Micro-architectural simulation of embedded core heterogeneity with gem5 and McPAT

Fernando A. Endo, Damien Couroussé and Henri-Pierre Charles
Univ. Grenoble Alpes, F-38000 Grenoble, France
CEA, LIST, MINATEC Campus, F-38054 Grenoble, France
{fernando.endo,damien.courousse,henri-pierre.charles}@cea.fr

ABSTRACT

Energy consumption is the major factor limiting performance in embedded systems. In addition, in the next generations of ICs, heat or energy constraints will not allow to power all transistors simultaneously. Heterogeneous multicore systems represent a possible solution to this problem: the diversity of cores provides energy and performance trade-offs.

Micro-architectural simulators allow a fast evaluation of such new hardware implementations. Currently, there is no open-source simulator that can estimate the energy and performance trade-offs of asymmetric ARM cores at the micro-architectural level.

This article presents a micro-architectural simulator of ARM Cortex-A cores, capable of estimating the performance, power and area of core asymmetry. Our simulation framework is based on the open-source gem5 and McPAT simulators. The main contribution is to report our experience with both simulators. We detail how we simulated the CPUs of a big.LITTLE system and validate area estimations and energy/performance trade-offs against published information.

Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: Simulation Support Systems—*Simulation environments*; C.4 [**Performance of Systems**]: Modeling techniques; C.1 [**Processor architectures**]: Single Data Stream Architectures

1. INTRODUCTION

In high-performance embedded systems, reduced energy consumption is essential for providing more performance. As current transistor technologies can not efficiently reduce the power densities in ICs, next generations of processors will need to improve performance with only incremental increases of power budget.

A good way to increase the total throughput of a processor with a limited power budget is to replace big and power-hungry out-of-order cores by little and energy-efficient in-order ones. However, as single-thread performance is still important in the embedded market, core asymmetry is a key strategy to have both single-thread peak performance and acceptable power dissipation when the system load is high.

To estimate the energy consumption of processors, instruction set simulators are fast and provide good estimations, nonetheless they are mostly calibrated to a specific hardware [9, 15]. In order to be flexible and able to simulate emerging hardware, micro-architectural simulators offer a good trade-off between precision and simulation speed. However, to the best of our knowledge, no open-source micro-architectural simulator can estimate the energy/performance trade-offs of asymmetric ARM cores.

In this paper, we present our performance, power and area simulation platform based on gem5 [5] and McPAT [13], which simulates in-order and out-of-order Cortex-A cores at the micro-architecture level. Both simulator are complex tools, and configuring them to simulate realistic hardware is not straightforward. Our main contribution is to report our experience with gem5 and McPAT. In more details:

- We detail how we simulated the CPUs of a big.LITTLE processor.
- We validate the area estimation of McPAT for the Cortex-A7 and Cortex-A15 CPUs against published data.
- We show energy and performance trade-offs of cluster asymmetry, and compare our estimations of the Dhrystone benchmark with published results.

2. RELATED WORK

SimpleScalar [6] was one of the first micro-architectural simulators, initially simulating a MIPS-like architecture and later also Alpha processors. A version supporting ARM was released, but only with functional and timing accurate modes. To support the ARM ISA, Sim-Panalyzer [21] augmented the out-of-order model of SimpleScalar and included power estimations of major sources. An well-known problem of SimpleScalar is that it can not accurately simulate multicore systems, because its trace-driven simulation is not adapted to model the communication between cores.

gem5 [5] is a cycle-accurate micro-architectural simulator, which can simulate multicore systems and supports the ARM ISA, including floating-point (VFP) and Advanced SIMD extensions (NEON). However, its in-order model is not currently functional for ARM.

McPAT [13] models together power, area and timing of multicore and manycore architectures, supporting both in-order and out-of-order pipelines. For power estimation, the utilization statistics of SoC components including those of the pipeline should be provided by hardware counters or a micro-architectural simulator.

SST [11] is a simulation framework which integrates specific versions of gem5, McPAT and HotSpot [17] as libraries and is able to simulate the feedback of temperature on leakage power. Unfortunately, SST currently only supports the x86 architecture.

