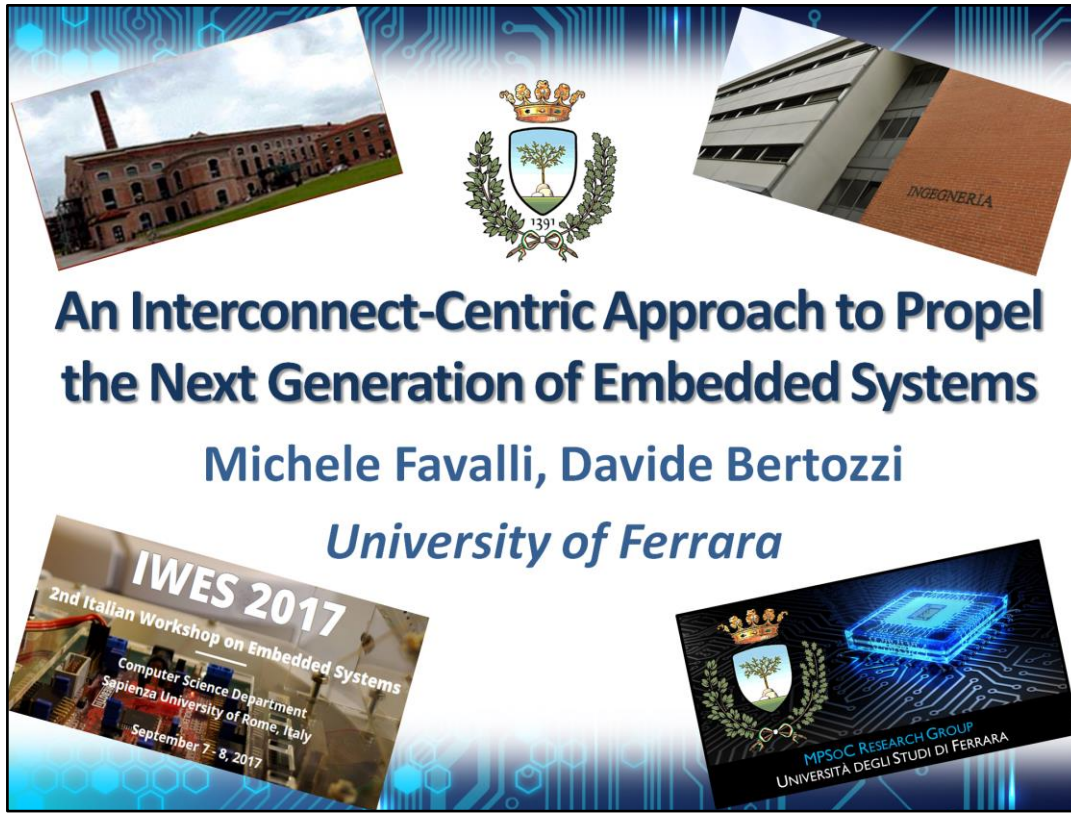
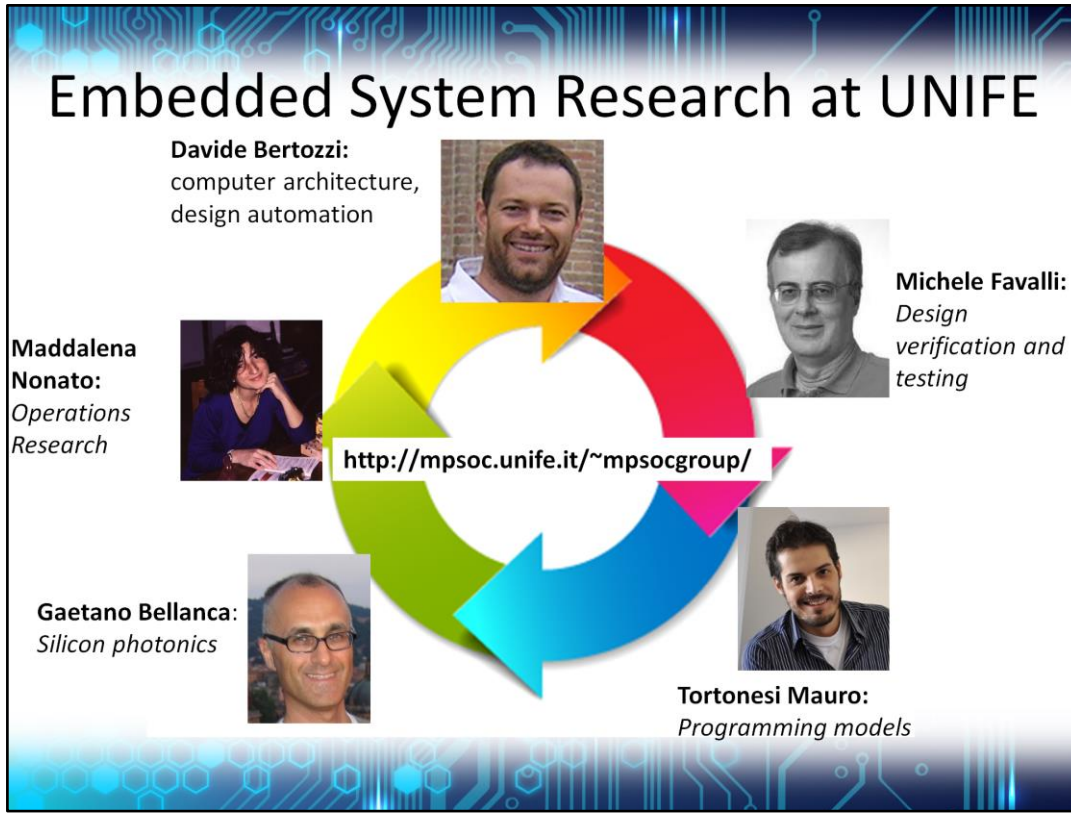
