

Year 3 Review
Brussels, February 24th, 2010

Transversal Activity

Achievements and Perspectives :

Design for Predictability and Performance

leader : Bengt Jonsson

Uppsala University

High-Level Objectives

- technology and design techniques for achieving predictability of systems
 - especially on modern (multi-core) platforms
- trade-offs between performance and predictability

Predictability transverses all levels of abstraction in embedded systems design:

- Verification, modeling, compilation, OS, execution platforms.

Industrial Sectors

- Safety-critical systems:
 - transportation, power automation, medical systems, ...
 - Market of over \$900 million in 2008 [int. ARC Advisory Group]
- Sectors where systems failure leads to severe economic consequences:
 - consumer electronics, telecom, ...
- Computer systems that require **both** precise execution time **and** high throughput

Partners

Modeling & Validation

- IST (Tom Henzinger)
- INRIA (Alain Girault)
- Uppsala (Bengt Jonsson)
- Trento (Alberto Sangiovanni–Vincentelli)

Code Generation & Timing analysis

- Dortmund (Peter Marwedel)
- Saarland (Reinhard Wilhelm)
- TU Vienna (Peter Puschner)
- AbsInt (Christian Ferdinand) [affiliated]
- Tidorum (Niklas Holsti) [affiliated]

OS & Networks

- Cantabria (Michael Gonzalez–Harbour)
- SSSA (Giorgio Buttazzo)
- York (Alan Burns)

Hardware Platforms & MPSoC

- Bologna (Luca Benini)
- Braunschweig (Rolf Ernst) [affiliated]
- ETHZ (Lothar Thiele)
- IMEC (Stylianos Mamagkakis)
- Linköping (Petru Eles)

Building Excellence

Most existing work was within one system level, e.g.,:

- Modeling and verification of timed component-based systems,
- Timing analysis for programs
- Compiler techniques for timing and memory predictability
- OS Scheduling and resource management
- Predictability in memory

Main goal of the predictability activity:

- To integrate research across different levels of abstraction

Time Predictability on Multiprocessor systems (Braunschweig + ETHZ)

- Modeling and Analysis method for multiprocessor systems with shared resources
 - Capture more accurately the interference between different cores that share common resources
 - improved analysis results
 - Investigate multiprocessor schedulability depending on the shared resource arbitration protocols
- COMPOSE project (Braunschweig + Intel Labs)

Modeling & Validation:

- IST - Austria (Tom Henzinger)
- INRIA – France (Alain Girault)
- Uppsala (Bengt Jonsson)
- Trento (Alberto Sangiovanni–Vincentelli)

Code Generation & Timing analysis

- Dortmund (Peter Marwedel)
- Saarland (Reinhard Wilhelm)
- TU Vienna (Peter Puschner)

OS & Networks

- Cantabria (Michael Gonzalez–Harbour)
- SSSA (Giorgio Buttazzo)
- York (Alan Burns)

Hardware Platforms & MPSoC

- Bologna (Luca Benini)
- Braunschweig (Rolf Ernst) [amiatoc]
- ETHZ – Zürich (Lothar Thiele)
- IMEC – Belgium (Stylianos Mamagkakis)
- Linköping (Petru Eles)

Cache Analysis for PLRU and FIFO (USAAR)

- FIFO–must–analysis is harder than LRU–must–analysis
- Similar ideas and results for may–analysis
- Significant improvements to FIFO–must/may–analyses, still efficiently implementable
- PLRU is significantly harder to predict than FIFO/LRU (e.g., Reineke-metrics)
- Improvement: Considering subtree distance (when subtree bits are flipped, their leaves cannot be evicted)
- May-analysis: still open problem

Modeling & Validation:

- IST - Austria (Tom Henzinger)
- INRIA – France (Alain Girault)
- Uppsala (Bengt Jonsson)
- Trento (Alberto Sangiovanni–Vincentelli)

Code Generation & Timing analysis

- Dortmund (Peter Marwedel)
- Saarland (Reinhard Wilhelm)
- TU Vienna (Peter Puschner)

OS & Networks

- Cantabria (Michael Gonzalez–Harbour)
- SSSA (Giorgio Buttazzo)
- York (Alan Burns)

