Energy Efficient Fully Associative Cache Model

S.Subha
SITE

Vellore Institute of Technology Vellore, India

ABSTRACT

Energy consumption in caches depends on the number of enabled sets/ways/blocks. The optimal energy consumption is the case of one set/one way/one block enabled. This paper proposes an algorithm to map cache line to one block in fully associative cache by XOR'ing the address with constant. Bit selection is applied to the result and the block accessed. Only one block is accessed in this mapping. The proposed model is simulated with SPEC2K benchmarks. The average memory access time is comparable with traditional fully associative cache with energy savings.

General Terms

Performance, Design

Keywords

Energy consumption, Fully associative cache, XOR logic

1. INTRODUCTION

Caches come in three models-direct mapped, set associative and fully associative [1, 4]. In direct mapped cache a block can be placed in one set. In w-way set associative model, a block can occupy any of the w-ways of the cache set. In fully associative model, a cache line can be placed in any of the cache blocks. Usually bit selection equivalent to mod function is used to choose cache set in direct mapped and set associative caches. In fully associative cache, the address along with data is matched with each of the cache blocks. On a hit, the block is accessed. The least recently used algorithm is used to replace cache miss in caches usually. For a fully associative cache with n blocks, the amount of energy consumed for match logic is nE where E is the energy consumed per block.

Energy consumption is widely studied topic in caches. The authors in [3] propose a power efficient cache with matching of the four least significant bits of tags. The authors in [5] propose a method to disable a set of ways in set associative cache during modest activity while the full cache may remain operational for more cache intensive periods to save energy. The authors in [7] suggest a method to access only the tag store during comparison. In [8] a method is proposed to enable only single way of set associative cache. In [9] a way is predicted which is probed during access of the set associative cache saving energy. Partial tags are compared in the method proposed in [10]. The cache is divided into sub-caches in [11]. Different techniques for data placement, sub cache prediction and selective probing are proposed in this paper. The author in [12] proposed a method to save energy in set associative cache. The author in [14] evaluates the idea of restriction the number of blocks to be searched in fully associative cache. The energy consumed by cache operation depends on the number of blocks/sets accessed along with the number of tag comparisons. The authors in [2] discuss the XOR mapping to direct mapped, set associative, victim cache,

hash-rehash cache, column associative, skew-symmetric caches.

This paper proposes an algorithm to map address to one block in fully associative cache of n blocks. The algorithm negates each bit in the address and uses bit selection to map to a block. This is equivalent to bitwise XOR'ing the address with a string of 1's which is $2^n - 1$ where $n = 2^n$. This mapping should not be confused with the mapping proposed in [12]. In [12] the address is XOR'ed with $0, 1, 2, \dots, 2^n - 1$ and bit selection is applied to the result to access or place a cache line. Here the address translation involves XOR'ing with $2^n - 1$. The proposed algorithm is simulated with SPEC2K benchmarks. The average memory access time (AMAT) is comparable with fully associative cache. Since one block is accessed in the proposed model, the match logic is applied to the mapped block. Hence there is energy saving in the match circuitry. The proposed algorithm performs comparable with direct mapped memory of same size.

The rest of the paper is organized as follows. Section 2 gives the motivation, section 3 the proposed model, section 4 the mathematical analysis of the proposed system, section 5 simulation, section 6 conclusion, section 7 acknowledgements and section 8 references.

2. MOTIVATION

Consider a fully associative cache of eight blocks. Consider the address stream 1, 2, 1, 3, 1, 4. The line corresponding to address 1 is placed in block 0, address 2 is placed in block 1, address 3 in block 2, and address 4 is placed in block 3 of a fully associative cache. There are four misses and two hits. All the blocks are searched for an address. If the energy consumed per block in match logic is 10J, 80J of energy is consumed per access. The total energy consumed for the given address stream is $6 \times 80J = 480J$. Consider the following algorithm.

