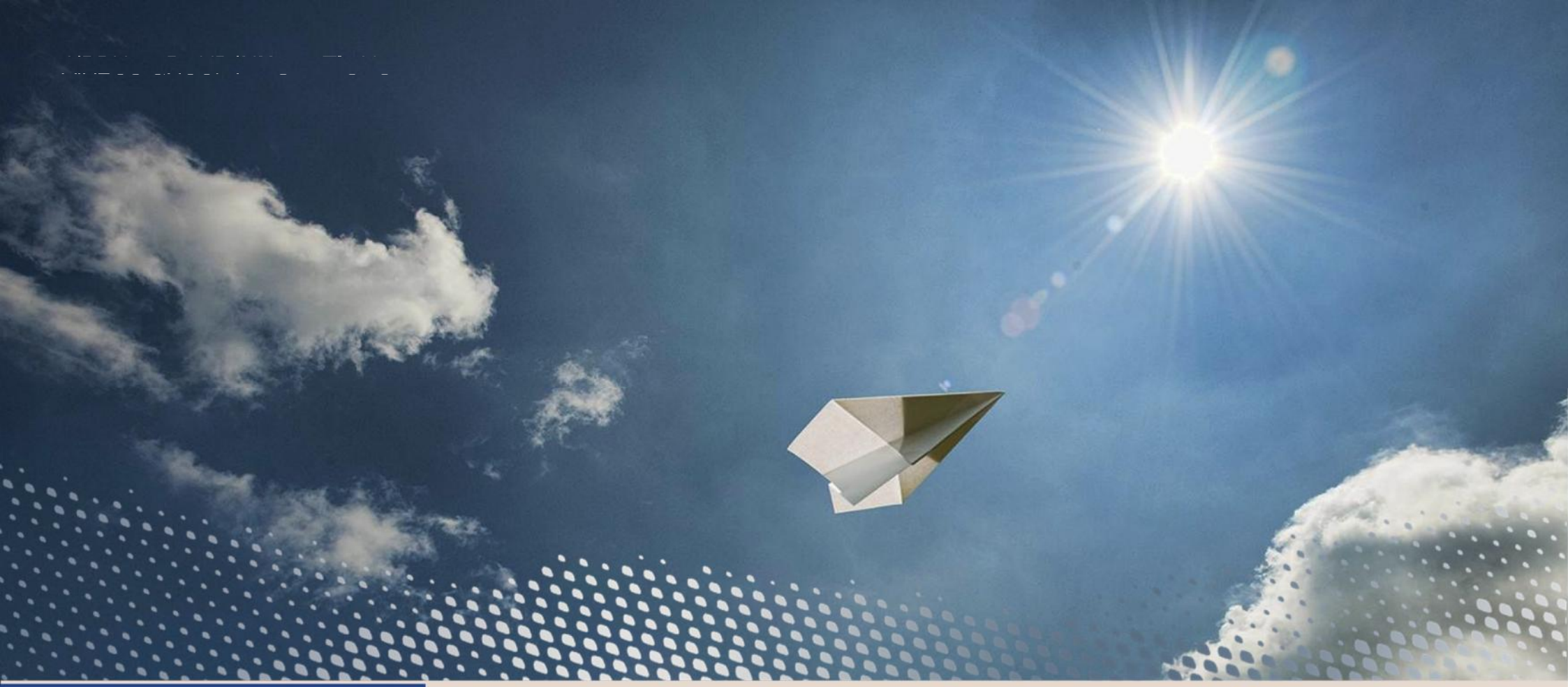


Challenges for the Use of Multicore Processors in Mixed-Criticality Systems with Focus on Temporal Aspects

Michael Paulitsch, RTAS 2014 – 16.4.2014

AIRBUS GROUP

Trends in Avionics

Trend towards new and additional IT-services and denser functional integration:

- Demand for new and additional IT-services
  on the aircraft itself and between the aircraft and the ground
  - Integrate formerly physically separated functions onto one platform
  - New failure modes and failures
  - New threats and vulnerabilities

Mixed-Criticality System
What is it?

Multiple safety criticalities (residing) on same platform

- Key requirement for platform: Simply defined, platform needs to fulfill safety requirements at minimum of highest safety requirement of application
- Chosen independence to minimize interaction between otherwise independent “chapters” (system level safety analysis extremely complicated.

