Performance/Energy Efficiency of Variable Line-Size Caches for Intelligent Memory Systems

Inoue, Koji
Department of Computer Science and Communication Engineering, Kyushu University

Kai, Koji
Institute of Systems & Information Technologies/KYUSHU

Murakami, Kazuaki
Department of Informatics, Kyushu University

出版情報：Lecture Notes in Computer Science. 2107, pp.169-169, 2000-11-12. Springer
バージョン：

Performance/Energy Efficiency of Variable Line-Size Caches on Intelligent Memory Systems

Koji Inoue[1,2], Koji Kai[1], and Kazuaki Murakami[3]

[1] Institute of Systems & Information Technologies/KYUSHU, 2-1-22 Momochihama, Sawara-ku, Fukuoka 814-0001 Japan
[2] Dept. of Computer Science and Comm. Eng., Kyushu University, 6–1 Kasuga-koen, Kasuga, Fukuoka 816-8580 Japan
[3] Dept. of Informatics, Kyushu University, 6–1 Kasuga-koen, Kasuga, Fukuoka 816-8580 Japan

1 Introduction

Integrating main memory (DRAM) and processors into a single chip, or merged DRAM/logic LSI, makes it possible to exploit high on-chip memory bandwidth by widening on-chip bus and on-chip DRAM array. In addition, from energy consumption point of view, the integration brings a significant improvement by decreasing the number of off-chip accesses.

For merged DRAM/logic LSIs having on-chip cache memory, we can exploit the high bandwidth by means of replacing a whole cache line at a time. This approach tends to increase the cache-line size if we attempt to exploit the attainable high bandwidth. A large cache-line size gives a benefit of prefetching effect if programs have rich spatial locality. Otherwise, however, it will bring the following disadvantages due to poor spatial locality:

1. A number of conflict misses will take place due to frequent evictions.
2. As a result, a lot of energy will be wasted for on-chip DRAM (main memory) due to a number of accesses.
3. Activating the wide on-chip bus and the DRAM array will also dissipate a lot of energy.

Employing set-associative caches is a conventional approach to solve the first and second problems, because it can improve cache-hit rates. Since increasing the cache associativity makes cache access time longer, however, it might worsen the memory system performance. In addition, we still have the third problem.

In order to solve all the problems without cache access time overhead, we have proposed *variable line-size cache (VLS cache)* architecture for merged DRAM/logic LSIs [3] [5]. The VLS cache exploits the high bandwidth by means of larger cache lines. At the same time, it can alleviate the negative effects of the larger cache-line size by partitioning it into multiple small cache lines (sublines). Activating only the DRAM subarrays corresponding to the replaced sublines makes a significant energy reduction. In [3] [5], we have discussed the performance only. This paper evaluates both the performance and energy improvements achieved by the VLS caches.

2 Variable Line-Size Cache Architectures

2.1 Concept

In the VLS cache, an SRAM (cache) cell array and a DRAM (main memory) cell array are divided into several subarrays. Data transfer for cache replacements is performed between corresponding SRAM and DRAM subarrays. A block of data associated with a single tag in the cache is referred as *subline*. *Line* is a block of data transferred from cache/main-memory to main-memory/cache for replacements.

Fig. 1 shows the mechanism of variable cache-line size. If programs have rich spatial locality, the line consists of many sublines and a large number of sublines would be involved on cache replacements. Contrarily, a few number of sublines would be replaced when programs have poor spatial locality. In case of Fig. 1, the cache-line sizes of 32-byte, 64-byte, and 128-byte are provided. Activating the DRAM subarrays and the on-chip buses corresponding to the replaced sublines can reduce the energy consumed for accessing to the on-chip main memory.

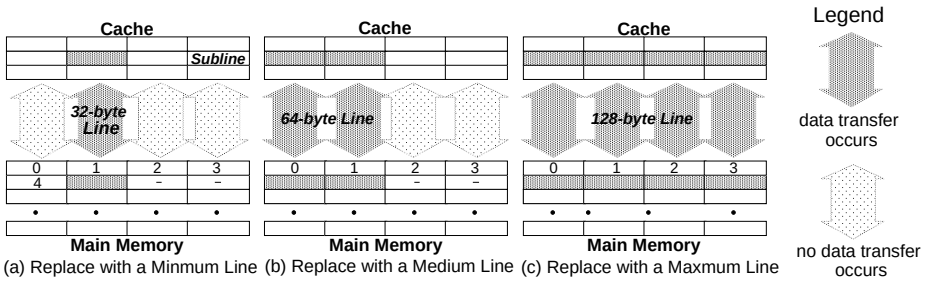

Fig. 1. Mechanism of Variable Cache-Line Size

The effectiveness of the VLS cache depends on how much the cache can choose appropriate line sizes (i.e., the number of sublines to be replaced). There are at least two approaches to the cache-line size determination: one is a static determination based on prior analysis; the other is a dynamic determination using hardware supports.

