Full-System Simulation of big.LITTLE Multicore Architecture for Performance and Energy Exploration

Anastasiia Butko, Florent Bruguier, Abdoulaye Gamatié, Gilles Sassatelli, David Novo, Lionel Torres, Michel Robert

Full-System Simulation of big.LITTLE Multicore Architecture for Performance and Energy Exploration

Anastasiia Butko, Florent Bruguier, Abdoulaye Gamatié,
Gilles Sassatelli, David Novo, Lionel Torres and Michel Robert
LIRMM (CNRS and University of Montpellier)
Montpellier, France
Email: {firstname.lastname}@lirmm.fr

***Abstract*—Single-ISA heterogeneous multicore processors have gained increasing popularity with the introduction of recent technologies such as ARM big.LITTLE. These processors offer increased energy efficiency through combining low power in-order cores with high performance out-of-order cores. Efficiently exploiting this attractive feature requires careful management so as to meet the demands of targeted applications. In this paper, we explore the design of those architectures based on the ARM big.LITTLE technology by modeling performance and power in gem5 and McPAT frameworks. Our models are validated w.r.t. the Samsung Exynos 5 Octa (5422) chip. We show average errors of 20% in execution time, 13% for power consumption and 24% for energy-to-solution.**

***Keywords*—Full-system simulation, single-ISA heterogeneous, multicore, gem5, McPAT, performance, energy, accuracy, ARM big.LITTLE.**

I. Introduction

To meet rapidly growing demands, future computing systems will need to be increasingly scalable and energy-efficient. To build architectures providing the required compromise in terms of performance and power dissipation, heterogeneous systems have become a promising direction. Such architectures usually consist of various processors/cores that differ from each other from their instruction set architectures (ISAs), their execution paradigms, e.g. in-order and out-of-order, their cache size and other fundamental characteristics. Particularly, single-ISA heterogeneous multicore processors [1] are made of multiple sets of cores that at the same time share a common ISA. Thereby they can run a unique standard operating system taking advantage of load-balancing features for fine control over performance and power consumption.

There are three software execution modes, which aim to explore the provided heterogeneity: (i) cluster migration, (ii) core migration and (iii) heterogeneous multiprocessing (HMP) [2]. Among other modes that imply only partial use of available resources, HMP mode allows using all of the cores simultaneously and enables fine-grained control for task scheduling.

In the mobile market, several system-on-chips (SoCs) operating on that principle exist. Nvidia Tegra 3/4 SoC [3] represents Variable Symmetric Multiprocessing (vSMP) technology that combines four faster power-hungry cores together with one 'companion' core dedicated to background tasks. All five cores have similar architecture, but the main cores are built in a standard silicon process to reach higher frequencies and the 'companion' core is built using a special low power silicon process that executes tasks at a low frequency [4].

ARM big.LITTLE technology integrated into Samsung Exynos 5/7 Octa SoC [5] combines two different types of cores. Developers reported over 50% in energy savings for popular activities such as web browsing and music playback with the duo Cortex-A7/Cortex-A15 configuration [6].

The design choice of architecture parameters such as the core types, the symmetric/asymmetric configurations, the cache size, is crucial for system energy efficiency. In [7] authors aim at providing some fundamental design insights based on a high-level analytical model analysis. Particularly, they claim two cores type being the most beneficial configuration and the task-to-core scheduling policy importance. Unlike analytical model-based estimation techniques [8] [9], full-system (FS) simulators provide a broad range of architecture configurations for detailed design exploration. They enable realistic software execution including operating system, runtime scheduling and parallel workloads.

Our contributions. In this work, we evaluate performance and power models of ARM big.LITTLE architecture for performance and energy trade-offs exploration. Models are implemented in gem5 [10] and McPAT [11] simulation frameworks. The accuracy in both performance and power estimations is assessed by comparing with a reference Exynos 5 Octa (5422) SoC integrated in the Odroid-XU3 computer board. This study is conducted using the Rodinia benchmark suite through its OpenMP implementation [12].

The main contributions of the present paper can be summarized as follows:

- Cycle-approximate performance and power models of ARM big.LITTLE heterogeneous processor are defined and implemented. These models are validated w.r.t. the real Exynos 5 Octa (5422) system-on-chip and show average errors around 20% for performance, 13% for

power consumption and 24% for EtoS. They are aimed to be freely available online [1].

- Based on the detailed analysis of above models, we report some useful insights about major simulation error sources and their associated impact on performance assessment. Despite the observed average error, we argue that our modeling is largely able to undertake the architecture exploration.

The rest of the paper is organized as follows: Section II presents related work on heterogeneous multicore architecture modeling and evaluation. In Section III the implementation of architecture and power models is described. The accuracy assessment of these models is discussed in Section IV. Section V brings our insights regarding the error sources of our modeling approach. Finally, Section VI gives concluding remarks and perspectives.

II. Related Work

A large part of studies on single-ISA heterogeneous multicores focuses on design space exploration, efficient task scheduling and performance/power evaluation.

In [7] authors use an analytical modeling for a large design space exploration of single-ISA heterogeneous architectures. Sarma et al. in [13] present a cross-layer exploration of heterogeneous multicore processor configurations. They demonstrate some predictive models for task allocation, performance-power models for different core types and workloads.