Our previous work detailed the implementation and validation of a cycle-approximate in-order pipeline in gem5, by modifying the out-of-order model [8]. This work differs from that previous one, because we integrate gem5 and McPAT and show energy/performance trade-offs of asymmetric ARM cores.

Previous work involved micro-architectural simulation of heterogeneous/asymmetric or configurable cores. Kumar et al. were the first to show the energy benefits of having different types of Alpha cores in a SoC [12]. Suleman et al. proposed an asymmetric x86 multicore architecture to accelerate the execution of critical sections [19]. Bahar and Manne proposed a technique called Pipeline Balancing, where an 8-way out-of-order Alpha machine can have its issue width changed to 6 or 4 in order to reduce energy consumption [3]. The studies of Shifer and Weiss [16] and Lukefahr et al. [14] analyzed the trade-offs of clustered asymmetric cores, in x86-64 and Alpha simulators respectively. Its worth observing that previous studies analyzed the energy and performance trade-offs of core asymmetry simulating x86 and Alpha ISAs. To the best of our knowledge, our work is the first to simulate such trade-offs with the ARM ISA, which is more relevant in embedded systems.

3. SIMULATION FRAMEWORK

This section details the performance and energy simulation of Cortex-A cores.

3.1 gem5

gem5 [5] is a cycle-accurate micro-architectural simulator. For micro-architectural simulation with the ARM ISA, currently only the out-of-order model (O3) is functional. This model can boot unmodified Linux images in the Full-System (FS) mode, which simulates a complete system. Our previous work detailed the components and configurations of the O3 model, and how we simulated an in-order pipeline based on this model [8].

3.2 McPAT

McPAT is a micro-architectural multicore/manycore power and area estimator [13]. The user interface with the simulator is provided through an XML file, which describes system parameters and utilization statistics of components. Given the processor parameters, McPAT builds an internal chip model, modeling and estimating the area of architectural elements like caches, NoCs and cores. Intra-core elements include each pipeline stage of in-order or out-of-order designs. For average power estimation, the XML file should contain the utilization statistics of core and other SoC components. In our framework, these statistics come from gem5, through a modified version of a publicly available parser [18].

Energy and power models. For most micro-architectural elements and caches, the energy cost per access is estimated. The energy consumption of these elements is then calculated multiplying the number of accesses by their energy cost. At a first glance, this technique based on the energy cost per access gives the impression that dynamic energy is independent of the instruction timing. However, it's exactly the role of a performance simulator to estimate the micro-architectural accesses in a dynamic environment, taking into account the timing of instructions. For example, in an out-of-order core, the speculation and hence the accesses to pipeline structures may vary depending on cache latencies. For McPAT, what matters is the total number of accesses (including speculative ones). Differently, the energy consumption of functional units (FUs) is not modeled per access, but per cycle. In consequence, multiple cycle accesses should be counted multiple times. McPAT also estimates the leakage power at a given temperature.

ARM support. The pipeline models are based on Intel and Alpha designs, but low-power embedded SoCs are also supported. For example, when the Embedded flag is activated in the input file, McPAT models the VFP/NEON functional unit of the Cortex-A9.

4. EXPERIMENTAL SETUP

This section presents the experimental setup. First, we detail the configuration of the asymmetric clusters of cores. Then, the benchmarks used in our comparisons are presented.

4.1 Configuring gem5 and McPAT to simulate big.LITTLE CPUs

For performance estimation, our simulator is based on a modified gem5 version [8], running in the FS mode. Power and area estimation are performed by McPAT 1.0 [13].

The system parameters of this experiment are based on the ODROID-XU3 board, which embeds an Exynos 5244 Application Processor. This processor is a big.LITTLE system with two clusters of four cores each, one with Cortex-A7 cores, and the other with Cortex-A15. Table 1 shows the main system and CPU parameters.

