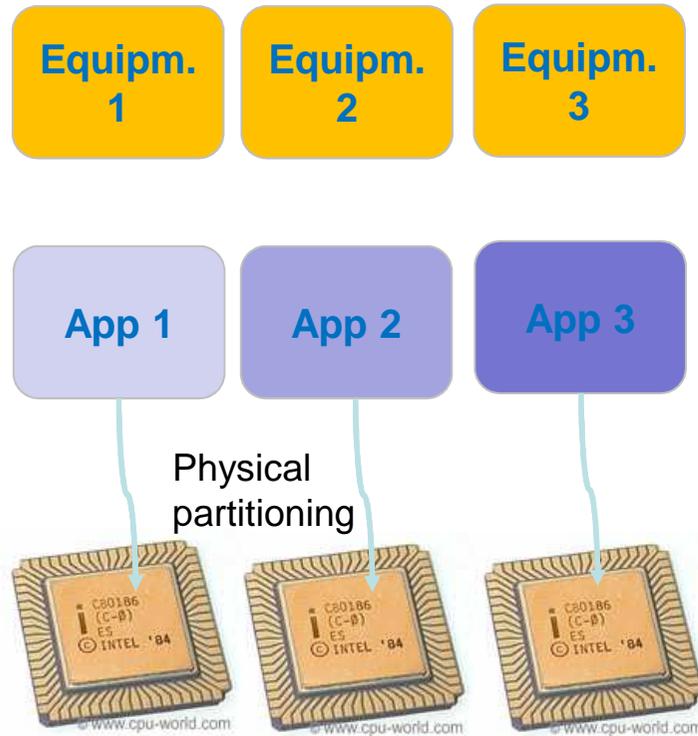


# **EMC<sup>2</sup> Use Case: Hybrid Avionics Integrated Architecture Demonstrator**

Bernd Koppenhöfer / Dietmar Geiger  
Airbus D&S Electronics

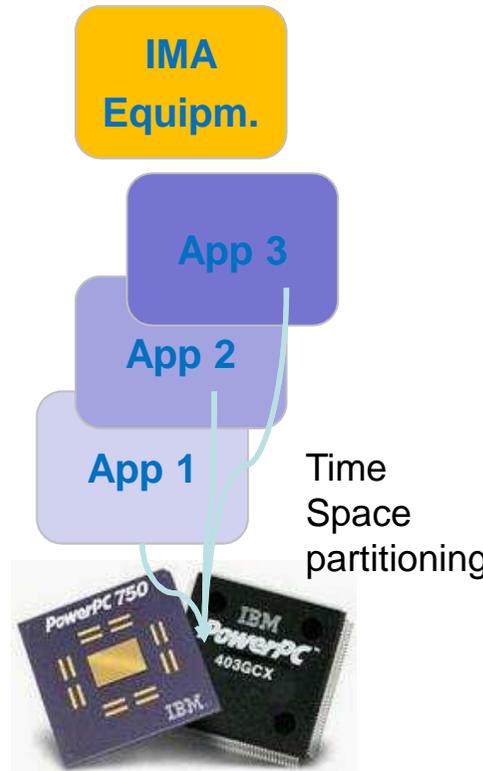


# From Single to Many Cores Processors



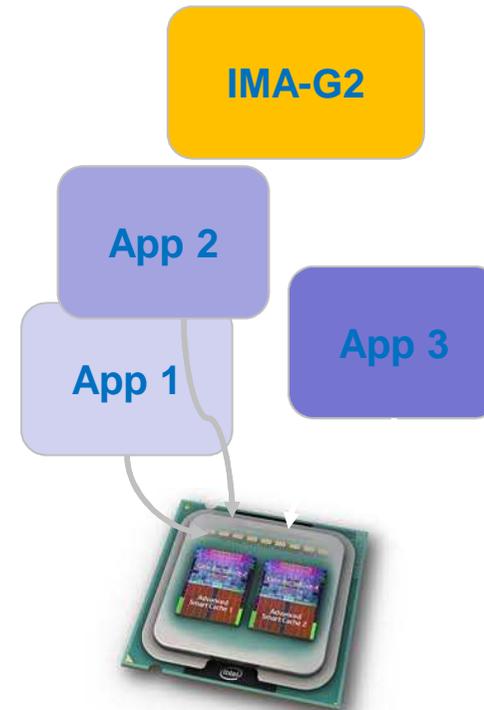
**Single CORE  
Real Time OS**