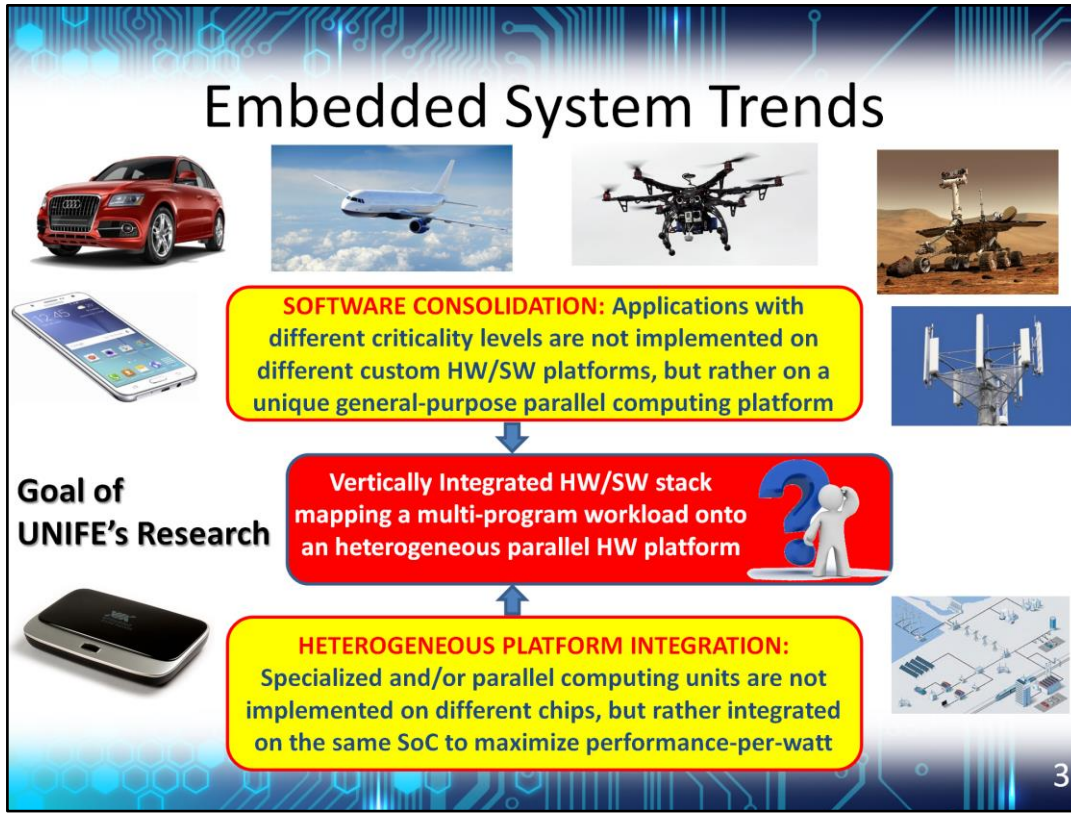


