ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(33), DOI: 10.17485/ijst/2015/v8i33/79871, December 2015* ISSN (Online) : 0974-5645

An Exclusive Cache Architecture with Power Saving

Srinivasan Subha*

Vellore Institute of Technology, Vellore - 632014, Tamil Nadu, India; ssubha@rocketmail.com

Abstract

Background: Tag caches models are proposed in improving cache performance. The tag cache and processor cache are in high energy mode in the proposed models. To reduce the power consumption in tag cache model is aim of this paper. **Methods:** A cache model to save power in tag cache model of exclusive cache is proposed. The SPEC2K benchmarks run against Simplescalar Toolkit is used for simulations. Routines in C language are written to simulate the proposed model and traditional processor cache models. The SPEC2K addresses are run with C program. The power consumption is calculated from Quartus2 using Verilog code for both proposed and traditional models. The proposed model is synthesized in Quartus2. The average memory access time between proposed and traditional models is compared. The power consumption is compared between traditional and proposed models. **Findings:** The power consumption in tag cache is improved in the proposed model by introducing AND gate. The average memory access time is constant with the base model. A power saving of 49% was observed in the proposed model over the model proposed in[5] with no change in average memory access time. The author in[5] reported energy saving of 23% with comparable average memory access time over model presented in[3]. The proposed model is scalable to higher cache levels. The proposed model can be implemented in any processor (uni-processor or multiprocessor)having exclusive cache model. **Applications/ Improvements:** The proposed model can be extended for multilateral caches with suitable logic.

Keywords: Cache Architecture, Cache Power Saving, Exclusive Cache, Logic Circuit, Tag Cache Model

1. Introduction

Cache is denoted by tuple (C, k, L) where C is the capacity with associativity k and line size L[1,2]. Caches are of two kinds based on line inclusion- inclusive cache and exclusive cache. A line is present in higher cache levels in inclusive cache. A line is present in only one cache level in exclusive cache[1]. Algorithms for line placement are proposed for exclusive cache[3,5]. During cache operation the entire cache is assumed to be in high power mode usually. The author proposed tag cache architecture[5]. In this model, the tag values of all cache levels are present in tag cache at level one. The line is accessed based on tag cache hit. Lines are placed if there is free way in level one or level two cache in this model. On conflicts the line is placed in level one cache placing the content of level one cache in level two and moving the level two cache content to main memory. The model differs from the exclusive cache proposed in[3]. A cache model to selectively enable occupied cache ways is proposed in[6]. The authors in[7] propose cache model for CMP that has exclusive level three cache with tag inclusion in cache directory. The performance of exclusive caches is presented in[8]. The cache reconfiguration by selectively choosing cache sets and ways is presented in[4]. The SRAM energy efficiency is improved by 20% if there are more columns than rows[9]. The operational voltage has to be increased by 20% for SRAM cells over ten year period to maintain reliability[10]. This paper proposes an architecture for tag cache model saving power consumption for the model proposed in[5]. The tag cache architecture is assumed.

2. Motivation

Consider two level exclusive tag cache model proposed in[5]. Let level one cache be two way set associative cache

* *Author for correspondence*

of two sets. Let level two cache be two way set associative cache of two sets. The cache line is assumed to operate two power modes- on and off. By default the cache line is in off mode. Consider the address trace 100, 200, 300, 400. Let cache line size be one byte. The addresses map to set zero. They are cache misses. The addresses 100, 200 are placed in level one cache set zero. The level one cache is in on mode during this operation. To place 300, 400 level two cache ways are in on mode. Thus level one cache is in on mode for four time units and level two cache is in on mode for two time units. Assume it takes 10mW power per cache way during operation. Assume it takes 40mW for the tag cache during cache operation. The total power consumed is (40+4*10*4+2*10*4)mW= 280mW. The first term is the power consumed by tag cache. The second term is the power consumed by level one cache and the third term is the power consumed in level two cache. Next consider the following cache model. The cache line is enabled during occupancy. This is achieved by introducing an AND gate with logic one as one of the inputs between the tag cache entry and the cache line in level one or level two cache. This model will have the cache line in set zero mapped to address 100 in on mode for four time units, for address 200 in on mode for three time units. The cache line in level two mapped to address 300 is in on mode for two time units and mapped to address 400 is in on mode for one time unit. The total power consumed is (40+(4+3+2+1)*10)mW = 140mW. A power saving of 50% is observed. This is the motivation for this paper.

