# A Low-Power Multicore Architecture at Work

Stefano Bernabovi*, Florent Bruguier†, Michael Chapman*, Abdoulaye Gamatie†, Thierry Gil†,
David Kerr-Munslow*, Philippe Naudin*, and Gilles Sassatelli†
*Cortus SAS, France.
†LIRMM (CNRS/Univ. de Montpellier), France

***Abstract*—This demo is dedicated to the preliminary version of an energy-efficient multicore architecture built with inherently low-power IP cores from Cortus SA, one of the world-leading semiconductor IP companies in the embedded domain. Among possible applications that can benefit of such an architecture are image or signal processing workloads. A key feature of the architecture is heterogeneity which makes it possible to exploit the specific characteristics of the integrated cores. An FPGA prototype of the architecture is used for this demo.**

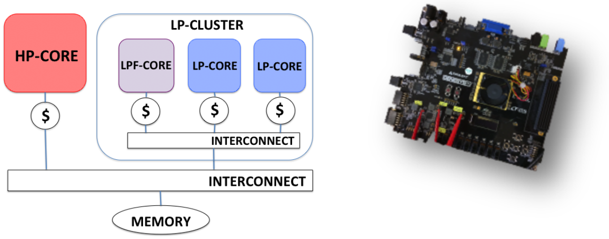
## I. Introduction

Heterogeneous computing usually refers to systems including various processing elements so as to meet both performance and power-efficiency requirements. Typical heterogeneous architectures combine CPUs and compute accelerators such as GPUs. While the former are well-suited for executing sequential workloads and the operating system, the latter are rather devoted to massively regular parallel workloads.

We have devised a novel asymmetric multicore architecture comprising two execution islands: parallel and sequential. While the former is devoted to highly parallelizable workloads for high throughput, the latter addresses weakly parallelizable workloads. Accordingly, the parallel island is composed of many low power cores and the sequential island is composed of a small number of high-performance cores. An originality of our proposal lies in the use of the cost-effective and inherently low power core technology provided by Cortus [1]. These cores are highly energy (MIPS/$\mu$W) and silicon efficient (MIPS/mm$^2$) compared to existing technologies. We believe the massive usage of such embedded cores deserves attention to design energy-efficient architectures required for high-performance embedded computing. The architectural solution promoted here is similar to the CPU/GPU heterogeneous design paradigm. However, an important difference is that the parallel island, which plays the same role as the GPU, can deal with both regular and irregular parallel workloads. In addition, both sequential and parallel islands support the same programming model, facilitating the job of programmers. GPUs require specific APIs such as OpenCL and CUDA, which are not necessarily supported by CPUs and always require extensive software support. Compared to ARM big.LITTLE architectures, which considers only application processors (APs) capable of running full-fledged operating systems, we here combine application processors on the ”big” side and micro-controllers on the ”LITTLE” side. Such compact ”LITTLE” cores (which are not intended to support a full OS) are key for aggressive energy optimization.

Another important trade-off considered in our solution is the support of floating point arithmetic, which benefits certain operations in embedded applications, e.g., matrix inversion required for Multiple Input / Multiple Output (MIMO) decoding and FFTs, which often suffer from scaling problems in fixed point. As floating point units (FPUs) can be expensive in terms of area and power in the very low power cores being considered, it will be supported only by a subset of these cores.

The 4-core architecture depicted in Fig. 1(a) comprises one high-performance core, referred to as HP-Core; and three low power cores, i.e., micro-controllers, with various features: one with floating point unit (LPF-Core) and two without (LP-Core). This provides a diversity for meeting various application requirements. Indeed, floating point operations are not always required in embedded workloads. All four cores are connected to a shared memory via a hierarchy of crossbars as illustrated in Fig. 1(a).



(a) Architecture sketch

(b) Genesys FPGA board

Fig. 1: Proposed multicore architecture and experimental board

A tailored lightweight and flexible multi-thread execution model has been also defined in order to enable the management of programs executed on the proposed architecture, which is synthesized on FPGA (see Fig. 1(b)). Some simple image processing-inspired workloads are experimented on this prototype, according to different core configurations of the architecture in order to highlight the performance / energy tradeoffs.

## Acknowledgements

This work has been partially funded by the French ANR agency under the grant ANR-15-CE25-0007-01, within the framework of the CONTINUUM project.

[1] http://www.cortus.com