What it is NOT

- A system where system approach sacrifices lower criticality applications for whatever purpose

Assurance in Aerospace – A Long Tradition of Safety

Civil Certification Standards (Large Airplane)

Part 21: Certification of Aircraft & Related Products, Parts & Appliances
CS 25: Certification Specifications for Large Aeroplanes
CS 25.1309: Equipment, Systems & Installations
AMC 25.1309: System Design & Analysis

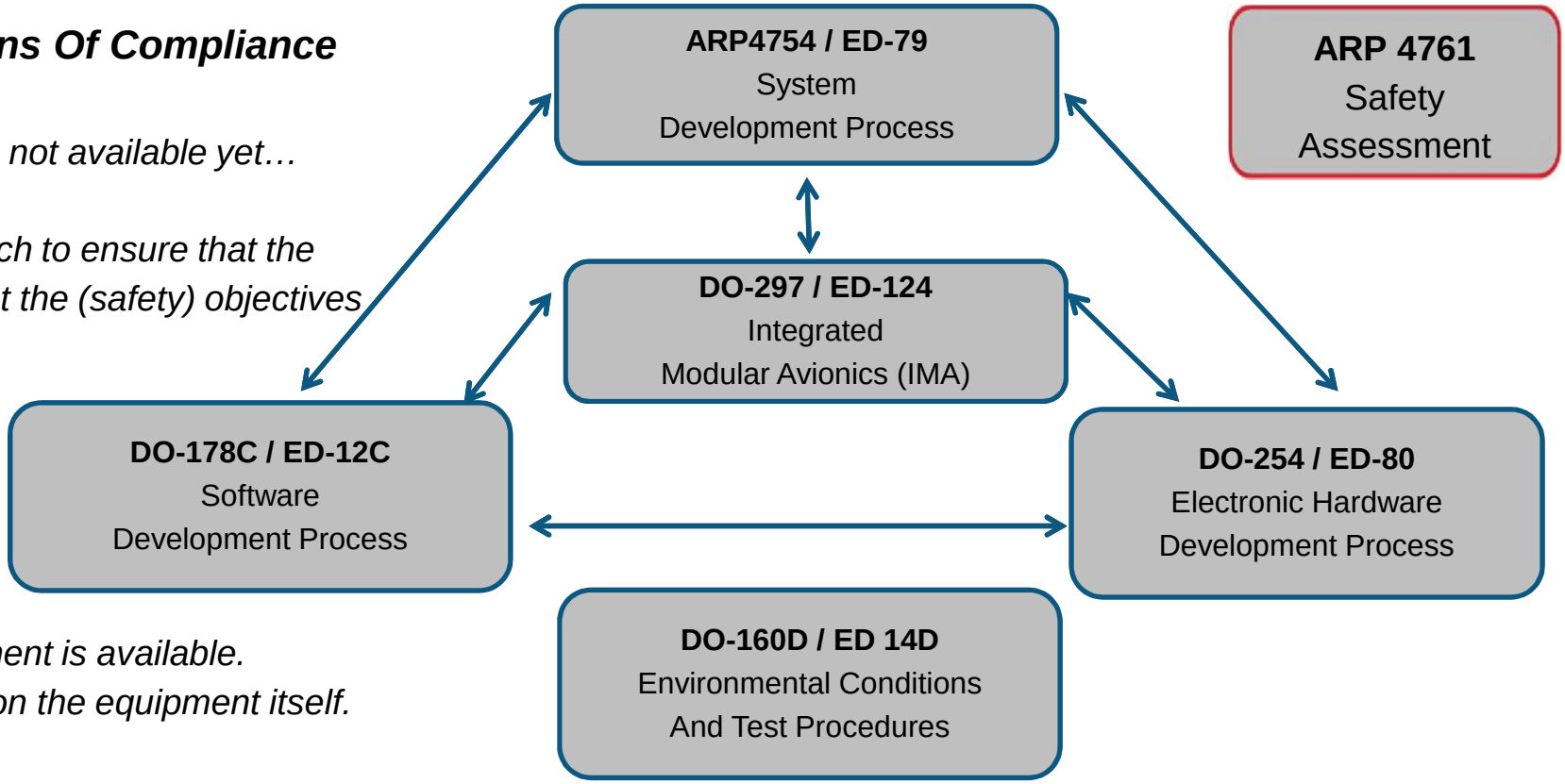
Airworthiness Standards

Set of requirements to ensure passengers' safety

Acceptable Means Of Compliance

The equipment is not available yet…

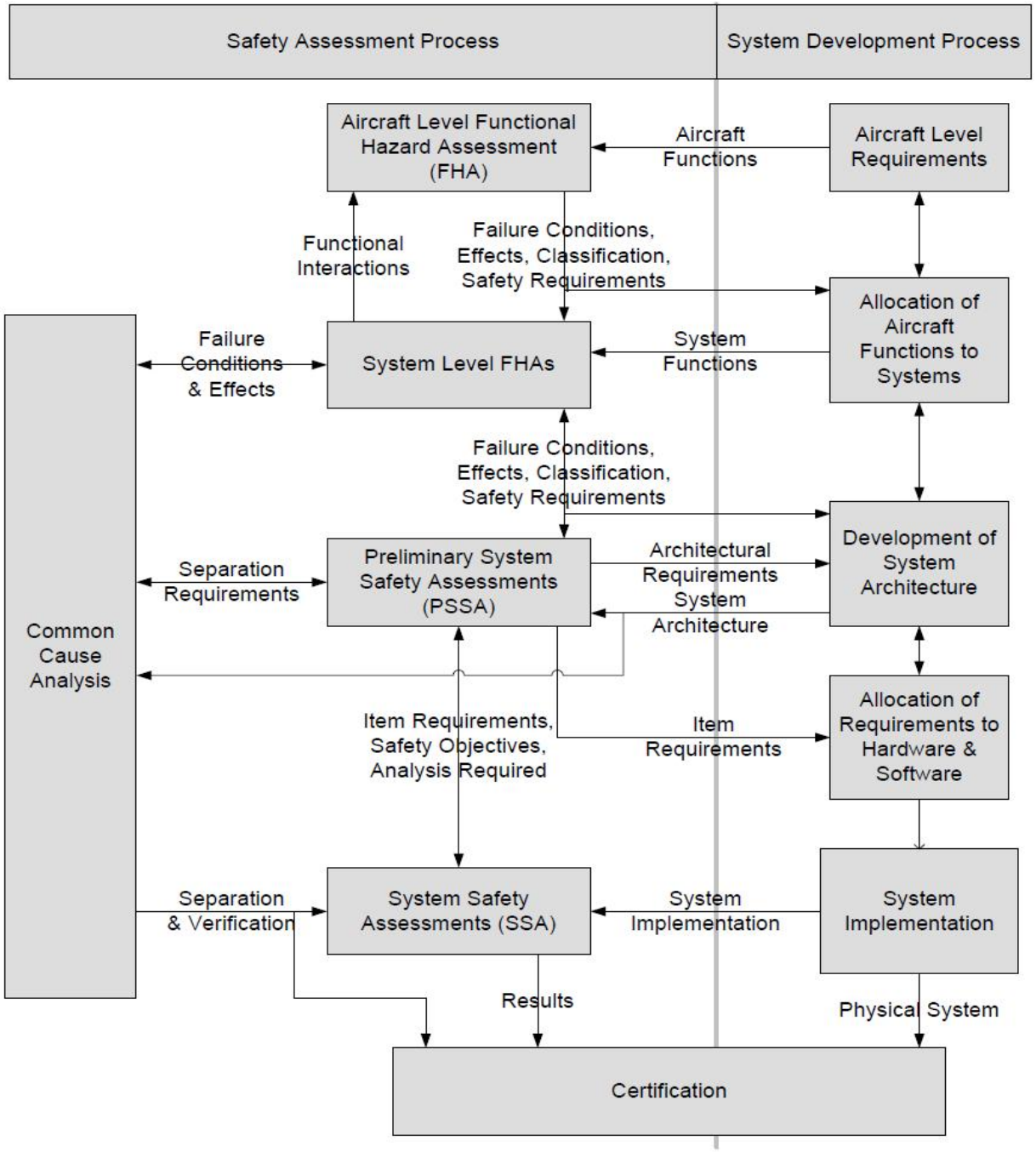
Structured approach to ensure that the equipment WILL meet the (safety) objectives



The equipment is available.
Tests are applied on the equipment itself.

