High-Resolution Online Power Monitoring for Modern Microprocessors

Fabian Oboril, Jos Ewert and Mehdi B. Tahoori
Chair of Dependable Nano Computing (CDNC), Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
Email: {fabian.oboril, mehdi.tahoori}@kit.edu, jos.ewert@student.kit.edu

***Abstract*—The power consumption of computing systems is nowadays a major design constraint that affects performance and reliability. To co-optimize these aspects, fine-grained adaptation techniques at runtime are of growing importance. However, to use these tools efficiently, fine-grained information about the power consumption of various on-chip components at runtime is required. Therefore, here we propose a novel software-implemented high-resolution (spatial and temporal) power monitoring approach that relies on micro-models to estimate the power consumption of all microarchitectural components inside a processor core. Combined with a self-calibration technique that uses an available on-chip power sensor, our power estimation approach can achieve an accuracy of more than 99 % and provides deep insights about the power dissipation inside a processor core during workload execution.**

I. Introduction

In recent years, the microprocessor power consumption emerged to be a major design aspect [1]. Moreover, due to the end of the *Dennard scaling* model [2], the power density increases with downscaling, which affects the microprocessor reliability [3]. Especially thermally accelerated effects such as wearout are nowadays a major issue [4].

To meet the power and reliability constraints, while maintaining a high performance, modern systems and processors employ various power and thermal management techniques such as task migration, power and clock gating, as well as Dynamic Voltage and Frequency Scaling (DVFS) [5]–[7]. In fact, to use these tools efficiently, i.e. to co-optimize performance, reliability and power, detailed and accurate power information of various on-chip components during the workload execution is necessary. In this regard, the efficiency of the adaptation (performance penalty vs. power/temperature reduction) strongly depends on the available spatial power information. In particular, the final system performance employing more localized adaption techniques will be considerably better [8].

Therefore, several methods to monitor the power consumption at runtime at different granularities have been explored. These can be classified into two categories: 1) Direct measurements via sensors [5], [9]; 2) Indirect estimation using information about the resource utilization [7], [10]–[15]. The advantages of the first class of approaches are the high temporal resolution and the high accuracy of the obtained power values. However, the spatial resolution is poor, as only few sensors are employed due to their high costs [7], [9]. The approaches in the second category are considerably cheaper, as no additional hardware is required. Instead, available performance counters are used in combination with analytical models for power estimation. However, due to massive chip-to-chip variations in the nanometer era and time-based variations due to voltage, frequency and temperature (VFT) changes, linear regression approaches [10]–[14] are often very inaccurate [16]. Moreover, many models can only monitor the power consumption at core-granularity (i.e. low spatial resolution) [10]–[14]. Just very few approaches such as [15] can model the power consumption of microarchitectural components such as caches, execution units or instruction decoders (i.e. high spatial resolution). However, as these models are very complex, they cannot be used at runtime (i.e. low temporal resolution). *In summary, an accurate and cheap power estimation methodology with a high spatial (i.e. per microarchitectural component) and temporal resolution, that can drive fine-grained system adaptation, is still missing.*

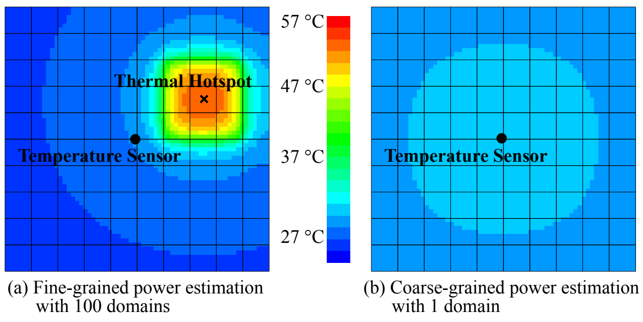
In this work we present a novel power estimation and monitoring approach that closes this gap. It is based on fine-grained models, i.e. power can be estimated for all microarchitectural components and VFT changes are taken into consideration. Yet, its temporal resolution is high enough (every 1-10 ms) to obtain the power consumption during workload executions. Moreover, our approach uses an available on-chip sensor to calibrate itself to minimize the estimation inaccuracy. Finally, the entire approach is generic, and thus, can be applied to different microprocessor architectures or different technologies. Our experimental results, based on measurements for different Intel processors manufactured in different technology nodes, show that the average inaccuracy for various applications is less than 1 %. Hence, our method is as accurate as an on-chip sensor, and in addition it reveals deep insights about the power behavior of the different microarchitectural components. Nevertheless, it is as flexible and low-cost as an analytical model running in software/firmware.

