A Leakage-Aware Shared L2 Cache Management Scheme for Low-Power Chip Multiprocessors

Hee-Joon Kang

MOBILE HANDSET R&D CENTER
LG Electronics Inc.
Seoul 153-801, Korea
+82-2-2033-7606, kanghj@lge.com

Wonil Choi, and Jihong Kim

School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
+82-2-880-1861, {choi11, jihong}@davinci.snu.ac.kr

Abstract An efficient leakage power management is critical in designing modern low-power CMPs. Since most CMPs employ a large on-chip L2 cache, reducing the leakage power consumption of the L2 cache is an important design goal. Existing cache leakage management techniques (such as the cache decay technique), however, were developed for single-processor systems and do not work well for CMPs with a shared L2 cache because of cache interferences and different access behaviors. In this paper, we propose a shared L2 cache leakage management scheme for low-power CMPs based on a leakage-aware cache partitioning technique and an adaptive task-aware timeout technique. Experimental results using a CMP simulator show that the proposed techniques can save on average 56% of the leakage energy consumption over the existing cache decay technique for the 4-way CMP configuration.

Keywords: leakage reduction, cache partitioning, L2 cache, shared cache, CMP, and adaptive timeout

Introduction Power dissipation is an important design concern in modern microprocessors such as chip multiprocessors (CMPs). As feature sizes further shrink down below 65nm, the leakage power consumption is quickly becoming a dominant power consumer in modern processors, thus reducing leakage power consumption is a critical design requirement for low-power CMPs.

Since most CMPs have a large on-chip shared L2 cache, reducing the leakage power consumption of the shared L2 cache is an important design goal. Existing cache line turn-off techniques (such as the *cache decay* technique [3]) are representative architectural-level leakage management techniques that can be useful for managing the leakage power consumption of the shared L2 cache. However, unlike in single-processors where these techniques are efficient, when applied for CMP systems, their efficiency suffers significantly from the frequent cache interferences among competing tasks for the shared L2 cache. Furthermore, existing cache partition techniques [2][5], which are useful in eliminating the cache interferences, are not suitable for low-power processors because they are difficult to combine with the cache decay technique without incurring a large energy overhead.

In this paper, we propose a novel shared L2 cache leakage management scheme that can be useful for CMPs. We also propose an adaptive per-task timeout management technique that can be better suited for CMPs by using per-task timeout values instead of one global timeout value.

As shown in Fig. 1, we consider CMP systems with a shared L2 cache in this study. We assume that all cache lines can be turned-off after pre-set idle cycles (under the cache decay technique) and the shared L2 cache can be partitioned using a cache partitioning technique such as the column caching technique [1].

Leakage Energy Management Techniques First, we propose a cache hit/sleep time expectation model which can be combined with the cache decay technique in an energy-efficient fashion. Based on these models, we propose a *sleep-aware cache partitioning* policy (SACP) which reduces the leakage energy consumption significantly with a small performance degradation. Second, we propose an adaptive *task-aware timeout management* technique (TATM) that analyzes per-task distributions of cache hit interval lengths and applies per-task timeout values when cache lines are transitioned to the sleep state.

Fig. 2 shows an overview of our leakage management scheme. The proposed scheme consists of S/W and H/W components, which reside in OS and the L2 cache, respectively. The S/W components, which consist of a cache hit/sleep time expectation model, a cache allocation manager, and an adaptive timeout manager, determine the best partition and the best timeout values based on information from a cache event monitor in the L2 cache. The H/W components, which consist of a cache event monitor, a modified cache line turn-off mechanism, and a column caching mechanism, observe the cache access patterns of executing tasks and

This work was supported in part by the Brain Korea 21 Project in 2008 and by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. R0A-2007-000-20116-0). The ICT at Seoul National University provides research facilities for this study.

deliver this information to the S/W components. In addition, they dynamically partition the L2 cache according to an allocation policy decided from the cache allocation manager and turn-off cache lines based on timeout values from the adaptive timeout manager. The cache allocation decisions and timeout values are passed from the S/W components in OS to the H/W components in the L2 cache via two global data structures, a cache way allocation table and a decay timeout table.

Under the SACP policy, cache ways are allocated for tasks in a greedy manner. Based on the cache hit/sleep time expectation model, for each cache way W to be allocated, we compute two expected marginal gains M_{hit} and M_{sleep} of each task τ assuming that the cache way W was allocated to τ. M_{hit} and M_{sleep} represent the changes in the cache hits and cache line sleep time with the extra cache way W, respectively. The cache allocation manager assigns W to the task with the largest weighted sum of M_{hit} and M_{sleep}. We repeat this allocation procedure until all the cache ways are assigned.

