Predictive Modeling based Power Estimation for Embedded Multicore Systems

Sriram Sankaran
Center for Cybersecurity Systems and Networks
Amrita University
Amritapuri, Kollam 690525
srirams@am.amrita.edu

ABSTRACT

The increasing number of cores in embedded devices results in improved performance compared to single-core systems. Further, the unique characteristics of these systems provide numerous opportunities for power management which require models for power estimation. In this work, a statistical approach that models the impact of the individual cores and memory hierarchy on overall power consumed by Chip Multiprocessors is developed using Performance Counters. In particular, we construct a per-core based power model using SPLASH2 benchmarks by leveraging concurrency for multicore systems. Our model is simple and technology independent and as a result executes faster incurring lesser overhead. Evaluation of the model shows a strong correlation between core-level activity and power consumption and that the model predicts power consumption for newer observations with minimal errors. In addition, we discuss a few applications where the model can be utilized towards estimating power consumption.

Keywords

Embedded Systems; Multicore; Energy Modeling; Linear Regression

1. INTRODUCTION

Advancements in computing systems have fueled the growth of mobile embedded devices with increasing amount of functionality causing significant battery drain. While battery technology has not kept up with the increasing demands in mobile devices, the need for developing energy-aware systems has gained prominence. Dynamic Frequency Scaling and Dynamic Power Management are two of the most popular mechanisms used to trade-off performance for conserving power in mobile embedded devices.

Embedded Multicore systems such as Chip Multi-Processors contain an ever-increasing number of cores which offer improved performance compared to single-core systems. These performance improvements come at an increased cost for power consumption since applications may not utilize all cores at any given time. However, the unique characteristics of these systems such as heterogeneous cores and shared memory provide numerous opportunities for power savings. The impact and granularity of savings depends on the individual application characteristics.

Typically, power consumed in a multicore system can be computed as a product of power consumed by a core and the number of cores. However, this assumes that each of the cores are homogeneous i.e. they have similar power and performance characteristics. In reality, these cores are heterogeneous [6] as a result of which utilization varies across cores. Thus, models for predicting power consumption in Chip Multi-Processors need to consider the activity of individual cores and memory hierarchy.

In this work, a statistical approach that models the impact of individual cores and memory hierarchy on overall power consumed by Chip Multiprocessors (CMP) is developed using Performance Counters. Our model is simple and technology dependent in that it contains lesser number of parameters and as a result executes faster than traditional models. Evaluation of the model shows a strong correlation between core-level activity and power consumption and that the model predicts power consumption for newer observations with minimal errors. In summary, our contributions include:

- Developing statistical power models using linear regression for estimating per-core power consumption by leveraging parameters such as concurrency for multicore systems
- Discussing applications where the model can be utilized towards estimating power consumption.

2. RELATED WORK

Prior works analyzed the applicability of multi-cores for mobile embedded devices. Through analysis, Nvidia claimed that multi-core processors are power-efficient compared to single-core processors [15]. Majority of the approaches for power-aware multicore systems can be classified into the following.

Energy Modeling: Economou *et al.* [8] and Isci *et al.* [12] modeled the energy consumption of embedded devices using performance counters. Khan *et. al* [13] modeled the energy consumption of multi-core systems using a statistical learning approach. Wang *et al.* [20] developed SPAN, a software power monitoring tool which correlates program segments

CF'16, May 16-19, 2016, Como, Italy

DOI: http://dx.doi.org/10.1145/2903150.2911714

with power consumption. Fan *et al.* [9] analyzed the power consumption characteristics of data centers and studied the optimal provisioning of resources. Sangaiah *et al.* [18] developed regression models to predict the performance of Chip Multiprocessors. In contrast to the above approaches, we develop a simple and a technology-independent model to estimate per-core power consumption as a function of core-level activity for embedded multicore systems.

Dynamic Power Management: Numerous approaches utilized Dynamic Voltage and Frequency Scaling for power management in multi-cores [3] [5] [14] [11] [4] [10] [2]. These approaches assume that per-core DVFS is feasible and that depending on the workload, core voltage/frequency can be adjusted for power savings. In addition, Kolpe *et al.* [14] investigated the possibility of clustering different cores depending on core activity and adjust the core voltage/frequency on a per-cluster basis. Fu *et al.* [10] proposed to consolidate the cores that were scaled down and shutdown unused cores for power savings.

