Fliegende Computer

Sascha Uhrig, Airbus Central Research & Technology

Juli 2020

AIRBUS

A commercial aircraft manufacturer, **with Defence and Space as well as Helicopters Divisions**

Airbus is the largest aeronautics and space company in Europe and a worldwide leader.

Hersteller von Verkehrsflugzeugen mit zwei Divisionen

Airbus Defence and Space und Airbus Helicopters

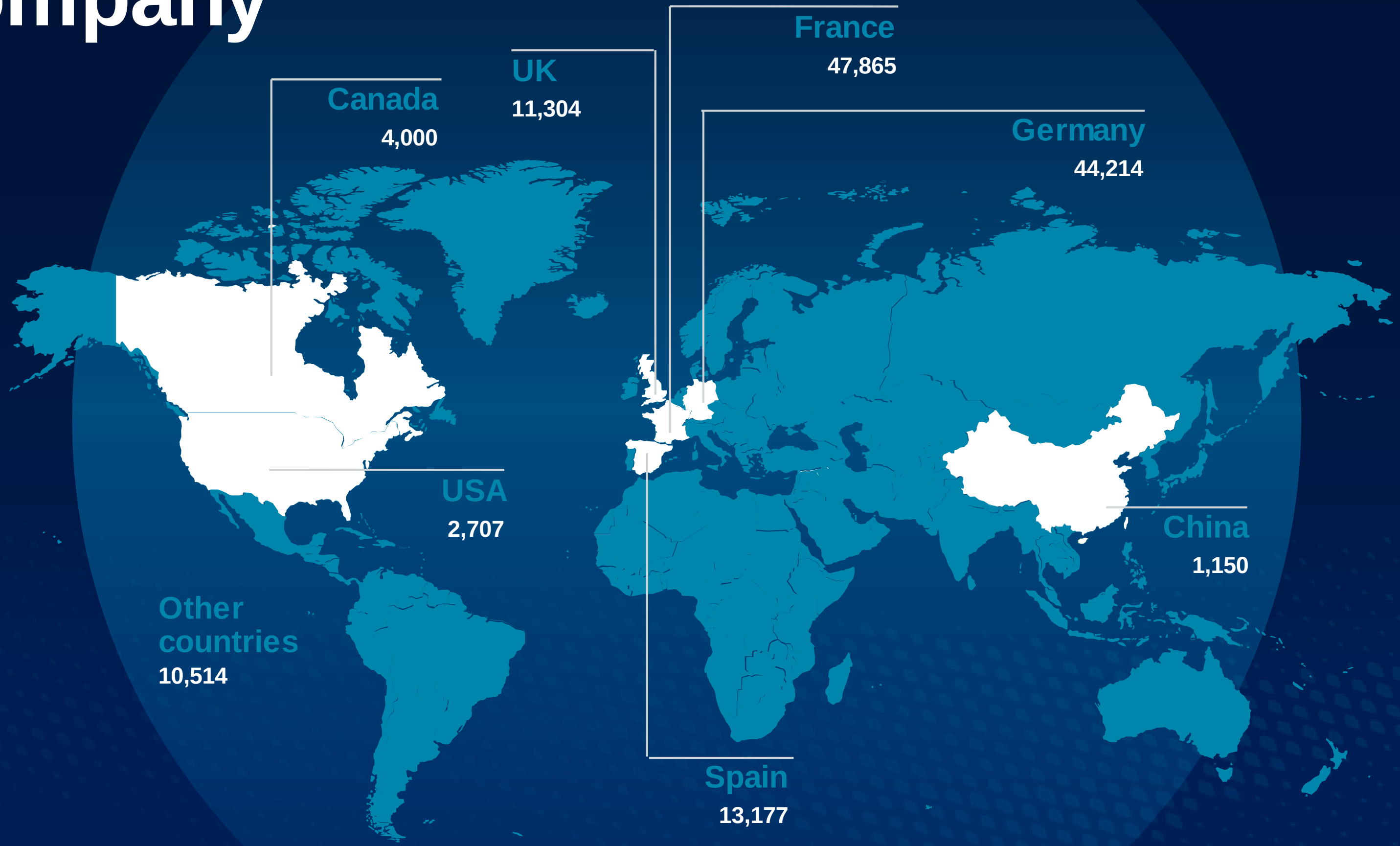
134.931
Mitarbeiter

€ 471 Milliarden
Auftragsbestand

€ 70,5 Milliarden
Jahresumsatz, angepasst gemäß IFRS 15

AIRBUS

Global company



134,931 employees
from 142 nationalities

Located across **35** countries
on more than 180 sites

31% Europe
69% Non-Europe
International Sales

AIRBUS

Vielseitig und zukunftsweisend: die Airbus-Flugzeugfamilien

Single-aisle Family

A321neo

A320neo

A319neo

A220-300

A220-100

Widebody Family

A350-1000

A350-900

A330-900

A330-800

Civil range

Intermediate Single

H125

H130

Light Twin

H135

H145

Medium

H155

H160

Super Medium

H175

Medium Heavy

H215

H225

Military range

Light

H125M

H135M

H145M

Medium

AS565 MBe

H160M

Heavy

H215M

H225M

Specialised

Tiger

NH90

HELICOPTERS

Technology demonstrators

VSR700

Racer

CityAirbus

DEFENCE AND SPACE

Airbus is Europe’s largest and most innovative defence and space company

We develop and manufacture world-class aerospace products. Our exceptional platforms and services allow our customers to address even their most challenging operational needs.

Strong, innovative and customer focused – Our portfolio

Military Aircraft

- A400M
- A330 Multi-Role Tanker Transport
- Special Mission Aircraft
- Combat Aircraft
- Full In-Service Support

Space Systems

- Telecommunication Satellites
- Earth Observation Satellites
- Navigation Satellites
- Orbital and Space Exploration Infrastructure
- Science Missions
- ArianeGroup (Launchers)

Communications, Intelligence and Security

- Intelligence
- CyberSecurity
- Security Solutions
- Secure Communications
- Secure Land Communications
- Future Applications

Unmanned Aerial Systems

- UAS and UAV solutions for airborne intelligence, surveillance and reconnaissance, and combat missions