Hardware Platforms & MPSoC

- Bologna (Luca Benini)
- Braunschweig (Rolf Ernst) [affiliated]
- ETHZ – Zürich (Lothar Thiele)
- IMEC – Belgium (Stylios Mamagkakis)
- Linköping (Petru Eles)

Timing Predictability of Hardware (USAAR)

Multi-Core architectures are notoriously problematic for timing analysis:

- Interferences between shared resources
 - *Virtual* interferences: Artifacts of the abstraction (*open problem*)
 - *Inherent* interferences: Actual interference which might in reality WCETs of tasks (*can be bounded*)
- Design principles and guidelines given to limit sharing and interferences while meeting cost/performance constraints
- Examples given for predictable configurations of recent multi-core architectures:
 - MPC5668G (e200z6/e200z0 cores), automotive domain
 - MPC8641D (derivative of MPC7448), avionics domain

Modeling & Validation:

- IST - Austria (Tom Henzinger)
- INRIA – France (Alain Girault)
- Uppsala (Bengt Jonsson)
- Trento (Alberto Sangiovanni–Vincentelli)

Code Generation & Timing analysis

- Dortmund (Peter Marwedel)
- Saarland (Reinhard Wilhelm)
- TU Vienna (Peter Puschner)

OS & Networks

- Cantabria (Michael Gonzalez–Harbour)
- SSSA (Giorgio Buttazzo)
- York (Alan Burns)

Hardware Platforms & MPSoC

- Bologna (Luca Benini)
- Braunschweig (Rolf Ernst) [affiliated]
- ETHZ – Zürich (Lothar Thiele)
- IMEC – Belgium (Stylianos Mamagkakis)
- Linköping (Petru Eles)

Dynamic Memory Allocation (USAAR)

Two techniques:

- *Replace dynamic allocation by static allocation:*
 - Analyse program to derive its allocation behaviour (with parametrical bounds on number of allocated objects)
 - Compute conflict-free cache-efficient allocation patterns
- *Perform predictable dynamic allocation:*
 - Allows to place allocated memory into specified cache sets
 - Caches sets can be chosen automatically or by analysis

Practical tests:

- Static memory allocation has comparable memory consumption
- Predictable dynamic allocation does not sacrifice speed

Modeling & Validation:

- IST - Austria (Tom Henzinger)
- INRIA – France (Alain Girault)
- Uppsala (Bengt Jonsson)
- Trento (Alberto Sangiovanni–Vincentelli)

Code Generation & Timing analysis

- Dortmund (Peter Marwedel)
- Saarland (Reinhard Wilhelm)
- TU Vienna (Peter Puschner)

OS & Networks

- Cantabria (Michael Gonzalez–Harbour)
- SSSA (Giorgio Buttazzo)
- York (Alan Burns)

Hardware Platforms & MPSoC

- Bologna (Luca Benini)
- Braunschweig (Rolf Ernst) [affiliated]
- ETHZ – Zürich (Lothar Thiele)
- IMEC – Belgium (Stylios Mamagkakis)
- Linköping (Petru Eles)

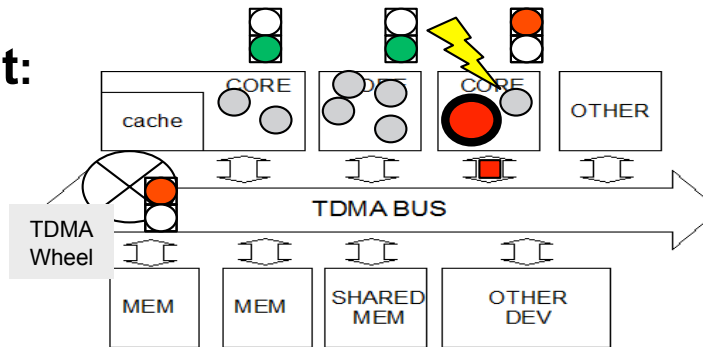
Adaptive TDMA Bus Allocation and Elastic Scheduling (UNIBO + SSSA)

Reconfigurable system:

- TDMA bus and OS scheduler work synergistically in case of run-time workload changes
- Ensures the highest utilization of processors