In [14] the authors propose a hierarchical power management framework for asymmetric multicores, in particular for ARM big.LITTLE architecture, in order to minimize energy consumption within the thermal design power constraint. Yu et al. in [15] evaluate ARM big.LITTLE power-aware task scheduling, via power saving techniques such as dynamic voltage and frequency scaling (DVFS) and dynamic hot plug. Tan et al. in [16] implement a computation approximation-aware scheduling framework in order to minimize energy consumption and maximize quality of service, while preserving performance and thermal design power constraints. They validate their framework on the Versatile Express Development Platform that includes a prototype version of the ARM big.LITTLE chip containing 3 Cortex-A7 cores and 2 Cortex-A15 cores. In [9], authors propose a performance/power model based on profiling information collected from the considered hardware platform. Hardware-based estimation often proves challenging compared to simulation-based investigations as it requires hardware counters for information collection and is hardly adaptable to the variety of core/memory configurations.

Endo et al. [17] show the micro-architectural simulation of ARM Cortex-A cores of the big.LITTLE processor by using the gem5/McPAT frameworks and validate area and energy/performance trade-offs against the published datasheet information. Their work does not focus on the multicore evaluation and only demonstrates the difference between Cortex-A7 and A15 single-cores running single-threaded applications. In [18] the authors design a gem5 model of CoreTile *Express* SoC and estimate the accuracy of Cortex-A15 core, memory system and interconnect. They deeply explore the micro-architectural simulation for the homogeneous dual-core system. Authors report the runtime error of the SPEC benchmark being within 40%.

Our work advances state-of-the-art by addressing the performance and power simulation of the heterogeneous ARM big.LITTLE multicore architecture. Models are shown to have sufficient accuracy compared to an actual SoC for enabling architectural investigations, and are further made freely available.

III. Performance and Power Models

The gem5 simulator [10] is a powerful cycle-approximate simulation framework [19] supporting multiple ISAs, CPU models, detailed memory systems including cache coherent protocols, interconnects and memory controllers. It further produces statistical information enabling to estimate power consumption and footprint area with the Multicore Power, Area, and Timing (McPAT) modeling framework [11].

A. Architecture modeling

The Odroid XU3 computer board built around the Exynos 5 Octa (5422) chip is used as reference platform. The general architecture parameters are taken from publicly available sources [5] reported in Table I.

*1) **Overview of the Exynos 5 Octa (5422) SoC**:* The Exynos 5 Octa (5422) chip shown in Figure 1 features two clusters, “big” and “LITTLE”, each of which consists of quad Cortex-A15 and quad Cortex-A7 cores respectively. Clusters operate at independent frequencies, from 200MHz up to 1.4GHz for the LITTLE and up to 2GHz for the big.

Each core has its private L1 instruction (I) and data (D) caches. And each of both clusters has its own L2 cache shared among all cluster cores. The L2 sizes differ, the Cortex-A7 cluster has a smaller 512kB L2 cache whereas the Cortex-A15

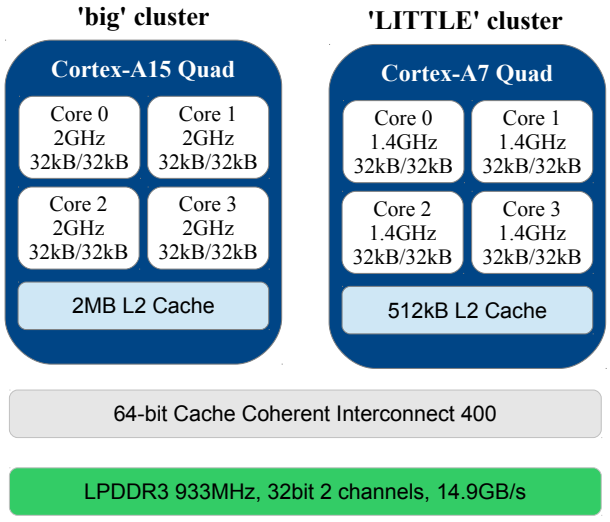


Fig. 1: ARM big.LITTLE technology.

[1] http://www.lirmm.fr/ADAC

TABLE I: 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
Interconnect		CCI-400 64-bit	
Memory		2 GB LPDDR3 RAM 933 MHz, 14.9 GB/s, 32-bit, 2 channels	

TABLE II: Exynos 5 Octa (5422) SoC technology.

Parameters	LITTLE	big
Power model		
Technology	28 nm CMOS	
V_{dd}@200/200MHz	0.91 V	0.91 V
V_{dd}@1.4/2GHz	1.24 V	1.3 V
$Temperature$ @200/200MHz	310-320 K	310-320 K
$Temperature$ @1.4/2GHz	310-320 K	320-330 K

has 2MB L2 cache. L2 caches are connected to the DRAM memory via the 64-bit Cache Coherent Interconnect (CCI) 400 [20]. The SoC incorporates its own system memory in the form of 2GB LPDDR3 RAM. It runs at 933MHz frequency and with 2x32 bit bus achieves 14.9GB/s memory bandwidth.

*2) **big.LITTLE model implementation in gem5**:* Following the reference Exynos 5 Octa (5422) SoC specification, we configure the simulation system. The gem5 framework provides a set of CPU models including in-order and out-of-order models. The fine-tuning of the micro-architectural parameters of the in-order Cortex-A7 and out-of-order Cortex-A15 cores are performed according to published recommendations [18][21][17][22]. This concerns features comprising the execution stage configuration, functional units, branch predictor, physical registers, etc.