3. BACKGROUND

3.1 Embedded Multicore Systems

Embedded Multicore systems have unique characteristics compared to single-core systems. These characteristics enable us to rethink the problem of energy modeling and management for multicore systems. Further, the working set of applications may not be known in advance which exacerbates the problem of energy management in multi-core systems since applications may use any number of cores to expedite the execution of tasks. In our model, we consider a particular class of multicore systems called the Chip Multiprocessors. Figure 1 describes a Chip Multiprocessor. The characteristics of a Chip Multiprocessor are the following.

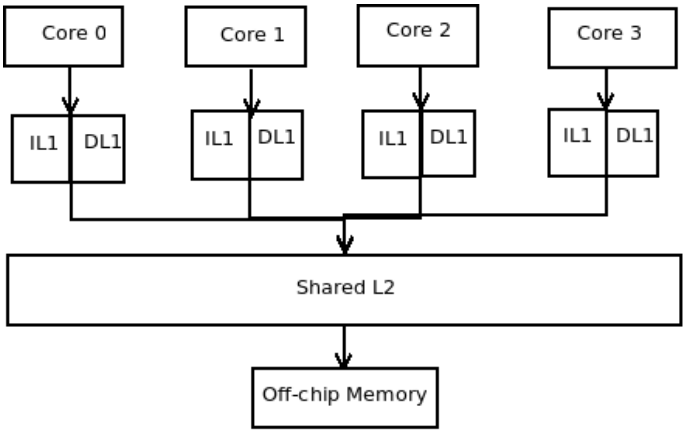

Figure 1: A Chip Multiprocessor

Heterogeneous Cores:

Heterogeneous cores [6] refer to varying computational capabilities for a given set of cores. For instance, certain subset of the cores can be high-power and high-performance cores while the rest are low-power cores. In this case, real-time applications need to execute on high-power cores so as to complete the tasks before their corresponding deadlines. Background applications may be placed on low-power cores due to their periodic and lightweight requirements. Thus applications and their corresponding execution times need to be analyzed before placing them on the appropriate cores.

Shared-L2:

In an embedded multicore system, L1 caches are private to each core while L2 caches typically called Last-Level Caches (LLC) are shared among different cores. The shared nature of the L2 cache makes it a critical bottleneck for the memory hierarchy since numerous applications compete for access to the L2 cache. To alleviate this bottleneck, shared memory can be partitioned among cores so as to avoid access to off-chip memory.

3.2 Power Modeling

Power Modeling is necessary to understand the various factors that impact power consumption in mobile devices. While power can be modeled at different levels of the system, architectural-level models capture a high-level view of the system and can be used to make better power-performance trade-offs by considering application behavior. In the recent past, power modeling has become a significant challenge due to the integration of numerous processing cores and Graphics Processing Units (GPUs) on a mobile device.

Current approaches for power modeling can be classified into utilization-based power modeling [19] and performance counters [12] based power estimation.

Utilization-based Modeling: These approaches model the utilization of individual system components and estimate the corresponding power consumption. In this approach, the main challenge lies in computing the utilization of individual components and using it to estimate power consumption. Traditional approaches rely on external power monitors which provide coarse-grained power estimates. These estimates can further be correlated with component-level utilization through statistical approaches to derive per-component power estimates.

Performance Counters: Embedded processors are augmented with special-purpose registers called Performance Counters for collecting component-specific statistics. These statistics can be used to build power models for individual components. The procedure lies in collecting events that are representative of the power consumed in mobile devices and understanding the relationship between performance parameters and power consumption. Statistics for each of these events can be measured using cycle-level simulation tools such as Wattch [7] for power estimation.

3.3 Linear Regression

Linear Regression is typically used to model the relationship between a dependent variable and multiple independent variables. This means that the dependent variable scales linearly in a increasing or decreasing manner depending on the variance in independent variables. A linear regression is expressed in the form Y= A + B * X where Y and X are the dependent and independent variables respectively. Within the context of this work, dependent variable can represent power consumption and the independent variables represent the factors affecting power consumption. For a given problem, data pertaining to the independent and dependent variables is periodically sampled and a linear model is constructed.

