Deliverable D4.1 – State of the art on performance and power estimation of embedded and high-performance cores

Anastasiia Butko, Abdoulaye Gamatié, Gilles Sassatelli, Stefano Bernabovi, Michael Chapman, Philippe Naudin

CONTINUUM

Project Ref. Number ANR-15-CE25-0007

D4.1 – State of the art on performance and power estimation of embedded and high-performance cores

Version 2.0
(2016)
Final

Public Distribution

Main contributors:
A. Butko, A. Gamatié, G. Sassatelli (LIRMM); S. Bernabovi, M. Chapman and P. Naudin (Cortus)

Project Partners: Cortus S.A.S, **Inria**, **LIRMM**

Project Partner Contact Information

Cortus S.A.S Michael Chapman 97 Rue de Freyr Le Génésis 34000 Montpellier France Tel: +33 430 967 000 E-mail: michael.chapman@cortus.com	**Inria** Erven Rohou Inria Rennes - Bretagne Atlantique Campus de Beaulieu 35042 Rennes Cedex France Tel: +33 299 847 493 E-mail: erven.rohou@inria.fr
LIRMM Abdoulaye Gamatié Rue Ada 161 34392 Montpellier France Tel: +33 4 674 19828 E-mail: abdoulaye.gamatie@lirmm.fr	

Table of Contents

Executive Summary

The goal of the CONTINUUM project is to define a new energy-efficient compute node model, which will benefit from a suitable combination of efficient compilation techniques, emerging memory, and communication technologies together with heterogeneous cores. The originality of the solution promoted by the project is to consider the core technology of the Cortus partner.

The current deliverable presents a number of candidate core technologies, mainly from Cortus and ARM. Performance and power consumption numbers are given as an assessment of all these technologies. The outcome of the present survey will serve in choosing the suitable core technologies in the expected heterogeneous architecture.

1 Introduction

In embedded computing, while the systems power budget is still confined to a few watts, the performance demand is growing. This comes from the continuous integration of new functionalities in systems, e.g. in mobile computing. To address this demand, the number of cores in systems has been increasing. At the same time, in the HPC domain, supercomputers are expected around 2020 to achieve 10^{18} floating-point operations per second (FLOPS) also referred to as exascale computing, within a power budget of 20MW [6]. With current technologies, such a supercomputer would require a similar power budget to that of a European mid-size city, therefore calling for new design solutions. These observations from both embedded and HPC domains draw their convergence towards finding the best ratio between performance and power consumption, i.e., energy-efficiency.

One solution consists in using embedded technologies in HPC systems in order to take advantage of their inherent low power consumption. This is the vision considered in the European MontBlanc project [38]. The project developed a prototype of large-scale HPC architecture integrating ARM embedded cores for energy-efficiency. In [34], the scalability and energy-efficiency of three multiprocessor-on-chip (MPSoCs) within compute clusters are evaluated. These MPSoCs are PandaBoard, Snowball and Tegra. They all contain ARM Cortex-A9 processors. In [5], a similar study is reported, which aims to assess the possible benefits of ARM core-based clusters compared to those relying on commodity processors such as x86, for HPC.

Another study [33] compared ARM-based clusters against Intel X86 workstation, by evaluating both their energy-efficiency and cost-efficiency. The reported experiments showed that the ARM clusters enable a better energy-efficiency ratio against the Intel workstation, e.g. up to 9.5 for in-memory database, and around 1.3 for Web server application. Note that the relevant measurement of the power consumption in the addressed systems highly depends on the reliability of the applied data collection tools. In [35], a platform-independent tool is devoted to this aim while targeting both homogeneous and heterogeneous systems. Such a tool a worth-mentioning for our forthcoming studies.

This deliverable presents an assessment of selected candidate core technologies, with relevant features to be explored for the compute node architecture targeted in the CONTINUUM project. Usual design assessment techniques rely on flexible system descriptions at different abstraction levels for a comfortable design space exploration [29, 30]. The techniques can follow general modeling paradigms, e.g. UML [19, 16, 37, 4], analytical modeling [11, 3, 2], transaction-level modeling[39, 27, 28, 31], cycle-accurate or cycle-approximate modeling [7, 10, 8, 9], and ultimately hardware prototyping [45].

In this work, we consider hardware prototypes as a baseline to evaluate the identified key metrics. As stated in the project proposal, the core technologies from the Cortus partner are given high attention in this project, as they are inherently energy-efficient. They offer a set of cores with different capabilities in terms of performance and power consumption tradeoff. This opens an interesting opportunity for building multicore heterogeneous architectures in which cores can be selected for execution depending on workload nature, so as to reduce as much as possible the dissipated energy while meeting the performance requirements [44, 26].

An existing similar heterogeneous multicore architecture is ARM big.LITTLE [22]. It basically consists of two clusters of cores as illustrated in Figure 1.

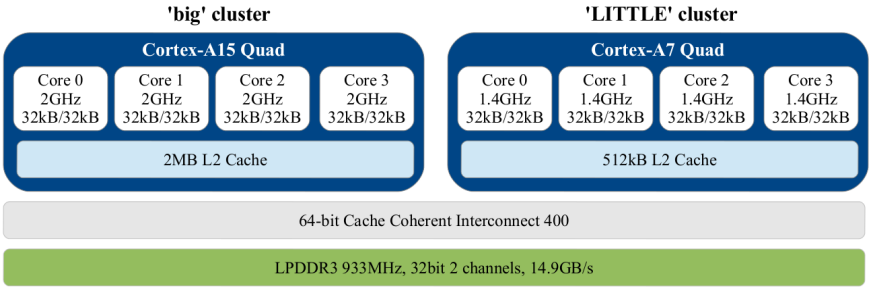


Figure 1: Sketch of the big.LITTLE technology integrated in the Exynos 5422 System-on-Chip.

The "big" cluster is composed of high-performance cores while the "LITTLE" cluster contains low power cores. The main idea behind this design is to select a suitable cluster according to the performance and power demand of executed workloads. While the traditional big.LITTLE configurations rely on application processors only, such as ARM Cortex-A7 or Cortex-A15, the CONTINUUM project aims to integrate also ultra-compact cores, such as Cortus cores which belong to the micro-controllers class. The target compute node architecture should be capable of supporting a full operating system, basically running on application processors. The presence of Cortus cores in the resulting heterogeneous architecture will increase its energy-efficiency. As there is not yet heterogeneous multicore architectures based on Cortus cores, we will consider some existing compute nodes integrating ARM big.LITTLE technology to carry out our preliminary investigations in the CONTINUUM project. The gained insights will serve in designing our solution with Cortus cores.

Outline. The rest of this deliverable is organized as follows. First, an overview of Cortus low power cores is given, together with a comparison against similar ARM low power cores in Section 2. Then, the energy-efficiency of a specific ARM big.LITTLE multicore system is evaluated based on the HPL and Rodinia benchmarks in Section 3. This case study gives some preliminary assessment of what one could expect from such a technology. A quick survey of some popular multi/manycore architectures is presented in Section 4. Finally, a few concluding remarks are provided in Section 5.

2 Very low power cores: Cortus versus ARM

We evaluate some selected CPU cores developed by the Cortus Company. The cores are compared with equivalent well-known ARM cores in order to provide the reader with a convenient comparison basis.

2.1 Overview

A possible design option investigated by the CONTINUUM project for energy-efficient compute nodes is a heterogeneous multicore architecture composed of many low power cores and a few high-performance cores. The candidate core technologies are those developed by the Cortus[1] partner. The current section briefly introduces the current processor families proposed by Cortus (Section 2.2). A comparison with similar ARM processors is also discussed (Section 2.3).

2.2 Cortus core technologies

The Cortus range of processors is all modern advanced 32-bit RISC processors, featuring the same core architecture and instruction set. However, they differ in silicon footprint and performance. Generally speaking, systems that are silicon and power-sensitive will find the APS processor family ideal. Applications demanding more performance and floating-point operations will find the high throughput FPS processors more suitable.

As illustrated in Table 1, within each family, one can distinguish processors that are designed so as to provide an *optimized core size* without compromising the performance. Other processors are designed in a way that increases their *code density*, therefore their instruction memory size. The increased code density comes at the expense of a slightly more complex processor core.

Table 1: Two families of Cortus processors (**typical area in 90 nm technology node**).

Family 1: optimized core size		**Family 2: optimized code density**	
APS1	0.039 mm2		
APS3R	0.043 mm2	APS23	0.049 mm2
APS3RP	0.079 mm2	APS23P	0.084 mm2
APS5	0.095 mm2	APS25	0.103 mm2
FPS6	0.185 mm2	FPS26	0.192 mm2

A typical architecture: APS25. The APS25 processor architecture depicted in Figure 2 is a fully 32-bit high-performance general purpose CPU, with an excellent code density (and a marginal increase in silicon area compared to APS5), designed specifically to meet the demands of embedded systems.

