THALES
Building a future we can all trust

Multicore Challenges

in

Avionics Applications

Frederic Bonamy, Thales AVS

14th June, 2024

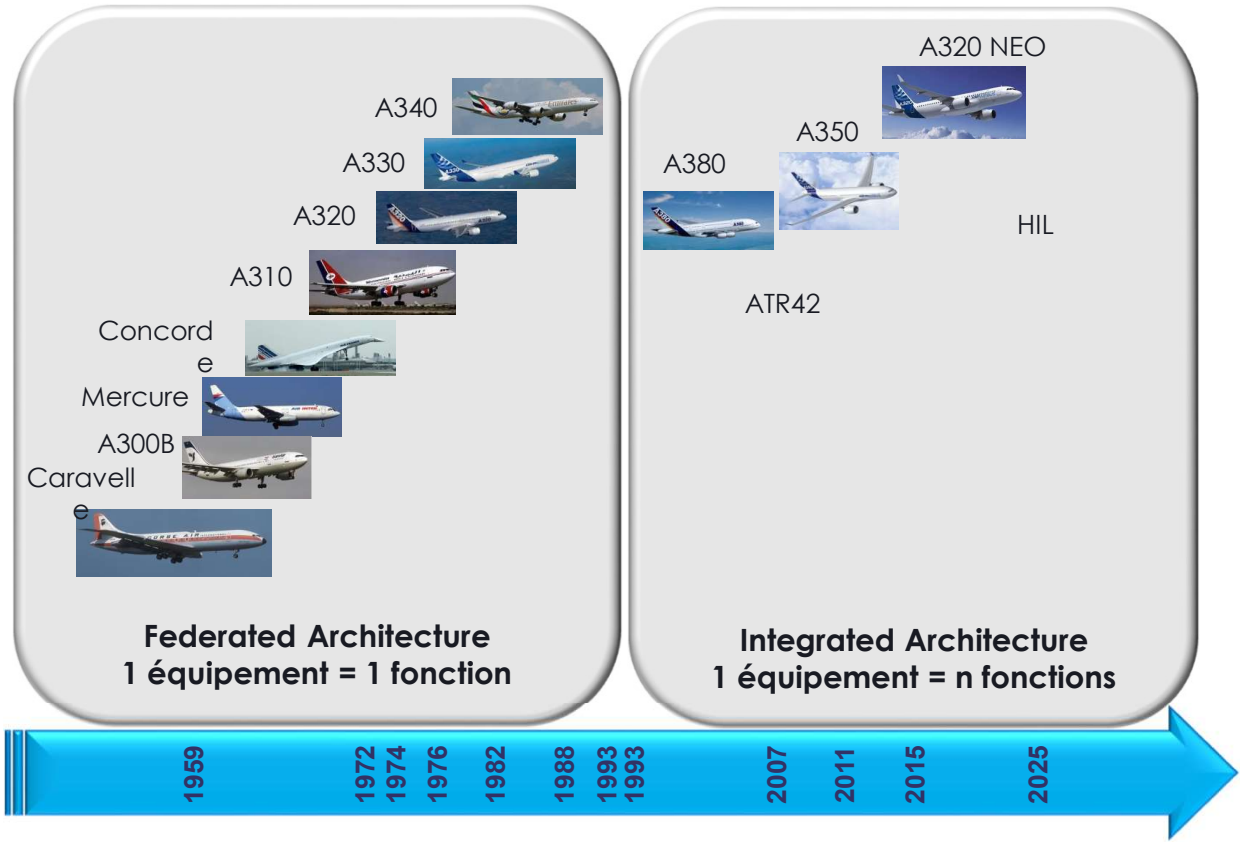
www.thalesgroup.com

Decades of computing evolution

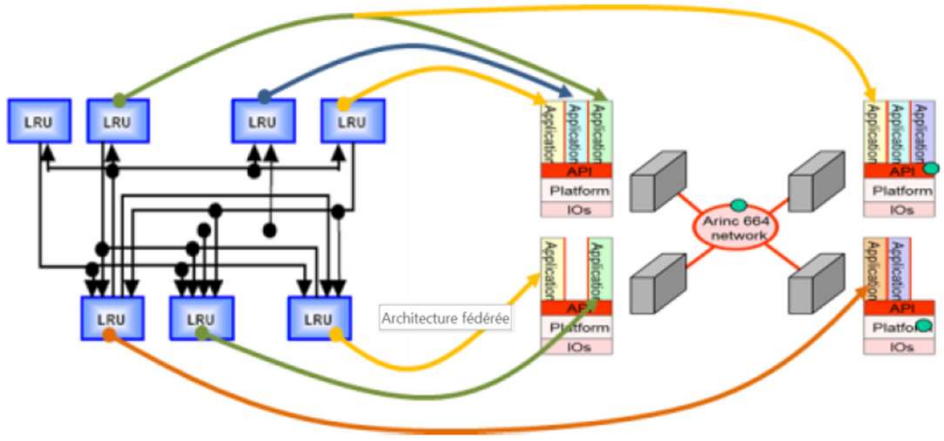
Now on air: introducing PureFlyt, the new-generation Flight Management System - Thales Aerospace BlogThales Aerospace Blog (thalesgroup.com)

Computing architecture evolutions always compliant with Certification regulation (DO178C, DO254, DO297,…)

Federated ➔ Integrated architecture



Federated ➔ Integrated architecture



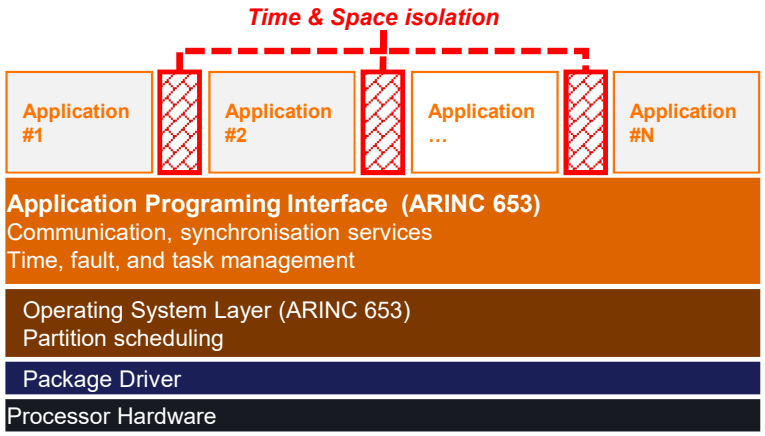
Federated Architecture

- Each function in implemented by its own dedicated computer.
- Need many connections.
- The System's complexity increases when a new function is added.

Integrated Architecture

- Configurable generic computers, that can host several applications (functions).
- Introduction of A664/P7 network allowing wiring/connections reduction.
- Adding a function does not necessarily increase the number of computers or connections : Mastery of the complexity of the system.

Integrated Modular Avionics (IMA)



Level	Failure Condition	Failure Rate
A	Catastrophic	<1 in 10^9 hours of flight
B	Hazardous	<1 in 10^7 hours of flight
C	Major	<1 in 10^5 hours of flight
D	Minor	<1 in 10^3 hours of flight
E	No Effect	

Several applications (Partitions) hosted on the same computer.

➔ Sharing resources

ARINC 653-1 (10/2003)

- Certification
- Determinism
- Functions isolation using partitioning
  - Spatial
  - Temporal

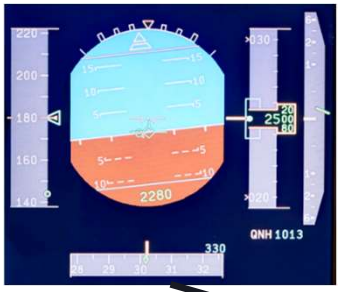
Safety:

- Logical splitting.
- Several applications with different Design Assurance Level (DAL) on the same computer.