II. Motivation

In this section, we present two examples that motivate the need for fine-grained power knowledge with high spatial and temporal resolution, that can combine high performance and high reliability.

If the power consumption of each microarchitectural block is available, only those blocks that consume too much power are targeted by fine-grained adaptation techniques. In contrast, if only per-core power information is available, and the combined power consumption is too high, the entire core has to be "reconfigured" (e.g. lower supply voltage and frequency). As a consequence, this per-core adaptation costs more performance [8], as the fine-grained changes reduce the throughput only for a small number of domains, while the majority can still operate as before.

Moreover, by having detailed power information, it is possible to predict temperature hotspots more accurately (in time and space) as temperature follows power. Hence, a proactive adaptation can be performed to avoid a critical temperature before it actually occurs. As a result, reliability can be significantly improved [18]. Instead, if only per-core power information is accessible, thermal hotspots may not be identified. This issue is illustrated in Fig. 1. Here, the power



(a) Fine-grained power estimation with 100 domains
(b) Coarse-grained power estimation with 1 domain

Fig. 1. Two cores with the same power consumption, but different temperature due to different power distribution (extracted with HotSpot [17])

Special Purpose Registers (SPRs)	Functionality	Micro-Model Usage	Monitored Component
L2_TRANS_ALL_X	L2 operations with X=Read/Write/Total	L2 accesses (read/write/hit/miss)	L2Cache, DCache
UOPS_DISPATCHED_PORT_X	Dispatched instructions at port X	Instructions for each functional unit	Execute, LSU, Registers, Scheduler
INSTS_RETIRED.ANY	Retired instructions	Total instruction count	Fetch, Retire
CPU_CLK_UNHALTED_*	Active cycles of the entire core	Total/Busy/Idle cycles	Overall core
BR_*_ALL_BRANCHES	Executed branches	Total/Mispredicted branch count	Branch Predictor
MEM_UOPS_RETIRED_X	Retired memory operations with X=Load/Store	Memory operations (load/store)	LSU, TLB, DCache
IDQ_MITE_ALL_UOPS	Decoded micro-instructions	Decoded instructions	ICache, Fetch, Decode, ROB
FP_COMP_OPS_EXE.*, SIMD_FP_256.*	Scalar/Vector FP operations	FP/Vector operations	Execute

TABLE I
EMPLOYED SPRS FOR THE PROPOSED POWER ESTIMATION (HERE FOR INTEL PROCESSORS)

density in the upper right chip corner is considerably higher than in the remaining part, which leads to a thermal hotspot. This hotspot can be detected, only if fine-grained power information is available, otherwise it remains undetected. Also, a temperature sensor does not detect this hotspot, as long as it is not placed in the affected area.

In summary, fine-grained power information at runtime is mandatory for effective power management and reliability-aware policies.

III. High-Resolution Power Monitoring

Based on the previous motivation, the general idea behind our proposed fine-grained power estimation approach is to obtain the power consumption of each microarchitectural block (domain). Therefore, each domain has its own power model (*micro-model*) that takes supply voltage, frequency, temperature and resource utilization into account. While the first three aspects can be usually accessed directly by reading certain Special Purpose Registers (SPRs) [5], [7], the utilization has to be obtained indirectly via performance counters. Having all this data, the dynamic and static power can be estimated for each domain, and thus also the combined power can be calculated.

The two major challenges of this approach are: 1.) Development of accurate yet fast micro-models that take VFT changes into account; and 2.) Finding appropriate performance counters to estimate the resource utilization and accessing them frequently (i.e. every 1-10 ms). In this work, we employ the micro-models from McPAT [15], which is a power estimation framework containing physical power models for all microarchitectural blocks in modern processors. These models are based on technology data and use the supply voltage, temperature and clock frequency as well as a variety of performance statistics as inputs to compute the power. The technology data (threshold voltage, oxide thickness, currents, feature size, etc.) for these models is based on the ITRS roadmap for technology nodes ranging from 90 nm to 16 nm. As a result of these detailed models, a power estimation with high spatial resolution can be achieved. However, the default models from McPAT are infeasible for a runtime power estimation with a high temporal resolution. This is due to the fact that these are computational intensive and that whenever a parameter (V,F,T) changes, all models have to be rebuilt which requires several seconds. Altogether this results in an evaluation time of several minutes, if the power consumption of a workload running for a few seconds should be analyzed. Therefore, we enhanced the models to be faster without impacting their accuracy. The first step was to remove all unnecessary computations. For example, McPAT always calculates the worst-case power for the last time period beside the actual power consumption. Since this worst-case data is not required for our power monitoring approach, such computations could be easily removed. In addition, we updated the micro-models according to the Equations (1) and (2) to take VFT changes into account [19] without requiring a re-initialization.