ACJ – Your world above the world

Airbus Corporate Jet

Erstklassig
und entspannt reisen

200+
In Betrieb

Inkrementelle Innovation

Neue Produktlinien oder NEO-Versionen

Demonstrator-basierte Forschung, 3,2 Mrd. € eigenfinanzierte F&E im Jahr 2018

Wegbereiter neuer Technologien

Nachhaltigkeit

Autonomie

Elektrifizierung

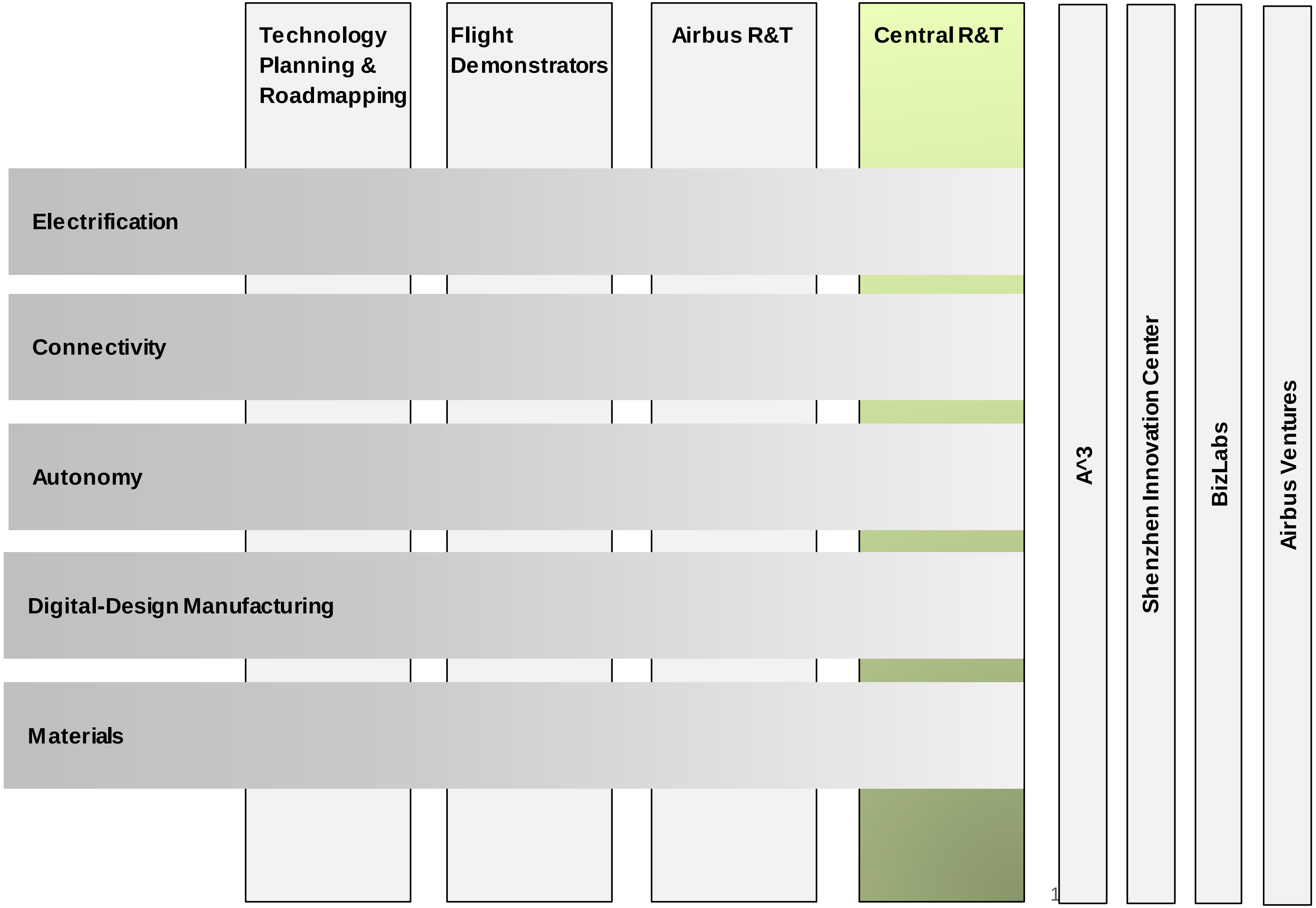
Konnektivität

Fortschrittliche Werkstoffe

Künstliche Intelligenz

Industrielle Systeme

Airbus Research & Technology Landscape



Technology Planning & Roadmapping

Flight Demonstrators

Airbus R&T

Central R&T

A^3

Shenzhen Innovation Center

BizLabs

Airbus Ventures

Electrification

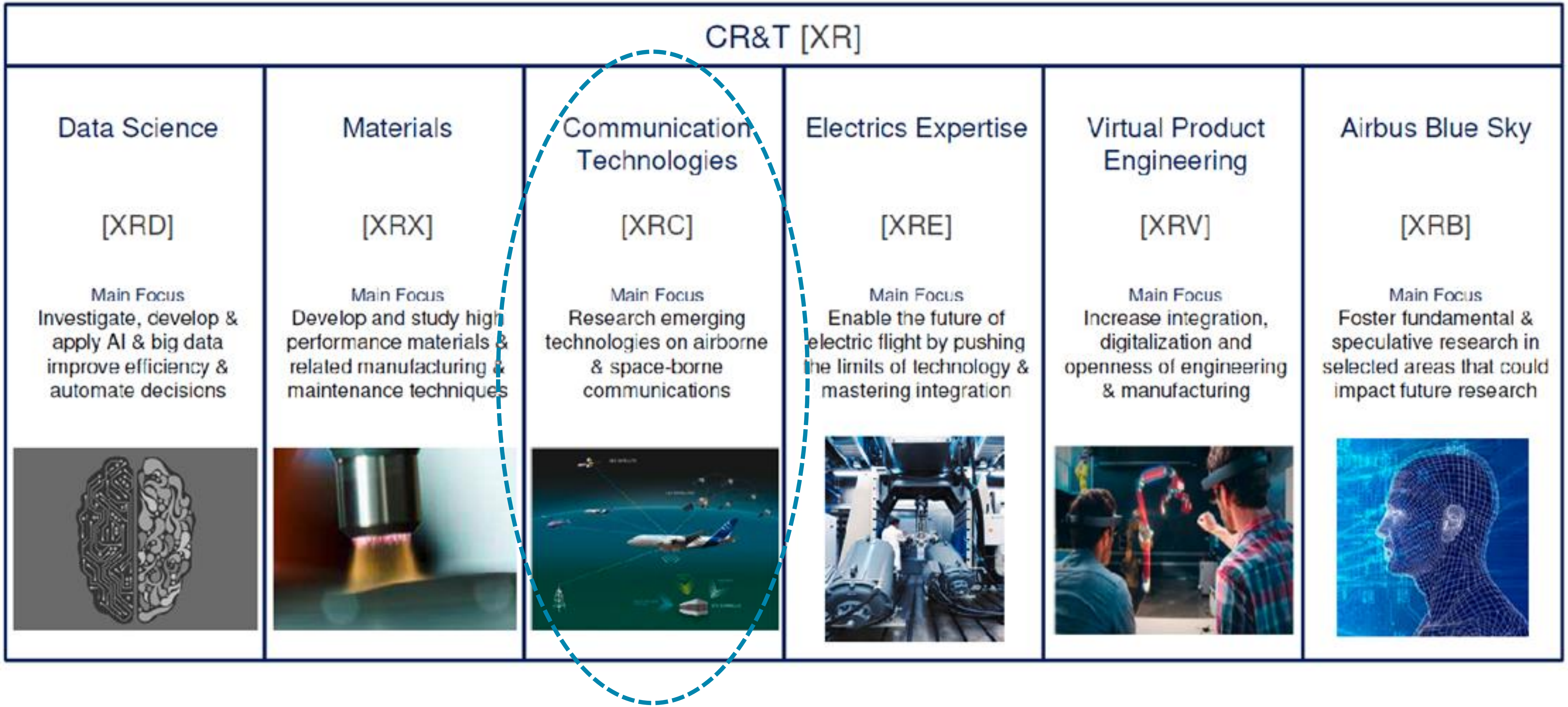
Connectivity

Autonomy

Digital-Design Manufacturing

Materials

CR&T Organisation



CR&T [XR]

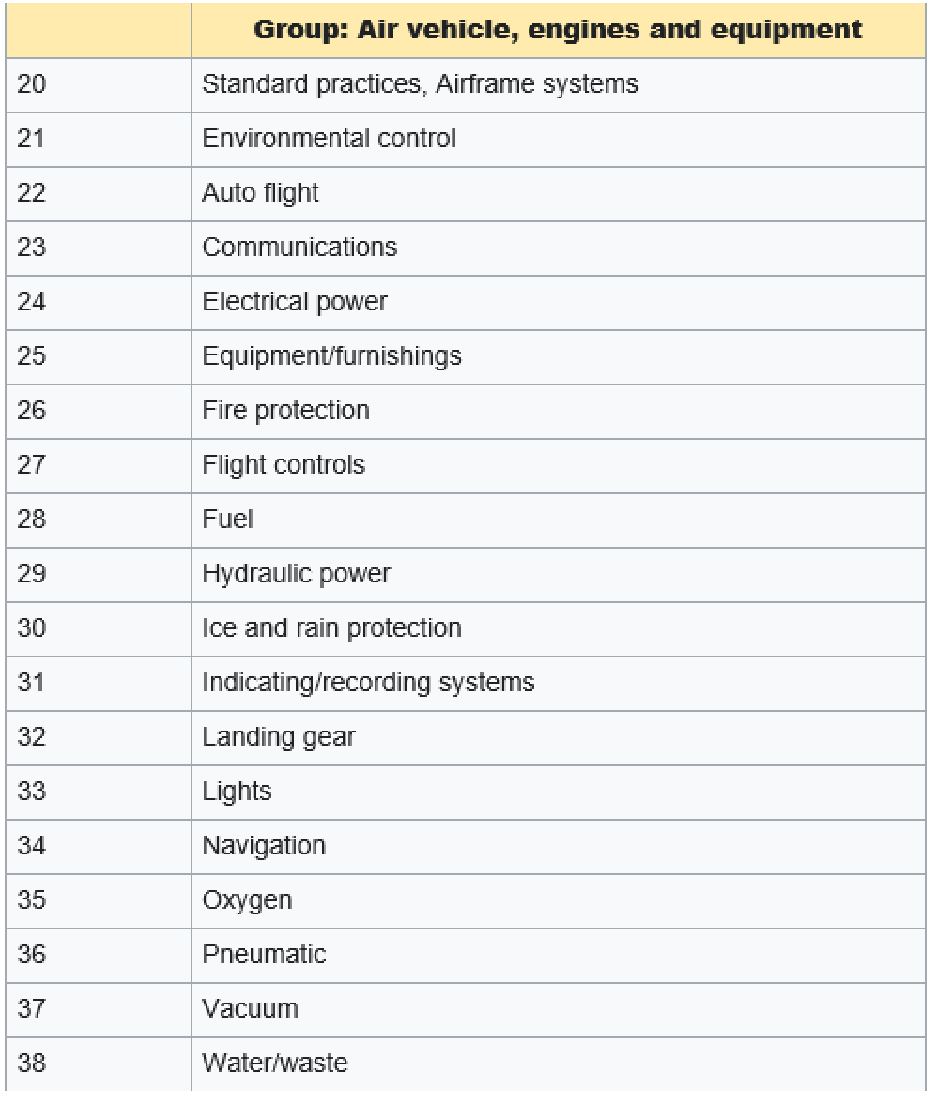
Data Science	Materials	Communication Technologies	Electrics Expertise	Virtual Product Engineering	Airbus Blue Sky
[XRD]	[XRX]	[XRC]	[XRE]	[XRV]	[XRB]
Main Focus Investigate, develop & apply AI & big data improve efficiency & automate decisions	Main Focus Develop and study high performance materials & related manufacturing & maintenance techniques	Main Focus Research emerging technologies on airborne & space-borne communications	Main Focus Enable the future of electric flight by pushing the limits of technology & mastering integration	Main Focus Increase integration, digitalization and openness of engineering & manufacturing	Main Focus Foster fundamental & speculative research in selected areas that could impact future research

Airbus CRT

Communication Technologies

Fly-by-Wire-Steuerung

General: ATA Chapters



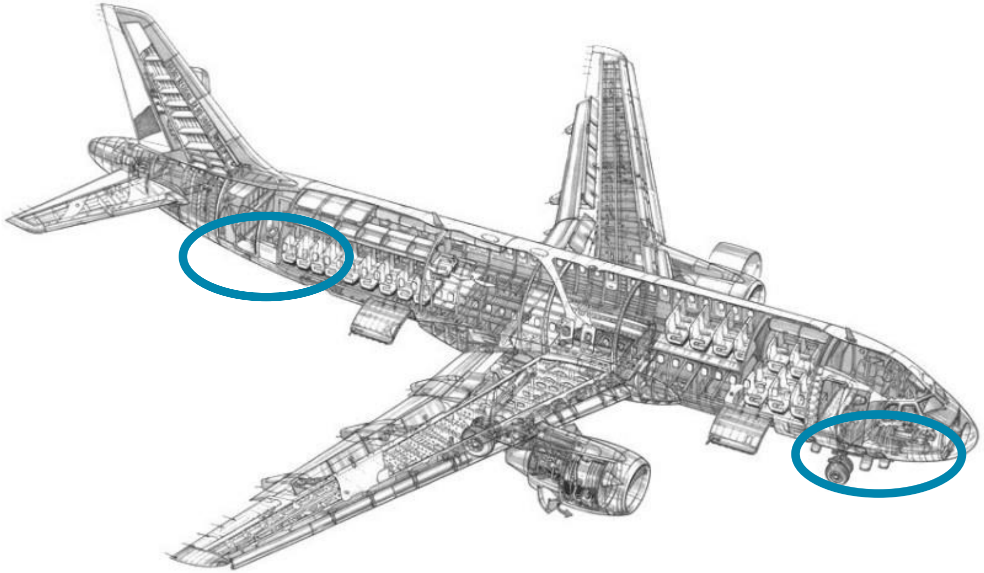
	Group: Air vehicle, engines and equipment
20	Standard practices, Airframe systems
21	Environmental control
22	Auto flight
23	Communications
24	Electrical power
25	Equipment/furnishings
26	Fire protection
27	Flight controls
28	Fuel
29	Hydraulic power
30	Ice and rain protection
31	Indicating/recording systems
32	Landing gear
33	Lights
34	Navigation
35	Oxygen
36	Pneumatic
37	Vacuum
38	Water/waste

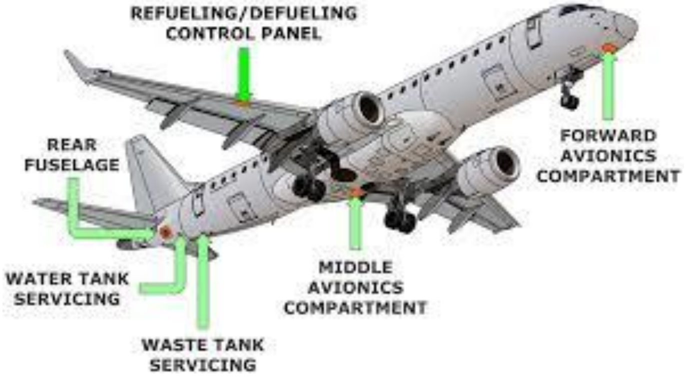
ATA = Air Transportation Association

…99 chapters

Avionics Bay

Avionics Bay A350




REFUELING/DEFUELING CONTROL PANEL
REAR FUSELAGE
FORWARD AVIONICS COMPARTMENT
WATER TANK SERVICING
MIDDLE AVIONICS COMPARTMENT
WASTE TANK SERVICING

Certification…

- …starts at the very first step of development by contacting certification authority (CA, e.g. EASA, FAA):
  I plan development of a new A/C
- CA:
  Ok, how do you plan to make sure it is safe?
- Avionic industry:
  I am applying standards A, B, C as part of good engineering practice (= certification plan)
- CA:
  Please start doing so, we will have regular reviews…

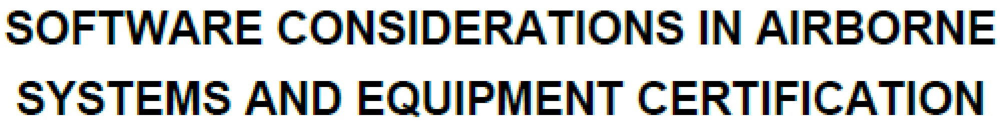
Certification plan and applied standards are mainly frozen at start of development!

Design Assurance Level (DAL)

- Level of design assurance ☺
- Means to what extend and what kind of “quality management” is used (“QM” is misleading since ordinary QM is on the lower end of DAL)
- DAL is assigned to every *item* before it is developed
- DAL assignment is a very complex process based on
  - A/C functions
  - Function criticality
  - A/C architecture
- DAL defines how an item is developed (and maintained)

Software Development

- Software development in avionics is mainly based on DO-178 (ED 12)
  - Targeting requirements engineering, development process, life cycle, configuration management etc.