Integrated Modular Avionics (IMA)

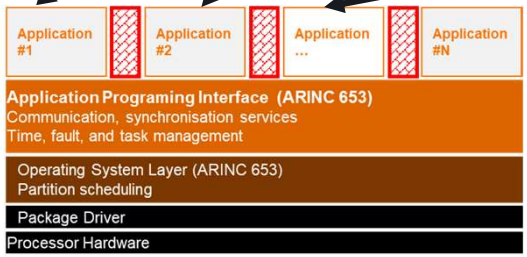
Example of a possible deployment of 3 applications on an IMA module

Primary Flight Display
(DAL A)

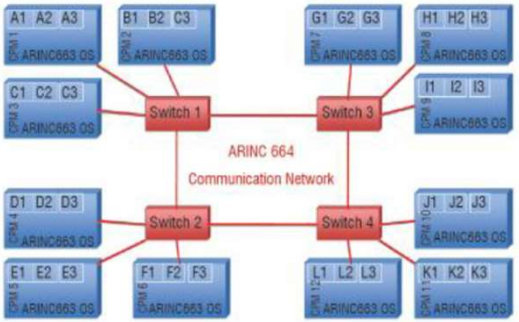


Mission Display
(DAL C)

Flight Management System
(DAL B)



Integrated Modular Avionics (IMA)



IMA allowed to :

- Integrate several applications on the same module
- Reduce the number of equipment in the A/C ➔ Integration
- Reduce the number of equipment types ➔ Modularity and Cost
- Reduce the number of connections ➔ Wiring

Applications' functional scope increases
+ a strong desire to reduce the application's footprint ➔ More and more integration is required

➔ The limits of processing on IMA Single-core architectures are reached.

From Single core to Multicore architecture

Facts

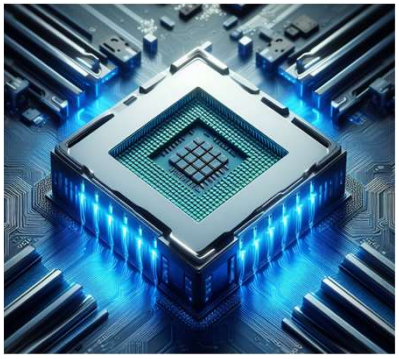
- **COTS SoC (System on Chip) already provide multicore architecture**

  Example : NXP T2081, ARM, Intel TigerLake…

- **Thales HW platforms are based on multicore CPU (example : T2081)**

- **Need to drastically reduce the SWaP-C of our solutions ➔ Performance versus Watts**

➔Use of Multicore capability of our platforms becomes mandatory

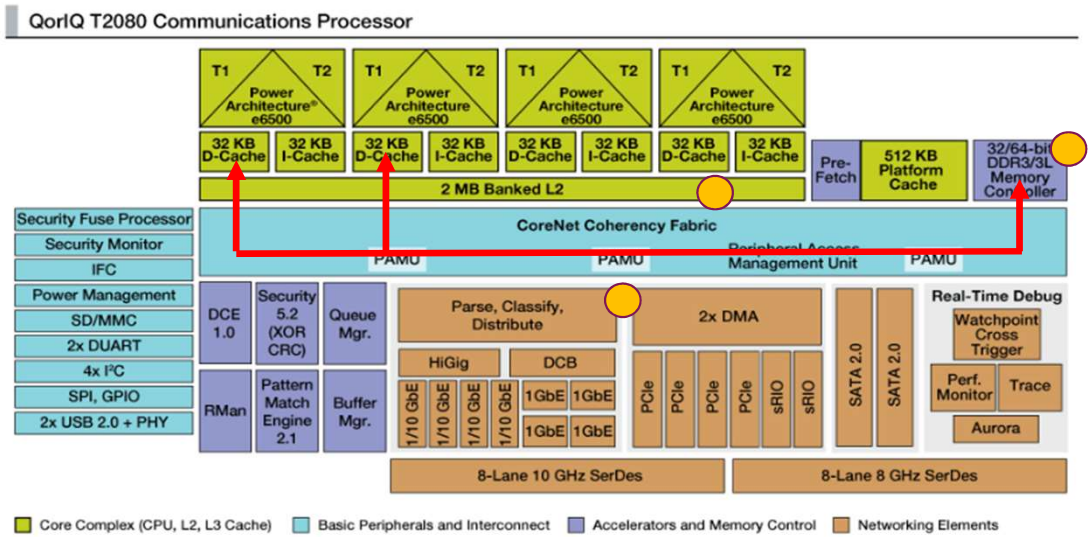


Impacts

- **Extension of Certification objectives (EASA AMC 20.193, 01/2022)**
  - Acceptable means for demonstrating compliance with airworthiness specifications related to multi-core processors (MCPs)
  - Impacts on certification activities depend on the DAL level of the application and the multi core deployment kind.
- Evolution of the ARINC 653 (08/2015) standard
- New capabilities of Real-Time Operating System (RTOS)
- New software architectures :
  - Processing of an application can be allocated to several cores.
  - Several applications can be executed in parallel on different cores.

