Year 1 Review
Brussels, January 23rd, 2008

Cluster

Achievements and Perspectives :

Software Synthesis, Code Generation and Timing Analysis

Cluster leader : Peter Marwedel

TU Dortmund

High-Level Objectives

- Limitations of increasing clock speeds any further
  ☞ focus on using MP platforms, in particular MPSoCs
  (increased importance since the proposal writing)
- Different from multi-core situation: Multiple applications, heterogeneous processors, and multiple objectives
- MP platforms pose threats to timing predictability;
  - make MPSoC architects aware of hazards; develop MP/MPSoC design principles for maximal predictability
  - develop models/methods for timing analysis of parallel software
- Efficient design + software synthesis also in the scope
  (see deliverables; coordination with other projects)
- Partners contribution to transversal clusters
  (e.g. for predictability: WCC)

Building Excellence

- NoE provides the required size of research teams, necessary to handle the complexity of technology.
- (Joint) Rheinfels workshop, St. Goar, June 2008
  Incorporating external experts
- Working meeting at Düsseldorf, Nov. 27-28, 2008
- TA meeting at Paris
- TA meeting at Prague
- Tutorials, summer schools, teaching at ALARI, WESE, …

State of the Integration in Europe

- Mapping to MP platforms generally seen as one of the largest challenges of current and future technology
- Interaction with Execution Platform Cluster
- Work from outside the cluster
  - DAEDALUS (Leiden)
  - SystemCoDesigner (Erlangen)
  - HOPES (Seoul)

  } Coming affiliates

  - Cooperation with hipeac, CoreGrid, KDUbiq, … NoEs
  - Cooperation with European projects Predator, Mnemee, EmBounded, …
  - Cooperation with companies (AbsInt, ACE, CoWare, ICD, Infineon, NXP, Rapita, ST)
- Liberation from integration into complex compilers

Overall Assessment and Vision at Y0+1

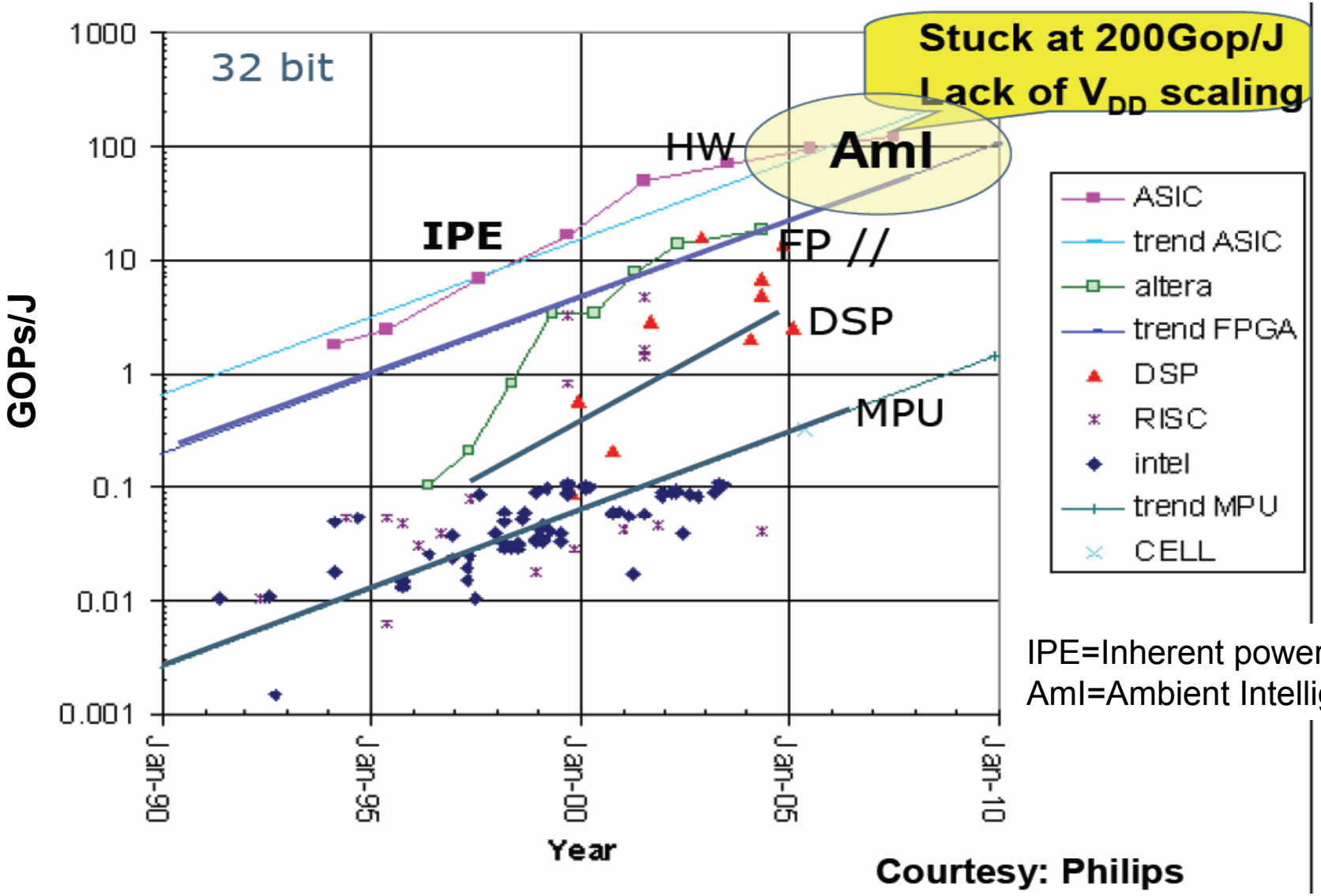
What went well:

- Abundant amount of results on resource-efficiency
- Results from mapping tools
- Establishment of a well-visible workshop
- Agreement to cooperate on benchmarks
- While ArtistDesign partners acted as a core, the involvement of researchers went far beyond ArtistDesign

Weaknesses:

- Limited support of full range of software synthesis
- Do timing analysis concerns affect MPSoC architecture?