It relies on a Harvard architecture with 2 $\times$ 4 GB address space. The instructions are 16, 24 and 32 bits in length. Most of them are single cycle, including load and store. The 5-7 stage pipeline ensures ultra low power consumption and high-performance while retaining a reasonable maximum

[1] `http://www.cortus.com/overview.php`

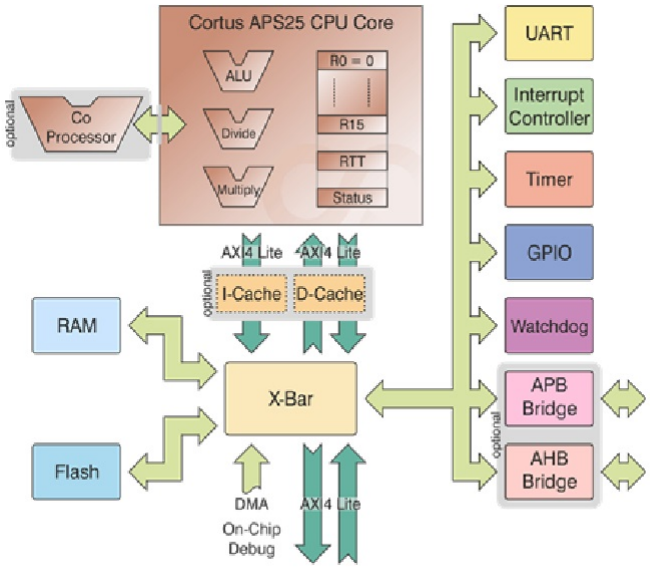


Figure 2: APS25 subsystem architecture

clock frequency. Out-of-order completion enables nearly all instructions to execute in a single cycle, including loads and stores. Interrupts are fully vectored and the architecture ensures a minimum of software overhead in task switches. The processor was designed to execute high level languages such as C. The entire GNU GCC toolsuite has been ported to this architecture.

Several standard peripherals are available. An optional trace buffer is also made available to make debugging rapid and simpler.

The Cortus APS bus is a simple and efficient synchronous bus that interfaces easily to synchronous memories (SRAM). It has a minimal number of signals that simplifies the interconnect reducing logic costs. With a sufficiently high-performance memory subsystem, it can offer zero latency memory accesses, with back-to-back reads and writes. The AXI 4 Lite bus is a high-performance bus that is compatible with other IP and interfaces easily with the APS bus. Efficient bridges between these bus interfaces, and to other popular standards such as AHB-Lite and APB are available.

Table 2 summarizes typical maximum operating frequencies of Cortus processor families. Here, the APS25 processor operates at the highest possible frequency.

Table 2: Maximum frequencies of APS and FPS processors in 90 nm.

Maximum frequency in UMC90			
APS1	307 MHz		
APS3R	312 MHz	APS23	235 MHz
APS3RP	285 MHz	APS23P	217 MHz
APS5	425 MHz	APS25	425 MHz
FPS6	400 MHz	FPS26	392 MHz

Table 3 provides a range of typical power consumption numbers for Cortus cores according to frequency scale.

Table 3: Power consumption per MHz in 90 nm.

Power consumption per frequency scale in UMC90	
Core type	Power per frequency scale (μW / MHz)
APS3R	10.42
APS3RP	11.42
FPS6	37.16
APS23	11.62
FPS5	17.56
FPS23P	12.66

The diversity of Cortus cores in terms of performance and power capabilities makes them attractive for usage in heterogeneous multicore systems, such as those explored in the CONTINUUM project. Typically, a system composed of a mix of such cores can run while its high-performance cores are in deep sleep mode in order to save power (e.g., running a network stack). Whenever an event requiring high-performance processing occurs, the high-performance subsystem is taken out of sleep mode in order to process the data already prepared by the low power companion cores. After all required data get processed, the high-performance cores can go back to sleep mode and the low power companion cores tidy up.

The ARM big.LITTLE technology [22] follows a similar principle by combining low power Cortex-A7 cores with high-performance Cortex-A15 cores so as to enable the adequate core selection depending on the nature of the executed workloads.

2.3 ARM core technologies

The Advanced RISC Machine (ARM) offers a family of Reduced Instruction Set Computing (RISC) architectures for computer processors, configured for various environments. A characteristic feature of ARM processors is their low electric power consumption. Almost all modern mobile phones and personal digital assistants contain ARM CPUs, making them one of the most widely used 32-bit microprocessor family in the world. The ARMv7-A cores, which rely on this 32-bit architecture will be evaluated in Section 3.

The Cortus processor technology introduced in the previous section is comparable to ARM microcontroller class processors, also known as Cortex-M class. Tables 4 and 5 compare both the size (in terms of gates count) and the maximum operating frequency for the two processor technology providers. The Cortus processors consume smaller area while providing higher frequency levels.

Table 4: Number of gates in core design: ARM Cortex-M versus Cortus APS.

Number of gates in ARM and Cortus cores			
Cortex M0	12 kgate	Cortus APS23	9 kgate
Cortex M3	33 kgate	Cortus APS5	17.4 kgate
Cortex M4	50 kgate	Cortus APS23P	15.3 kgate

Finally, Table 6 reports a comparison of both technologies in terms of Dhrystone Million Instruction per Second (DMIPS). The performance obtained with Cortus processor technology is generally higher than that of ARM Cortex-M class.

Table 5: Maximum core frequencies in 90 nm : ARM Cortex-M versus Cortus APS.

Maximum core frequencies			
Cortex M0	180 MHz	Cortus APS23	235 MHz
Cortex M3	180 MHz	Cortus APS5	425 MHz
Cortex M4	204 MHz	Cortus APS23P	217 MHz

Table 6: Performance comparison: APS/FPS versus Cortex-M cores.

DMIPS values			
APS3R	2.76 DMIPS/MHz	Cortex M0	1.27 DMIPS/MHz
APS3RP	2.76 DMIPS/MHz	Cortex M3	1.89 DMIPS/MHz
APS5	2.33 DMIPS/MHz	Cortex M4	1.91 DMIPS/MHz
FPS6	2.33 DMIPS/MHz		
APS23P	2.79 DMIPS/MHz		
APS25	2.52 DMIPS/MHz		

2.4 Summary

This section briefly introduced the Cortus processor families, which are envisioned as major building blocks in the heterogeneous multicore architecture explored by the CONTINUUM project for energy-efficient compute nodes. The massive usage of small embedded cores (such as those from Cortus) as promoted in the project proposal can provide an interesting compromise expected for energy-efficiency and cost-effectiveness. These small embedded cores are highly energy and silicon efficient offering more MIPS/mm2 or MIPS/μW of energy consumed than bigger application cores.

3 Evaluation of a big.LITTLE technology

Among the recent ARM-based technologies, big.LITTLE is certainly a very popular and promising solution that is worth-mentioning in our vision. Indeed, the CONTINUUM project aims at system designs with similar features as ARM big.LITTLE.

3.1 Introduction

We explore the potential of a state-of-the-art ARM-based computer board named Odroid XU3 for building energy-efficient many- and multicore systems. This board integrates the Samsung Exynos 5422 chip relying on ARM big.LITTLE technology [22] that enables to dynamically migrate applications between two different clusters of ARM cores: a low-power cluster composed of four Cortex-A7 cores *versus* a high-performance cluster composed of four Cortex-A15 cores. It also includes a GPU and further peripherals. The migration of applications between the 4-core clusters depends on their workload, i.e. it is steered by performance needs. Our study provides insightful performance and energy results in typical compute-intensive benchmarks.

3.2 Evaluated system setup information

3.2.1 Hardware characteristics

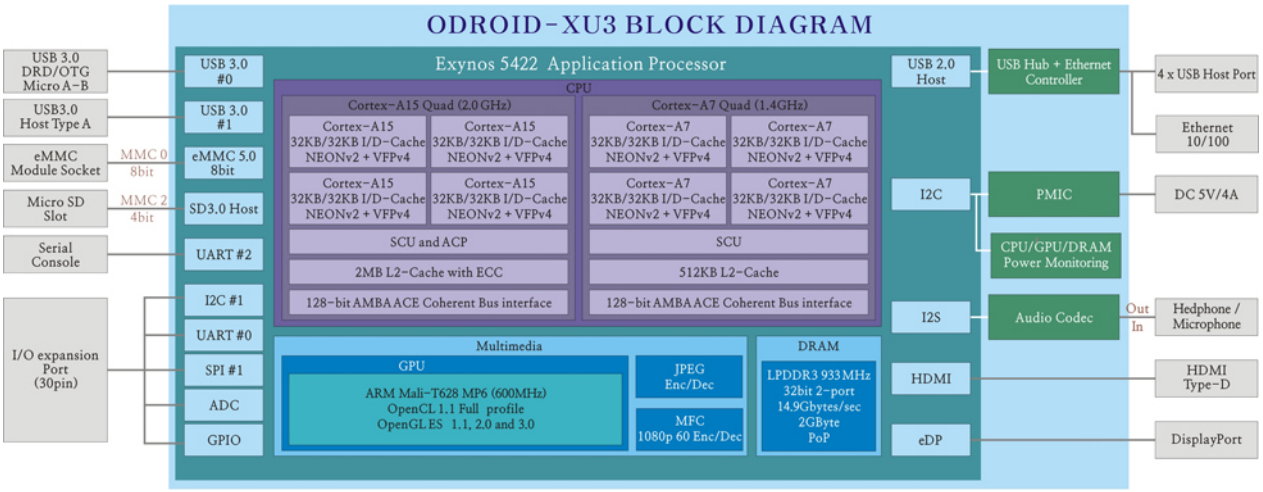


Figure 3: ODROID XU3 block diagram

To explore the capabilities of the Exynos 5422 chip, we consider the Odroid XU3 embedded board, developed by Hardkernel company[2]. This board embeds useful power and thermal sensors that facilitate the evaluation of the energy consumed by its relevant hardware components. Its block diagram is shown in Figure 3. The board is a complete embedded system featuring several components among which:

[2] `http://www.hardkernel.com`.

