Journal of Theoretical and Applied Information Technology
28th February 2014. Vol. 60 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A FULLY ASSOCIATIVE CACHE ALGORITHM WITH IMPROVED PERFORMANCE

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

E-mail: ssubha@rocketmail.com

ABSTRACT

Fully associative caches enable all blocks to map address. This paper proposes an algorithm to enable one fully associative block. The address a is right shifted by certain prefixed number of bits. The result is XOR'ed with n-1, n being the number of fully associative cache blocks. The bit selection of the result to map to one of the n blocks in fully associative cache is performed. The chosen block is enabled to access a line or place cache line. The proposed algorithm is simulated with SPEC2K benchmarks. The average memory access time is improved by 16% with energy savings of 9%. The proposed model shows 30% degradation in average memory access time with no change in energy consumption over direct mapped cache of same size.

Keywords: *Average Memory Access Time, Cache Algorithm, Cache Performance, Energy Saving, Fully Associative Cache*

1. INTRODUCTION

Caches are denoted by (C,k,L) for capacity C with associativity k and line size L bytes. Caches are of three kinds –direct mapped, set associative and fully associative [1,3] A line is placed in fixed position in direct mapped cache. It can occupy any of the k-ways in k-way set associative cache in the mapped set. It can occupy any of the n blocks in fully associative cache of n blocks. In fully associative cache, the address and data are placed in block. The address match is performed, the block accessed on hit. In case of miss, the least recently used block is replaced. This operation involves performing address match with all blocks. Hence all blocks are enabled during address mapping. Ideally, it is desirable if only one block is enabled during address mapping. The author in [4, 5] proposes algorithm to enable subset of the fully associative cache. In [7] only one block is enabled. However there is increase in average memory access time. A method to save energy in fully associative cache is presented in [6]. The use of XOR functions improves the performance of set associative caches is shown in [2].

This paper proposes algorithm with improved AMAT and energy saving in fully associative cache. The address is shifted right by prefixed number of bits. The result is XOR'ed with n-1, n number of fully associative cache blocks. Bit selection is done to map to block. The mapped block is accessed. The line is accessed if present (hit) or placed in the mapped block (miss). The proposed model is simulated with SPEC2K benchmarks. The average memory access time is improved by 16% with energy savings of 9%. The proposed model shows 30% degradation in average memory access time with no change in energy saving over direct mapped cache of same size.

The rest of the paper is organized as follows. Section 2 gives motivation, section 3 the proposed model, section 4 mathematical analysis of proposed model, section 5 simulations, section 6 conclusion followed by acknowledgments and references.

2. MOTIVATION

Consider a cache system of one level. Let level one be fully associative cache of four blocks. Consider the address trace 100, 101, 104, 123, 140 . These are placed in blocks 0,1,2,3,0 respecting the least recently used replacement policy. Consider the following algorithm for address 100.

Journal of Theoretical and Applied Information Technology
28th February 2014. Vol. 60 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

1. Right shift 100 one position. This gives 50
2. Perform XOR with (4-1). This gives 49.
3. Extract 2 bits from 49 i.e. perform 49%4. This gives 1. Address 100 is mapped to block 1.
4. Stop.

Repeating the above algorithm for addresses 101, 104, 123, 140 maps to blocks 1, 3, 2, 1. In this mapping only one block is accessed. If the cache consumes 10J per block during operation, the total energy consumed during operation in traditional cache is 4*5*10J=200J. Assume the cache operates in two modes viz. high power mode and low power mode in the proposed model. A block is in low power mode during non-operation. Assume the cache consumes 5J per block during non-operation. The energy consumed in this model is 5*4*5J+5*5J = 125J. An improvement of 37.5% in energy consumption is observed. This is the motivation of this paper.

3. PROPOSED MODEL

Consider a cache system. The performance of the cache system is measured in terms of average memory access time. The energy consumed by the cache system depends on the number of active components. The aim of cache operation is

Min Energy consumed

Subject to

Min average memory access time.

(1)

The following algorithm aims at achieving (1)

Consider fully associative cache of n blocks. Consider address a. The following algorithm maps the address to block b.