Ladies and gentleman good morning, I am presenting the research of Ferrara on embedded systems
This research is based on an interconnect centric approach for the next generation of embedded systems



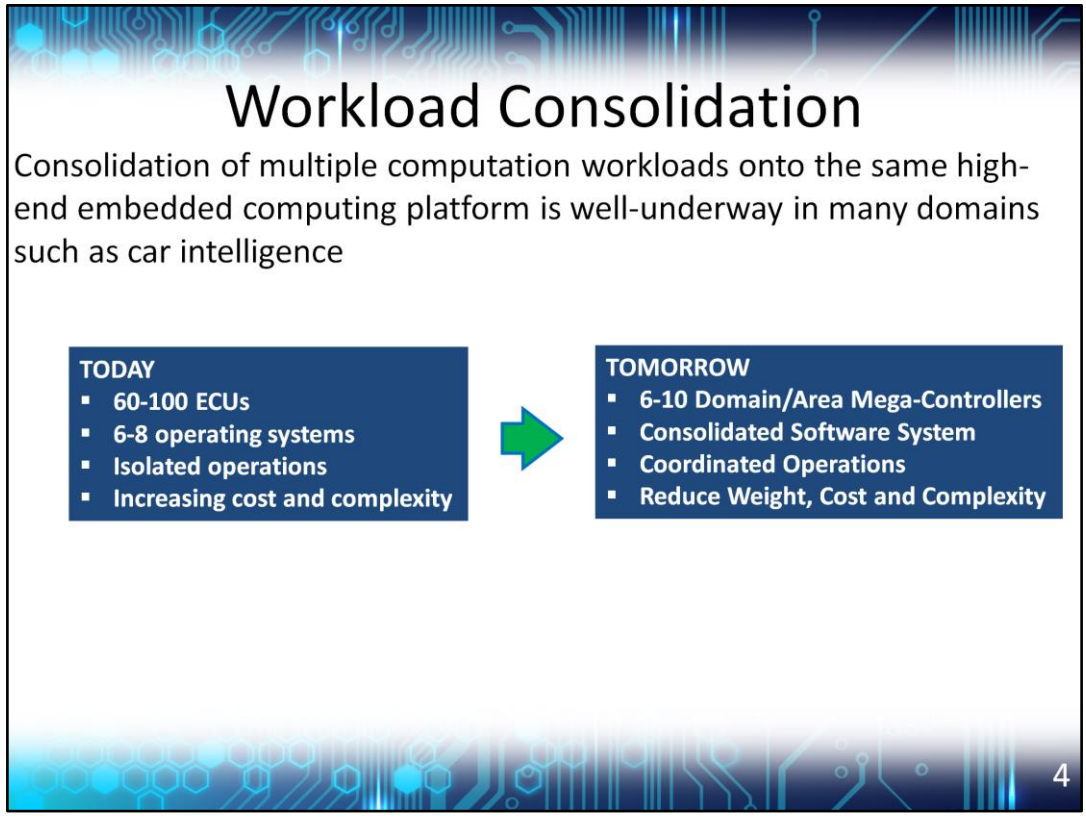
The research is coordinated by Davide Bertozzi and is based on an interdisciplinary approach that,
starting from a computer architecture and design automation core, includes software, photonics, operations research and
design verification and testing



One trend in embedded computing is the move from architectures based on several custom computing elements to a general purpose computing platform while providing a programming path that does not require fundamental changes for software developers.

Such a computing platform will be integrated on a single SOC to maximize performance and typically features heterogeneous
modules to satisfy the different kinds of criticality levels of the embedded applications