- Samsung Exynos 5422 chip with big.LITTLE processor, characterized by the following parameters:

Table 7: Exynos 5 Octa (5422) SoC specification.

Parameters		LITTLE	big
Architecture model			
Core type		Cortex-A7 (in-order)	Cortex-A15 (out-of-order)
Number of cores		4	4
Core clocks		200 MHz - 1.4 GHz	200 MHz - 2 GHz
L1	Size	32 kB	32 kB
	Assoc.	2-way	2-way
	Latency	3 cycles	4 cycles
L2	Size	512 kB	2 MB
	Assoc.	8-way	16-way
	Latency	15 cycles	21 cycles

- CCI-400 64-bit interconnect
- PowerVR SGX544MP3 GPU,
- 2 GB LPDDR3 RAM (933 MHz, 14.9 GB/s, 32-bit, 2 channels),
- eMMC 4.5 Flash Storage (64GB),
- Current and voltage sensors to measure power consumed by the two quad-core clusters, RAM memory and GPU.

3.2.2 Software parameters

To run the Odroid board we use the following software:

- Operating system: xUbuntu 13.10[3],
- Compilation: GCC 4.8.1,
- Libraries: OpenMPI 1.6.4 and automatically tuned linear algebra software (ATLAS) 3.10.1-2.

The first benchmark we consider is high-performance Linpack (HPL) [1]. It solves random dense linear systems in double-precision arithmetic (64 bits). The Top500 ranking of the most powerful supercomputers relies on this benchmark. The performance numbers obtained with HPL provide a good correction of theoretical peak performance. In our case, the Odroid board is evaluated according to the following setup: on each quad-core cluster, four MPI tasks (one task per core) will be executed;

[3] `http://www.odroid.in/Ubuntu_XU` – Linux 3.4.67 #5 SMP PREEMPT Sun Nov 24 19:25:46 KST 2013 armv7l armv7l armv7l GNU/Linux

the benchmark data fills around 1.2 Gbits of the LPDDR3 RAM memory; there is no SWAP and the GPU will not be used in all our experiments.

The second benchmark suite is Rodinia [13]. It is composed of applications and kernels of different nature in terms of workload, from domains such as bioinformatics, image processing, data mining, medical imaging and physics simulation. It also includes classical algorithms like LU decomposition and graph traversal. For our experiments, the following setup is considered: for each quad-core cluster, each application and kernel is executed through its OpenMP implementation configured with 4 threads (except for the *kmeans_serial* kernel which is executed with a single thread).

3.3 Energy-efficiency evaluation based on HPL

We evaluate the performance, power and energy-efficiency by considering the entire Odroid board at core peak performance level with HPL.

3.3.1 Performance and energy-efficiency of Odroid-XU3

The peak performance is evaluated in terms of Giga FLoating point Operations Per Second (GFLOPS) for both Cortex-A15 and Cortex-A7 quad-core clusters. The power consumption of the different components is monitored via on-board sensors. Results given in the following rely on an average of 10 iterations of HPL execution. Table 8 shows the HPL score for different cluster frequencies. As expected, at similar frequencies (i.e., 1.4Ghz, 0.8GHz and 0.2GHz) the Cortex-A15 cluster provides a higher performance than the Cortex-A7 cluster. The peak performance of Cortex-A15 cluster at 1.4GHz is around 4.96 GFLOPS, which is around 3 times higher than that of the Cortex-A7 cluster at the same frequency.

	A15				A7		
Freq. (GHz)	2.0	1.4	**0.8**	0.2	1.4	0.8	0.2
HPL score (GFLOPS)	4.7	4.96	**3.42**	0.96	1.68	1.04	0.26
Average Power (W)	12.5	7.5	**4.6**	2.76	3.46	2.58	2.18
EtoS (J)	221.7	127.7	**113.1**	240	172.1	206.4	710
Energy eff. (GFLOPS/W)	0.376	0.662	**0.746**	0.347	0.484	0.404	0.118

Table 8: HPL results for different frequencies of Odroid board.

Now, let us consider the energy-efficiency of the system in terms of GFLOPS per Watt (GFLOPS/W), which is computed from the HPL score, execution time and average power consumption. The corresponding results are given in the last row of Table 8. From the entire board level, the most energy efficient configuration corresponds to the Cortex-A15 cluster running at 800 MHz. Despite the fact that Cortex-A7 core consumes less power than Cortex-A15 core, the extremely compute-intensive feature of the HPL benchmark makes the Cortex-A15 quad-core cluster more energy-efficient than the Cortex-A7 one.

3.3.2 Comparison with other computer systems

Let us consider the most energy-efficient board-level configuration of the Odroid board identified previously, i.e. the Cortex-A15 cluster at 800 MHz. Now, we compare it with other systems running the HPL benchmark [24, 5, 34] in Table 9. In [34], the scalability and energy-efficiency of three multiprocessor-on-chip (MPSoCs) in a cluster are evaluated. These MPSoCs are PandaBoard, Snowball and Tegra. They all contain ARM Cortex-A9 processors. The obtained results show that Snowball is the most energy-efficient while Tegra 2 is the most scalable. In [5], a similar study is reported, which assesses the benefits of ARM core clusters compared to those relying on commodity processors such as x86. Compared to mentioned works using Cortex-A9, our identified configuration is more energy-efficient. This is explained by two reasons: first, the Cortex-A15 processor belongs to the third generation of Cortex-A family, which is more optimized than the second generation to which belongs the Cortex-A9 processor; second, the number of cores available in a system appears proportional to the energy-efficiency of that system.

In Table 9, the Viridis system contains four Cortex-A9 cores while both PandaBoard and Tegra 2 systems contain only two Cortex-A9 cores. The two AMD dual-core systems are less energy-efficient than all mentioned ARM-based systems. The only system that proves better than Odroid is composed of four i7 cores. However, the cost of such a node ($1000 each) is around 5 times higher than the Odroid XU3 board.

	i7 [24]	Atom64 [24]	amdf [24]	viridis[24]	Pandaboard [5]	Tegra 2 [34]	Odroid XU3
CPU	Intel Core i7-3615	Intel Atom N2600	AMD Fusion G-T40N	Cortex-A9	Cortex-A9	Cortex-A9	Cortex-A15
Num. of cores	4(8 threads)	2	4	2	2	2	4
HPL score (GFLOPS)	39.63	0.9575	1.609	3.218	1.601	0.9206	3.42
Energy eff. (MFLOPS/W)	1059	69	85	593	291	161	746

Table 9: Energy efficiency of single system node for HPL benchmark

3.4 Evaluation of the board using Rodinia

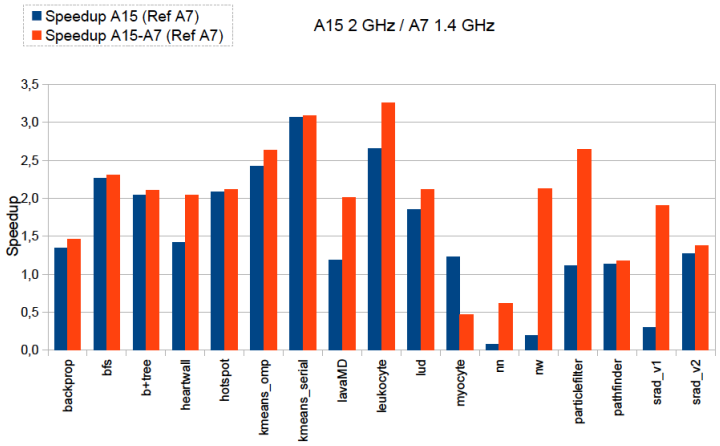
In this section, we evaluate the performance and energy-efficiency of the Odroid board when executing the Rodinia benchmark suite.

3.4.1 Evaluation scenarios

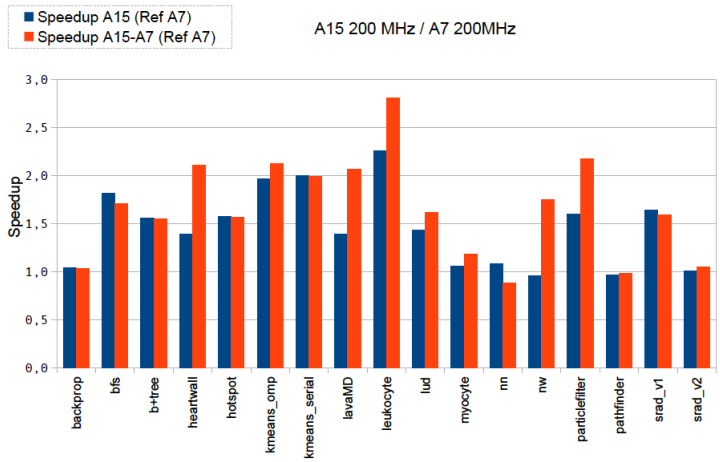
In the considered experiments, three execution configurations are considered: *i) execution only performed on the Cortex-A7 cluster, ii) execution only performed on the Cortex-A15 cluster, and iii) execution performed on both clusters, i.e., HMP mode*. Typically, for applications with low workloads, i.e. which are not performance-demanding, the Cortex-A7 cluster is generally preferable for low power execution. The Cortex-A15 cluster will be preferred for applications with high workloads. In the next paragraphs, for all experiments, the results are normalized regarding the configuration (i).