SOFTWARE CONSIDERATIONS IN AIRBORNE
SYSTEMS AND EQUIPMENT CERTIFICATION

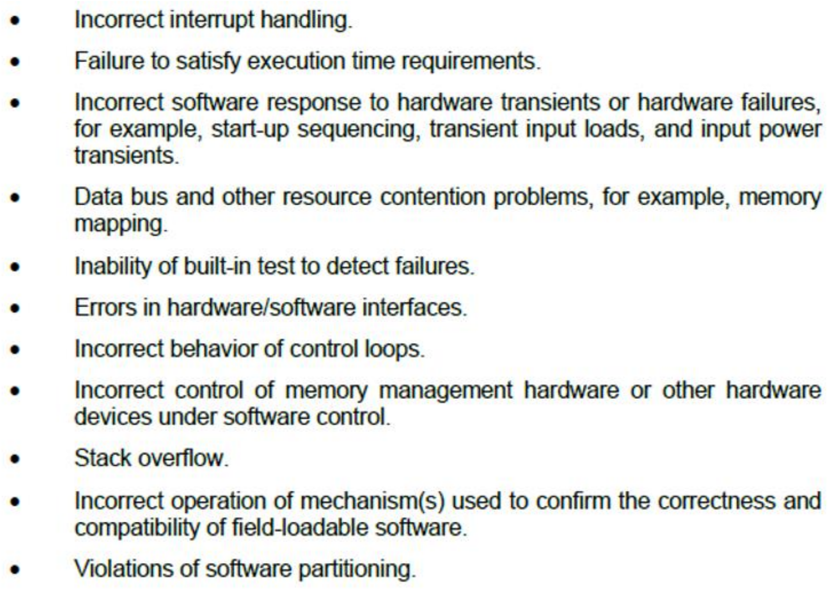
- Use of tools is based on ED 215
  - Rational: software development only valid if tools are of high quality

SOFTWARE TOOL QUALIFICATION CONSIDERATIONS

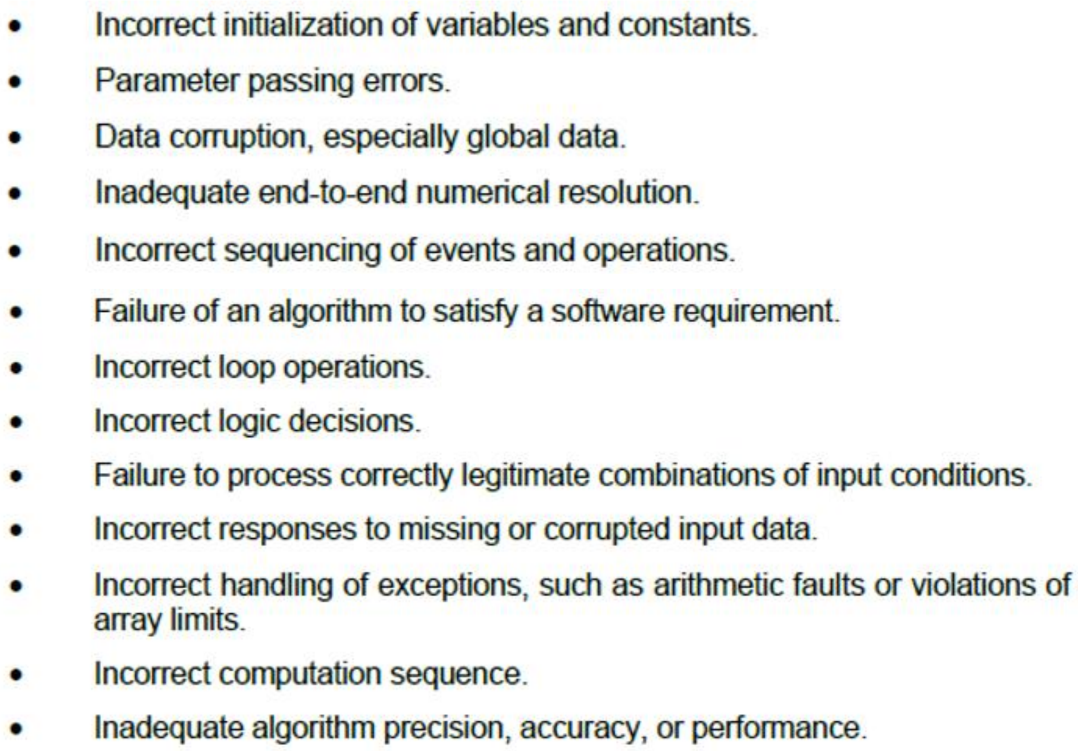
Software Development

- DO-178 addresses (not complete)
    - System requirements
    - Information flow between stakeholders
    - Safety assessment
    - Architectural aspects
    - System/software life cycles process
    - Software planning and development processes
    - Software quality assurance
    - Certification and liaison process

Software Testing - Examples



- Incorrect interrupt handling.
- Failure to satisfy execution time requirements.
- Incorrect software response to hardware transients or hardware failures, for example, start-up sequencing, transient input loads, and input power transients.
- Data bus and other resource contention problems, for example, memory mapping.
- Inability of built-in test to detect failures.
- Errors in hardware/software interfaces.
- Incorrect behavior of control loops.
- Incorrect control of memory management hardware or other hardware devices under software control.
- Stack overflow.
- Incorrect operation of mechanism(s) used to confirm the correctness and compatibility of field-loadable software.
- Violations of software partitioning.



- Incorrect initialization of variables and constants.
- Parameter passing errors.
- Data corruption, especially global data.
- Inadequate end-to-end numerical resolution.
- Incorrect sequencing of events and operations.
- Failure of an algorithm to satisfy a software requirement.
- Incorrect loop operations.
- Incorrect logic decisions.
- Failure to process correctly legitimate combinations of input conditions.
- Incorrect responses to missing or corrupted input data.
- Incorrect handling of exceptions, such as arithmetic faults or violations of array limits.
- Incorrect computation sequence.
- Inadequate algorithm precision, accuracy, or performance.

Special Challenge on Multicores

- DO-178 describes how to develop software and which steps to go for sufficient quality
- Multicores are seen as completely different (more complex) systems than single-cores.
  - ➔There is no official statement/*advice* how to deal with multicores
  - ➔There are no multicore systems available, yet
  - ➔There are no experiences…
- Multicore systems are not covered by any standard (so far)!
- "inofficial" guideline given in CAST32a position paper
  - ➔Parallel software explicitly **excluded**

Special Challenge on Multicores

- DO-178 descri … for sufficient
- Multicores … ns than single
  - ➔There is … es
  - ➔There … mul
  - ➔There are no experiences…
- Multicore systems are not covered by any standard (so far)!
- “inofficial” guideline given in CAST32a position paper
  - ➔Parallel software explicitly **excluded**

No experiences…not really true:

There is a single (non-Airbus) Dualcore flying...
by accident

Some Remarks

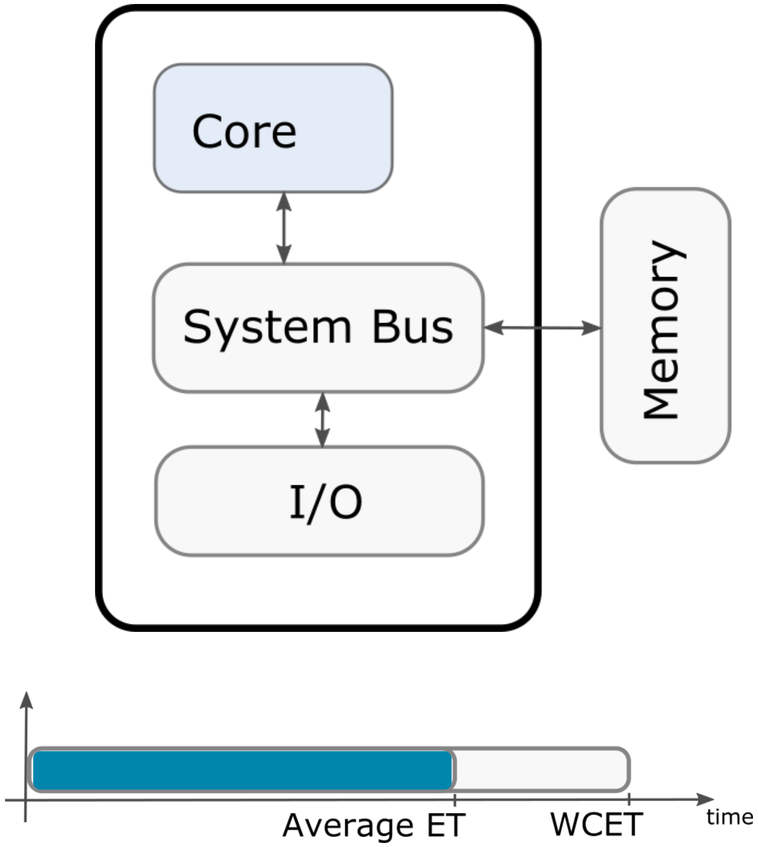
- Certification is not a burden
- Standards are useful
  - Not directly provided by Certification Authorities
  - Developed by industrial consortia (incl. Airbus)
  - Guide to useful and acknowledged actions

EMC^2 Research Project

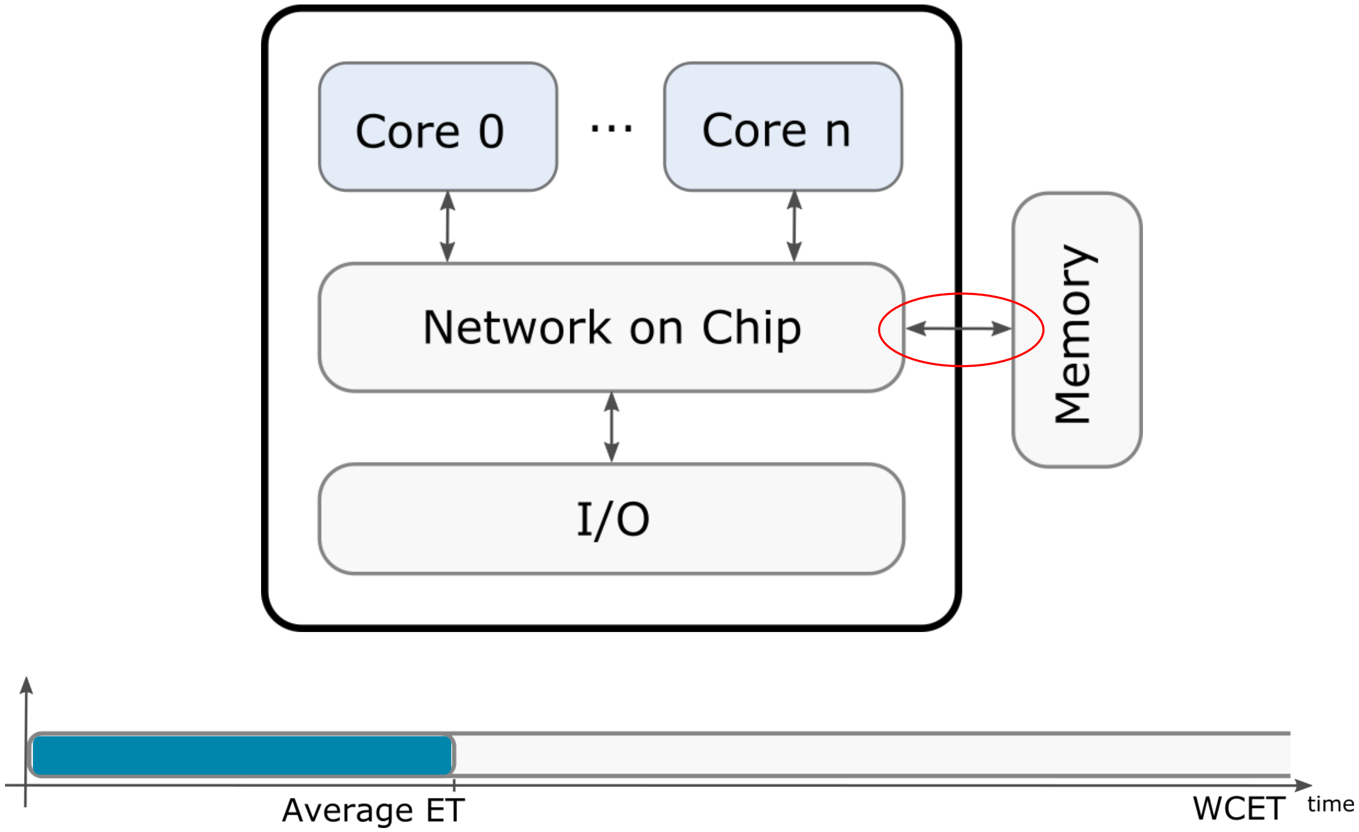
- Addresses the *famous* multicore interference problem
- Dream:
  Take a legacy single-core application and run it on a multicore
- Challenge:
  Does not work – interferences destroy any timing analysis
- Approach:
  Let someone monitor execution progress

