The Effect of Temperature on Cache Size Tuning for Low Energy Embedded Systems

Noori, Hamid
Department of Informatics, Graduate School of Information Science and Electrical Engineering, Kyushu University

Goudarzi, Maziar
System LSI Research Center, Kyushu University

Inoue, Koji
Department of Informatics, Graduate School of Information Science and Electrical Engineering, Kyushu University

Murakami, Kazuaki
Department of Informatics, Graduate School of Information Science and Electrical Engineering, Kyushu University

出版情報：Proc. of 17th ACM Great Lakes Symposium on VLSI, pp.453-456, 2007-03-13. ACM Great Lakes Symposium on VLSI
バージョン：
権利関係：

The Effect of Temperature on Cache Size Tuning for Low Energy Embedded Systems

Hamid Noori† Maziar Goudarzi‡ Koji Inoue† Kazuaki Murakami†
†Department of Informatics, Graduate School of Information Science and Electrical Engineering, Kyushu University
‡System LSI Research Center, Kyushu University
3rd Floor Institute of System LSI Design Industry, Fukuoka
3-8-33 Momochihama, Sawara-ku, Fukuoka 814-0001 Japan
+81-92-847-5190
noori@c.csce.kysuhu-u.ac.jp goudarzi@slrc.kyushu-u.ac.jp {inoue, murakami}@i.kyushu-u.ac.jp

ABSTRACT
Energy consumption is a major concern in embedded computing systems. Several studies have shown that cache memories account for about 40% or more of the total energy consumed in these systems. In older technology nodes, active power was the primary contributor to total power dissipation of a CMOS design. However, with the scaling of feature sizes, the share of leakage in total power consumption of digital systems continues to grow. Temperature is a factor which exponentially increases the leakage current. In this paper, we show the effects of temperature on the selection of optimal cache size for low energy embedded systems. Our results show that for a given application, the optimal cache size selection is affected by the temperature. Our experiments have been done for 100nm technology. Our study reveals that the cache size selection for different temperatures depends on the rate at which cache miss increases when reducing the cache size. When the miss rate increases sharply the optimal point is the same for all examined temperatures, however when it becomes smoother, the optimal point for different temperatures begin to get farther.

Categories and Subject Descriptors
B.3.2 [**Memory Structures**]: Design Styles – *cache memories.*

General Terms: Design.

Keywords: Temperature-Aware Design, Cache Memory, Leakage Current, Low Energy, Embedded Systems.

1. INTRODUCTION
Leakage (also known as static) power is becoming the dominant contributor to the power consumption of CMOS integrated circuits. Unlike dynamic power, which depends on the activity in the circuit, leakage exists as long as power supply is applied to the circuit even if there is no activity. Consequently, in general, bigger circuits dissipate higher leakage.

GLSVLSI'07, March 11-13, 2007, Stresa-Lago Maggiore, Italy.

Energy consumption is an important issue for battery powered embedded systems. On-chip cache consumes almost half of a microprocessor's total energy [1]. Energy efficient cache architecture design is thus a critical issue in the design of processor-based embedded systems. Larger cache size may improve the hit rate for some programs, at the expense of more power consumed to fetch (dynamic power) and to hold (static power) the data. Performance-oriented applications benefit from a large cache size. Energy-oriented applications however want the cache size such that the energy savings from higher hit rate outweigh the energy increase from more leakage.

The impact of cache parameters such as cache size, number of ways and line size on energy and performance has been shown in [1][2] [3]. Reliability is another important factor that has been considered besides energy and performance, for cache size selection [4]. In this paper we investigate the effects of temperature as yet another parameter for cache size selection.

Temperature is a parameter that changes for different reasons. The sources of temperature variation can be the environment (location or time) or the system itself (when running applications). Temperature has an exponential effect on the leakage power [5]. Fig. 1 shows the leakage power of a 32K cache for six different temperatures when target technology is 100nm. The data were obtained using CACTI 4.1 [5].