Figures 4 and 5 show the average speedup of configurations ii) and iii) *versus* configuration i). In the former case, all cores simultaneously operate either at their maximum or minimum frequency levels. In the latter case, cores simultaneously operate at different frequency levels.

While the two configurations always provide a better speedup than the reference configuration, the observed gains vary with the application kernels. The best speedup results are provided in the scenario captured by Figure 5. More generally, the HMP mode appears as the best, except for a few scenarios (e.g., Lud and Myocyte kernels in Figure 5).



(a) At maximum core operating frequencies

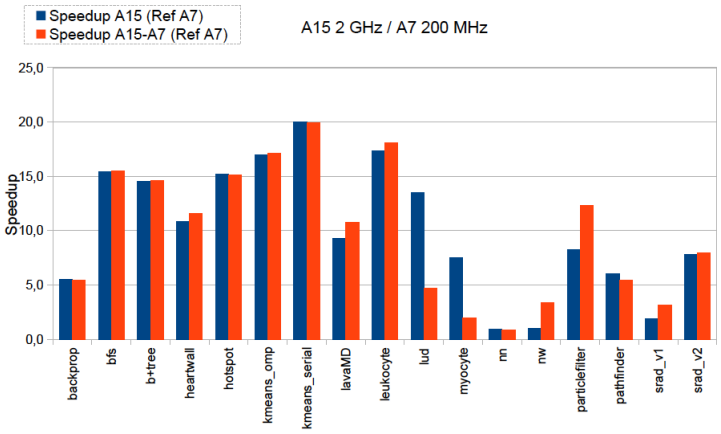


(b) At minimum core operating frequencies

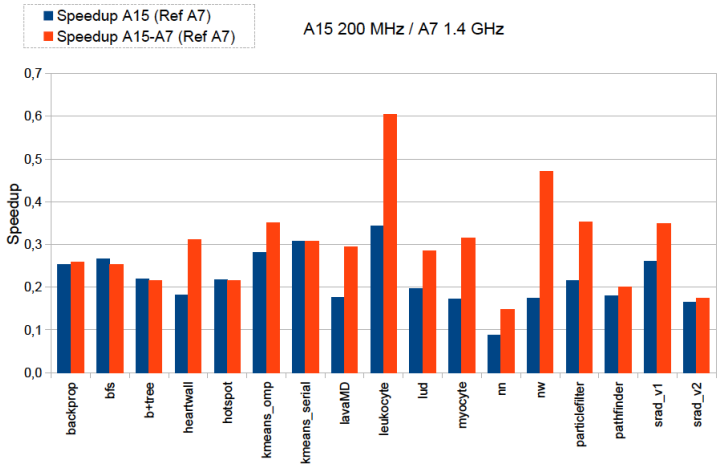
Figure 4: Speedup of A15 cluster and HMP execution modes vs. A7 cluster for Rodinia: all cores operating simultaneously at maximum/minimum frequency levels.

Figures 6, 7 and 8 details the energy consumption for each evaluated Rodinia kernel or application. The energy-to-solution measured when only using the Cortex-A7 cluster is globally less than that obtained with other configurations, i.e., when using only the Cortex-A15 cluster or the HMP mode. Contrarily to HPL, this evaluation shows that for a large part of Rodinia applications and kernels the Cortex-A7 mode appears more energy-efficient at board level.

This suggests that application nature has an impact on the energy consumption induced in the different clusters at board level: HPL permanently exploits the peak performance of available cores during execution while Rodinia applications and kernels, due to their irregular computational nature, imply a load fluctuation on cores.



(a) At maximum A15 and minimum A7 core operating frequencies



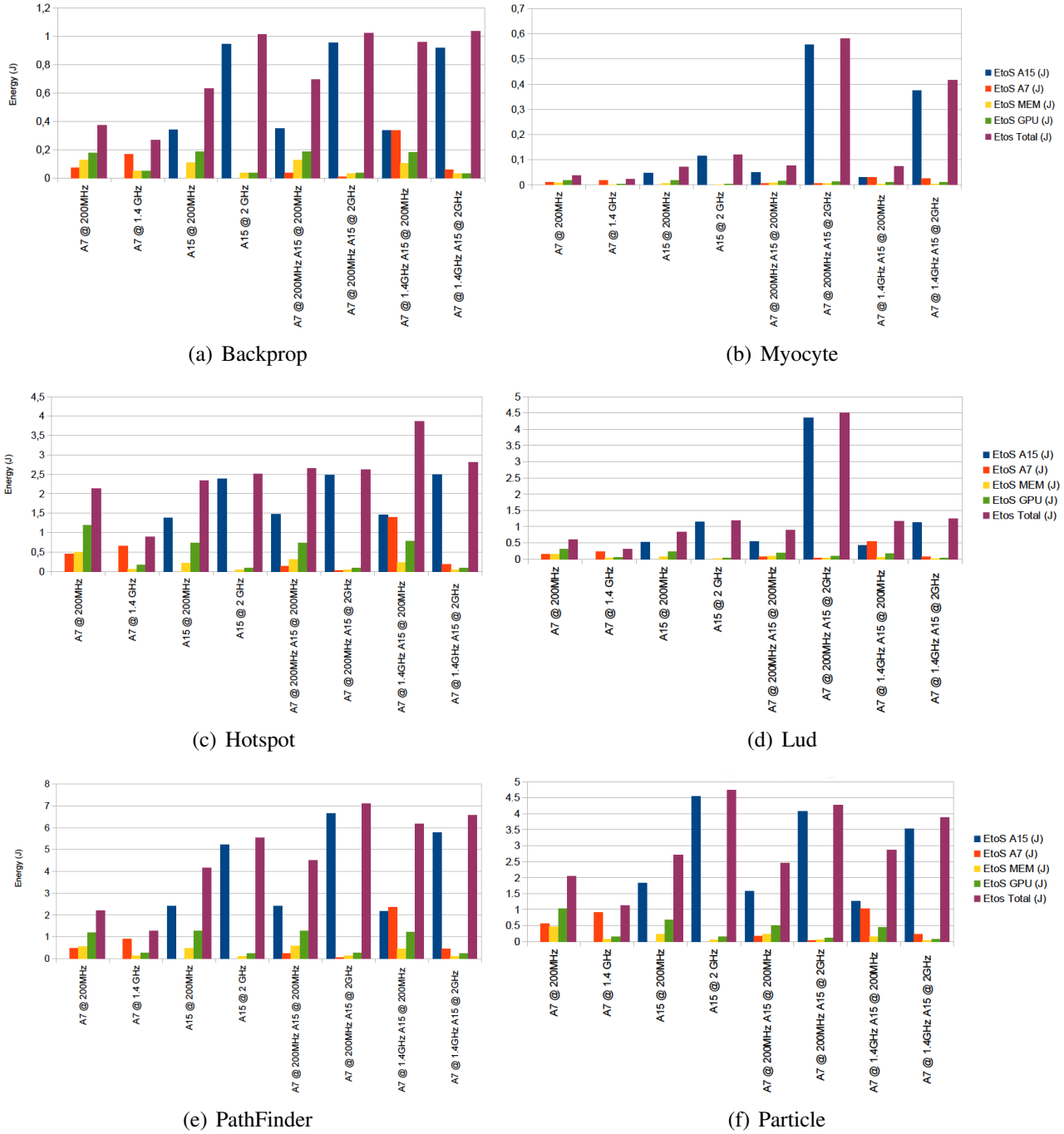
(b) At minimum A15 and maximum A7 core operating frequencies

Figure 5: Speedup of A15 cluster and HMP execution modes vs. A7 cluster for Rodinia: all cores operating simultaneously at different frequency levels.

3.5 Summary and remarks

This chapter presented an evaluation of opportunities and limitations of state-of-the-art and reasonable cost embedded multicore computer systems, integrating ARM big.LITTLE technology for energy-efficient mini-clusters. It provided insightful performance and energy-efficiency results based on two compute-intensive benchmarks, high-performance Linpack and Rodinia. The performance scalability of a mini-cluster composed of these boards has been analyzed.

These results showed that the big.LITTLE architecture of the considered Odroid board calls for adequate migration policies, which are capable of adequately addressing heterogeneous application workloads. A characterization of applications/tasks/threads is required, e.g., regular vs. irregular, computation-intensive vs. memory-intensive. This information can be used either offline or online together with data monitoring (power and energy consumption, CPU workload, etc.) to exploit as much as possible the energy-efficiency of clusters.

