Improving the Accuracy of the Leakage Power Estimation of Embedded CPUs

Ting-Wu Chin, Shiao-Li Tsao, Kuo-Wei Hung, and Pei-Shu Huang
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan

***Abstract*—Previous studies have used on-chip thermal sensors (diodes) to estimate the leakage power of a CPU. However, an embedded CPU equips only a few thermal sensors and may suffer from considerable spatial temperature variances across the CPU core, and leakage power estimation based on insufficient temperature information introduces errors. According to our experiments, the conventional leakage power models may have up to 22.9% estimation error for a 70-nm embedded CPU. In this study, we first evaluated the accuracy of leakage power estimates based on thermal sensors on different locations of a CPU and suggested locations that can reduce the error to 0.9%. Then, we proposed temperature-referred and counter-tracked estimation (TRACE) that relies on temperature sensors and hardware activity counters to estimate leakage power. The simulation results demonstrated that employing TRACE could reduce the error to 3.4%. Experiments were also conducted on a real platform to verify our findings.**

Keywords—leakage power; modeling; thermal sensor;

I. Introduction

Dynamic power and thermal management are critical for battery-operated embedded systems. The efficiency and effectiveness of these schemes depend heavily on the accuracy of real-time power estimation [1]. Leakage power occupies a substantial portion of the total power consumption of modern CPUs; therefore, estimating leakage power accurately is critical [5]. In this study, we focus on the improvement of leakage power estimation models of CPUs.

Embedded CPUs are usually equipped with only a few thermal sensors to reduce costs, and they are often placed at the hot spots [5]. The temperature information from hot spots results in inaccurate leakage power estimations. We conducted experiments on a commercial multiprocessor system-on-chip (MPSoC), the Samsung Exynos 5422, which has only one thermal sensor per core, and used an infrared camera to observe the spatial temperature variance. Fig. 1 shows the spatial temperature variance when the CPU2006 benchmark [10], hmmer, is running under the ambient temperature of 20ºC. If we use the peak temperature to estimate the leakage power, an error up to 12.7% can be introduced.

Hence, in this study, we first evaluated the accuracy of leakage power estimates based on thermal sensors at different locations and determined locations that could improve the accuracy of these estimates. We then proposed temperature-referred and counter-tracked estimation (TRACE) which is a counter–sensor hybrid approach for estimating the leakage power of a CPU. The major contributions of this study are:

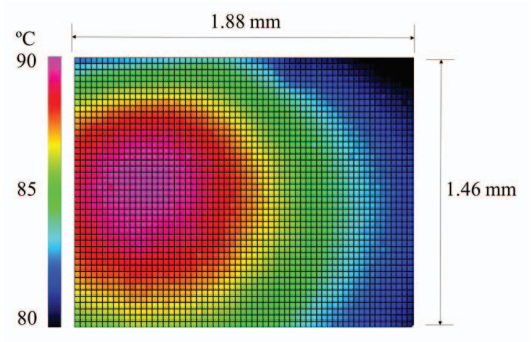

Fig. 1. The spatial temperature variance of a Samsung Exynos 5422 running the CPU2006 benchmark, hmmer.

- To the best of our knowledge, this is the first study to investigate inaccuracy in leakage power estimation caused by spatial temperature variances and insufficient temperature information from thermal sensors.
- We evaluated leakage power estimates based on thermal sensors on different locations and determined locations that could improve their accuracy.
- We proposed TRACE, which can provide accurate leakage power estimates for CPUs without the needs for additional hardware resources.

The rest of the paper is organized as follows: Section II summarizes related works; Section III illustrates the problem of leakage power estimation; Section IV introduces the proposed method; Section V discusses the simulation and experimental results; and Section VI concludes this study.

II. Related Works

Estimating the leakage power of a CPU with a single thermal sensor, [1, 3, 9] implicitly assume that the whole CPU core shares the same temperature. However, the spatial temperature variances across a CPU core appear and errors are introduced by models relying on only one hot-spot thermal sensor. Reference [6, 7, 8] suggest estimating the leakage power of regions of a CPU based on per-region temperatures. However, an embedded CPU usually equips only one or a few thermal sensors and the model based on per-region temperature information cannot be directly applied. Our work investigated the accuracy of the

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

sensor-based leakage power estimation across different numbers and locations of thermal sensors, and presented TRACE to estimate leakage power more accurately for embedded CPUs.

