www.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Time-randomized Processors for Secure and Reliable High-Performance Computing

David Trilla[†,‡], **Carles Hernández**[‡], Jaume Abella[‡], Francisco J. Cazorla[‡,*]

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

‡

†

CSIC
Consejo Superior de Investigaciones Científicas

*

1st Workshop on Pioneering Processor Paradigms

Austin, TX, US

Time-randomized Processors (TRPs)

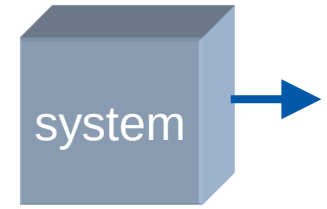
- Proposed to reduce the timing verification costs of critical software using complex processors
  - Autonomous automotive driving systems
  - Unmaned aerial vehicles
  - Industry 4.0

- High performance features challenge worst-case execution time (WCET) estimation
  - Caches
  - Speculation
  - Shared resources

Time-randomized Processors (TRPs)

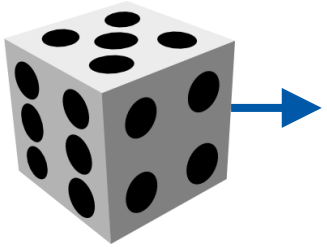
- Control the sources of execution time variability
  - Randomizing → probabilistically
    - Caches replacement and placement policies
    - Shared resources Arbitration policies
  - Upperbounding → when randomizing is not possible
    - Fixed-Latency floating point unit

- In TRPs execution time can be probabilistically modeled



1, 4, 5, 2,… unknown

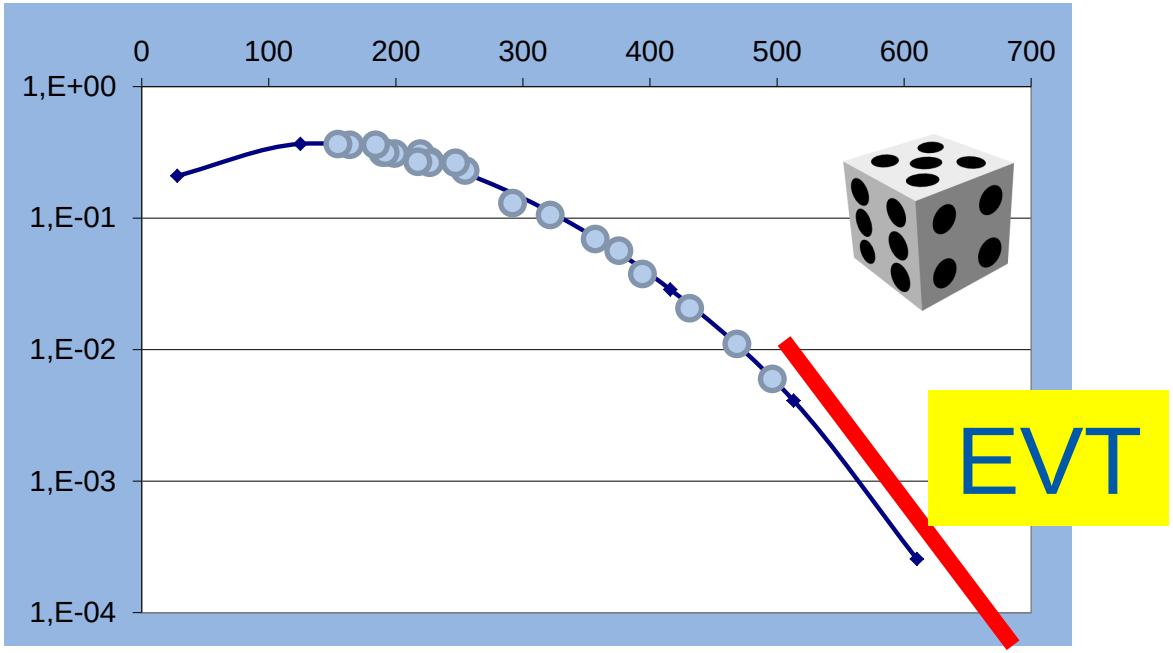
Probability of 1, 1, 1, 1, 1?



