Way-Predicting Set-Associative Cache for High Performance and Low Energy Consumption

Koji Inoue, Tohru Ishihara, and Kazuaki Murakami

Department of Computer Science and Communication Engineering Kyushu University
6-1 Kasuga-Koen, Kasuga, Fukuoka 816-8580 JAPAN
ppram@c.csce.kyushu-u.ac.jp

Abstract

This paper proposes a new approach using way prediction for achieving high performance and low energy consumption of set-associative caches. By accessing only a single cache way predicted, instead of accessing all the ways in a set, the energy consumption can be reduced. This paper shows that the way-predicting set-associative cache improves the ED (energy-delay) product by 60–70% compared to a conventional set-associative cache.

1 Introduction

On-chip cache has been playing an important role in achieving high memory performance and low energy consumption, because it can reduce the number of access to slower and higher energy off-chip next-level memory (usually DRAM). As on-chip cache size has increased, however, the energy dissipated by on-chip caches has become significant.

There have been several proposals for reducing the power consumption of on-chip caches. MDM (Multiple-Divided Module) cache[8] attempts to reduce the power consumption by means of partitioning the cache into several small sub-caches. Block buffering[3][9], filter cache[7], and L-cache[4] achieve low power consumption by adding a very small L0-cache between the processor and the L1-cache. However, these caches require large modifications of cache structure or memory hierarchy. Hasegawa et al.[5] proposed a low-power set-associative cache, which is referred to as phased cache in this paper and detailed in Section 2.2. The phased cache suffers from longer cache-hit time.

Many modern processors employ *set-associative caches* as L1 or L2 caches. Although increasing cache associativity introduces higher hit rate, it makes the cache access time longer due to the delay for way selection. To compensate for this disadvantage, several researchers have proposed way-predictable set-associative caches [1][2][6][10].

In this paper, we attempt to use the way-prediction for achieving not only high performance but also low energy consumption of set-associative caches. The set-associative cache with way-prediction, or *way-predicting cache*, speculatively selects one way before it starts a normal cache access. By accessing only the one way, on the way prediction is accurate, the energy consumption can be reduced without performance degradation compared to accessing all the ways. Since the cache structure and memory hierarchy of conventional memory system is maintained, this approach can be implemented with small hardware overhead.

2 Set-Associative Cache

2.1 Conventional Cache

Total energy consumed for an access to a set-associative cache (E_{Cache}) can be approximated by the sum of following terms [9]:

- E_{Decode}: Energy consumed to drive the address bus and decode the memory address.
- E_{Memory}: Energy consumed to access the tag-subarrays and data-subarrays, mainly to drive word lines and bit lines, and to activate sense amplifiers, and so on.
- $E_{I/O}$: Energy consumed to drive external I/O pins when a cache replacement occurs.

Because E_{Decode} is much smaller than E_{Memory} and $E_{I/O}$ has little influence on caches with high hit rates [3], E_{Cache} can be simplified as follows:

$$E_{Cache} \sim E_{Memory} \quad (1)$$

$$= N_{Tag} \times E_{Tag} + N_{Data} \times E_{Data} \quad (2)$$

- N_{Tag}, N_{Data}: The number of tag-subarrays and data-subarrays accessed, respectively, while a cache access is performed.
- E_{Tag}, E_{Data}: Energy consumed for accessing a tag-subarray and data-subarray, respectively.

Figure 1(A) shows a general organization of a set-associative cache, which consists of four *ways* with a *tag-subarray* and a *data-subarray* for each way. Regardless of a hit or miss, all ways are activated, and the cache access can be completed in one cycle. Accordingly, total energy consumption for an access (E_{Cache}) and the average cache-access time (T_{Cache}) in term of clock cycles of a conventional four-way set-associative cache (4SACache) can be expressed by the following equations:

$$E_{4SACache} = 4E_{Tag} + 4E_{Data} \quad (3)$$

$$T_{4SACache} = 1 \quad (4)$$

2.2 Phased Cache

The energy consumption of set-associative cache tends to be higher than that of direct-mapped cache, because all the ways in a set are accessed in parallel although at most only one way has the desired data. To solve the energy issue, Hasegwa et al. proposed a low-power set-associative cache architecture [5], which is referred to as *phased cache* in this paper. As shown in Figure 1(b), the phased cache divides the cache-access process into the following two phases. First, all the tags in the set are examined in parallel, and no data accesses occur during this phase. Next, if there is a hit, then a data access is performed for the hit way.