$$P_{dynamic} \sim f \cdot V_{dd}^2 \tag{1}$$

$$P_{static} \sim V_{dd} \cdot \left(aT^2 \cdot \exp(\frac{\alpha V_{dd} + \beta}{T}) + b \cdot \exp\left(\gamma V_{dd} + \delta\right) \right) \tag{2}$$

At runtime V_{dd} (supply voltage), T (temperature) and f (clock frequency) are obtained from SPRs (here: `IA32_PERF_STATUS` and `IA32_THERM_STATUS`). The (constant) fitting parameters $a, b, \alpha, \beta, \gamma, \delta$ that describe the temperature and voltage impact on the static power were extracted using McPAT's default static power model. As a result of these modifications, the time required to estimate the power consumption of all microarchitectural blocks within a single core is less than $50\,\mu$s for an Intel Core i5-2400 compared to 7.7 s (five orders of magnitude improvement). Hence, if the power estimation is performed once every 10 ms, our model can be used to monitor the power consumption of a workload, without inferring a huge performance penalty ($< 0.7\,\%$). Please note that this temporal resolution is more than enough to drive system adaptations, as pointed out in [10].

Beside the micro-models themselves, it is a challenge to select and frequently access the performance counters that are required to feed the micro-models. In particular, the access is a very demanding task, as various SPRs (here: 31) need to be accessed at almost the same time (to make sure that the gathered data is correlated). Therefore, we developed our own access routines to meet the tight timing constraints. These are written in C++ and use the `/dev/cpu/CPUNUM/msr` interface of Linux to access the required SPRs. As soon as all data for the last time period is gathered, this data is directly fed to the micro-models such that the overall power estimation time is less than the mentioned $50\,\mu$s.

The selection of appropriate SPRs is challenging as well, although the micro-models require very specific input data about the resource utilization of each microarchitectural block. The difficulty arises from the fact that for many microarchitectural components the utilization cannot be directly obtained from a single SPR. In fact, for most of the blocks a combination of several SPRs has to be considered. For example, the Intel processors of our experimental case study (see Section V) do not allow to directly count the number of instructions executed by ALU_0. Instead, only the number of instructions dispatched via Port_0 can be accessed. However, this port is also used by FPU_0. Thus, the number of dispatched instructions at Port_0 is obtained, and by using additional SPRs also the overall ratio of integer to floating point instructions is extracted. Finally, using both information together, an approximation for the number of instructions executed by ALU_0 is computed and used by the corresponding micro-model. Consequently, for the specific Intel processors we employed for our study, in total 31 SPRs had to be accessed, although the number of microarchitectural components is considerably lower. Three of the SPRs are employed to obtain the clock frequency, supply voltage, temperature as well as the combined power consumption for all cores (here: `MSR_PP0_ENERGY_STATUS`). The remaining 28 SPRs, detailed in Table I, are used to infer the resource utilization.

Please note that the same temperature is used by all micro-models that belong to one core, as usually only the per-core temperature is available. To alleviate this issue, an analytical temperature model such as hotspot [17] can be added to the framework to use more accurate temperature data, and thus increase the overall accuracy.

Putting all this together results in a fine-grained power estimation approach with high temporal and spatial resolution (evaluation every 1-10 ms for all microarchitectural components) that considers VFT changes at runtime.

It is important to note that this approach is platform independent. Hence, once the technology parameters are set and the required performance counters are selected, the power model can be applied to all processors of the same family (i.e. same architecture and process technology). If the architecture or technology node changes, only the corresponding parameters have to be adjusted, e.g. the number of functional units or accessed SPRs (architecture) or threshold voltage (technology), while the underlying framework can be kept. This is a great advantage over regression-based techniques that require a new training every time a single parameter changes. However, as microprocessors fabricated at nanoscale CMOS nodes have significant process, and thus power consumption variation, also a non-calibrated micro-model estimation can be very inaccurate from one chip to another. Therefore, our power estimation technique makes use of a single available on-chip power sensor to calibrate the micro-models at runtime (see Section IV). Consequently, our proposed technique delivers very accurate results (see Section V).