Single-Core & Multi-Core

Single-Core

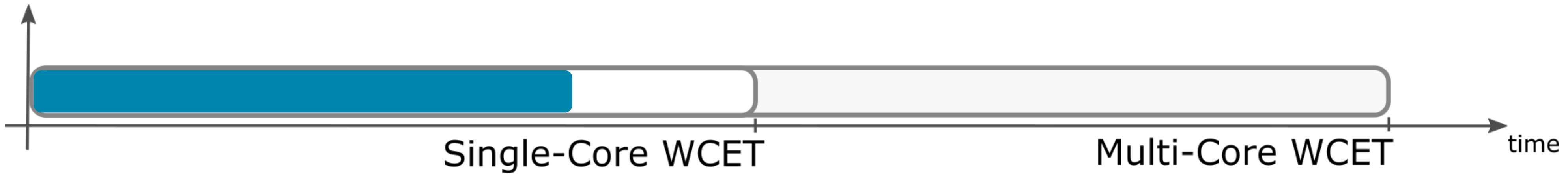


Multi-Core

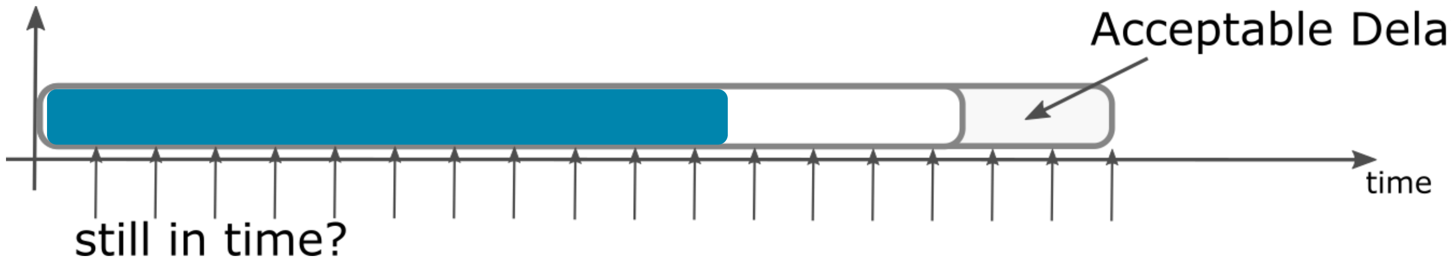


Basic Idea

- WCET approach

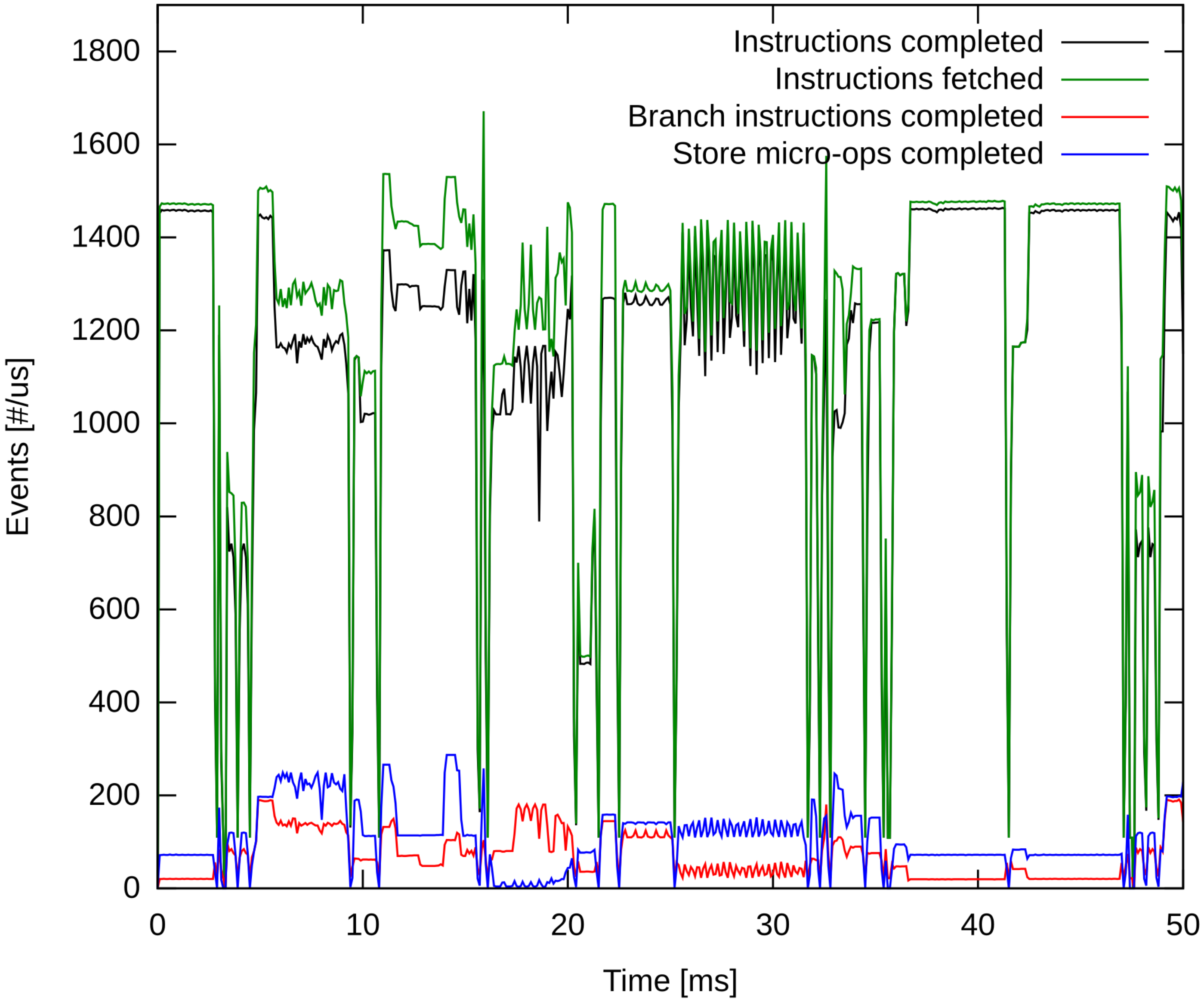


- Progress measurement during runtime



Application Fingerprint

- Performance counters in every processor core
- Application fingerprint:
  the recorded amount of events per second over the execution of the application
- Unique per application



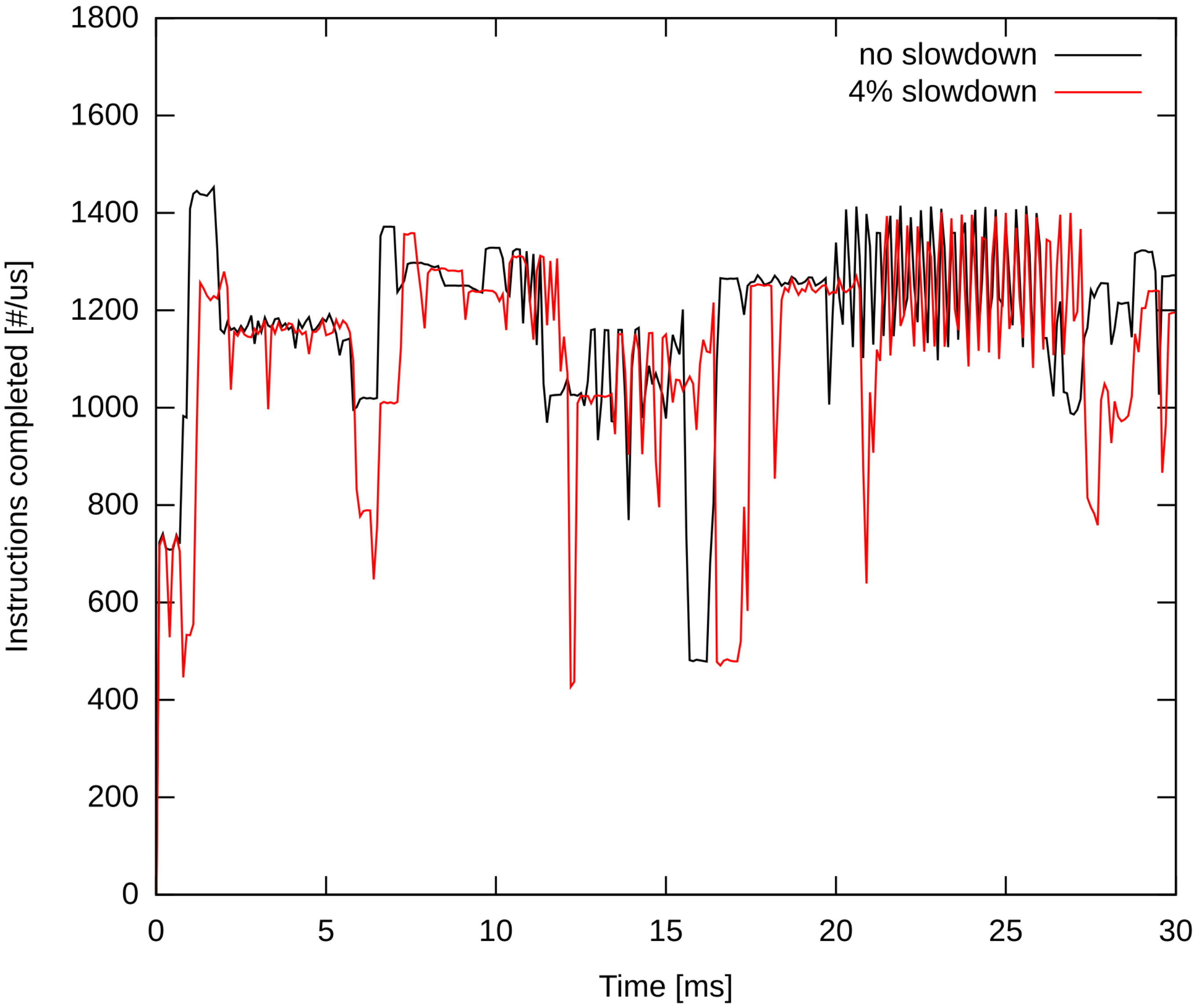
Application Fingerprint

- Performance counters in every processor core

- Application fingerprint:
  the recorded amount of events per second over the execution of the application