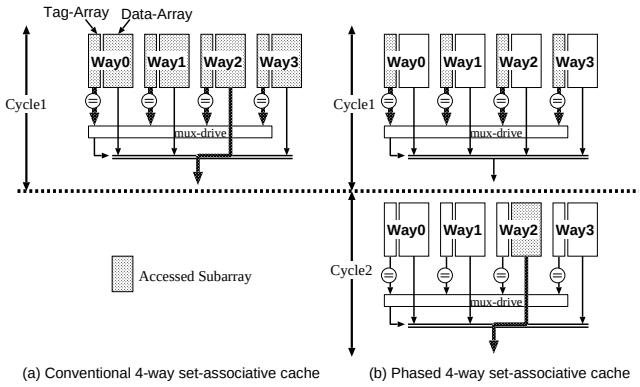

Figure 1: Phased Set-Associative Cache

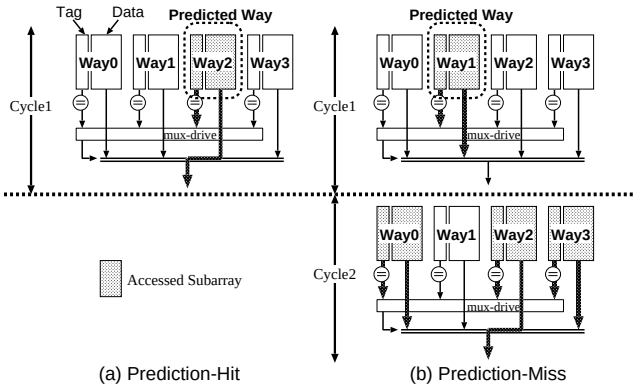

Figure 2: Way-Predicting Set-Associative Cache

The average energy consumption (E_{Cache}) and the average cache-access time (T_{Cache}) for the phased four-way set-associative cache (P4SACache) can be expressed as follows:

$$E_{P4SACache} = 4E_{Tag} + CHR \times E_{Data} \quad (5)$$

$$T_{P4SACache} = 1 + CHR \times 1 \quad (6)$$

Here, CHR is the cache-hit rate.

3 Way-Predicting Set-Associative Cache

We now propose a new set-associative cache architecture, called *way-predicting cache* for lower energy consumption. The way-predicting cache speculatively chooses one way before starting the normal cache-access process, and then accesses the predicted way as shown in Figure 2(a). If the prediction is correct, the cache access has been completed successfully. Otherwise, the cache then searches the other remaining ways as shown in Figure 2(b).

On a prediction-hit, shown in Figure 2(a), the way-predicting cache consumes only energy for activating the predicted way. In addition, the cache access can be completed in one cycle. On prediction-misses (or cache misses), however, the cache-access time of the way-predicting cache increases due to the successive process of two phases as shown in Figure 2(b). Since all the remaining ways are activated in the same manner as a conventional set-associative cache, the way-predicting cache could not reduce energy consumption in this scenario. The performance/energy efficiency of the way-predicting cache largely depends on the accuracy of the way prediction.

In this paper, we have employed a MRU (Most Recently Used) algorithm for the way prediction [1][2][6][10]. In case of a 16 KB four-way set-associative cache, the MRU region is only 4 KB. The MRU information for each set, which is a two-bit flag, is used to speculatively choose one way from the corresponding set. These two-bit flags are stored in a table accessed by the set-index address. Reading the MRU information before starting the cache access might make cache access time longer. However, it can be hidden by calculating the set-index address at an earlier pipe-line stage [1]. In addition, way prediction helps reduce cache access-time due to eliminating of a delay for way selection. So, we assumed that the cache-access time on prediction hit of the way-predicting cache is same as that of conventional set-associative cache.