In a real system the micro-models and the required SPR accesses should be handled by a firmware or driver which runs in the background as part of the operating system. Ideally, it is provided by the manufacturer directly to have lightweight and accurate models. Nevertheless, also opensource solutions like the one presented in this paper based on McPAT can provide good accuracy and resolution.

IV. Self-Calibration Methodology

As mentioned in the previous section, self-calibration for power estimation approaches is nowadays required, due to huge amount of process variation. This variation cannot be captured by the micro-models using the Equations (1) and (2), as the underlying technology data is inherently incapable to account for chip-to-chip and core-to-core variations. Therefore, a single available power sensor can be accessed at runtime to calibrate the model such that it delivers accurate power estimation results irrespective of the process corner the chip belongs to. Another reason to use a power sensor for calibration is the fact that all software-based models, including our proposed technique can only capture an "average" behavior of the system. For example, the estimated power consumption for an ALU will always be the same, no matter if the accesses are simple AND operations or complex 64-bit additions. This under- or overestimation of workload influences on the power consumption can be reduced with a calibration technique.

To improve the accuracy of our power estimation technique we use a periodic self-calibration method based on an *exponential moving average* (EMA) correction, which is calculated as follows:

$$s_1 = \frac{p_{real,1}}{p_{est,1}}, \quad s_t = \alpha \cdot \frac{p_{real,t}}{p_{est,t}} + (1-\alpha) \cdot s_{t-1} \qquad (3)$$

In this regard, $p_{real,t}$ is the real power consumption at step t obtained from an on-chip power sensor, $p_{est,t}$ is the corresponding estimated power consumption, and α reflects the influence of past power values as well as the actual one. For this work, α is set to 0.9. If α is smaller, sudden workload changes that are not captured by the micro-models (e.g. ADD instead of AND operations) will not be corrected fast enough. On the other hand, if α is larger, the history effect vanishes, which means that the calibration is only dependent on the current estimation error but not on the history. The result of the EMA calculation, s_t, is used as *correction factor* by our self-calibration approach for all micro-models, i.e. $p_{est,t,new} = s_t \cdot p_{est,t}$.

Processor	Intel Core i5-2400 (32 nm, 1.6 GHz–3.1 GHz, 1.06 V–1.22 V) Intel Core i5-3450 (22 nm, 1.6 GHz–3.1 GHz, 0.88 V–1.07 V) All sleep modes + DVFS activated
OS / Benchmarks	Linux Kernel 3.2 / SPEC2006
Measurements	Power estimation & Update of EMA: Every 1 ms Update of correction factor: Every 10 ms

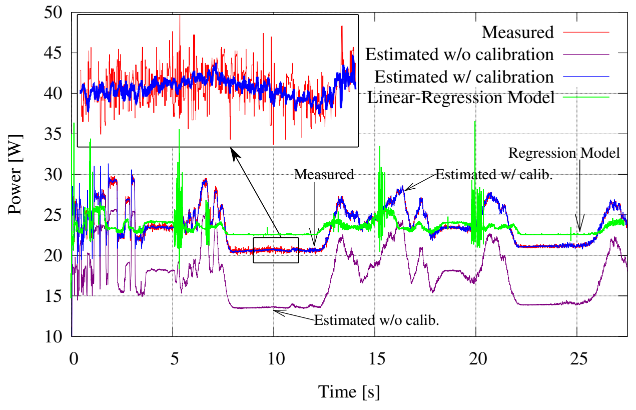
TABLE II
Experimental Setup

Furthermore it is important to note that using this self-calibration approach, the power consumption of all sub-core components is adjusted with the same factor. This is very reasonable, since our results show that not a single microarchitectural block is responsible for the estimation inaccuracy. If this was the case, an additional training for the corresponding micro-model would be required before all components could be adjusted with same factor.

