

From Single-Core to Multi-Core Platforms – Systematic Migration of Hard Real-Time Software

Ursula Kelling, Jens Harnisch (Infineon Technologies AG)

SYMTAVISION NewsConference
September 2010, Braunschweig



Never stop thinking

Outline

- **Drivers and use cases for automotive multi-core architectures**
- **Science to real world: Today**
- **Future controller architectures**
- **Practical challenges when moving to multi-core**
- **Development flow for multi-core software development**
- **Multi-core scheduling analysis with SymT/AS**
- **Summary and outlook**

Drivers and use cases for automotive multi-core architectures

- Drivers:
 - High-Integration: Functionality of two or more ECUs is combined on one ECU and hence on one microcontroller
 - Higher performance requirements: New control algorithms require more computational power; however, this can only be realized with multi-core approaches, and not with higher clock rate (e.g. due to EMC and thermal problems): clearly challenging for software development

- Use cases:
 - Power-train (Gas/Diesel Direct Injection, Hybrid Control, E-Motor Drive)
 - Body, chassis, safety



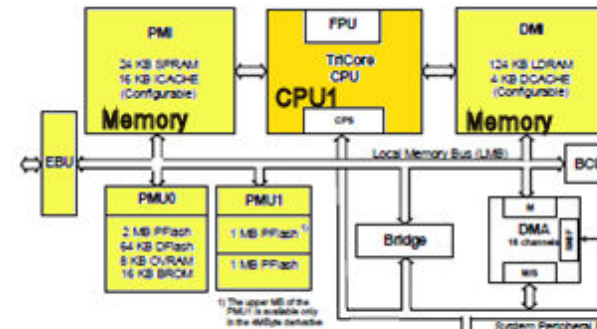
Note: Scientific notations are according to „Scheduling in Computer and Manufacturing Systems“ published by Błazewicz et al.

Science to real world: Today

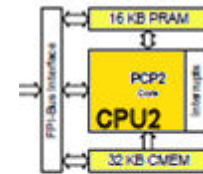
Science

- R2 |
- Two processors, having a different feature set. (unrelated processors)

Real World



Abbreviations:
 ICACHE: Instruction Cache
 DCACHE: Data Cache
 SDRAM: SDRAM
 LDRAM: Local Data RAM
 OVRAM: Overlay RAM
 BRAM: Boot ROM
 PFlash: Program Flash
 OFlash: Data Flash
 PRAM: Parameter RAM in PCP
 CMEM: Code RAM in PCP

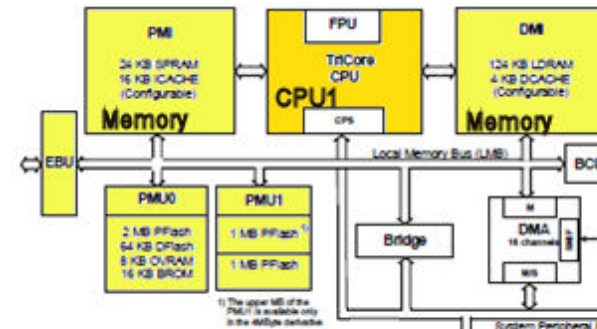


Science to real world: Today

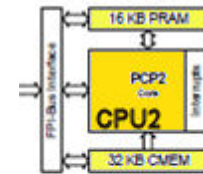
Science

- R2 | pmtn,
- Two processors, having a different feature set. (unrelated processors)
- Preemptions are allowed

Real World



Abbreviations:
 ICACHE: Instruction Cache
 DCACHE: Data Cache
 SDRAM: SDRAM
 LDRAM: Local Data RAM
 OVRAM: Overlay RAM
 BRAM: Boot ROM
 PFlash: Program Flash
 OFlash: Data Flash
 PRAM: Parameter RAM in PCP
 CMEM: Code RAM in PCP