Energy Efficiency

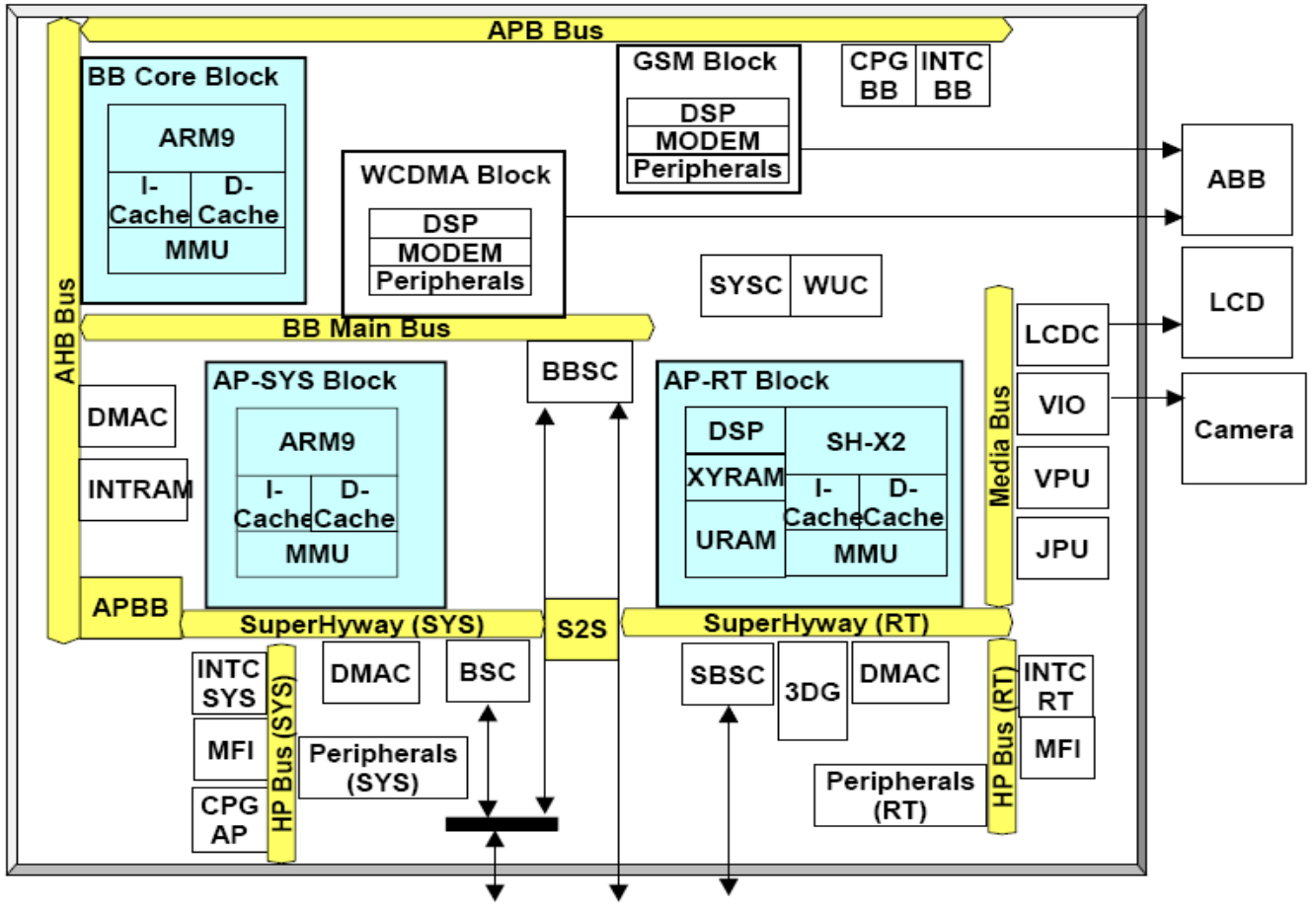


Courtesy: Philips
© Hugo De Man, IMEC, 2007

IPE=Inherent power efficiency
AmI=Ambient Intelligence

Heterogeneous Architectures

G1 Module Diagram



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

Problem Description

- **Given**
  - A set of applications
  - Use cases
  - 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

Not many contributions yet!

Tools urgently needed!