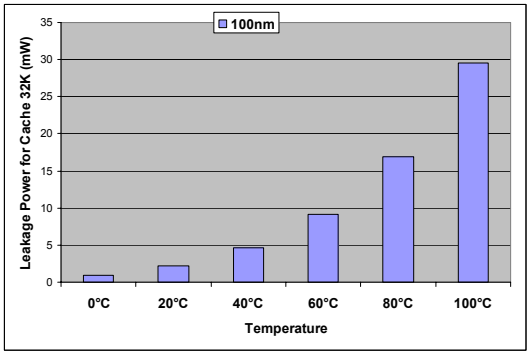

Figure 1. The impact of temperature on the leakage power.

The focus of this paper is mainly to study the effects of different temperatures caused by environment on the cache size selection for minimal energy consumption. We show that when the temperature changes, a new cache size may need to be selected due to the increase of leakage current, if minimum energy consumption is a

must. In other word, our results show that a cache size selected for minimum energy which is optimal size for a cold environment may not work efficient when used in a warmer environment and vice versa.

2. PROBLEM FORMULATION

In this section, first we discuss our energy evaluation model and then define the problem.

2.1 Energy Evaluation

The power consumption of CMOS circuits includes dynamic, static, and short circuit power. The short circuit power consumption is much smaller than dynamic and static power, and is negligible. Dynamic power attributes to signal transition in activated bit- and word-lines, sense-amplifiers, comparators, and selectors during reads/writes, while static power is due to the total amount of leakage current through inactive or OFF transistors. Energy equals power times time. The dynamic energy consumed per access is a sum of the energy spent on searching within the cache, an extra energy required for handling the writes and energy consumed by block replacement on cache miss. Dynamic energy per cache access equals the dynamic power of all the circuits in the cache multiplied by the time per cache access; while the static energy of a cache equals static power multiplied by the time.

Dynamic energy consumption causes most of the total energy dissipation in micrometer-scale technologies and lower temperatures, but static energy dissipation will contribute an increasingly larger portion of total energy dissipation in nanometer-scale technologies and higher temperatures. We consider both types of energies.

Furthermore, we should not disregard energy consumption due to accessing off-chip memory, since fetching instructions and data from off-chip memory is energy expensive because of the high off-chip capacitance and large off-chip memory storage. Additionally, when accessing the off-chip memory, the processor may stall while waiting for the instruction and/or data, and such waiting still consumes some energy. Thus, we calculate the total energy due to memory accesses using Equation 1. In our evaluation the energy is a function of size of cache (S) and temperature (T). Other parameters assumed to be constant.

$$energy\_memory(S,T) = energy\_dynamic(S) + energy\_static(S,T) \quad (1)$$

where

$$energy\_dynamic(S) = cache\_access(S) * energy\_cache\_access(S) + cache\_misses(S) * energy\_miss(S), \quad (2)$$

$$energy\_miss(S) = energy\_off\_chip\_access + energy\_uP\_stall + energy\_cache\_block\_refill(S) \quad (3)$$

$$energy\_static(S,T) = executed\_clock\_cycles(S) * clock\_period * leakage\_power(S,T), \quad (4)$$

cache_access, *cache_misses* and *executed_clock_cycles* are computed by running SimpleScalar [6] for applications with desired cache configuration.

energy_cache_access, *energy_cache_block_refill*, and *leakage_power* are energy for accessing the cache, cache block refilling after a cache miss and leakage power (static power) of a given cache, respectively, which are computed using CACTI [5].

The *energy_off_chip_access* is the energy of accessing off-chip memory when there is a miss, and the *energy_uP_stall* is the energy consumed when the processor is stalled waiting for the memory system to provide missed instruction or data. According to the explanations and experiments done in [1], we assumed:

$$energy\_off\_chip\_access + energy\_uP\_stall = 200 * energy\_cache\_access \quad (5)$$

(for 32K cache size in 100 nm)

2.2 Problem Definition

In the study that we report in this paper, we assume that the cache organization (for both instruction and data cache), the embedded application, the processor and the target technology used are all invariant and only the temperature and total size of the cache is changing. This reflects a scenario where a processor-based embedded system is used in different environments with various temperatures. So in our experiments, the optimization problem is defined as follows: "*For a given application, processor architecture, technology, temperature and instruction- and data-cache organization, find the cache size that results in minimum energy consumption (i.e. minimizes Equation 1 for a given temperature) over the entire application run.*"

3. EXPERIMENTAL RESULTS

3.1 Experiment Setup

We use applications from Mibench [7] benchmark suite. As mentioned, SimpleScalar and CACTI (version 4.1) are used as our simulation tool and power modeling tool, respectively. The cache hit is assumed to take one clock cycle and cache miss 100 cycles. The clock frequency of the single-issue base processor is assumed to be 200 MHz. Our experiments are done for 100nm technology.