To overcome the limitations of gem5 full-system mode, we implemented a set of enhancements:

- **Support for 8 ARM cores**: the first limitation is related to actually available ARM MPCore processor model which contains maximum 4 ARM v7 cores. To run 8 ARM-cores system we modify the description of the Snoop Control Unit (SCU) register [23]. The SCU count therefore contains no masked number of cores.
- **Heterogeneous multicore**: to build the clustered system, e.g. quad-core ARM Cortex-A15 with quad-core ARM Cortex-A7, the creation script has been enriched by the possibility to include various CPU models.
- **Multiple frequency domains**: to enable the big and LITTLE clusters operations at different frequencies we supplied full-system simulation mode with the ability to assign distinct clocks to individual cores.
- **Multiple shared L2 caches**: we add the option to identify the L2 cache number to full-system simulation mode. The big.LITTLE technology assumes cache coherency even when all eight cores are working simultaneously. This sophisticated task is performed at hardware level by means of coherent interconnect. Due to the fact that the particular ARM CCI-400 is not implemented in gem5, we use the CoherentXBar component. It can be used as a template for modeling coherent buses and is typically used for the L1-to-L2 buses and as the main system interconnect [10].

Some modifications were performed in the Linux kernel source code so as to enable gem5 full-system support:

- **Ability to boot 8 cores simultaneously**. This modification relates to that described in Section III-A aimed at enabling a higher core count in the hardware model, at the SCU-level. The corresponding function fetching the number of available cores from the hardware register has been modified accordingly.
- **Global Interrupt Controller support**: The *cpu_logical_map* function presented in Linux kernel 3.10 does not allow to use it in gem5, the former implementation (Linux kernel 3.7) is here used.

B. Power estimation

The McPAT framework allows estimating power consumption based on the statistics collected during gem5 simulation. We configure the general architecture parameters and McPAT parameters according to Table II.

The Exynos 5 Octa (5422) SoC is built using a 28nm CMOS process. The supply voltage, V_{dd} and operating temperature, T, are experimentally measured on the Odroid XU3 board by means of querying internal sensors. V_{dd} values depend on the Linux kernel configuration and are related to the Adaptive Supply Voltage (ASV) technique used in Samsung SoCs.

The operating temperature strongly depends on the cluster architecture and application nature. For the Cortex-A7 cluster the temperature always remains below 323K and the board fan stays off. For the Cortex-A15 cluster the temperature rises above 323K and the board fan is quickly triggered so as to ensure proper cooling.

gem5 does not generate statistics concerning operating temperature. So that we analyzed benchmark execution on the board and explicitly specified an averaged temperature per application execution.

IV. Accuracy assessment

A. Experimental setup

*1) **System configurations**:* The reference Odroid XU3 board runs Ubuntu 14.04 OS on Linux kernel LTS 3.10. For a better power saving, the Linux kernel offers a set of CPU frequency scaling features. The general frequency scaling policy for CPU is defined by the *scaling governor* thanks to a dedicated power scheme [24].