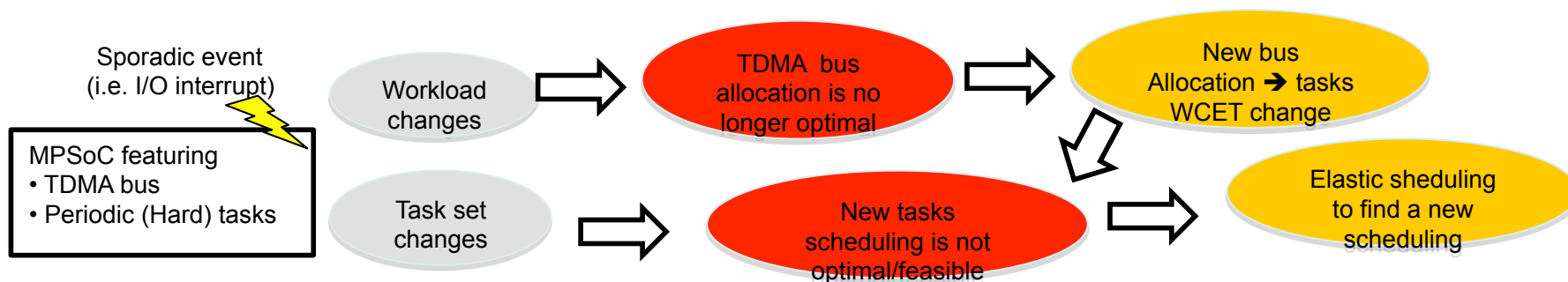
Experiments on RT task set:

- Avionic Tasks
- Mathematical Task
- Coremark benchmark suite



Our algorithm:

- Reacts in less than 100ms after workload variation
- Overhead is < 5% of task computation time



Modeling & Validation:

- . IST - Austria (Tom Henzinger)
- . INRIA – France (Alain Girault)
- . Uppsala (Bengt Jonsson)
- . Trento (Alberto Sangiovanni–Vincentelli)

Code Generation & Timing analysis

- . Dortmund (Peter Marwedel)
- . Saarland (Reinhard Wilhelm)
- . TU Vienna (Peter Puschner)