3.2 The Effect of Temperature on the Instruction Cache

Fig. 2 shows the number of execution clock cycles for *qsort* for different instruction cache sizes when the size of data cache is fixed. The cache configuration is direct mapped with 16 bytes for cache line size. As it is expected, when the cache becomes smaller the execution time increases, due to increased misses.

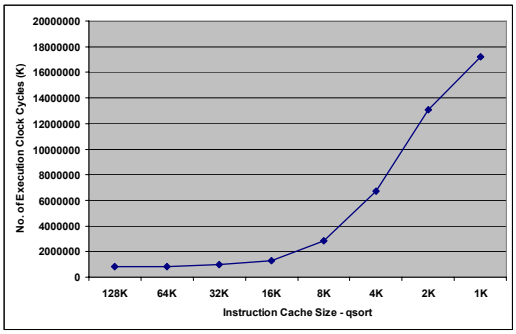

Figure 2. No. of execution clock cycles for different cache sizes.

Fig. 3 and 4 show the dynamic energy and static energy for various temperatures. As the number of misses increases, the dynamic energy grows up as well. In Fig. 4 as the temperature increases the static energy of a given cache increases, so that the static energy of 0°C becomes negligible compared to 100°C. The curves of the static energy for all different temperatures are almost the same and similar

to curve of 100°C but since the magnitudes at lower temperatures are much lower they can not be well distinguished.

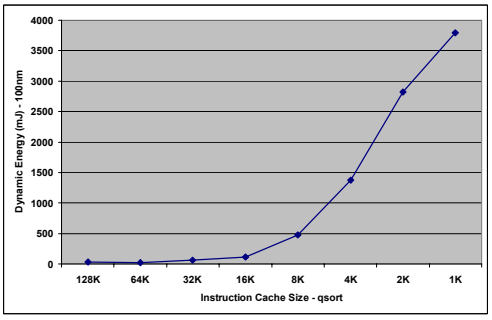

Figure 3. Dynamic energy - instruction cache.

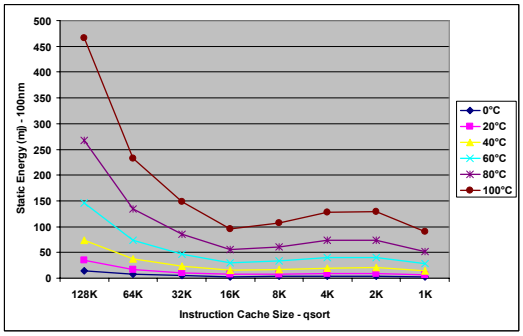

Figure 4. Static energy – instruction cache.

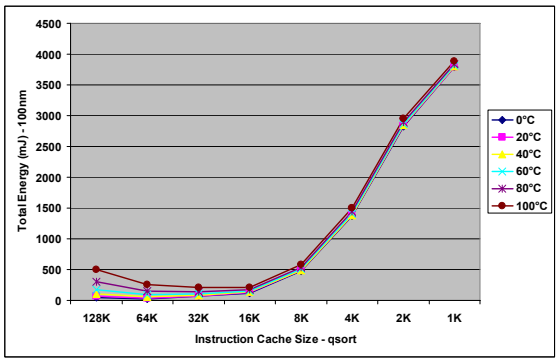

Figure 5. Total energy – instruction cache.

It is interesting to note in Fig. 4 that the static energy does not constantly decrease with lower cache. As the cache size decreases we expect the reduction of the static energy similar to what happens for 128K, 64K, 32K and 16K. However, for 8K we see an increase in static energy. The reason can be understood by looking at Fig. 2. From this figure we learn that for 8K the number of misses increases sharply and consequently the execution time increases. A big increase in execution time causes a big increase in static energy (Equation 4) which can be seen for 8K, 4K and 2K. Since the ratio of increase in misses and static energy for 1K is less than three other previous sizes we see again a decrease in static energy.