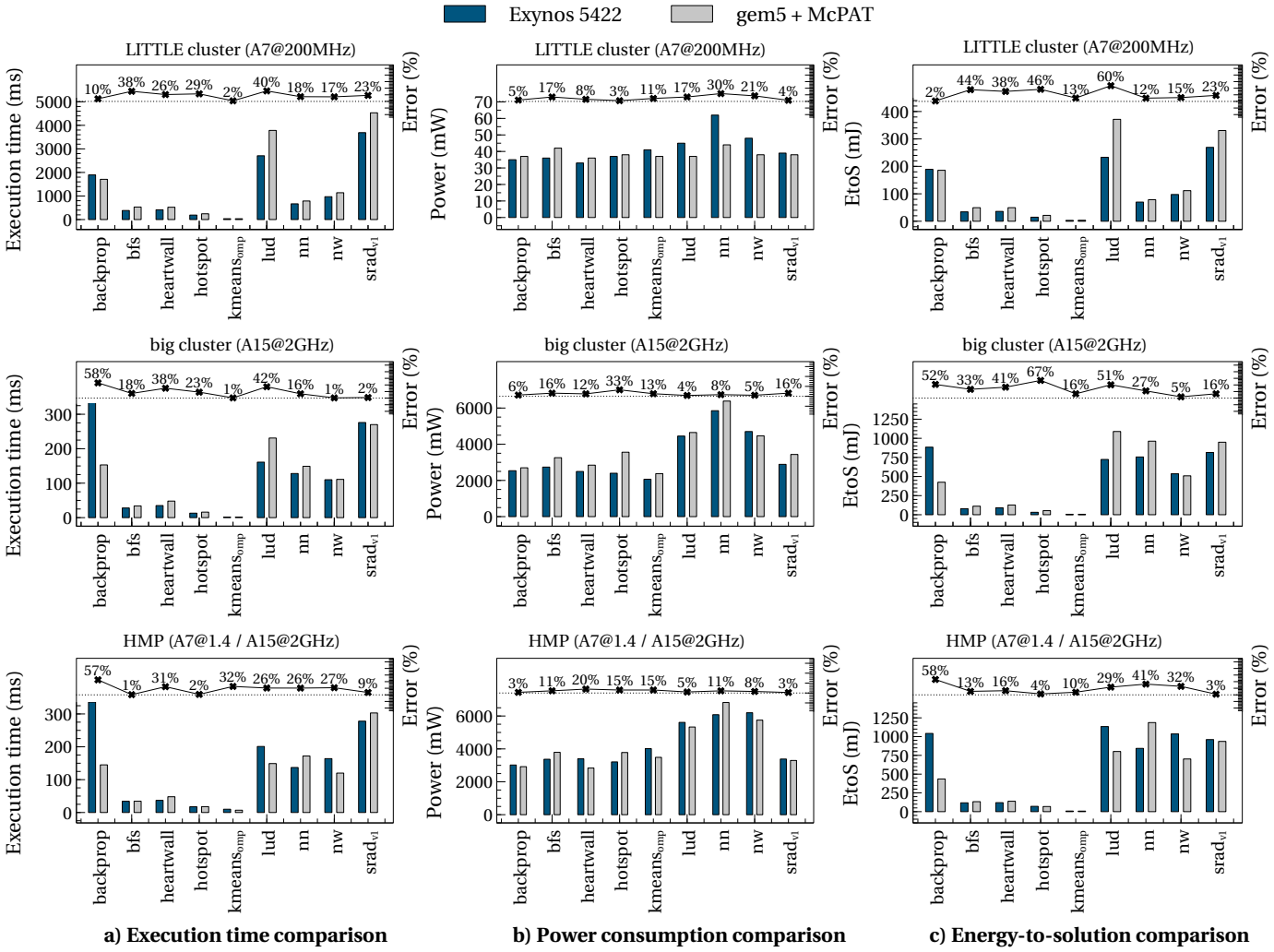


Fig. 2: Execution time and power consumption comparison (gem5/McPAT versus Exynos 5 Octa).

The *ondemand* and the *conservative* governors dynamically set CPU frequencies depending on the current usage and according to the specified thresholds. While the *ondemand* governor jumps from the maximum to the minimum frequency, the *conservative* governor increases and decreases the CPU speed gracefully.

The *performance* and the *powersave* governors set the CPU statically to the highest and to the lowest frequency respectively denoted by `scaling_min_freq` and `scaling_max_freq` values.

For our experiments, we statically set the CPU frequency by means of the *performance* governor. So, the DVFS is disabled. In order to evaluate the impact of different CPU frequencies on the simulation error, we consider three different frequencies: minimum, medium and maximum. Due to the fact that the LITTLE and the big clusters have different clock limits (see Table I), the chosen frequency sets are 200MHz, 800MHz and 1.4GHz for the LITTLE cluster and 200MHz, 1.1GHz and 2GHz for the big cluster.

In addition to CPU frequency scaling, the Linux kernel includes support for CPU and core migration. We do not consider these features in our work. However, we use the provided functionality to activate and deactivate the big and LITTLE clusters. Indeed, when only one cluster is activated the workloads are executed in SMP mode. In such way, we isolate a specific part of the system to separately assess simulation accuracy.

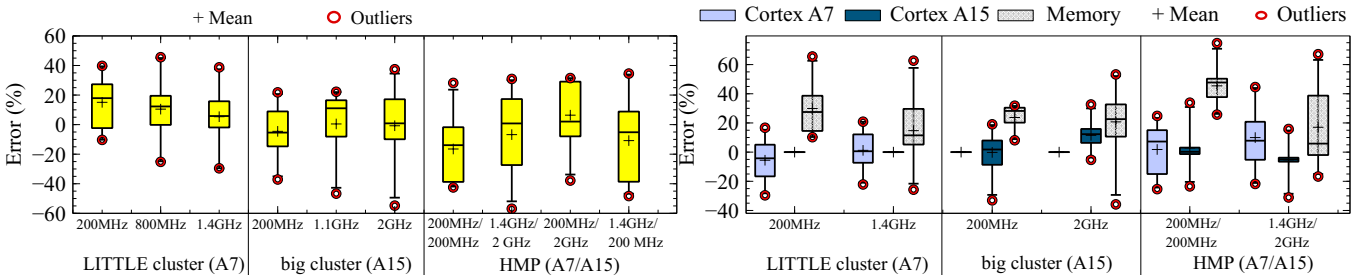
Note that throughout all experiments we do not use the embedded GPU.

Thereby, the following three scenarios are investigated:

- **Scenario I:** Cortex-A7 cluster, *LITTLE*, only running in SMP mode at 200MHz, 800MHz and 1.4GHz,
- **Scenario II:** Cortex-A15 cluster, *big*, only running in SMP mode at 200MHz, 1.1GHz and 2GHz,
- **Scenario III:** Cortex-A7 and Cortex-A15, *big.LITTLE*, running in HMP mode at 200/200MHz, 200MHz/2GHz, 1.4GHz/200MHz and 1.4/2GHz respectively.

*2) **Benchmarks**:* The study is conducted using the `Rodinia` benchmark suite and `lmbench` micro-benchmark.

The `Rodinia` benchmark [12] is used throughout the rest



(a) Execution time error distribution

(b) Power error distribution

Fig. 3: Error distribution.

TABLE III: Benchmark parameters.