**Limitation  
Weight, power, size**



**Single CORE  
Robust Partitioning  
Arinc 653 OS**

**Limitation  
Performance**



**Dual/Multi CORE  
A653-1 OS  
AMP\* model**

**Challenge  
Certification Approach**

AMP: Asymmetric multiprocessing with exactly one application per core

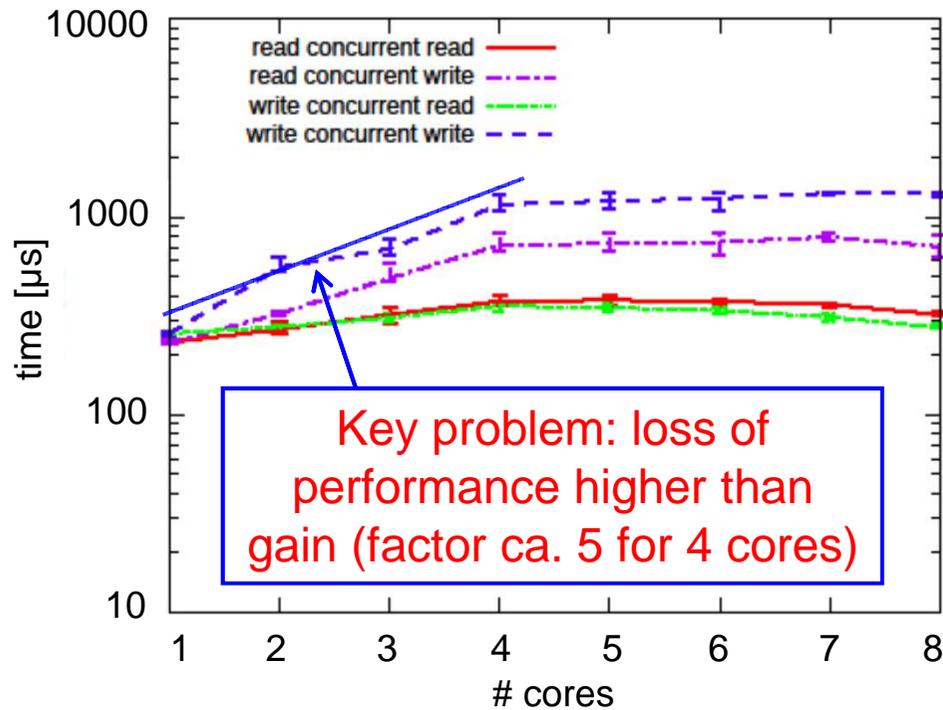
# P4080 Evaluation Results (1)

