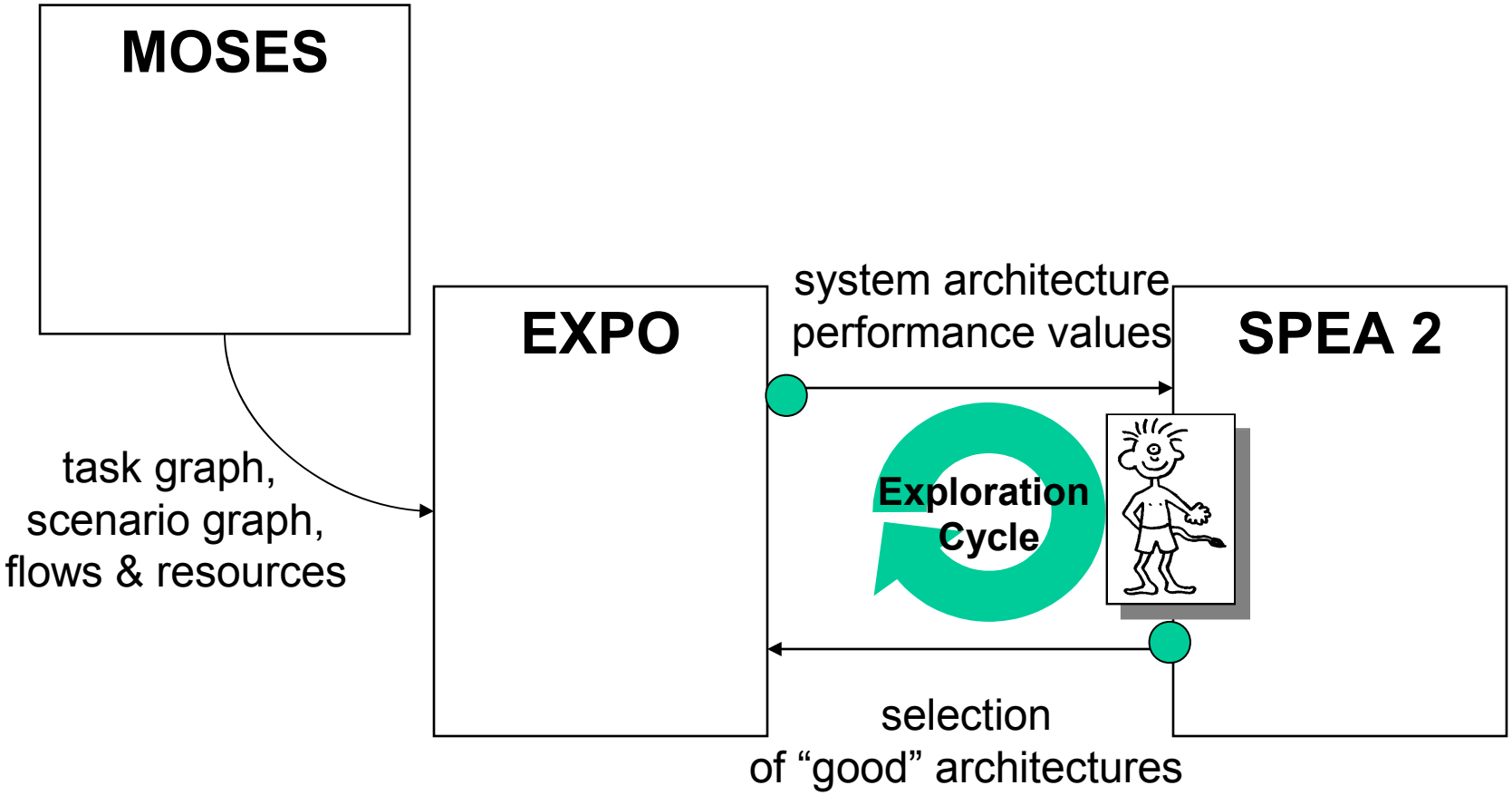


Basic Model: Specification Graph

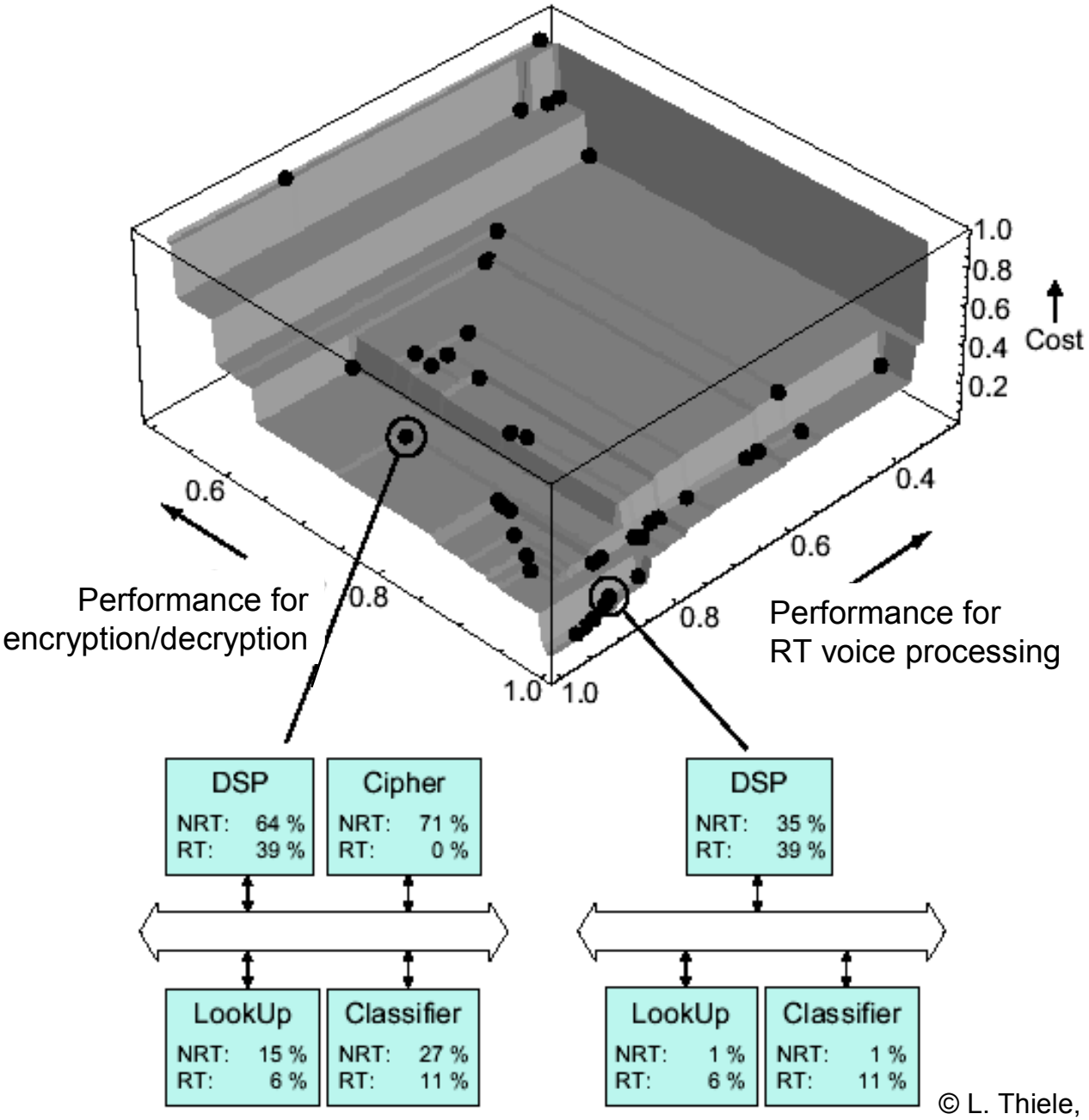
Definition: A specification graph is a graph $G_S=(V_S,E_S)$ consisting of a problem graph G_P, an architecture graph G_A, and edges E_M. In particular, $V_S=V_P \cup V_A$, $E_S=E_P \cup E_A \cup E_M$