Under the TATM technique, the adaptive timeout manager determines timeout values for each task. Based on per-task cumulative distributions of the cache hit interval lengths (which were collected by the cache event monitor), the adaptive timeout manager finds the threshold cache interval length θ such that the predetermined ρ % of cache hit intervals have the length of shorter than θ. (In the experiments below, ρ was set to 90.) We use θ as a per-task timeout value. We update the cache partition and per-task timeout values every five million cycles. Due to space limitations, [7] describes our policies in more detail.

Experiments In order to validate the effectiveness of the proposed scheme, we have performed several experiments using the CATS CMP simulator [4] and SPEC CPU2000 benchmarks [6]. As shown in Tables I and II, we used a 4-way CMP configuration with a shared L2 cache for our experiments. The original cache decay technique (without any other extensions) was used as a base case for comparisons.

Fig. 3 presents the normalized leakage energy consumption of the proposed techniques. Fig. 3(a) shows that HBCP and SACP saved on average 23% and 36% of the leakage energy consumption, respectively. HBCP [5] is the hit-based cache partitioning policy optimized for increasing the total number of cache hits. SACP reduced more leakage energy consumption than HBCP because SACP allocates cache ways to tasks so that cache lines can stay longer in the sleep state even though doing so may increase the number of cache misses. As shown in Fig. 3(b), the combined policy of SACP and TATM saves on average 56% of the leakage energy consumption, which includes the dynamic energy consumption from the additional misses.

Fig. 4 shows cache miss rate changes by the proposed techniques. As shown in Fig. 4, the proposed scheme has a low performance overhead. Except the combined policy of SACP and TATM, our proposed techniques virtually incurred no additional cache misses over the existing cache decay technique. For the combined policy of SACP and TATM, even if it increased the cache miss rate on average by 0.06 over the cache decay technique, it aggressively reduced the leakage energy consumption as shown in Fig. 3(b).

Conclusions Reducing the leakage power consumption of the L2 cache is an important design goal of modern CMPs because of a large on-chip L2 cache. The existing cache leakage management techniques, which have been proposed for single-processor systems, are not efficient when used for a shared cache of CMPs because of different cache interference patterns and access behaviors.

In this paper, we mitigate the effect of cache interferences using the sleep-aware cache partitioning technique which can be employed for reducing the leakage energy consumption for CMPs. The adaptive task-aware timeout management technique also reduced the leakage energy consumption significantly by applying more appropriate timeout values for each task. Our experimental results showed that the proposed techniques can save on average 56% of the leakage energy consumption over the existing cache decay technique for the 4-way CMP configuration.

[1] D. Chiou, P. Jain, L. Rudolph, and S. Devadas, "Application-specific memory management for embedded systems using software-controlled caches," Proc. of DAC, pp. 416-419, 2000.

[2] H. Dybdahl and P. Stenström, "An adaptive shared/private NUCA cache partitioning scheme for chip multiprocessors," Proc. of HPCA, pp. 2-12, 2007.

[3] S. Kaxiras, Z. Hu, and M. Martonosi, "Cache decay: exploiting generational behavior to reduce cache leakage power," Proc. of ISCA, pp. 240-251, 2001.

[4] D. Kim, S. Ha, and R. Gupta, "CATS: cycle accurate transaction-driven simulation with multiple processor simulators," Proc. of DATE, pp. 749-754, 2007.

[5] G. E. Suh, L. Rudolph, and S. Devadas, "Dynamic partitioning of shared cache memory," Journal of Supercomputing, vol. 28, no. 1, pp. 7-26, 2004.

[6] SPEC CPU2000 benchmark, http://www.spec.org/.

[7] H. Kang, W. Choi, and J. Kim, "Leakage Management Schemes", http://davinci.snu.ac.kr/algorithms.pptx.

Table I
Configuration used in simulations

Architectural parameter	Specifications
Number of processors	4
L1 I/D caches	Private, 8KB, 4-way, 64B block
L2 unified cache	Shared, 1MB, 16-way, 64B block
L1 latency	2 cycles
L2 latency	20 cycles
Memory latency	260 cycles

Table II
Power/energy parameters used in simulations

Parameter	Specifications
CMOS manufacturing process	65 nm
L2 tag array leakage power	417.6 mW
L2 cache block leakage power	468 nW
Off-chip memory access energy	101.4 nJ
1bit register transition energy	.05 pJ
1bit register leakage power	1.534 nW