1st Workshop on Mapping Applications To MPSoCs, Rheinfels castle, June, 2008

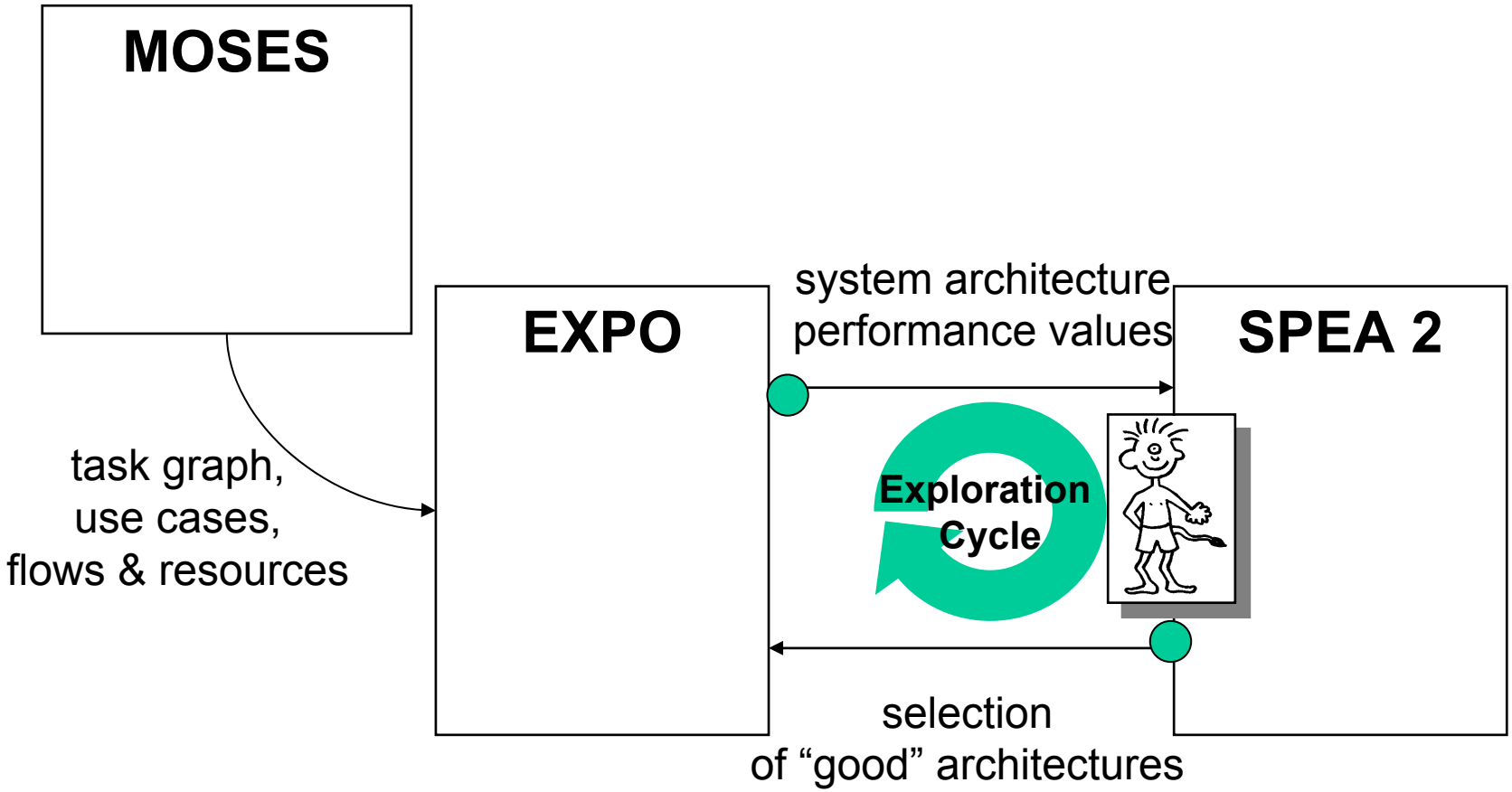
- Future architectures of MPSoCs (John Goodacre, ARM)
- SPEA2/DOL (Lothar Thiele, ETHZ)
- Car-Entertainment Applications ☞ MP Systems (Marco Bekooij, NXP)
- MAPS (Rainer Leupers, Aachen)
- Overview over parallelization techniques (C. Lengauer, U. Passau)
- DSE of Heterogeneous MPSoCs (Ristau, Fettweis, TU Dresden)
- Daedalus (Ed Deprettere, U. Leiden)
- Mapping to the CELL processor (U. Bologna and others)
- Timing analysis issues (various)

Programme and slides: http://www.artist-embedded.org/artist/-map2mpsoc-2008-.html