EXPO/SPEA2 – Tool architecture



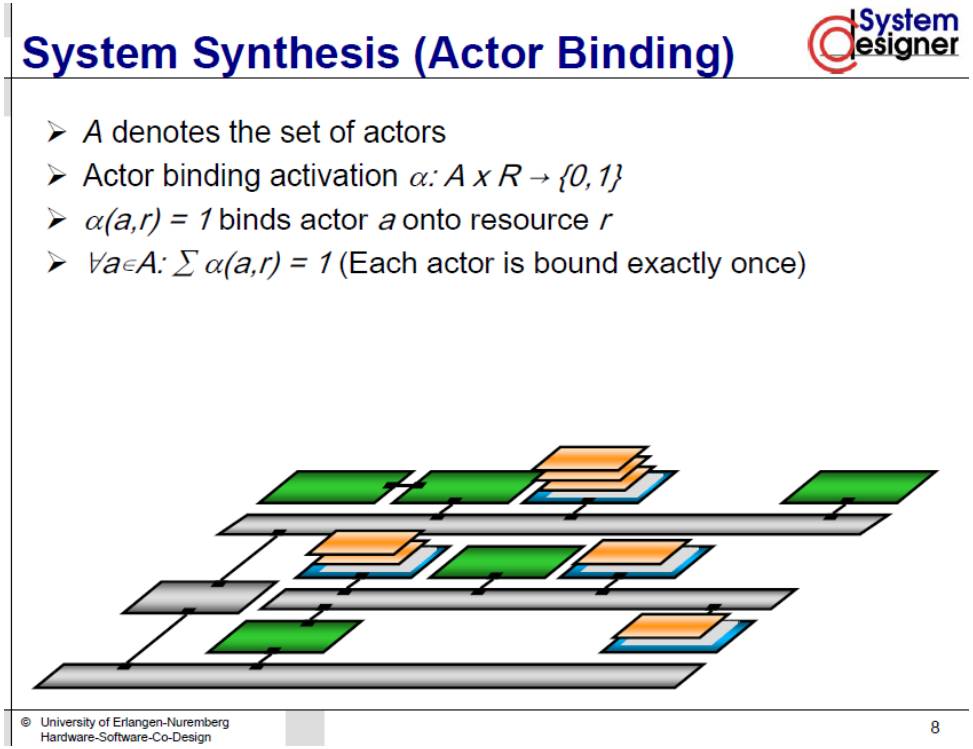
Results



© L. Thiele, ETHZ

Design Space Exploration with SystemCoDesigner
(Teich, Erlangen)

- System Synthesis comprises:
  - Resource allocation
  - Actor binding
  - Channel mapping
  - Transaction modeling
- Idea:
  - Formulate synthesis problem as 0-1 ILP
  - Use Pseudo-Boolean (PB) solver to find feasible solution
  - Use multi-objective Evolutionary algorithm (MOEA) to optimize Decision Strategy of the PB solver