1, 4, 5, 2,… $\left(\frac{1}{6}\right)^5$

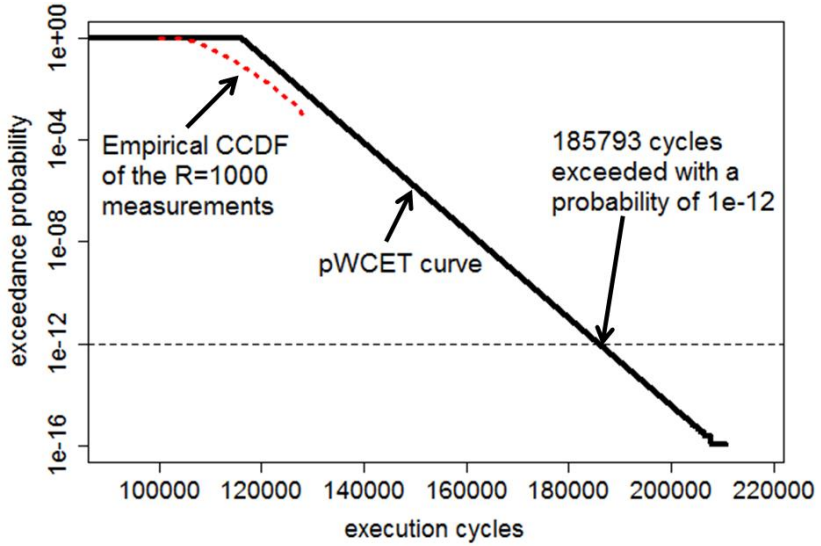
Time-randomized Processors

- Timing verification step → collecting measurements
- Probabilistic WCET → Extreme Value Theory (EVT)
  - I.I.D properties
  - Representative measurements



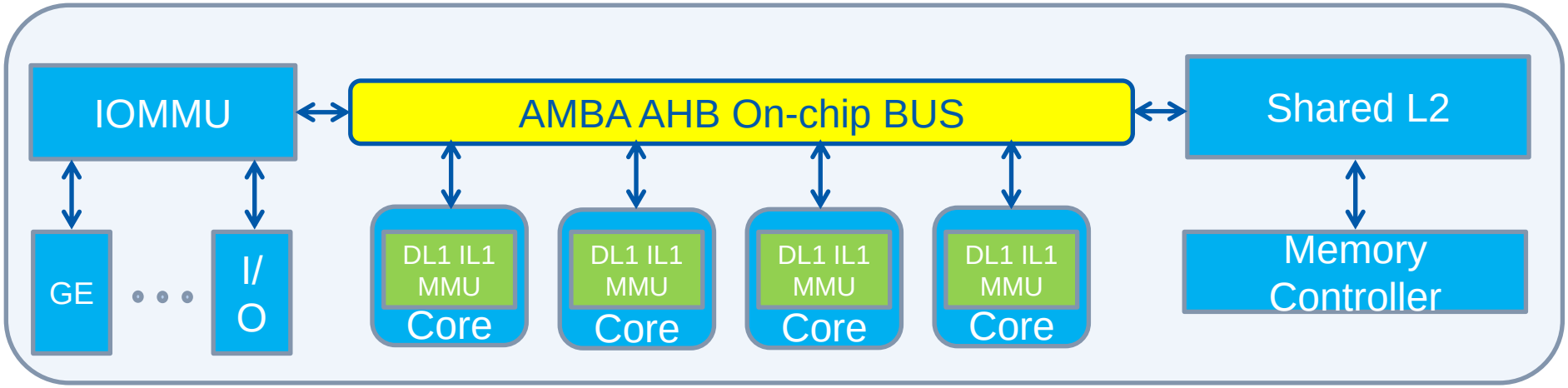
Extreme Value Theory (EVT)

- Considers the system as a black box
- Technique to derive the combined probability of appearance of those events observed (captured)
- Cannot predict those events that are not observed
- EVT must be fed with *meaningful* (**representative**) observations data from the execution of that program at operation → derived bounds hold at operation



Time-randomized Processors (TRPs)