1. For address a XOR with 7.
2. Use bit selection to map to a block.
3. Stop.

According to the proposed algorithm, the address 1 is XOR'ed with 7 which is 111 in binary. This gives 110. The address 1 is placed in block 6. Address 2 on XOR'ing with 7 gives 101 (010 XOR 111) and is placed in block 5. Address 3 on XOR'ing with 7 gives 100 (011 XOR 111) and is placed in block 4. Address 4 XOR'ed with 7 gives 011 (100 XOR 111) is placed in block 3. The number of hits is two and misses is four. Only one block is accessed for mapping an address. Hence the total energy consumed per access is 10J. For the given address stream the total energy consumed is 60J. There is 87.5% savings in energy consumption. This is the motivation of this paper.

3. PROPOSED MODEL

Consider a fully associative cache with n blocks. Consider an address stream. The addresses are mapped to the cache based on the fully associative cache model. All the blocks of the fully associative cache are enabled for the entire trace. The match logic circuitry is enabled for all the blocks. Now, consider the following algorithm for an address a .

1. XOR the address with $n-1$.
2. Use bit selection to map to any of the n blocks.
3. Stop.

Only one block is accessed in this algorithm. The match circuitry of the accessed block is enabled. This saves energy. The address mapping is depicted in Figure 1.

Figure 1. Architecture of the proposed system

4. MATHEMATICAL ANALYSIS OF PROPOSED MODEL

Consider fully associative cache with n blocks. Let this be denoted as C_{trad} . Consider address trace of R references. Let the references be placed according to the logic of the fully associative cache. Let H be the number of hits, t the hit time per reference, M miss penalty. The AMAT for this model is given by

$$AMAT(C_{trad}) = \frac{1}{R} (Ht + (R - H)M) \quad (1)$$

Next, consider the algorithm proposed in section 3. Let the system implementing this algorithm be denoted as C_{prop} . Let h be the number of hits, T the time to access one cache block, k the XOR mapping time for one address. The AMAT of the proposed system is given by

$$AMAT(C_{prop}) = \frac{1}{R} (hT + (R - h)M + Rk) \quad (2)$$

An improvement in AMAT is seen if

$$Ht + (R - H)M > hT + (R - h)M + Rk \quad (3)$$

Consider the energy consumed in fully associative cache. In the traditional cache, all the match logic is performed in

parallel for n blocks. If the energy consumed per block is E Joules, the total energy consumed is

$$REn \quad (4)$$

In the proposed model, only one cache block is accessed for match logic. The total energy consumed in the proposed model is

$$RE \quad (5)$$

A performance improvement of $\frac{n-1}{n}$ is observed.

5. SIMULATION

The proposed **model** was simulated with SPEC2000 benchmarks. The addresses were collected using Simplescalar toolkit. Routines in C language were written to simulate the proposed model. The hits and misses were collected for the benchmarks. The parameters for the simulation are given in Table 1. The proposed model was compared with fully associative and direct mapped cache models for AMAT. As seen from Figure 2 the AMAT is comparable with these models. The AMAT is calculated as follows. The cache is simulated as an array. Each address is mapped to a block in fully associative cache. The number of hits and misses are gathered from the address trace. The AMAT is calculated by substituting the values in the equations (1) and (2) for the traditional and proposed models respectively.

Table 1 Simulation Parameters

Name	Value
L1 size	32768 sets/blocks
Line size	32 bytes
Hit time of direct mapped	1 cycle
Miss penalty of any cache	40 cycles
Hit time of fully associative cache	6 cycles
Hit time of proposed model	1 cycles
Address translation by using XOR Gate	2 cycles

Figure 2. AMAT comparison

The energy savings in the proposed model over the fully associative cache is shown in Table 2. It is assumed that the cache operates in two modes viz, high power mode and low

power model. By default the cache is in low power model. It is assumed that 10J energy is consumed during cache operation. Thus for example if there are 1000 references in the address trace for the assumed parameters the energy consumed in traditional fully associative cache is $1000 \times 32768 \times 10J$ as all the blocks are enabled. In the proposed model, only one block is in high power mode during operation. Thus for 1000 references the total energy consumed during cache operation is $1000 \times 10J$. The percentage improvement in energy consumption is calculated with the above model.