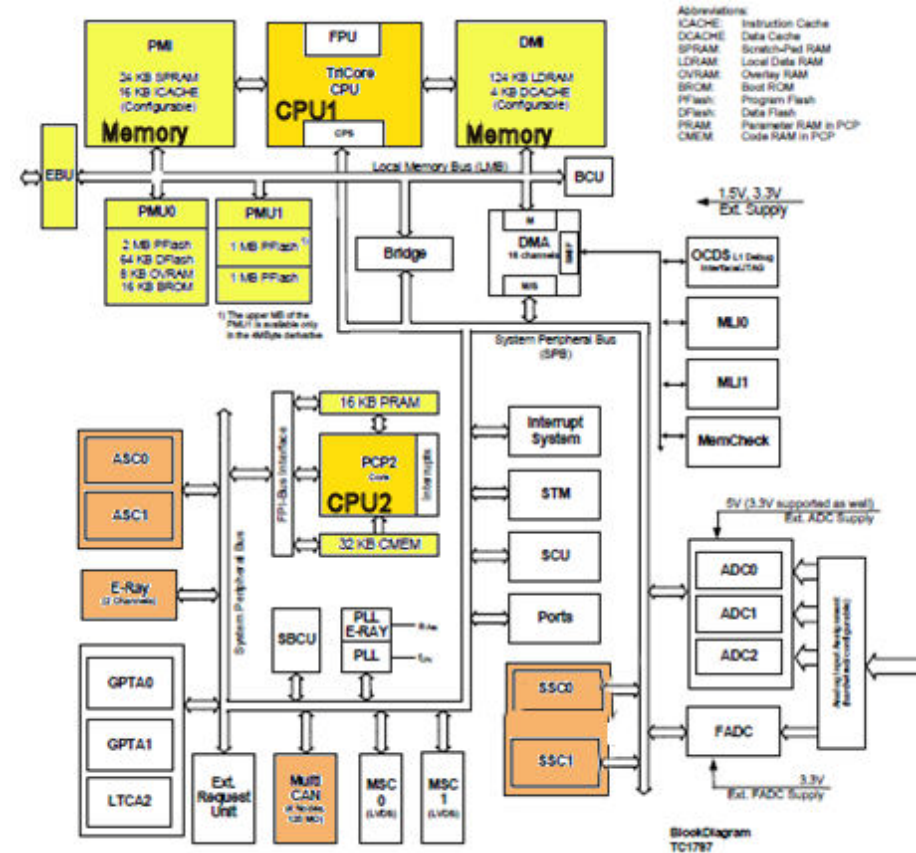


Science to real world: Today

Science

- R2 | pmtn, res,
- Two processors, having a different feature set. (unrelated processors)
- Preemptions are allowed
- Resources have to be managed, to avoid deadlocks

Real World

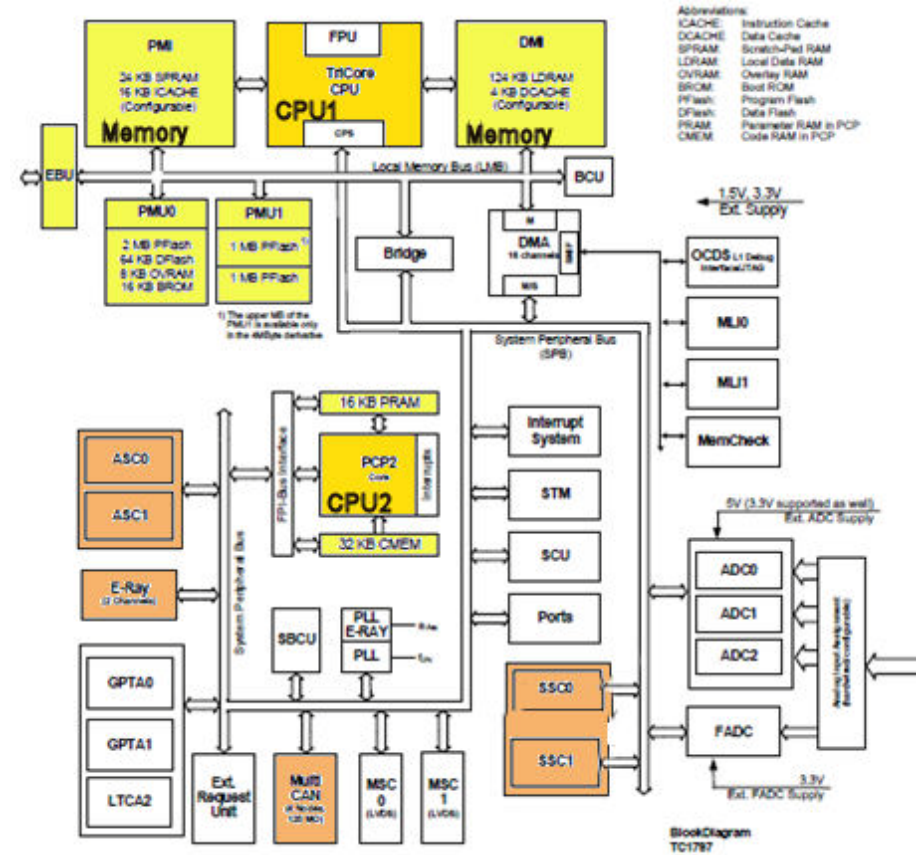


Science to real world: Today

Science

- R2 | pmtn, res, prec, r_j , (d_j) ,
- Two processors, having a different feature set. (unrelated processors)
- Preemptions are allowed
- Resources have to be managed, to avoid deadlocks
- Dependencies between tasks
- Release times
- Deadlines may exist