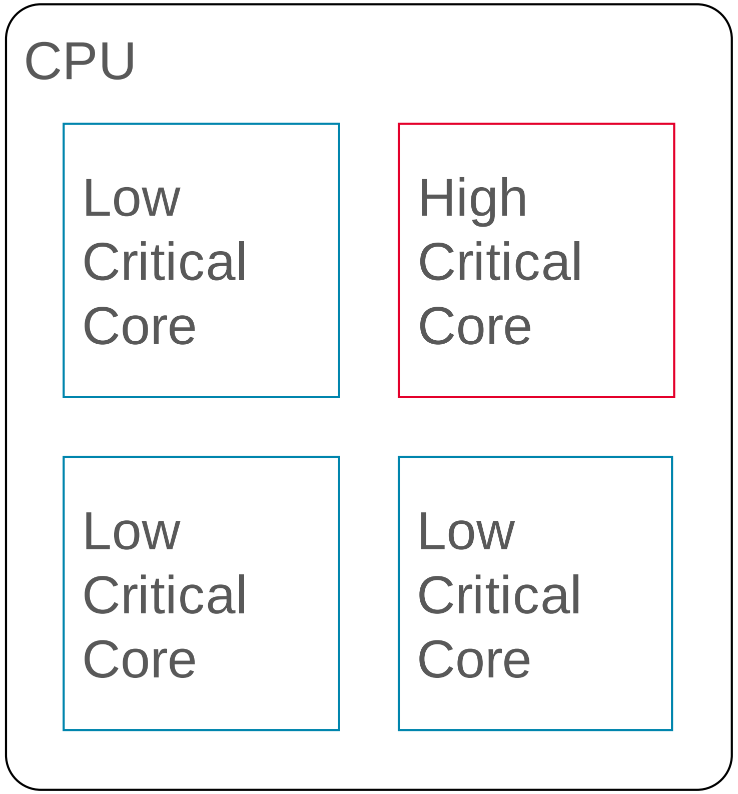
- Unique per application

- Similar but stretched and scaled curve when app is slowed-down



Basic Idea

Setup



CPU

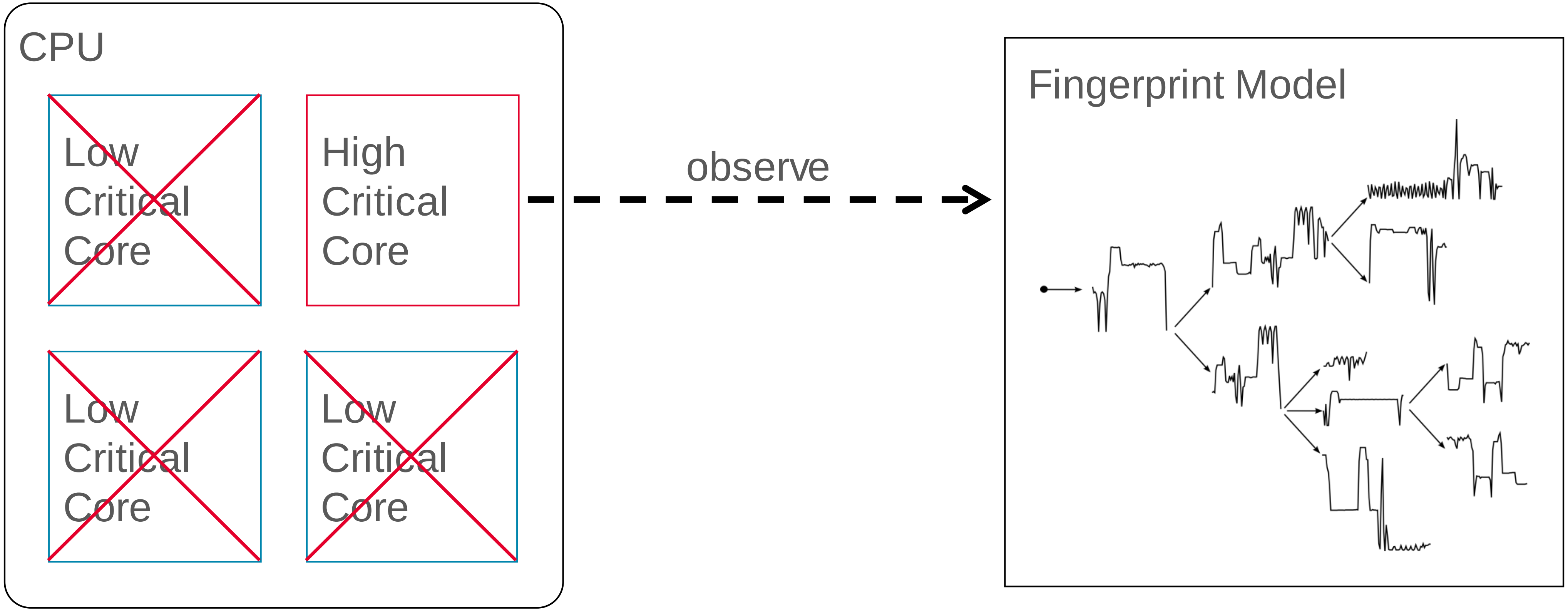
Low Critical Core

High Critical Core

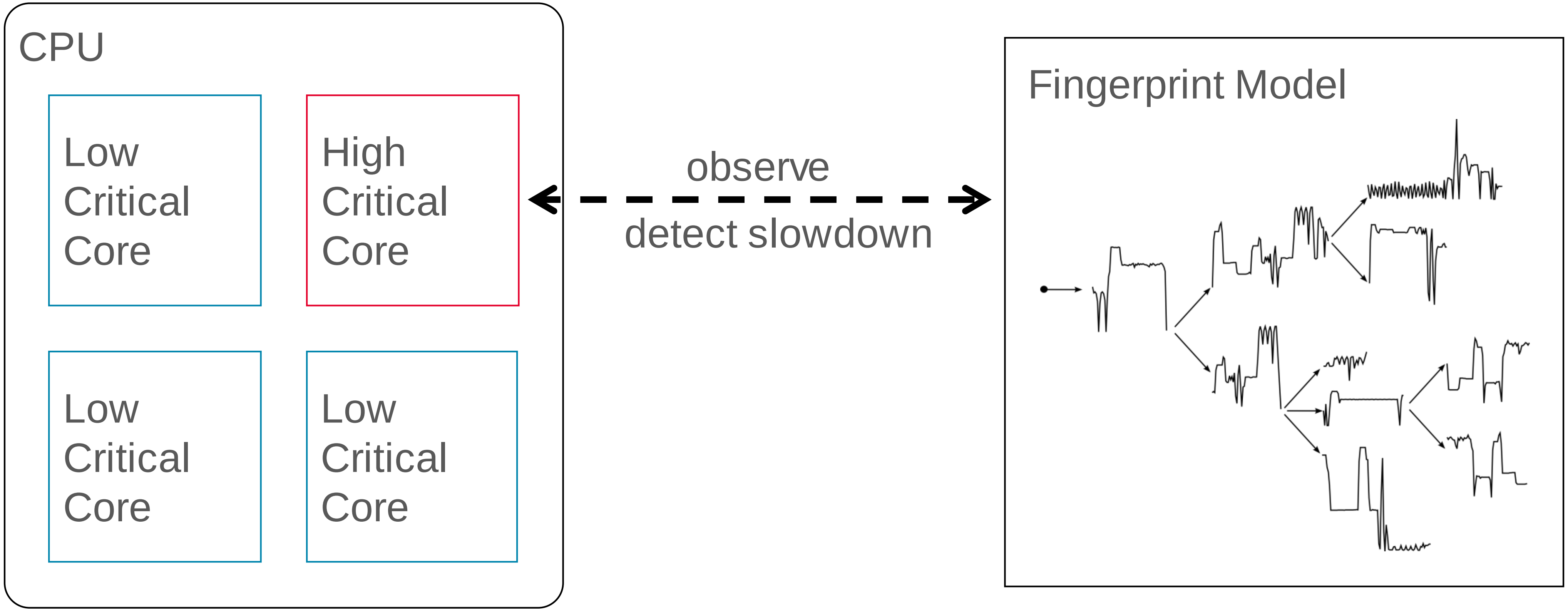
Low Critical Core

Low Critical Core

Basic Idea
At Development Time



Basic Idea
During Productive Execution



CPU

Low Critical Core

High Critical Core

Low Critical Core

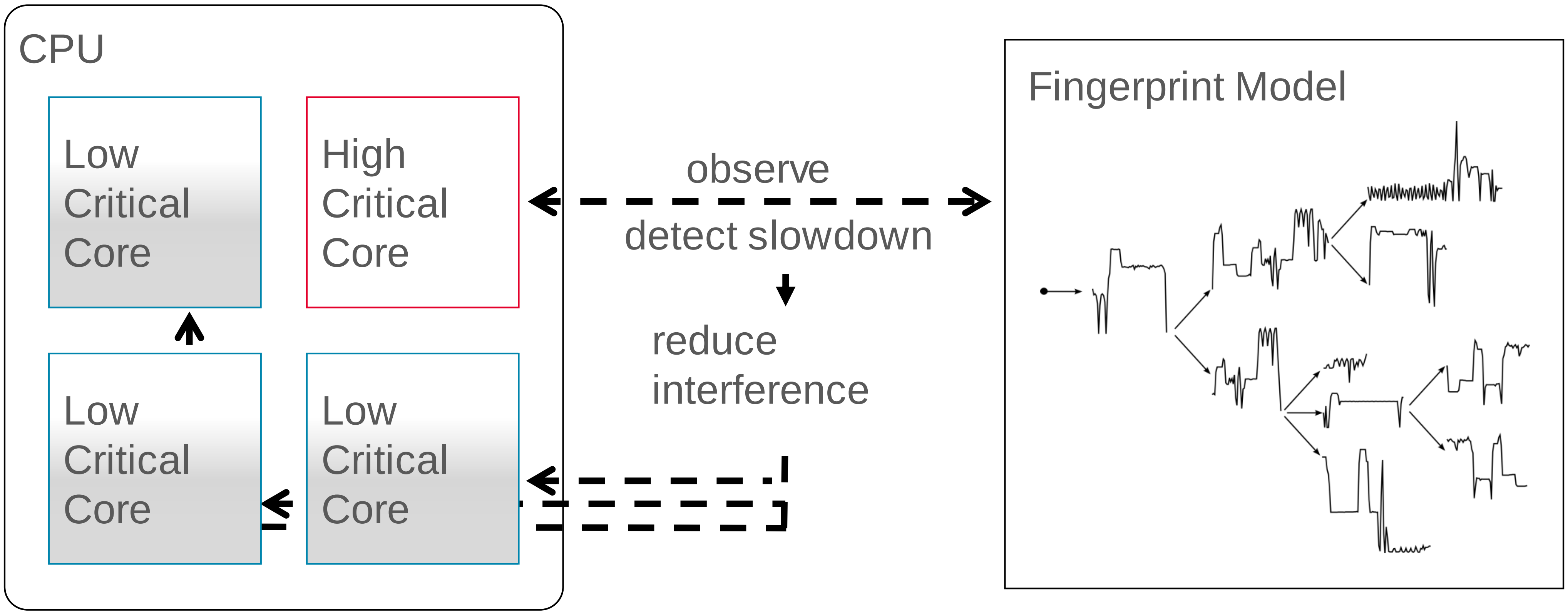
Low Critical Core

observe

detect slowdown

Fingerprint Model

Basic Idea
During Productive Execution



CPU

Low Critical Core

High Critical Core

Low Critical Core

Low Critical Core

observe

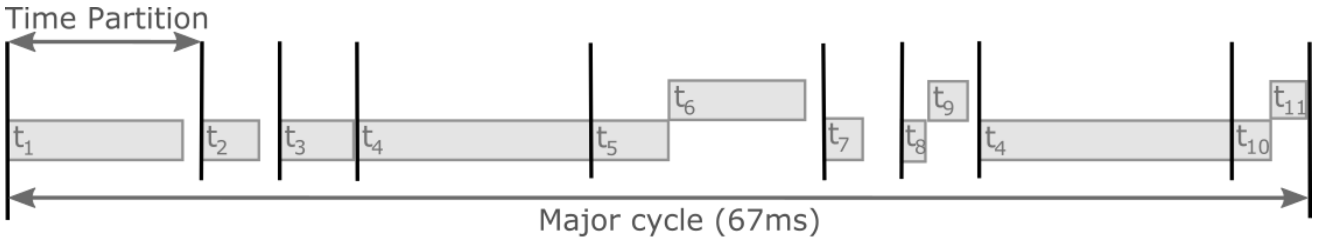
detect slowdown

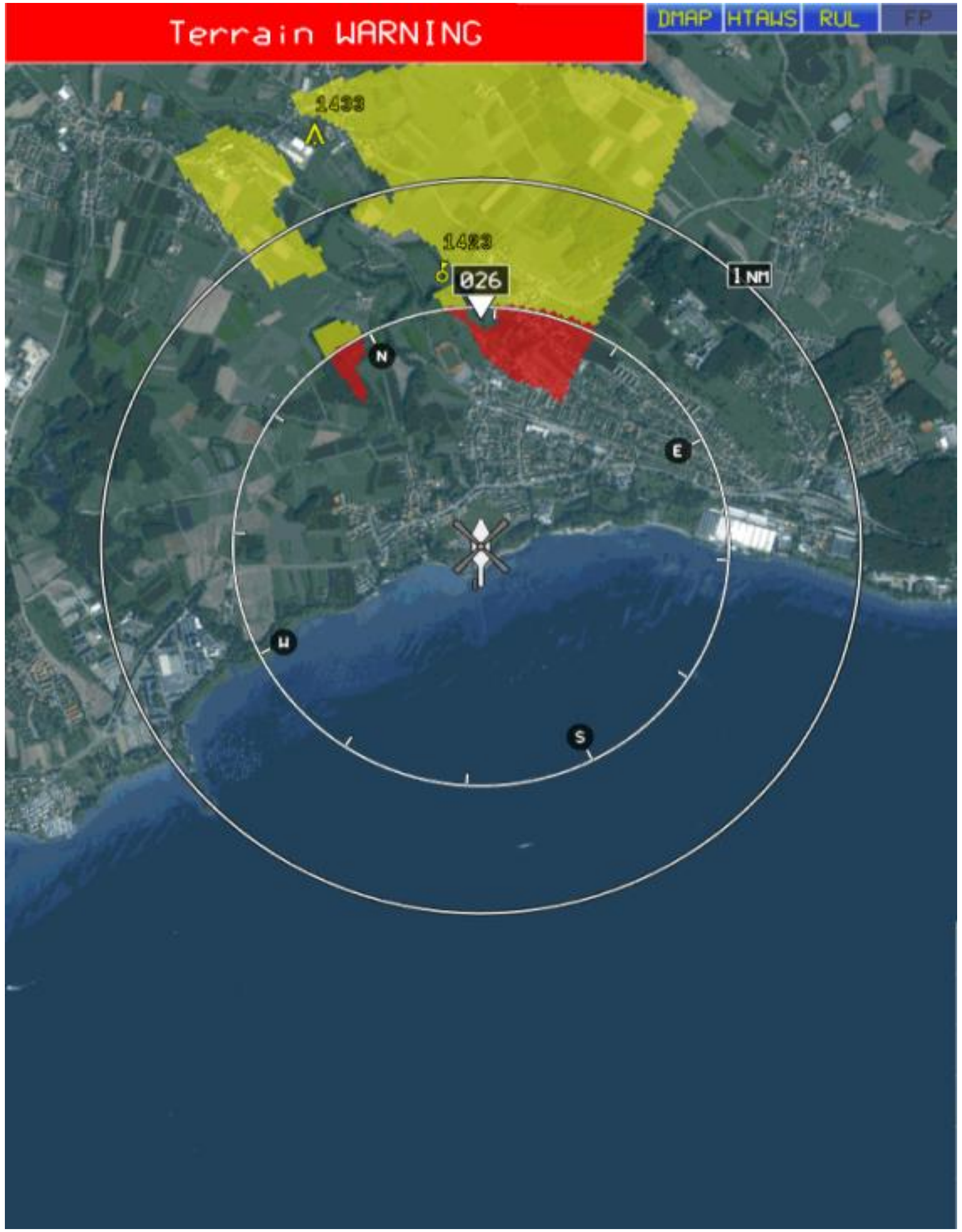
reduce interference

Fingerprint Model

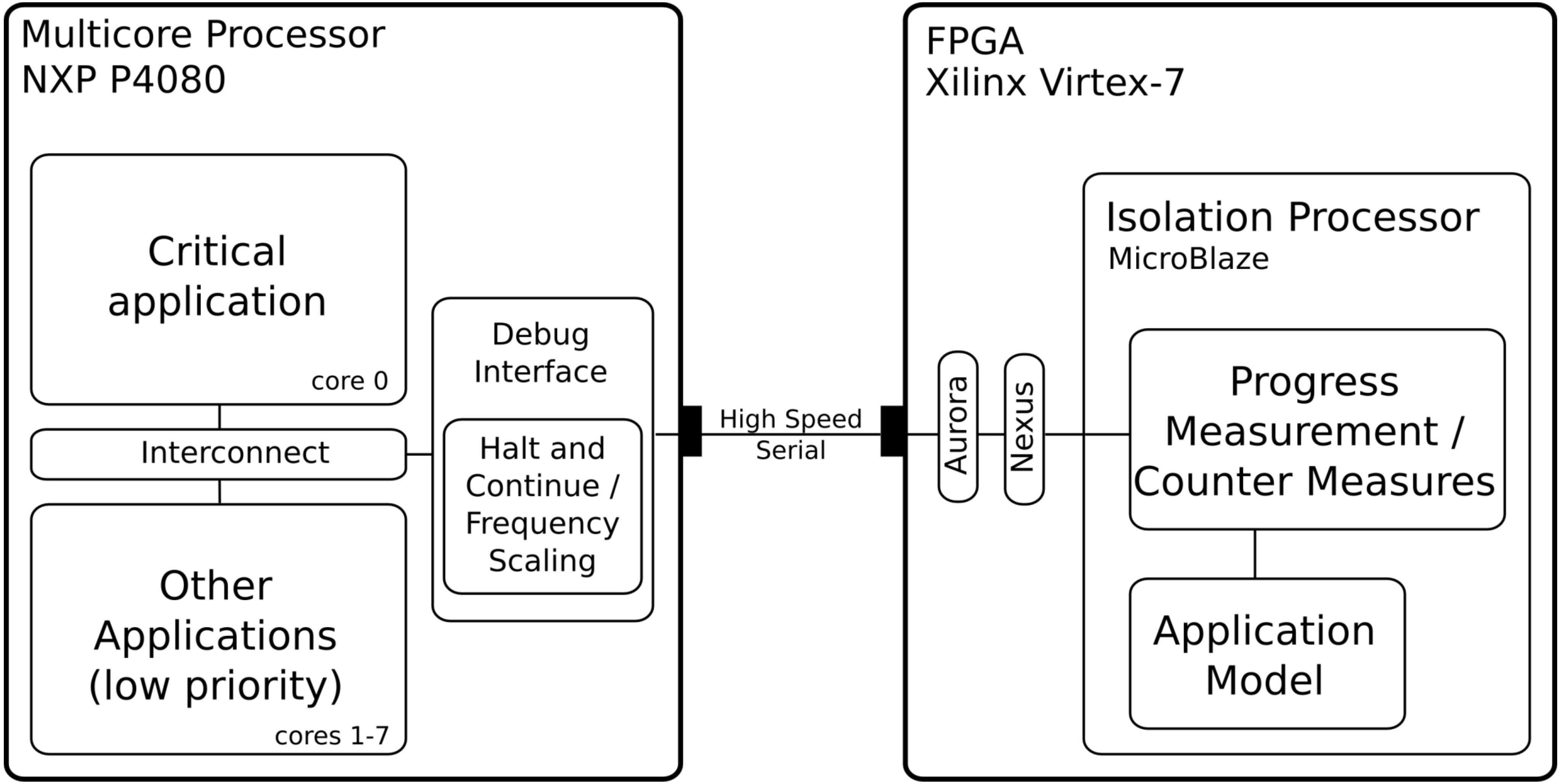
Helicopter Terrain Awareness and Warning System (HTAWS)

- Pilot support system
- Design Assurance Level: C
- Single core application
- Scheduled tasks within time space partitions