1. computer a>>line size to get the address
2. compute a>>x for some x
3. Compute a XOR (n-1).
4. choose the block b=a%n
5. If the address and data are present in b, increment number of hits, access the block and stop.
6. Place the block in b , increment number of misses, access the block and stop.

The number of bits used for right shift in step 2 of the above algorithm can be varied.

The architecture of the proposed model is shown in Figure 1. Only one block is accessed in the proposed model.

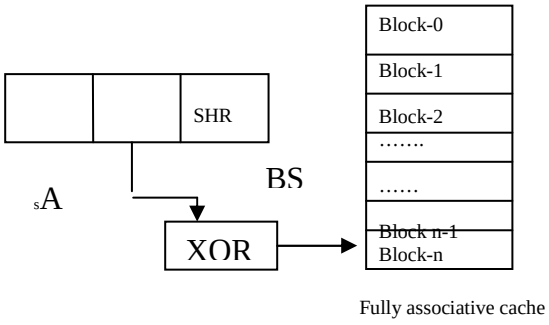

Figure 1 Architecture of proposed model. Address A is right shifted SHR bits. The result is XOR'd wih n-1, and bit selection BS is done to map to block in fully associative cache of n blocks.

4. MATHEMATICAL ANALYSIS OF PROPOSED MODEL

Consider two level inclusive cache system . Let level one be fully associative cache of n blocks. Let level two be w_1-way set associative cache of S sets. Denote this system as C_{prop} .Consider address trace with R references. Consider the algorithm proposed in section 3. Let k be the time taken to perform the address mapping. Let h_1, h_2 be the hits in level one and level two cache respectively. Let t_1, t_2, t_{12}, M be level one hit time, level two hit time, time to transfer block of data between level one and level two and miss penalty respectively. The average memory access

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

time is given by

$$AMAT(C_{prop}) = \frac{1}{R}\begin{pmatrix} Rk + h_1 t_1 + h_2(t_1 + t_2 + t_{12}) + \\ (R - h_1 - h_2)M \end{pmatrix} \quad (2)$$

The first term in (2) is the time taken to perform address mapping. The second term is the time taken to access level one hits. The third term is the time taken to access level two hits. The fourth term is the time taken to access the misses in the cache system.

Consider the traditional cache system. Let level one be fully associative cache and level two be set associative cache with same capacity as C_{prop}. Denote this system as C_{trad}. The address mapping in this system is done as usual. Let H_1, H_2 be the number of hits in level one and level two respectively. Let T_1, T_2, T_{12}, m be level one hit time, level two hit time, transfer time between level one and level two, and miss penalty respectively. The average memory access time is given by

$$AMAT(C_{trad}) = \frac{1}{R}\begin{pmatrix} H_1 T_1 + \\ H_2(T_1 + T_2 + T_{12}) + \\ (R - H_1 - H_2)m \end{pmatrix} \quad (3)$$

The first term in (3) is the level one hit time, second term is the level one miss and level two hit, third term is the time to service miss in level one and level two caches. An improvement in AMAT is observed if

$$\frac{1}{R}\begin{pmatrix} Rk + h_1 t_1 + h_2(t_1 + t_2 + t_{12}) + \\ (R - h_1 - h_2)M \end{pmatrix} <=$$

$$\frac{1}{R}\begin{pmatrix} H_1 T_1 + H_2(T_1 + T_2 + T_{12}) + \\ (R - H_1 - H_2)m \end{pmatrix} \quad (4)$$

Consider direct mapped cache of same size. Let the number of sets be S_d. Let h_{1d}, h_{2d} be level one and level two cache hits. Let $t_{1d}, t_{2d}, t_{12d}, m_d$ be level one access time, level two access time, transfer time between level one and level two and miss penalty in direct mapped cache. The average memory access time is given by

$$AMAT(C_{direct}) = \frac{1}{R}\begin{pmatrix} h_{1d} t_{1d} + \\ h_{2d}(t_{1d} + t_{2d} + t_{12d}) + \\ m_d(R - h_{1d} - h_{2d}) \end{pmatrix} \quad (5)$$

A performance improvement in AMAT with direct mapped cache is observed if

$$\frac{1}{R}\begin{pmatrix} Rk + h_1 t_1 + h_2(t_1 + t_2 + t_{12}) + \\ (R - h_1 - h_2)M \end{pmatrix} <=$$

$$\frac{1}{R}\begin{pmatrix} h_{1d} t_{1d} + \\ h_{2d}(t_{1d} + t_{2d} + t_{12d}) + \\ m_d(R - h_{1d} - h_{2d}) \end{pmatrix} \quad (6)$$

Consider the energy consumption. In the proposed model only one block in level one is accessed. The entire cache set is accessed in level two access. Assume the cache system operates in two energy modes – high energy mode and low energy mode. The cache system is in low energy mode during non-operation. Let the energy consumed per cache block during operational and non-operational mode be E_{high}, E_{low} respectively. In the proposed model the following happens.

