Exploring Energy Efficiency Model Generalization on Multicore Embedded Platforms

Rexha, Hergys; Lafond, Sebastien

Published in:
26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)

DOI:
10.1109/PDP2018.2018.00084

Published: 01/01/2018

Exploring Energy Efficiency Model Generalization on Multicore Embedded Platforms

*Hergys Rexha, Sébastien Lafond†
Faculty of Science and Engineering, Åbo Akademi University, Turku, Finland
Email: *hergys.rexha@abo.fi, †sebastien.lafond@abo.fi

***Abstract*—In this paper we investigate the relation between energy efficiency model and workload type executed in modern embedded architectures. From the energy efficiency model obtained in our previous work we select a few configuration points to verify that the prediction in terms of relative energy efficiency is maintained through different workload scenarios. A configuration point is defined as a set of platform tunable metrics, such as DVFS point, DPM level and utilization rate. As workloads, we use a combination of synthetic generators and real world applications from the embedded domain. In our experiments we use two different architectures for testing the model generality, which provide examples of real systems. First we have a comparison of the efficiency obtained by the two architecturally different chips (ARM and INTEL) in different configuration points and different workload scenarios. Second we try to explain the different results through the thermal management done by the two different chips. At the end we show that only in the case of workloads highly composed by integer instructions the results from the two architectures converge and show the need for a specific model trained with integer operations.**

I. Introduction

Energy consumption is a key issue in today's electronic systems, ranging from IoT nodes to server farms. Information technology has acquired a big part of our everyday life, requiring a large amount of energy. The number of computing systems we use today has never been so high in the past, and is rapidly increasing [1]. The current energy production figures show that in Europe, only a small part of the energy required is produced from renewable sources, utilizing mostly fossil fuels, with a related strong impact in the environment in terms of GHG (green house gas) emissions. Recently the world meteorological organization in its buletin said that the levels of CO_2 in 2016 had an unprecedented increment [2], giving an alarm about the levels of greenhouse gases and the resulting climate changes. As Europe has always been a leader in policies for environmental protection and carbon reduction initiatives, in the future guidelines, there is a high pressure to increase the energy efficiency of electronic devices [3]. No matter if we consider a data center or mobile devices, the imperative is still the same: a decrease in energy consumption is nedded. Depending from the computing domain, approaches have been long proposed for achieving reduction in the energy consumption. One of the largest markets in electronic devices which has had the fastest rise in the past years, is the mobile systems domain. Exposed to a huge number of use cases, mobile devices face different requirements which often trade-off with low energy consumption. The most obvious requirement is performance, which affects directly the power dissipation of mobile systems. Today's processor chips are reaching levels of performance which are able to cope with the most performance hungry applications. In the future, different applications like virtual reality, artificial intelligence and machine learning will increase the level of performance required from processor chips. The problem in this scenario is that we need also to be efficient. Recent approaches from the industry try to achieve better energy efficiency through the use of heterogeneous systems, which enclose different computing elements inside a single chip [4]. Recently, industry proposes an increased level of heterogeneity present on a multiprocessor system on chip (MPSoC) with approaches like [5] and [6] where we go beyond the idea of a two cluster heterogeneity, by adding another cluster of cores considered as middle level performance, obtaining a tri-cluster heterogeneity. In this way authors promise to cut power dissipation by 50%[1]. Also, from ARM we have the latest technology named DynamIQ [2] providing many options for organizing high performance and energy efficient cores inside a cluster. In this paper we follow the work done in [7], which builds an energy efficiency model based on platform configuration points. Platform configuration points are combinations of available actuators present in today's heterogeneous architectures. In the next section, we will present the ideas behind this work and the questions raised which we try to answer in this paper.

A. Why we did this work?

In our previous work we experimentally build an energy efficiency model for two widely used ARM core types which compose the ARM big.LITTLE architecture (Cortex A-15 and Cortex A-7). The energy efficiency model is based on synthetic workload composed mostly from floating point instructions. In the following steps we would like to validate the generality of the model with different instruction mixes and verify that the relative efficiencies of the model points are still valid. The main research questions that we try to address in this work are the followings:

1) Can we have a general energy efficiency model, without looking at the type of load/instructions?
2) Are the relative energy efficiency values for different configuration points kept for different instruction mixes?