Real World

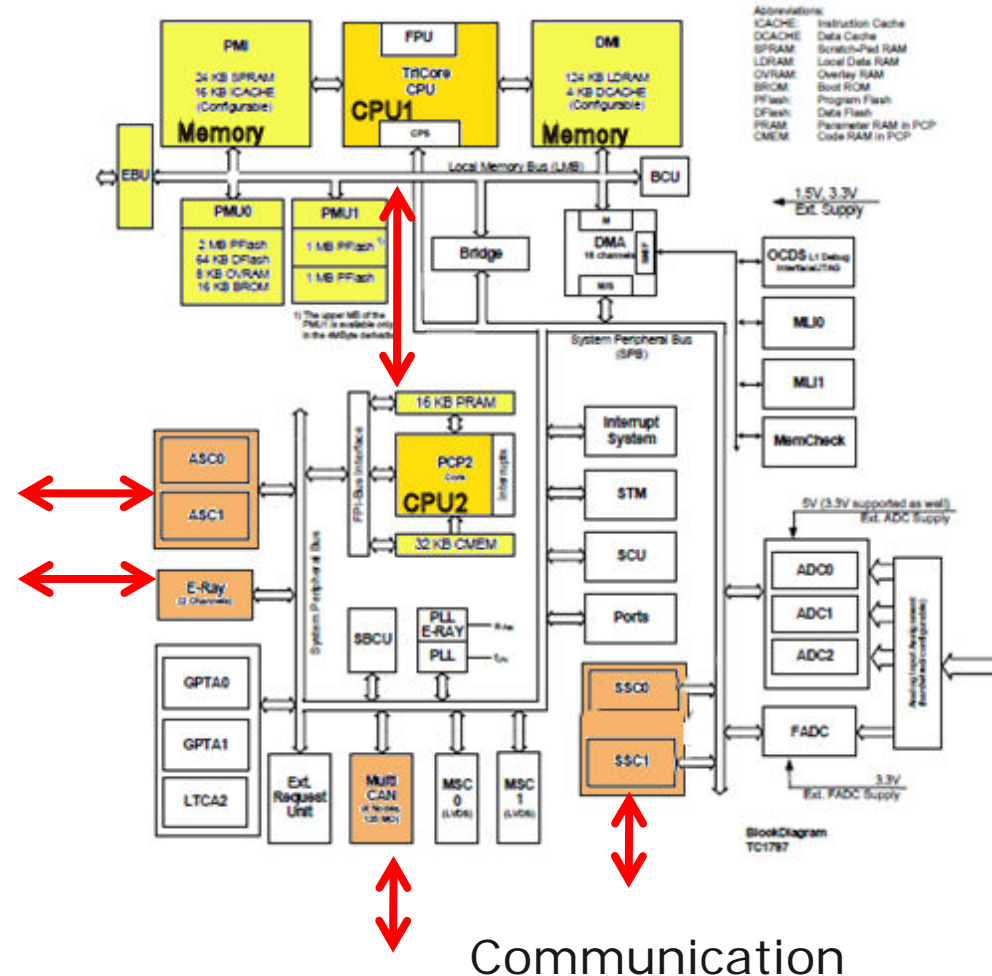


Science to real world: Today

Science

- R2 | pmtn, res, prec, r_j , (d_j) , c_j | C_{\max}
- Two processors, having a different feature set. (unrelated processors)
- Preemptions are allowed
- Resources have to be managed, to avoid deadlocks
- Dependencies between tasks
- Release times
- Deadlines may exist
- Communication times
- Optimization criterion is the maximum processing time, or it could be L_{\max} the lateness of tasks
- In any case, the problem is NP hard, so more than one sub-optimal solution exists