3. Proposed Cache Architecture

The proposed cache architecture consists of the tag cache at level one. Let there be two cache levels. The caches are set associative caches. They are exclusive in nature as defined by the algorithm given in[5]. The cache lines are in off position initially. On cache line placement/replacement the cache way is enabled. The tag cache drives the cache lines by the introduction of AND gate with logic one. The comparison of tag bits is done in parallel. Energy can be saved for comparison if selective bit wise comparison is done though this might introduce delay in the tag match process. This model is functional if the tag cache entry is non-zero. The proposed architecture is shown in Figure 1. It is scalable. This model achieves exclusive cache with variable ways in the enabled sets.

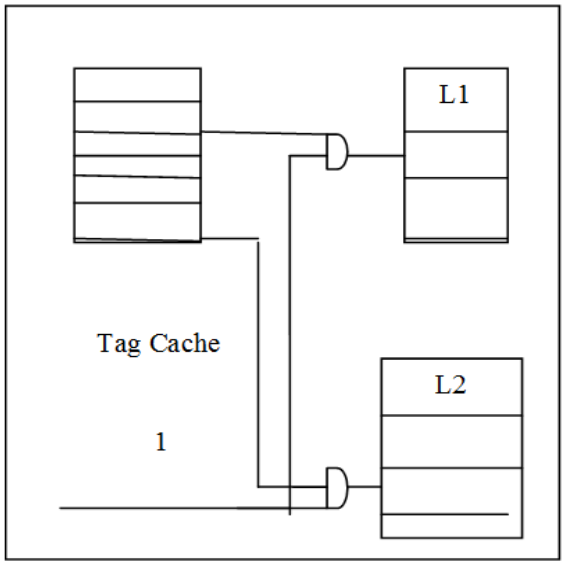

Figure 1. Proposed architecture. The tag cache enables the level one and level two caches.

4. Mathematical Analysis of Proposed Model

Consider two level exclusive cache model proposed in[5]. Let level one be w_1 set associative cache with S_1 sets and level two be w_2 set associative cache of S_2 sets. Let tag cache be present in level one of size T_g lines. The line placement/replacement is as described in[5]. Let this cache be denoted as C_{trad}. Consider the cache architecture described in section 3. Let this be denoted as C_{prop}. Consider the average memory access time as performance measure. Let the parameters be defined as in Table1.
Then,

$$AMAT(C_{trad}) = \frac{1}{R}\begin{pmatrix} Rt_0 + h_1t_1 + h_2t_2 + \\ cmiss1(t_1 + 2t_0) + \\ cmiss2(t_2 + 3t_0) + \\ miss\begin{pmatrix} 4t_0 + t_1 + t_2 + \\ t_{12} + t_{2m} + t_{1m} \end{pmatrix} \end{pmatrix} \quad (1)$$

The first term in (1) is to access the tag cache. The second and third terms are the hit time access to level one and level two caches respectively. The fourth term is the time taken to fill empty way in level one cache. This involves accessing tag cache in level one, fetching the line to level one cache and updating the tag cache entry. The fifth term is the time taken to fill empty way in level two

cache. This includes accessing the tag cache to check for match in level one cache and level two cache and placing the line in level two cache, updating the tag cache entry. The sixth term is the time taken to bring in a line in fully filled level one and level two cache. This involves checking for tag match in level one cache, level two cache, replacing the level one cache line and updating the tag cache entries. As the tag cache contains the tags in consecutive locations, it may be the case that the tag entries in level one and level two are in two different cache blocks. Consider the proposed model. Let the parameters be as defined in Table 1. The AMAT of the proposed system is given as