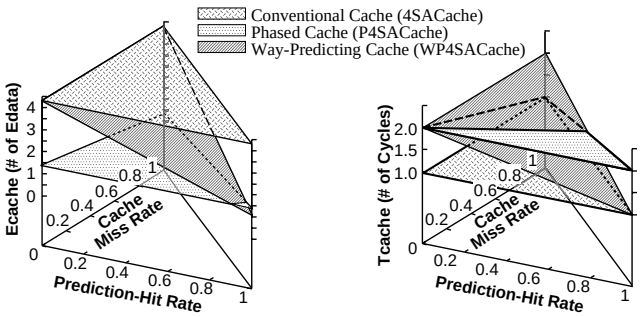

Figure 3: Average Energy Consumption per Cache Access and Average Cache-Access Time

The average energy consumption (E_{Cache}) and the average cache-access time (T_{Cache}) for the way-predicting four-way set-associative cache (WP4SACache) can be expressed as follows:

$$E_{WP4SACache} = (E_{Tag} + E_{Data}) + (1 - PHR) \times (3E_{Tag} + 3E_{Data}) \quad (7)$$

$$T_{WP4SACache} = 1 + (1 - PHR) \times 1 \quad (8)$$

Here, PHR is prediction-hit rate.

4 Evaluation

4.1 Static Analysis

Figure 3 shows the average energy consumption and the average cache-access time based on equations from (3) to (8) for a conventional four-way set-associative cache (4SACache), a phased four-way set-associative cache (P4SACache), and a way-predicting four-way set-associative cache (WP4SACache). For every cache, the cache size, cache-line size, and associativity are 16K bytes, 32 bytes, and 4, respectively. Because the same replacement algorithm (usually LRU) is used for every cache, the cache-hit rate (CHR) is common to all the caches. The address size, set-index size, and byte-offset size are 32 bits, 7 bits ($= \log_2 128$), and 5 bits ($= \log_2 32$), respectively. Thus the tag size is 20 bits. From this calculation, we assumed that $E_{Tag} = 0.078E_{Data}$ because the ratio of the tag size to the cache-line size is 20 : 256 (in terms of bits), or 0.078 : 1.

Table 1: Benchmark Results: Average Energy Consumption (E_{Cache}) and Average cache-access time (T_{Cache})

Benchmarks	I-Cache				D-Cache			
	P4SACache		WP4SACache		P4SACache		WP4SACache	
	T_{Cache}	E_{Cache}	T_{Cache}	E_{Cache}	T_{Cache}	E_{Cache}	T_{Cache}	E_{Cache}
099.go	198.6%	30.1%	105.5%	29.1%	198.8%	30.1%	118.7%	39.0%
124.m88ksim	199.8%	30.4%	104.2%	28.2%	199.1%	30.2%	104.5%	28.4%
126.gcc	197.4%	29.8%	107.7%	30.8%	197.0%	29.7%	112.6%	34.5%
129.compress	200.0%	30.4%	100.0%	25.0%	195.3%	29.3%	108.4%	31.3%
130.li	200.0%	30.4%	102.7%	27.0%	196.7%	29.7%	107.2%	30.4%
132.ijpeg	200.0%	30.4%	100.3%	25.2%	199.0%	30.2%	107.4%	30.6%
134.perl	199.6%	30.3%	105.1%	28.8%	198.4%	30.1%	107.4%	30.5%
147.vortex	198.8%	30.2%	108.4%	31.3%	198.5%	30.1%	110.6%	33.0%
101.tomcatv	198.9%	30.2%	108.4%	31.3%	197.9%	29.9%	112.0%	34.0%
102.swim	200.0%	30.4%	102.0%	26.5%	182.0%	26.3%	149.7%	62.3%
103.su2cor	199.7%	30.4%	103.5%	27.6%	193.4%	28.9%	106.6%	36.1%
104.hydro2d	199.7%	30.4%	101.7%	26.3%	193.0%	28.8%	110.6%	32.9%

Figure 3 plots the average energy consumption per cache access and the cache-access time as a function of the prediction-hit rate (PHR) and the cache-miss rate ($CMR = 1 - CHR$) for each cache (4SACache, P4SACache, and WP4SACache). When $PHR = 100\%$ (i.e., $CHR = 100\%$), the way-predicting cache performs best. Compared to the conventional set-associative cache, the average energy consumption is reduced by 75% without any performance degradation. Compared to the phased cache, the average energy consumption and the average cache-access time are reduced by 18% and 50%, respectively. On the other hand, when $PHR = 0\%$, the way-predicting cache performs worst even if $CHR = 100\%$. Compared to the conventional set-associative cache, the average cache-access time increases by 100% while the average energy consumption is unchanged. Compared to the phased cache, the average energy consumption is greater by 229% and the cache-access time is the same.