Fig. 5 shows the total energy for different cache sizes and temperatures. This figure depicts that for {0°C ~ 80°C} and {80°C ~ 100°C} there are two different points for optimal cache sizes (minimizes Eq. 1) which are 64K and 32K respectively. For {0°C ~ 80°C}, the curve of total energy is similar to dynamic energy, due to the smaller effect of static energy than dynamic energy. In {80°C ~ 100°C} the curve becomes similar to the curve of static energy, due to the impact of static energy which causes an increase for total energy of big caches such as 128K, 64K and 32K. Since the dynamic energy dominates due to huge number of misses and the effect of static energy decreases due to small size of the caches, the curve of total energy becomes similar to curve of dynamic energy, with the cache size smaller than 16K. For 64K cache (optimal size in 0°C), energy consumption increases by 855% when temperature changes from 0°C to 100°C. However, by resizing the cache to 32K (optimal size 100°C) it decreases to 692% which means 17% improvement.

3.3 The Effect of Temperature on the Data Cache

We did similar experiments for data cache for *qsort*. Fig.6 depicts number of execution clock cycles for *qsort* when the size of data cache varies from 512K to 1K while instruction cache size is fixed. The data cache is 2-way set-associative with 16 bytes line size.

Fig. 7 and Fig. 8 show the dynamic energy and static energy for different data cache sizes for *qsort*, respectively. As in Fig. 3, in which the curve is similar to the curve in Fig. 2 (number of execution clock cycles), the curve of dynamic energy in Fig. 7 is also similar to the curve of Fig. 6. However, we do not see similar changes in static energy of data cache (Fig. 8) as we see for instruction cache (Fig. 4). Comparing Fig. 2 and Fig. 6 shows that changes in instruction cache size affects execution time (clock cycles) much more than changes in data cache size. The execution time increases by 2031% when the instruction cache changes from 128K to 1K while for 512K to 1K data cache size, the increase in execution time is around 17%. These big differences in execution time for different instruction cache sizes highly impact the static energy, unlike data cache.

Fig. 9 shows the total cache energy for different data cache sizes. According to the results, 64K, 32K, 16K, 8K, 4K and 2K are optimal data cache sizes for 0°C, 20°C, 40°C, 60°C, 80°C and 100°C, respectively. The curve of total energy for large caches is similar to curve of static energy and for small caches it is getting similar to the curve of dynamic energy. In 100°C, 64K data cache (optimal size for 0°C) consumes 6.5 times more energy compared to 0°C. However, for the optimal cache size at 100°C (2K) the energy consumption is 1.8 times of optimal cache size at 0°C (64K) which shows 72% improvement.

For instruction cache with sharper miss rate there are two optimal sizes while temperature changes from 0°C ~ 100°C, however for the data cache with a miss rate almost 120 times smoother than instruction cache there are six points. From these results we conclude that when the slope of miss rate is very sharp as in Fig. 2, dynamic energy becomes dominant compared to static energy, due to the huge number of off-chip memory accesses and therefore number of optimal points are less and closer to each other for different temperatures. However for data cache, the sharpness in miss rate, becomes much smoother (reduction in off-chip memory access) and again the static energy becomes more effective.

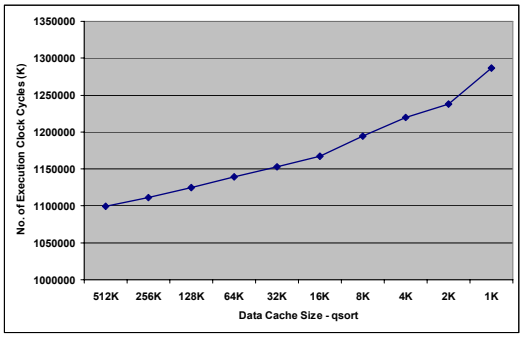

Fig. 6. No. of execution clock cycles for different cache sizes.

