A Tag Cache Architecture for Two-Type Data Cache Model

S. Subha
SITE,
Vellore Institute of Technology,
Vellore, India

ABSTRACT

For the two level two type data cache model proposed in literature, the two cache levels are accessed for each reference to determine the cache is to be inclusive or exclusive. This is achieved by probing the index array of the cache levels. This paper proposes tag cache architecture for the two type data cache model. The tag array at level one is accessed to determine if a line is present in level one or level two by comparing the tag values. The cache levels are accessed based on the result of tag comparison. The tag array is enabled during the entire operation selectively enabling the cache lines in two levels. Energy consumption is reduced by this operation. A mathematical model for energy saving is developed and validated with SPEC2K benchmark with 99% energy saving.

General Terms

Computer Architecture, Cache Memory, Energy Saving

Keywords

tag cache, two-type data cache, energy savings

1. INTRODUCTION

There are three kinds of cache: direct mapped, set associative and fully associative [3]. Blocks are placed in fixed locations in direct mapped cache, while they can occupy any location in a fully associative cache. In set associative cache, a block can occupy a subset of locations in the cache. Address translation places a block in the cache. Blocks are placed according to a replacement policy whenever there is a conflict. Some caches have a property in which the blocks in any level of the cache are also present in higher levels of the cache. This property is called inclusive property, and such a cache is called an inclusive cache. The author in [1] discusses the various issues in cache memories.

Jouppi [7] proposed a cache model where the blocks in lower levels of the cache are not present in higher levels of the cache. This property is called the exclusive property, and such caches are known as exclusive caches. The authors in [2] discuss analytical model for cache. In their paper Ying Zing et al [12] study performance of exclusive cache hierarchies. The author in [9] proposed two type data cache architecture. In this model the cache ways are inclusive or exclusive based on reuse of the cache line. Initially the levels L1 and L2 are exclusive in nature. On the first occurrence of an address, the data is placed in one of the levels only making that way of the cache exclusive. The data is placed in level one if there is place else it is placed in level two if it is free. If the data is accessed for consecutive time in level one cache i.e. the second time, the cache way is made inclusive in nature i.e. its copy resides in both level one and level two of the cache. During this process, the data in level two may be replaced. The Least Recently Used (LRU) algorithm is used to replace the data in the level two cache. In this, the line which is the least recently used is replaced with the incoming line. If the data is reused in level two cache it is made exclusive in nature. On a cold miss with both the levels occupied, the block is placed in level one cache making it exclusive. As the number of ways to place the data increases, the performance increases in terms of the average memory access time (AMAT). Conditions for achieving performance improvement in the proposed model over the inclusive and exclusive models are derived in this paper.

The energy consumed in the cache system depends on the number of ways enabled during cache operation. The authors in [4] state that duplication of data is expensive from leakage perspective. They propose a strategy to reduce energy consumption in level two cache. The authors in [6] propose a method to reduce energy consumption by way-prediction and direct mapping. The author in [10] proposes tag cache model for exclusive cache. In the model proposed in [9], the two cache levels are enabled during the entire operation. This can be reduced to operating the two cache sets in the two cache levels. If $w_1 and w_2$ are the number of ways in level one and level two caches, E is the energy consumed per cache way, for R references in the address trace, the total energy consumed is $\left(w_1 + w_2\right)RE$.

This paper proposes tag cache architecture for the two type data cache model. A tag cache consisting of the tag values of cache ways at all levels is placed in level one. The address is checked for hit in the cache levels by probing into the tag cache. The algorithm for accessing cache levels as proposed in [9] is implemented based on the tag comparison in the tag array. In this model, the entries of the tag cache are enabled during the cache operation. The cache ways are enabled selectively based on the algorithm in [9]. An energy saving is achieved. The proposed model is simulated with SPEC2K benchmarks. Energy savings of 99% is observed with same performance.

The paper is organized as follows. Section 2 gives a motivating example. Section 3 presents the architecture, address mapping/translation of the proposed model. Section 4 gives the performance metrics, section 5 gives simulation, section 6 gives conclusion followed by the references in section 7.