Table 1. Parameter definitions

Parameter	Description
R	Total references
h_1	Hits in level one cache in traditional tag cache
h_2	Hit in level two cache in traditional cache
cmiss1	Misses filled in empty level one cache in traditional cache
cmiss2	Misses filled in mpty level two cache in traditional cache
miss	Conflict misses in level one and level two cache in traditional cache
t_0	Time to access tag cache
t_1	Time to access level one cache
t_2	Time to access level two cache
t_{1m}	Transfer time between main memory and level one cache
t_{2m}	Transfer time between main memory and level two cache
t_{12}	Transfer time between level one and level two cache
w_1	Level one cache associativity
w_2	Level two cache associativity
H_1	Hits in level one cache in proposed tag cache
H_2	Hit in level two cache in proposed tag cache
CMISS1	Misses filled in empty level one cache in proposed cache
CMISS2	Misses filled in empty level two cache in proposed tag cache
MISS	Conflict misses in level one and level two cache in proposed cache

$$AMAT(C_{prop}) = \frac{1}{R}\begin{pmatrix} Rt_0 + H_1t_1 + H_2t_2 + \\ CMISS1(t_1 + 2t_0) + \\ CMISS2(t_2 + 3t_0) + \\ MISS\begin{pmatrix} 4t_0 + t_1 + t_2 + \\ t_{12} + t_{2m} + t_{1m} \end{pmatrix} \end{pmatrix} \quad (2)$$

An improvement in AMAT is observed if

$$\frac{1}{R}\begin{pmatrix} Rt_0 + h_1t_1 + h_2t_2 + \\ cmiss1(t_1 + 2t_0) + \\ cmiss2(t_2 + 3t_0) + \\ miss\begin{pmatrix} 4t_0 + t_1 + t_2 + \\ t_{12} + t_{2m} + t_{1m} \end{pmatrix} \end{pmatrix} \quad (3)$$

$>=$

$$\frac{1}{R}\begin{pmatrix} Rt_0 + H_1t_1 + H_2t_2 + \\ CMISS1(t_1 + 2t_0) + \\ CMISS2(t_2 + 3t_0) + \\ MISS\begin{pmatrix} 4t_0 + t_1 + t_2 + \\ t_{12} + t_{2m} + t_{1m} \end{pmatrix} \end{pmatrix}$$

Consider the power consumption. Assume the cache operates in two modes- on and off mode. In the traditional cache, the tag cache, level one cache and level two cache are in high power mode. Let the cache be operational for T time units.The power consumed in this cache is given by

$$(T_g + w_1S_1 + w_2S_2)TE_{new} \quad (4)$$

Where E_{new} is the energy consumed in high energy mode. The energy in the proposed model is presented next. On cache line occupancy the line is in on mode. By default it is in off mode. Let $x_1\ x_2,\ldots,\ x_t$ be the number of enabled cache ways during time $t_1,t_2,\ldots,t_t$ The total power consumed is

$$\begin{pmatrix} x_1(T - t_1) + (x_2 - x_1)(T - t_2) + (x_3 - x_2)(T - t_3) \\ +.. \\ .+ (x_t - x_{t-1})(T - t_t) \end{pmatrix} E_{new} \quad (5)$$

An improvement in power consumption is observed if

$$(T_g + w_1S_1 + w_2S_2)TE_{new} >= \begin{pmatrix} x_1(T - t_1) + (x_2 - x_1)(T - t_2) + (x_3 - x_2)(T - t_3) \\ +\ldots + (x_t - x_{t-1})(T - t_t) \end{pmatrix} E_{new} \quad (6)$$

In the above discussion, the energy consumed by cache way involves the energy for the tag, control bits, data bits of cache way.

5. Simulation

The proposed model is simulated with SPEC2K

benchmarks using Simple scalar Toolkit. Routines in C language are written simulating the cache model. The simulation parameters are shown in Table 2. The proposed model is compared with the model in[5]. The values of hits and misses are collected from the program execution. The AMAT is calculated. As shown in Figure 2 the AMAT remains same. The power consumed is calculated using Quartus 2. Code in Verilog is written for 2-way set associative cache of two sets in level one and level two. The code is compiled and synthesized and power consumption is obtained from Quartus 2. The power consumed per cache way is calculated from this. This is listed in Table 2. The power consumed is calculated for the values obtained from C program execution on SPEC2K benchmarks. The total enabled cache ways is taken for calculations in the proposed model. The power consumption is shown in Table 3. As seen from Table 3there is 49%power improvement in the proposed model.