III. Evaluation of Sensor Locations for Leakage power Estimation

According to our experiments, the thermal sensors on the hot spots could not provide sufficient temperature information for estimating the leakage power of a CPU, i.e. up to 22.9% of estimation error.

Sensor-based leakage power estimation relies on runtime temperatures reported by on-chip thermal sensors. According to the BSIM3 model [11], the simplified leakage power of a transistor can be denoted as P_s^t :

where I_l and V_d are the leakage current and supply voltage of a transistor, respectively. T is the temperature of the transistor; $C1$, $C2$, and $C3$ are coefficients according to a particular CMOS production. According to [12], we can assume

$$P_s^t = I_l \times V_d = (C1 \times T^2 e^{\frac{C2}{T}} + C3) \times V_d \quad (1)$$

the same temperature of all transistors in a region with a uniform design such as a functional block. Liu et al. also showed we can approximate leakage power with second order Taylor expansion within the temperature range of 25ºC and 120ºC. The summation of the leakage power of all functional blocks (i.e., the leakage power of a CPU), denoted as P_s^c, is:

$$P_s^c \approx \sum_{i=1}^{K} (\alpha_i + \beta_i \times (T_i - T_f) + \gamma_i \times (T_i - T_f)^2) \times V_d \quad (2)$$

where K is the number of functional blocks of a CPU, α_i, β_i, and γ_i are the regression coefficients for functional block i; T_i is the temperature of functional block i and T_f is the reference temperature for Taylor expansion.

Embedded CPUs involve concerns about cost and area, and are usually equipped with a few thermal sensors on a CPU core. Therefore, we must choose only a few sensors, say S from K where $S \ll K$, and still effectively model the leakage power. We thus build a regression model (3) by using temperature information from only S sensors to approximate the leakage power of the CPU.

$$P_s^c \approx \sum_{i=1}^{S} (\alpha_i + \beta_i \times (T_i - T_f) + \gamma_i \times (T_i - T_f)^2) \times V_d \quad (3)$$

This is a feature selection problem which is NP-hard and hence not feasible for a brute-force solution. The goal is to establish an approximation function with S variables that can effectively represent the original function with a total of K parameters. Out of a number of metaheuristics methods for feature selection, we leveraged two common solutions. The first is called sequential forward selection (SFS) [18]. However, an overfitting problem may occur if an insufficient training dataset is provided. Therefore, we consider a second method called correlation-based feature selection (CFS) [17].

Our experimental results showed that a single thermal sensor placed at the proper location could reduce the estimation error considerably. However, we are not suggesting that the sensor be removed from the original location that is used for overheating prevention, but demonstrating that accurate leakage power estimation is feasible through the choice of an appropriate sensor location. We discuss the results in Section V.

IV. Temperature-Referred And Counter-Tracked Estimation

The idea behind the proposed TRACE estimation approach is to build a regression model based on relevant parameters such as ambient temperature, dynamic power, and the leakage power of the last timestamp to estimate the leakage power of the current timestamp of a CPU. The dynamic power is tracked by the hardware activity counters and the ambient temperature is calculated with reference to the on-chip thermal sensor. We learned the relationship between these parameters from an RC thermal model. The relationship can be represented as:

$$P^c[t] = C_c \frac{d(T_c[t] - T_a[t])}{dt} + \frac{T_c[t] - T_a[t]}{R_{ca}} \quad (4)$$

where C_c is the thermal capacitance of a CPU, R_{ca} represents the thermal resistance between the CPU and the ambient air, $T_c[t]$ is the temperature of the CPU at time t, $T_a[t]$ is the ambient temperature at time t, and $P^c[t] = P_s^c[t] + P_d^c[t]$ is the total power consumption including the leakage power and dynamic power of the CPU at time t. We can further discretize (4) with a fixed sample period Δt, and assume that the ambient temperature is the same during a short Δt as follows:

$$T_c[t + \Delta t] = A_s T_c[t] + B_s T_a[t] + C_s P^c[t] \quad (5)$$

where $A_s = 1 + \frac{\Delta t}{R_{ca} C_c}$, $B_s = 1 + \frac{R_{ca} C_c}{\Delta t}$, and $C_s = -\frac{R_{ca} C_c + \Delta t}{R_{ca} \Delta t}$. Assuming that there is only one thermal sensor on the CPU, (3) can be reduced to: $P_s^c[t] = \alpha + \beta \times (T_c[t] - T_f) + \gamma \times (T_c[t] - T_f)^2$ at time t. We can derive $T_c[t]$ as:

$$T_c[t] = \frac{P_s^c[t + \Delta t] - P_s^c[t] - \beta \Delta t + 2\gamma T_f \Delta t}{2\gamma \Delta t} \quad (6)$$