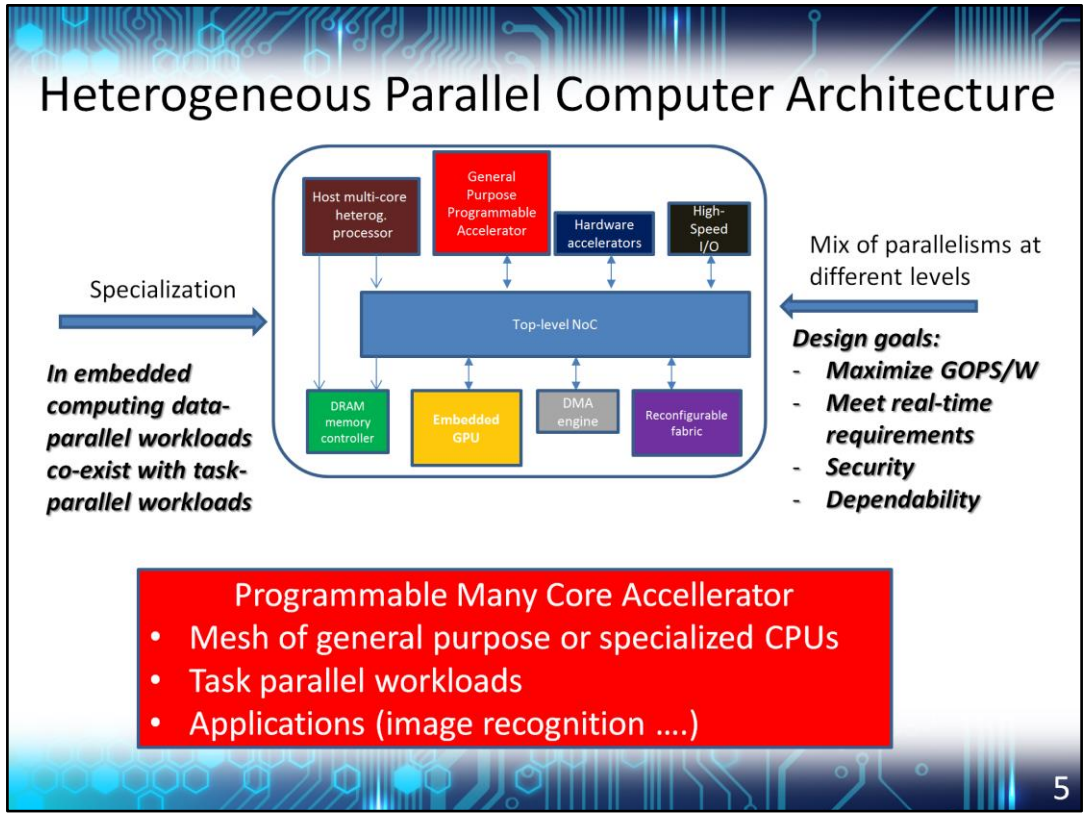
In this context, the goal of UNIFE research is to support a vertically integrated HW/SW stack where a multi-program workload is
mapped on a single HW platform integrating heterogeneous modules in a parallel environment



As an example of the aggregation of multiple workloads on a few high-end computing platforms is given by car intelligence that today features a complex network of several specialized and physically separated electronic control units which in turn present also relevant differences in software support due to the use of different operating systems

It is rather evident that this approach is leading to increases in cost and complexity

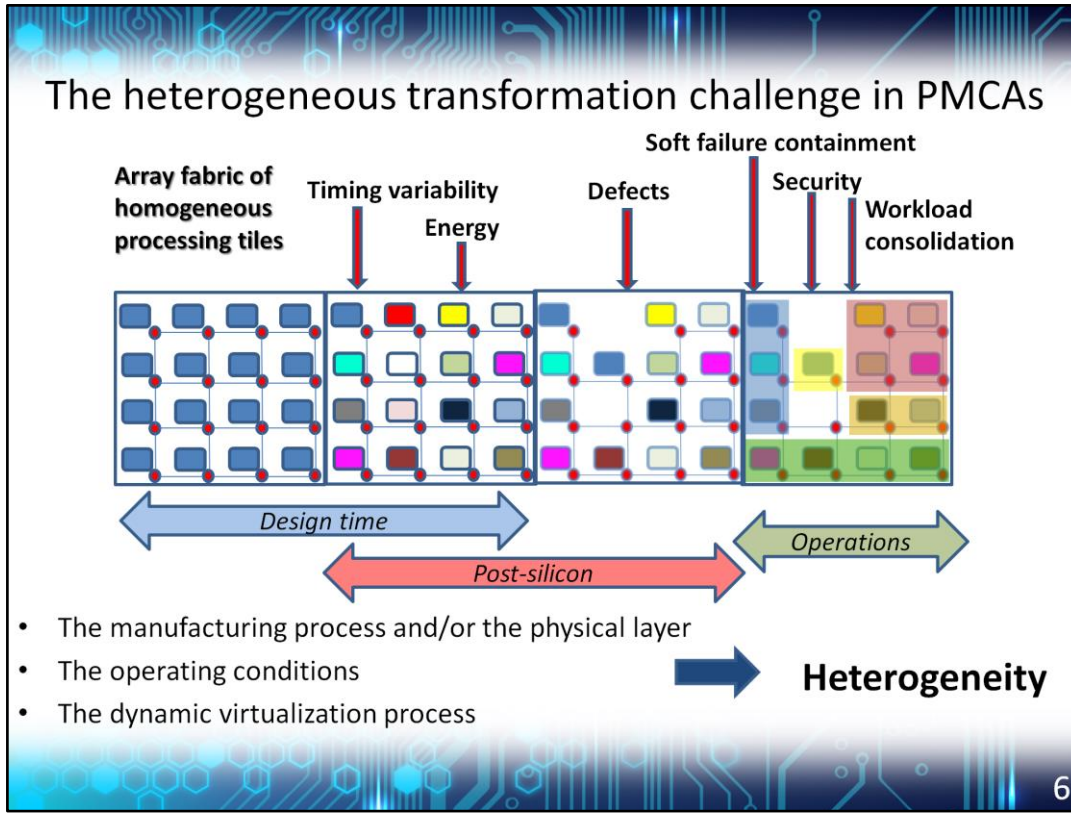
The expected trend, instead, is given by the use of up to 10 domain and area high-end controllers with a consolidated software system that is expected to reduce weight, costs and complexity