[1]http://www.mediatek.com/products/smartphones/mediatek-helio-x30
[2]http://developer.arm.com/technologies/dynamiq

TABLE I
PLATFORM CONFIGURATION POINTS USED IN THE EXPERIMENTS WITH ARM ARCHITECTURE

Name	Configuration	Energy Efficiency (op/J)
C1	4A15/1.1GHz/100% + 4A7/0.6GHz/100%	900442
C2	4A15/2GHz/100% + 4A7/1.4GHz/100%	572154
C3	4A15/1.1GHz/60% + 4A7/0.6GHz/60%	1010780
C4	4A15/2GHz/60% + 4A7/1.4GHz/60%	657773

In our definition of platform configuration point we use configurable actuators available in today's platforms. In heterogeneous platforms they are defined by:

- The number and type of cores to utilize for computations
- The frequency each type of core can have
- The utilization rate to be used by each type of core

The combination of the previous parameters defines a platform configuration point. From the energy efficiency model in [7] we derive a lookup table composed of all the configuration points available on a given platform and the related performance and energy efficiency values they provide.

Furthermore, in our experiments we use another architecture to compare energy efficiency results achieved by platform configuration points in different workload types. We choose two sets of configuration points to be used during the experiments. In the ARM architecture, the selected platform configuration points are described in Table I. As a second architecture we choose Intel Atom, and define the configuration points presented in Table II. For both architectures, in the four configuration points chosen we use the maximum executing parallelism available and a mix of choices when applying or not, utilization control and DVFS. We enforce utilization control on a particular thread by means of a real time type scheduler, which is named *sched_deadline*. For more information on the methodology used and the details of the configurations, refer to [7]. With utilization control, which is expressed in percentage, we select the load level the core will reach while running the computations. Beside the description of each configuration point, the resulting energy efficiency values are reported in the tables. These values originate from the energy efficiency model in [7]. Our investigations for the previous questions lead us to the forthcoming questioning:

3) Do ARM and Intel architectures provide the same energy efficiency?
4) Do the thermal characteristics play a role in the efficiency of the configurations, especially when controlling the load?

We will try to answer these questions through a wide set of experiments as they will be presented in section III.

II. RELATED WORK

There are not many works who analyse the relation between the composition of the workload being executed and the different configuration choices to execute it in a highly energy efficient way. The closest to this topic is the work in [8] where the authors show that by taking into account the mix of instructions from the workload, an energy efficient mapping could be done on a heterogeneous systems. By knowing which core type is best for a certain workload the scheduling decisions could be taken in such a way to achieve high levels of energy efficiency. The authors promise to save energy in the interval 7.1% to 31.3% if a workload-aware scheduler will be used. In contrast with their work we want to validate the results obtained before with our energy model, in the context of different workload type. In our second test case we use real world applications which represent embedded applications.

TABLE II
PLATFORM CONFIGURATION POINTS USED IN THE EXPERIMENTS WITH INTEL ARCHITECTURE

Name	Configuration	Energy Efficiency (op/J)
C1	4 Intel Atom/1.1GHz/100%	195890
C2	4 Intel Atom/1.92GHz/100%	104742
C3	4 Intel Atom/1.1GHz/60%	205216
C4	4 Intel Atom/1.92GHz/60%	99615

III. EXPERIMENTAL SETTINGS

In this section we will present the hardware and software tools that we used during our experiments and also the used measurement framework.

A. Hardware platforms

We use two hardware platforms, ODROID XU4 development board from Hardkernel [3] and UpBoard[4].

B. Measurement framework, interval definition with the oscilloscope

The ODROID board is installed with Linux Ubuntu 14.04, kernel 4.2.0, GCC 4.9 while the UpBoard is installed with Ubilinux 3.0, kernel 4.4.0 GCC 4.9. While performing the experiments on both boards the minimal services of Linux system are running. For power measurements we use an external power supply with a current/power IC monitor [9]. To investigate on the effects of chip temperature on the power dissipation we use a two channel PC oscilloscope, PicoScope 2205 [10]. The experimental framework involves the two development boards on which we run synthetic and real world applications, power measuring device and a refrigerated environment for testing temperature effects on power dissipation.