Table 2. Simulation parameters

Sl.No.	Parameter	Value
1.	Level one cache size	128KB
2.	Level one associativity	4
3.	Level two cache size	256KB
4.	Level two associativity	8
5.	Level one access time	3 cycles
6.	Level two access time	18 cycles
7.	Level one to level two transfer time	18 cycles
8.	Level one to memory access time	60 cycles
9.	Level two to memory access time	90 cycles
10.	Line size	32 bytes
11.	Power per cache way	8332.8 μW

Table 3. Power consumption comparison

Name	Power(t)KW	Power(p)KW	%Improve
256.bzip2	73271.56	73271.56	1.98603E-14
181.mcf	621.5154	229.5009	63.07397959
197.parser	5086.737	2716.116	46.60395408
300.twolf	707.5931	262.1133	62.95705782
255.vortex	1888.692	710.3674	62.38839286
175.vpr	1372.853	516.3521	62.38839286
	Average		49.56862954

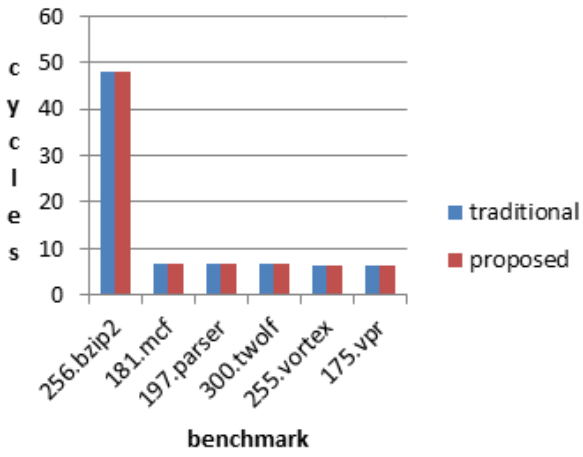

Figure 2. AMAT comparison.

6. Conclusion

An architecture for two level tag cache mode proposed in[5]with power savings is proposed in this paper. The occupied ways determined from tag cache are enabled by AND'ing with one logic and the tag cache entry. All other ways are in off mode. An exclusive cache architecture with variable cache ways for occupied sets is achieved in the proposed model. Expressions for AMAT and power consumption are derived for the proposed model. Conditions for performance improvement are derived compared to the model listed in[5]. The proposed model is scalable. The proposed model is simulated with SPEC2k benchmarks. An improvement in power consumption of 49% with no change in AMAT is observed.

7. Acknowledgements

The author thanks Santa Clara University, CA, USA for providing Simple Scalar Toolkit, SPEC2K benchmarks.

8. References

1. Smith AJ. Cache memories. Computing Surveys. 1982; 14(3):473–530.
2. Patterson DA, Hennessy JL. Computer architecture: A quantitative approach. 3rd ed. USA: Morgan Kaufmann Publishers; 2003.
3. Jouppi NP, Wilton SJE. Tradeoffs in two-level on chip caching. Proceedings of the 21stAnnual International Symposium on Computer Architecture; Chicago, IL, USA. 1994. p.

34–45.
4. Yang S-H, Powell MD, Falsafi B,Vijaykumar TN. Exploiting choice in resizable cache design to optimize deep-submicron processor energy-delay. Proceedings 8th International Symposium on High-Performance Computer Architecture; 2002. p. 151–61.
5. Subha S. An energy saving model of exclusive cache. HPCS; 2011. p. 233–8.
6. Subha S.A reconfigurable cache architecture. 2014 International Conference on High Performance Computing Applications; Bhubaneswar. 2014. p. 1–5.
7. Zhao L, Iyer R, Makineni S, Newell D, Cheng L. **NCID: A** non-inclusive cache, inclusive directory architecture for flexible and efficient cache hierarchies. Proceedings of the 7th ACM International Conference on Computing frontiers; Bertinoro, Italy. 2010. p. 121–30.
8. Zheng Y, Davis BT, Jordan M. Performance evaluation of exclusion cache hierarchy. 2004 IEEE International Symposium on Performance Analysis of Systems and Software; Austin, Texas, USA. 2004. p. 89–96.
9. Ashwin JS, Praveen JS, Manoharan N. Optimization of SRAM array structure for energy efficiency improvement in advanced CMOS technology. Indian Journal of Science and Technology. 2014; 7(S6):35–9.
10. Koushik SK, Lakshmi B. Ageing degradation impact on stability of 6T-SRAM bit cell. Indian Journal of Science and Technology. 2015; 8(20):1–7.