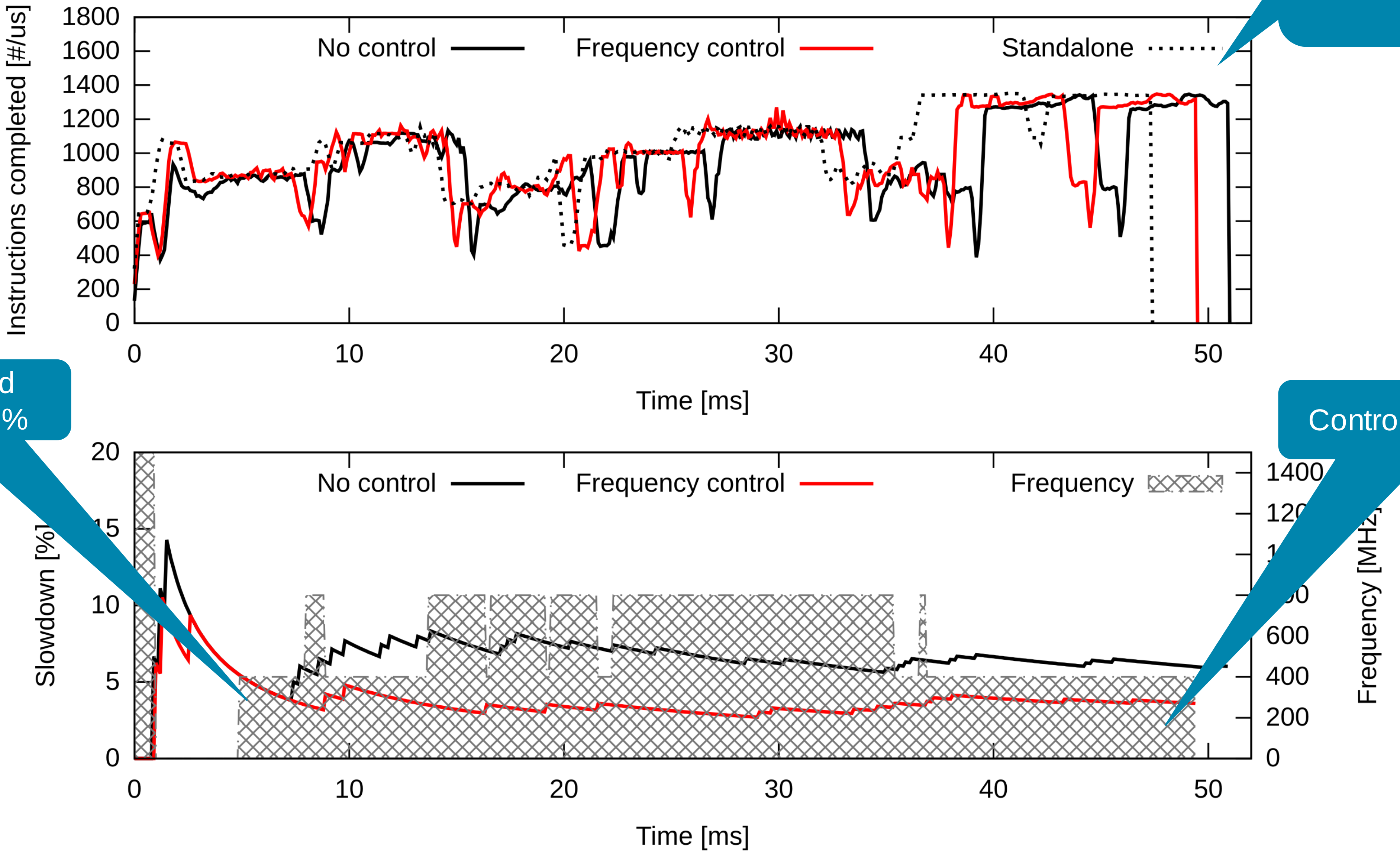




How?
Reduce Interference from other Cores



Control Measure: Frequency Scaling

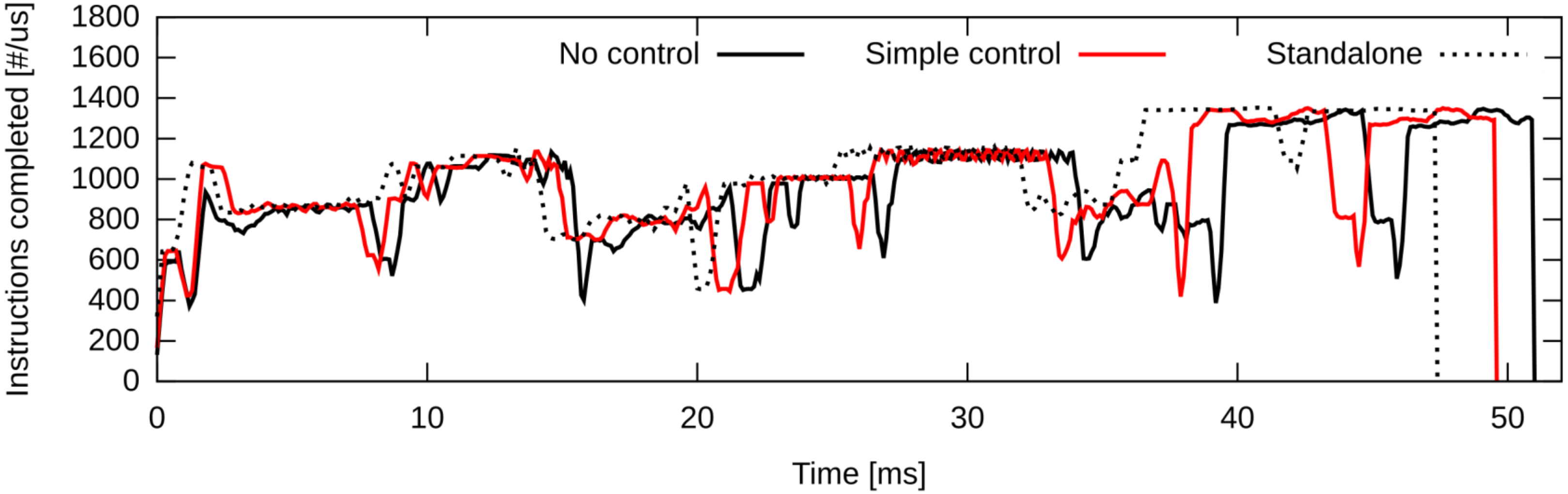


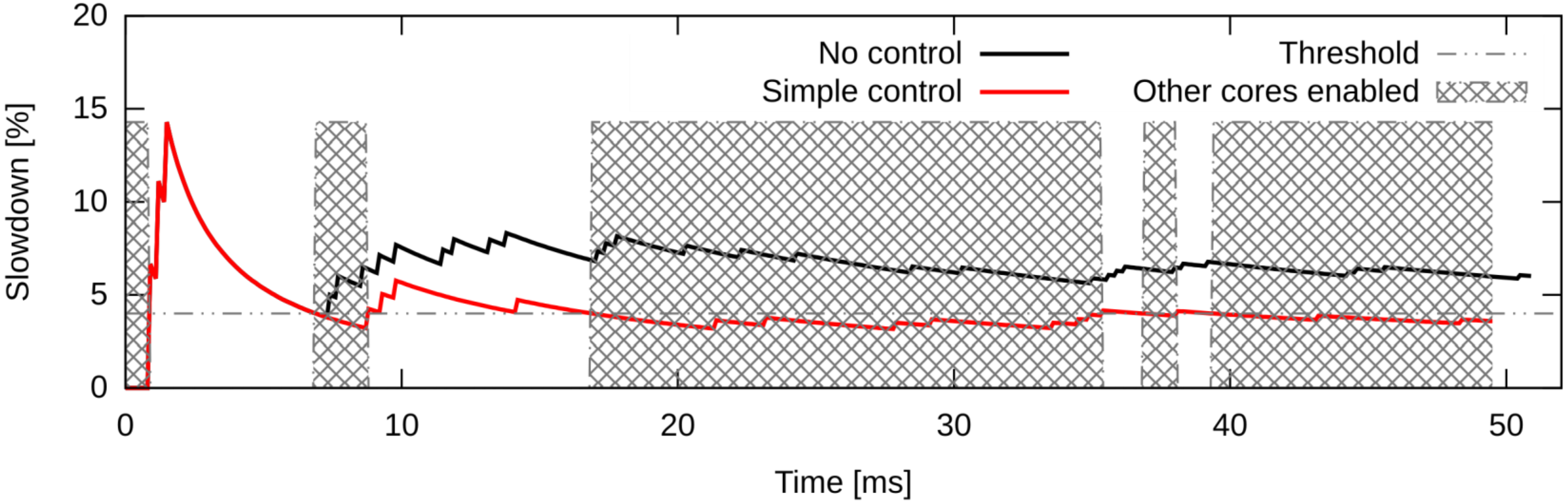
Measurements of main/critical core

Max. allowed slow-down: 4%

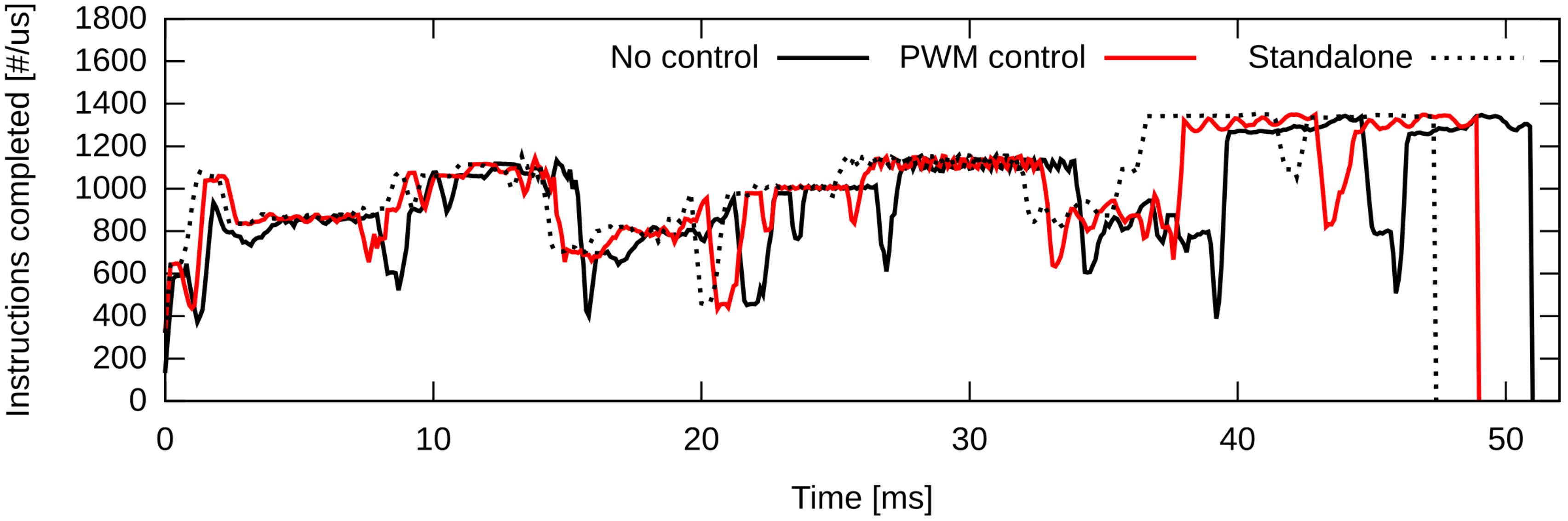
Control of other cores

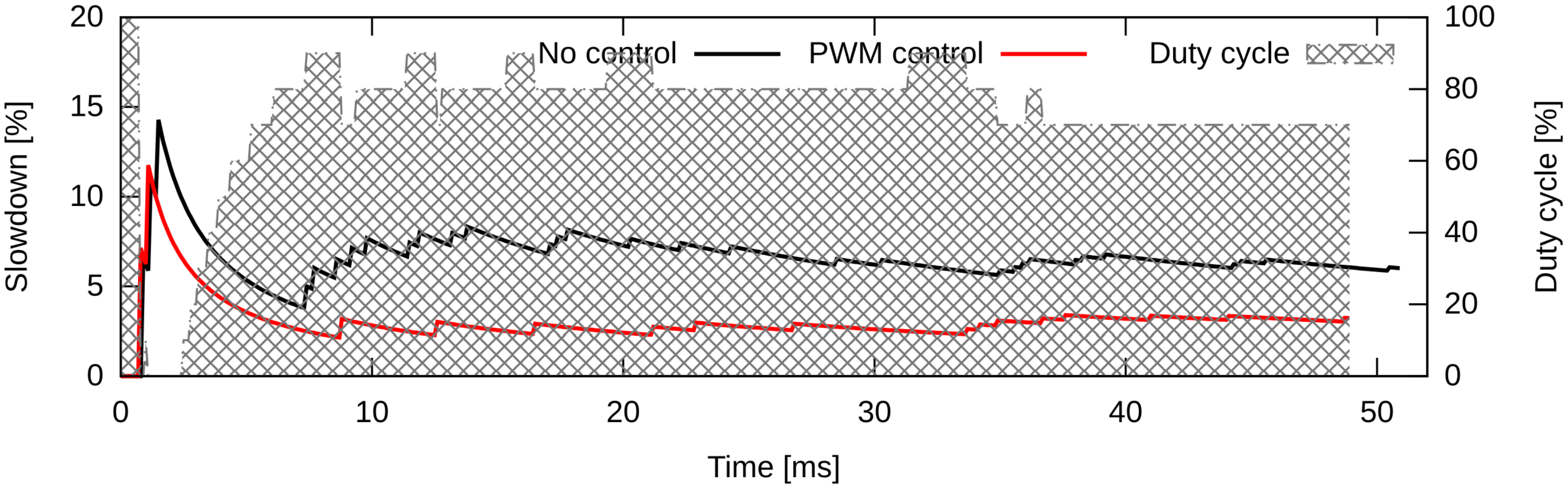
Control Measure: Threshold





Control Measure: PWM