More precisely, for a given observation in the testing set, dependent variable is predicted using the following equation.

$$Y = B_0 + B_1 * X_1 + B_2 * X_2 + ...B_n * X_n \quad (1)$$

where $B_0....B_n$ refer to the co-efficients used for describing the model.

Typically, sampled data is divided into training and testing sets for model fitting. A model generated using the training test is typically applied on the testing set to measure its accuracy. To fit a linear regression model for all the observations in the training set, the best-fitting approach can be used by minimizing the sum of squared errors of the actual power consumption values to the predicted ones.

4. MODELING POWER CONSUMPTION

In this section, we discuss the parameters used for constructing the per-core power model and further describe the procedure for sampling data and model construction.

4.1 Model Parameters

Our model considers parameters such as Concurrency, Power and Execution time for power estimation. In particular, we characterize the workload for each of the applications using these parameters and further utilize the sampled data to construct statistical power models for per-core power estimation. Below, we elaborate on each of the parameters contained in the model.

Concurrency:

Concurrency involves executing applications in parallel on multiple cores resulting in improved performance. To utilize multiple cores, we can choose to specify the number of cores at compile time. The advantage of executing applications concurrently is that it decreases execution time although it leads to increased power consumption.

Power:

Power can refer to dynamic or leakage power in an embedded multi-core system. Energy consumption is dependent on the execution time of the applications and power consumed by each of the cores. As transistor technology scales into nanometers, leakage power becomes a significant component of total system power and is projected to exceed dynamic power consumption. One of the popular strategies to reduce leakage power is to utilize Race-to-halt [1] where applications are executed at the highest frequency so that processors can be placed in sleep mode after the execution.

Execution Time:

Execution Time refers to the total time taken when an application is executed in one or more cores. It involves assessing the performance benefits and analyzing the power and performance trade-offs of individual cores. Since cores contain private L1 caches, access rates of the individual caches also contribute to the overall performance of the cores.

4.2 Data Sampling

We sample data using SPLASH2 benchmarks [21] by leveraging concurrency in multicore systems. In particular, data points pertaining to individual cores and memory hierarchy were gathered by running benchmarks on varying number of cores. Table 1 contains a brief description of SPLASH2 benchmarks that were ran for model generation.

We gather data points using the following procedure. Since we are interested in modeling power consumption at a core-level, applications were scheduled to execute on N cores and N data points were obtained. These data points refer to the activity at the core-level and memory hierarchy for each of the benchmarks. We extracted the performance statistics for the benchmarks using SESC simulator [16] while power consumption was measured using Wattch [7].

Table 1: SPLASH2 Benchmarks

Benchmark	Description
crafty	Chess Program with operations such as AND, OR, Exclusive OR and SHIFT
fft	Fast Fourier Transform algorithm for minimizing Inter-process communication
lunon	Implements sparse-matrix incomplete LU decomposition
ocean	Scientific workload for Performance Evaluation of parallel machines
radix	Iterative Algorithm for integer radix sort
waterspatial	Uses a different algorithm to solve molecular dynamics N-body problem

The question that arises is the need for another model since Wattch contains models built into it. We claim that the model constructed in this work is simpler containing lesser number of parameters than Wattch. Yet, we obtain accuracy of power estimates closer to Wattch as will be demonstrated in the evaluation section. Since the proposed model is simple, it executes faster as well.

4.3 Model Construction

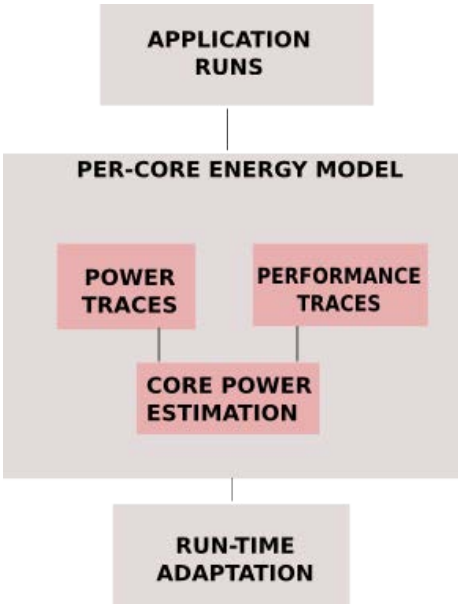

Figure 2: Power Model Construction

Our core-level energy model is built in the following manner. Figure 2 pictorially describes the per-core model for Embedded Multicore Systems. We ran benchmarks sched-