We assume a short sampling period Δt, i.e. 10 ms in our setting, and the leakage power does not change excessively within a sampling period. We can make this approximation: $\frac{dP_s^c[t]}{dt} \approx \frac{dP_s^c[t + \Delta t]}{dt} \approx \frac{P_s^c[t + \Delta t] - P_s^c[t]}{\Delta t}$. By combining (5) and (6), we can obtain (7):

$$P_s^c[t + \Delta t] = a P_s^c[t] + b P_d^c[t] + c T_a[t] + d \quad (7)$$

where a, b, c, and d are coefficients that can be obtained by the regression process. Equation (7) reveals that to estimate the leakage power of a CPU at time $t + \Delta t$, we require information about the leakage power, dynamic power of the CPU, and the ambient temperature at time t.

In the initial state (i.e., $t = 0$), we can assume $P_s^c[0] = 0$. To derive the dynamic power of a CPU, $P_d^c[t]$, we rely on hardware activity counters. As mentioned in [13], a linear regression model that accounts for hardware activity counters can accurately estimate the total dynamic power of a CPU. To estimate the ambient temperature, we count on RC thermal

model to model the ambient and the functional block where the sensor is located. Hence, we can understand the relationship among the power generated by the hot spot, the temperature of the hot spot, and the ambient temperature.

V. Experimental Setup and Evaluations

For the simulation environment, we used the Gem5 cycle-accurate simulator [14] to generate hardware activity traces for each functional block of a CPU, and then forwarded the results to McPAT [15] to produce the dynamic power traces. The dynamic power traces were further fed into HotSpot [4] to simulate the temperatures and leakage power. We shrank the process technology of the referred CPU, Alpha EV6 [16], to 70nm CMOS production. We used 8 benchmarks collected from CPU2006 [10] and MiBench.

For the environment of the real platform, we evaluated the Samsung Exynos 5422 on ODROID-XU3. To monitor the detailed thermal distribution of the CPU in real-time, we leveraged an infrared camera from FLIR. We also relied on a chamber from FIRSTEK to control the ambient temperature. The overall experimental environment is shown in Fig. 2. On the real platform, since there is no way to gather the leakage power of the CPU, we used an approximated and indirect model. We constructed a theoretical model comparable to (3) by running negligible tasks and changing the ambient temperature of the chamber. In this way, the spatial temperature variance is less than ± 2ºC, which is negligible. Because the ODROID-XU3 has a current sensor, we consider that the reading of the current sensor is all leakage power under this experimental setting.

A. Placement of Thermal Sensors

In the simulations, we had 30 functional blocks ($K = 30$) on an Alpha EV6 CPU. After running the eight benchmarks under four different ambient temperatures, we obtained 8,810,352 data points in total. Because the sensor-based power model is time independent, the datasets can be regarded as independent data samples. Therefore, for our experiments of sensor selection, we divided the whole dataset into the training set and testing set in an arbitrary ratio of 7:3.

Fig. 3 illustrates the relationship between the accuracy of the leakage power estimation and the number of thermal sensors that SFS and CFS considered. The results show that the estimates based on a single thermal sensor could achieve 99.5% accuracy if we placed the sensor in an appropriate location. If we added more thermal sensors to the CPU, both models could increase

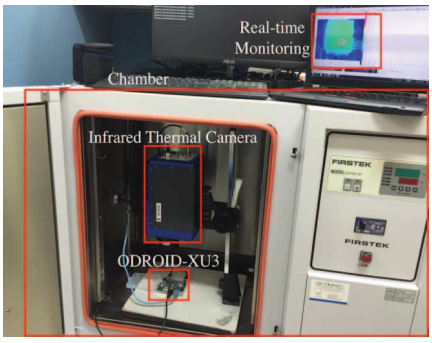

Fig. 2. Experimental environment of the real platform.