Source: ARTEMIS Project Recomp

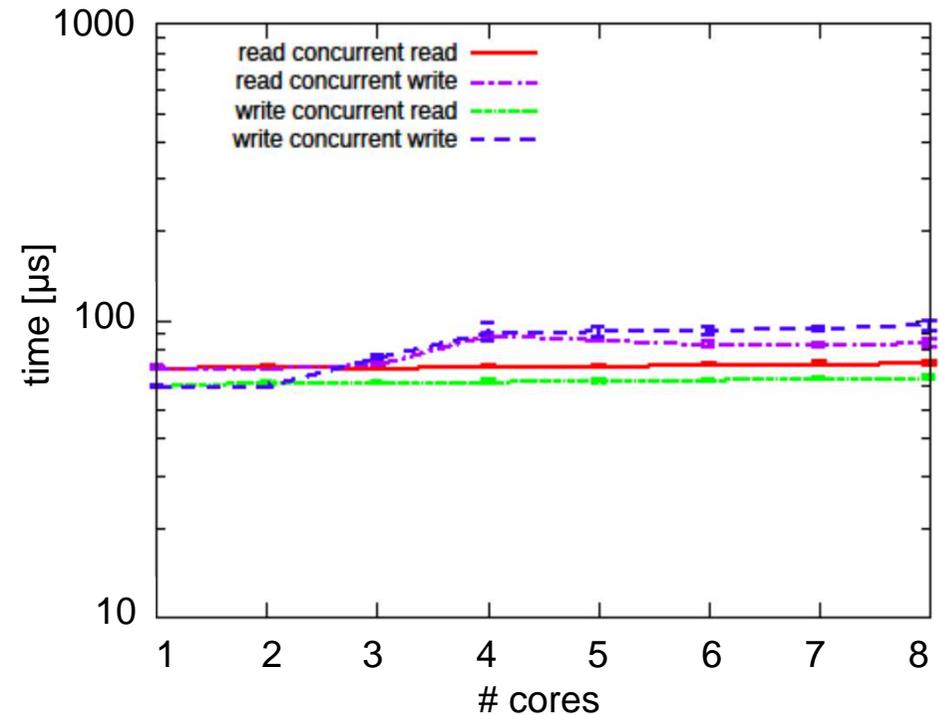


- Maximum interference of one core by other cores test setup for DDR vs. SRAM (Level 3 Cache)
- Concurrent access to memory regions with 4 cores (4kB regions, 64B gap, 1 to 8 cores, ...)

### DDR



### SRAM



Airworthiness Authorities view:

E.g. Integrated Modular Avionics (IMA) - Guidance DO297

Key Applications Characteristics	Description
Application developed independent from each other Incremental acceptance	Applications independent verified on the platform
Applications implemented <b>without unintended interactions</b> with other applications	Applications can be verified independently Finally complete suit of applications needs to be implemented and interactions to be verified
Applications may be <b>reusable</b>	Application modularity and portability
Applications may be <b>modified independently</b>	Each application modifiable with limited or no impact on other applications



*See workshop "MCS" for details on IMA/DO297.*



- **Stay with single Core processors**  
→ Not really a long term solution
- **The deterministic MultiCore**  
→ Where is the Chip Vendor / Market  
→ we love this approach, but is it realistic ?
- **Only use Core intrinsic Resources (Cache)**  
→ Still possible conflicts with external resources
- **In Service Experience**  
→ How many hours do we need – in which system configuration  
→ Focus on limited number of different devices and configuration settings
- **System Safety Net** - Monitoring and Mitigation on System level  
→ Specific for the Application
- **HW Safety Net** - Monitoring of the device function independent of the application on HW-level  
→ Versatile approach

# HW Safety NET: Safety-driven Design constraints



## Requirements:

- Issues inside the MCP shall be detected
- Decision needs to be outside the MCP
- Monitoring should be independent of application (otherwise no open platform)

**Effectivity of HW Safety NET has to be demonstrated**

## Approval from Authority required

- This can only be a stepwise approach
- Implement on a less critical System
- Gain Experience

# HW Safety NET: Monitoring Approach



- Carefully Select Chip Manufacturer and Target MCP
- Investigate possible interference channels on the chip
- Cooperate with Chip Manufacturer to understand nature of interferences
- Define Correct Configuration and Usage Domain of the MCP
- Propose Monitoring Strategy – Together with Chip Manufacturer
- Propose Mitigation Strategy
- Demonstrate Effectiveness of Mitigations
- Involve Authority in Investigation