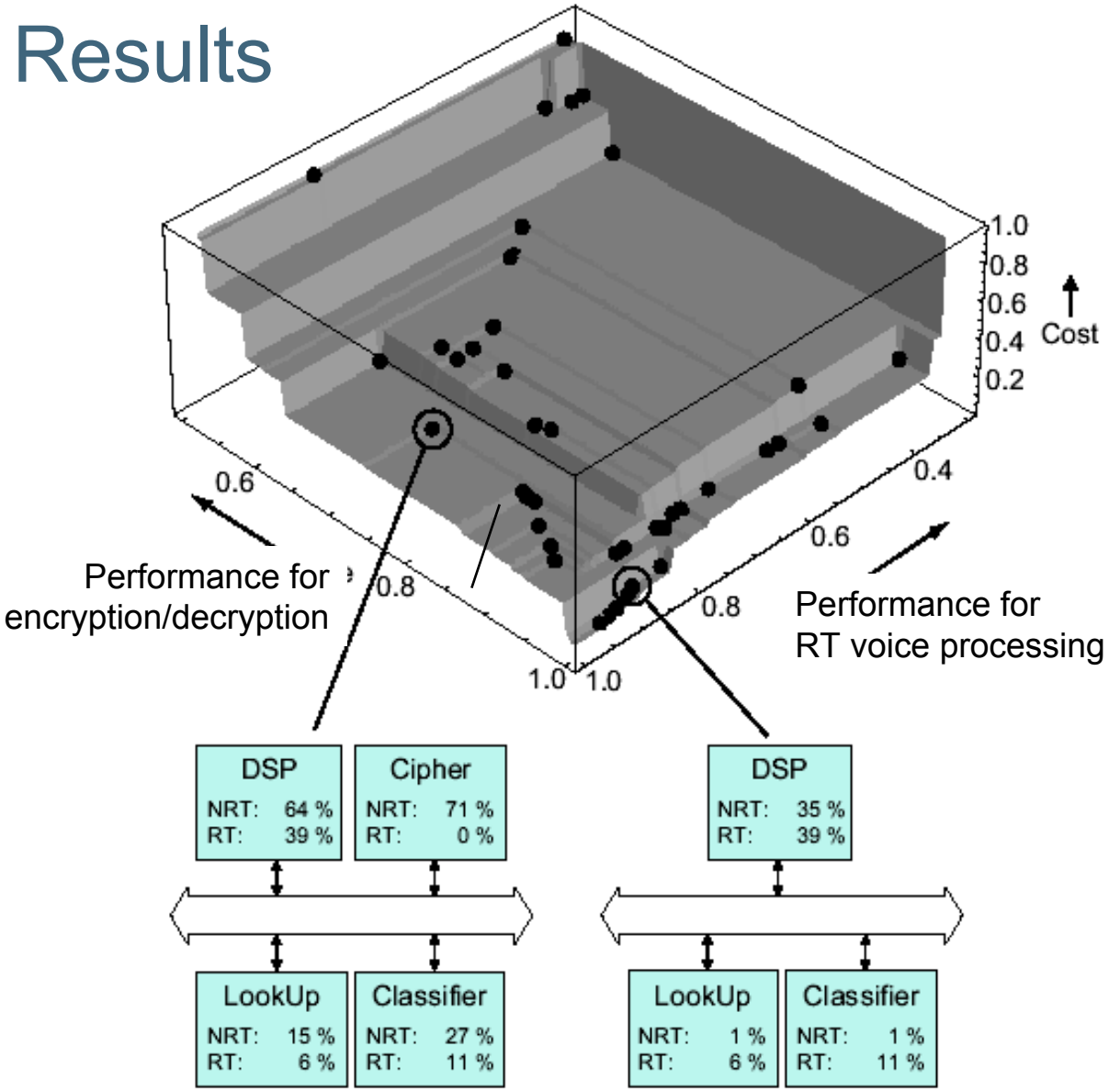
A Simple Classification

Architecture fixed/ Auto-parallelizing	Fixed Architecture	Architecture to be designed
Starting from given model	Map to CELL, HOPES, ETHAM	COOL codesign tool; EXPO/SPEA2/ DOL
Auto-parallelizing	U. Edinburgh Mnemee MAPS	Daedalus

