Adaptive cache decay

Paolo Bennati, Roberto Giorgi

Dept. Ingegneria dell'Informazione, University of Siena, Via Roma 56, 53100 Siena, Italy

---

ABSTRACT

Leakage power in data cache memories represents a sizable fraction of total power consumption, and many techniques have been proposed to reduce it. Previous techniques put unused lines for example to drowsy state or switch them off completely (cache decay) in order to save power. Our idea is to adaptively select mostly used cache lines. We found that this can be achieved automatically by using a tiny cache acting as a filter "L0" cache.
Our experiments, with complete MiBench suite for ARM based processor, show 13% improvement in leakage saving and 21% in EDP versus drowsy cache and 52% improvement in leakage saving and 65% in EDP versus cache decay (in average).

KEYWORDS: cache decay; drowsy cache; filter cache; low-power; leakage.

1 Introduction

Recent studies have demonstrated that the power consumption of cache memories accounts for about 50% of the total power consumed in embedded computing systems [1, 2]. Static power has increased in importance in recent CMOS technologies and it can be as much as 50% of total power dissipation.
Many research projects have focused on reducing leakage power in the caches by putting unused lines into a low-leakage state. The proposed techniques can mainly be divided into state-preserving (e.g. *drowsy* [3]) and non-state-preserving ones (e.g. *gated-Vdd* [4]). Both techniques work well if the selection of the lines that will be put in power-saving mode is done accurately. It is important to carefully select which lines to deactivate and when. This is necessary to avoid performance loss while achieving leakage saving. As a matter of fact, during a fixed period of time, only a small subset of cache lines is used [7], so there are lot of opportunities for power saving.
Filter "L0" cache [8] is very small cache placed between L1 and CPU. The filter cache works as a buffer that stores recently accessed cache lines. This approach reduces dramatically the activity of the L1 cache.
In our research, we are investigating the addition of a filter cache to the conventional L1 power-saving caches. By reducing the activity of L1, due to the filtering of the recently accessed lines, the power-saving policies in L1 can be more aggressive, hence they can put more lines in power-saving state. The filter cache doesn't have any power-saving techniques, and provides fast access time. Because of the fact that the filter cache is very small compared to the L1, the additional leakage it introduces is almost negligible.

2 Related Work

It has been demonstrated that leakage power in cache memories is more important than dynamic with current and next generation technologies [2]. Unfortunately the fastest implementation is not always the most beneficial from the energy stand-point [4]. Cache utilization varies widely across a range of applications and it varies significantly also during the execution of a single application [5], so there are a lot of opportunities for switching off cache lines in order to reduce leakage.

Recent mainly consist on a better organization of the cache by reducing its size dynamically. Unused lines can be put into a low-leakage state. When a cache block is put in power-saving state, the technique is called state-preserving if the block content is maintained and non state-preserving if it is destroyed. The main architectural methods of those two categories are drowsy caches [3] for the first one and cache decay [6] for the second one. Cache Decay uses the circuital gated-Vdd technique [7]; it introduces an extra transistor that gates the supply of the cache SRAM cells. A dramatically reduction of the leakage current is achieved, but the loss in performance is not negligible and it causes some increase in dynamic power dissipation. On the other hand, drowsy caches decrease leakage by reducing the power supply without losing information. No additional access to lower memory level is necessary during an access into a drowsy line but the leakage reduction is smaller than in gated-Vdd. A comparison between the two proposal has been done in [8]. After these two techniques have been introduced, many others have suggested their improvements.

Meng et al. [9] explored the limit of leakage power reduction in caches and they found that, with the perfect knowledge of the access pattern it is possible to find the exact moment when to put a line into drowsy state or when to switch it completely off.

A trade-off approach between performance and dynamic energy consumption, has been proposed in [10]. A tiny filter cache is positioned behind the processor and the standard L1 data cache to reduce the performance loss.

3 New cache hierarchy organization

Figure 1 shows the memory hierarchy organization with the L0 cache introduced between the CPU and the upper levels of the hierarchy.

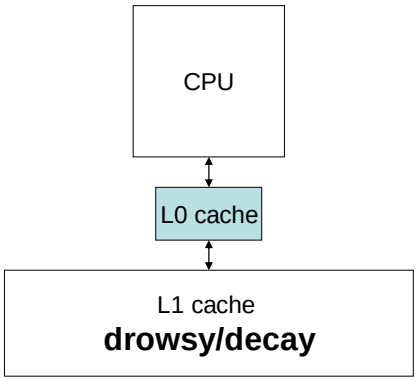

Figure 1. New memory hierarchy organization for leakage reduction.

L0 cache is a very tiny (e.g. 128B) and fast cache. The latency between the L0 cache and the processor is as small as possible as well as the one between L0 and L1. L1 cache is a standard cache (e.g. 64KB) with power-saving capabilities. We have focused on the two most important technique: drowsy cache [3] for the state-preserving category, and cache decay [6] for the non-state-preserving.
The addition of L0 cache has several advantages over other proposed leakage saving techniques. The fact that it filters out significant amount of L1 accesses suggests that the reduction of IPC that occurs in other techniques will be significantly smaller. L0 is a simple and very small cache, so the additional cost of this solution is less than 1% of the L1 cost and it doesn't impact IPC significantly;

4 Simulations

We performed simulation with HotLeakage [10] simulator, retargeted for ARM based processor and modified it in order to permit our configuration. We have used technology parameters values for a 70 nm process at V_{dd}=0.9V. We simulated the complete suite of MiBench [13] benchmarks, compiled for ARM based processor. We compared six different configurations, as shown in Table 1. For each of these configurations, we tested various size for L0 (from 128B direct mapped to 64B fully associative) and L1 (16KB-64KB) as in common ARM xScale (Table 1).
For the leakage evaluation, we took into account all the extra leakage that each low-power technique introduces.