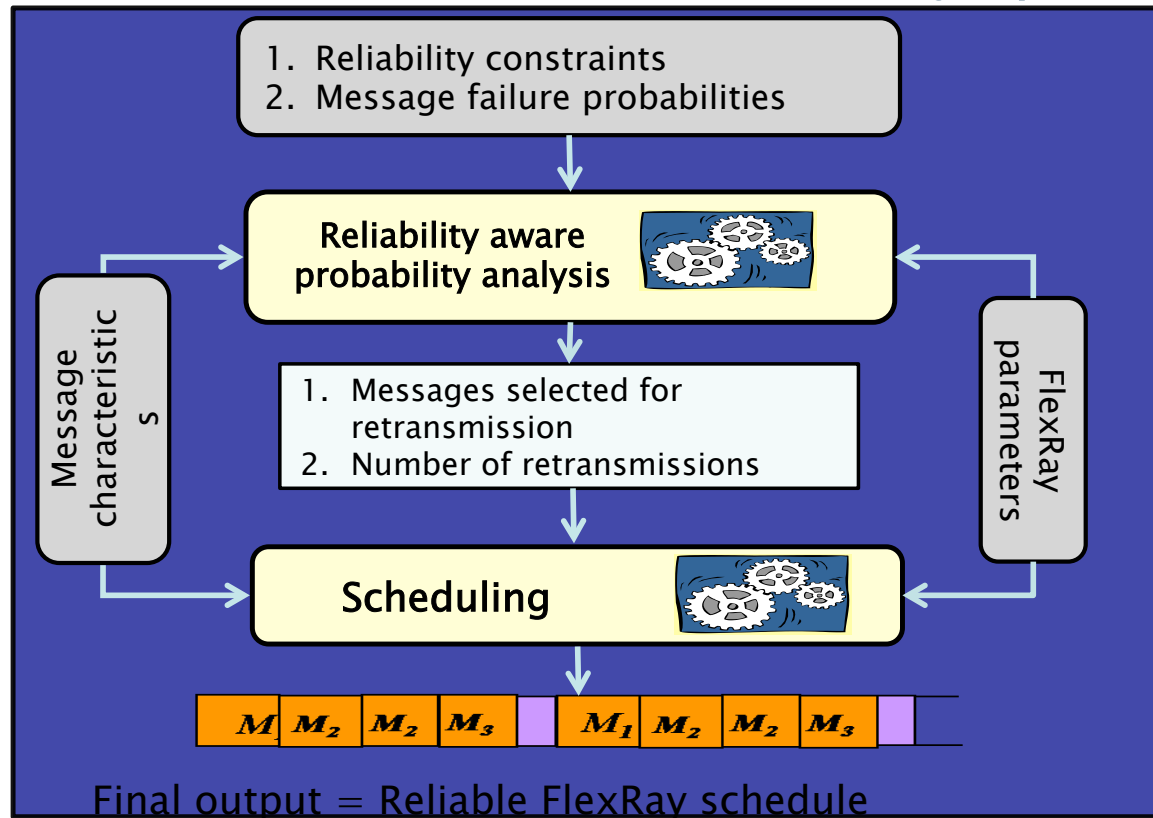
OS & Networks

- . Cantabria (Michael Gonzalez Harbour)
- . SSSA (Giorgio Buttazzo)
- . York (Alan Burns)

Hardware Platforms & MPSoC

- . Bologna (Luca Benini)
- . Braunschweig (Rolf Ernst) [affiliated]
- . ETHZ – Zürich (Lothar Thiele)
- . IMEC – Belgium (Stylios Mamagkakis)
- . Linköping (Petru Eles)

Scheduling for predictable, fault-tolerant communication on FlexRay (Linköping + DTU)



Modeling & Validation:

- IST - Austria (Tom Henzinger)
- INRIA – France (Alain Girault)
- Uppsala (Bengt Jonsson)
- Trento (Alberto Sangiovanni-Vincentelli)

Code Generation & Timing analysis

- Dortmund (Peter Marwedel)
- Saarland (Reinhard Wilhelm)
- TU Vienna (Peter Puschner)

OS & Networks

- Cantabria (Michael Gonzalez-Harbour)
- SSSA (Giorgio Buttazzo)
- York (Alan Burns)

Hardware Platforms & MPSoC

- Bologna (Luca Benini)
- Braunschweig (Rolf Ernst) [affiliated]
- ETHZ – Zürich (Lothar Thiele)
- IMEC – Belgium (Stylianos Mamagkakis)
- Linköping (Petru Eles)

In-vehicle communication over FlexRay based networks

Reliability on FlexRay bus is achieved via retransmissions

- At a minimum bandwidth utilization cost

Component-based design under RT constraints (Cantabria)

Strategy for configuring the scheduling of component-based real-time applications:

- Implemented on top of the **RT-CCM** component technology
 - Scheduling configuration managed by the containers in an opaque way
- Scheduling configuration obtained from the analysis of the real-time model of the application
 - **CBS-MAST**: Extension of MAST adapted to Component-Based Design
- The application designers only manage the metadata included in the component descriptors
 - **RT-D&C**: Real-time extension of the *Deployment and Configuration of Component-Based Applications Specification* of the OMG

Modeling & Validation:

- IST - Austria (Tom Henzinger)
- INRIA – France (Alain Girault)
- Uppsala (Bengt Jonsson)
- Trento (Alberto Sangiovanni-Vincentelli)

Code Generation & Timing analysis

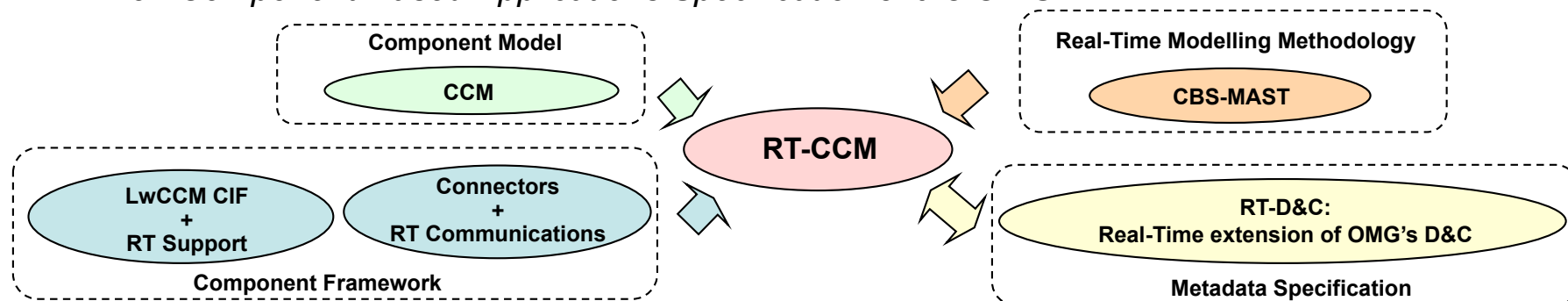
- Dortmund (Peter Marwedel)
- Saarland (Reinhard Wilhelm)
- TU Vienna (Peter Puschner)