SWaP-C : Size Weight and Power and Costs

Multicore HW architecture example



T2081 datasheet - The 28 nm QorIQ T2080 and T2081 communications processors (digchip.com)

All interference channels lead to contentions in the processing running in parallel.
Worst Case Execution Times is impacted

➔ Certification authorities published an update of the guidelines

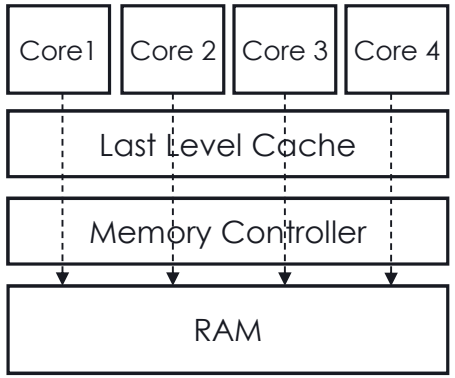
Multi-core: Avionics Regulation AMC 20-193

> **General Acceptable Means of Compliance for Airworthiness of Products, Parts and Appliances (AMC-20), Amendment 23, 21 January 2022**

- AMC 20-193 Use of multi-core processors

Therefore, several software applications and/or hardware functions may attempt to access the same shared resources of the MCP (such as memory, cache, 'coherency fabric' / 'module interconnect', or external interfaces) at the same time, causing contention for those resources.

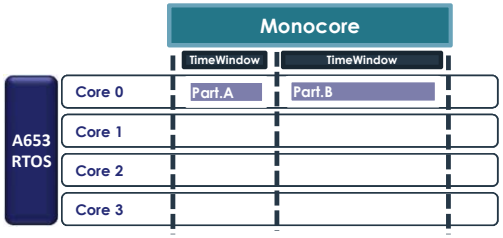
Simplified diagram of concurrent access to shared Memory



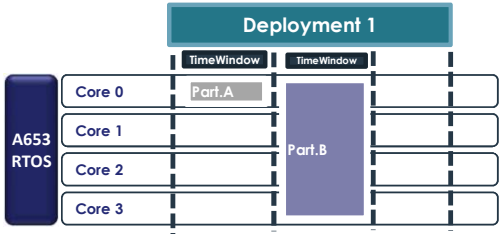
Main focus

- Platform **characterization** : **Interference** channels and **contentions**.
- **Demonstration** of the **Worst Case Execution Time** (WCET) on the multicore architecture (required by DO178C).

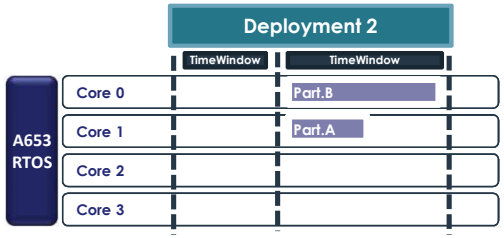
Multicore deployment examples



All partitions run on the same computing core in different Time Windows.
No parallel execution.

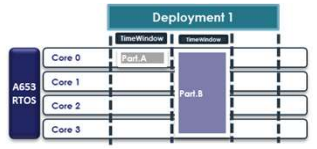


Processing of a single partition is mapped on several computing cores.
This deployment may lead to performance (CPU) impacts due to concurrent accesses to shared resources, parallel processing.
No impacts on other partition allocated to other time windows.