These new computing platforms are expected to be heterogeneous by aggregating data level (GPUs) parallel accelerators, hardwired and reconfigurable accelerators and peripheral controllers under the management of a multi-core host processor. A possibly multilevel Network on Chip provide communications

The designer's goals for this kind of devices are the optimization of giga ops per watt, the satisfaction of real-time requirements possibly under security and dependability constraints.

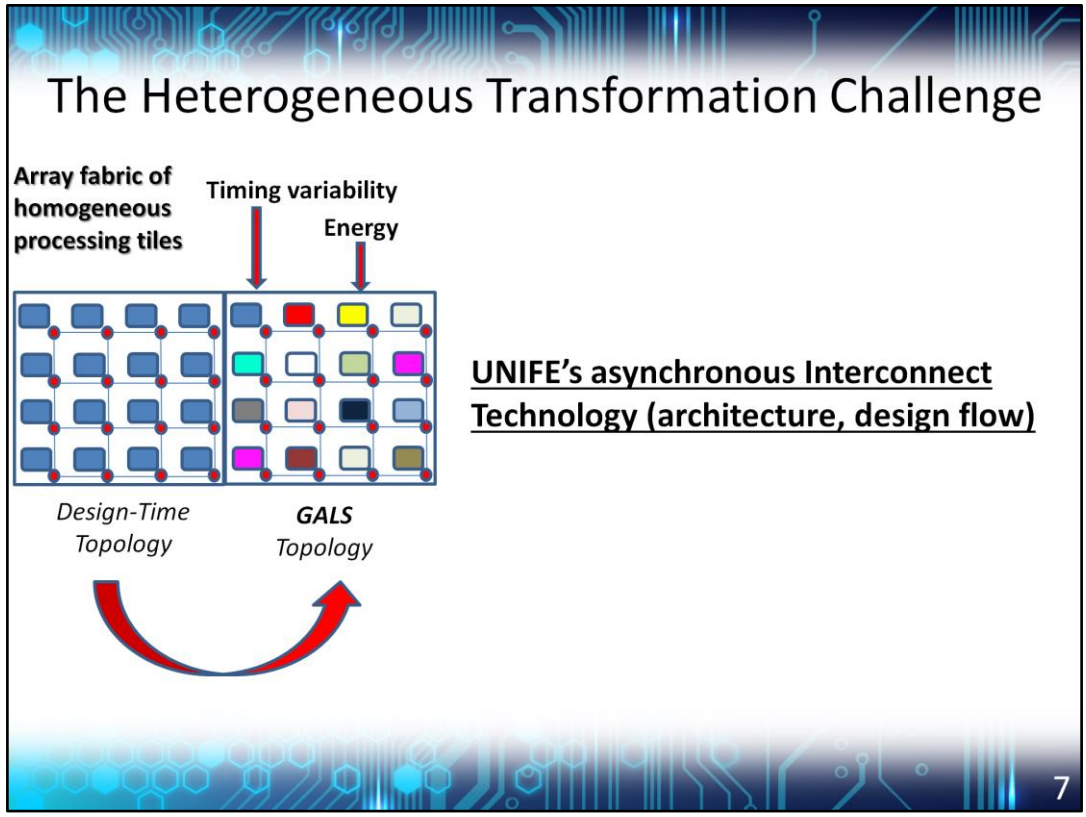
In this kind of devices one emerging trend is the co-existence of traditional data parallel workloads with task-parallel workloads (image recognition) that is expected to be satisfied by programmable many core accelerators