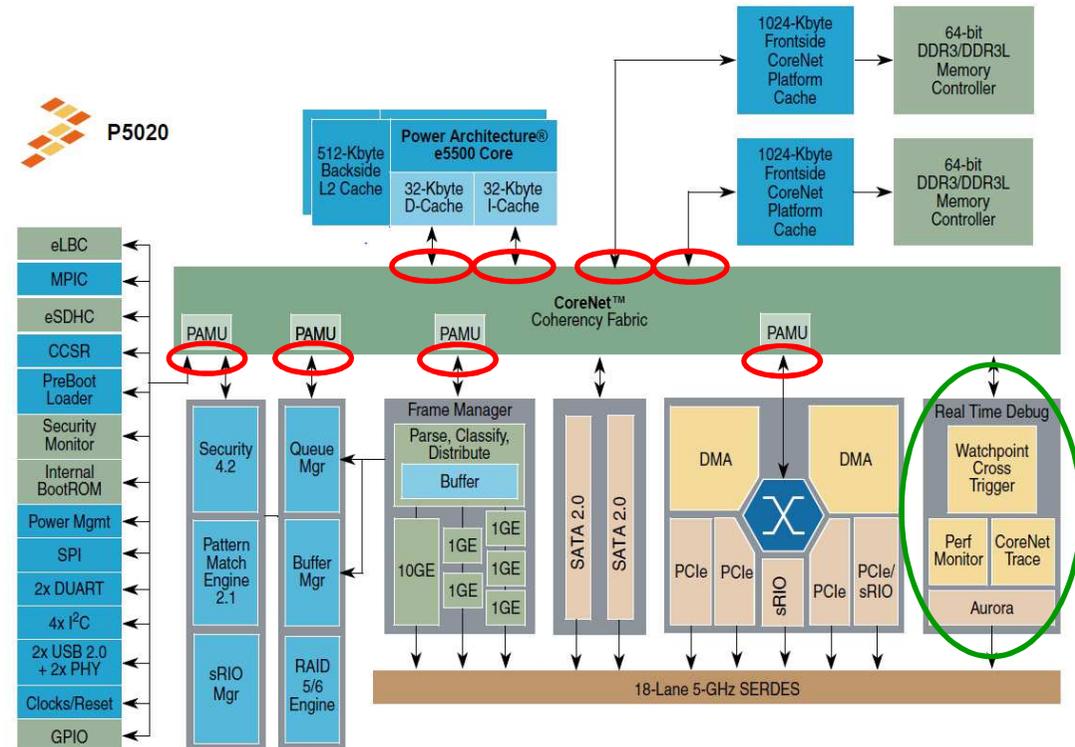
## Test Platform: Freescale P5020