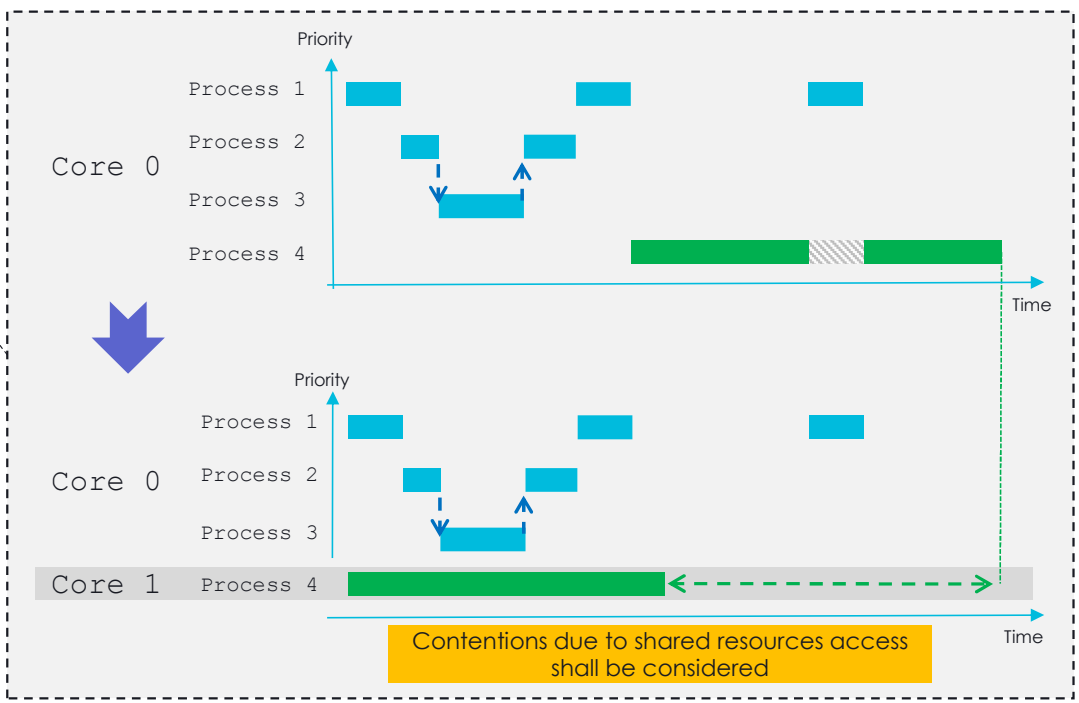


Partitions are allocated to the same Time Window on different cores running in parallel.
This deployment may lead to performance (CPU) impacts due to concurrent accesses to shared resources.

Legacy partition : How to adapt application to multicore

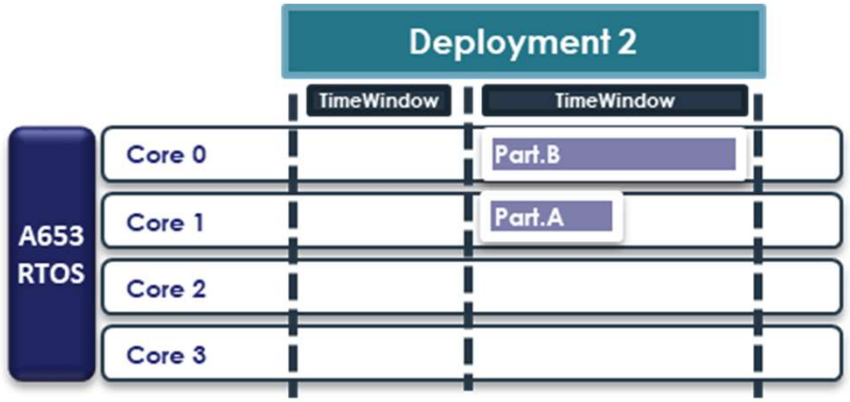


- **Case 1 : Multi A653 process application**
- Strength:
  - No major modification, map A653 process on cores
- Weakness
  - Software design could not be designed to data parallel access
  - Limited to already multithreaded partitions
  - Solution my not be optimal because not originally designed for multicore
- **Case 2 : Mono A653 process application**
- Strength:
  - Optimal multi core solution can be reached
- Weakness
  - Major modification of the legacy application
  - Cost