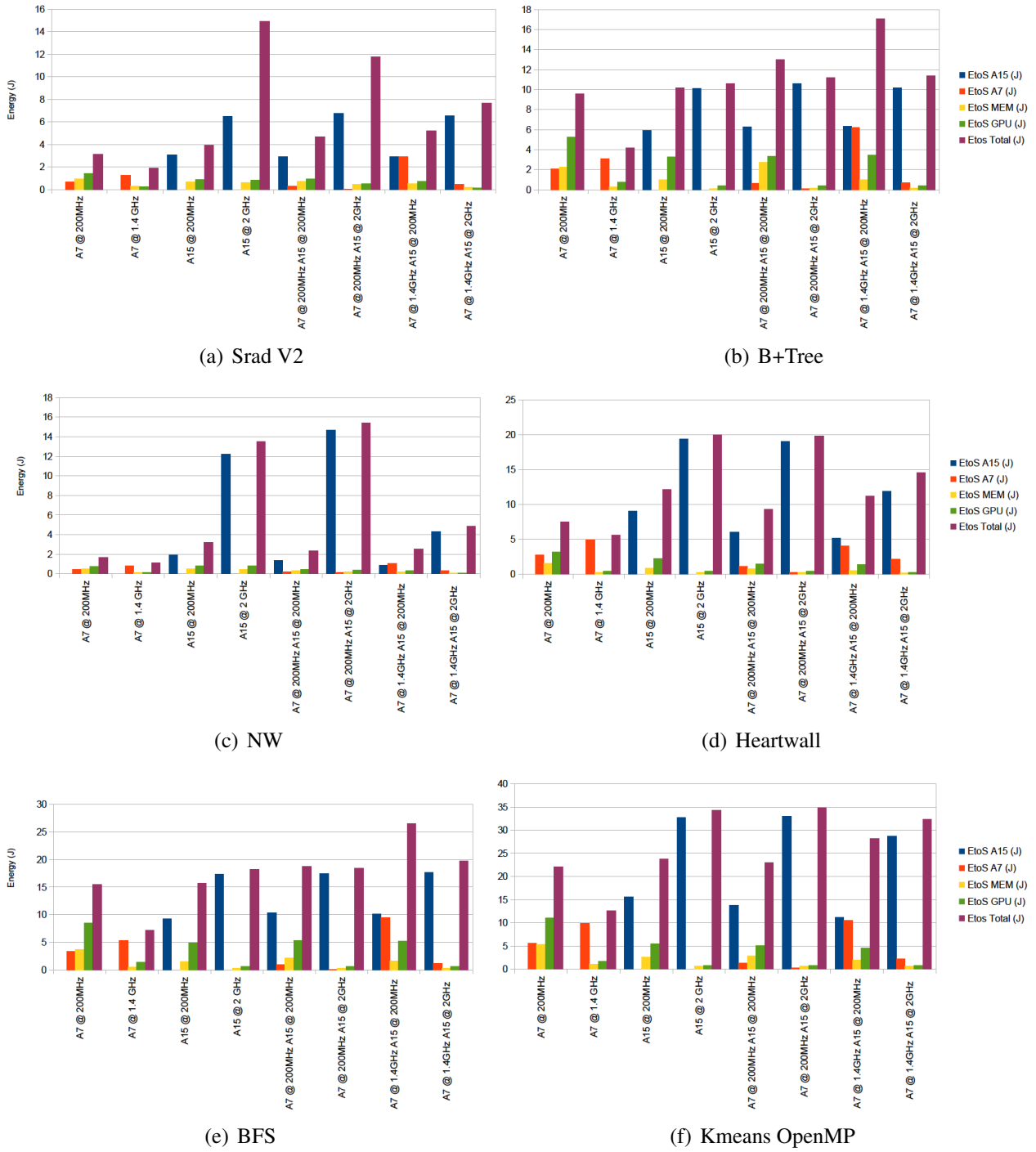


(a) Backprop (b) Myocyte

(c) Hotspot (d) Lud

(e) PathFinder (f) Particle

Figure 6: Energy-to-Solution of Rodinia kernels according different clustering modes and operating frequencies.

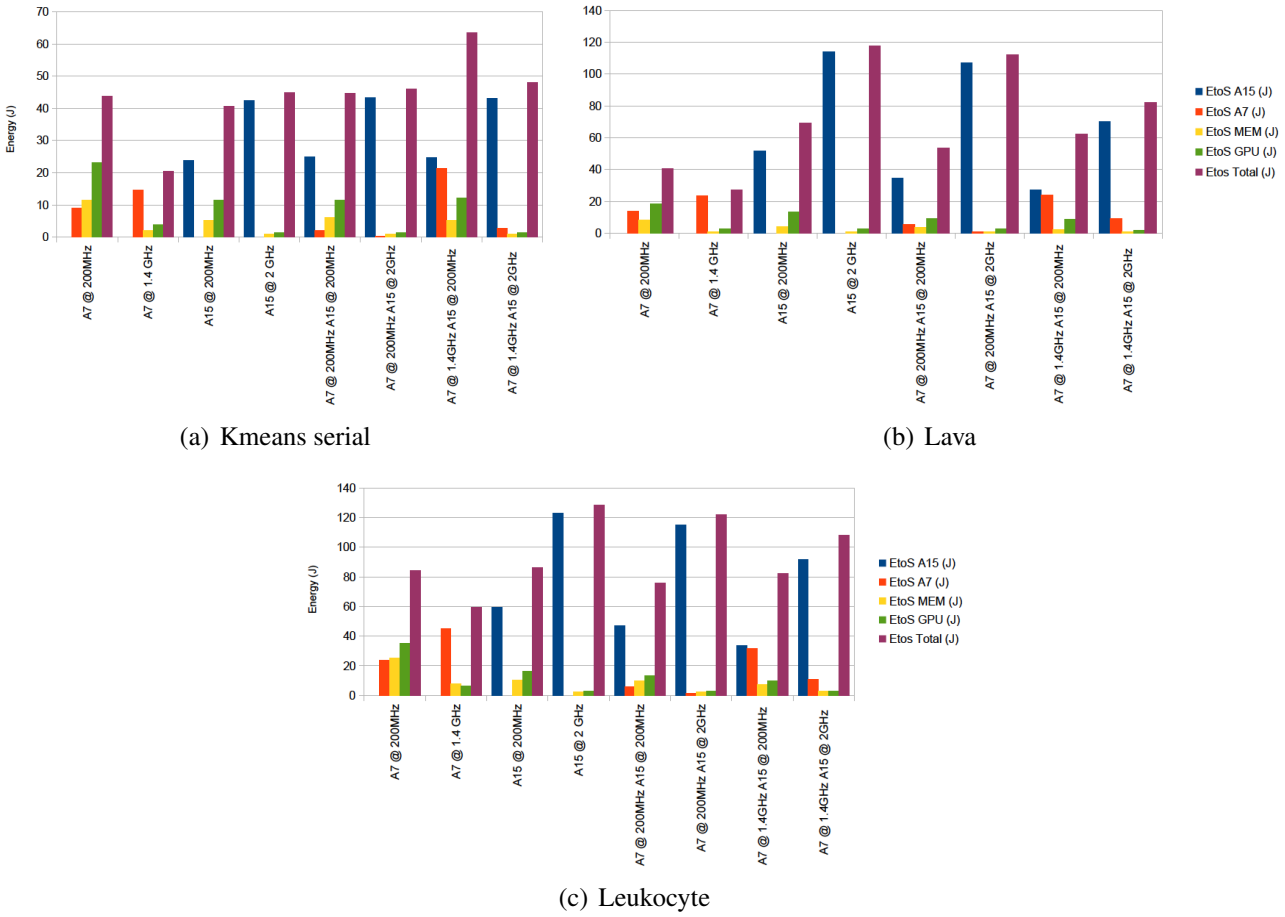


(a) Srad V2 (b) B+Tree

(c) NW (d) Heartwall

(e) BFS (f) Kmeans OpenMP

Figure 7: Energy-to-Solution of Rodinia kernels according different clustering modes and operating frequencies.



(a) Kmeans serial

(b) Lava

(c) Leukocyte

Figure 8: Energy-to-Solution of Rodinia kernels according different clustering modes and operating frequencies.

4 Further multicore architectures

Beyond the big.LITTLE technology evaluated in the previous section, there are further manycore architectures that can deserve attention. These architectures have different characteristics that could be considered in the design of the compute node architecture investigated in the CONTINUUM project. Thanks to their high number of cores, they represent interesting compute accelerators adopted in a number of execution infrastructures. In the next sections, we survey some of these architectures.

4.1 Graphical Processing Units of Nvidia

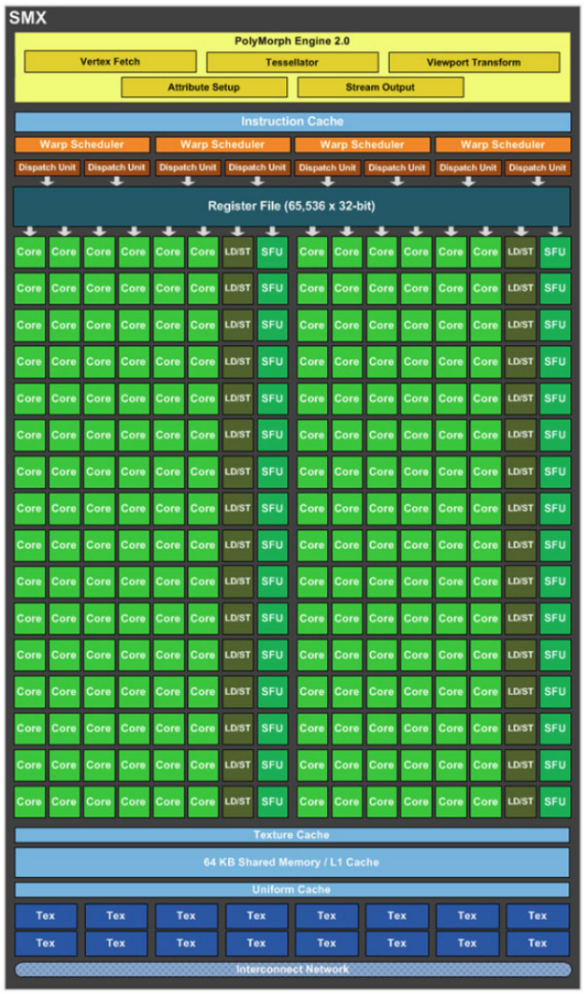


Figure 9: Kepler SMX architecture (source: `http://www.bit-tech.net`).