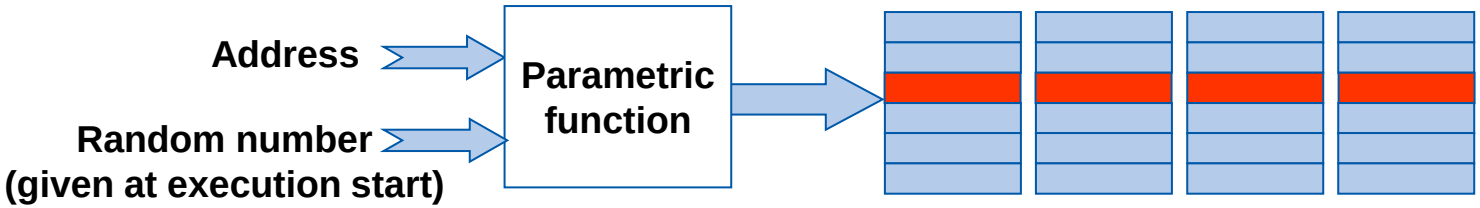
- Example: LEOPARD processor (joint Gaisler-BSC work)



  - Leon-based Multicore
    - Simple for HPC ←→ Complex for critical applications
  - Modifications
    - Random Replacement and Random placement in Caches
    - Random permutations arbiter in the bus and memory controller
    - Worst-latency FPU
  - Can be enabled/disabled selectively

Architectural Modifications

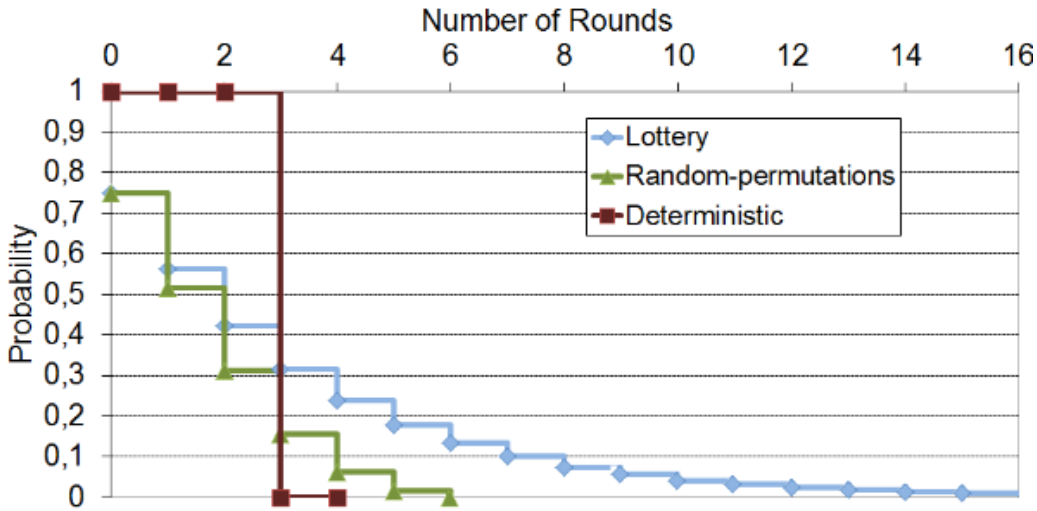
- Randomized Cache Designs
  - To make cache conflicts behave probabilistically and remove pathological cases
  - Random replacement
  - Random placement
    - Covers probabilistically conflicts due to variations in the memory mapping



Architectural Modifications

- Random Arbitration in the on-chip bus and memory controller
  - Round-robin is time analysable but forces assuming the worst-possible alignment of requests
    - In a 4 core multicore → 3 arbitration rounds is the only safe bound
  - Random arbitration
    - Requests aligment is captured probabilistically
      - Lottery