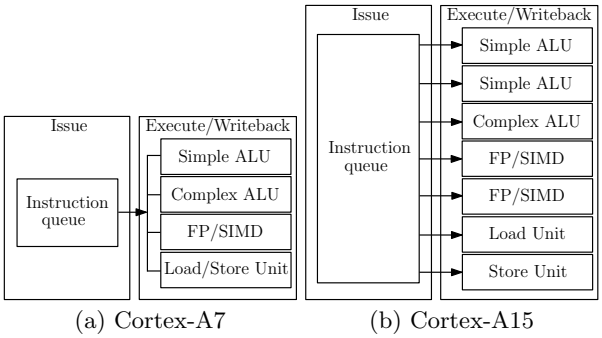
For both cores, instruction timing in the execution stage is not publicly available. To configure them in gem5, we proceeded as follows. In both cores, integer instructions have the same latencies of one, four and twelve cycles for ALU, multiply and divide, respectively. For floating-point, we estimated their timing taking as reference our gem5 configu-

Table 1: Parameters of Cortex-A7 and A15 CPUs

Parameter		Cortex-A15 (out-of-order)	Cortex-A7 (in-order)
Core clocks		2.0 GHz	1.4 GHz
DRAM	Size	256 MB[1]	256 MB[1]
	Clock	933 MHz	933 MHz
	Latency (ns)	81[2]	81[2]
L2	Size	2048 kB	512 kB
	Associativity	16	8
	Latency[3]	8	3
	MSHRs	11	8
	Write buffers	16	16
L1-I	Size	32 kB	32 kB
	Associativity	2	2
	Latency[3]	1	1
	MSHRs	2	2
L1-D	Size	32 kB	32 kB
	Associativity	2	4
	Latency[3]	1	1
	MSHRs	6	4
	Write buffers	16	4
Stride prefet.	Cache level	2	1
	Degree	1	1
	Buffer size	16	8[2]
Global BP	Entries	4096[4]	256
	Bits	2[4]	2[2]
Local BP	Entries	1024[4]	N/A
	Bits	3[4]	N/A
BTB entries		4096[4]	256[5]
RAS entries		48	8
ITLB/DTLB entries		128 each[6]	32 each[6]
Front-end width		3	1[7]
Back-end width		7[7]	1[7]
Pipeline depth (INT/FP)		15/24	8/10
Physical INT/FP registers		90/256[2]	N/A
IQ entries		48	16[2]
LSQ entries		16 each	8 each[2]
ROB entries		60	N/A

[1] gem5 FS mode limitation.
[2] Educated guess.
[3] Latencies in core clock cycles.
[4] Based on Alpha 21064.
[5] The A7 does not implement a BTB, but an equivalent structure holding instructions.
[6] Both A7 and A15 have two levels of TLB. Here, ITLBs and DTLBs are over-dimensioned to compensate the absent second level.
[7] The A7 is partial dual-issue, while the A15 has a peak issue width of eight instructions. Here, in both cores we do not simulate the branch unit.

ration for the Cortex-A9 [8]. For the A7, we subtract one cycle (two for multiply-accumulate), because it has a tightly integrated VFP/NEON to the ARM pipeline, compared to the rectangular design of VFP/NEON in the A9. For the A15 we multiply the A9 latencies by $10/4 = 2.5$, which is the ratio of VFP pipeline depths. Table 2 shows the latencies of main instructions.



(a) Cortex-A7 (b) Cortex-A15

Figure 1: gem5 execution stage configuration.

Based on published ARM diagrams [10], we configured the execution stages as Figure 1 shows.

McPAT can simulate the three transistor types described by ITRS, including the Low operating power (LOP) configuration used in high-performance embedded processors. We set this type of transistor for cores and caches. The technology node is 28 ns and the temperature is fixed at 27 °C. The area and energy dissipation of functional units are empirically modeled in McPAT (of a Cortex-A9 in the embedded mode). We developed a patch to allow the user to specify energy and area ratios for each FU. To set the area of FUs in both cores, we used the rule in Eq. 1 taken from Intel area scaling studies of out-of-order cores [16]

$$FU_{area} \propto issue\_width^2 \quad (1)$$

In this experiment, we considered the front-end width instead of the issue (back-end) width, because that rule considers the sustainable issue width. We considered that the energy cost per access also follows the Eq. 1. The cost per cycle is then roughly the cost per access divided by the average latencies of instructions. Table 3 summarizes the normalized FU areas and energy costs of the reference and studied cores.

