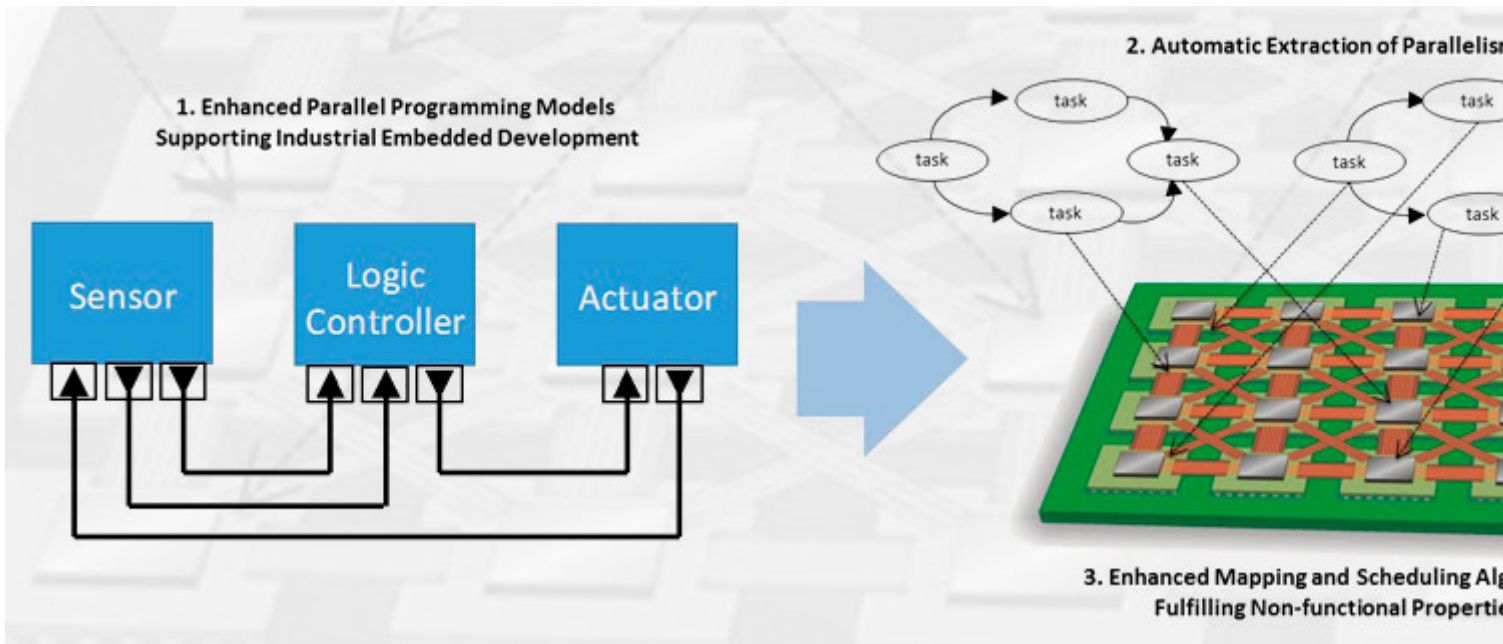# Time-predictable parallel programming models



To exploit the computation capabilities of next-generation many-core embedded architectures, it is mandatory to tame the complexity of parallel programming, exacerbated by the cyber-physical interactions of embedded systems that imposes extra non-functional constraints such as time-predictability.

# Summary

Parallel hardware architectures are rapidly evolving, with parallel processors becoming mainstream in all computing domains, from embedded to high-performance computing systems. However, software development is not evolving as fast as parallel hardware, and taking advantage of the potential extra performance of these architectures still requires dealing with the extra complexity of the parallel execution.

This evolution disparity of (parallel) hardware and software is exacerbated by the advent of next-generation many-core embedded heterogeneous architectures. In order to exploit the massively parallel computation capabilities of these architectures, it is mandatory to tame the complexity of parallel programming. In that respect, parallel programming models become a vital element to provide the desired performance while providing better programmability.

Despite the efforts of simplifying the programmability of many-core architectures, the performance obtained can significantly change depending on the parallelisation strategy selected, having an unpredictable impact on the timing behaviour. In general, analysing the data-flow of a program can be very difficult due to the uncertainty about the exact moment parallel parts execute. Indeed, the complexity of parallel programming

has already been identified as a major challenge in general purpose computing, and it is now magnified in the embedded systems domain, due to cyber-physical interactions and non-functional properties, such as energy-efficiency, time-predictability and dependability.

Therefore, despite many-core architectures can provide the required performance for cyber-physical systems, they are still far from being efficiently used due to the extra complexity of parallel programming.

# Objectives

In this research front we pursue the following objectives:

1. Integrate current industrial embedded software development practices (e.g. using component-based approaches) with current parallel programing models provided by the newest COTS many-core embedded platforms.
2. Develop novel compiler techniques and tools to automatically extract the parallelism existing on applications.
3. Develop a timing analysis methods capable of expressing the timing implications of parallel applications and potential resource-contentions of task-to-core mappings

Barcelona Supercomputing Center - Centro Nacional de Supercomputación