Graphical Processing Units (GPUs) such as Kepler[4] can have upto 15 streaming multiprocessors (SMX), each able to handle 192 single-precision cores and 64 double-precision units. Figure 9 illustrates a massively parallel architecture of an SMX. Each core has fully pipelined floating-point

[4] `http://www.nvidia.com/object/nvidia-kepler.html`

and integer arithmetic logic units. Threads are scheduled in SMX by groups of 32 parallel lightweight threads called warps. All these features make GPUs powerful compute accelerators able to provide high performance per watt, especially for regular data-parallel computations as found in graphics and scientific computing applications. Their main limitation comes when dealing with irregular applications, e.g. with conditional branches. In addition, when using an SMX for all its corresponding cores become active, thus energy-consuming. This means that to have an energy-efficient execution on the SMX, all its cores must be used.

4.2 Intel Many Integrated Core Architecture

Many Integrated Core Architecture (MIC)[5] [18] is the architecture adopted by Intel Xeon Phi co-processors, used as compute accelerators in the second ranked world's fastest[6] supercomputer (Thiane-2) in 2014. It is composed of 61 cores interconnected by a bi-directional ring network (see Figure 10). Intel Xeon Phi co-processors provide power gating of cores, L2 cache and memory controllers for leakage power reduction. Compared to GPUs for which performance optimization requires to run lightweight threads maximizing parallelism, the MIC architecture maximizes core performance through coarse-grained parallelism. However, a study indicated that the ring network and the ECC memory overhead are performance bottlenecks in MIC architecture, showing poor scalability beyond 32 cores [25].

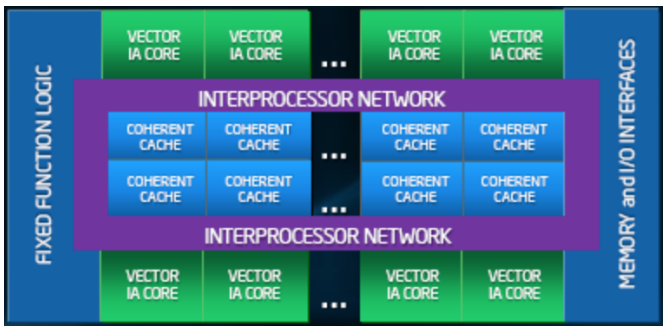


Figure 10: Intel MIC architecture (source: `http://wiki.expertiza.ncsu.edu`).

4.3 Tile-Gx of Tilera

TILE-Gx[7] architecture is another multicore processor, which comprises up to 72 cores interconnected by a 2D mesh NoC using wormhole routing packets. Figure 11 depicts a Tile-Gx architecture composed of 36 cores. From a global point of view, TILE-Gx processor in composed of a two-dimensional array

[5] `http://www.intel.com/content/www/us/en/architecture-and-technology/many-integrated-core/intel-many-integrated-core-architecture.html?_ga=1.31218297.751580311.1426595792`

[6] `https://www.top500.org/lists/2016/06`

[7] `http://www.mellanox.com/repository/solutions/tile-scm/docs/UG130-ArchOverview-TILE-Gx.pdf`

of identical so-called "tiles". Each tile consists of a core, 64KB L1 cache, 256 L2 cache and a non-blocking switch connecting tiles to the NoC. A shared address space with cache coherence maintained by hardware is considered. A study [32] showed that such a processor has a well-balanced architecture for achieving an excellent ratio of performance per watt. This makes TILE-Gx a good candidate for building energy-efficient compute accelerators. But, the weak point of its architectures mainly lies in the lack of floating point units (FPUs) required for compute-intensive applications.

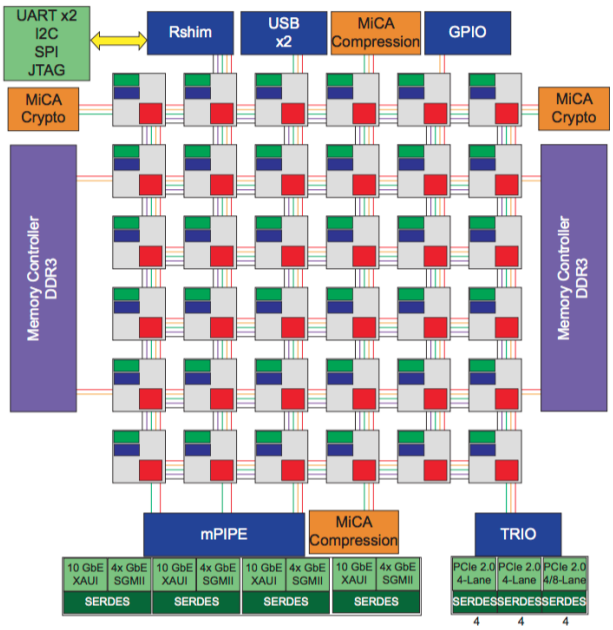


Figure 11: Tile-Gx36 block diagram (source `http://www.mellanox.com/repository/solutions/tile-scm/docs/UG130-ArchOverview-TILE-Gx.pdf`).

4.4 Multi-Purpose Processor Array of Kalray

Multi-Purpose Processor Array (MPPA)[8] [14] is a manycore architecture that integrates 256 cores, where cores are distributed across 16 compute clusters (see Figure 12). Each compute cluster has a private local memory and cache coherence is enforced by software. Communication and synchronization between compute clusters are ensured by a proprietary NoC using a 2D torus topology with a wormhole routing. A recent study [15] compares three accelerators: (i) MPPA, Intel i7-3820 quad-core, and Nvidia Tesla C2075 GPU. It showed that although MPPA has a lower peak performance than Intel i7-3820 and GPU for double precision floating-point arithmetic, for irregular application it outperforms Intel i7-3820 by a factor of 2.4 and only twice worse than the GPU. When comparing energy consumption, MPPA is over 20 and 6 times more efficient than Intel i7-3820 and GPU respectively. Indeed in a study [12] authors showed that a single thread execution on a cluster dissipates 3.73watts while 16 threads require 3.98watts. This suggests that a partial exploitation of the 16 cores available in a cluster leads to low energy-efficiency as with GPUs.

[8] `http://www.kalrayinc.com/kalray/products/#processors`

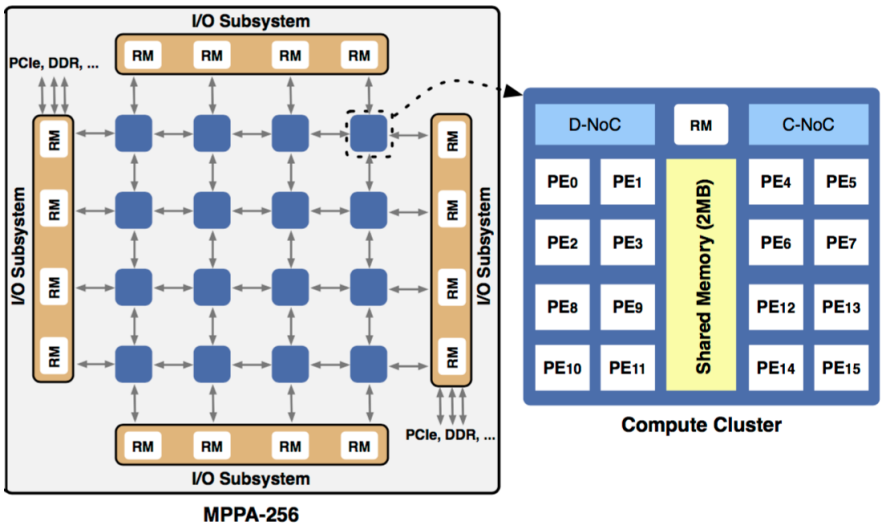


Figure 12: MPPA - 256 block diagram (source: [12]).

4.5 Tera-Scale ARchitecture

Tera-Scale ARchitecture (TSAR) [23, 20] is a scalable general-purpose cache-coherent globally asynchronous locally synchronous (GALS) multicore architecture. It consists of a set of clusters (see Figure 13) interconnected by a 2D mesh NoC. Each cluster is composed of 32 bits RISC cores without superscalar features. TSAR architecture aims to solve two major technical issues: i) scalability by targeting up to 4096 cores and ii) power consumption by using small cores to obtain the best MIPS/microwatt ratio. TSAR architecture physically implements MMU in the L1 cache controller. This creates an overhead in terms of silicon area and increased latency communication to maintain cache coherency. Therefore, network latency of distributed MMU may be very sensitive to the growth of the system.

A recent multicore architecture proposal [21] adopting distributed-memory design, shows promising performance, area and energy consumption improvements. This is enabled by the scalability of the architecture.

4.6 Summary

This section presented a number of existing compute accelerator architectures from which some interesting features could be borrowed for the compute node architecture explored in CONTINUUM. For instance, the cluster-based design of the TSAR architecture with low power cores certainly deserves to be considered as it shares a number of characteristics in the foreseen compute node architecture. However, an important innovation expected in CONTINUUM is the integration of emerging non-volatile memory technologies in the memory hierarchy, while exploiting core heterogeneity so as to achieve the best compromise in terms of performance and power consumption.