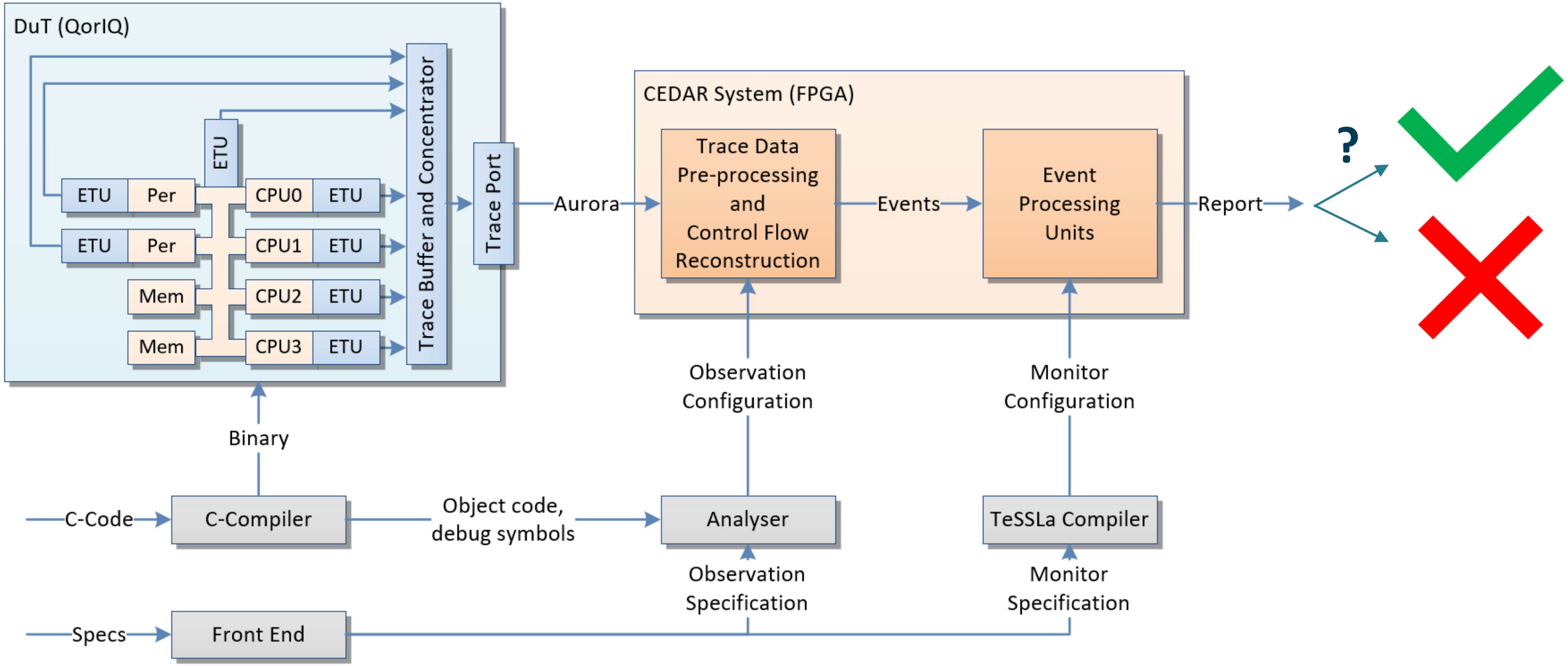




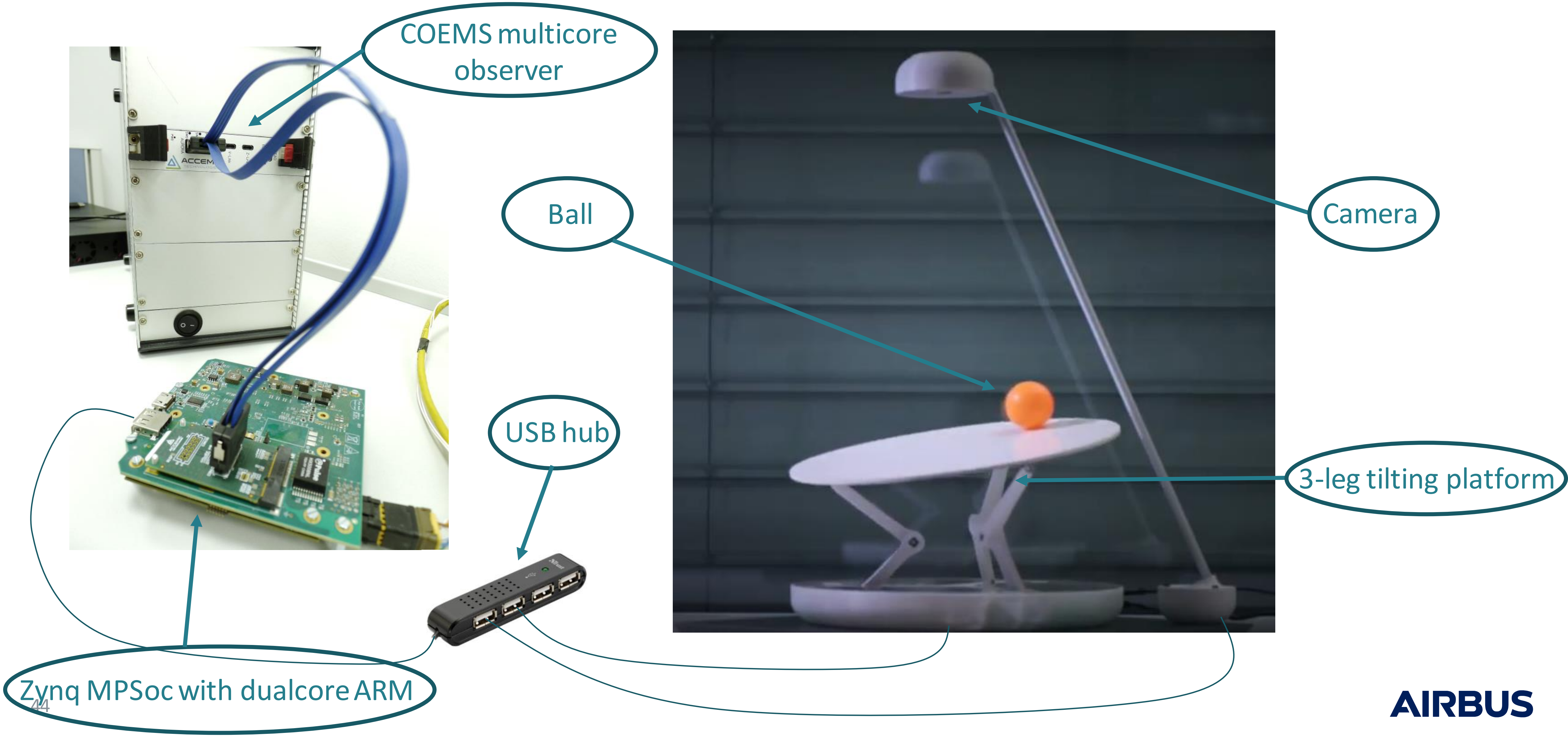
COEMS Research Project

- How to observe multicore behavior for longer periods?
- Dream:
  An instance checking if there are
  - any timing violations or “close to be violations”
  - any critical races
- Challenge:
  Does not work – only offline analysis possible, trace data from multicores too extensive
- Approach:
  Let evaluate behavior directly during execution