For energy estimations we did not consider the snooping unit, because we compared the clusters with only one core activated.

4.2 Benchmarks

Here, we describe the benchmarks used to compare the energy and performance trade-offs of Cortex-A7 and A15 CPUs.

4.2.1 Dhrystone

Dhrystone is a very simple benchmark. However, processor manufacturers still employ it to compare relative energy and performance. In addition, results of big.LITTLE CPUs running this benchmark were published [10]. Nonetheless, because the detailed environment was not described, we used Dhrystone 2.1 and assumed similar CPU configurations. The benchmark is compiled with Code Sourcery gcc 4.7.2 with the flags `-static -mthumb -O3 -mcpu=cortex-a15`.

4.2.2 PARSEC 3.0

To better evaluate the energy and performance trade-offs of the CPUs, we ran 10 of the 13 benchmarks of PARSEC

Table 2: Configuration of the functional units (FUs) for integer and VFP instructions

gem5 FU	gem5 opClass	Example of instructions	Cortex-A15 (out-of-order)		Cortex-A7 (in-order)	
			Latency	Pipelined	Latency	Pipelined
Simple ALU	IntAlu	MOV, ADD, SUB, AND, ORR	1	Yes	1	Yes
Complex ALU	IntMult	MUL, MLA	4	Yes	4	Yes
	IntDiv	UDIV, SDIV	12	No	12	No
FP/SIMD Unit[1]	SimdFloatAdd	VADD, VSUB	10	Yes	3	Yes
	SimdFloatMult	VMUL, VNMUL	12	Yes	4	Yes
	SimdFloatMultAcc	VMLA, VMLS, VNMLA, VNMLS	20	Yes	6	Yes
Load/Store Unit[2]	MemRead	LDR, VLDR	3	Yes	1	Yes
	MemWrite	STR, VSTR	2	Yes	1	Yes

[1] In gem5, both VFP and Advanced SIMD instructions are regrouped under the SimdFloat* operation classes.
[2] The A15 can issue one load and one store per cycle. In gem5, this is simulated by separated Load and Store units. Here, we regrouped them for simplicity.

Table 3: Normalized FU area and energy costs

Core	Width	Area		Energy per access (EPA)		Inst. lat.		Energy per cycle	
		INT	FP/SIMD	INT	FP/SIMD	INT	FP/SIMD	INT	FP/SIMD
Generic	W	$\propto W^2$		$\propto W^2$		L		EPA/L	
A7	1	0.25	0.25	0.25	∼0.25	1	∼1	0.25	∼0.25
A9	2	1	1	1	1	1	1	1	1
A15	3	2.25	2.25	2.25	∼2.5	1	∼2.5	2.25	1

3.0 [4], a modern suite which covers several application domains. The compilation environment is described in our previous work [8]. We also used the Simsmall input set.

5. EXPERIMENTAL RESULTS

First, we validate the area of cores and clusters estimated with McPAT 1.0. Then, energy and performance trade-offs of Cortex-A7 and A15 CPUs are presented.

5.1 McPAT area validation

Table 4 shows the core, L2 and cluster area estimation compared to published data. For core area, our estimations showed an error of only 3.6 % for the A15 and exactly matched the area of the A7. In the cluster estimations, we did not model the snooping unit, which can explain the underestimations of -13 and -1.4 % for the A7 and A15 clusters respectively. Based on the McPAT example configuration for the A9, the snooping unit may represent less than 1 % of the A15 cluster and around 4 % in the A7.

The floorplan of the Cortex-A7 at 28 ns was published [1]. Based on that information, we evaluated the area estimated by our McPAT configuration. Table 5 presents the relative areas of five main structures in the core. The greatest difference is in the TLB, where McPAT underestimates its area by 60 %. This can be explained by the lack of second level of TLB in our model. However, if we weight the errors by the relative core area of each structure, the Load/store unit has the greatest error of only 5.3 %.