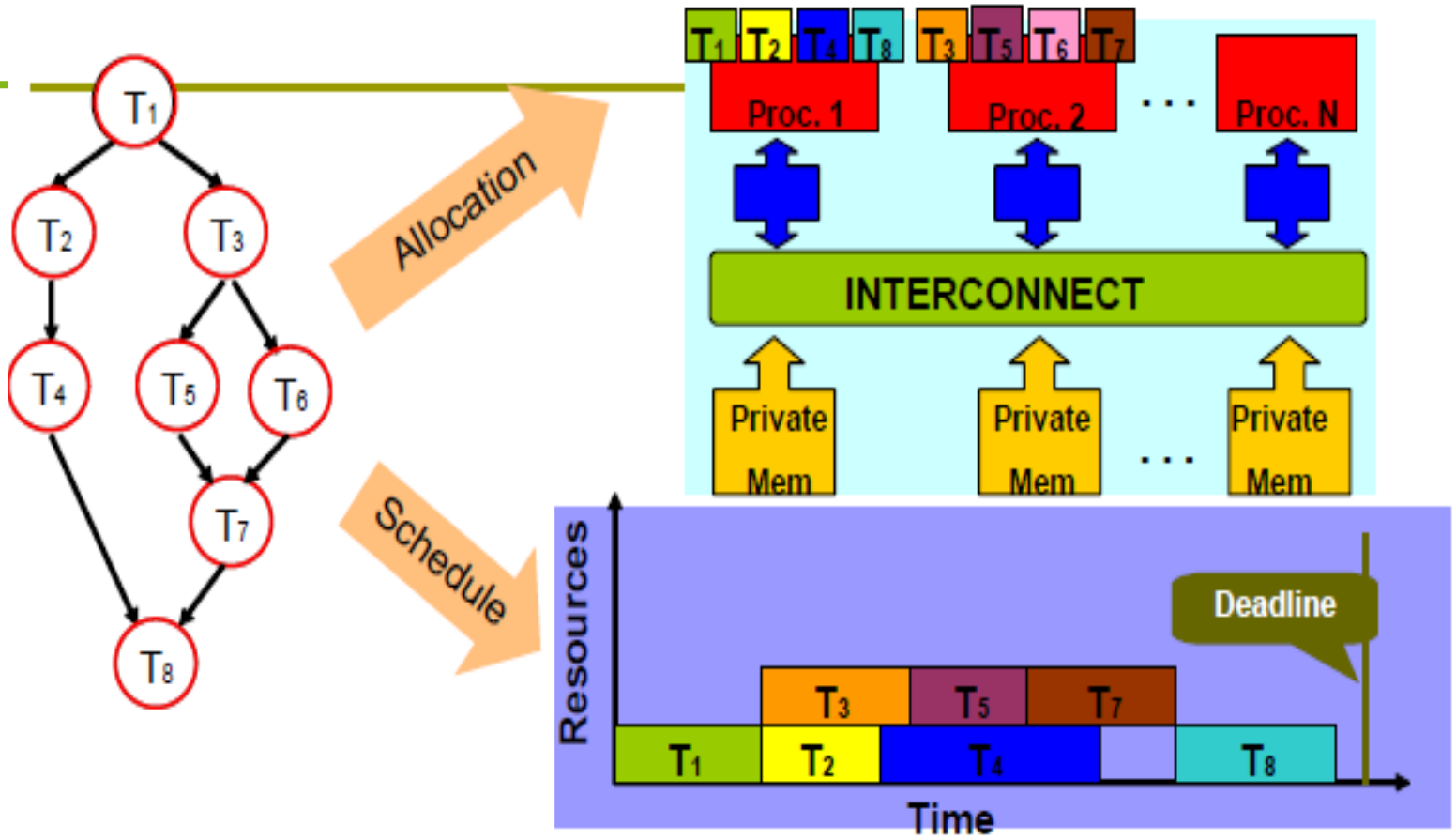
Additional Contributions

- Compaan (B. Kienhuis et al.)
- Sympa/S (R. Ernst et al.)
- Milan (J. David et al.)
- Charmed (S. Bhattacharyya et al.)
- Metropolis (A. Sangiovanni-Vincentelli et al.)
- …

A Simple Classification

Architecture fixed/ Auto-parallelizing	Fixed Architecture	Architecture to be designed
Starting from given task graph	Map to CELL, Hopes Qiang XU (HK) Simunic (UCSD)	COOL codesign tool; EXPO/SPEA2 SystemCodesigner
Auto-parallelizing	Mnemee (Dortmund) Franke (Edinburgh) MAPS	Daedalus

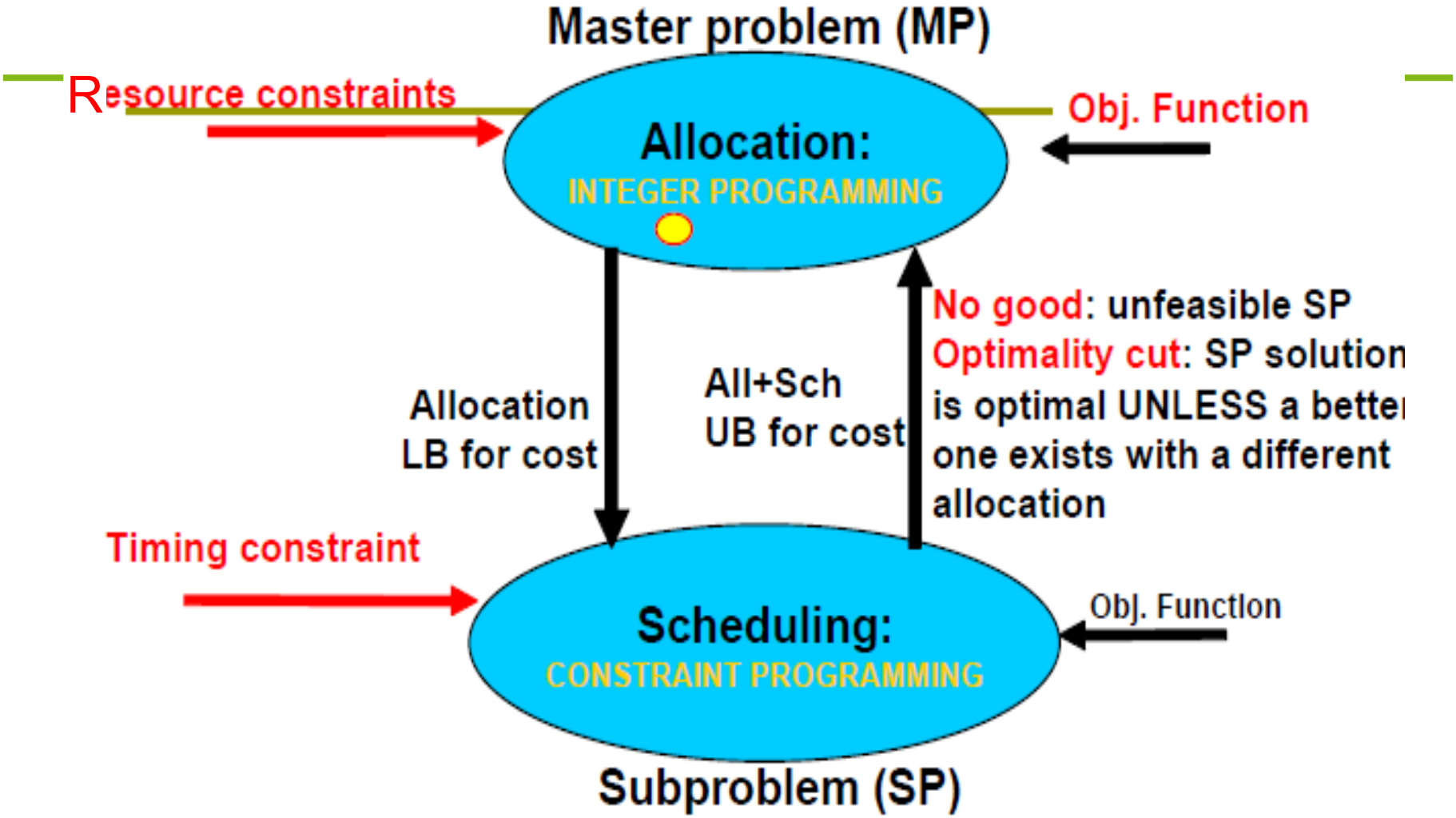
A fixed architecture approach: Map → CELL



- The problem of allocating and scheduling task graphs on processors in a distributed real-time system is NP-hard.

Martino Ruggiero, Luca Benini: Mapping task graphs to the CELL BE processor, *1st Workshop on Mapping of Applications to MPSoCs, Rheinfels Castle*, 2008