Application	Acronym	Problem size
Rodinia benchmark		
Back Propagation	*backprop*	65 536
Breadth-First Search	*bfs*	4096
Heart Wall	*heartwall*	test.avi, 1 frame
HotSpot	*hotspot*	64x64
Kmeans	*kmeans openmp*	100
LU Decomposition	*lud*	256
k-Nearest Neighbors	*nn*	42760
Needleman-Wunsch	*nw*	1024
SRAD	*srad v1*	1x502x458
	srad v2	512x512
Lmbench benchmark		
Memory read latency	*lat_mem_rd*	64M 512 stide
Memory bandwidth	*STREAM*	2.000.000
CPU operations latency	*lat_ops*	11 repetitions

of the paper. It is composed of applications and kernels from different domains such as bioinformatics, image processing, data mining, medical imaging and physics simulation. It also includes simpler compute-intensive kernels such as LU decomposition and graph traversal. `Rodinia` targets performance benchmarking of heterogeneous systems, for that reason CUDA, OpenMP and OpenCL implementations are available. Here, the OpenMP implementation is chosen, with four threads per cluster, i.e., one thread per core. We use *static* loop scheduling policy. Threads are bound to specific CPU cores by using `GOMP_CPU_AFFINITY` environment variable. This ensures a similar workload execution on the real SoC and our gem5 model.

Out of the `Rodinia` benchmark suite we select the following subset of benchmarks: *backprop*, *bfs*, *heartwall*, *hotspot*, *kmeans openmp*, *lud*, *nn*, *nw* and *srad v1/v2*. Based on dwarves classification proposed in [25], the chosen subset contains Structured Grid (*heartwall*, *hotspot*, *srad*), Unstructured Grid (*backprop*), Dynamic Programming (*nw*), Dense Linear Algebra (*kmeans*, *lud*, *nn*) and Graph Traversal (*bfs*) dwarves.

The selected problem size is presented in Table III.

The `lmbench` micro-benchmark [26] is used to evaluate the latencies provided by our performance model and compare them to the Exynos 5 Octa SoC. The selected application subset is presented in Table III and includes *lat_mem_rd*, *STREAM* and *lat_ops*.

- *lat_mem_rd* provides the latency of the entire memory hierarchy including data cache, L2 cache and main memory. It measures the time to do about 1.000.000 loads varying two parameters, array size and array stride. The reported time represents only memory latency and does not include load instruction execution time. It is assumed that all cores can do a load instruction in one cycle. The benchmark has been validated by logic analyzer measurements on an SGI Indy [26].
- *STREAM* is a synthetic program that measures the memory bandwidth (in MB/s) and calculates the corresponding rate for simple vector kernels, such as copy, scale, add and triad [27]. To measure the bandwidth from the main memory each array must be at least 4x the size of the sum of all the last-level caches used in the run. For the chosen platform with two L2 caches of 512kB and 2MB, we use 2.000.000 array size, which meets the condition.
- *lat_ops* measures the latency of basic CPU operations, such as integer ADD, float MUL, uint64 XOR, etc. The benchmark is configured to use interlocking operations, so it measures the time of an individual operation. In addition, it uses relatively short vectors thus the operations should be going to/from L1 or L2 caches, rather than main memory, which reduces the memory overheads. Authors however affirm that the benchmark is experimental and may give erroneous results [28].

B. Results

Running the `Rodinia` benchmark, we evaluate performance, power and energy metrics, i.e. application execution time, power consumption and energy-to-solution (EtoS). The values given by the gem5 and McPAT models are compared with the measured on the Odroid XU3 board. Comparison results for three configurations, e.g. LITTLE cluster at 200MHz, big cluster at 2GHz and big.LITTLE at 1.4/2GHz, are presented in Figure 2.

1) ***Performance comparison****:* The execution time values reported in Figure 2 a) are averaged over 5 subsequent runs for ensuring the consistency. The absolute error varies significantly depending on the configuration and application, ranging from 1% to 57%. For the most applications, LITTLE

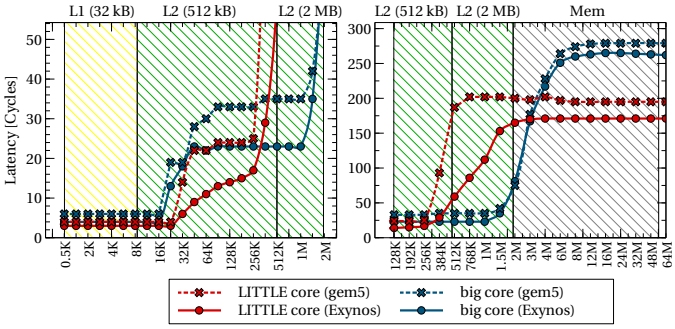


Fig. 4: Memory hierarchy latency measurements.

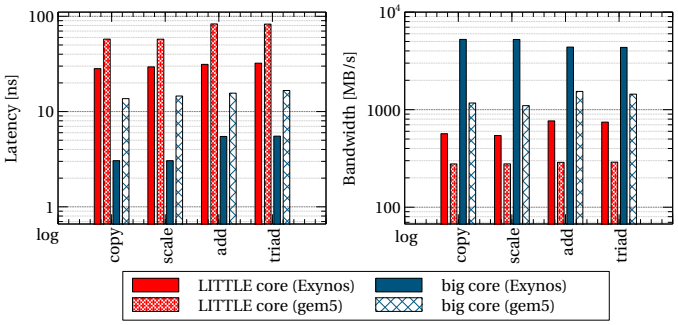


Fig. 5: Latency and bandwidth of the main memory controller.

cluster model provides slower execution time compared to the reference SoC. Similarly, big cluster model shows an important slowdown for more than a half of given applications. In HMP mode, big.LITTLE model provides more variety, e.g. *lud* and *nw* run faster on the model than on the SoC.