Figure 2 shows the relative area of core and structures of both modeled cores. The A7 is seven times smaller than the A15. The structures of the big core are not as balanced as those in the small one: the execution stage in the A15

Table 4: Area validation (mm^2 at 28 nm) of core and cluster estimated with McPAT 1.0

Cluster	Comp.	McPAT	Publ.	Error (%)	Ref.
A7	Core	0.45	0.45	0.0	[1]
	L2	1.52	-	-	-
	Total[1]	3.32	3.8	-13	[7]
A15	Core	3.21	3.1	3.6	[20]
	L2	5.88	-	-	-
	Total[1]	18.7	19	-1.4	[7]

[1] In our McPAT model, it's the area of four cores + L2, the snooping unit was not accounted.

occupies 82 % of the core, while in the A7 it represents only 35 %. Unfortunately, the A15 floorplan was not published to validate the results. Nonetheless, if we look at the Cortex-A9 floorplan [2], its execution stage represents around 60 % of the core area. If we consider that the A15 embeds two FP/ NEON units instead of one, our estimations are reasonable for the A15.

5.2 Simulating big.LITTLE energy/performance trade-offs

As a relative energy and performance validation of our simulation framework, we present here trade-offs between Cortex-A7 and A15 CPUs. To highlight the energy and performance differences of those cores, we simulated only one active core in each cluster, running single-threaded benchmarks. The energy comparisons take into account the active core and the L2 cache.

Table 6 shows the relative energy and performance of A7

Table 5: Validation of relative area estimations compared to published data for the Cortex-A7 [1]

Component(s)		Area (%)		Error (%)	Weighted error (%)
McPAT	ARM A7 floorplan	McPAT	ARM A7		
Instruction fetch unit	PFU, I-cache, ICU	24.6	25.6	-3.63	-0.93
Execution stage	DPU	34.7	38.2	-9.40	-3.60
Memory management unit	TLB	1.77	4.45	-60.2	-2.68
Load/store unit	STB, D-cache, DCU	29.7	24.4	21.8	5.31
Other	BIU	9.25	7.36	25.7	1.89

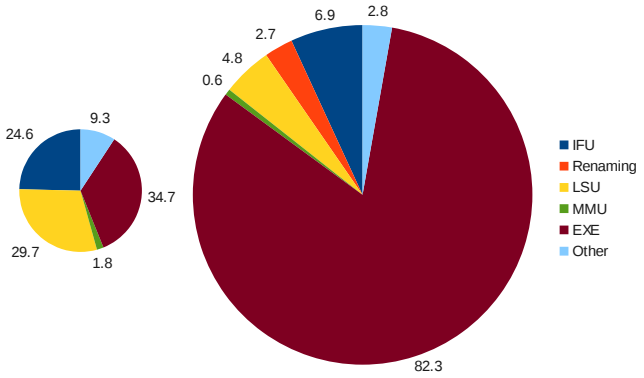


Figure 2: Relative core and structure areas of the modeled Cortex-A7 (left) and Cortex-A15 (right).

Table 6: Relative energy and performance of Cortex-A7 and A15 CPUs (one core ative)

Benchmark	A15 speedup	A7 energy eff.
Dhrystone	1.84	3.69
Blacksholes	1.14	4.42
Bodytrack	1.45	4.09
Dedup	1.46	4.29
Ferret	1.11	5.47
Fluidanimate	1.15	5.29
Freqmine	1.45	4.32
Streamcluster	1.32	3.88
Swaptions	1.34	4.12
Vips	1.89	3.56
x264	2.46	3.08
Geometric mean	1.47	4.15

and A15 clusters. In the Dhrystone benchmark, the A15 provided a speedup of 1.84, while the A7 consumes 3.69 times less energy. These results are very close to those published by ARM [10]: 1.9 and 3.5, respectively. In the PARSEC benchmarks, we observed varying degrees of trade-offs. Ferret has a speedup of only 1.11 in the A15, with an energy efficiency of 5.47 in the A7. x264 showed the greatest speedup of 2.46 in the A15, exchanged by an energy efficiency of 3.08 in the A7. In average, the A15 is 1.5 times faster, but the A7 consumes 4.1 times less energy.