OS & Networks

- Cantabria (Michael Gonzalez-Harbour)
- SSSA (Giorgio Buttazzo)
- York (Alan Burns)

Hardware Platforms & MPSoC

- Bologna (Luca Benini)
- Braunschweig (Rolf Ernst) [affiliated]
- ETHZ – Zürich (Lothar Thiele)
- IMEC – Belgium (Stylianios Mamagkakis)
- Linköping (Petru Eles)



Standardization of UML MARTE (Cantabria)

- Participation to SysML standardization
- Participation to MARTE, Real-time and Embedded systems profile for UML
 - Major role of Univ. Cantabria in the development of this standard
 - Evolution into MARTE 1.2, and soon MARTE 1.3
- Dissemination of MARTE
 - Built a collaborative web page for dissemination of the standard
 - Now it's the official OMG web page for MARTE
 - <http://www.omgmarTE.org>
 - Participated in the organization of the OMG's Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems

Modeling & Validation:

- IST - Austria (Tom Henzinger)
- INRIA – France (Alain Girault)
- Uppsala (Bengt Jonsson)
- Trento (Alberto Sangiovanni–Vincentelli)

Code Generation & Timing analysis

- Dortmund (Peter Marwedel)
- Saarland (Reinhard Wilhelm)
- TU Vienna (Peter Puschner)

OS & Networks

- Cantabria (Michael Gonzalez–Harbour)
- SSSA (Giorgio Buttazzo)
- York (Alan Burns)

Hardware Platforms & MPSoC

- Bologna (Luca Benini)
- Braunschweig (Rolf Ernst) [affiliated]
- ETHZ – Zürich (Lothar Thiele)
- IMEC – Belgium (Stylios Mamagkakis)
- Linköping (Petru Eles)

PRET_C: predictable multithreaded C (INRIA + Auckland)

- Simple synchronous extensions to C:
 - Reactive inputs and outputs
 - Synchronization barrier
 - Synchronous parallel thread spawning
 - Strong and weak abort
- Thread safe shared memory access
- Deterministic synchronous semantics
- ARPRET: specific hardware accelerator working with a softcore processor
- Predictable execution without sacrificing throughput
- Model-checking based timing analysis with UPPAAL

Modeling & Validation:

- IST - Austria (Tom Henzinger)
- INRIA - France (Alain Girault)
- Uppsala (Bengt Jonsson)
- Trento (Alberto Sangiovanni-Vincentelli)

Code Generation & Timing analysis

- Dortmund (Peter Marwedel)
- Saarland (Reinhard Wilhelm)
- TU Vienna (Peter Puschner)