1. For h_1 references, one fully associative block is enabled in high energy mode.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. For h_2 references, one fully associative block and one set in level two is enabled in high energy mode.

3. For $R-h_1-h_2$ references, one block in level one and one set in level two is enabled in high energy mode.

The total energy consumed in the proposed model is given by

$$E(C_{prop}) = RE_{low}(n+w_1S) + h_1(E_{high}-E_{low}) + h_2((E_{high}-E_{low}) + w_1(E_{high}-E_{low})) + (R-h_1-h_2)(E_{high}-E_{low})(1+w_1) \quad (7)$$

The first term in (7) is the energy consumed by cache system during non-operation. The second term is the energy consumed during level one hits. The third term is the energy consumed in level two hits. The fourth term is the energy consumed during miss. In the traditional cache the following happens.

1. The entire fully associative cache is in high energy mode during operation.

2. The level two set associative cache is in non-operational mode. During access one set is in operational mode.

The energy consumed in the traditional cache is given by

$$E(C_{trad}) = RE_{low}w_1S + RnE_{high} + h_2(w_1(E_{high}-E_{low})) + (R-h_1-h_2)w_1(E_{high}-E_{low}) \quad (8)$$

The first term in (8) is the energy consumed by level two cache during non-operation. The second term is the energy consumed by fully associative cache in level one. The third term is the energy consumed during level two hits. The fourth term is the energy consumed during miss. An improvement in energy consumption is observed if

$$RE_{low}(n+w_1S) + h_1(E_{high}-E_{low}) + h_2((E_{high}-E_{low}) + w_1(E_{high}-E_{low})) + (R-h_1-h_2)(E_{high}-E_{low})(1+w_1)$$

$$<= RE_{low}w_1S + RnE_{high} + h_2(w_1(E_{high}-E_{low})) + (R-h_1-h_2)w_1(E_{high}-E_{low}) \quad (9)$$

Consider the direct mapped system. The following happens.

1. The direct mapped cache is in low power mode during non-operation. The accessed set is in high power mode during operation.

2. The level two set associative cache is in low power mode during non-operation. The accessed set is in high power mode during operation.

The energy consumed in the direct mapped cache is given by

$$E(C_{direct}) = RE_{low}(S_d + w_1S) + h_{1d}(E_{high}-E_{low}) + h_{2d}(1+w_1)(E_{high}-E_{low}) + (R-h_{1d}-h_{2d})(1+w_1)(E_{high}-E_{low}) \quad (10)$$

The first term in (10) is the energy consumed by the cache system during non-operation. The second term is the energy consumed during direct mapped cache hits. The third term is the energy consumed during level one miss, level two hits. The fourth term is the energy consumed during miss. An improvement in energy consumption is observed if

$$RE_{low}(n+w_1S) + h_1(E_{high}-E_{low}) + h_2((E_{high}-E_{low}) + w_1(E_{high}-E_{low})) + (R-h_1-h_2)(E_{high}-E_{low})(1+w_1)$$

$$<= RE_{low}(S_d + w_1S) + h_{1d}(E_{high}-E_{low}) + h_{2d}(1+w_1)(E_{high}-E_{low}) + (R-h_{1d}-h_{2d})(1+w_1)(E_{high}-E_{low}) \quad (11)$$

Journal of Theoretical and Applied Information Technology
28th February 2014. Vol. 60 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. SIMULATIONS

The proposed model is simulated with SPEC2K benchmarks. The simplescalar Toolset was used to gather addresses from SPEC2K benchmarks. The program was run on a Linux system with uniprogram environment. Routines in C program were written to simulate the proposed model. A two level data cache model was simulated with the parameters shown in Table 1. The addresses were shifted by fourteen bits in this simulation. The simulations resulted in same performance increasing the number of bits used for right shift in the proposed model. The number of hits and misses were collected.