C. Presentation of the benchmarks: type of loads, synthetic vs. real

We used two categories of workloads for our experiments. The first category is composed of synthetic instruction mix, which are obtained from a synthetic workload generator called epEBench [11]. The second workload category is composed of real world applications related to the embedded system domain. These applications are selected from CoreMarkPro[5] benchmark suite. It offers real-world examples of applications with different instruction mix. The workloads in CoreMarkPro

[3]http://wiki.odroid.com/odroid-xu4/odroid-xu4
[4]http://forum.up-community.org/categories/up-board/
[5]http://www.eembc.org/coremark/index.php?b=pro.htm

are divided in two main categories: floating-point and integer. In our experiments we choose 4 workloads, 2 from each type. For more information refer to [12]. The workloads used are:

- Linear Algebra workload which is a mathematical solver of equations through the Gaussian elimination method.
- FFT Radix 2 workload performs transformations with Radix2 on the input.
- XML parser workload parses an XML string and creates an ezxml structure with subsequent final validation of the results.
- Secure Hash Algorithm (SHA256) workload is composed from a subset of cryptographic hash functions with digests of 256 bits.

D. Temperature experiments

We have conducted experiments to measure the relation between power dissipation and core temperature during workloads execution. Because of the large power dissipation during the execution of some of the workloads, especially in the Exynos 5422 chip, all the experiments concerning CoreMarkPro benchmark were run in a refrigerated environment with a controlled temperature of -18°C. Also the development boards were equipped with a fan powered with an external supply. With this infrastructure we were able to perform experiments without reaching the critical temperature of the core, where if so happens, the system will be turned off in order to prevent physical damages of the silicon.

IV. Results

In this section we present some of the results which shed some light over the questions presented in the first sections. We first explore the generality of the energy efficiency model with regard to the executed workload. Then, through the synthetic benchmark, we value the impact of utilization control on the energy efficiency of different instruction types. At the end, a comparison of the two architectures regarding energy efficiency and a possible explanation for their difference will be discussed.

A. (Answer to question 1): The impact of workload type on the relative energy efficiency of the configuration points

We run the selected workloads from the CoreMarkPro benchmark at the highest degree of parallelism offered from the chosen platforms. The workloads are executed with 1000 iterations and the throughput is collected in terms of workloads per second as a performance metric. The average power dissipation is measured during execution time. In Figure 1 we show the energy efficiency results from the Exynos chip with the real world workloads for the selected platform configuration points. In general for the first three workloads there is not a significative difference in the efficiency provided by C1 and C3 configuration points. We remind that the only difference between these configuration points is the utilization control which is enforced in C3 at 60%, while in C1 the core reaches 100%. Both of these configurations set the cores at their middle level frequency (1GHz for A15 and 600MHz for A7) choosing a relaxed execution strategy or otherwise called the pace-to-idle approach where the execution is set at the lowest possible speed while still keeping the performance requirements. The difference although in energy efficiency remains high with C2 and C4 which enforce the race-to-idle strategy, or running at the fastest speed. Demonstrating that going at the fastest speed is not energy efficient. Remarkably during the cryptographic function execution, we have different results of relative energy efficiency for the platform configuration points. Here, C1 and C2 show better energy efficiency than C3 and C4. This behaviour is against the model predictions.

The results for Intel are presented in Figure 2. We observe the same behaviour as in the case of ARM, but with better numbers in the provided energy efficiency . Still in the Secure Hash application the best energy efficiency of the group is achieved by the configuration C1. Apparently in the case of a strong presence of integer instructions in the executed workload, the actual model is not able to select the highest energy efficiency configuration point.

B. (Answer to question 2):Impact of load control on efficiency

We want to answer the question of whether the utilization control has an impact on efficiency, and evaluate if this impact is more important in some type of workloads rather than others. We use epEBench multi-core energy benchmark for generating workloads based on the function models presented in [11]. The workloads are executed through the set of previously described configuration points (C1-C4). The performance data are collected in terms of instructions per second and the power data are logged through the external meter. The results are presented in Figures 3, and 4. According to the results for the ARM platform, C1 and C3 provide almost the same level of energy efficiency through all the model types used. The same happens between C2 and C4. In the case of workloads composed by integer additions and multiplications we cannot notice the behaviour observed from SHA256 workload in Figure 1.
For the Intel platform (Figure 4), configurations C1 and C3 still provide a better energy efficiency compared to C2 and C4 which are together at the same level. While again as in ARM this observance is repeated through all the model used. Apparently for both architectures there is no significant difference in the energy efficiency levels achieved by C1 and C3 for different instruction types.