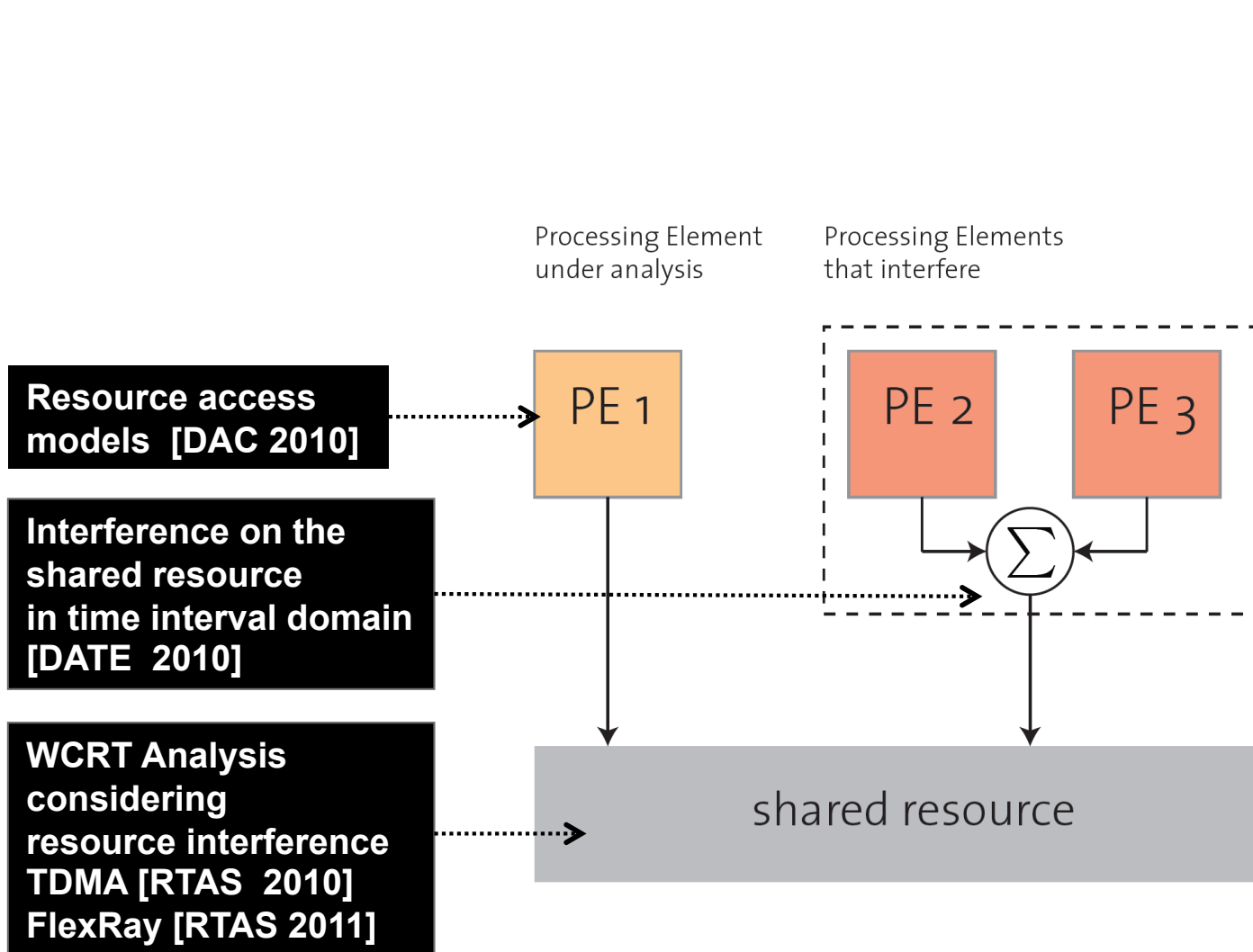
OS & Networks

- Cantabria (Michael Gonzalez-Harbour)
- SSSA (Giorgio Buttazzo)
- York (Alan Burns)

Hardware Platforms & MPSoC

- Bologna (Luca Benini)
- Braunschweig (Rolf Ernst) [affiliated]
- ETHZ - Zürich (Lothar Thiele)
- IMEC - Belgium (Stylios Mamagkakis)
- Linköping (Petru Eles)

Predictable Multi-core Communication (ETHZ)



Modeling & Validation:

- IST - Austria (Tom Henzinger)
- INRIA – France (Alain Girault)
- Uppsala (Bengt Jonsson)
- Trento (Alberto Sangiovanni–Vincentelli)

Code Generation & Timing analysis

- Dortmund (Peter Marwedel)
- Saarland (Reinhard Wilhelm)
- TU Vienna (Peter Puschner)

OS & Networks

- Cantabria (Michael Gonzalez–Harbour)
- SSSA (Giorgio Buttazzo)
- York (Alan Burns)

Hardware Platforms & MPSoC

- Bologna (Luca Benini)
- Braunschweig (Rolf Ernst) [affiliated]
- ETHZ – Zürich (Lothar Thiele)
- IMEC – Belgium (Stylianos Mavragkakis)
- Linköping (Petru Eles)