Table 1 Simulation Parameters

Parameter	Value
L1 size in blocks	32768
Block size	32 bytes
L2 size in sets	65536
L2 associativity	4
L1 access time in fully associative cache in proposed model	1 cycle
L1 access time in traditional fully associative cache	6 cycles
L2 access time	12 cycles
Miss penalty	65 cycles
XOR calculation	1cycle
L1 to L2 transfer time	10 cycles
L1 access time in direct mapped cache	1 cycles

The AMAT is shown in Figure 2. As seen from the Fig.2 there is an improvement in AMAT by 16% over traditional fully associative cache and degradation of 30% over direct mapped cache of same size.

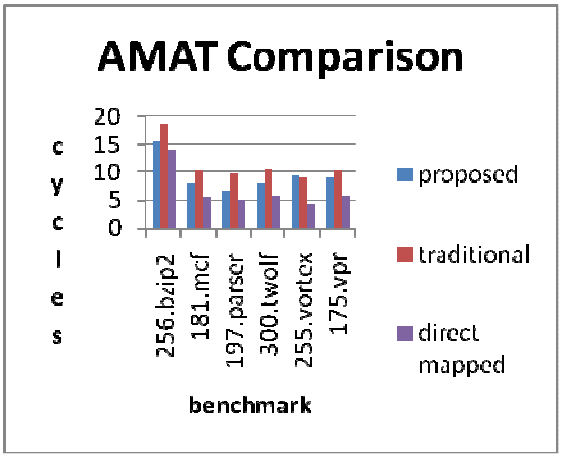

Figure 2 AMAT comparison

The energy consumed with respect to traditional cache system is shown in Table 2. The cache is assumed to consume 5J during non-operation and 10J during operation per cache way. As seen from Table 2 there is improvement of 9% in energy consumed.

Table 2 Energy comparison with traditional cache system

name	Energy(p)	Energy(t)	%improve
256.bzip2	6.89E+11	7.66E+11	9.999776
181.mcf	5.84E+09	6.49E+09	9.999603
197.parser	4.78E+10	5.32E+10	9.99966
300.twolf	6.65E+09	7.39E+09	9.999616
255.vortex	1.78E+10	1.97E+10	9.999465
175.vpr	1.29E+10	1.43E+10	9.999552
		average	9.999612

There is no change in energy consumption of the proposed model with respect to direct mapped cache.

6. CONCLUSION

A fully associative cache enabling one block during operation is proposed in this paper. The address is shifted right by pre-fixed number of bits. It is XOR'd with n-1 in fully associative cache of n blocks. The result is bit extracted to map to block. The proposed model is simulated with SPEC2K benchmarks. The average memory access time is improved by 16% with energy savings of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

9%. The proposed model shows 30% degradation in average memory access time with no change in energy consumption over direct mapped cache of same size.

ACKNOWLEDGMENTS

The authors expresses thanks to Santa Clara University, Santa Clara, CA, USA for providing Simplescalar Toolkit and SPEC2K benchmarks.

REFERENCES

[1] Alan Jay Smith, "Cache memories", Computing Surveys, Vol. 14, No.3, pp. 473-530, September 1982

[2] Antonio Gonzalez, Mateo Valero, Nigel Topham, Joan M Parciresa, "Eliminating Cache Conflict Misses Through XOR-Based Placement Function", Proceedings of ICS, 1997

[3] David A.Patterson, John L.Hennessey, Computer System Architecture, 3rd ed., Morgan Kaufmann Inc., 2003

[4] S.Subha, "An Algorithm for Fully Associative Cache Memory, Proceedings of Second Bihar Science Conference, 2009

[5] S.Subha, "Performance analysis of variable number of sets in fully associative cache", Proceedings of International Conference on Computing, Engineering and Information, pp. 295-298 , 2009

[6] S.Subha, "An Energy Saving Model for Fully Associative Cache", Proceedings of TISC, 2011

[7] S.Subha, "Energy efficient fully associative cache model", International Journal of Computing Applications, Vol. 47, No.6, pp. 16-18, June 2012