2.2 Statically Variable Line-Size Cache

The statically variable line-size cache (S-VLS cache) changes its cache-line size program by program. Application programs are analyzed by using cache simulators in advance in order to determine an appropriate cache-line size. In case that the S-VLS cache provides 32-byte, 64-byte, and 128-byte lines, for example, we could determine the appropriate line size in the following manner. First, the program is simulated three times to measure hit rates with fixed line size of

32 bytes, 64 bytes, and 128 bytes. Then we choose the best cache-line size as the appropriate cache-line size. All replacements might be performed under the control of some cache-line-size specifier hardware.

2.3 Dynamically Variable Line-Size Cache

It may be possible to adopt the static approach when target programs have regular access patterns within well-structured loops. However, a number of programs have non-regular access patterns. In addition, the amount of spatial locality may vary both within and among program executions. Against to the static approach explained in section 2.2, the dynamically variable line-size cache (D-VLS cache) selects adequate cache-line sizes based on recently observed data reference behavior at run time. The cache has some hardware components to optimize the cache-line size. The detail of D-VLS cache behavior and an algorithm to optimize the cache-line size have been described in [3].

3 Evaluations

We have evaluated the performance/energy efficiency of on-chip memory systems employing the following conventional and VLS caches:

- Fix128 and Fix128W2 : 16 KB conventional caches having fixed 128-byte cache lines. Fix128 is a direct-mapped cache and Fix128W2 is a 2-way set-associative cache.
- Fix128db : 32 KB conventional direct-mapped cache having fixed 128-byte cache lines.
- SVLS128-32 and DVLS128-32 : 16 KB direct-mapped variable line-size caches having 32-byte, 64-byte, and 128-byte cache lines, based on the static approach (SVLS) and the dynamic approach(DVLS), respectively.

We measured the cache-miss rates and replace counts for each cache using benchmark programs: six integer programs and three floating-point programs from the SPEC95 benchmark suite. Then we calculated "*average memory access time (AMAT)*" as a performance metric. To find the cache access time of each cache, we used the CACTI 2.0 which is the updated version of CACTI model [1]. Since the VLS caches do not have any access time overhead, we assumed that their access times are the same as that of Fix128. Moreover, we calculated "*average memory access energy (AMAE)*" which is the average energy consumption per memory access. The energy consumed for a cache access in each model is based on Kamble's model [4]. In this evaluation, we refer to the cache access time and cache access energy of Fix128 as T_{unit}, and E_{unit}, respectively.

Fig. 2 shows the performance ($AMAT$) of each memory system in case that the access time of on-chip DRAM is six times longer than T_{unit}. Increasing associativity improves miss rates. As the set-associative cache is slow, however, Fix128W2 does not bring good results for some programs. On the other hand, the VLS caches can maintain the fast access of direct mapping. Thus, in all programs

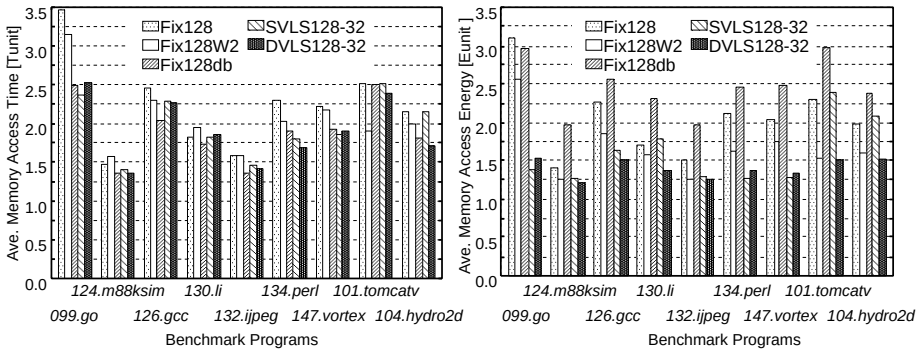

Fig. 2. Average Memory Access Time and Energy

except for *101.tomcatv*, the VLS caches make significant improvements, which are comparable with the doubled size conventional direct-mapped cache (Fix128db).