Legacy partition : How to adapt application to multicore

- **Several partitions in parallel execution**
- Strength:
  - No impact on the partitions architecture: Still Single core from its point of view.
- Weakness
  - Mutual impacts (SW and HW) due to shared resource access shall be analysed and taken into account for the WCET demonstration.
  - Solution may not be optimal



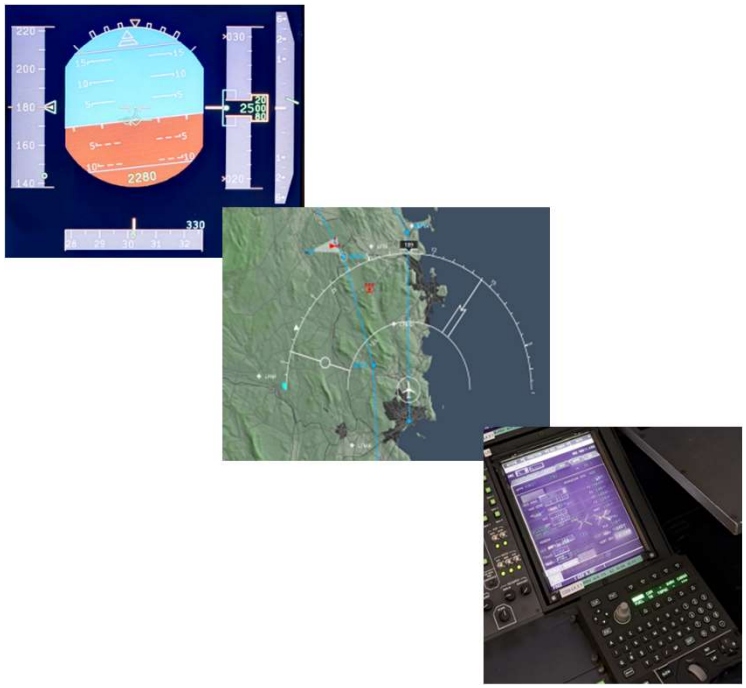
Contentions due to shared resources access shall be considered

Avionics application

- **Sizing**
- From 50 K up to more than 1 million lines of code.
- Application data size in mega bytes.
- Required CPU processing from 1ms to x10ms.
- Execution period from 5ms to 50ms.

- **Complexity**
- Periodic and Asynchronous processing
- Multithreaded architecture
- Sporadic events
- Heterogeneous Input/Output communications (A664-P7/A429/CAN/MM/GPU…)
- Complex hardware (Cache L1/L2 and interconnect)

- **Design Assurance Level (DAL)**
- From DAL D (ex: utility application) to DAL A (ex: Flight controls)
- WCET demonstration



Characteristics of avionics application make the WCET demonstration more tricky in multi core context

Observability and Tooling for WCET demonstration

> Non intrusive Observability

- Allows deep understanding of the software : Timing, branching,…
- No RAM data path usage.
  - Trace Off-Chip : non intrusive profiling traces (Nexus traces extracted through Aurora bus)
- Big data : Raw data require data processing…

> Data mining and presentation

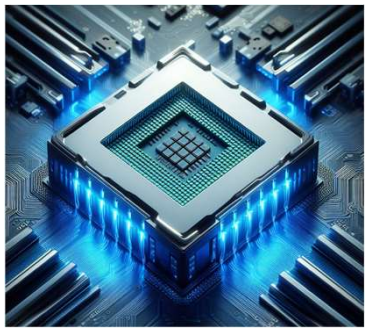
- Performance data shall be extracted from Raw data
  - Interfaced with Non intrusive Observability records
- Performance data shall be summarized
- Graphical synthesis
- Need technologies to explore

Observability tools used for WCET demonstration need to comply with multi core complexity

Synthesis of Main challenges for avionics application

- Adapt platforms (RTOS) from Single to multi core processing in IMA
- Legacy : Adapt existing applications
- New development : Take multi core natively in the design
- Deal with avionics software complexity and size
- Optimize multi core deployment architecture
- Adapt WCET demonstration process, tools and explore new approaches

Beyond multi core architectures



Other accelerators in perspective
for audio/video and optimization processing
GP/GPU, FPGA, TPU…

The WCET demonstration story will continue…

Questions

{OPEN}