Table2 Energy Comparison

Name	prop	trad	%improvement
256.bzip2	5E+06	1.5E+11	99.99695
181.mcf	79270	2.6E+09	99.99695
197.parser	648850	2.1E+10	99.99695
300.twolf	90250	3E+09	99.99695
255.vortex	240910	7.9E+09	99.99695
175.vpr	175110	5.7E+09	99.99695
		average	99.99695

6. CONCLUSION

A fully associative address mapping algorithm is proposed in this paper. An address is XOR'ed with n-1 for a fully associative cache of size n blocks. Bit selection is applied to the result and the corresponding block is accessed. The proposed model is simulated with SPEC2000 benchmarks. The AMAT for the proposed system is comparable with fully associative cache and direct mapped cache of the same size. An average energy savings of 99% in operational mode is observed.

7. ACKNOWLEDGEMENTS

The author thanks Santa Clara University, Santa Clara, CA, USA for providing SPEC2000 benchmark and SimpleScalar Toolkit.

8. REFERENCES

- [1] A.J.Smith, Cache memories, ACM Computing Surveys, Vol.14 No.3, pp.473-530, September 1982
- [2] Antonio Gonzalez, Mateo Valero, Nigel Topham, Joan.M. Parcerisa, Eliminating Cache Conflict Misses Through XOR-Based Placement Functions, Proceedings of International Conference on Supercomputing, 1997, pp. 76-83
- [3] Chuanjun Zhang, Frank Vahid, Jun Yang and Walid Najjar, A Way-Halting Cache for Low-Energy High Performance Systems, ACM Transactions on Architecture and Code Optimization, Vol.2, No.1, March 2003, pp. 34-54
- [4] David.A.Patterson and John.L.Hennessy, Computer Architecture: A Quantitative Approach, Morgan Kaufmann Publishers, 3rd Edition, 2003
- [5] David.H.Albonesi, Selective Cache Ways: On demand cache resource Allocation, Proceedings of International Symposium on Microarchitecture, 1999
- [6] G.Rivera, C.W.Tseng, Data Transformations for Eliminating Conflict Misses, PLDI, 1998
- [7] Huiyang Zhou, Mark.C.Toburen, Eric Rotenberg, Thomas.M.Conte, Adaptive Mode Control: A Static-Power-Efficient Cache Design, ACM Transactions on Embedded Computing Systems, Vol.2, No. 3, August 2003, pp. 347-372
- [8] Jung-Wook Park, Gi-Ho Park, Sung-Bae Park, Shin-Dug Kim, Power-aware deterministic block allocation for low-power way-selective cache structure, Proceedings of ICCD, 2004
- [9] Michael.D.Powell, Amit Agarwal, T.N.Vijaykumar, Babak Falsafi, Kaushik Roy, Reducing set-associative cache energy via way-prediction and selective direct mapping, Proceedings of MICRO, 2001
- [10] RuiMin, Zhiyong Xu, Yiming Hu, Wen-ben Jone, Partial tag comparison: a new technology for power-efficient set-associative cache designs, Proceedings of 17th International Conference on VLSI Design, 2004, pp. 183-188
- [11] S.Kim, N.Vijaykrishnan, M.Kandemir, A.Sivasubramaniam, M.J.Irwin, E.Geethanjali, Power-aware partitioned cache architectures, Proceedings of ISPLED, 2001, pp. 64-67
- [12] S.Subha, Set Associative Cache Model with Energy Saving, Proceedings of ITNG, 2008, pp.1249-1250
- [13] S.Subha, An Algorithm for Fully Associative Cache Memory, Proceedings of Second Bihar Science Conference, 2009
- [14] S.Subha, Performance Analysis of Variable number of Sets in Fully Associative Cache, Proceedings of ICC, 2009, pp. 295-298
- [15] Zhigang Hu, Stefanos Kaxiras, Margaret Martonosi, Improving Cache Power Efficiency with an Asymmetric Set Associative Cache, Proceedings of Workshop of Memory Performance, 2001