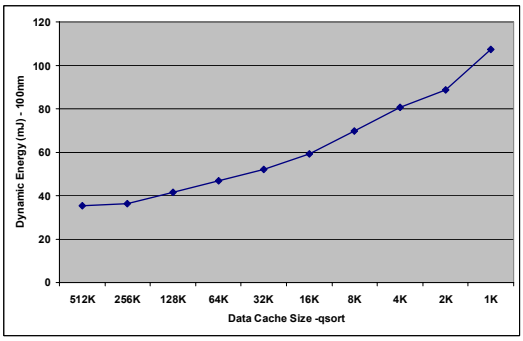

Fig. 7. Dynamic energy for different data cache sizes.

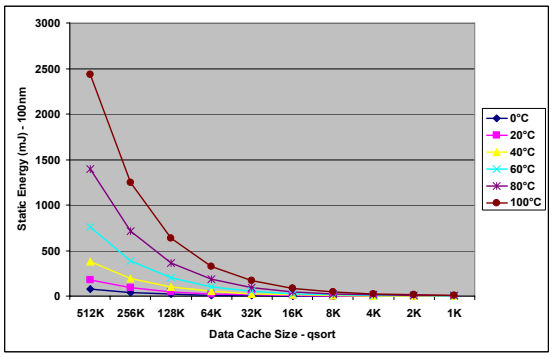

Fig. 8. Static energy for different data cache sizes.

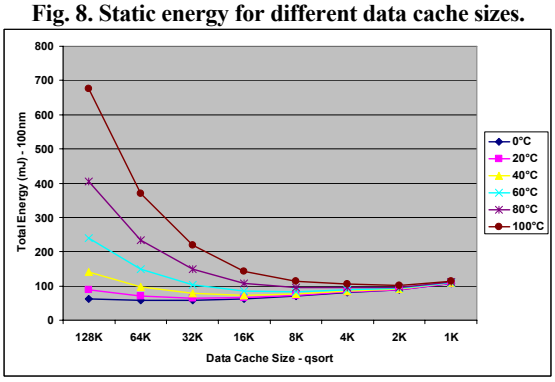

Fig. 9. Total energy for different data cache sizes.

4. CONLCUSIONS

In this paper we study the effects of temperature as another parameter while tuning cache size for different applications targeting low energy embedded computing systems. The results show that for using low energy embedded systems in different environments with various temperatures, the cache may need to be re-tuned. Our study shows that the sharper the slope of miss rate for different cache sizes, the less variation in optimal cache size for different temperatures. So that when the slope is very sharp we can have one optimal point for any temperature and when it is getting softer the distance of optimal cache sizes for different temperatures becomes more. The reason is that when the miss rate is sharp, the dynamic energy becomes dominant, due to increase in off-chip memory accesses, however when it becomes softer the static energy impacts the total energy.

As another point, in all cases the optimal cache size reduces in higher temperatures, although it increases misses and consequently dynamic energy. This is because of the high impact of static energy in higher temperatures.

The results of our study show the need to either keep the temperature fixed for low energy embedded systems with fixed cache size or employ a reconfigurable cache to be temperature-aware tunable in addition to other factors affecting optimal cache size selection.

5. ACKNOWLEDGMENTS

This research was supported in part by the Grant-in-Aid for Creative Basic Research, 14GS0218, Encouragement of Young Scientists (A), 17680005, and the 21st Century COE Program. The second author is supported by Core Research for Evolutional Science and Technology (CREST) project of Japan Science and Technology Corporation (JST). We are grateful for their support.

6. REFERENCES

[1] Zhang C., Vahid F., and Najjar W., A Highly Configurable Cache Architecture for Embedded Systems. *ACM Transactions on Embedded Computing Systems*, Vol. 4, No. 2, May 2005.

[2] Albonesi D. H., Selective cache ways: On-demand cache resource allocation, *32nd Annual ACM/IEEE International Symposium on Microarchitecture*. 1999.

[3] Zhang C., Vahid F. and Lysecky R., A self-Tuning Cache Architecture for Embedded Systems. *Design Automation and Test Europe*, 2004.

[4] Cai Y. et al. Cache Size Selection for Performance, Energy and Reliability of Time-Constrained Systems, *ASP-DAC* 2006.

[5] Tarjan D., Thoziyoor Sh., Jouppi N. P., *Cacti 4.0*, HP Laboratories, Technical Report, 2006.

[6] SimpleScalar http://www.simplescalar.com/

[7] Mibench http://www.eecs.umich.edu/mibench/