Assurance in Aerospace –

Example ARP4754: System Development Process with strong safety focus



[Source: ED202, © ARINC]

View of Aerospace Certification Body Related to Timing

- **Currently under discussion** and concerns amongst others
  - **(Functional) interference channels**
  - **Shared resources** like memory / cache / interconnect / I/O / …
  - **Resource Usage** plan und its verification

AIRBUS GROUP

Multicore: General Possible Undesired Effects (Temporal)

Other possible undesired effects affecting temporal determinism

Details in paper
O. Kotaba, J. Nowotsch, M. Paulitsch, S. Petters, H. Theiling. Multicore In Real-Time Systems - Temporal Isolation Challenges Due To Shared Resources. WICERT workshop as part of DATE 2013.
Other overview paper:
D. Dasari, B. Akesson, V. Nelis, M.A. Awan, S.M. Petters. Identifying the Sources of Unpredictability in COTS-based Multicore Systems. SIES conf. 2013.

Shared resource	Mechanism
System bus	Contention by multiple cores Contention by other device - IO, DMA, etc. Contention by coherency mechanism traffic
Bridges	Contention by other connected busses
Memory bus and controller	Concurrent access
Memory (DRAM)	Interleaved access by multiple cores causes address set-up delay Delay by memory refresh
Shared cache	Cache line eviction Contention due to concurrent access Coherency: Read delayed due to invalidated entry Coherency: Delay due to contention by coherency mechanism read requested by lower level cache Coherency: Contention by coherency mechanism on this level
Local cache	Coherency: Read delayed due to invalidated entry Coherency: Contention by coherency mechanism read
TLBs	Coherency overhead
Addressable devices	Overhead of locking mechanism accessing the memory I/O Device state altered by other thread/application Interrupt routing overhead Contention on the addressable device - e.g. DMA, Interrupt controller, etc. Synchronous access of other bus by the addressable device (e.g. DMA)
Pipeline stages	Contention by parallel hyperthreads
Logical units	Contention by parallel applications
	Other platform-specific effects, e.g. BIOS Handlers, Automated task migration, Cache stashing, etc.

Assessment of Multi-Core Worst-Case Execution Behavior

Overview

(work with Cassidian in RECOMP)

Motivation:

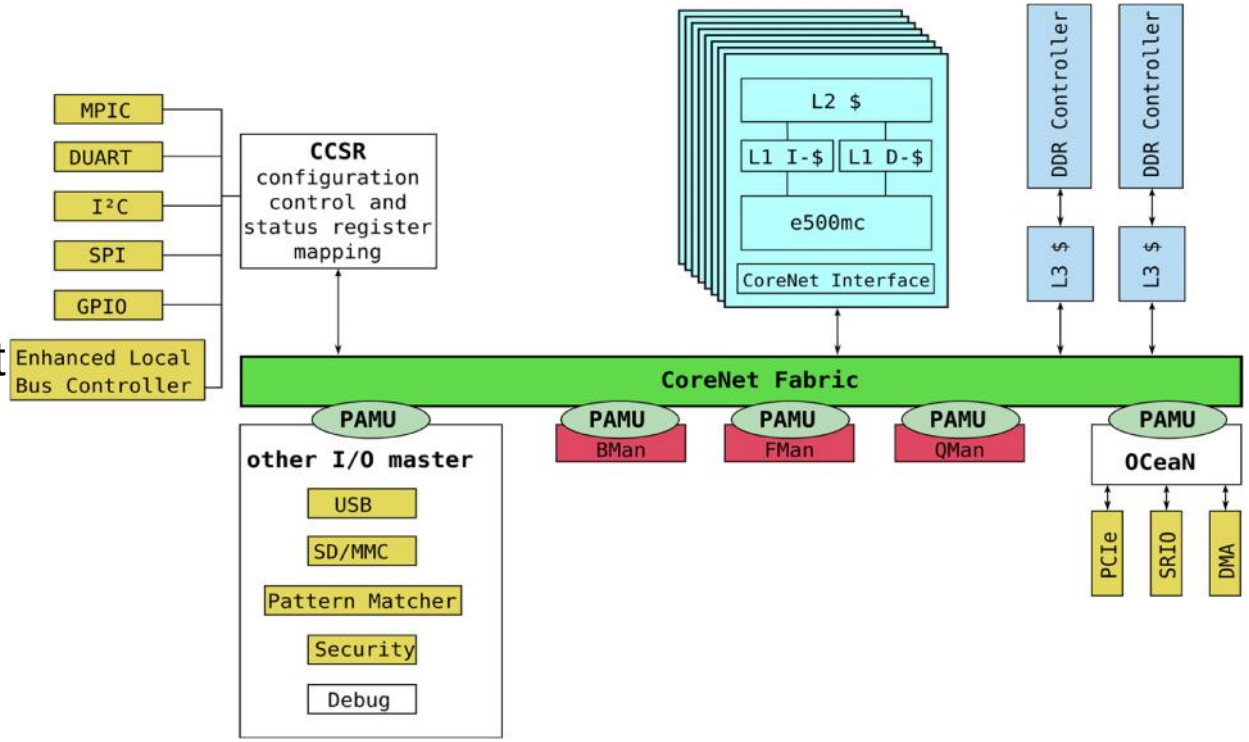
- Integration leads to common use of shared resources. Partitioning impact needs to be evaluated for safety-critical applications, such as IMA