C. (Answer to question 3):ARM vs. Intel, which architecture provides better energy efficiency?

This question is highly debated in the academic and industrial community circles. In our experiments we tackle both sides of the problem with a double scenario. On one side we use real applications from the embedded domain to measure the energy efficiency levels achieved in both architectures, on the other side we test again the architectures by using an energy benchmark which enables us to define the workload model to execute. In the experiments conducted with the real world applications, Intel succeeded to be more energy

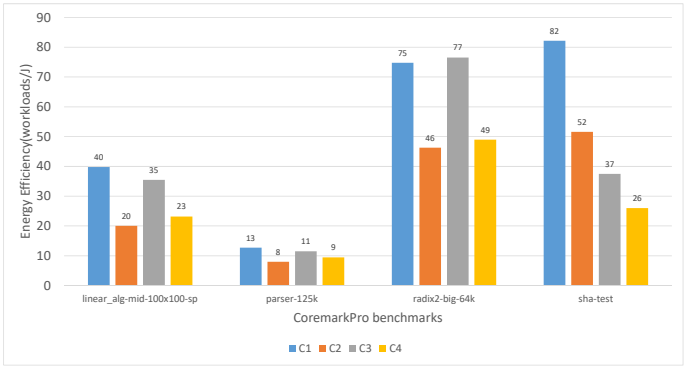

Fig. 1. Energy efficiency for the Exynos 5422 SoC workloads from CoreMarkPro

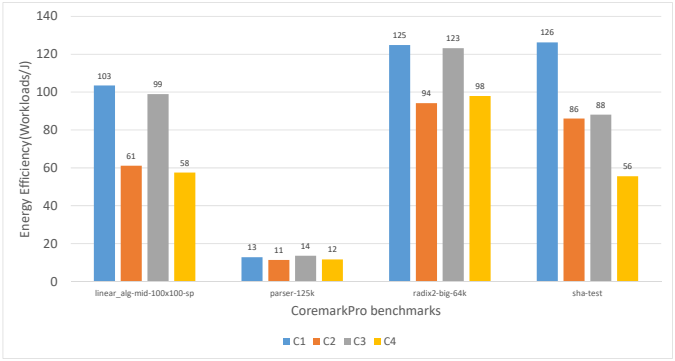

Fig. 2. Energy efficiency for the Atom x5 8350 SoC with workloads from CoreMarkPro

efficient than ARM, in all the workloads. Even in the XML parser which shows the lowest score in terms of energy efficiency. A possible explanation for a low score is the poor ILP present in the application, versus the highest ILP of the workload set found in the SHA256 application which shows the highest result in energy efficiency. A possible explanation for the difference between ARM and INTEL can be found by the presence of more static power dissipated in the Exynos chip. In our experiments we measure the relation between chip temperature and power dissipation during the execution Linear Algebra workload. While executing, the gradient of the temperature curve in the Exynos chip is higher than in Atom. This proves that the generated heat is not dissipated rapidly, causing a fast increase in the core temperature. This observation guides us to the conclusion that more static power is present in the power data, so even though ARM shows better performance (more cores present in chip), still energy efficiency scores are in favour of INTEL Atom. As a comparison we have conducted the same experiments for the INTEL architecture and measured Atom z-8350 temperature and power with the same workload. From the results we have noticed that the temperature of the chip reaches a steady state from the beginning of the experiment and remains constant through the execution, leading to less static power dissipation.

A total different picture is obtained in the second case, with different function models used in the epEBench synthetic benchmark. ARM architecture provides better energy efficiency in almost all the models used, except for branches, which show better results in Intel. This could be related to the better branch predictor present in the Atom processor. This intuition is confirmed with epEBench benchmark where we stress both chips with 200G branch instructions and Atom has 15% less branch misses than the Exynos chip.

Furthermore in Intel, there is not much difference in efficiency between the single point and double point operations, while in ARM single precision operations are more energy efficient than double precision. We believe this is due to the presence of heterogeneity in ARM architecture, with the A7 cores providing increased levels of efficiency. Also, the energy efficiency achieved in the workloads composed by SIMD instruction is higher in ARM than Intel (we don't show the results since they are very low), understandably if we consider the presence of NEON execution engine in the Exynos chip.