Table 1. Exploration space.

Hierarchy configurations	
L1:	common (no power-saving technique) level 1 data cache with no L0
L0 + L1	common (no power-saving technique) level 1 data cache with L0
L1drowsy	level 1 drowsy data cache without L0 cache
L1decay	level 1 decay data cache without L0 cache
L0 + L1drowsy	level 1 drowsy data cache plus L0 cache
L0+ L1decay	level 1 decay data cache plus L0 cache

Cache configurations		
	L0	L1
Cache size	64B 128B	16KB 64KB
Block size	16B	16B
Associativity	DM - FA	2 - 4
Latency	1 cycle	2 cycles

We analyzed three metrics: leakage, IPC and energy-delay product (in our case leakage*number_of_cycle). Figure 2 shows EDP over MiBench

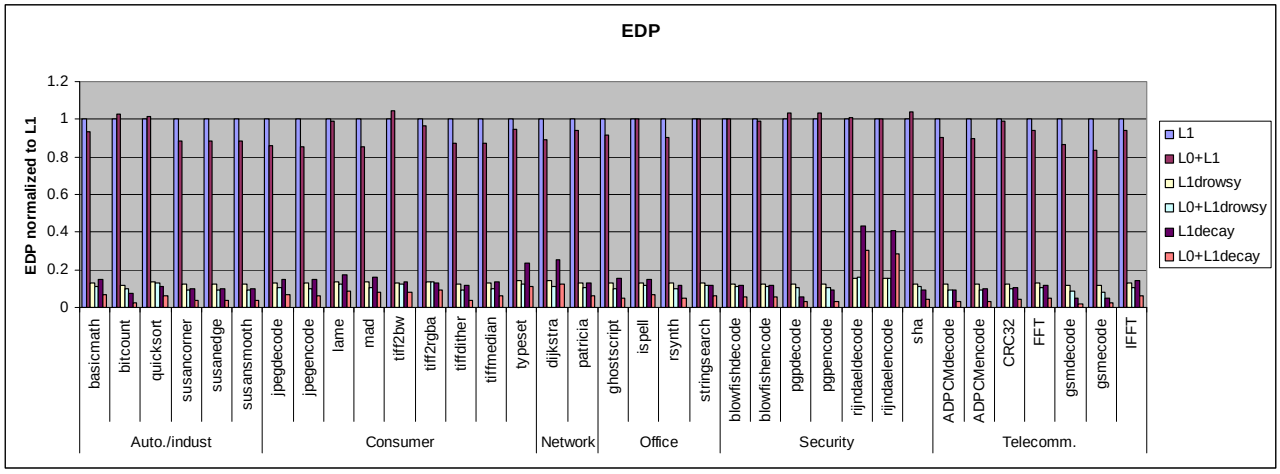

Figure 2. EDP (leakage*sim_cycle). The average values are shown.

On average, our proposal versus drowsy cache increases the leakage-saving of 13% and the IPC of 5%; the EDP is improved of 21%. Versus cache decay these values increase at 52% for the leakage saving, 10% for the IPC and 65% for EDP.

5 Acknowledgment

We are particularly grateful to Sally McKee for allowing us to access to the HotLeakage version retargeted for ARM ISA realized at the Cornell University, Ithaca (USA).

References

[1] N. S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayanan, "Leakage current: Moore's law meets static power," *Computer*, vol. 36, pp. 68-75, 2003.

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

[3] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: simple techniques for reducing leakage power," presented at Annual International Symposium on Computer Architecture (ISCA'02), 2002.

[4] L. Benini, A. Macii, E. Macii, and M. Poncino, "Analysis of Energy Dissipation in the Memory Hierarchy of Embedded Systems: A Case Study," presented at 10th Mediterranean Electrotechnical Conference, MEleCon, Lemesos, Cyprus, 2000.

[5] S. Somogyi, T. F. Wenisch, A. Ailamaki, B. Falsafi, and A. Moshovos, "Spatial Memory Streaming," presented at Proceedings of the 33rd International Symposium on Computer Architecture (ISCA'06)-Volume 00, 2006.

[6] S. Kaxiras, Z. Hu, and M. Martonosi, "Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power," presented at Proceedings of the 28th annual international symposium on Computer architecture, 2001.

[7] 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," presented at 2000 International Symposium on Low Power Electronics and Design (ISPLED'00), Rapallo, Italy, 2000.

[8] Y. Li, D. Parikh, Y. Zhang, K. Sankaranarayanan, M. Stan, and K. Skadron, "State-preserving vs. non-state-preserving leakage control in caches," presented at Design, Automation and Test in Europe Conference and Exhibition (DATE'04), 2004.

[9] Y. Meng, T. Sherwood, and R. Kastner, "Exploring the limits of leakage power reduction in caches," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 2, pp. 221-246, 2005.

[10] J. Kin, M. Gupta, and W. H. Mangione-Smith, "The Filter Cache: An Energy Efficient Memory Structure," presented at 30th annual ACM/IEEE international symposium on Microarchitecture (MICRO'97), 1997.