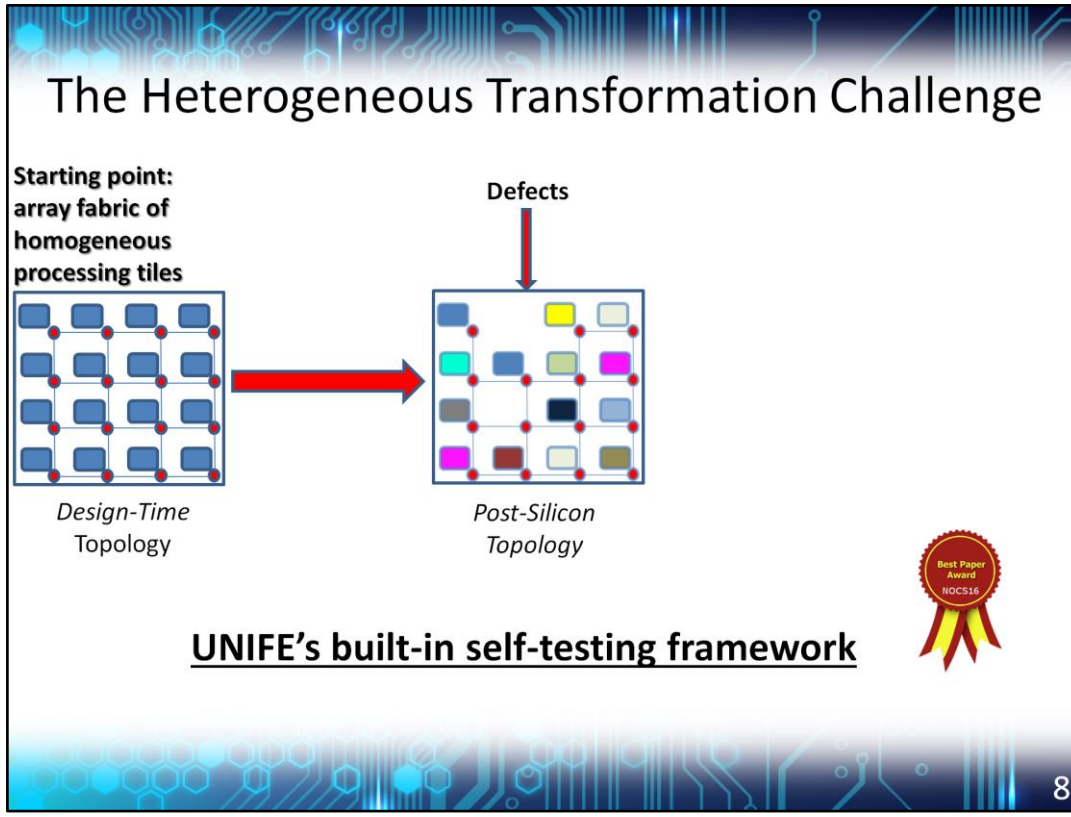
From an abstract architectural point of view, programmable many core accelerators are designed as an array fabric of homogeneous processing tiles, but at either design time or in the post silicon phase they face an hetereogeneous transformation.

For instance, energy and dissipation budgeting at design time may impose different clocking and power supply choices to different tiles. The same may occur also in the post silicon phase when tiles are tested with respect to timing variability problems.Inaddition, testing for hard defects may require the deletion of tiles or interconnect components.

Finally, there is also a run time heterogeneity due to workload consolidation in a dynamic virtualization process on different partition tiles. This last process ensures computing efficiency but also to soft failure containment and security.