Goal:

- Analysis of partitioning features of modern multicore computer in context of use in IMA
- Impact of integration on worst-case timing (WCET) of application

Approach

- memory-intensive tests



Freescale P4080

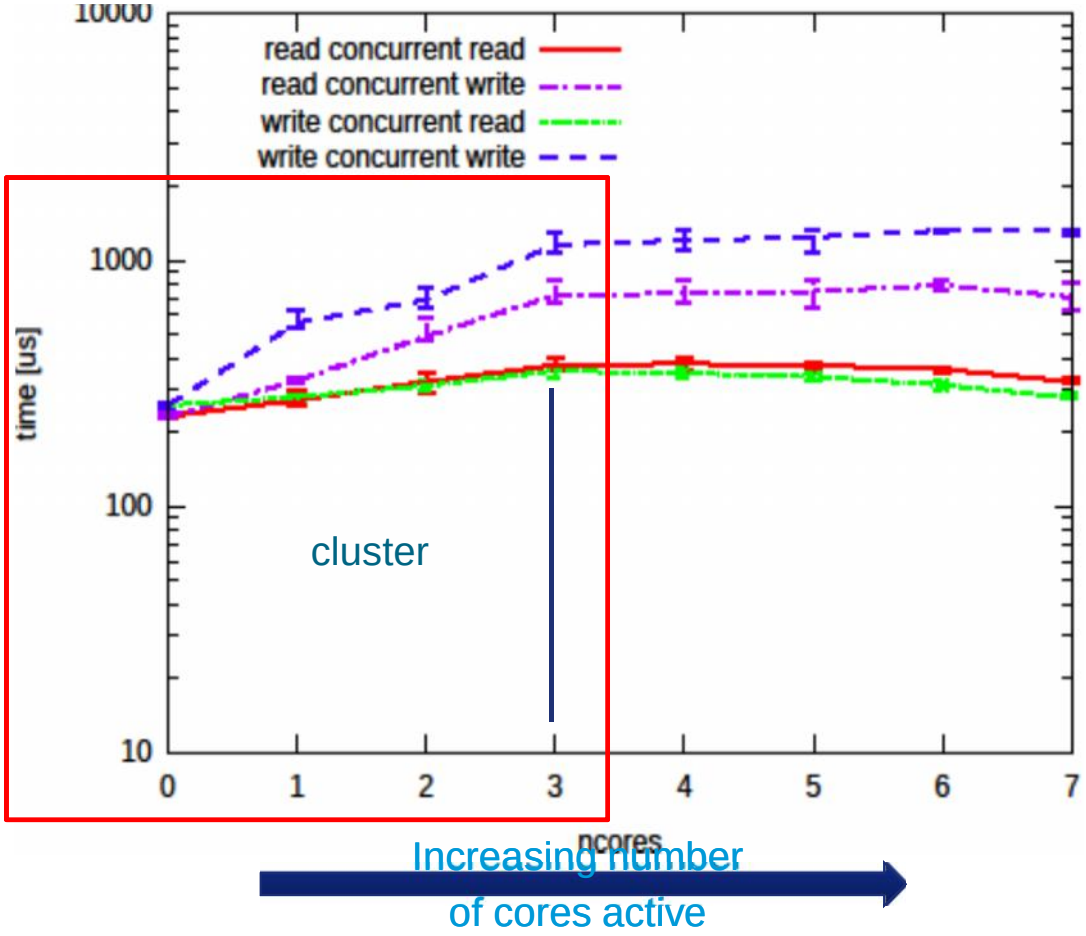
Focus of work:

- Network on Chip (not much data available); some memory access performance tests

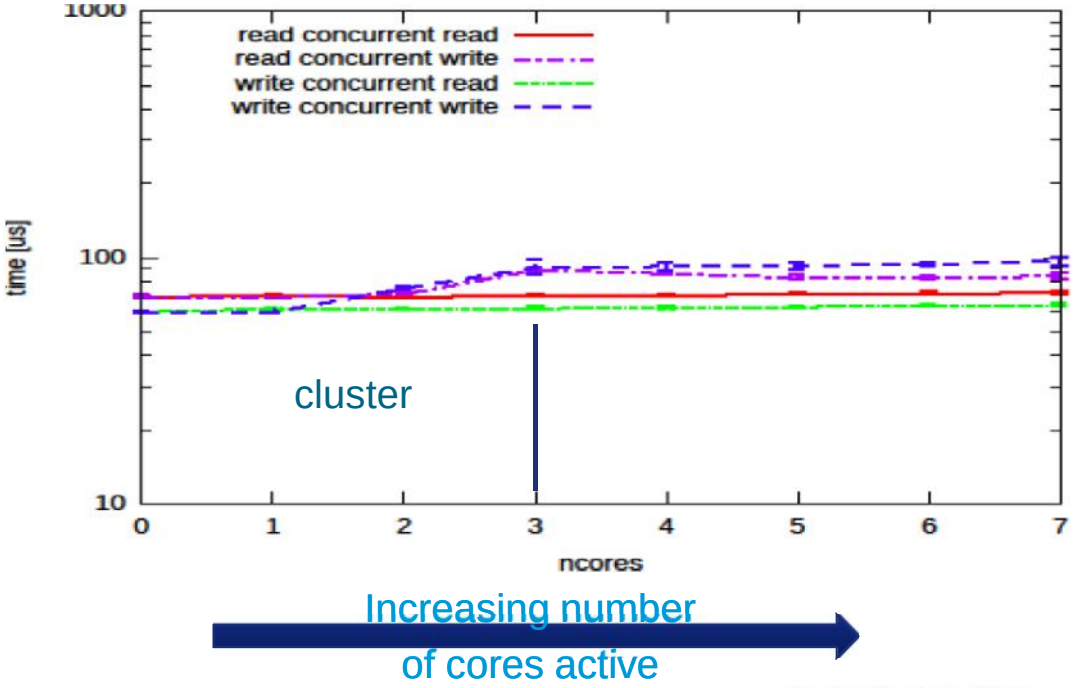
Details of work published at EDCC2012 (J. Nowotsch, M. Paulitsch)

Assessment of Multi-Core WCET
Results: Access DDR vs. SRAM on Freescale P4080

- Concurrency setup; Parameter: 4kB regions, 64B gap, 1 to 8 cores, coh. req. flag off
- Key take away: worst case access time increases over-proportionally with more cores



DDR – Results: saturation at 4 cores (1 cluster)



SRAM - Results: no influence of certain accesses

WCET for Multi-Core Computers
Problem Statement

Goal: deploy multi-core processors for safety-critical real-time applications (avionics, automotive, ...)