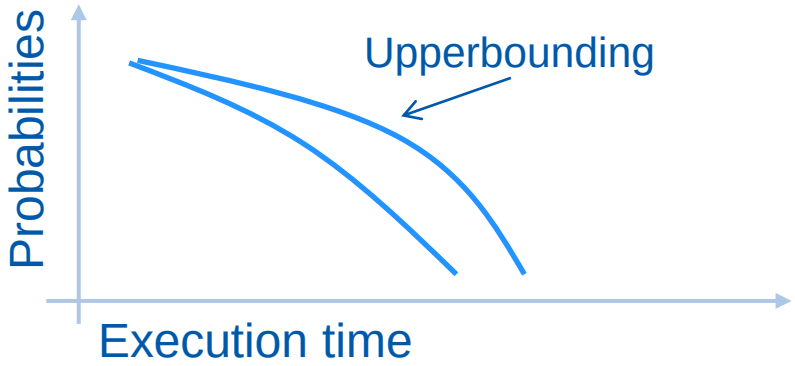
      - Random permutations



Architectural Modifications

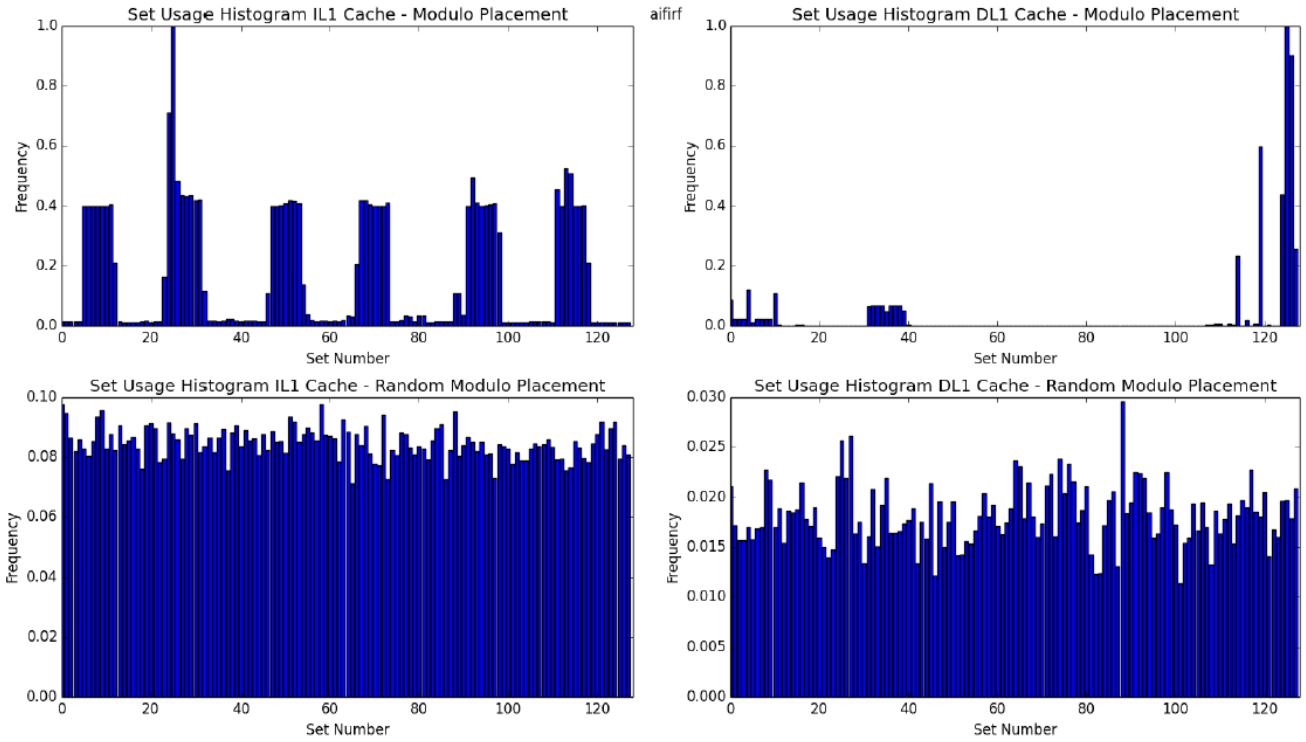
- Worst-Latency Floating Point Unit (FPU)
  - In general the end-user cannot reason about the operated values
  - Delay upperbounding is the only safe assumption

- LEON FPU (FDIV, FSQRT)
  - FDIV latency: 15-18 cycles
    - Enforce always 18
  - FSQRT latency: 23-26 cycles
    - Enforce always 26
  - End user does not need to control input values



