Filtering drowsy instruction cache to achieve better efficiency

Roberto Giorgi
University of Siena - Via Roma, 56
53100 Siena – Italy +39 0577 234630
giorgi@dii.unisi.it

Paolo Bennati
University of Siena - Via Roma, 56
53100 Siena – Italy +39 0577 233601
bennati@dii.unisi.it

ABSTRACT
Leakage power in cache memories represents a sizable fraction of total power consumption, and many techniques have been proposed to reduce it. As a matter of fact, during a fixed period of time, only a small subset of cache lines is used. Previous techniques put unused lines, for example, to drowsy in order to save power.
Our idea is to adaptively select the most used cache lines. In the case of instruction cache, we found that this can automatically achieved by coupling a tiny cache acting as a filter cache (IL0 cache) with a drowsy-cache.
Our experiments, with complete MiBench suite for ARM based processor, show a 25% improvement in leakage saving versus drowsy.

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

General Terms
Performance, Design, Experimentation.

Keywords
Leakage; low-power; filter cache; drowsy cache.

1. INTRODUCTION
A large fraction of the total power consumption in a computing system is due to the power consumed in cache memories and it accounts for about 50% of microprocessor's energy. There are two main factors that cause power consumption in cache memories: *dynamic power* and *static power*. The second one in the 70nm generation constitutes or a large portion of the total power dissipation [1] and recently many projects have tried to reduce it [2, 3]. The common idea they share is to put unused cache lines in a power-saving state that allows to save leakage energy. To reduce the cache activity, the use of a L0 cache (*filter cache*), that is very small relative to the conventional L1 cache placed between CPU and L1 cache, has been proposed in [4]. The filter cache acts as a buffer and caches recently accessed lines. This solution efficiently reduces the dynamic power [4]: therefore, we will not discuss further the issues related to the dynamic power in presence of a filter cache. We focus on leakage-energy budget in presence of a filter cache: this issue has not been studied so far, as our knowledge, for instruction caches.
In this paper, we propose a solution that places a filter cache (IL0) between the processor and a conventional L1 instruction cache with power-saving drowsy-capabilities (IL1). This solution reduces the activity of IL1 because IL0 automatically filters the more recently accessed lines. IL1 cache is a conventional cache (e.g., 16KB) with power-saving capabilities. IL0 cache is a very tiny (e.g., 128B) cache with a latency lower than IL0. Experiments for data caches are discussed in a separate paper [5]. IL0 cache captures the most recently used accesses. Lines can remain off in IL1 since the last used instructions are stored also in IL0 and an hit in the filter cache solves the access.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.
Copyright 2008 ACM 978-1-59593-753-7/08/0003…$5.00.

2. EXPERIMENTS AND RESULTS
We have used the HotLeakage [6], retargeted for ARM based processor and modified to permit the cache configurations that we propose, with a configuration similar to the Intel XScale processor as in [7]. Technology parameters are for a 70 nm process with Vdd=0.9V, as presented in current studies [5]. We simulated the complete suite of MiBench benchmarks [7] for ARM based processor. We mainly compared our proposal (*filtered-drowsy instruction cache*) with conventional drowsy cache (*drowsy instruction cache*) . The size of IL0 we used is 128B (1 cycle latency) while the IL1 is a 16KB 2-way cache (2 cycles latency).
For the leakage evaluation, we take into account the dynamic energy, that each power-saving technique introduces; in other terms, we consider the total leakage, where we account for the contributions from activity in the counters the power-saving techniques use to periodically put lines into low-power state, the leakage of extra circuitry and we consider also the dynamic power of such extra circuitry (as an additional cost to implement the low-power technique). The leakage, for the configuration where L0 is included, accounts for the L0 contribution: the total leakage considered is the sum of the leakage spent in L0 and the leakage spent in L1.
Fig. 1 shows the leakage saving across the full MiBench. In average, the additional saving relative to drowsy-cache is 25%. The IPC behaviour is shown in Fig. 2; performance increases for each benchmark (in average IPC increments of about 1.5%). In Fig. 3 the leakage energy delay (*leakage energy * execution time*) is shown. This metric is particularly useful to understand the global behaviour in order to achieve a high-performance low-power cache hierarchy (the average additional saving is about 24%).

3. ACKNOWLEDGEMENTS
We are particularly grateful to Prof. Sally McKee and her research group at the Cornell University for providing us with a modified version of HotLeakage targeted for ARM ISA. This research is also supported by the European Commission in the context of the HiPEAC Network of Excellence (FP6), contract IST- 004408.

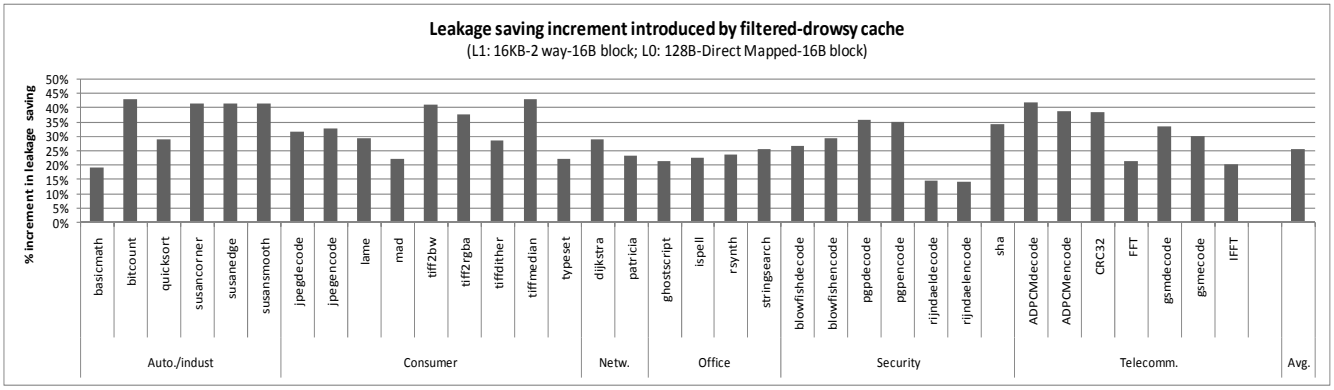

Fig. 1: Increment in leakage-saving across MiBench suite (higher is better): baseline (0%) is the leakage-saving of drowsy instruction cache.