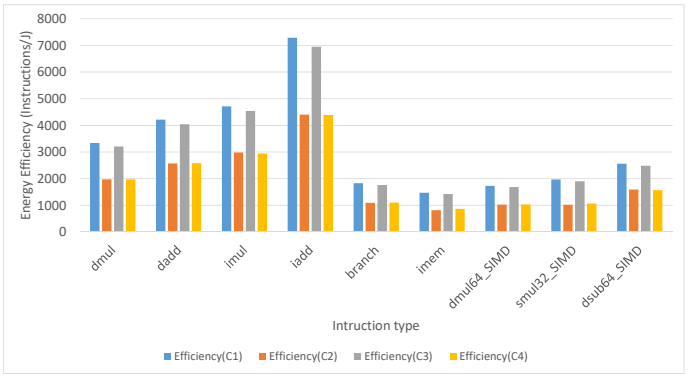

Fig. 3. Energy efficiency for the Exynos 5422 SoC with function models from epEBench

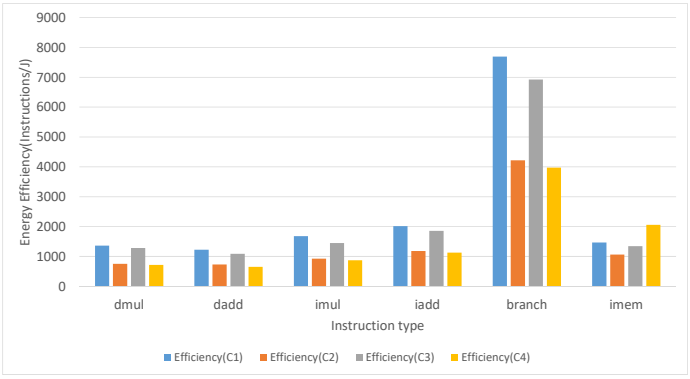

Fig. 4. Energy efficiency for the Atom x5 8350 SoC with function models from epEBench

D. (Answer to question 4): Thermal characteristic influence on the energy efficiency results

From the temperature experiments conducted in the previous subsection we could notice the presence of an increased power dissipation in the ARM chip compared to Intel, leading to a difference in energy efficiency scores. The previous experiments were conducted in a highly refrigerated environment, as a result a controlled rise in chip temperature. In a second experiment we remove the boards from the controlled environment and try to execute again the Linear workload with 1000

iterations and with only the active fan as a cooling system. We monitor the current consumed during the execution with an oscilloscope. During the multi iteration workload execution we notice a gradual increase in the current consumption, starting from the first iteration of the workload. After the first iteration the subsequent increase in power dissipation is due to static power. The same experiment is conducted in Intel Atom where the current consumption remains stable in the execution window which shows better management of the thermal effects with subsequent control of the static power dissipated. On the other side, the execution of synthetic loads produces different levels of static power dissipated. We observe that only special type of instructions, like memory instructions, while executed inside a loop produce more static power than others.

V. Conclusions

In this paper we answer questions regarding the generality of the previous obtained energy efficiency model [7], in the context of the embedded systems domain. We select two hardware platforms for running experiments, which are based on two popular embedded architectures such as ARM and INTEL. In the experiments with real applications from the CoreMarkPro suite, ARM and INTEL show results approximately in accordance with the model values for workloads with evenly distributed instruction mixes. In case of workloads with high levels of integer instructions, we need another model to find high energy efficiency configuration points.

When we try to investigate the effect of utilization control on the energy efficiency with different instruction types, we use the multicore energy benchmark epEBench, which executes custom function models. We use 9 type of function models which test most common instruction types, with results in ARM and INTEL that show no significant difference in the energy efficiency provided by utilization control on any special type of workload. Comparing the absolute results of energy efficiency between the two architectures produces a two faced picture. When using real applications as workloads Intel shows better efficiency levels than ARM in all the applications tested. We get a total different result when using synthetic workloads, where ARM shows better absolute values of energy efficiency in almost all the instruction types used. The performance provided by 8 cores in ARM is larger than the additional increase in dissipated power. Exception is made for branches and memory operations. With branches, Intel shows far better efficiency than ARM. The performance with this type of load in the Atom chip is better, and less power is dissipated in Intel and we believe this is due to a more efficient branch predictor in the Atom core. This is confirmed by performance counters which show less branch misprediction. With purely memory load, we measure the same results in energy efficiency, even though in ARM the performance is much higher in terms of operations per second. The results are explained with a more stable level of power dissipated in Intel compared to ARM with this type of load. Another observation that can be made by results obtained from these experiments, is the limited efficacy of synthetic benchmarks when assessing properly energy efficiency comparison. This can be explained by the fact that static power effects are not strongly present while executing synthetic benchmark with only certain type of execution units stressed, as shown in section IVD, which explains the different energy efficiency results between synthetic and real workloads.