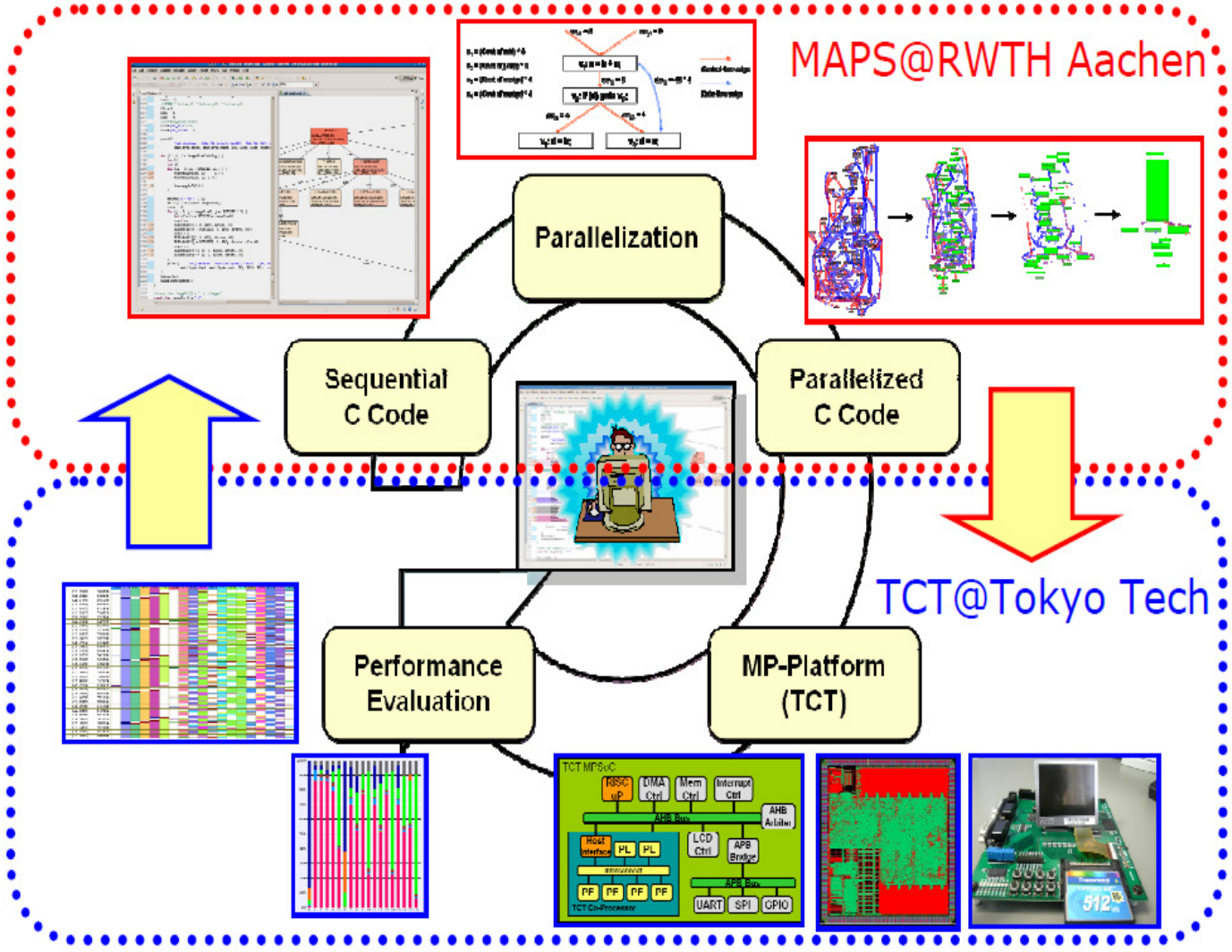
EXPO/SPEA2/DOL – Tool architecture



Results

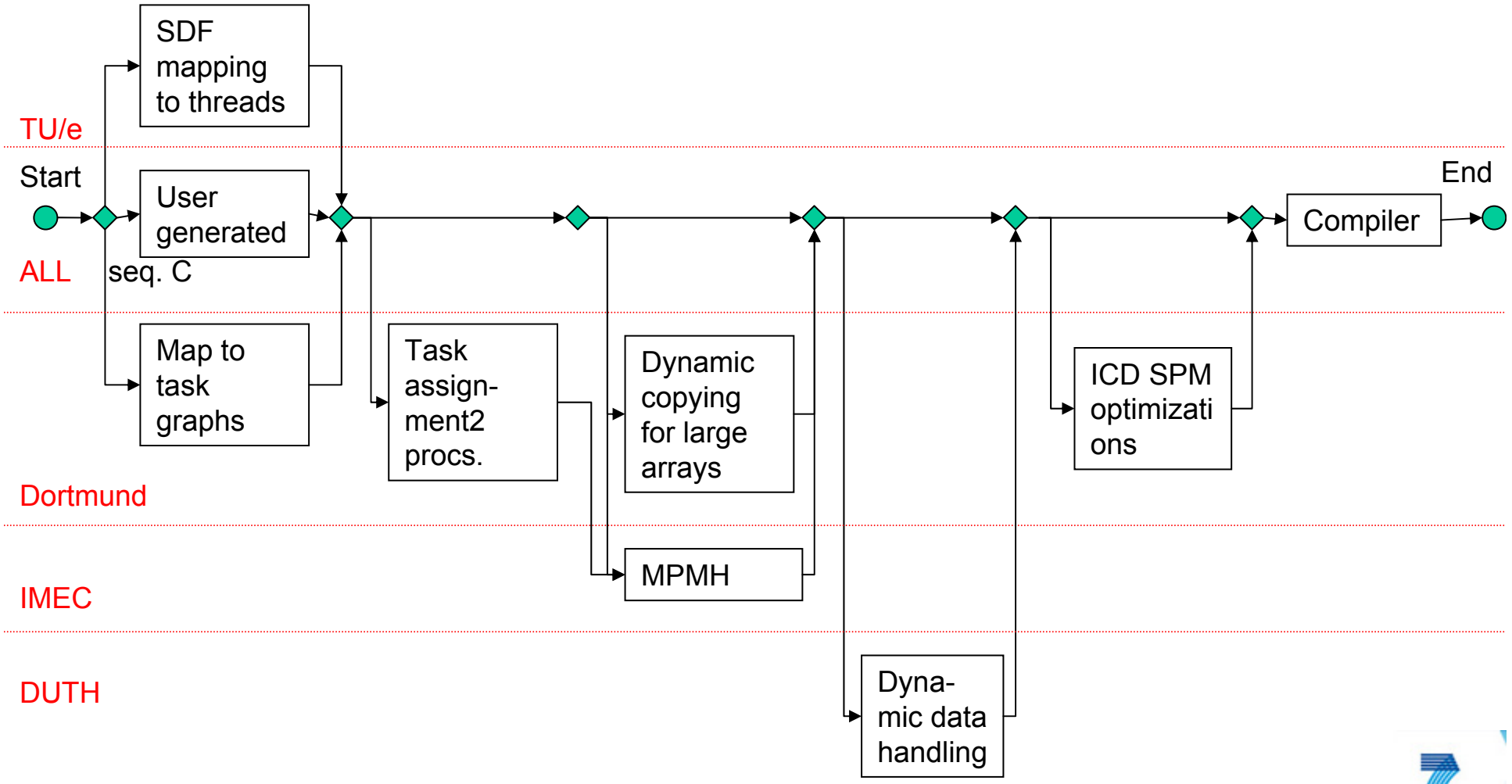


MAPS-TCT Framework



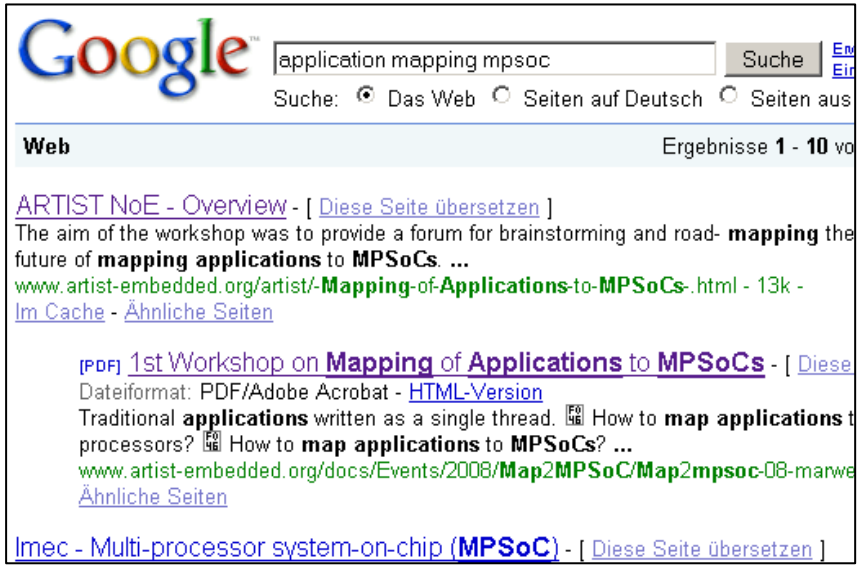
Rainer Leupers, Weihua Sheng: MAPS: An Integrated Framework for MPSoC Application Parallelization, *1st Workshop on Mapping of Applications to MPSoCs, Rheinfels Castle*, 2008

Proposed Mnemee Tool Flow (Simplified)



Meeting Results

- Cooperation:
  - On Mapping: Aachen, Dresden, Zürich, Leiden, Dortmund
  - Code Generation: Passau, Saarbrücken, Edinburgh, Leiden
  - Scenarios: Eindhoven, IMEC, Edinburgh
  - CleanC: IMEC, Edinburgh
- Report at the CASA Workshop, Embedded Systems Week, Atlanta, 2008
- Web site (incl. Slides)