Table 2: Baseline Configuration

Parameter	Value
Number of Cores	8
Technology node	70 nm
Frequency	5 Ghz
L1 Instruction Cache	Cache Size: 32768 Line Size: 32 Associativity: 2
L1 Data Cache	Cache Size: 32768 Line Size: 32 Associativity: 4
L2 Unified Cache	Cache Size: 1048576 Line Size: 32 Associativity: 8

uled on varying number of cores and accumulated data for CPU cycles and cache access rates for each core and the corresponding power consumption. In particular, we selected CPU cycles and L1 cache (Instruction and Data Cache) access rates since they reflect the activity of CPU and L1 caches respectively for each of the cores. We claim that these parameters are suitable for predicting per-core power consumption. By choosing high-level parameters that contribute towards the power consumed by individual cores, our model offers a trade-off between simplicity and accuracy. Further collected data was used to create the statistical model for core-level power estimation which is of the following form.

$$P_{core} = B_1 * CPI + B_2 * ICache_{access} + B_3 * DCache_{access} + B_0 \quad (2)$$

where P_{core}, CPI, $ICache_{access}$ and $DCache_{access}$ refer to Average power consumed by core, Cycles Per Instruction, Average Instruction Cache access rate and Average Data Cache access rate respectively. B_0,B_1,B_2 and B_3 refer to the co-efficients obtained using the model.

5. MODEL EVALUATION

We develop our energy model using SESC [16], a microprocessor simulator for superscalar processors. SESC simulates both single and multi-core processors and emulates the MIPS instruction set processor. Our baseline configuration is displayed in table 2.

5.1 Model Training

We train the model using R [17], a statistical software package. Particularly, we use Multiple Linear Regression towards developing the statistical power model since there exists multiple independent variables and a single dependent variable. Table 3 contains the list of parameters and their corresponding model co-efficients .

Table 4 contains the results for R-Squared values for the per-core model. R-Squared values indicate the goodness of fit in accounting for variation in the data. An R-Squared value close to 1 indicates that there exists a strong correlation between independent and dependent variables. Similarly, R-Squared value of 1 shows a perfect correlation and that the model accounts for all variations in the data. In the case of multiple linear regression, Adjusted R-Squared values need to be considered since the model involves multiple independent variables.

Table 3: Model Co-efficients for Core-level Model

Parameter	Co-efficient
CPI	0.993
L1 Instruction Cache	245.779
L1 Data Cache	110.406
Intercept	-6.047

Table 4: Core-level Model Evaluation

Parameter	Value
Adjusted R-Squared	0.91
Predicted R-Squared	0.89
Root Mean Square Error	15.82

From the table, it is clear that R-Squared values lie closer to 1 which indicate the strong correlation between average power consumption and selected performance events such as Cycles Per Instruction, Average L1 Instruction and Data Cache Access rate. In the next subsection, we validate the model and provide analysis on the error rates associated with prediction.

5.2 Model Validation

To validate the model, we applied the model trained using the training set to the testing set consisting of completely new observations. In particular, we classified the data into 80% training and 20% testing and validated the model on the testing set. Table 4 contains the results for Predicted R-Squared and Root Mean Square error. It indicates that Predicted R-Squared values lie closer to 1 which means that the model is able to account for most of the variations on the testing set. The error associated with prediction called Root Mean Square Error (RMSE) denoted by $Error_{RMS}$ can be computed using the following equation.

$$Error_{RMS} = \sqrt{\sum_{i=1}^{n}(P_{Predicted}(i) - P_{Actual}(i))^2} \quad (3)$$

We obtained a root mean square error of 15.82 which shows that the model incurs a minimal error associated with prediction. Figure 3 shows a graph of predicted and actual values for power consumption for each of the samples in the testing set. The graph indicates that the model predicts observations in the testing set with increased accuracy.