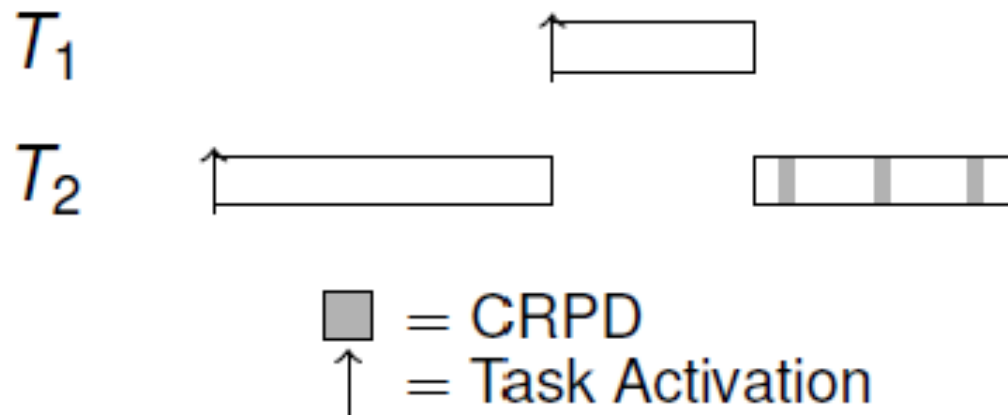
Scientific Highlights

- WCET Analysis in the Presence of Context Switches (USaar, AbsInt, SSSA)

Preemptive Scheduling

Cache related preemption delay (CRPD):

- Impact of preemption on the cache content
- Overall cost of additional reloads due to preemption



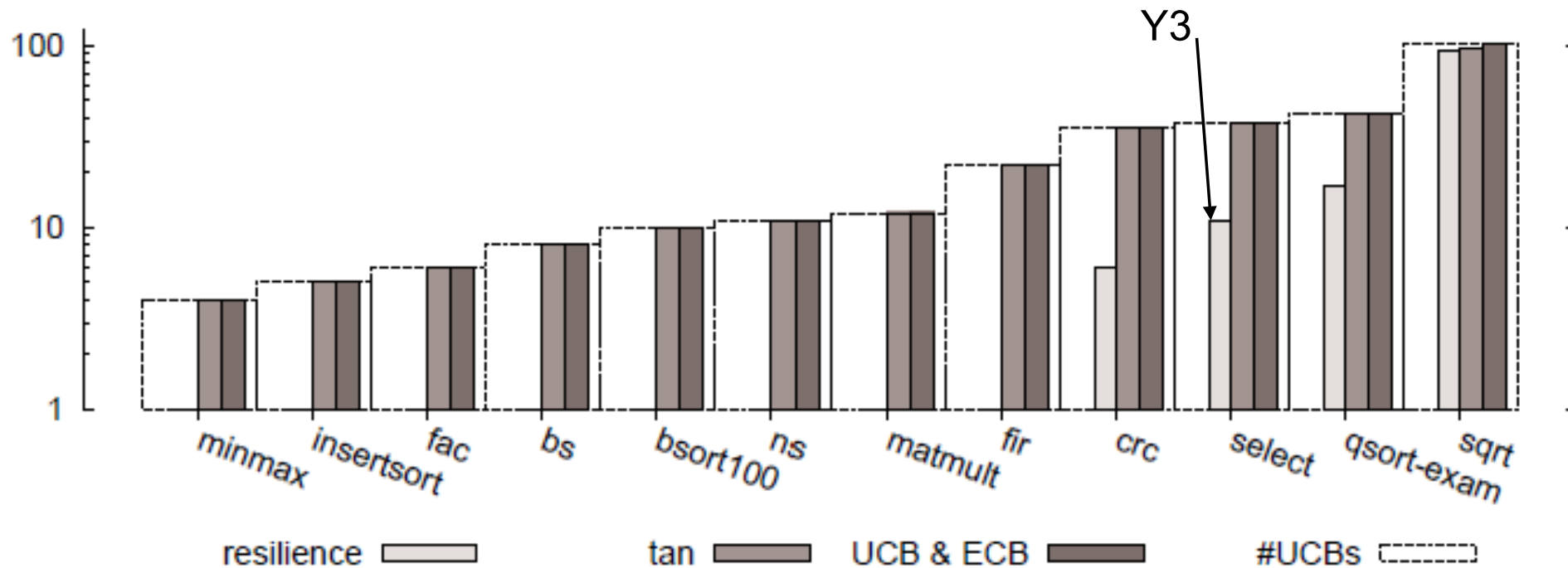
CRPD for set-associative LRU caches

- Cache blocks computation:
 - preempted task: Useful Cache Blocks (UCB)
 - preempting task: Evicting Cache Blocks (ECB)
- CRPD computation from **both** UCB **and** ECB:
 - Previous combination overestimates
 - Some UCBs remain useful under preemption
- **Resilience Analysis:**
 - How many evictions does a cache block survive?
 - Determines which UCBs remain useful
 - Reduces the CRPD bound (which improves predictability)

Evaluation

Test scenario: simple preemption of Task „qurt“ by the others tasks.