Real World



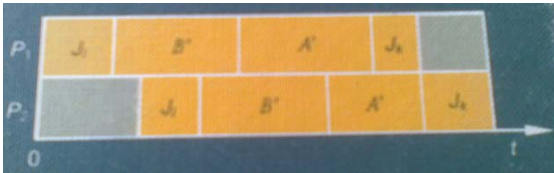
Communication

Runtime of a task

- Dependent on the location of a task, the runtime might change.
- Due to the fact, that RAM has less wait states than (un-cached) Flash, and Flash has less wait states than external memory, every task has memory dependent runtimes.
- Already today, accessing the local RAM of the CPU is faster than accessing the RAM of another CPU (communication time or wait states have to be added)

Growing complexity

- $Q_m \mid pmtn, res, prec, r_j, (d_j), c_j \mid C_{max}$ (or L_{max})
- Which means, that uniform processors will be put into the system (same capabilities, but different speeds)
- Additional local and "external" memories will exist.
- As $m = 1, \dots, n$ a help for scheduling tasks is necessary.
- Gantt charts are a tool to show schedules:

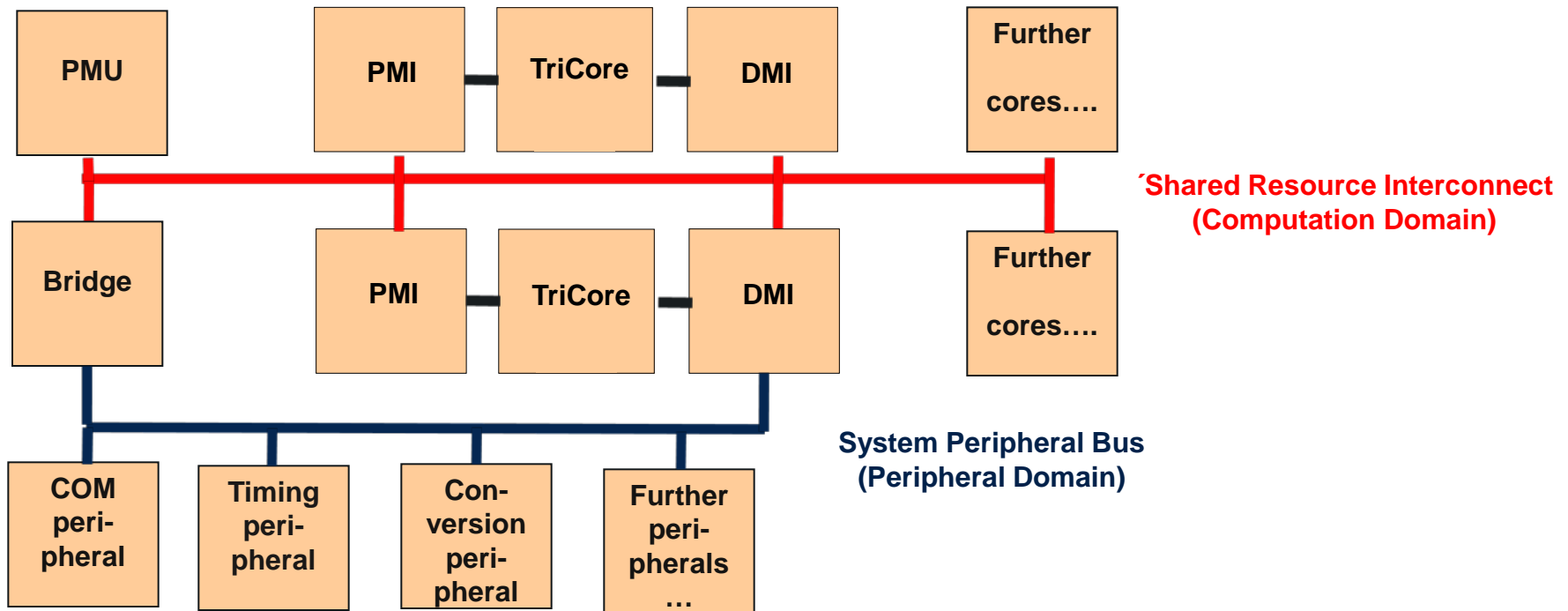


- Up to 6 tasks of course you could use these:



How to connect many cores on an automotive microcontroller?

- Attach all cores to a bus? Bandwidth limitation!
- Attach all cores to a cross bar? Does not scale to many cores!
- Use a network on chip? Missing real time capability!
- Use different domains (e.g. computation domain, communication domain) and cross bars within domains? May work!