Monitoring Points 

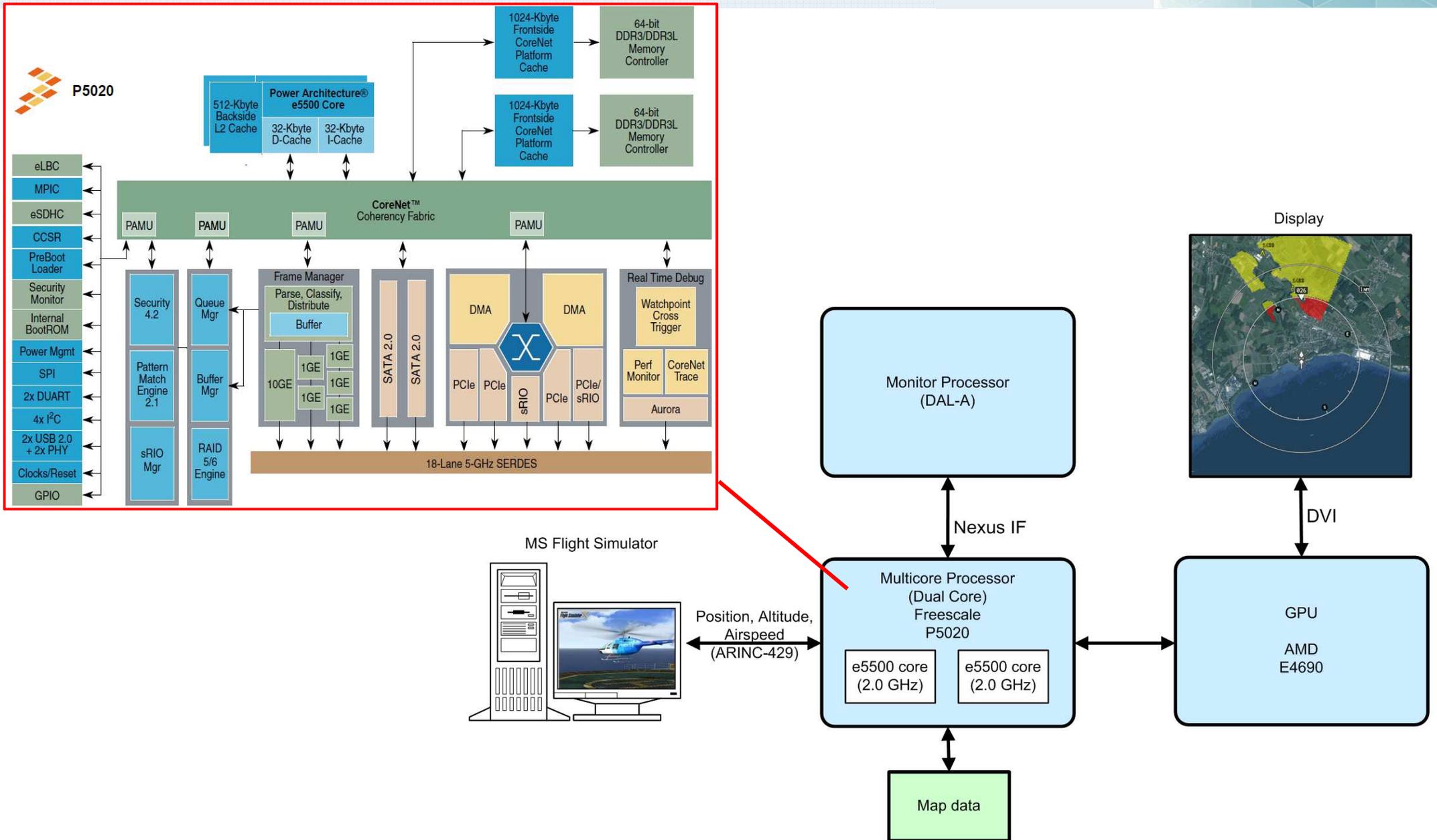
Bandwidth – Timing ??