Fig. 2 also shows energy consumption ($AMAE$) of each memory system in case that the energy consumed for accessing to on-chip DRAM is ten times larger than E_{unit} [2]. In conventional caches, the energy consumed for accessing to the on-chip DRAM depends on only miss rates (i.e., the total number of main memory accesses). While the energy depends not only on miss rates but also on cache-line size to be replaced in the VLS caches. The minimum and maximum average cache-line size on replacements are 42.82 bytes for *099.go* and 89.34 bytes for *104.hydro2d*, respectively. Although the miss rate of VLS caches is higher than that of 2-way set-associative cache (Fix128W2), the VLS caches produce more energy reduction due to DRAM sub-banking effects.

Finally, in Fig. 3, we show the energy-delay product to evaluate the performance and energy consumption at the same time. For each program, all the results are normalized to Fix128. In conventional caches, the performance improvement achieved by increasing cache capacity (Fix128db) is negated by the more energy consumption. Contrarily, energy improvement produced by increasing associativity (Fix128W2) is negated by low-performance due to long cache access time. The VLS caches do not have this kind of negations because they can produce both the performance and energy improvements. In the best case of *099.go*, the VLS caches reduce the energy-delay product more than 65 % from the conventional direct-mapped cache (Fix128). While the improvements of conventional approach of increasing associativity or capacity are only from 25 % to 30 %.

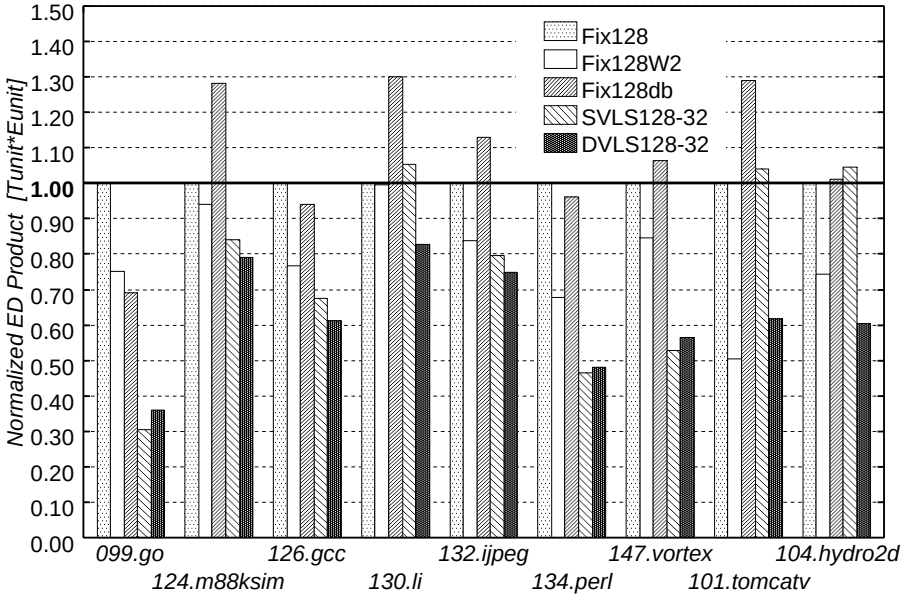

Fig. 3. Energy Delay Products ($AMAT \times AMAE$)

References

1. Jouppi, P., Norman,
   http://www.research.digital.com/wrl/people/jouppi/CACTI.html
2. Fromm, R., Perissakis, S., Cardwell, N., Kozyrakis, C., McGaughy, B., Patterson, D., Anderson, T., and Yelick, K., "The Energy Efficiency of IRAM Architectures," *Proc. of the 24rd Annual International Symposium on Computer Architecture*, pp.327–337, May 1997.
3. Inoue, K., Koji, K., and Murakami, K., "Dynamically Variable Line-Size Cache Exploiting High On-Chip Memory Bandwidth of Merged DRAM/Logic LSIs," *Porc. of the 5th International Symposium on High-Performance Computer Architecture*, pp.218–222, Jan. 1999.
4. Kamble, M. B., and Ghose, K., "Analytical Energy Dissipation Models For Low Power Caches," *Proc. of the 1997 International Symposium on Low Power Electronics and Design*, pp.143–148, Aug. 1997.
5. Murakami, K., Shirakawa, S., and Miyajima, H., "Parallel Processing RAM Chip with 256Mb DRAM and Quad Processors," *1997 ISSCC Digest of Technical Papers*, pp.228–229, Feb 1997.

This article was processed using the LaTeX macro package with LLNCS style