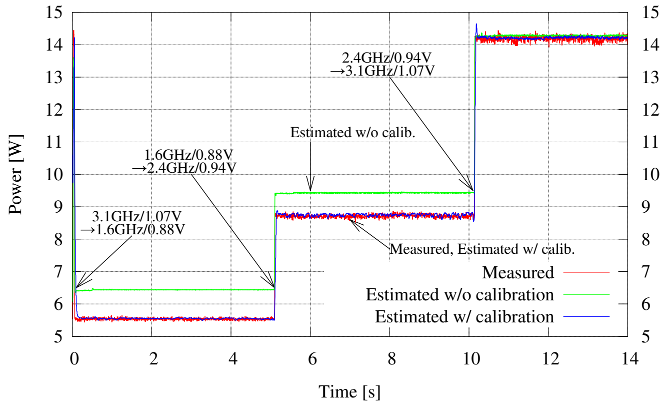
V. Experimental Results

In order to evaluate the accuracy and performance overhead of our proposed online power estimation approach, we employed the technique in a real system with the configuration shown in Table II. For the calibration the available on-chip power sensor is employed (accessed via the SPR which measures the combined power consumption of all cores including their L1- and L2-caches. Moreover, we used the on-chip temperature sensor to obtain information about the per-core temperature which is used by our power models. As workloads we use the SPEC2006 benchmark suite. Since these applications are single-threaded, we execute four instances in parallel to fully utilize the quad-core processors. The power estimation interval is set to 1 ms to demonstrate the capabilities of our proposed power monitoring methodology. As a result, the performance impact due to the additional computations for accessing 119 SPRs (29 SPRs for each core plus the SPRs for voltage, frequency and combined power) and estimating the power consumption is around 6 % on average over all workloads. If the monitoring interval is extended to 10 ms, which is still fine-grained enough according to [10], the performance impact is less than 0.7 %. In addition, if at least one core is not used, there is no performance overhead at all. Beside a performance impact, the additional calculations also infer an energy overhead. However, since the average power consumption decreases due to switching threads (from workload to power model and back), the average energy consumption increases by less than 6 % for a monitoring interval of 1 ms. In case of 10 ms time steps, the overhead is negligible.

The first observation is depicted in Fig. 2, namely that our model can capture the workload trend including VFT changes at runtime very accurately, even if the self-calibration is not used. This behavior can be seen in all applications. *Hence, we can conclude that our micro-models can capture the workload impact on different microarchitectural blocks very well.* Otherwise, there would be one benchmark in which the estimated power trend would not match the real power trend. Please note that due to the lack of sensors in the different components, only this indirect proof of accuracy is possible. Of course, the power estimation without calibration is inaccurate when looking at the absolute power numbers, which is due to the fact that the technology models do not capture the impact of process variation and that the models are based on ITRS predictions. With more accurate technology data, the inaccuracy would be lower. However, if the self-calibration is employed, the inaccuracy for the combined power consumption is reduced to a negligible value. Over all SPEC2006 applications the average and maximum estimation error is less than 0.1 % and 0.4 % respectively for both processors, i.e. for different technology nodes. Compared to the linear regression based model proposed in [13] that estimates the combined power for each core, our model is much more accurate. In particular, it can capture



(a) gcc



(b) hmmer with manual DVFS

Fig. 2. Power consumption of two applications for the Intel Core i5-3450 (measured=sensor readout)

the workload trend, which is not possible with simpler models as shown in Fig. 2(a). As a result, it is impossible with these simpler models to get any accurate sub-core information.

As described in Section IV, our self-calibration technique continuously compares the estimated and the measured power consumption, and if there is a mismatch the correction factor is adjusted. We observed that this correction factor converges very fast to 1 after a short period at the beginning of a workload, i.e. no further correction is applied. Only if in the following very sharp power gradients occur, the power estimation becomes inaccurate again, and hence the correction factor deviates in these cases from 1. In both cases the inaccuracy is due to two reasons: 1) At the beginning as well as in case of sharp power gradients, frequency and supply voltage are typically not very stable. However, as those parameters are read only every 1 ms, the "reaction" of the power estimation is slightly delayed and hence not very accurate. 2) The SPR values reflect the average behavior in the last period and hence are a kind of low-pass filter, which makes it hard to capture almost instantaneous power changes that can arise from waking up or sending a component from/to sleep.

Beside having an accurate core-level power estimation, it is also possible to get deep insights about the power consumption of different microarchitectural components as shown in Fig. 3. In other words, our monitoring framework increases the spatial resolution of the single sensor employed for the calibration.