6. APPLICATIONS

We discuss a few mechanisms where the model can be utilized for power estimation and further optimize power consumption in embedded multicore systems. Since the generated model is simple and that power estimates can be computed on the fly, it can be used as feedback for mechanisms

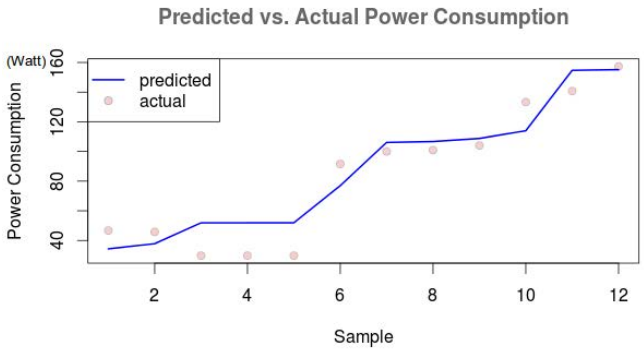

Figure 3: Model Validation

such as Dynamic Voltage and Frequency Scaling and Task Scheduling which are discussed below.

6.1 Dynamic Voltage and Frequency Scaling

Dynamic Voltage and Frequency scaling is used to reduce power consumption by scaling down the supply voltage. The decrease in supply voltage also incurs a corresponding reduction in frequency in computing systems. One of the main limitations of DVFS is that it increases the execution time of computational tasks. Thus it becomes necessary to configure DVFS according to the needs of applications.

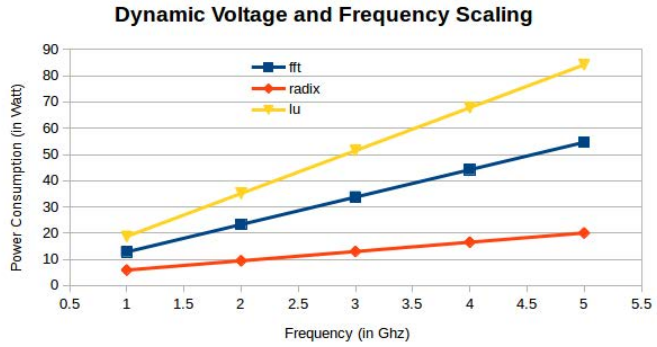

Figure 4: Impact of Dynamic Voltage and Frequency Scaling

The problem of dynamic voltage and frequency scaling is further exacerbated in multicore systems since there exists numerous heterogeneous cores and that applications can be concurrently executed on them. Since the model generates per-core power estimates using the activity of the individual cores and memory hierarchy, it can be used to scale frequency based on the estimates.

We use the generated model to estimate power consumption at varying frequencies using SPLASH2 benchmarks. Figure 4 contains the results for the Dynamic Voltage and Frequency Scaling. The results indicate a linear relationship between the frequencies and the corresponding power consumption for each of the applications. However the rate at which power consumption increases is the highest for Lunon benchmark compared to FFT and Radix applications since Lunon implements sparse matrix multiplication. Consequently, we also observed the Instructions Per Cycle (IPC) to be highest for the Lunon benchmark.

6.2 Task Scheduling

Task Scheduling is of increased importance in multicore systems due to the need for assigning tasks to cores that match their computational requirements. In addition to mapping tasks to cores, deadline requirements of the tasks need to be met. Thus depending on the schedulability of the tasks, tasks can be migrated across cores which may result in unused cores to be powered down thus conserving energy.

Power-aware task scheduling can be performed since the utilization of the individual cores and the resulting power estimates from the model can be monitored in real-time. This can further lead to task consolidation and core power-down resulting in energy savings.

The concept of static analysis is used to study the characteristics of applications before assigning them to the cores for task scheduling. Static analysis becomes a potential research challenge when multiple applications concurrently execute in multitasking systems. Thus the problem lies in accurately attributing resources to individual applications. Such a static analysis can be combined with our model for energy accounting purposes. We propose to investigate these ideas as part of future work.