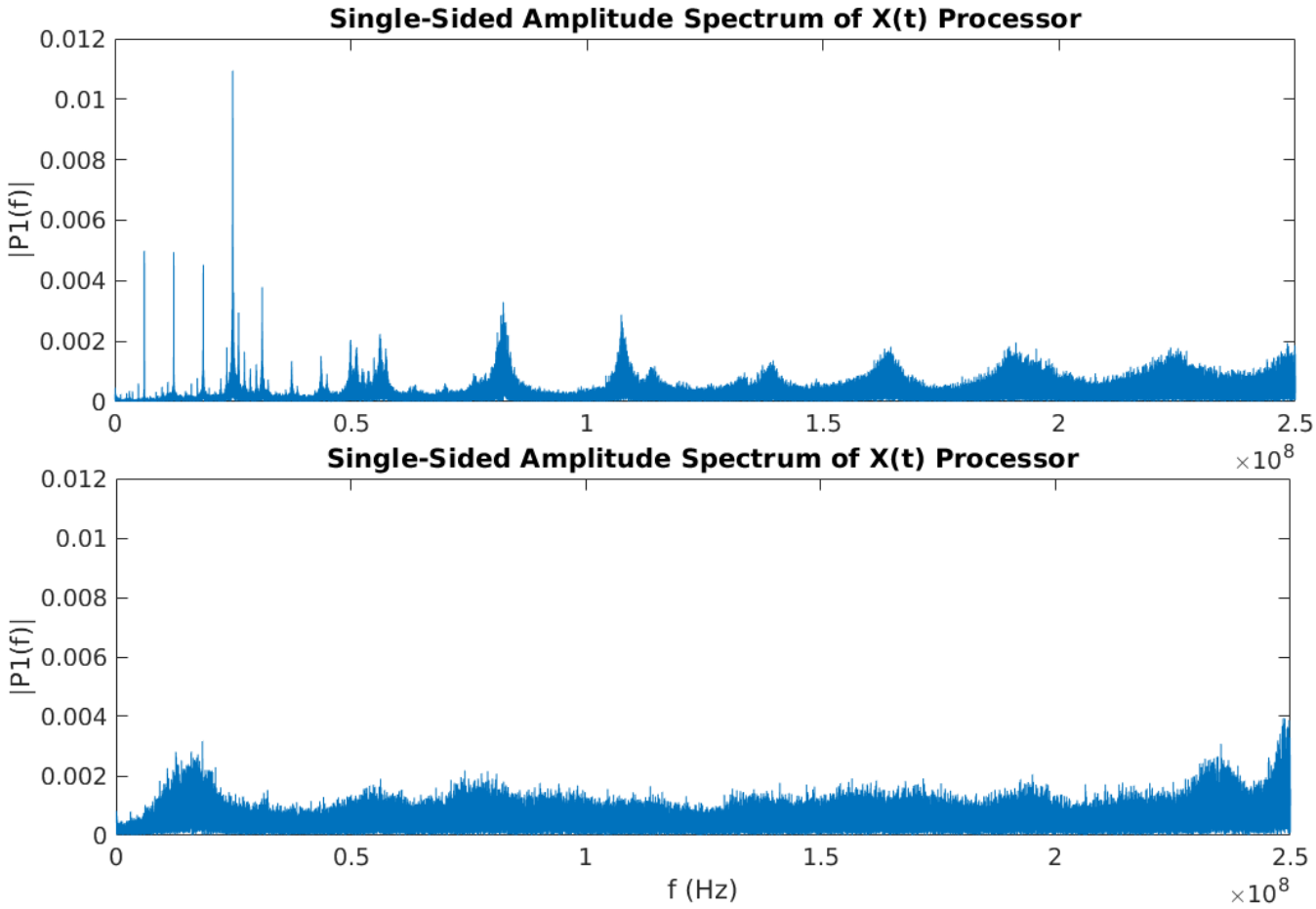
Enhanced Reliability Properties

- Example: Aging effect reduction in caches
  - Hot carrier injection effects are proportional to utilization
  - Randomized caches provide uniform set access distribution