Problem: concurrent use of shared resources (e.g. interconnect, main memory)

→ unknown access latency for a concrete resource access

→ complicating timing analysis

Approach:

- Extend state-of-the art timing analysis to
  - Analyse the use of shared resources → compute upper bound
  - Compute interference delay based on timing and resource information
- Runtime monitoring to enforce resource usage bounds
- Increase average performance (response times, ...) using dynamic re-computation of resource usage bounds at runtime without violating static guarantees

Benefit:

- Robust execution framework for multicore processors
- Tooling extension for multicore processors

WCET for Multi-Core Computers Combined with Monitoring

Basic idea to benchmark/analyze hardware and include access interference and monitor memory accesses (RTNS 2013 paper, expected ECRTS 2014 paper)

-Extension of timing analysis

- Applied to AbsInt's aiT – commercial static WCET framework (extension memory accesses)

-Runtime Monitoring

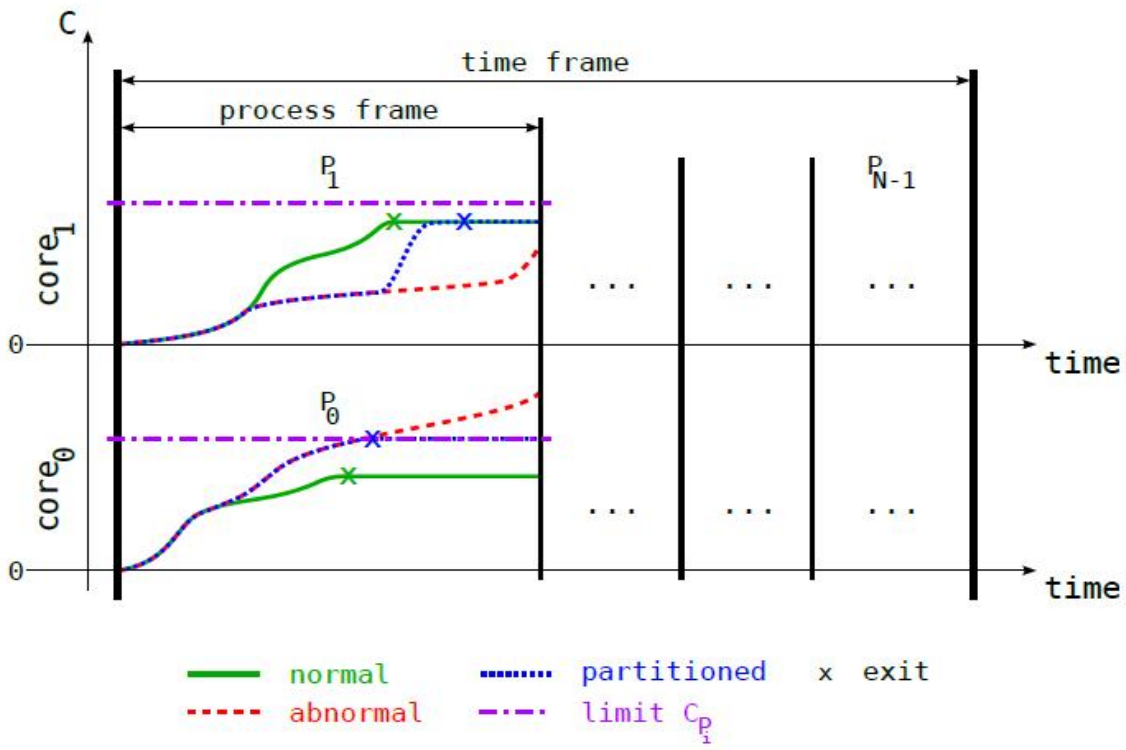
- Applied to bare-metal OS layer
- Applied to SYSGO's PikeOS

-**Average-Case Extension**

- Applied to bare-metal OS layer

-**Evaluation**

- Based on Freescale's P4080
- Benchmarks deduced from EEMBC Autobench benchmark suite
- WCET reduction:
  - Utilisation increase: core 98.9%, system 55%
  - Additional accesses: 2 to 70 times the accesses that were statically assigned



Implications on Research Needs

WCET:

- Optimizations for COTS devices (bank, cache, …)
  - Need to be careful about experimental conclusions: current multicore platforms are very complex and measured behavior is not easily explainable
- New WCET approaches for multicore (measurements, analytical, hybrid)
- Approaches to new monitoring considering different criticalities and different design integrity guarantees
- Consider I/O (as it is less feasible to control interference in COTS devices)

System Level (**Scheduling**):

- New approaches to scheduling (in real mixed-criticality systems)
  - Consider interference, COTS architecture (banks, cache hiercharchies, …)
- Consider “dynamic” behavior as “add-on”; improvements while guaranteed

Security: attacks on timing (destroying partitioning / virtualization)

Thank you!

Michael Paulitsch
Michael.Paulitsch @ airbus.com