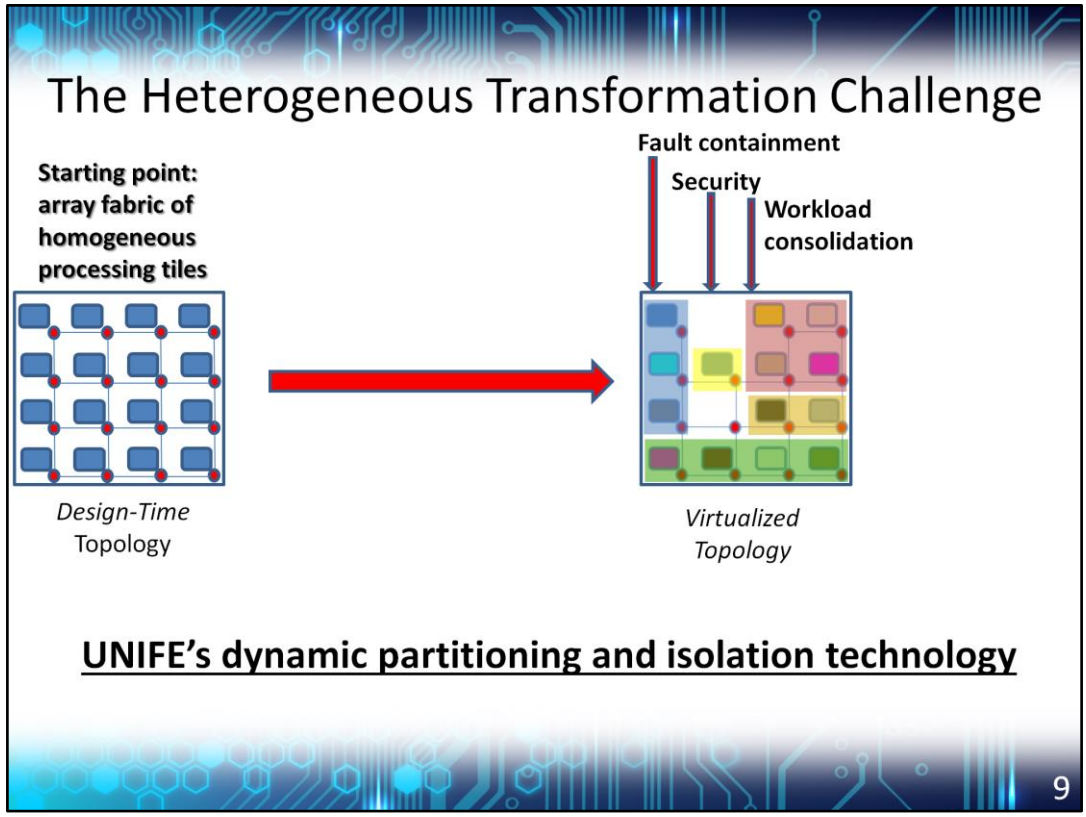


In this context, the university of Ferrara proposes an asynchronous interconnect technology to support a PMCA designed under the Globally Asynchronous Locally Synchronous paradigm that well support a mesh featuring tiles working at different clock frequencies to account for timing variability or energy considerations.



An additional cause of heterogeneity is given by the need to maximize yield by detecting and insulating defective tiles and network components

In this case, UNIFE developed a self-testing framework for asynchronous network on chips that received a Bets Paper Award at ASYNC 2016.



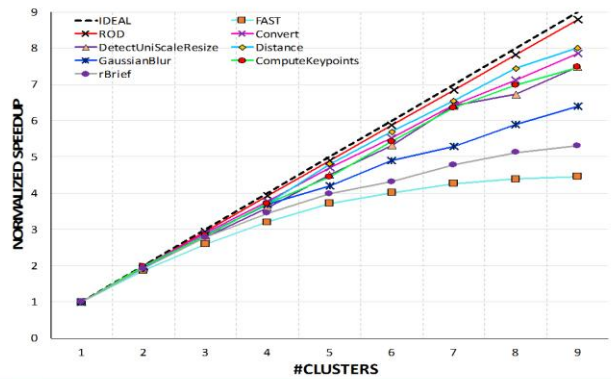
The final step in the proposed approach is the support to dynamic partitioning and isolation

Time-Division Multiplexing?

TDM is the most straightforward way of consolidating workloads onto the PMCA. Are embedded applications parallel enough for this?