⇒ Selection of Monitoring Interface

Can we use the **Debug Interface** for Monitoring



# Avionics Demonstrator: Blockdiagram



# Avionics Demonstrator: Sample Application HTAWS



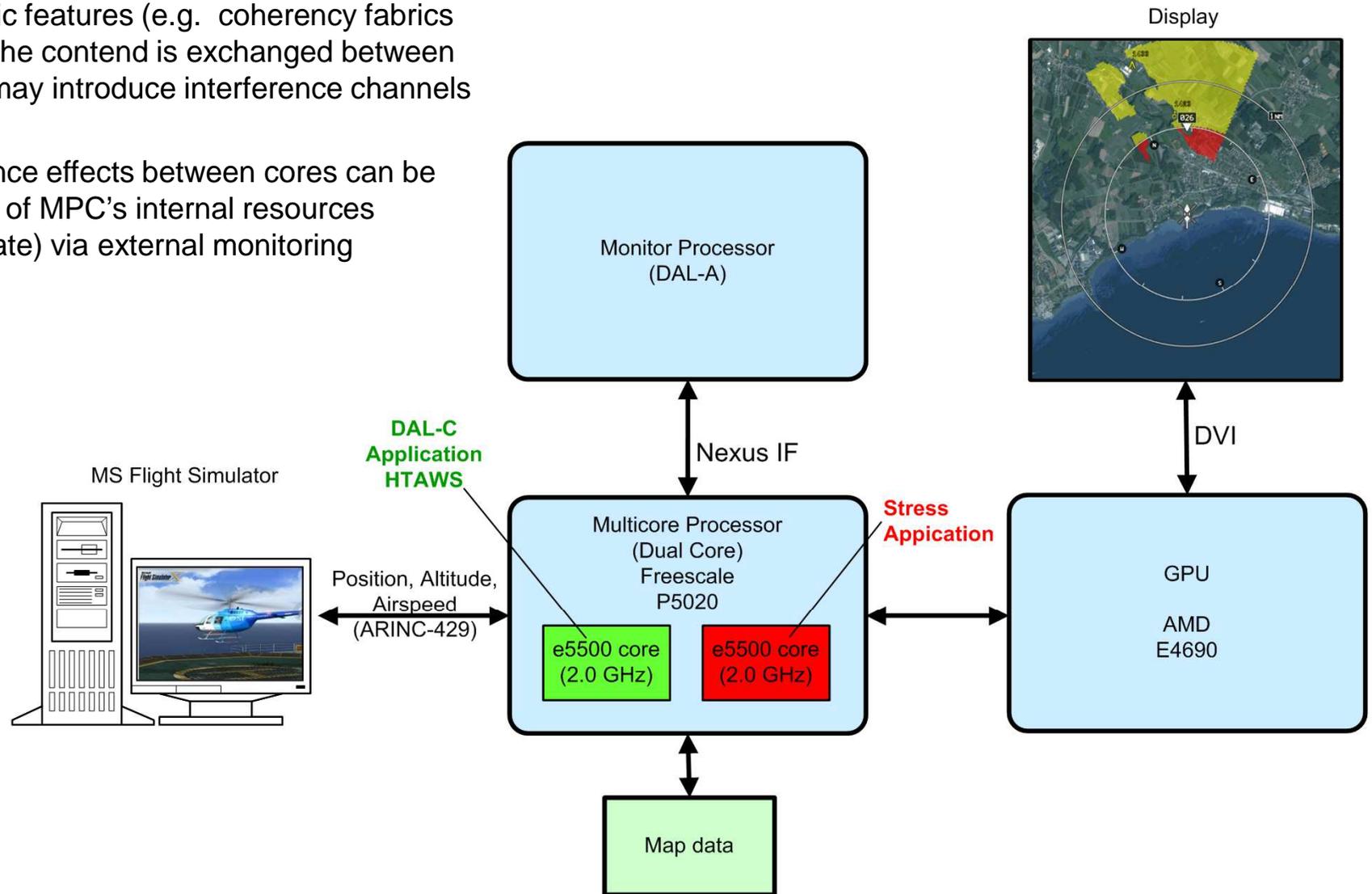
## Helicopter Terrain Awareness and Warning System

- Supports pilots
  - flying at night
  - in changing weather with poor visibility
  - rough terrain or at low altitudes
- To prevent avoidable collisions with ground or obstacles
- Provides a comprehensive, map based overview of helicopter's surroundings