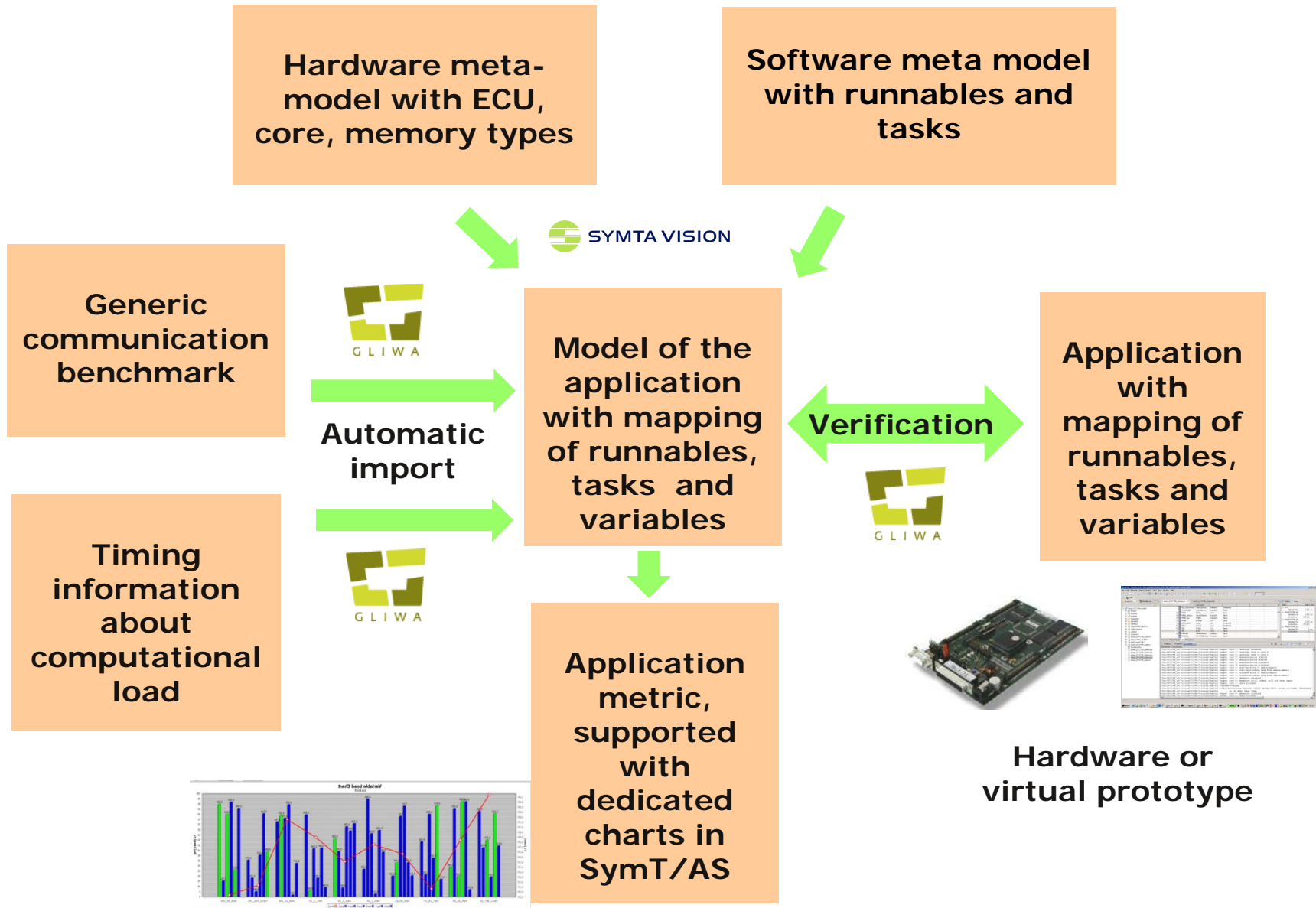
- **Scheduling analysis on chip will become more complicated, with possibly different execution times on cores and communication overhead.**