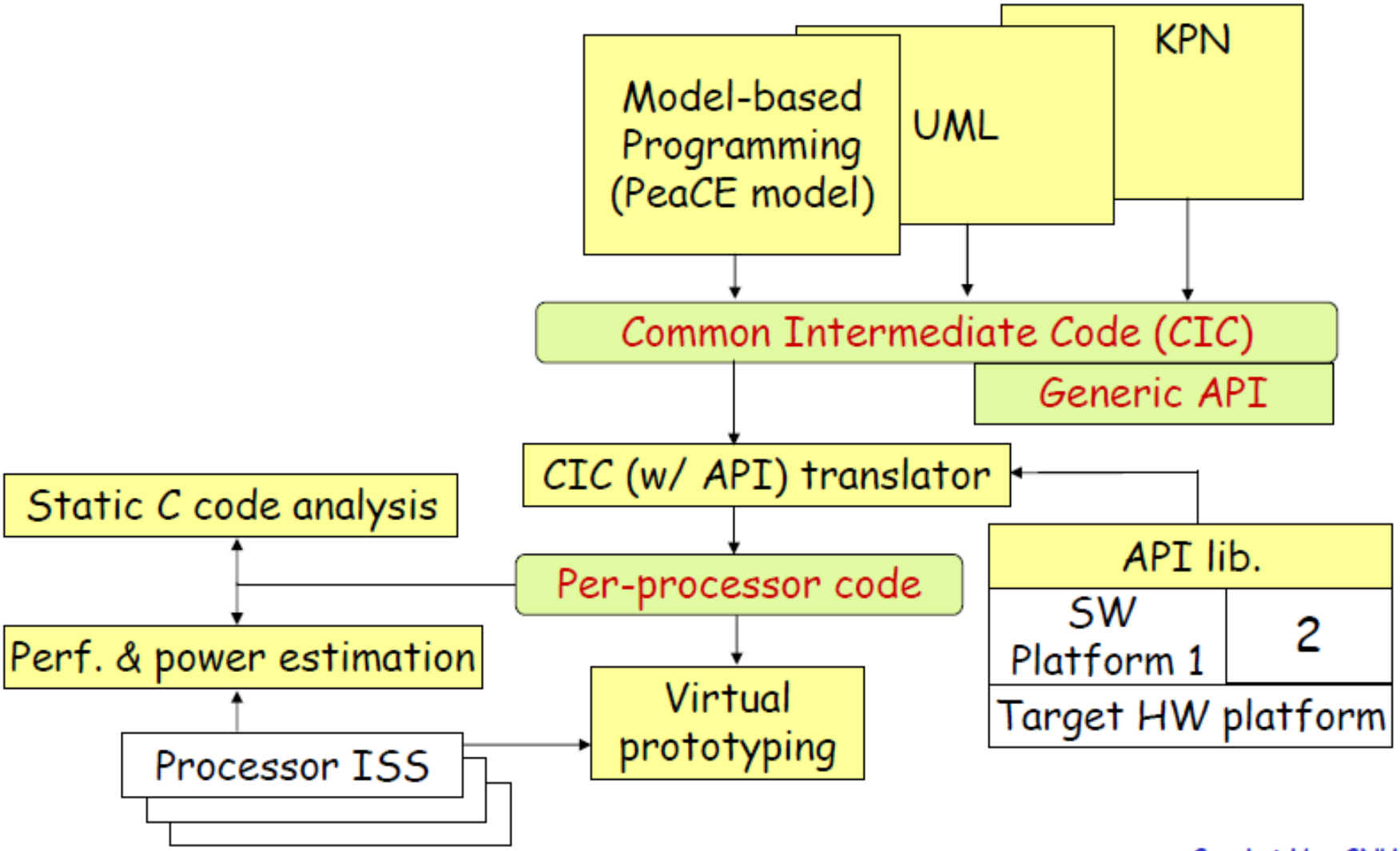
Partitioning into Allocation and Scheduling



Iterations stop when MP becomes unfeasible!

HOPES Proposal

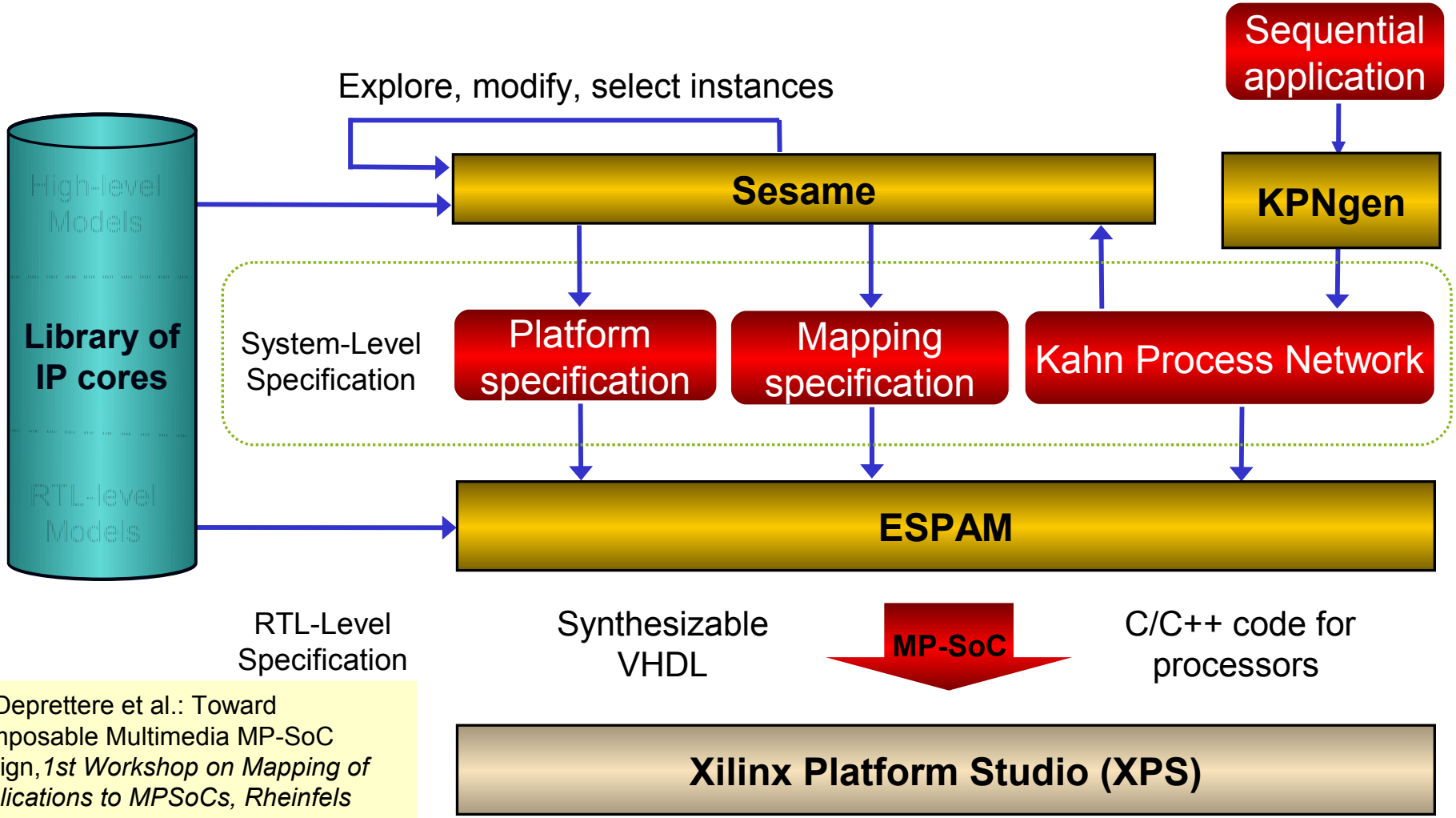
HOPES



A Simple Classification

Architecture fixed/ Auto-parallelizing	Fixed Architecture	Architecture to be designed
Starting from given task graph	Map to CELL, Hopes, Qiang XU (HK) Simunic (UCSD)	COOL codesign tool; EXPO/SPEA2 SystemCodesigner
Auto-parallelizing	Mnemee (Dortmund) Franke (Edinburgh) MAPS	Daedalus