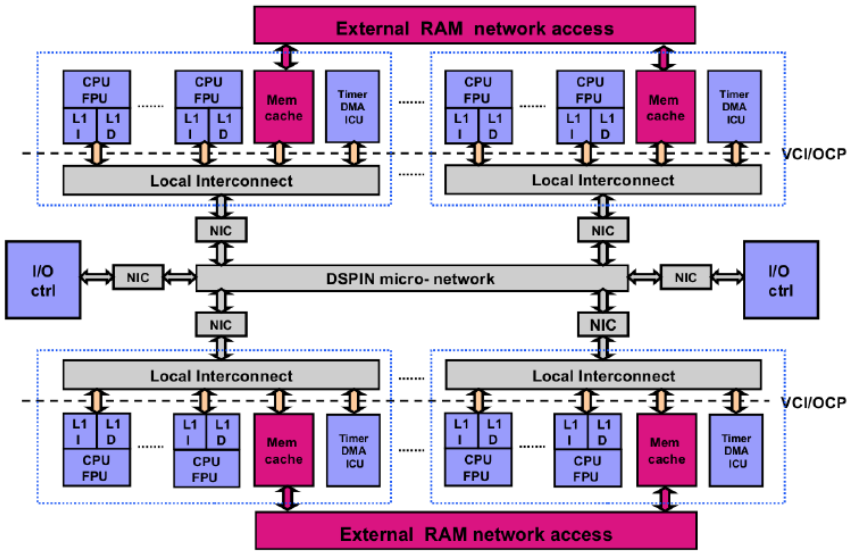


Figure 13: TSAR architecture (source [23]).

5 Conclusions and future work

In this deliverable, we presented a number of candidate core technologies for the compute node architecture studied in the CONTINUUM project. The core technologies from the Cortus partner are central building blocks in the definition of the expected architecture. Indeed, they appear as good candidates for offering the required capabilities in terms of performance and power consumption trade-off in order to conveniently reach energy-efficiency. The foreseen design solution shares several points with the ARM big.LITTLE technology, which consists of heterogeneous cores that can be selected according to the performance and power demand of executed workloads, for energy-efficiency purposes. The energy-efficiency of a system-on-chip integrating this technology has been evaluated in this deliverable. On the other hand, some complementary manycore architectures have been surveyed, as a possible inspiration basis for the compute node targeted in our project.

From this preliminary study on relevant candidate technologies about cores, we can properly now address the design of our target compute node based on all interesting features identified from the state-of-the-art. In particular:

- as big.LITTLE design paradigm shows several benefits in terms of performance and power consumption trade-off, we would like to focus on its heterogeneous feature by considering the ultra-compact core technology from Cortus, which are more energy-efficient than traditional big.LITTLE configurations;

- further interesting design paradigms identified from the reviewed literature, e.g., mesh interconnects and cluster-based partitioning [23, 21], deserve high attention as part of the design options to be considered in the next steps of the project;

- emerging non-volatile memories [43], such as magnetic memories integrated in last-level caches [41, 40, 36, 42, 17], are other design ingredients to take into account for an aggressive energy reduction.

References

[1] A. Petitet and R. C. Whaley and J. Dongarra and A. Cleary. HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. http://www.netlib.org/benchmark/hpl, 2016.

[2] An, X., Boumedien, S., Gamatié, A., and Rutten, É. CLASSY: a clock analysis system for rapid prototyping of embedded applications on mpsocs. In Corporaal, H. and Stuijk, S., editors, *Workshop on Software and Compilers for Embedded Systems, Map2MPSoC/SCOPES 2012, Sankt Goar, Germany, May 15-16, 2012*, pages 3–12. ACM, 2012. doi: 10.1145/2236576.2236577. URL https://doi.org/10.1145/2236576.2236577.

[3] An, X., Gamatié, A., and Rutten, É. High-level design space exploration for adaptive applications on multiprocessor systems-on-chip. *J. Syst. Archit.*, 61(3-4):172–184, 2015. doi: 10.1016/j.sysarc.2015.02.002. URL https://doi.org/10.1016/j.sysarc.2015.02.002.

[4] Apvrille, L. and Bécoulet, A. Prototyping an Embedded Automotive System from its UML/SysML Models. In *Embedded Real Time Software and Systems (ERTS2012)*, Toulouse, France, February 2012. URL https://hal.archives-ouvertes.fr/hal-02191862.

[5] Balakrishnan, N. Building and benchmarking a low power arm cluster. Master's thesis, Univ. of Edinburgh, Scottland, 2012.

[6] Bergman, K., Borkar, S., Campbell, D., Carlson, W., Dally, W., Denneau, M., Franzon, P., Harrod, W., Hiller, J., Karp, S., Keckler, S., Klein, D., Lucas, R., Richards, M., Scarpelli, A., Scott, S., Snavely, A., Sterling, T., Williams, R. S., Yelick, K., Bergman, K., Borkar, S., Campbell, D., Carlson, W., Dally, W., Denneau, M., Franzon, P., Harrod, W., Hiller, J., Keckler, S., Klein, D., Kogge, P., Williams, R. S., and Yelick, K. Exascale computing study: Technology challenges in achieving exascale systems peter kogge, editor & study lead, 2008.

[7] Binkert, N., Beckmann, B., Black, G., Reinhardt, S. K., Saidi, A., Basu, A., Hestness, J., Hower, D. R., Krishna, T., Sardashti, S., Sen, R., Sewell, K., Shoaib, M., Vaish, N., Hill, M. D., and Wood, D. A. The gem5 simulator. *SIGARCH Comput. Archit. News*, 39(2):1–7, August 2011. ISSN 0163-5964. doi: 10.1145/2024716.2024718. URL https://doi.org/10.1145/2024716.2024718.

[8] Butko, A., Gamatié, A., Sassatelli, G., Torres, L., and Robert, M. Design exploration for next generation high-performance manycore on-chip systems: Application to big.little architectures. In *2015 IEEE Computer Society Annual Symposium on VLSI*, pages 551–556, 2015. doi: 10.1109/ISVLSI.2015.28.

[9] Butko, A., Garibotti, R., Ost, L., Lapotre, V., Gamatié, A., Sassatelli, G., and Adeniyi-Jones, C. A trace-driven approach for fast and accurate simulation of manycore architectures. In *The 20th Asia and South Pacific Design Automation Conference, ASP-DAC 2015, Chiba, Japan, January 19-22, 2015*, pages 707–712. IEEE, 2015. doi: 10.1109/ASPDAC.2015.7059093. URL https://doi.org/10.1109/ASPDAC.2015.7059093.

[10] Butko, A., Bruguier, F., Gamatié, A., Sassatelli, G., Novo, D., Torres, L., and Robert, M. Full-system simulation of big.little multicore architecture for performance and energy exploration. In *10th IEEE International Symposium on Embedded Multicore/Many-core Systems-on-Chip, MCSOC 2016, Lyon, France, September 21-23, 2016*, pages 201–208. IEEE Computer Society, 2016. doi: 10.1109/MCSoC.2016.20. URL https://doi.org/10.1109/MCSoC.2016.20.

[11] Caliri, G. V. Introduction to analytical modeling. In *26th International Computer Measurement Group Conference, December 10-15, 2000, Orlando, FL, USA, Proceedings*, pages 31–36. Computer Measurement Group, 2000. URL http://www.cmg.org/?s2member_file_download=/proceedings/2000/0004.pdf.

[12] Castro, M., Francesquini, E., Nguélé, T. M., and Méhaut, J.-F. Analysis of computing and energy performance of multicore, numa, and manycore platforms for an irregular application. In *Proceedings of the 3rd Workshop on Irregular Applications: Architectures and Algorithms*, IA3 '13, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450325035. doi: 10.1145/2535753.2535757. URL https://doi.org/10.1145/2535753.2535757.

[13] Che, S., Sheaffer, J. W., Boyer, M., Szafaryn, L. G., Liang Wang, and Skadron, K. A characterization of the rodinia benchmark suite with comparison to contemporary cmp workloads. In *IEEE International Symposium on Workload Characterization (IISWC'10)*, pages 1–11, 2010. doi: 10.1109/IISWC.2010.5650274.

[14] de Dinechin, B. D. Kalray mppa®: Massively parallel processor array: Revisiting dsp acceleration with the kalray mppa manycore processor. In *2015 IEEE Hot Chips 27 Symposium (HCS)*, pages 1–27, 2015. doi: 10.1109/HOTCHIPS.2015.7477332.

[15] de Dinechin, B. D., Ayrignac, R., Beaucamps, P., Couvert, P., Ganne, B., de Massas, P. G., Jacquet, F., Jones, S., Chaisemartin, N. M., Riss, F., and Strudel, T. A clustered manycore processor architecture for embedded and accelerated applications. In *2013 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6, 2013. doi: 10.1109/HPEC.2013.6670342.

[16] Dekeyser, J.-L., Gamatié, A., Etien, A., Ben Atitallah, R., and Boulet, P. Using the UML Profile for MARTE to MPSoC Co-Design. In *First International Conference on Embedded Systems & Critical Applications (ICESCA'08)*, Tunis, Tunisia, May 2008. URL https://hal.inria.fr/inria-00524363.