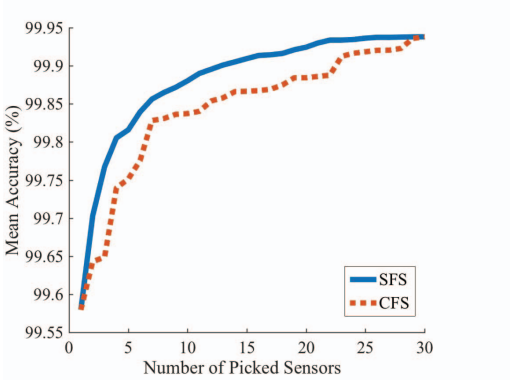

Fig. 3. The number of sensors and the accuracy of models under two different feature selection algorithms.

the accuracy of the leakage power estimation. Notice that the accuracy here is averaged across testing samples rather than benchmark based. The simulation results indicate that a single sensor in an appropriate location is sufficient for accurate leakage power modeling.

We further analyzed the estimation errors of the leakage power model based on a single thermal sensor on H-location, hot spot location, and R-location, representative location selected by both SFS and CFS as the first candidate. According to the results of the H-location model, we categorized the eight benchmarks into the three types and only showed one example per type. In Fig. 4, bitcount H/lbm H/bzip2 H are the results of the H-location model and the results of the R-location model are bitcount R/lbm R/bzip2 R. The first type of benchmark includes only bitcount, and the estimation error of the H-location model is extremely high. This is because it stresses the functional block at H-location, which results in overestimating. The second type of benchmark includes lbm and patricia. This type of benchmark stresses positions other than H-location; therefore, the estimation error based on H-location is reduced. The third type of benchmark (bzip2, fft and dijkstra) stresses both the functional block on H-location and other locations, such that the error is between the first and second type.

B. TRACE

Since TRACE is a time-dependent model, we must use the entire set of benchmark traces to train the model. The training set benchmarks we used were bitcount, hmmer, qsort, and lbm. According to our experiments, TRACE can reduce the average

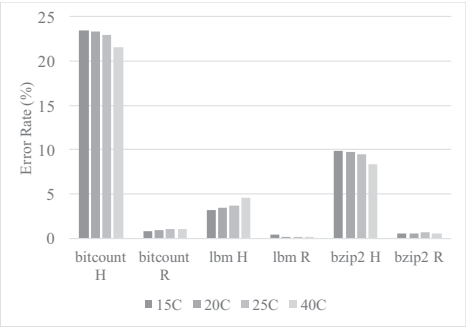

Fig. 4. The mean error rates of the leakage power estimates based on the R-location and H-location.

estimation error of the eight benchmarks from 14.6% to 2.2% compared to the conventional leakage power estimates based on the H-location model. The results are shown in Fig. 5. We found that TRACE's inaccuracies come from ambient temperature estimation, not dynamic power model, because the latter one has an average error rate of merely 0.2%. We leave the improvement of ambient temperature estimation to future research.

C. Verification on Real Platform

To verify the effectiveness of the sensor selection method, we used microbenchmarks to identify the location of the CPU core and placed 12 virtual thermal sensors evenly on the core. The benchmarks we used for the real platform include bwaves, bzip2, calculix, dealII, hmmer, and lbm. The leakage power estimation based on the H-location sensor introduces an average error of 8.4% under the six benchmarks, whereas the estimation based on the R-location sensor introduces only 0.4%. A complete verification of TRACE on real platforms is planned for a future article.

VI. Conclusions

In this study, we demonstrated that the conventional leakage power estimation may be erroneous for embedded CPUs suffering spatial temperature variances. We discussed the numbers and locations of the thermal sensors on a CPU and their impacts on the sensor-based leakage power model. Two feature selection algorithms were evaluated and the results reveal that a power model with the thermal sensor in an optimal location (the instruction cache, in our simulations) can substantially improve the accuracy of the estimation compared with the model relying on hot-spot sensors. Moreover, we proposed TRACE without additional hardware resources. Simulation results show that the TRACE can reduce the average estimation errors of all conducted benchmarks to less than 2.2%.

Acknowledgment

The authors would like to thank Ministry of Science and Technology of the Republic of China (Taiwan) for financially supporting this research under contract No. 105-3011-E-009 -001 -, 103-2221-E-009 -205 -MY3, and 105-2218-E-194 -006 -.