Working Meeting Düsseldorf, Nov. 27-28, 2009

- Cooperation of realistic test benches/benchmarks
- Subset of C as a common exchange format
- Polylib format commonly used by various partners.
- Cooperation Alain Dartes / Leiden on buffer sizing
- **ETHZ/Dortmund cooperation on memory modeling**
- ETHZ/Aachen cooperation on virtual simulation platforms
- Leiden/Passau cooperation on dependence analysis
- Leiden/Aachen: dependence analysis for dynamic applications
- Using Passau's approach for more sophisticated parallelization
- Cooperation Erlangen and TU/e on the exploration of SDF
- **Joint front-ends**?
- Consideration of explicitly parallel code
- Common "intermediate" representation?
- Memory mapping

Persistent Web site: http://www.artist-embedded.org/artist/Mapping-of-Applications-to-MPSoCs,1590.html

Plans for Y2

- Implementing agreed-upon cooperation
- Working Meeting: DATE 2009
- 2nd Workshop on Mapping of Applications to MPSoCs
  - To be held June 29-30, 2009, at Rheinfels Castle
  - Information: http://www.artist-embedded.org/artist/-map2mpsoc-2009-.html
  - Followed by Mnemee meeting

Timing Analysis Y1

- Main task for Y1: initiate research on timing analysis for MC/MPSoC systems
- Very little previous research
- Success of timing analysis critically dependent on the organization of these systems (both HW and SW)
- Timing analysis cannot be considered in isolation anymore
- Much work during Y1 on formulating system design principles to maximize timing predictability while not sacrificing too much performance

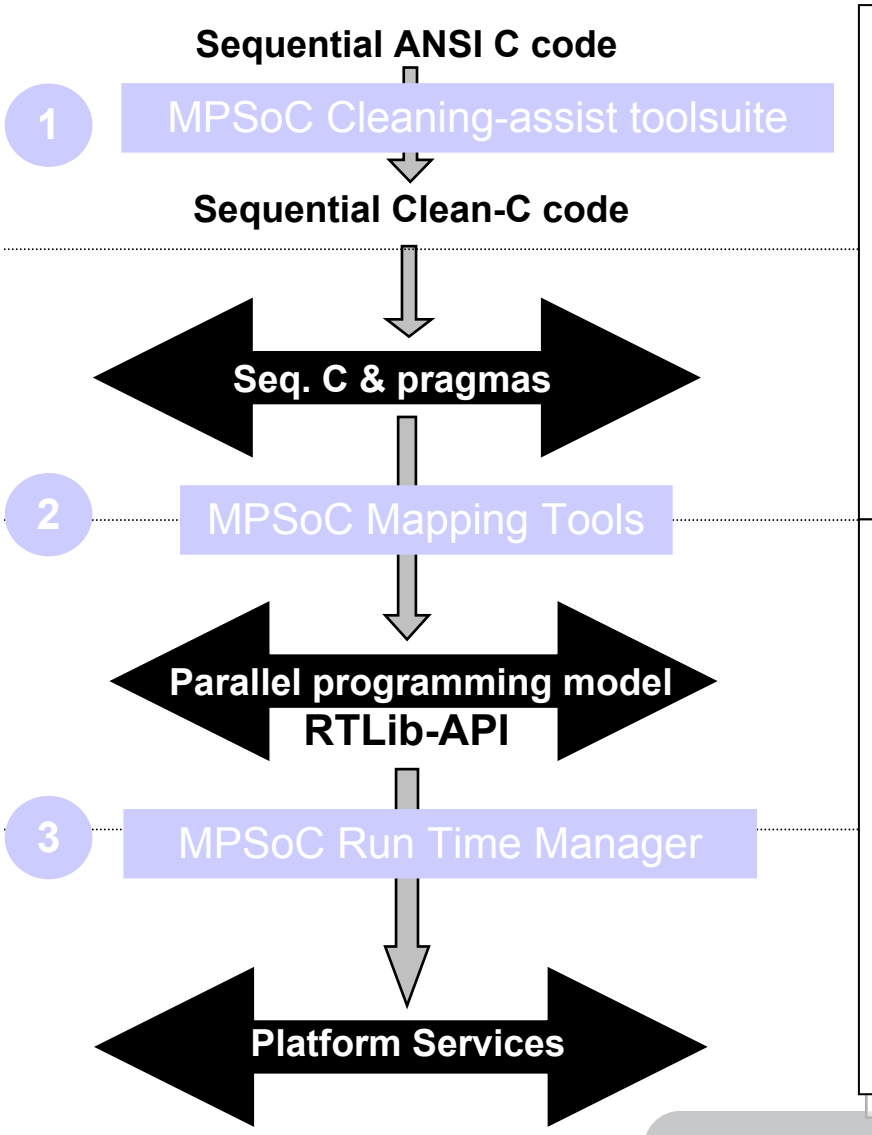
System Design for Timing Predictability

- Two different efforts (will consider relation/possible merge during Y2)
- (No general solution, we believe solutions will be application-specific to a high degree)
- Main problem is shared resources: they allow side effects that affect timing predictability adversely
- Thus, both efforts aim at *minimizing the use of common resources*, and subsequently *put the remaining use under strict control*

Plans for Y2

- WCET Tool Challenge
- WCET Workshop 2009
- Meeting with MPSoC design cluster at DATE 2009, Nice
- Write a position paper on timing predictability for parallel systems (challenges, design issues, analysis issues, ...)
- Continue initiated work on design principles for predictability
- Initiate work on timing analysis of explicitly parallel programs
- Continue work on measurement-based methods