As a possible explanation of energy efficiency values is the difference of the two architectures in managing thermal effects. As we show by the experiments done with the oscilloscope INTEL is better in managing thermal effects than ARM, which results in less static power present in Intel compared to ARM.

Acknowledgment

This research was supported by a grant from the Erasmus Mundus EUROWEB+ Scholarship Programme.

References

[1] E. Gelenbe and Y. Caseau, "The Impact of Information Technology on Energy Consumption and Carbon Emissions," *Ubiquity*, vol. 2015, no. June, pp. 1:1–1:15, Jun. 2015. [Online]. Available: http://doi.acm.org/10.1145/2755977

[2] "World meteorological organization: Co2 levels surged to record high in 2016," https://www.cbsnews.com/news.

[3] "Europe's Energy Transition - 1st Edition." [Online]. Available: https://www.elsevier.com/books/europe-s-energy-transition/manuel-welsch/978-0-12-809806-6#utm_source=SciTech%20Connect&utm_medium=Energy%20New%20Releases&utm_campaign=STC317

[4] A. Holdings. (2013) big.little technology: The future of mobile making very high performance available in a mobile envelope without sacrificing energy efficiency. [Online]. Available: https://www.arm.com/files/pdf/big_LITTLE_Technology_the_Futue_of_Mobile.pdf

[5] H. T. Mair, G. Gammie, A. Wang, R. Lagerquist, C. J. Chung, S. Gururajarao, P. Kao, A. Rajagopalan, A. Saha, A. Jain, E. Wang, S. Ouyang, H. Wen, A. Thippana, H. Chen, S. Rahman, M. Chau, A. Varma, B. Flachs, M. Peng, A. Tsai, V. Lin, U. Fu, W. Kuo, L. K. Yong, C. Peng, L. Shieh, J. Wu, and U. Ko, "4.3 A 20nm 2.5ghz ultra-low-power tri-cluster CPU subsystem with adaptive power allocation for optimal mobile SoC performance," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan. 2016, pp. 76–77.

[6] H. Mair, E. Wang, A. Wang, P. Kao, Y. Tsai, S. Gururajarao, R. Lagerquist, J. Son, G. Gammie, G. Lin, A. Thippana, K. Li, M. Rahman, W. Kuo, D. Yen, Y. C. Zhuang, U. Fu, H. W. Wang, M. Peng, C. Y. Wu, T. Dosluoglu, A. Gelman, D. Dia, G. Gurumurthy, T. Hsieh, W. Lin, R. Tzeng, J. Wu, C. Wang, and U. Ko, "3.4 A 10nm FinFET 2.8ghz tri-gear deca-core CPU complex with optimized power-delivery network for mobile SoC performance," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb. 2017, pp. 56–57.

[7] H. Rexha, S. Holmbacka, and S. Lafond, "Core level utilization for achieving energy efficiency in heterogeneous systems," in *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, March 2017, pp. 401–407.

[8] S. Holmbacka and J. Keller, "Workload Type-Aware Scheduling on big.LITTLE Platforms," in *Algorithms and Architectures for Parallel Processing*, ser. Lecture Notes in Computer Science. Springer, Cham, Aug. 2017, pp. 3–17. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-65482-9_1

[9] HARDKERNEL. (2016) smartpower2 power supply. [Online]. Available: http://wiki.odroid.com/accessory/power_supply_battery/smartpower2

[10] PicoTech. Picoscope 2205. [Online]. Available: http://logicanalysers.com/testequipment/picotech/pc_oscilloscope/picoscope_2203_2204_2205_user_guide.pdf

[11] S. Holmbacka and R. Mller, "epebench: True energy benchmark," in *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, March 2017, pp. 426–429.

[12] EEMBC. Introduction to coremark-pro. [Online]. Available: http://www.eembc.org/coremark/CoreMark-Pro_intro.pdf