4. REFERENCES

[1] A. Allan, D. Edenfeld, W. H. Joyner Jr, A. B. Kahng, M. Rodgers, and Y. Zorian, "2001 technology roadmap for semiconductors", in *Computer*, vol. 35, pp. 42-53, 2002.

[2] N. S. Kim, K. Flautner, D. Blaauw, and T. Mudge, "Drowsy instruction caches: leakage power reduction using dynamic voltage scaling and cache sub-bank prediction", in *Annual ACM/IEEE International Symposium on Microarchitecture (MICRO'02)*, pp. 1072-4451/02, 2002.

[3] M. Powell, S.-H. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar, "Gated-Vdd: a circuit technique to reduce leakage in deep-submicron cache memories", *International Symposium on Low Power Electronics and Design (ISPLED'00)*, Rapallo, Italy, 2000, pp. 90--95.

[4] J. Kin, M. Gupta, and W. H. Mangione-Smith, "The Filter Cache: An Energy Efficient Memory Structure", *Annual ACM/IEEE International Symposium on Microarchitecture (MICRO'97)*, 1997, pp. 184-193.

[5] R. Giorgi and P. Bennati, "Reducing leakage in power-saving capable caches for embedded systems by using a filter cache", *MEmory performance: DEaling with Applications, systems and architecture Workshop (MEDEA'07)*, Brasov, Romania, 2007, pp. 97--104.

[6] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, "HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects", University of Virginia Tech report, Charlottesville 2003

[7] M. R. a. R. Guthaus, J.S. and Ernst, D. and Austin, T.M. and Mudge, T. and Brown, R.B., "MiBench: A free, commercially representative embedded benchmark suite", *Annual Workshop on Workload Characterization (WWC'01)*, 2001, pp. 83--94.

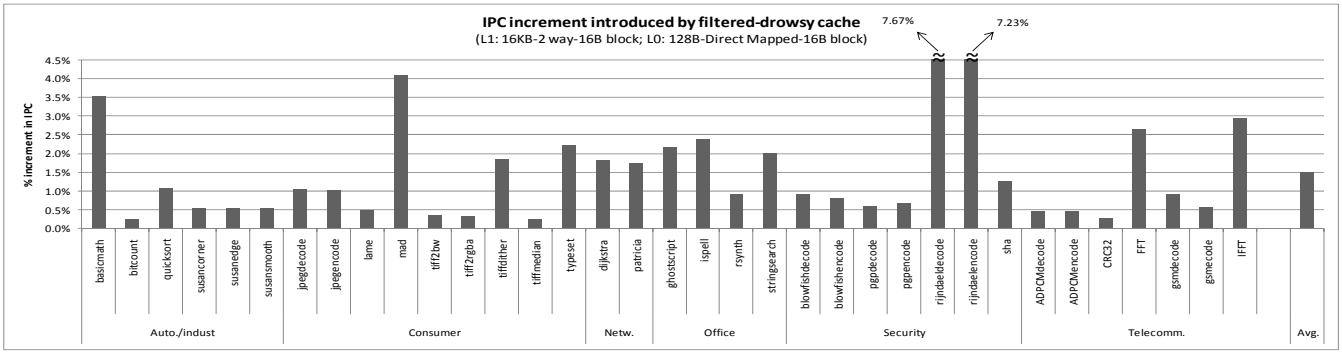

Fig. 2: Increment in IPC across MiBench suite (higher is better): baseline (0%) is the IPC of drowsy instruction cache.

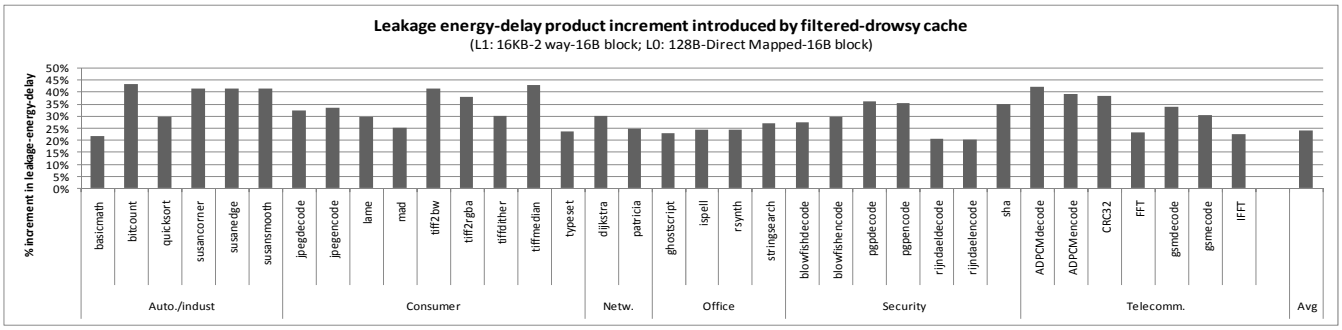

Fig. 3: Increment in leakage energy-delay product across MiBench suite (higher is better): baseline (0%) is the leakage energy-delay product of drowsy instruction cache.