Figure 3 a) shows a box-plot reporting observed mean error between the model and the board. The average absolute error percentage throughout all applications for the LITTLE cluster is 18.8%, 20.1% for the big cluster and 22.9% for the big.LITTLE system. On an average, the gem5 model predicts performance with a 20% error.

2) ***Power comparison****:* The power consumption values estimated with the McPAT framework account for runtime dynamic power and leakage power. The reported total power consumption contains the Cortex-A7 cluster power, Cortex-A15 cluster power and memory controller power. Other peripherals such as storage, network and cooling are therefore here not considered. The Exynos 5 Octa SoC values are measured over 5 subsequent runs and represent average power consumption per application. It does not introduce power variation during an execution.

Comparison results are shown in Figure 2 b). The corresponding absolute error percentage of total power is averaged throughout all given applications. It is 12.7%, 11.7% and 10.8% for the LITTLE cluster, big cluster and big.LITTLE HMP respectively.

The error percentage distribution is shown in Figure 3 b) in form of a box-plot. It presents the error separately for LITTLE cluster, big cluster and memory controller. The average absolute error percentage throughout all applications for the LITTLE cluster is 14.9%, 11% for the big cluster and 28.5% for the memory.

The largest error relates to the memory controller. The external memory controller error is likely that most influenced by cache and interconnect modeling inaccuracy. Nevertheless, these results allow estimating SoC power consumption ranging from tens of mW to several W.

3) ***Energy comparison****:* Based on the simulated execution time and the above total power results we calculate the EtoS and compared it with the values measured on the board. Comparison results for all applications are shown in Figure 2 c). Observed average absolute error is 21.9%, 27.9% and 22.1% for the LITTLE cluster, big cluster and HMP big.LITTLE respectively.

The EtoS error percentage includes both, gem5 execution time error and McPAT power consumption error. Therefore, such scenarios as Cortex-A7 cluster running at 200MHz and Cortex-A15 cluster running at 2GHz cumulates the error and shows higher mismatch.

4) ***Latency comparison****:* Figure 4 shows the results from the `lmbench` *lat_mem_rd* latency measurements plotted as a series of data sets. Each data set represents a stride size, with the array size varying from 512 bytes up to 64M. The horizontal plateaus represent different levels in the memory hierarchy, i.e. data L1 cache, L2 caches and main memory. The point where each plateau ends shows the end of that portion of the memory hierarchy.

The LITTLE and the big cores have the same 32 kB size of L1 cache. The L1 data cache latency of the Exynos 5 Octa (5422) SoC is 3 and 4 cycles for LITTLE and big cores respectively. These values correspond to the parameters specified in the model and presented in Table I. However, the benchmark measures 4 and 6 cycles for LITTLE and big cores in our gem5 model. L2 caches and main memory also show higher latency in our gem5 model compared to the real SoC. A significant mismatch is observed at the transition point between L2 cache and main memory of LITTLE core.

Figure 5 shows the results of *STREAM* execution on our model and on the Exynos 5 Octa SoC. All operations running on the model provide significant slowdown and consequently lower memory bandwidth. These results correspond to the measurements of the memory hierarchy latency shown in Figure 4.

Figure 6 shows the results of *lat_ops* execution and reports latency of the following basic CPU operations:

- interger BIT, ADD, MUL, DIV, MOD;
- (u)int64 BIT, ADD, MUL, DIV, MOD;
- float ADD, MUL, DIV, bogomflops;
- double ADD, MUL, DIV, bogomflops.

The LITTLE core model shows divergent latency mismatches. Operations such as integer BIT, MUL, DIV, MOD and int64 ADD, DIV, MOD provide up to 46% higher latency compared to the ARM Cortex-A7 core in the real SoC. All other operations show an opposite behaviour and provides

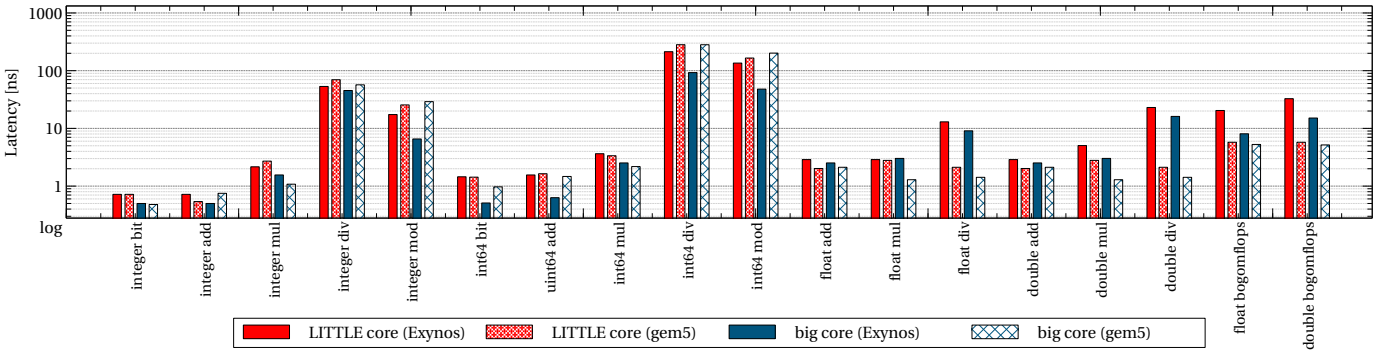


Fig. 6: Latency of basic CPU operations.

faster operation execution time. In some cases, e.g. float DIV and double DIV, gem5 model operates almost 10 times faster than the real SoC. In average, the absolute error percentage for all CPU operations compared to the Exynos 5 Octa (5422) SoC is around 30%.