Daedalus Design-flow

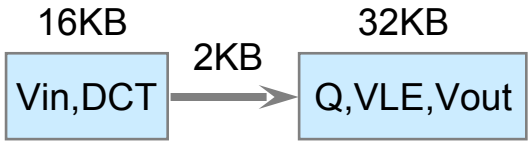


Ed Deprettere et al.: Toward Composable Multimedia MP-SoC Design, *1st Workshop on Mapping of Applications to MPSoCs, Rheinfels Castle*, 2008

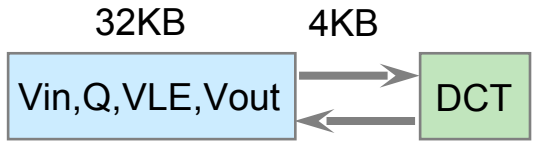
© E. Deprettere, U. Leiden

JPEG/JPEG 2000 case study

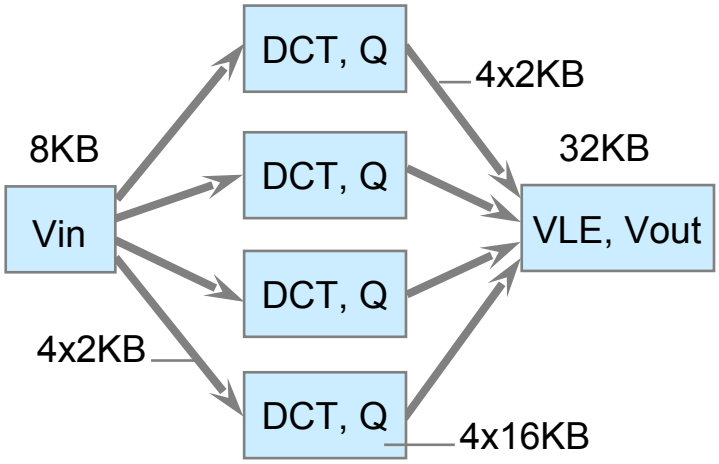
Example architecture instances for a single-tile JPEG encoder:



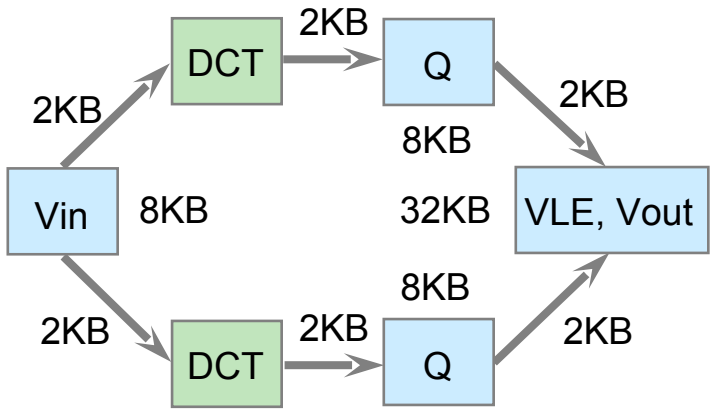
2 MicroBlaze processors (50KB)



1 MicroBlaze, 1HW DCT (36KB)

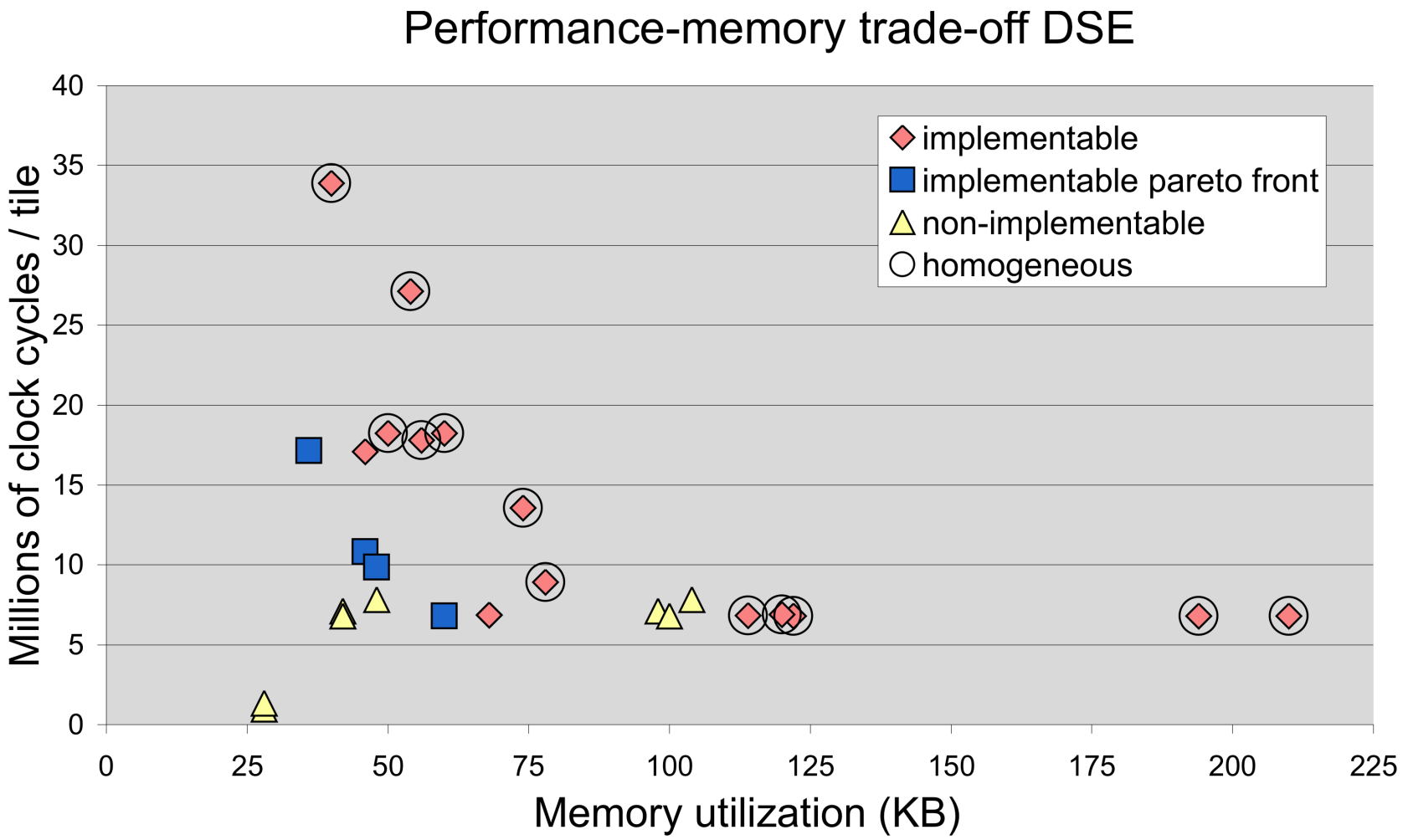


6 MicroBlaze processors (120KB)



4 MicroBlaze, 2HW DCT (68KB)

Sesame DSE results:
Single JPEG encoder DSE



A Simple Classification

Architecture fixed/ Auto-parallelizing	Fixed Architecture	Architecture to be designed
Starting from given task graph	Map to CELL, Hopes, Qiang XU (HK) Simunic (UCSD)	COOL codesign tool; EXPO/SPEA2 SystemCodesigner
Auto-parallelizing	Mnemee (Dortmund) Franke (Edinburgh) MAPS	Daedalus