4.2 Experimental Analysis

We made some experiments using a cache simulator. The cache simulator gets an address trace as its input, and simulates the LRU cache replacement algorithm and the MRU way-prediction algorithm. And then, the cache simulator reports the prediction-hit rate (PHR), prediction-miss rate (PMR), and cache-miss rate (CMR) as its outputs. All benchmark programs were compiled by GNU CC (–O2) for the UltraSPARC.

For the I-cache, all the programs achieve quite high prediction-hit rates (PHR) of over 90%. For the D-cache, more than half of the programs also achieve high prediction-hit rates (PHR) of over 90%. The average PHR for I-cache and D-cache are about 96% and 86%, respectively.

Based on the models of energy consumption and cache-access time expressed by equations (3) to (8), and on the benchmark simulation results, Table 1 shows the average energy consumption per cache access and the average cache-access time for the phased four-way set-associative cache (P4SACache) and the way-predicting four-way set-associative cache (WP4SACache). All the results are reported as relative numbers which are normalized to the results of the conventional four-way set-associative cache (4SACache).

Compared to the conventional cache, for most of the programs, the phased cache reduces the average energy consumption by about 70%, but it increases the average cache-access time by about 100%. On the other hand, for most of the programs, the way-predicting cache achieves mostly the same energy reduction as the phased cache. At the same time, the average cache-access time of the conventional cache is maintained. In average, the way-predicting cache improves the mean ED product (=average energy consumption per cache access × average cache-access time) by about 70% and 60% with the I-cache and the D-cache, respectively, compared with the conventional set-associative cache.

5 Conclusions

In this paper, the way-predicting set-associative cache for low energy consumption has been proposed. The way-predicting cache speculatively selects one way from the set, before beginning a normal cache access. By accessing only the one way predicted, instead of accessing all the ways, the energy consumption can be reduced. The experimental results show that the way-predicting cache improves the ED product by 60–70% over the conventional set-associative cache.

References

[1] Brad, C., Dirk, G., and Joel, E., "predictive Sequential Associative Cache," *Proc. of the 2nd International Symposium on High-Performance Computer Architecture*, pp244–253, Feb. 1996.

[2] Chang, J. H, Chao, H., and So, K., "Cache Design of A Sub-Micron CMOS System370," *Proc. of the 14th International Symposium on Computer Architecture*, pp208–213, June 1987.

[3] Ghose, K., and Kamble, M. B, "Energy Efficient Cache Organizations for Superscalar Processors," *Power-Driven Microarchitecture Workshop In Conjunction With ISCA98 in Barcelona*, http://www.cs.colorado.edu/ grunwald/LowPowerWorkshop/, June 1998.

[4] Haji, N. B. I., Polychronopoulos, C., and Stamoulis, G., "Architectural and Compiler Support for Energy Reduction in the Memory Hierarchy of High Performance Microprocessors," *Proc. of the 1998 International Symposium on Low Power Electronics and Design*, pp.70–75, Aug. 1998.

[5] Hasegawa, A., et al.,"SH3: High Code Density, Low Power," *IEEE Micro*, pp.11–19, Dec. 1995.

[6] Kessler, R. E, Jooss, R., Lebeck, A., and Hill, M. D, " Inexpensive Implementations of Set-Associativity," *Proc. of the 16th International Symposium on Computer Architecture*, pp131–139, 1989.

[7] Kin, J., Gupta, M., and Mangione-Smith, W. H., "The Filter Cache: An Energy Efficient Memory Structure," *Proc. of the 30th Annual International Symposium on Microarchitecture*, pp.184–193, Dec. 1997.

[8] Ko, U., Balsara, P. T., and Nanda, A. K, "Energy Optimization of Multi-Level Processor Cache Architecture," *Proc. of the 1995 International Symposium on Low Power Design*, pp.45–49, Apr. 1995.

[9] Su, C. L., and Despain, A. M, "Cache Design Trade-offs for Power and Performance Optimization:A Case Study," *Proc. of the 1995 International Symposium on Low Power Design*, pp.69–74, Apr. 1995.

[10] Yeager, K. C., "The Mips R10000 Superscalar Microprocessor," *IEEE Micro*, Vol. 16, Num. 2, pp.28–40, Apr. 1996.