Spare Slides: IMEC ArtistDesign highlights



ArtistDesign SSCGTA cluster

ArtistDesign MPSoC&HW Cluster

ArtistDesign OS&Networks cluster

Highlights

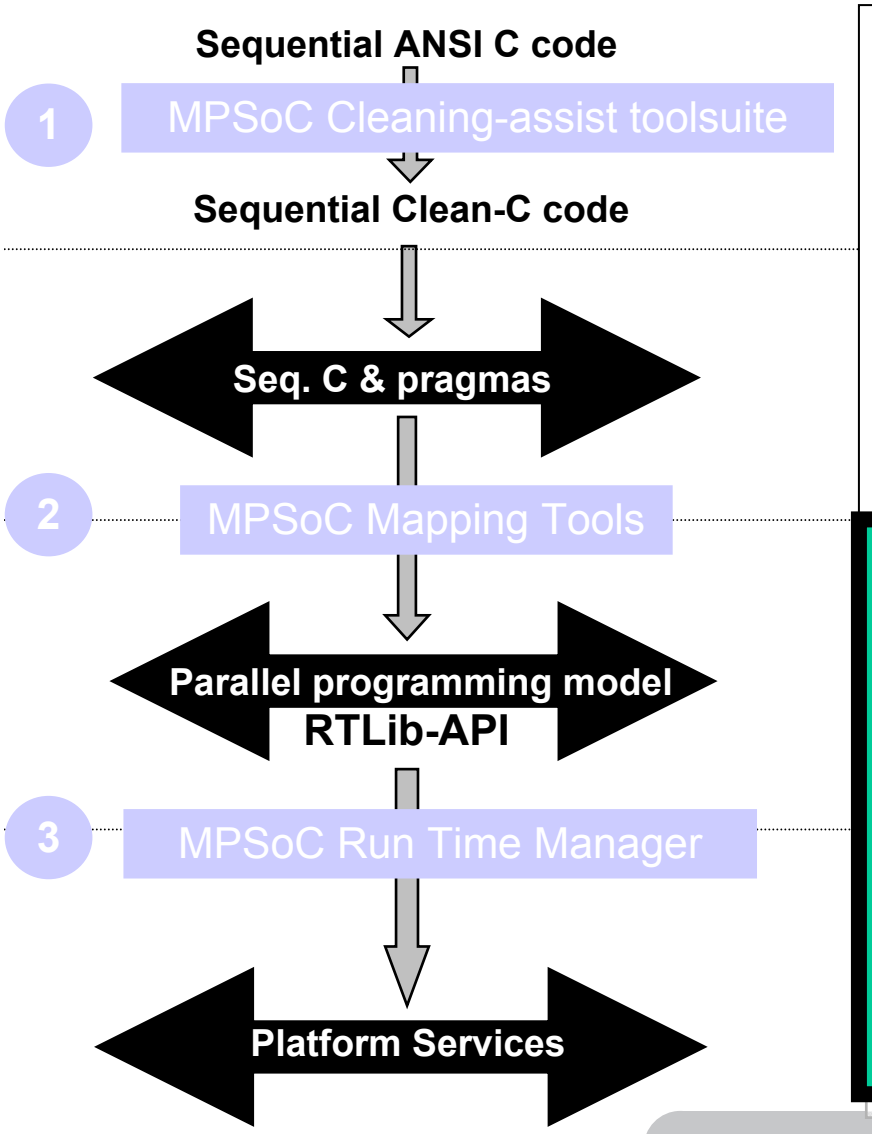
- Transforms ANSI C code to **CleanC** (= easy to parallelize code)
- Open Source

- Memory Hierarchy (**MH**) tool for source to source transformations to optimize memory transfer and storage
- MNEMEE FP7 project

- MPSoC Parallelization Assistant (**MPA**) tool for semi-automatic source code parallelization
- MOSART FP7 project

- Metadata based Task Concurrency Management (**TCM**) scheduler for MPSoC
- OptiMMA IWT project

IMEC MPSoC&HW cluster highlights (MPA+TCM)



ArtistDesign SSCGTA cluster

ArtistDesign MPSoC&HW Cluster

ArtistDesign OS&Networks cluster

Highlights

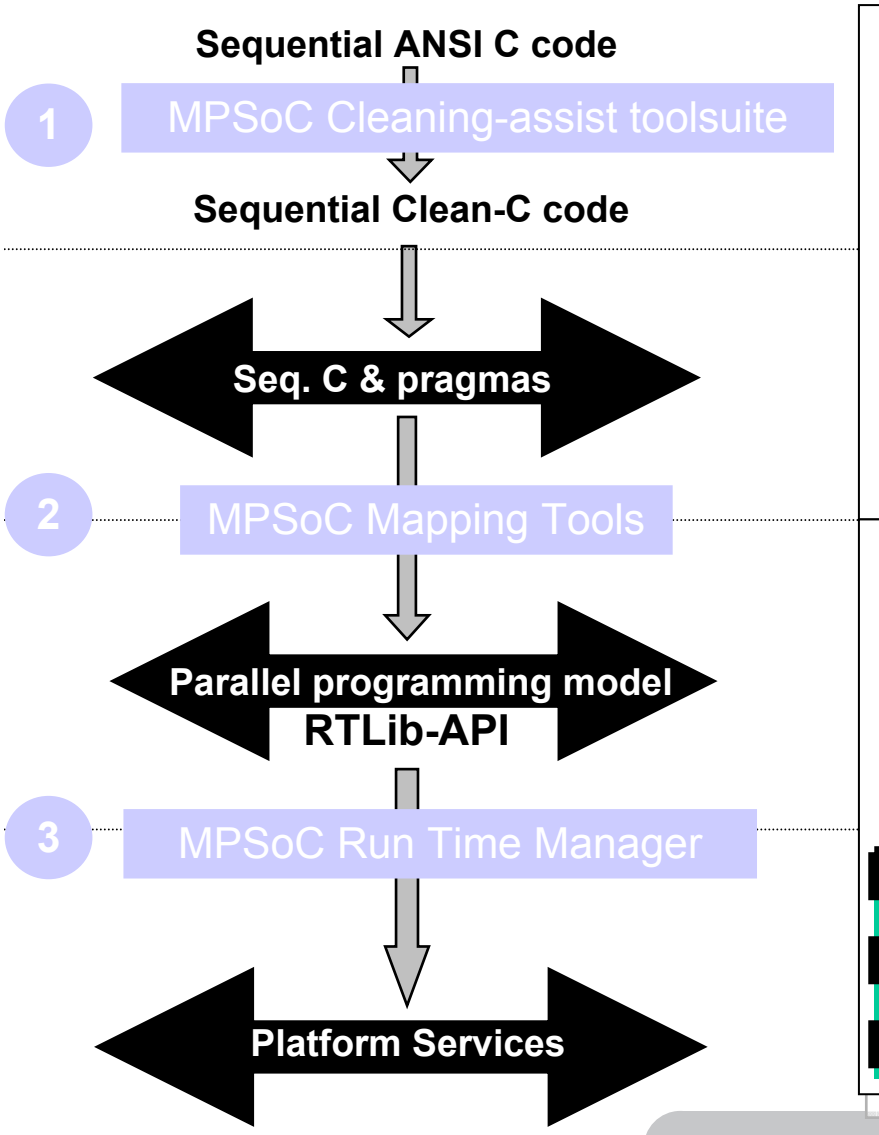
- Transforms ANSI C code to **CleanC** (= easy to parallelize code)
- Open Source