Auto-Parallelizing Compilers

Discipline "High Performance Computing":

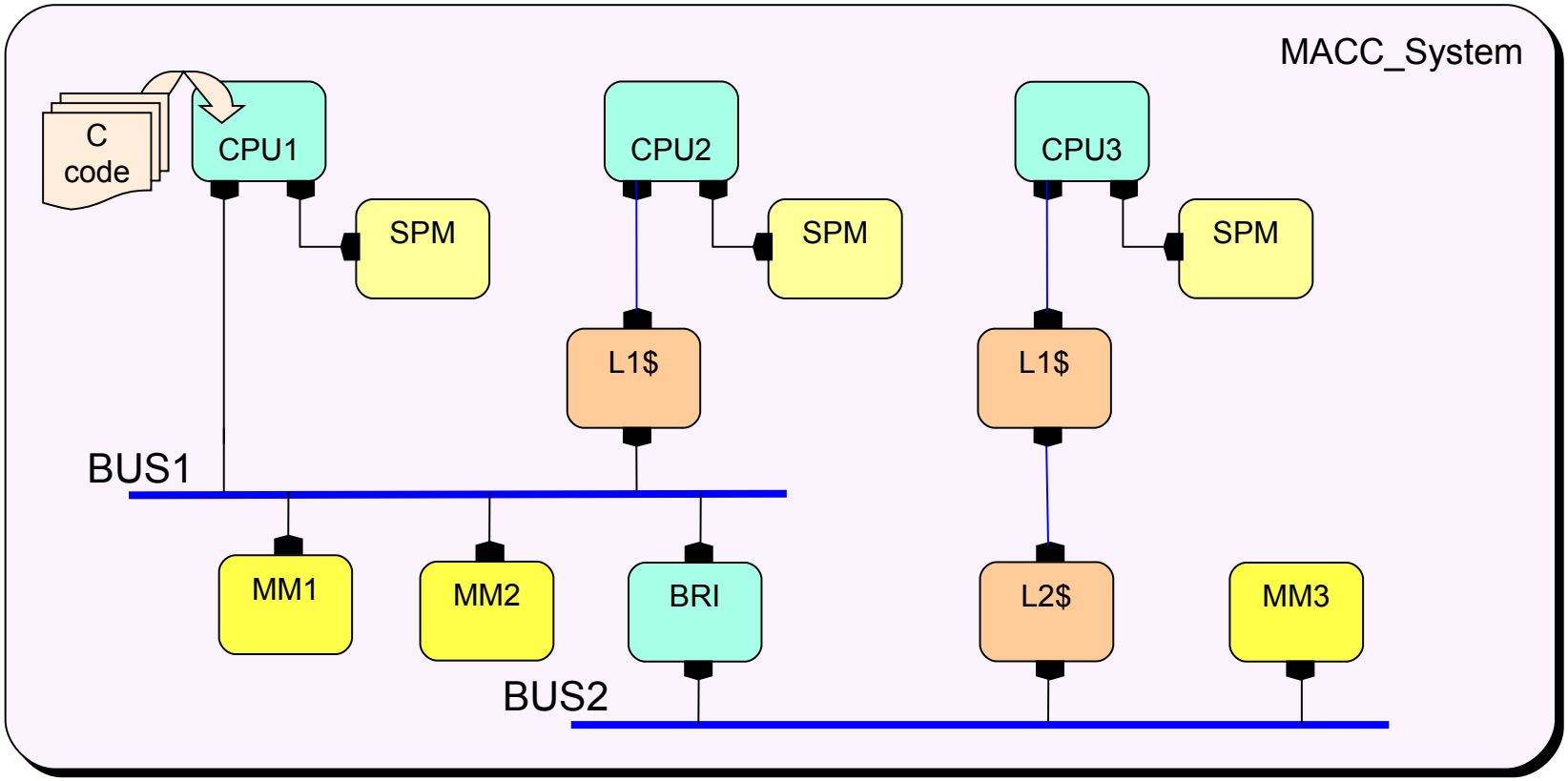
- Research on vectorizing compilers for more than 25 years.
- Traditionally: Fortran compilers.
- Such vectorizing compilers usually inappropriate for Multi-DSPs, since assumptions on memory model unrealistic:
  - Communication between processors via *shared memory*
  - Memory has only *one single* common *address space*