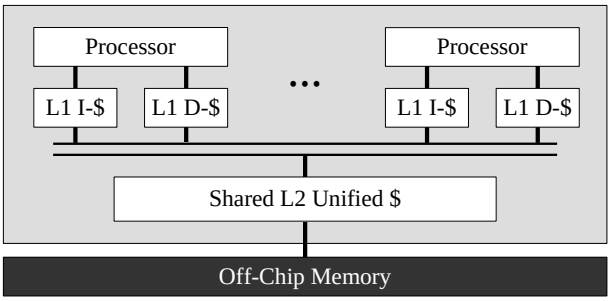


Fig. 1 An overview of target CMPs with a shared L2 unified cache

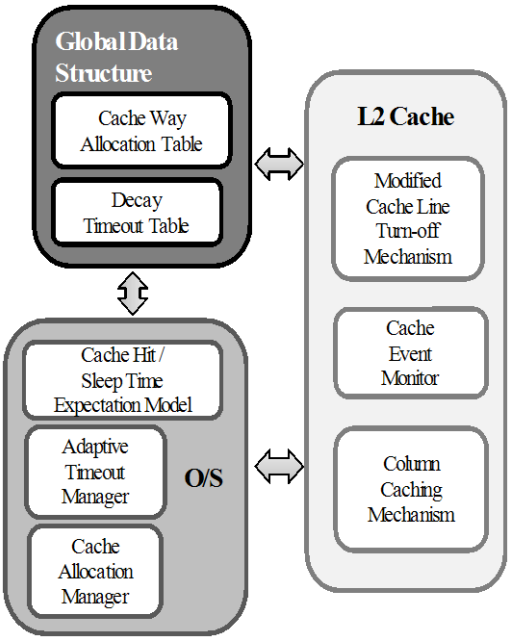
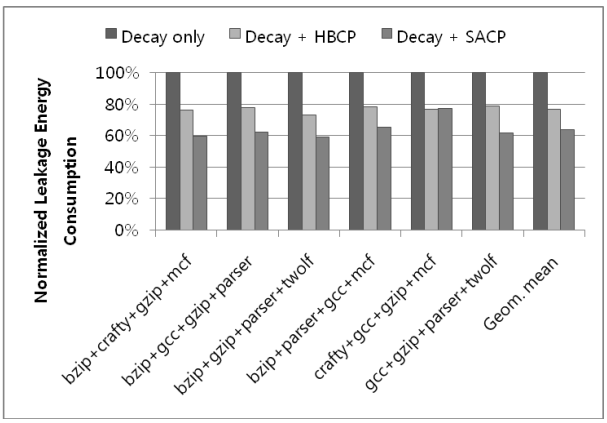
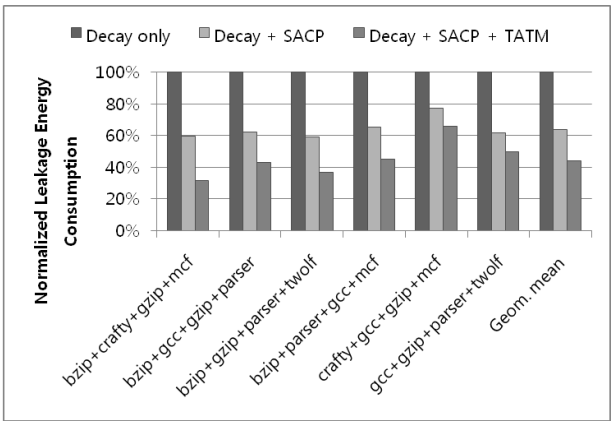


Fig. 2 An overview of the proposed scheme



(a) Comparison of hit-based cache partitioning (HBCP) and sleep-aware cache partitioning (SACP)



(b) Comparison of non-timeout-management and task-aware timeout management (TATM)

Fig 3. Normalized leakage consumption for the benchmarks

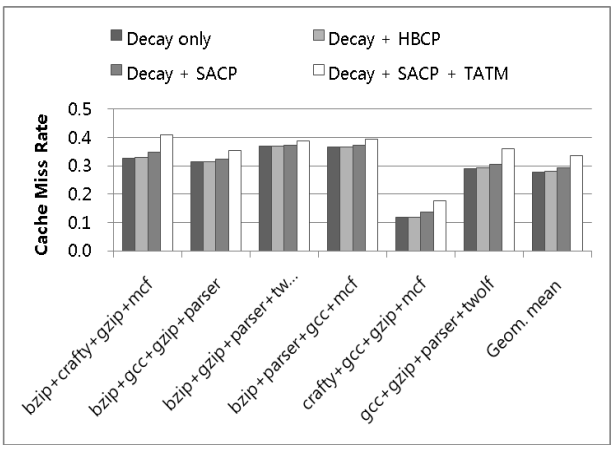


Fig 4. Cache miss rates for the benchmarks