# Avionics Demonstrator: Further Activities (1)

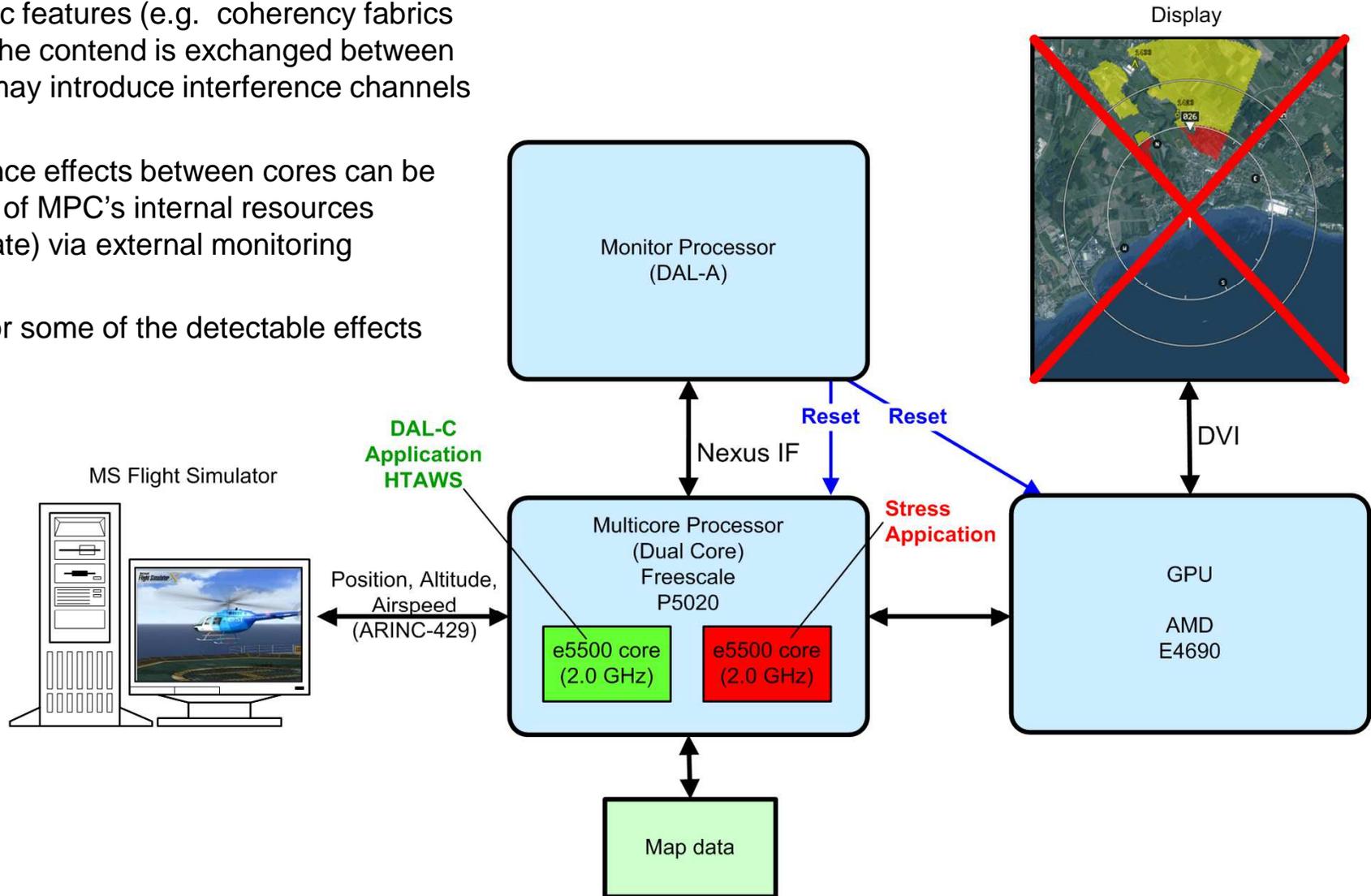
1. Implement **HTAWS** at one e5500 core and a **Stress Application** on the other e5500 core
2. Search for MPC specific features (e.g. coherency fabrics which ensures that cache content is exchanged between different cores) which may introduce interference channels between cores.
3. Investigate, if interference effects between cores can be detected by monitoring of MPC's internal resources (e.g. memory access rate) via external monitoring processor



# Avionics Demonstrator: Further Activities (2)



1. Implement **HTAWS** at one e5500 core and a **Stress Application** on the other e5500 core
2. Search for MPC specific features (e.g. coherency fabrics which ensures that cache content is exchanged between different cores) which may introduce interference channels between cores.
3. Investigate, if interference effects between cores can be detected by monitoring of MPC's internal resources (e.g. memory access rate) via external monitoring processor
4. Implement mitigation for some of the detectable effects





Thank you for your attention!

This investigations were partly funded by:  
Bundesministerium für Bildung und Forschung 53170 Bonn  
Forderkennzeichen: 01 IS14002D  
Verbundprojekt EMC2

Joint Technology Initiative – Collaborative Project (ARTEMIS) ; ARTEMIS-2013-1  
Grant Agreement Number 621429