7. CONCLUSIONS

We developed a statistical power model to estimate the power consumed by embedded multicore systems. The proposed power model quantifies the power consumed by a core as a function of activity of the individual cores and memory hierarchy. Linear Regression was used to generate the power model. The proposed model is simple and technology independent, in that it contains lesser number of parameters compared to existing models and yet obtaining accurate power estimates. Evaluation of the model showed a strong correlation between core-level activity and power consumption and that the model predicted power consumption for newer observations with minimal errors. In addition, we discussed a few applications where the model can be utilized towards estimating power consumption.

8. REFERENCES

[1] M. A. Awan and S. M. Petters. Enhanced race-to-halt: A leakage-aware energy management approach for dynamic priority systems. In *Proceedings of the 2011 23rd Euromicro Conference on Real-Time Systems*, ECRTS '11, pages 92–101, Washington, DC, USA, 2011. IEEE Computer Society.

[2] R. Basmadjian and H. De Meer. Evaluating and modeling power consumption of multi-core processors. In *Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference on*, pages 1–10, May 2012.

[3] D. Bautista, J. Sahuquillo, H. Hassan, S. Petit, and J. Duato. A simple power-aware scheduling for multicore systems when running real-time applications. In *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pages 1–7, April 2008.

[4] R. Bergamaschi, G. Han, A. Buyuktosunoglu, H. Patel, I. Nair, G. Dittmann, G. Janssen, N. Dhanwada, Z. Hu, P. Bose, and J. Darringer. Exploring power management in multi-core systems. In *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*, pages 708–713, March 2008.

[5] W. L. Bircher and L. K. John. Analysis of dynamic power management on multi-core processors. In *Proceedings of the 22Nd Annual International Conference on Supercomputing*, ICS '08, pages 327–338, New York, NY, USA, 2008. ACM.

[6] F. Bower, D. Sorin, and L. Cox. The impact of dynamically heterogeneous multicore processors on thread scheduling. *Micro, IEEE*, 28(3):17–25, May 2008.

[7] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th annual international symposium on Computer architecture*, ISCA '00, pages 83–94. ACM, 2000.

[8] D. Economou, S. Rivoire, and C. Kozyrakis. Full-system power analysis and modeling for server environments. In *In Workshop on Modeling Benchmarking and Simulation (MOBS*, 2006.

[9] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ISCA '07, pages 13–23, New York, NY, USA, 2007. ACM.

[10] X. Fu and X. Wang. Utilization-controlled task consolidation for power optimization in multi-core real-time systems. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2011 IEEE 17th International Conference on*, volume 1, pages 73–82, Aug 2011.

[11] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi. An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 39, pages 347–358, Washington, DC, USA, 2006. IEEE Computer Society.

[12] C. Isci and M. Martonosi. Runtime power monitoring in high-end processors: Methodology and empirical data. Technical report, Princeton University Electrical Eng. Dept., September 2003.

[13] S. Khan, P. Xekalakis, J. Cavazos, and M. Cintra. Using predictive modeling for cross-program design space exploration in multicore systems. In *Proceedings of the 16th International Conference on Parallel Architecture and Compilation Techniques*, PACT '07, pages 327–338, Washington, DC, USA, 2007. IEEE Computer Society.

[14] T. Kolpe, A. Zhai, and S. Sapatnekar. Enabling improved power management in multicore processors through clustered dvfs. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, pages 1–6, March 2011.

[15] NVIDIA. Variable smp - a multi-core cpu architecture for low power and high performance. White Paper.

[16] P. M. Ortego and P. Sack. Sesc: Superescalar simulator. Technical report, 2004.

[17] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[18] K. Sangaiah, M. Hempstead, and B. Taskin. Uncore rpd: Rapid design space exploration of the uncore via regression modeling. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '15, pages 365–372, Piscataway, NJ, USA, 2015. IEEE Press.

[19] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welsh. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 188–200. ACM, 2004.

[20] S. Wang, H. Chen, and W. Shi. Span: A software power analyzer for multicore computer systems. *Elsevier Sustainable Computing: Informatics and Systems*, page In press, 2011.

[21] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta. The splash-2 programs: Characterization and methodological considerations. pages 24–36. Proceedings of the 22nd Annual International Symposium on Computer Architecture, June 1995.