- Memory Hierarchy (**MH**) tool for source to source transformations to optimize memory transfer and storage
- MNEMEE FP7 project

- MPSoC Parallelization Assistant (**MPA**) tool for semi-automatic source code parallelization
- MOSART FP7 project

- Metadata based Task Concurrency Management (**TCM**) scheduler for MPSoC
- OptiMMA IWT project

IMEC OS&Networks cluster highlights (TCM)



ArtistDesign SSCGTA cluster

ArtistDesign MPSoC&HW Cluster

ArtistDesign OS&Networks cluster

Highlights

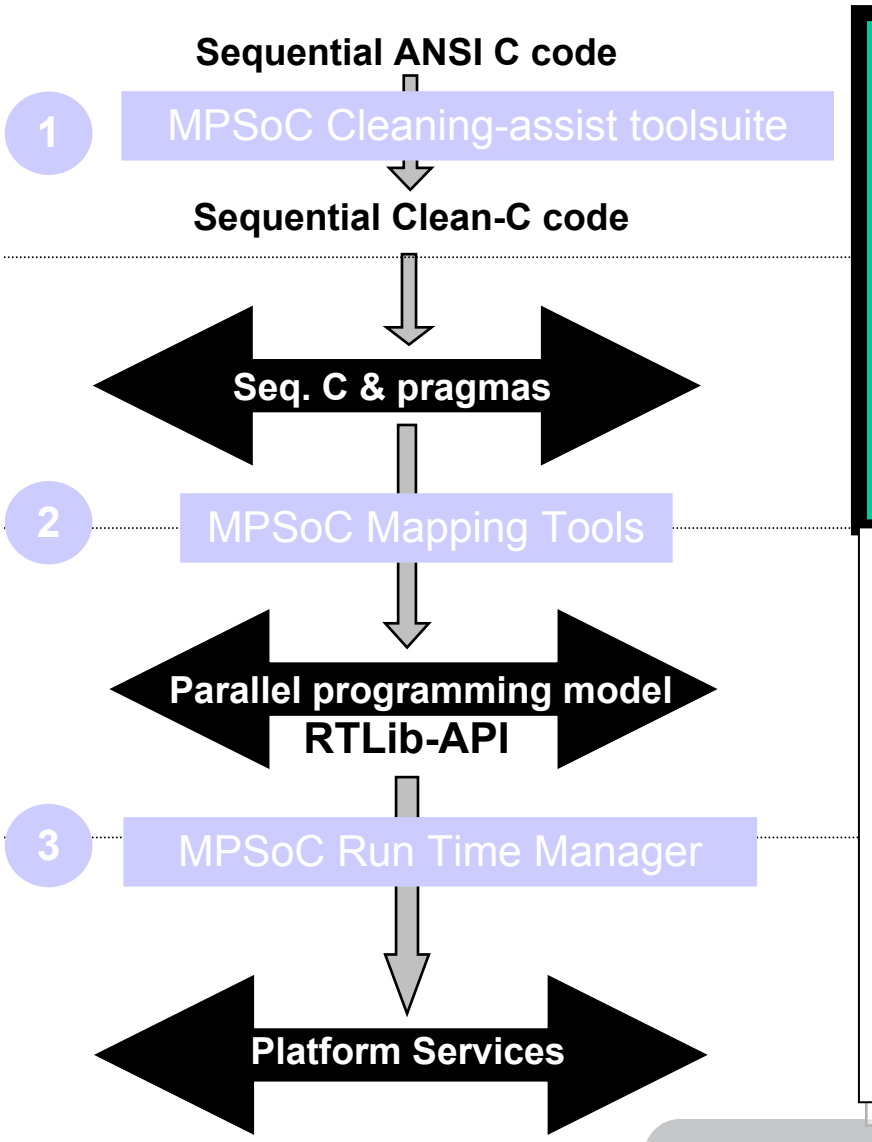
- Transforms ANSI C code to **CleanC** (= easy to parallelize code)
- Open Source

- Memory Hierarchy (**MH**) tool for source to source transformations to optimize memory transfer and storage
- MNEMEE FP7 project

- MPSoC Parallelization Assistant (**MPA**) tool for semi-automatic source code parallelization
- MOSART FP7 project

- Metadata based Task Concurrency Management (**TCM**) scheduler for MPSoC
- OptiMMA IWT project

IMEC SSCGTA cluster highlights (CleanC+MH)



ArtistDesign SSCGTA cluster

ArtistDesign MPSoC&HW Cluster

ArtistDesign OS&Networks cluster

Highlights

- Transforms ANSI C code to **CleanC** (= easy to parallelize code)
- Open Source

- Memory Hierarchy (**MH**) tool for source to source transformations to optimize memory transfer and storage
- MNEMEE FP7 project

- MPSoC Parallelization Assistant (**MPA**) tool for semi-automatic source code parallelization
- MOSART FP7 project

- Metadata based Task Concurrency Management (**TCM**) scheduler for MPSoC
- OptiMMA IWT project

Related Work

- 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