It's believed that A15 cores provides speedups of 2-3x and that A7 cores are 3-4x more energy efficient, while our results showed that the A15 provide only an average speedup of 1.5. One explanation is that our benchmarks are compiled with high optimization levels, compared to moderately optimized binaries and libraries or legacy-applications found in real-life, which out-of-order pipelines can accelerate. Another explanation comes from the nature of the PARSEC suite, which focus on thread-level parallelism instead of instruction-level parallelism (ILP), what superscalar pipelines are designed for. Indeed, x264 and Vips that showed the greatest speedups have higher ILP compared to the others.

Figures 3 and 4 show the average energy consumption of PARSEC benchmarks per structure in the Cortex-A7 and A15 respectively. In both CPUs, the L2 cache is the main energy consumer, mostly contributing with leakage. In the in-order core, L1 caches and the FP/SIMD unit are large structures frequently accessed, and hence contribute both with dynamic and leakage energy, while small components with high activity comprise the decoder, load/store queue, register files and bypass buses, which contribute mainly with dynamic energy. In the out-of-order core, the execution stage consumes most of the energy, with the most energy-hungry component in the core being the FP/SIMD units, followed by the instruction scheduler, integer register renaming, registers files, instruction buffer and decoder, which are also examples of small structures highly accessed.

6. CONCLUSIONS

In this paper, we detailed the simulation of asymmetric embedded cores with gem5 and McPAT. Our previous work validated the timing accuracy of our simulation framework of Cortex-A cores. In this work, we extended our simulator to estimate area and energy consumption with McPAT. We validated the area estimations of Cortex-A7 and A15 cores and clusters. We also showed their energy and performance trade-offs running 11 benchmarks.

7. ACKNOWLEDGMENTS

The authors would like to thank Thierno Barry's review.

8. REFERENCES

[1] ARM website. Cortex-A7 processor.
[2] ARM website. Cortex-A9 Processor.
[3] R. I. Bahar and S. Manne. Power and energy reduction via pipeline balancing. ISCA '01, pages 218–229, New York, NY, USA, 2001. ACM.
[4] C. Bienia. *Benchmarking Modern Multiprocessors*. PhD thesis, Princeton University, January 2011.
[5] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell,

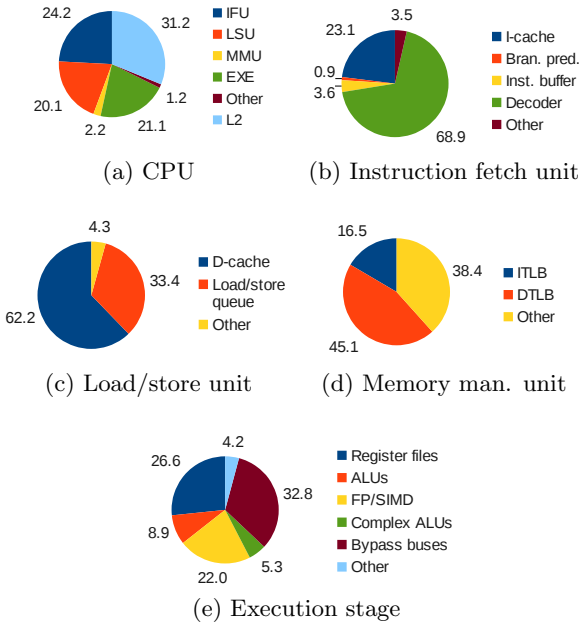


Figure 3: Average energy consumption of PARSEC benchmarks per structure in the Cortex-A7.