The first group of CPU operations that are executed slower includes integer ADD, DIV, MOD and int64 BIT, ADD, DIV, MOD operations. Some of them are executed 3 times slower compared to the real SoC. The remaining operations are executed faster and show up to 10x speedup. Except these operations that show peak mismatch, the absolute error percentage of the big core model corresponding to the ARM Cortex-A15 in the Exynos 5 Octa (5422) SoC is around 36%.

V. Insights on Simulation Error Sources

Black and Shen [29] distinguished three separate categories of error sources: (i) *Modeling error* occurs when the simulator functionality is implemented erroneously due to the developer fault; (ii) *Specification error* occurs when the model developer has untruthful information or has no access to the relevant information; (iii) *Abstraction error* occurs when some components are abstracted or simplified. Based on the above classification, we discuss the sources of error related to our modeling approach. We are mainly concerned by abstraction and specification errors.

The first source of error relates to non-fully cycle-accurate modeling of the processor microarchitecture. The modeling approach proposed in [21] has been validated for the Cortex-A8 processor and demonstrated the execution time error around 8%. It has not been validated for the Cortex-A7 core. In Figure 3 a) we observe that the error mean of Cortex-A7 cluster is between 5% and 10% higher than that of the Cortex-A15 cluster. Thus the in-order implementation is less accurate than the out-of-order model.

The second source of error is related to the specification error. Namely, it includes the lack of information about processor microarchitecture, which is confidential and cannot be accessed by research community. This includes unknown optimizations, such as DRAM controller or the proprietary coherent interconnect, implemented by the chip manufacturer and IP provider, i.e. Samsung and ARM in our case respectively. Accordingly, we were forced to do a best effort modeling of those components. Based on the results, we conclude that these modeling and specification errors account for a major part of the total modeling error.

The third source of error concerns simplistic implementation of the cache coherent interconnect and coherency protocol, as well as specific timings of the used DDR memory controller. The `lmbench` latency measurements show delayed memory hierarchy access that in case of communication intensive applications may lead to significant slowdown.

In addition to the listed error sources, the variability of the measurement process can provide an important impact on the results. This introduces the fourth source of error, e.g. *observation* or *measurement* error [30].

A statistical analysis of the values measured on the Odroid XU3 board shows the average of the absolute deviation between 0% and 5% among the considered set of `Rodinia` benchmarks. In gem5 full-system simulation, a couple of OS boot and single workload execution repeated multiple times does not provide values variation. However, multiple workload runs under a single OS simulation give significant deviation between 0% and 22% showing a high mismatch for the first run. Due to the cache hierarchy warm-up, the following runs demonstrate negligible variation.

Based on the above analysis, we observe that the primary efforts for more accurate modeling should be spent on the memory system implementation. For that we will certainly need standardization efforts that render the required information public. This conclusion is also aligned with the work presented in [18], in which authors state that microarchitecture-level variations do not lead to significant changes in execution time as opposed to variations on memory architecture.

Despite the assumptions that had to be taken, measurements show that our average modeling error is about 20%. Importantly, we observe that our model is able to largely track the dynamics exposed in the execution of different benchmark kernels. Therefore, we conclude its suitability to undertake the architecture exploration.

VI. Conclusion

In this paper, we propose performance and power models of the ARM big. LITTLE multicore architecture implemented in gem5 and McPAT simulation frameworks. These models have

been calibrated and validated w.r.t. the Exynos 5 Octa (5422) chip running the `Rodinia` benchmark suite and `lmbench` synthetic micro-benchmark. The presented results have shown average error around 20% for performance, 13% for power consumption and 24% for energy-to-solution. We provide some useful insights about major simulation error sources and their associated impact on performance assessment.

Future work includes modeling of ARMv8 big.LITTLE configurations, alongside with the use of more suitable programming models that enable a better runtime assignment of threads to cores depending on their respective nature.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2015] under the Mont-blanc 2 Project (www.montblanc-project.eu), grant agreement nº 610402.

REFERENCES

[1] R. Kumar, D. M. Tullsen, P. Ranganathan, N. P. Jouppi, and K. I. Farkas, "Single-isa heterogeneous multi-core architectures for multithreaded workload performance," in *Proceedings of the 31st Annual International Symposium on Computer Architecture*, ser. ISCA '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 64–.

[2] B. Jeff, "big.little technology moves towards fully heterogeneous global task scheduling," November 2013, White Paper. [Online]. Available: http://www.arm.com/files/pdf/

[3] NVIDIA, "Tegra mobile processors," http://www.nvidia.com, 2015.