2. MOTIVATION

Consider a two level cache system. Let the levels be L1 and L2 for level one and level two respectively. Let both the levels be 2-way set associative cache. Let the number of sets in level one be four and in level two be eight. Consider the following address requests. 0, 4, 0, 8, 0, 12, 0, 16, 0, 32, 0, 64, 0, 0, 0, 0. Number these sixteen references from 1 to 16. Let the size of cache line by 32 bytes. Consider the two type data cache model proposed in [9]. The address mapping proceeds as follows. There is a miss for references 1, 2, 4, 6, 8, 10, 12. The remaining references viz. 3, 5, 7, 9, 11, 13 to a are hits. The total number of hits is nine and misses is seven. Assume the cache operates in two power modes viz. high power mode and low power mode. Let the energy consumed per cache line be 10 joules and 5 joules in high power and low power mode respectively. During the cache operation, both level one and level two caches are in high energy mode. The total energy consumed is (2*4+2*8)*10*16J=3840J. Consider a tag cache in level one. It consists of the tag values of all the cache ways at all levels arranged in an array. Consider the following algorithm for address a.

1. Compute i1=a%4, t1=a div 4, i2=a%8, t2=a div 8.
2. Check the tag cache locations i1 and 8+i2 for match with tag values t1 and t2 respectively. If there is match with t1 and t2, access the line in level one cache and stop. If there is match with t1 and not with t2 access the line in t1, make the cache way of i2 inclusive. If there is no match with t1 but match with t2, access the cache way i2 in level two. If there is no match for t1 and t2, access the least recently used way in level one, replace the line, update the tag cache.
3. Stop.

In the above algorithm, the tag cache is accessed first and the cache lines are accessed later. The cache is in low power mode during not operational period. For the example, the cache level one is accessed for references one and two. Cache level one and two are accessed for reference 3, level two for reference 4, 6. Cache level one is accessed for all the other references. The addresses are assumed to be 32 bits. The size of tag cache for L1 entries is 4*2*25 bits=200 bits. The size of tag cache for L2 entries is 8*2*24 bits=384 bits. The size of tag cache is 584/8 bytes = 73bytes. The energy consumed in tag cache in this (73*10)/32 J=22.8125J. The total energy consumed for the sixteen references is (22.8125)*16 + 15*10+2*10 + (4*2+8*2)*5 = 655J. The first term is the energy consumed by tag cache, the second term is the energy consumed for accessing one cache way, the third term is the energy consumed in making cache inclusive for reference 3. The fourth term is the energy consumed during non-operation. There is energy savings of 82%. This is the motivation of this paper.

3. ARCHITECTURE OF PROPOSED SYSTEM

The proposed architecture is depicted in Figure 1. The tag cache architecture for two type data cache model is shown in Figure. 1. The system consists of level one and level two cache. The cache can be direct mapped or set associative. The number of sets in each cache need not be equal. The two levels can be of any associativity. A tag array consisting of the tag values of the cache ways in two levels is present in level one. Let the sizes of level one and level two caches are $S_1 and S_2$ respectively. The algorithm for address translation is given next. Consider an address request a to the cache.

1. Compute $i_1 = a\%S_1, i_2 = a\%S_2$
   $t_1 = adivS_1, t_2 = adivS_2$
2. Check for matching of tag values t_1 and t_2 in tag cache. The following conditions are observed.
   a. If there is match for both t_1 and t_2 access level one cache, stop.
   b. If there is match for t_1 and not for t_2, place the line in level two cache, access the line. Update the tag cache entry for level two cache and stop. Both level one and level two cache ways are accessed in this case.
   c. If there is no match for t_1 and match for t_2, access level two cache way and stop.
   d. If there is no match for t_1 and no match for t_2 place the line in least recently used way of level one , access the line . Update the tag cache entry for level one cache and stop.

The tag array is operational during the entire address trace. The cache ways in level one and level two cache are enabled based on the algorithm. Atmost two cache ways are enabled during the address mapping.

As the block is toggled between inclusive and exclusive based on its access, the main memory is updated on each replacement. This is not write through as the proposed model is not strictly inclusive in nature.

4. PERFORMANCE METRICS

Consider cache system with two levels. Let level one cache be w_1-way set associative cache with S_1 sets. Let level two cache be w_2 -way set associative cache with S_2 sets. Let L be the cache line size. Denote this system as C_{trad}. Consider the address trace with R references. Consider the two type data cache model as proposed in [9]. Let the cache operate in two modes viz. high power mode and low power mode. The cache is in high power mode during operation. Let E be the energy consumed in high power mode per cache line. The total energy consumed in this model is given by

$$E(C_{trad}) = (w_1 S_1 + w_2 S_2)ER \tag{1}$$

Consider the cache model proposed in section 3. Denote it as C_{prop}. Let the tag cache capacity be T cache lines. Let α be the number of references for which there is tag match in level one and not in level two (case 2b) of algorithm in section 3). Let β be the number of references that have no match in level one and level two in tag cache (case 2d). The energy consumed in this system during operation is given by

$$E(C_{prop}) = RTE + 2E\alpha + (R - \alpha)E + E_1 \tag{2}$$

Where E_1 is the energy in low power mode for the entire cache system. An improvement in energy consumed is seen if