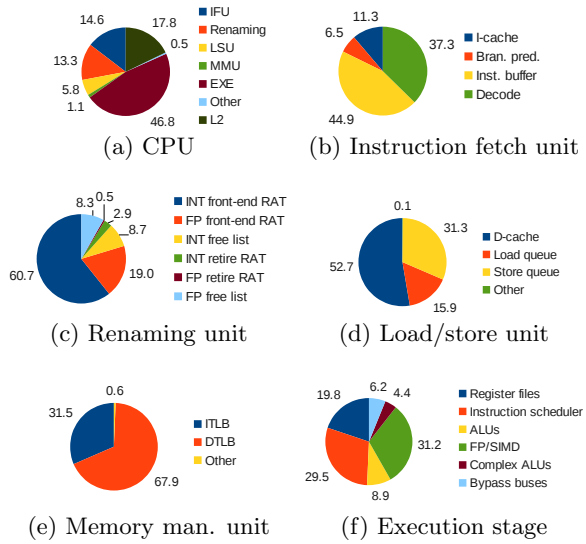


Figure 4: Average energy consumption of PARSEC benchmarks per structure in the Cortex-A15.

M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood. The gem5 simulator. *SIGARCH Comput. Archit. News*, 39(2):1–7, Aug. 2011.

[6] D. Burger and T. M. Austin. The SimpleScalar tool set, version 2.0. *SIGARCH Comput. Archit. News*, 25(3):13–25, June 1997.

[7] EETimes. Slideshow: Samsung cagey on smartphone SoC at ISSCC. `http://www.eetimes.com/document.asp?doc_id=1263082&page_number=2` [accessed 5 Nov. 2014].

[8] F. A. Endo, D. Couroussé, and H.-P. Charles. Micro-architectural simulation of in-order and out-of-order ARM microprocessors with gem5. SAMOS XIV, 2014.

[9] N. Fournel. *Estimation et optimisation de performances temporelles et énergétiques pour la conception de logiciels embarqués*. PhD thesis, École Normale Supérieure de Lyon, 2007.

[10] P. Greenhalgh. Big.LITTLE Processing with ARM Cortex-A15 & Cortex-A7. *ARM White paper*, 2011.

[11] M.-y. Hsieh. A scalable simulation framework for evaluating thermal management techniques and the lifetime reliability of multithreaded multicore systems. IGCC '11, 2011.

[12] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen. Single-ISA heterogeneous multi-core architectures: The potential for processor power reduction. MICRO 36, 2003.

[13] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. The McPAT framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing. *ACM Trans. Archit. Code Optim.*, 10(1):5:1–5:29, Apr. 2013.

[14] A. Lukefahr, S. Padmanabha, R. Das, F. M. Sleiman, R. Dreslinski, T. F. Wenisch, and S. Mahlke. Composite cores: Pushing heterogeneity into a core. MICRO '12, 2012.

[15] S. K. Rethinagiri, O. Palomar, R. Ben Atitallah, S. Niar, O. Unsal, and A. C. Kestelman. System-level power estimation tool for embedded processor based platforms. RAPIDO '14, 2014.

[16] E. Shifer and S. Weiss. Low-latency adaptive mode transitions and hierarchical power management in asymmetric clustered cores. *ACM Trans. Archit. Code Optim.*, 10(3):10:1–10:25, Sept. 2008.

[17] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. ISCA '03, 2003.

[18] R. Strong. m5-mcpat-parser. `https://bitbucket.org/rickshin/m5-mcpat-parser` [accessed 5 Nov. 2014].

[19] M. A. Suleman, O. Mutlu, M. K. Qureshi, and Y. N. Patt. Accelerating critical section execution with asymmetric multi-core architectures. *SIGARCH Comput. Archit. News*, 37(1):253–264, Mar. 2009.

[20] The Tech Report. AMD's A4-5000 'Kabini' APU reviewed. `http://techreport.com/review/24856/amd-a4-5000-kabini-apu-reviewed/11` [accessed 5 Nov. 2014].

[21] University of Michigan and University of Colorado. DARPA Technical Report v2.7.fm. Technical report. `http://web.eecs.umich.edu/~panalyzer/pdfs/Sim-Panalyzer2.0_ReferenceManual.pdf` [accessed 7 Feb. 2013].