(from Mälardalen Benchmark suite)



(tan, UCB&ECB, #UCB show the bounds by former analyses)

Future use of resilience analysis: Multicore cache analysis

Lift the concept of resilience to **multicores**:

- Parallel tasks instead of preempting/preempted task
- To prove: which cache blocks remain cached even in case of interference by parallel tasks

Current CRPD-Integration in a3 Timing Analyzer:

- DC-UCB analysis implemented
- Available for Arm7, MPC603E and MPC55XX
- Status: Prototype for LRU cache replacement strategy (MPC5553 can be round-robin)
- Integration with RT_Druid and Symta/S via XTC extension

Conclusion

- Overall assesment
- Spreading excellence
- Plans for Y4

Overall Assessment and Vision at Y0+3

- Many collaborations working very well
 - WCC Compiler, MPSoC architecture analysis, Definition and assessment of “predictability”, Predictable implementation for Model-based Design,
 - New developments
 - Scheduling on Multicore Platforms
 - Time-predictive HW and SW implementations
 - Global Event for year 4
- 8 joint publications in Y3, several mutual visits and joint projects

To be improved:

- Wider transversal integration across levels of abstraction
 - E.g., Towards implementation of model-based design, predictable multicore architecture and operating system

Workshops

- **13th International Workshop on Software and Compilers for Embedded Systems (SCOPES 2010)**
Schloss Rheinfels, St. Goar, Germany – June 29-30, 2010
- **8th IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS 2010)**
Waidhofen/Ybbs, Austria, October 13-15, 2010
- **The 8th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS 2010)**
IST Austria, Klosterneuburg, Austria. 8-10 September 2010
- **Green and Smart Embedded System Technology: Infrastructures, Methods and Tools (GREEMBED 2010)**
Stockholm, Sweden, April 12, 2010

KeyNotes, Invited Talks, Tutorials

- **VMCAI 2010** (Reinhard Wilhelm)
- **DATE 2010** (Alberto Sangiovanni-Vincentelli)
- **CPSWEEK 2010** (Peter Marwedel)
- **CPSWEEK 2010** (Alberto Sangiovanni-Vincentelli)
- **SIES 2010** (Alberto Sangiovanni-Vincentelli)
- **SoCC 2010** (Alberto Sangiovanni-Vincentelli)
- **ICFEM 2010** (Wang Yi)
- **ISCA 2010 Tutorial** (Lothar Thiele & Reinhard Wilhelm)
- **ARTIST Summer School China 2010** (Several speakers)
- **ARTIST Summer School Europe 2010** (Several speakers)

Tools and Platforms

- **AiT**, the leading tool for computing WCETs [AbsInt, Dortmund, Saarland]
- **WCC**, the WCET aware compiler [AbsInt, Dortmund, Saarland]
- **MAST**, modeling and analysis suite for real-time applications [Cantabria]
- **MPA toolbox**, analysis of distributed embedded real-time systems, based on the real-time calculus [ETHZ]
- **MPARM**, virtual SoC platform, written in SystemC, to model system HW and SW [Bologna]
- **UPPAAL**, leading tool for precise automata-based analysis of timed systems [Uppsala, Aalborg]
- **PRET_C**, predictable multithreaded programming in C [INRIA, Auckland]

Plans for Y4

Continued transversal integration:

- Principles for definition of predictable multi-core architectures
- Focus on resource management for multicore platforms
- Extension of WCC to multi-task and multicore systems
- Continued work on Predictable HW and SW designs
- Fault-tolerance: predictable reliability

Joint event with Predator and Merasa: PPES'11 workshop
affiliated with DATE'11: *Bringing Theory to Practice: Predictability
and Performance in Embedded Systems*

Collaborative paper: State-of-the art on predictability