☞ ***De Facto no auto-parallelizing compiler for Multi-DSPs!***

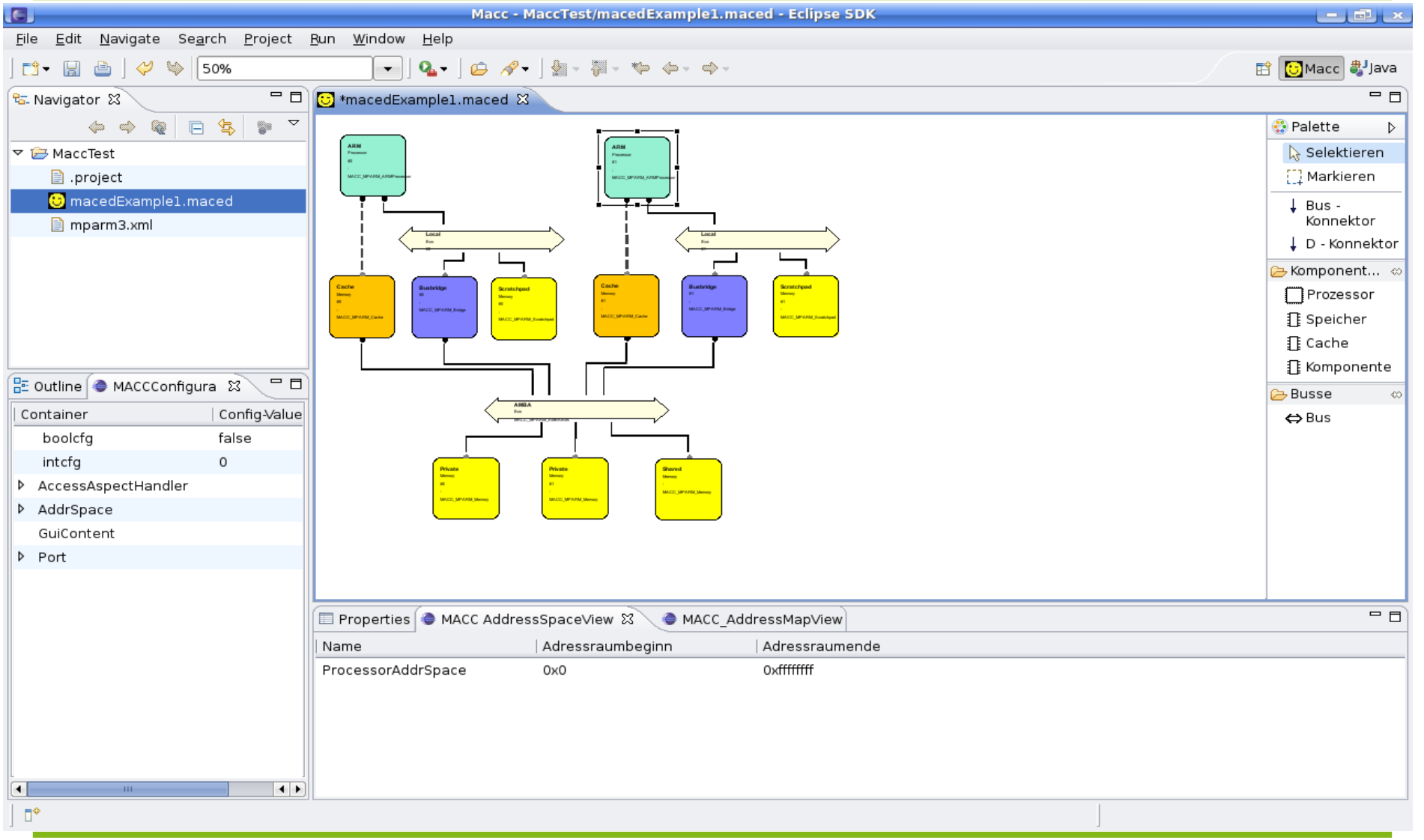
☞ Work of Franke, O'Boyle (Edinburgh)

Introduction of Memory Architecture-Aware Optimization
The MACC PMS (Processor/Memory/Switch) Model

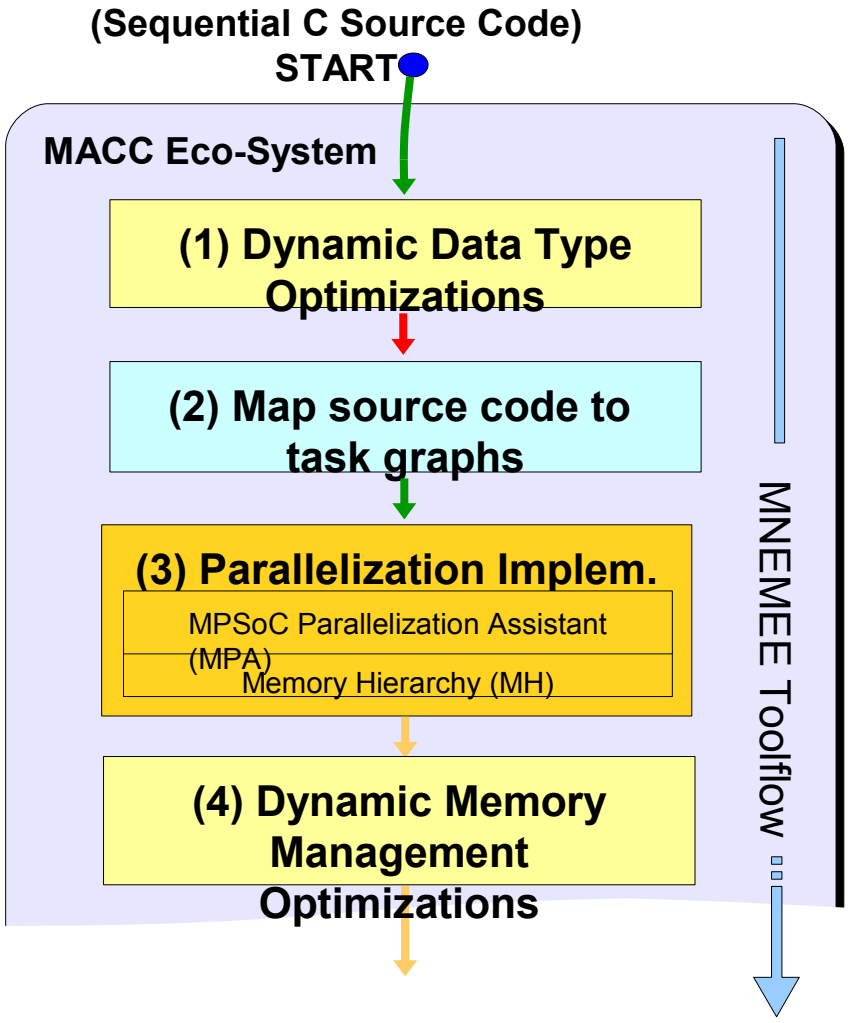
- Explicit memory architecture
- API provides access to memory information