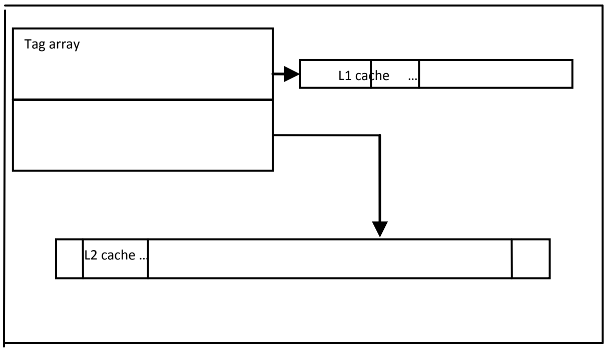

Fig 1 Architecture of proposed system

$$(w_1 S_1 + w_2 S_2)ER >=$$

$$RTE + 2E\alpha + (R-\alpha)E + E_1 \qquad (3)$$

The AMAT is a performance metric used for measuring cache performance. Let the parameters for the system proposed in [9] be as shown in Table 1. The AMAT for this system for address trace of R references is given by

$$AMAT(C_{trad}) = \frac{1}{R}\begin{pmatrix} R(t_1 + t_2) + \alpha t_{12} + \beta(t_{12} + t_{2m}) \\ + x_4 t_1 + \delta t_{1m} \end{pmatrix} \qquad (4)$$

The AMAT for the proposed system is given by

$$AMAT(C_{prop}) = \frac{1}{R}\begin{pmatrix} Rt_c + x_1 t_1 + \alpha t_{12} + \\ \beta(t_{12} + t_{2m}) + x_3 t_2 + \\ x_4 t_1 + \delta t_{1m} \end{pmatrix} \qquad (5)$$

A performance improvement in AMAT is seen if

$$\frac{1}{R}\begin{pmatrix} R(t_1 + t_2) + \alpha t_{12} + \beta(t_{12} + t_{2m}) \\ + x_4 t_1 + \delta t_{1m} \end{pmatrix} >=$$

$$\frac{1}{R}\begin{pmatrix} Rt_c + x_1 t_1 + \alpha t_{12} + \\ \beta(t_{12} + t_{2m}) + x_3 t_2 + \\ x_4 t_1 + \delta t_{1m} \end{pmatrix} \qquad (6)$$

5. SIMULATION

The proposed model was simulated on SPEC 2000 benchmarks using Simplescalar 3.0 binaries. Only certain binaries were built and these were used for the simulation. The benchmark was chosen as it is widely referred to. The addresses of the accessed data were collected using simcache.c program in Simplescalar 3.0. The proposed model was simulated with the following parameters shown in Table 2 . The results of the simulation for AMAT are shown in Figure 2. As seen from this figure, there is an improvement in AMAT of 58% in the proposed system over the two type data cache model proposed in [9]. The results of simulation for energy consumption for the proposed model are shown in Figure 3. The cache line is assumed to consume 10J of energy during operation and 5J of energy during non-operation. As seen from Figure 3 there is improvement in energy consumption by 99%.

Table 1 List of parameters

S.No	Parameter	Description
1	R	Number of references
2	α	Number of vacant level two ways replaced
3	β	Number of occupied level two ways replaced
4	x_4	Number of misses in level one and level two caches
5	t_1	Time to access level one cache
6	t_2	Time to access level two access
7	t_{12}	Transfer time between level one and level two
8	t_{2m}	Access time to access level two from main memory
9	t_{2m}	Access time to access level one from main memory
10	t_c	Tag cache access time
11	x_1	Number of hits in level one and level two cache
12	x_3	Number of misses in level one and hits in level two cache
13	δ	Number of occupied level one ways replaced

Table 2 Simulation Parameters

S.No	Parameter	Value
1	Level one cache size	32KB with block size 32 bytes
2	Level one cache associativity	4
3	Level two cache size	128KB with block size 32 bytes
4	Level two cache associativity	8
5	Level one cache access time	3 cycles
6	Level two cache access time	12 cycles
7	Level one to memory access time	50 cycles
8	Level two to memory access time	65 cycles
9	Level one to level two cache transfer time	20 cycles