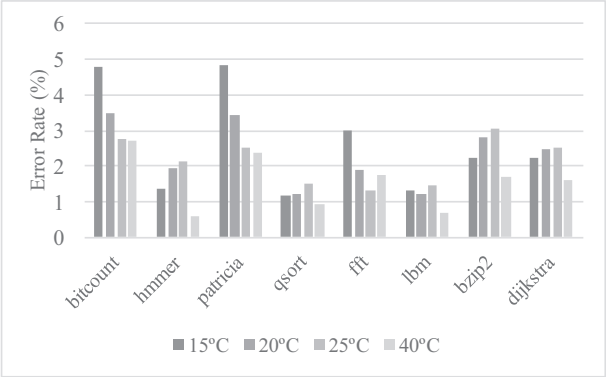

Fig. 5. Error rates of TRACE under different ambient temperatures and benchmarks.

References

[1] G. Singla, G. Kaur, A. K. Unver, and U. Y. Ogras, "Predictive dynamic thermal and power management for heterogeneous mobile platforms," Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (pp. 960-965). EDA Consortium.

[2] N. S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J. S. Hu, et al., "Leakage current: Moore's law meets leakage power," computer, 36(12), 68-75.

[3] M. Zapater, O. Tuncer, J. L. Ayala, J. M. Moya, K. Vaidyanathan, K. Gross, and A. K. Coskun, "Leakage-aware cooling management for improving server energy efficiency," Parallel and Distributed Systems, IEEE Transactions on, 26(10), 2764-2777.

[4] K. Skadron, M. R. Stan, R. J. Ribando, S. Gurumurthi, W. Huang, K. Sankaranarayanan, and G. Link, "Hotspot 5.0."

[5] R. Mukherjee, S. Mondal, and S. O. Memik, S., "Thermal sensor allocation and placement for reconfigurable systems," Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design (pp. 437-442). ACM.

[6] A. Joseph, A. Haridass, C. Lefurgy, S. Rachamalla, S. Pai, D. Chinnakkonda, and V. Goyal, "FirmLeak: A framework for efficient and accurate runtime estimation of leakage power by firmware," VLSI Design (VLSID), 2015 28th International Conference on (pp. 464-469). IEEE.

[7] P. C. Monferrer, G. Magklis, J. Gonzalez, A. Gonzalez, "Leakage power estimation," U.S. Patent No. 7,814,339, Washington, DC: U.S. Patent and Trademark Office.

[8] A. Ibrahim, A. Dwarakanath, and D. P. Shimizu, "Realtime power management of integrated circuits," U.S. Patent No. 8,527,794. Washington, DC: U.S. Patent and Trademark Office.

[9] J. C. Salinas-Hilburg and M. Zapater, "Unsupervised power modeling of co-allocated workloads for energy efficiency in data centers," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 1345-1350). IEEE.

[10] J. L. Henning, "SPEC CPU2006 benchmark descriptions," ACM SIGARCH Computer Architecture News, 34(4), 1-17.

[11] Y. Cheng and C. Hu, "MOSFET modeling & BSIM3 user's guide," Springer Science & Business Media.

[12] Y. Liu, R. P. Dick, L. Shang, and H. Yang, "Accurate temperature-dependent integrated circuit leakage power estimation is easy," Proceedings of the conference on Design, automation and test in Europe (pp. 1526-1531). EDA Consortium..

[13] R. Joseph and M. Martonosi, "Run-time power estimation in high performance microprocessors," Proceedings of the 2001 international symposium on Low power electronics and design (pp. 135-140). ACM.

[14] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, and R. Sen, "The gem5 simulator," ACM SIGARCH Computer Architecture News, 39(2), 1-7.

[15] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures," Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (pp. 469-480). ACM.

[16] R. E. Kessler, "The alpha 21264 microprocessor," Micro, IEEE, 19(2), 24-36.

[17] A. M. Hall, "Correlation-based feature selection for machine learning," (Doctoral dissertation, The University of Waikato).

[18] Whitney, A. Wayne. "A direct method of nonparametric measurement selection." IEEE Transactions on Computers 100.9 (1971): 1100-1103.

[19] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop on (pp. 3-14). IEEE.