[17] Delobelle, T., Péneau, P.-Y., Senni, S., Bruguier, F., Gamatié, A., Sassatelli, G., and Torres, L. Flot automatique d'évaluation pour l'exploration d'architectures à base de mémoires non volatiles. In *Conférence d'informatique en Parallélisme, Architecture et Système, Compas'16, Lorient, France*, 2016.

[18] Duran, A. and Klemm, M. The intel® many integrated core architecture. In *2012 International Conference on High Performance Computing Simulation (HPCS)*, pages 365–366, 2012. doi: 10.1109/HPCSim.2012.6266938.

[19] Gamatié, A., Beux, S. L., Piel, É., Atitallah, R. B., Etien, A., Marquet, P., and Dekeyser, J. A model-driven design framework for massively parallel embedded systems. *ACM Trans. Embed. Comput. Syst.*, 10(4):39:1–39:36, 2011. doi: 10.1145/2043662.2043663. URL https://doi.org/10.1145/2043662.2043663.

[20] Gao, Y. *Contrôleur de cache générique pour une architecture manycore massivement parallèle à mémoire partagée cohérente*. PhD thesis, 2011. URL http://www.theses.fr/2011PA066296. Thèse de doctorat dirigée par Greiner, Alain Informatique Paris 6 2011.

[21] Garibotti, R., Butko, A., Ost, L., Gamatié, A., Sassatelli, G., and Adeniyi-Jones, C. Efficient embedded software migration towards clusterized distributed-memory architectures. *IEEE Trans. Computers*, 65(8):2645–2651, 2016. doi: 10.1109/TC.2015.2485202. URL https://doi.org/10.1109/TC.2015.2485202.

[22] Greenhalgh, P. big.LITTLE processing with ARM Cortex-a15 & Cortex-a7. ARM White Paper, 2011.

[23] Greiner, A. Tsar: a scalable, shared memory, many-cores architecture with global cache coherence. In *9th International Forum on Embedded MPSoC and Multicore (MPSoC–09)*, 2009.

[24] Jarus, M., Varrette, S., Oleksiak, A., and Bouvry, P. Performance evaluation and energy efficiency of high-density hpc platforms based on intel, amd and arm processors. In *Revised Selected Papers of the COST IC0804 European Conference on Energy Efficiency in Large Scale Distributed Systems - Volume 8046*, EE-LSDS 2013, page 182–200, Berlin, Heidelberg, 2013. Springer-Verlag. ISBN 9783642405167. doi: 10.1007/978-3-642-40517-4_16. URL https://doi.org/10.1007/978-3-642-40517-4_16.

[25] Jeong, K., Kahng, A. B., Kang, S., Rosing, T. S., and Strong, R. Mapg: Memory access power gating. In *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1054–1059, 2012. doi: 10.1109/DATE.2012.6176651.

[26] Kumar, R., Tullsen, D. M., Jouppi, N. P., and Ranganathan, P. Heterogeneous chip multiprocessors. *Computer*, 38(11):32–38, 2005. doi: 10.1109/MC.2005.379.

[27] Latif, K., Effiong, C. E., Gamatié, A., Sassatelli, G., Zordan, L. B., Ost, L., Dziurzanski, P., and Soares Indrusiak, L. An Integrated Framework for Model-Based Design and Analysis of Automotive Multi-Core Systems. In *FDL: Forum on specification & Design Languages*, Work-in-Progress Session, Barcelona, Spain, September 2015. URL https://hal-lirmm.ccsd.cnrs.fr/lirmm-01418748.

[28] Latif, K., Selva, M., Effiong, C., Ursu, R., Gamatié, A., Sassatelli, G., Zordan, L., Ost, L., Dziurzanski, P., and Indrusiak, L. S. Design space exploration for complex automotive applications: An engine control system case study. In *Proceedings of the 2016 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*, RAPIDO '16, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340724. doi: 10.1145/2852339.2852341. URL https://doi.org/10.1145/2852339.2852341.

[29] Martin, G., Bailey, B., and Piziali, A. *ESL Design and Verification: A Prescription for Electronic System Level Methodology*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007. ISBN 9780080488837.

[30] Marwedel, P. *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*. Springer Publishing Company, Incorporated, 2nd edition, 2010. ISBN 9789400702561.

[31] Mello, A., Maia, I., Greiner, A., and Pêcheux, F. Parallel simulation of systemc TLM 2.0 compliant mpsoc on SMP workstations. In Micheli, G. D., Al-Hashimi, B. M., Müller, W., and Macii, E., editors, *Design, Automation and Test in Europe, DATE 2010, Dresden, Germany, March 8-12, 2010*, pages 606–609. IEEE Computer Society, 2010. doi: 10.1109/DATE.2010.5457136. URL https://doi.org/10.1109/DATE.2010.5457136.

[32] Munir, A., Gordon-Ross, A., and Ranka, S. Parallelized benchmark-driven performance evaluation of smps and tiled multi-core architectures for embedded systems. In *2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC)*, pages 416–423, 2012. doi: 10.1109/PCCC.2012.6407785.

[33] Ou, Z., Pang, B., Deng, Y., Nurminen, J. K., Ylä-Jääski, A., and Hui, P. Energy- and cost-efficiency analysis of arm-based clusters. In *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pages 115–123, 2012. doi: 10.1109/CCGrid.2012.84.

[34] Padoin, E. L., de Oliveira, D. A. G., Velho, P., Navaux, P. O. A., Videau, B., Degomme, A., and Mehaut, J.-F. Scalability and energy efficiency of hpc cluster with arm mpsoc. In *Workshop of Parallel and Distributed Processing, orto Alegre, RS*, 2013.

[35] Padoin, E. L., Pilla, L. L., Castro, M., Boito, F. Z., Olivier, P., Navaux, A., and Méhaut, J.-F. *IET Computers Digital Techniques*, 9:27–35(8), January 2015. ISSN 1751-8601. URL https://digital-library.theiet.org/content/journals/10.1049/iet-cdt.2014.0074.

[36] Péneau, P., Bouziane, R., Gamatié, A., Rohou, E., Bruguier, F., Sassatelli, G., Torres, L., and Senni, S. Loop optimization in presence of STT-MRAM caches: A study of performance-energy tradeoffs. In *26th International Workshop on Power and Timing Modeling, Optimization and Simulation, PATMOS 2016, Bremen, Germany, September 21-23, 2016*, pages 162–169. IEEE, 2016. doi: 10.1109/PATMOS.2016.7833682. URL https://doi.org/10.1109/PATMOS.2016.7833682.

[37] Quadri, I. R., Gamatié, A., Boulet, P., and Dekeyser, J.-L. Modeling of Configurations for Embedded System Implementations in MARTE. In *1st workshop on Model Based Engineering for Embedded Systems Design - Design, Automation and Test in Europe (DATE 2010)*, Dresden, Germany, March 2010. URL https://hal.inria.fr/inria-00486845.

[38] Rajovic, N., Rico, A., Puzovic, N., Adeniyi-Jones, C., and Ramirez, A. Tibidabo: Making the case for an arm-based hpc system. *Future Generation Computer Systems*, 36:322 – 334, 2014. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2013.07.013. URL http://

www.sciencedirect.com/science/article/pii/S0167739X13001581. Special Section: Intelligent Big Data Processing Special Section: Behavior Data Security Issues in Network Information Propagation Special Section: Energy-efficiency in Large Distributed Computing Architectures Special Section: eScience Infrastructure and Applications.

[39] Schirner, G. and Dömer, R. Quantitative analysis of the speed/accuracy trade-off in transaction level modeling. *ACM Trans. Embed. Comput. Syst.*, 8(1), January 2009. ISSN 1539-9087. doi: 10.1145/1457246.1457250. URL https://doi.org/10.1145/1457246.1457250.

[40] Senni, S., Brum, R. M., Torres, L., Sassatelli, G., Gamatié, A., and Mussard, B. Potential applications based on NVM emerging technologies. In Nebel, W. and Atienza, D., editors, *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, Grenoble, France, March 9-13, 2015*, pages 1012–1017. ACM, 2015. URL http://dl.acm.org/citation.cfm?id=2757049.

[41] Senni, S., Torres, L., Sassatelli, G., Gamatié, A., and Mussard, B. Emerging non-volatile memory technologies exploration flow for processor architecture. In *2015 IEEE Computer Society Annual Symposium on VLSI, ISVLSI 2015, Montpellier, France, July 8-10, 2015*, page 460. IEEE Computer Society, 2015. doi: 10.1109/ISVLSI.2015.126. URL https://doi.org/10.1109/ISVLSI.2015.126.

[42] Senni, S., Torres, L., Sassatelli, G., Gamatié, A., and Mussard, B. Exploring MRAM technologies for energy efficient systems-on-chip. *IEEE J. Emerg. Sel. Topics Circuits Syst.*, 6(3):279–292, 2016. doi: 10.1109/JETCAS.2016.2547680. URL https://doi.org/10.1109/JETCAS.2016.2547680.

[43] The CONTINUUM Project Consortium. Survey on emerging memory and communication technologies. Technical Report Deliverable D3.1, June 2016.

[44] Van Craeynest, K. and Eeckhout, L. Understanding fundamental design choices in single-isa heterogeneous multicore architectures. *ACM Trans. Archit. Code Optim.*, 9(4), January 2013. ISSN 1544-3566. doi: 10.1145/2400682.2400691. URL https://doi.org/10.1145/2400682.2400691.

[45] Wolf, W. *FPGA-Based System Design*. Prentice Hall PTR, USA, 2004. ISBN 0131424610.