[4] NVIDIA Corporation, "The benefits of quad core CPUs in mobile devices," 2011, White Paper. [Online]. Available: http://www.nvidia.com/object/white-papers.html

[5] Samsung, "Exynos Octa SoC," https://http://www.samsung.com/, 2015.

[6] ARM Ltd., "big.little technology: The future of mobile," 2013, White Paper. [Online]. Available: https://www.arm.com/

[7] K. Van Craeynest and L. Eeckhout, "Understanding fundamental design choices in single-isa heterogeneous multicore architectures," *ACM Trans. Archit. Code Optim.*, vol. 9, no. 4, pp. 32:1–32:23, Jan. 2013.

[8] K. Van Craeynest, A. Jaleel, L. Eeckhout, P. Narvaez, and J. Emer, "Scheduling heterogeneous multi-cores through performance impact estimation (pie)," in *Proceedings of the 39th Annual International Symposium on Computer Architecture*, ser. ISCA '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 213–224.

[9] M. Pricopi, T. Muthukaruppan, V. Venkataramani, T. Mitra, and S. Vishin, "Power-performance modeling on asymmetric multi-cores," in *Compilers, Architecture and Synthesis for Embedded Systems (CASES), 2013 International Conference on*, Sept 2013, pp. 1–10.

[10] "The gem5 Simulator," http://www.gem5.org/docs/, 2015.

[11] Hewlett-Packard, "Mcpat," http://www.hpl.hp.com/research/mcpat/, 2008.

[12] S. Che, M. Boyer, J. Meng, D. Tarjan, J. Sheaffer, S.-H. Lee, and K. Skadron, "Rodinia: A benchmark suite for heterogeneous computing," in *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*, Oct 2009, pp. 44–54.

[13] S. Sarma and N. Dutt, "Cross-layer exploration of heterogeneous multicore processor configurations," in *VLSI Design (VLSID), 2015 28th International Conference on*, Jan 2015, pp. 147–152.

[14] T. Muthukaruppan, M. Pricopi, V. Venkataramani, T. Mitra, and S. Vishin, "Hierarchical power management for asymmetric multi-core in dark silicon era," in *Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE*, May 2013, pp. 1–9.

[15] K. Yu, D. Han, C. Youn, S. Hwang, and J. Lee, "Power-aware task scheduling for big.little mobile processor," in *SoC Design Conference (ISOCC), 2013 International*, Nov 2013, pp. 208–212.

[16] C. Tan, T. Muthukaruppan, T. Mitra, and L. Ju, "Approximation-aware scheduling on heterogeneous multi-core architectures," in *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*, Jan 2015, pp. 618–623.

[17] F. A. Endo, D. Couroussé, and H.-P. Charles, "Micro-architectural simulation of embedded core heterogeneity with gem5 and mcpat," in *Proceedings of the 2015 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*, ser. RAPIDO '15. New York, NY, USA: ACM, 2015, pp. 7:1–7:6.

[18] A. Gutierrez, J. Pusdesris, R. Dreslinski, T. Mudge, C. Sudanthi, C. Emmons, M. Hayenga, and N. Paver, "Sources of error in full-system simulation," in *Performance Analysis of Systems and Software (ISPASS), 2014 IEEE International Symposium on*, March 2014, pp. 13–22.

[19] A. Butko, R. Garibotti, L. Ost, and G. Sassatelli, "Accuracy evaluation of gem5 simulator system," in *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2012 7th International Workshop on*, July 2012, pp. 1–7.

[20] *CoreLink CCI-400 Cache Coherent Interconnect Technical Reference Manual*, ARM, November 16 2012, revision r1p1.

[21] F. Endo, D. Courousse, and H.-P. Charles, "Micro-architectural simulation of in-order and out-of-order arm microprocessors with gem5," in *Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV), 2014 International Conference on*, 2014, pp. 266–273.

[22] I. Pavlov, "7-zip lzma benchmark samsung exynos 5250 arm cortex-a15," http://7-cpu.com/cpu/Cortex-A15.html, 2015.

[23] *Cortex-A9 MPCore*, ARM, November 27 2009, revision r2p0.

[24] D. Brodowski and N. Goldeg, "Rodinia:accelerating compute-intensive applications with accelerators," http://lava.cs.virginia.edu/Rodinia/, 2014.

[25] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, and K. A. Yelick, "The landscape of parallel computing research: A view from berkeley," TECHNICAL REPORT, UC BERKELEY, Tech. Rep., 2006.

[26] L. McVoy and C. Staelin, "Lmbench: Portable tools for performance analysis," in *Proceedings of the 1996 Annual Conference on USENIX Annual Technical Conference*, ser. ATEC '96. Berkeley, CA, USA: USENIX Association, 1996, pp. 23–23. [Online]. Available: http://dl.acm.org/citation.cfm?id=1268299.1268322

[27] J. D. McCalpin, "Memory bandwidth and machine balance in current high performance computers," *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pp. 19–25, Dec. 1995.

[28] C. Staelin and L. McVoy, "Lmbench - tools for performance analysis," http://lmbench.sourceforge.net.

[29] B. Black and J. Shen, "Calibration of microprocessor performance models," *Computer*, May 1998.

[30] M. M. Bland and D. G. Altman, "Statistics Notes: Measurement error," *BMJ*, vol. 313, no. 7059, 1996. [Online]. Available: http://bmj.bmjjournals.com