Enhanced Reliability Properties

- Power/Performance stability
  - Pathological cases occur with a quantifiable (low) probability



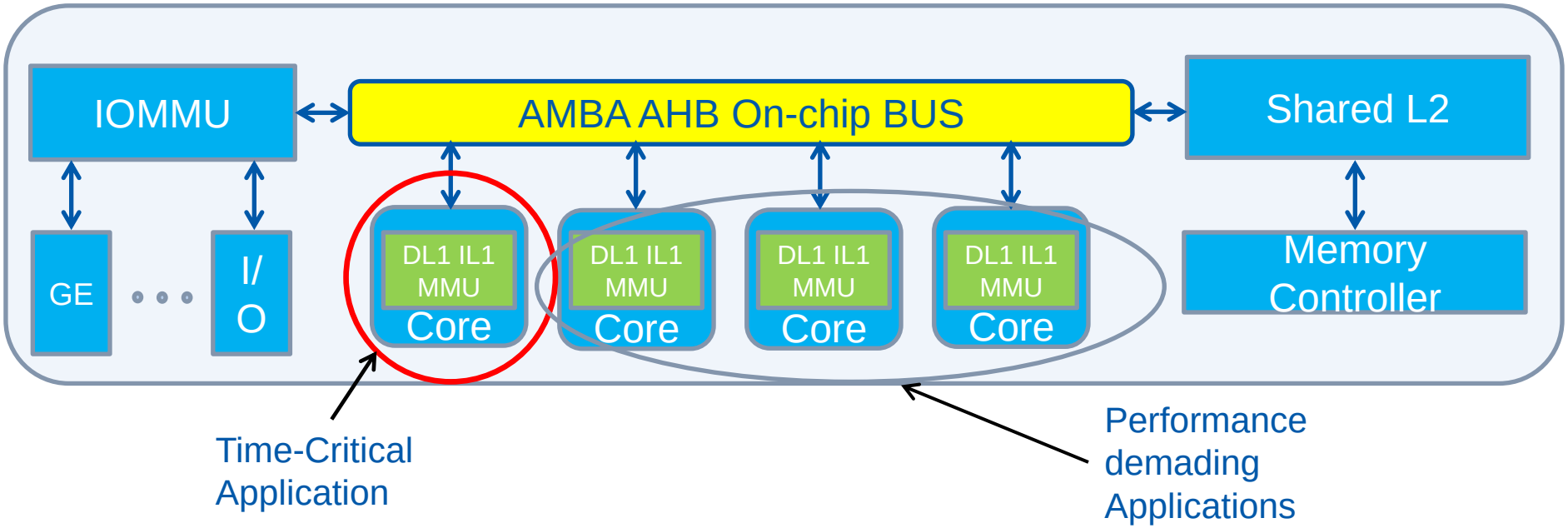
Enhanced Security Properties

- Randomization techniques has been proposed for Security
  - Caches
    - Layout randomization
    - Random fill cache
  - Pipeline
    - Random stalls insertion
- Randomization protect the processor against :
  - Cache side channel attacks
  - Power analysis attacks

- TRPs → Inherit security properties from randomization
- Randomization security techniques do not meet TRPs requirements

Time-randomized Processors for HPC

- What they can offer?
  - Guaranteed Performance at a low cost



  - Enhanced Security and reliability properties

Time-randomized Processors (TRPs) for HPC

- TRPs reduce verification effort required to test extreme situations
  - With deterministic processors a huge effort is required to trigger worst-case conditions (e.g power viruses)
    - Current processors complexity is very high
    - Post-silicon validation is thorough process

  - In TRPs extreme events will emerge naturally → reducing the complexity of the verification process
    - An arbitrarily low probability can

    be attached to the non-observed events

    →avoiding safe guardbands

    for the unfeasible situations

Time-randomized Processors (TRPs) for HPC

- With TRPs extreme events occur with a given probability
  - The architect is provided with a solid argument
    - Extremely low probable events (e.g below meteorit impact) can be simply discarded
    - Having quantifiable events helps finding the most adequate safety measurement (as for random hardware failures)
  - Extreme value theory can be employed to argue about extreme power, temperature, execution conditions


BSC
Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación
QUESTIONS?