Image Processing **benchmarks** executed on a gem5-based PMCA simulator

9 clusters of 1 core each. ***Ideal NoC and 1 cycle memory access latency***



Instead of exploiting a marginal speedup, a better option may be to allow multiple offloaded tasks to co-exist in the PMCA at a given time

10

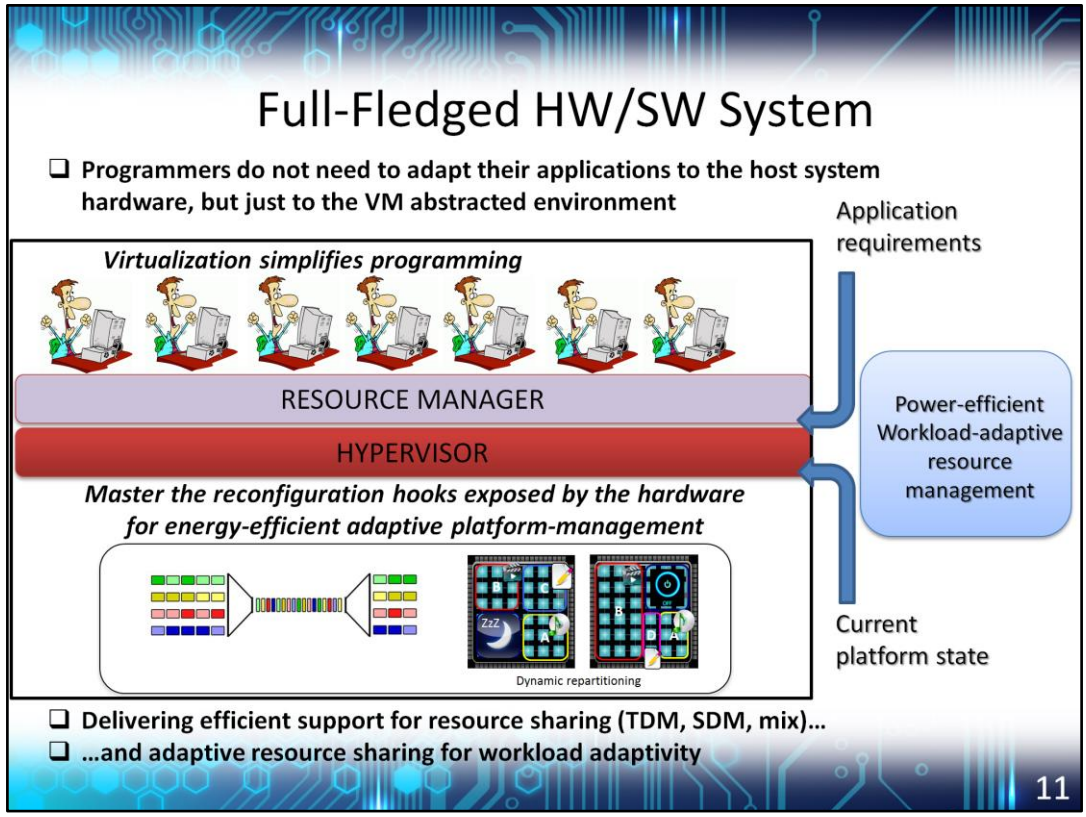
The push toward dynamic partitioning is based on an analysis of time division multiplexing in the use of PMCAs where tasks are sequentially scheduled for certain time slots

This approach requires to save the context and to switch between contexts (management of registers, stack and of the heap) – this is a typical approach for Graphics Processor Units which, however, refer to a SIMD paradigm different from that of PMCA

The University of Ferrara investigated whether typical PMCA applications are parallel enough for this approach to result effective. In this case, a PMCA simulator has been used to evaluate the performance of a 3x3 mesh for a set of image processing benchmarks.

The results show that in some benchmark, the performance do not scale linearly with the number of processing tiles, thus meaning that the TDM approach may not efficently use some slot. The problem is expected to grow in the presence of workloads coming from different application domains.

In the view of UNIFE, a better option may be to allow multiple tasks to coexist in PMCA at a given time



From the point of view of the hardware software-interface, the view of the University of Ferrara is that programmers should not adapt their applications to the system hardware, but just to the Virtual Machine environment by specifying not only functionality but also other requirements (latency, bandwidth….) to a resource manager and hypervisor unit that provide a power efficient workload adaptive resource management that determine the current platform state

Thank you for your attention