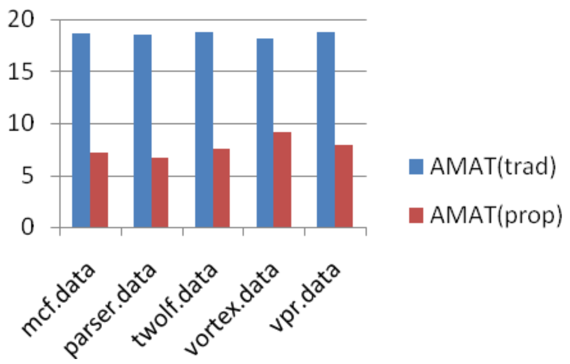

Fig 2 AMAT comparison

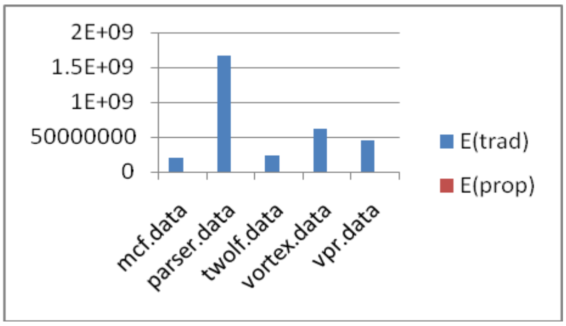

Fig 3 Energy Comparison

6. CONCLUSION

The conditions for the proposed tag cache model for two type data cache model to outperform the two type data cache model in performance are derived. The expressions for the energy consumption of the proposed model are derived. The proposed model is simulated with SPEC2K benchmarks. An improvement in AMAT of 58% with 99% energy saving is observed.

7. REFERENCES

[1] Smith, A. J. 1982. Cache memories. ACM Computing Surveys (CSUR), 14(3), 473-530.

[2] Agarwal, A., Hennessy, J., & Horowitz, M. 1989. An analytical cache model.ACM Transactions on Computer Systems (TOCS), 7(2), 184-215.

[3] David.A.Patterson and John. L. Hennessy 2003 Computer Architecture: A Quantitative Approach, Morgan Kaufmann Publishers, 3rd edititon,

[4] Li, L., Kadayif, I., Tsai, Y. F., Vijaykrishnan, N., Kandemir, M., Irwin, M. J., & Sivasubramaniam, A. 2002. Leakage energy management in cache hierarchies. In Parallel Architectures and Compilation Techniques, 2002. Proceedings. 2002 International Conference on (pp. 131-140). IEEE.

[5] McFarling, S. 1992, April. Cache replacement with dynamic exclusion. InACM SIGARCH Computer Architecture News (Vol. 20, No. 2, pp. 191-200). ACM.

[6] Powell, M. D., Agarwal, A., Vijaykumar, T. N., Falsafi, B., & Roy, K. 2001, December. Reducing set-associative cache energy via way-prediction and selective direct-mapping. In Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture (pp. 54-65). IEEE Computer Society.

[7] Jouppi, N. P., & Wilton, S. J. 1994, April. Tradeoffs in two-level on-chip caching. In Computer Architecture, 1994., Proceedings the 21st Annual International Symposium on (pp. 34-45). IEEE.

[8] Min, R., Xu, Z., Hu, Y., & Jone, W. B. 2004. Partial tag comparison: A new technology for power-efficient set-associative cache designs. In VLSI Design, 2004. Proceedings. 17th International Conference on (pp. 183-188). IEEE.

[9] Subha, S. 2009, June. A two-type data cache model. In Electro/Information Technology, 2009. eit'09. IEEE International Conference on (pp. 476-481). IEEE.

[10] Subha S. 2011 December. An Energy Saving Tag Cache Model, IJCA, 36(8), pp. 38-43

[11] Zhao, L., Iyer, R., Makineni, S., Newell, D., & Cheng, L. 2010, May. NCID: a non-inclusive cache, inclusive directory architecture for flexible and efficient cache hierarchies. In Proceedings of the 7th ACM international conference on Computing frontiers (pp. 121-130). ACM.

[12] Zheng, Y., Davis, B. T., & Jordan, M. 2004. Performance evaluation of exclusive cache hierarchies. In Performance Analysis of Systems and Software, 2004 IEEE International Symposium on-ISPASS (pp. 89-96). IEEE.