VI. Conclusion

The microprocessor power consumption is a major design constraint, which affects performance and reliability. Therefore, it is of decisive importance to co-optimize performance, reliability and power at runtime by applying fine-grained adaptation techniques. However, to use these efficiently, accurate information about the power consumption with a high spatial and temporal resolution is required. For this purpose, we presented in this work a novel, software-implemented high-resolution power monitoring approach to obtain the power consumption of all microarchitectural components at runtime. In addition, it is platform independent and considers voltage, frequency and temperature changes at runtime. Moreover, it employs a self-calibration technique that uses the information of a single on-chip power sensor to improve the estimation accuracy. By this means, an estimation accuracy of more than 99 % can be achieved. Hence, it is as accurate as an on-chip sensor and in addition capable of monitoring the power behavior of various microarchitectural components. Nevertheless, it is as flexible and low-cost as an analytical model running in software.

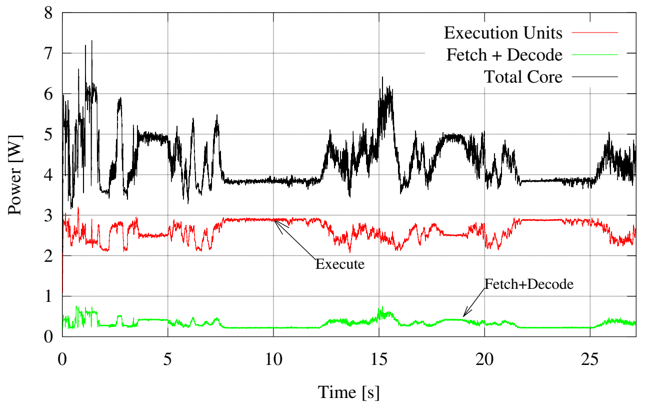


Fig. 3. Power trace for the execution unit as well as Fetch+Decode of a single core (Intel Core i5-3450, gcc benchmark)

References

[1] ITRS, http://www.itrs.net, 2013.
[2] R. Dennard *et al.*, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE SSC*, pp. 256–268, Oct 1974.
[3] S. Borkar *et al.*, "The Future of Microprocessors," *Commun. ACM*, pp. 67–77, May 2011.
[4] F. Oboril *et al.*, "Aging-Aware Design of Microprocessor Instruction Pipelines," *IEEE TCAD*, pp. 704–716, May 2014.
[5] Intel, "Desktop 3rd Gen Intel Core Processor Family: Datasheet," 2012.
[6] Qualcomm Inc., *Snapdragon S4 Processors: System on Chip Solutions for a New Mobile Age – White Paper*, Oct. 2011.
[7] M. Floyd *et al.*, "Introducing the Adaptive Energy Management Features of the Power7 Chip," *IEEE Micro*, pp. 60–75, March 2011.
[8] P. Petrica *et al.*, "Flicker: A Dynamically Adaptive Architecture for Power Limited Multicore Systems," *Comp. Arch. News*, pp. 13–23, Jun. 2013.
[9] M. Ware *et al.*, "Architecting for Power Management: The IBM POWER7 Approach," in *HPCA*, Jan. 2010, pp. 1–11.
[10] K. Rajamani *et al.*, "Online Power and Performance Estimation for Dynamic Power Management," *IBM, RC-24007, Tech. Rep*, 2006.
[11] P. Gschwandtner *et al.*, "Modeling CPU Energy Consumption of HPC Applications on the IBM POWER7," in *PDP*, Feb 2014, pp. 536–543.
[12] K. Singh *et al.*, "Real Time Power Estimation and Thread Scheduling via Performance Counters," *Comp. Arch. News*, pp. 46–55, Jul. 2009.
[13] Y. Sun *et al.*, "Low-cost Estimation of Sub-system Power," in *Green Comp. Conf.*, June 2012, pp. 1–10.
[14] S. Wang *et al.*, "SPAN: A software power analyzer for multicore computer systems," *Sus. Comp.: Info. and Sys.*, pp. 23–34, 2011.
[15] S. Li *et al.*, "McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures," in *Micro*, Dec. 2009, pp. 469–480.
[16] J. McCullough *et al.*, "Evaluating the effectiveness of model-based power characterization," in *USENIX Annual Technical Conf*, 2011.
[17] W. Huang *et al.*, "HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design," *IEEE TVLSI*, pp. 501–513, May 2006.
[18] J. Henkel *et al.*, "Reliable On-chip Systems in the Nano-era: Lessons Learnt and Future Trends," in *DAC*, 2013, pp. 99:1–99:10.
[19] W. Liao *et al.*, "Temperature and Supply Voltage Aware Performance and Power Modeling at Microarchitecture Level," *IEEE TCAD*, pp. 1042–1053, July 2005.