Practical challenges when moving to multi-core

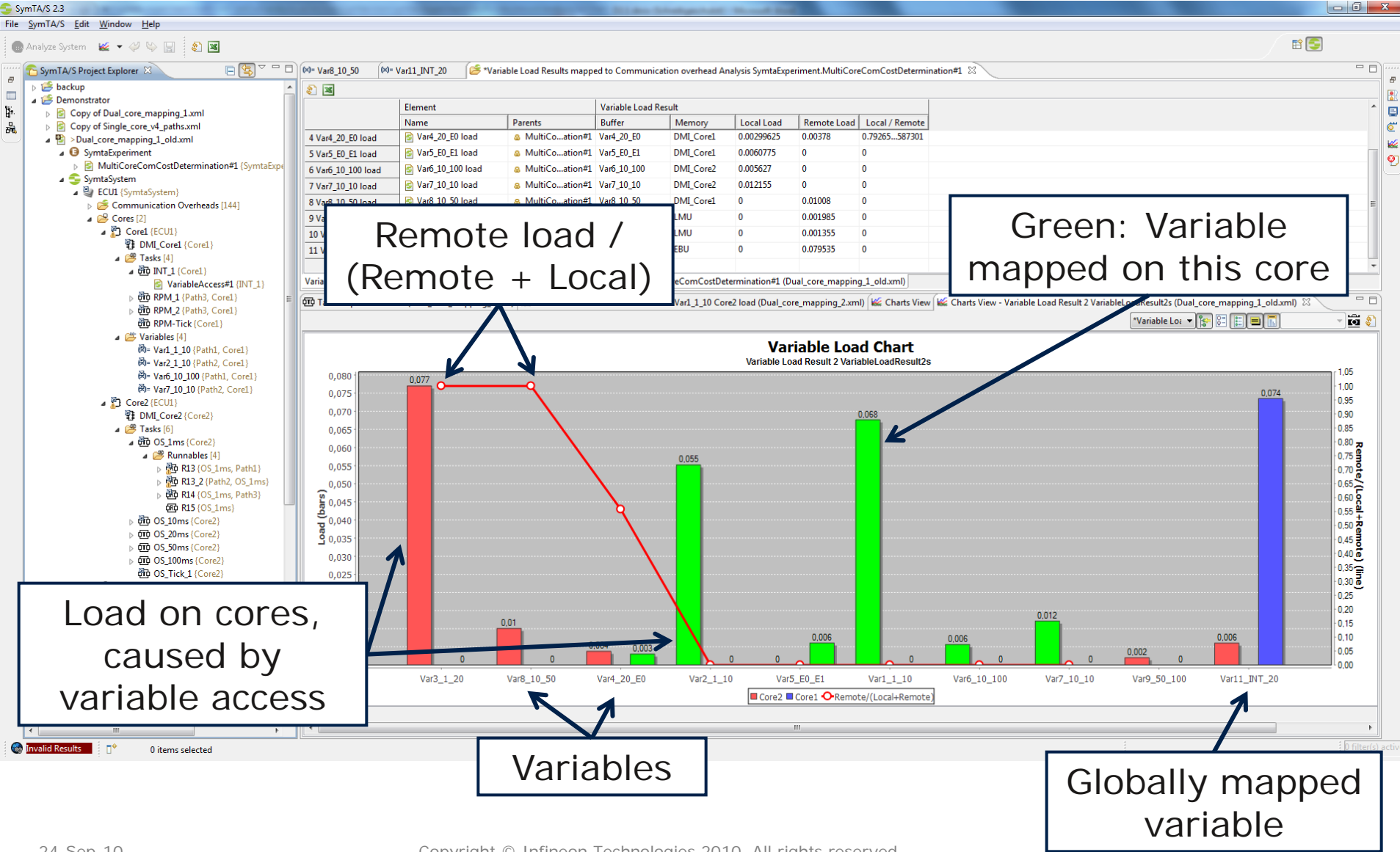


- New hardware requires new compiler and new OS version
- Architectures are not always directly comparable to predecessors due to different clock rates, connectivity and memory characteristics
- Software mapping to cores uses new communication primitives
- **Many changes at once! The practice:
Why is the performance not as expected?**
- **Solution for investigation:**
 - Use a model based approach
 - Use communication cost catalogue

Development flow for multi-core software development



Multi-core scheduling analysis with SymT/AS



Summary and outlook



- Moving to multi-core systems does not always make a transition in small steps possible, model based approaches and tool support gain importance
- The project developed a flow with consideration of communication and computational costs, to conduct system analysis on a model level, and investigate/analyze more efficiently than on the real target only
- Flow could be used in the future to automatically generate advice for better application mapping
- Please check the demonstrator!

Acknowledgements

- Christoph Ficek, Ingo Houben and Dr. Kai Richter (SYMTA VISION GmbH)
- Peter Gliwa and Nick Merriam (Gliwa GmbH)
- Gerd Punsmann, Harry Siebert and Arndt Voigtländer (Infineon Technologies AG)

Thank you for your attention!

ENERGY EFFICIENCY
COMMUNICATIONS
SECURITY

Innovative semiconductor solutions for energy efficiency, communications and security.