MaCC Modeling Example via GUI



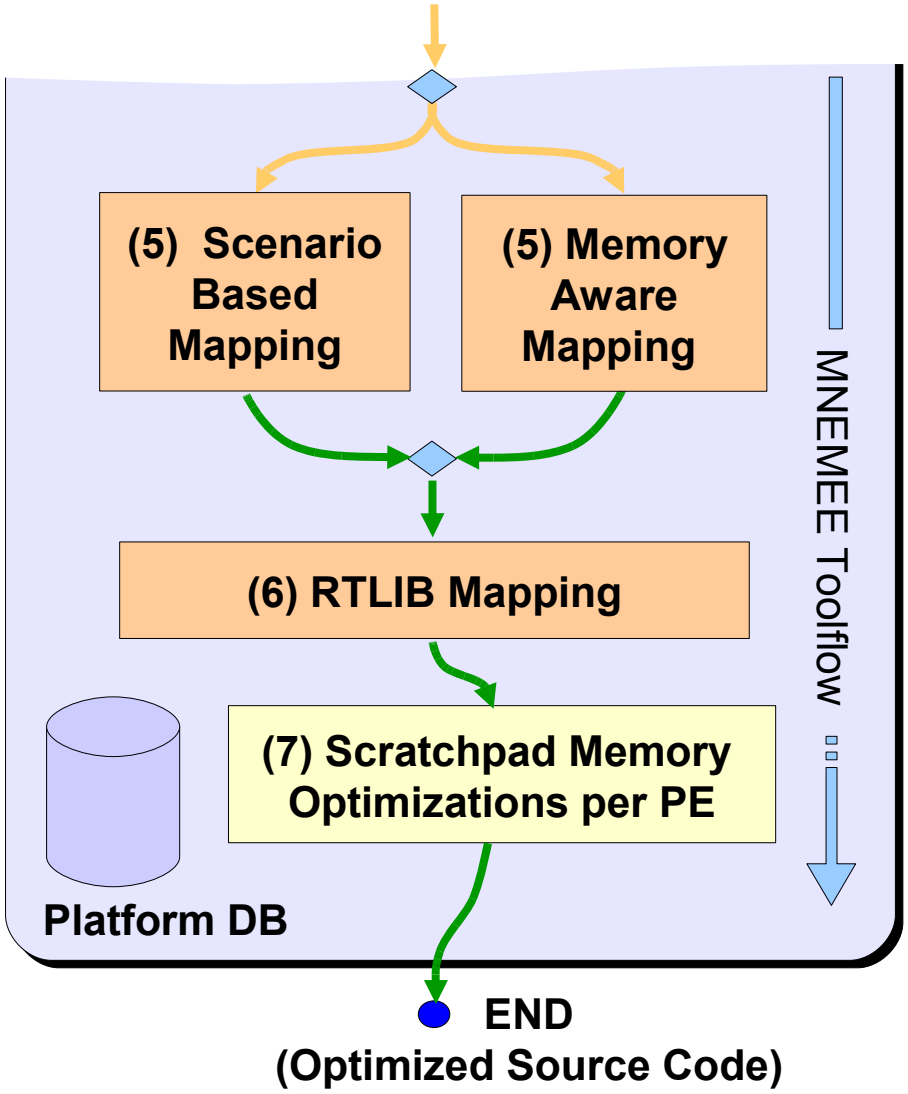
Toolflow Detailed View



1. Optimization of dynamic data structures
2. Extraction of potential parallelism
3. Implementation of parallelism; placement of static data
4. Placement of dynamic data

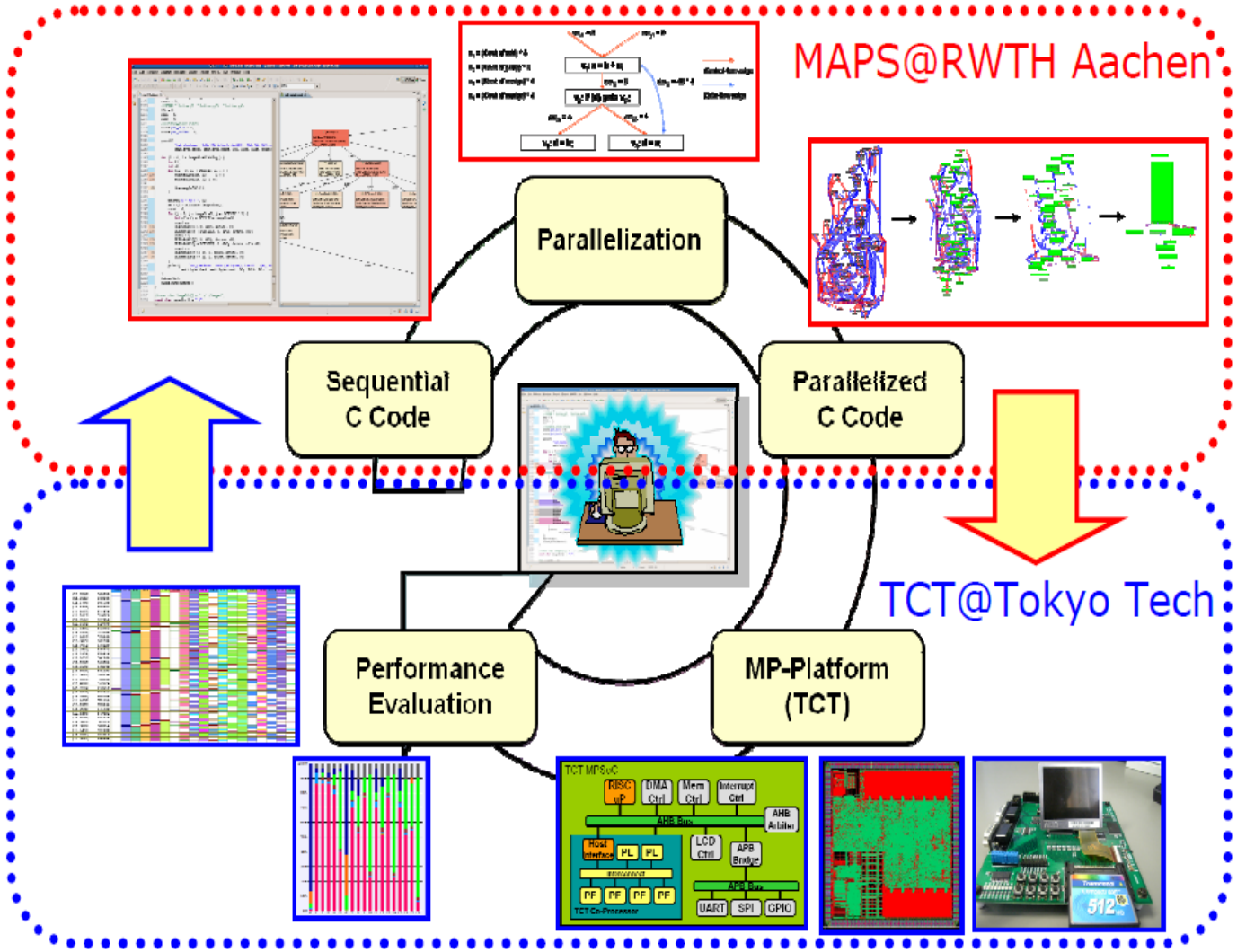
Status:
- Fully automated
- Known automated approach
- Manual transformation required

Toolflow Detailed View



5. Perform mapping to processing elements
   - Scenario based
   - Memory aware
5. Transform the code to implement the mapping
6. Perform scratchpad memory optimizations for each processing element

MAPS-TCT Framework



Rainer Leupers, Weihua Sheng: MAPS: An Integrated Framework for MPSoC Application Parallelization, *1st Workshop on Mapping of Applications to MPSoCs, Rheinfels Castle*, 2008

Future Work

- Completion of the Mnemee tool flow
- 3rd Workshop on Mapping of Applications to MPSoCs
  - To be held June 28-29, 2010, at Rheinfels Castle
  - Information: http://www.artist-embedded.org/artist/-map2mpsoc-2010-.html
- Work by other researchers in the area

Summary

- Clear trend toward multi-processor systems for embedded systems
- Using architecture **crucially** depends on **mapping tools**
- ArtistDesign network cluster focusing on mapping
- Providing an overview of available techniques
- Two criteria for classification
  - Fixed / flexible architecture
  - Auto parallelizing / non-parallelizing
- Introduction to proposed Mnemee tool chain
- Future work
- Summary