Dream, Idea


DuT (QorIQ)
ETU
Per
ETU
Per
Mem
Mem
ETU
CPU0
ETU
CPU1
ETU
CPU2
ETU
CPU3
ETU
Trace Buffer and Concentrator
Trace Port
Aurora
CEDAR System (FPGA)
Trace Data Pre-processing and Control Flow Reconstruction
Events
Event Processing Units
Report
?
Binary
Observation Configuration
Monitor Configuration
C-Code
C-Compiler
Object code, debug symbols
Analyser
TeSSLa Compiler
Observation Specification
Monitor Specification
Specs
Front End

Demo: Ball Balancing System



COEMS multicore observer

Ball

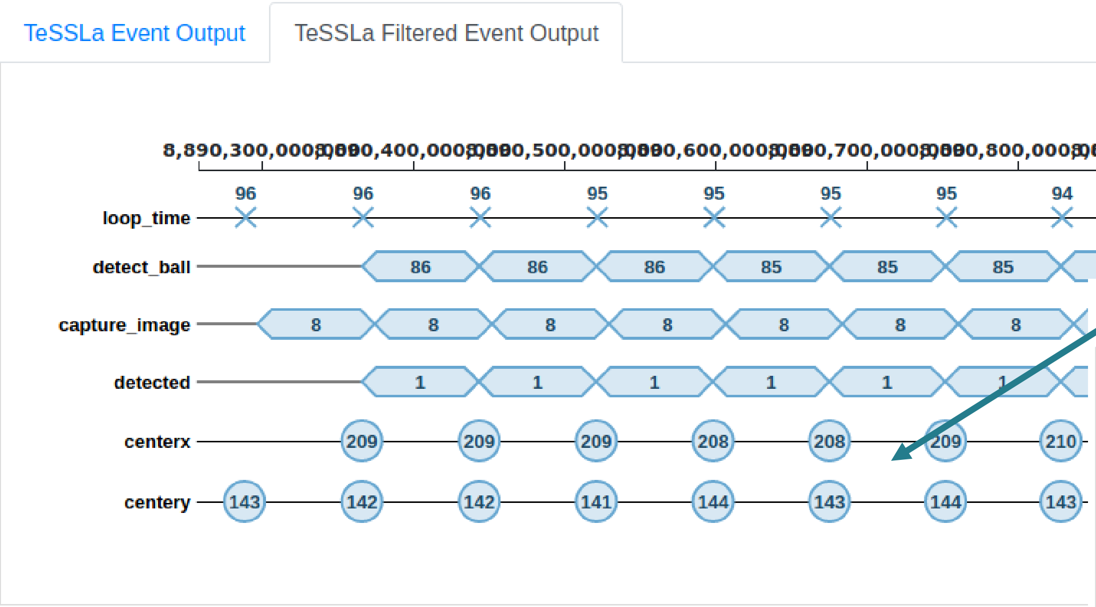
Camera

USB hub

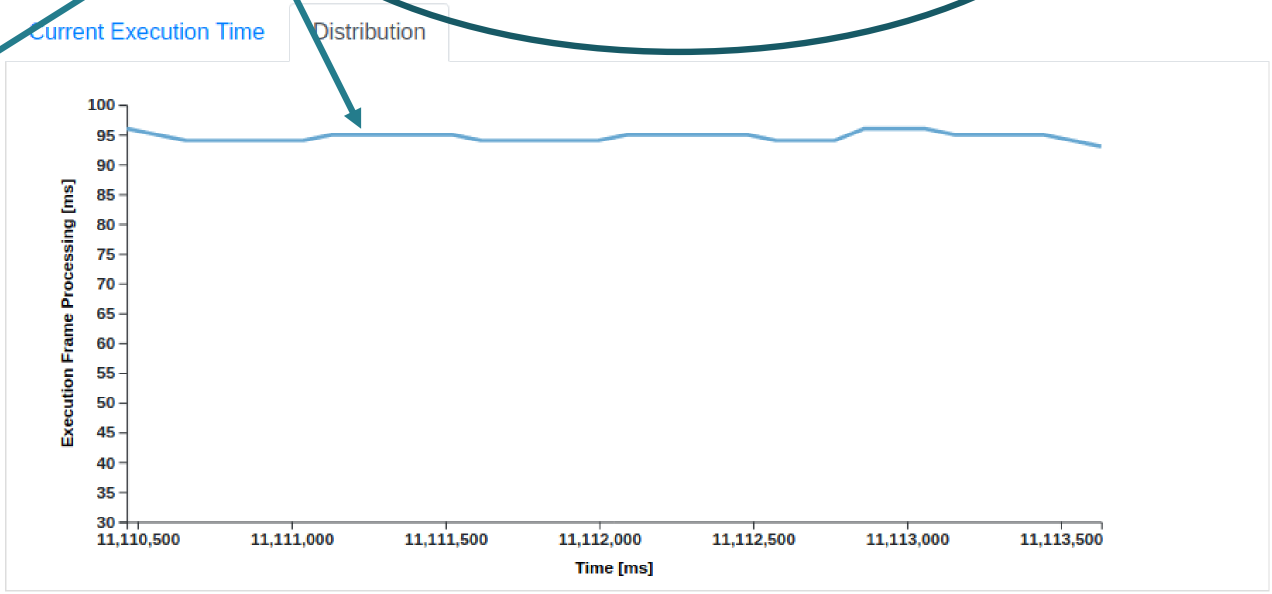
3-leg tilting platform

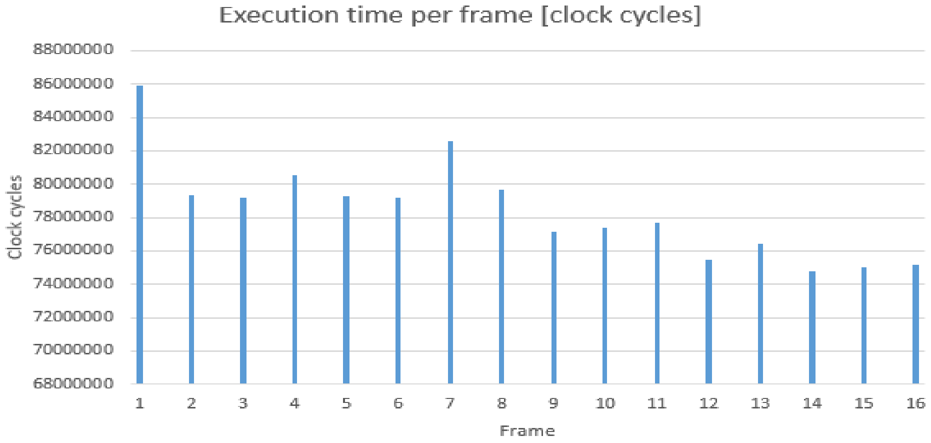
Zynq MPSoc with dualcore ARM

Related Timings



Relationship between ball coordinates and execution time:
Ball not in centre → lifted ball
→ larger ball → longer execution time

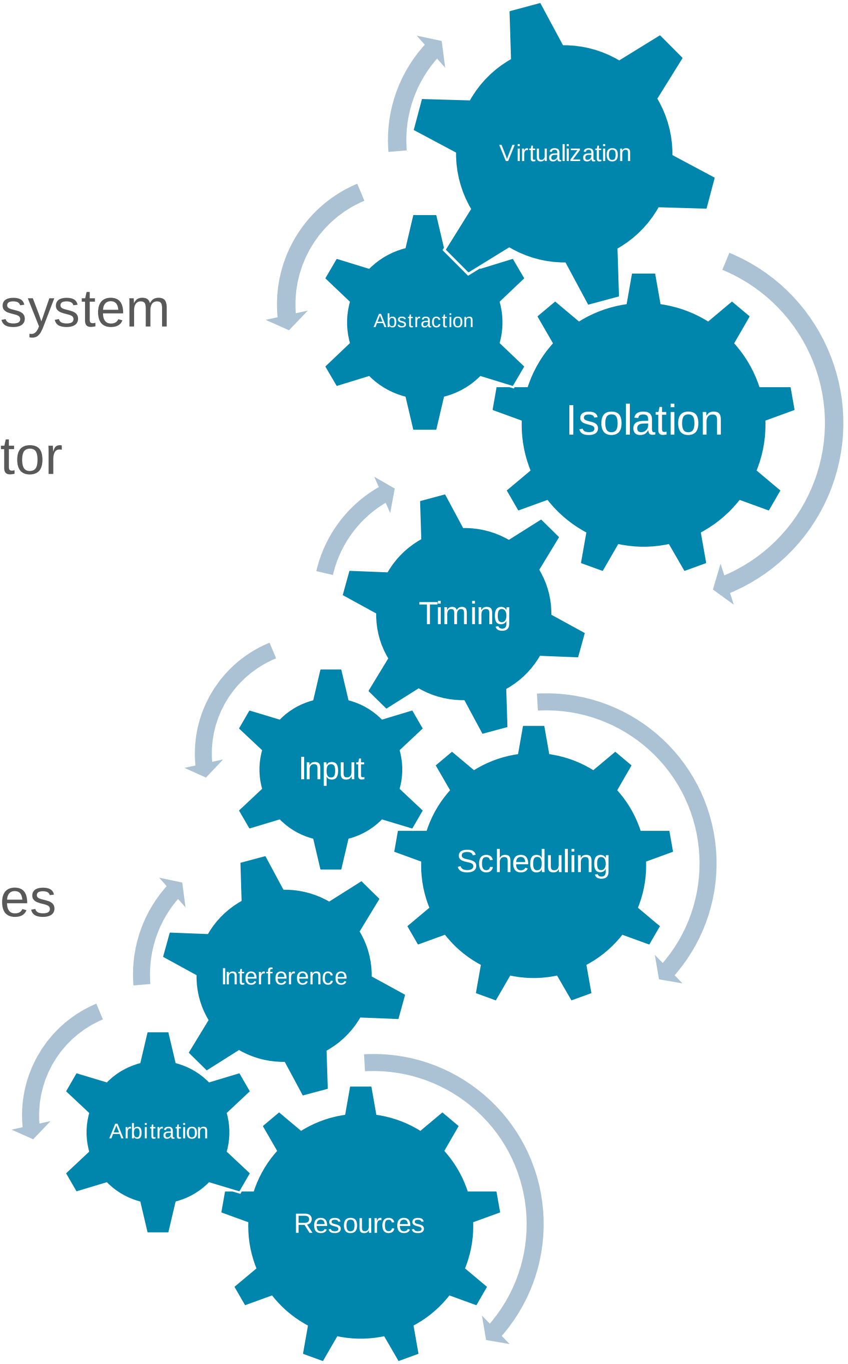




Academic contribution…

(not complete!)

- Certification targets the complete system
  - Including all layers
  - Must be done by system integrator (= Airbus)
- Why not starting from scratch?
  - Address the **root cause**
  - Forget about modern cores
  - Forget about traditional multicores
  - **Invent something new! Less complex, more useful (for us)**



AIRBUS

Many thanks…

Discussion!

Sascha Uhrig, Airbus Central Research & Technology