A Literature Survey of CPU Exclusive Caches

S.Subha
SITE, Vellore Institute of Technology, Vellore, T.Nadu, India
ssubha@rocketmail.com

***Abstract*— CPU caches can be inclusive or exclusive. In inclusive caches a line is present in cache level and higher cache levels. In exclusive cache a line present in one cache level is not present in any other cache level. Algorithms to access exclusive caches proposed in literature are presented in this paper. Exclusive caches are used in multi core systems. The performance evaluation and power saving methods on exclusive cache hierarchies as proposed in literature are presented.**

***Index Terms*— average memory access time, exclusive cache, energy saving**

I.INTRODUCTION

A cache is denoted by tuple (C,k,L) where C is the capacity, k the associativity and L the line size [1,2]. A cache can be data cache or instruction cache or unified cache. In data cache, the cache lines are for the data accessed in the program. The instruction cache consists of instructions in program. The unified cache contains data and instructions [2]. A computer system contains more than one cache level. A cache can be inclusive or exclusive in nature. In inclusive cache the line in lower cache levels is present in higher cache levels. In exclusive cache, a line is present in only one cache level [1, 3]. A two type data cache model in which line becomes inclusive on repeated access is presented in [4]. Modern processors have more than one core. Exclusive caches are used in these processors widely.

This paper presents the algorithms proposed for exclusive caches. Section 2 gives the algorithms, section 3 the performance and power consumption in exclusive caches, section 4 conclusion followed by references.

II. EXCLUSIVE CACHE ALGORITHMS

The author in [3] proposed the following for exclusive cache with two levels. Both the cache levels are set associative caches. The number of sets in both levels is equal.

1. Start
2. If level one cache contains the line, the line is accessed and stop.
3. If level two cache contains the line, the line is swapped with level one cache accessed and stop.
4. If the line is cache miss i.e. not found in level one and level two caches, the line is fetched into level one cache from main memory evicting the level one cache victim to level two cache. The level two cache victim is evicted to higher memory. The line is accessed in level one cache, accessed and stop.

   There is line for placing main memory line into level one cache to access in the model proposed in [3]. The author reported improvement in average time per instruction in this model. In case the number of sets in two cache levels are not the same, there is chance for inclusion.

The author in [5] proposed exclusive cache algorithm for two level set associative caches. The number of sets in both levels need not be equal. The method involves designating a line as logically belonging to level one or level two. The replacement decision is based on this. Initially all blocks are free. The algorithm is given next.

1. Start
2. If there is a cache hit in level one or level two, treat that block as logical level one and stop.
3. If level one is free, place the block in level one, treat as logical level one and stop.
4. If level two is free, place the block in level two, treat as logical level one and stop.
5. If level one is treated as logical level two and vice versa, place the block in level one, treat physical level one as logical level one, physical level two as logical level two, and stop. Else, place the block in level two, treat physical level one and level two as logical level two and level one respectively and stop.

A index array indicates the logical number of the line. The author reported AMAT improvement of direct-direct by 67%, set associative-set associative by 64% configurations over the model proposed in [3] for SPEC2K benchmarks. The model has scalability. The AMAT degraded by 1% in direct-set associative configuration when compared with model proposed in [3].

III. PERFORMANCE AND POWER CONSUMPTION OF EXCLUSIVE CACHES

The performance evaluation of exclusive cache proposed in [3] was reported in [8]. The authors present results on exclusive cache hierarchies. They found significant performance improvement over inclusive caches in certain benchmarks. There was 16% reduction in execution time. The smallest cache configuration showed reduction of 8% in execution time in the simulated benchmarks. The authors found that if the victim buffer in exclusive cache and victim cache in inclusive cache are of equal size exclusive cache performed better than inclusive cache. This is because conflict misses are reduces significantly in victim cache whereas worst case penalties are introduced in victim buffer. They found inconsistent performance improvement in exclusive cache hierarchy. Hence they suggested that the need for exclusive caches be justified based upon the application specifics and system specifics.

Energy and power consumption is of wide interest in any system today. In caches similar to electronic circuits, the energy or power consumed is directly proportional to the number of active components and the activity duration. The author in [6] proposed energy saving for exclusive cache model. The proposed model has two cache levels. There is one port in each cache level. There are S_1, S_2 sets in level one and level two cache respectively. A tag cache containing values of the tags at all cache levels is arranged consecutively in level one. The block size is set to the capacity of larger of sets in cache levels. One block is accessed to access all the tag entries in set. The access time of tag cache access time is assumed to be equal to level one cache access time. The proposed algorithm is given next.

1. Start
2. Compute the following.
   Set1 = a mod S_1
   Tag1 = a div S_1
   Set2 = a mod S_2
   Tag2 = a div S_2
3. Check for match of Tag1 in tag cache. If it is hit i.e. the line is present in level one cache it is level one hit. The level one cache line is accessed and stop. If there is no match in level one cache check for match of Tag2. If a match is found, it is level two cache hit. Access the level two cache line and stop.
4. This is a cache miss. Place the least recently used line in Set2 of level two in main memory. Transfer the least recently used line of Set1 in level one cache in the evicted level two line. Place the line with address a in level one cache. Update the entries in the tag cache.
5. Stop.

The author assumed that the entire cache system is in two modes – high power mode and low power mode. The cache is in low power mode by default. On accessthe cache line is in high power mode. The tag cache is in high power mode during cache operation. The author simulated the model with SPEC2K benchmarks. An improvement in energy of 23% with comparable AMAT with traditional exclusive cache as proposed in [3] was reported. This model assumes the cache is operational in two modes – high and low. In order to save power, a cache model assuming on or off mode for the cache lines in level one and level two is proposed in [7] . This model as proposed in [7] assumes same cache line placement/replacement policy as in [6]. The cache ways are enabled on access by introducing AND gate with logic one as one of the inputs. This model holds good for non-zero tag values. An improvement in power by 49% is reported.

IV. CONCLUSION

Exclusive caches find wide usage in multicore systems. The algorithms for placement/replacement in exclusive caches are surveyed in this paper. With increase in power consumption, algorithms to improve power or energy consumption in exclusive caches are surveyed in this paper.

REFERENCES

[1] Alan Jay Smith, "Cache Memories", Computing Surveys. 14(3), September1982.

[2] David.A. Patterson, John L Hennessey, "Computer System Architecture,A Quantitative Approach", 3rd Edition, Morgan Kaufmann Publishers Inc., 2003

[3] Jouppi NP, Wilton SJE, "Tradeoffs in two-level on-chip caching", Proceedings of 21st Annual International Symposium on Computer Architecture, pp. 34-45, 1994

[4] S. Subha, " A Two-Type Data Cache Model" Proceedings of 2009 IEEE International Conference on Electro/Information Technology, pp. 476-481, 2009

[5] Subha S, "An exclusive cache model", Proceedings of ITNG , pp. 1715-16, 2009

[6] Subha S, "An energy saving model for exclusive cache", HPCS , pp.233-8, 2011

[7] Subha.S, "An exclusive cache architecture with power saving", IJST, 8(33), December, 2015

[8] Zheng.Y, Davis BT, Jordan M, "Performance evaluation of exclusive cache hierarchies", Proceedings of ISPASS, pp, 89-96, 2004