An Algorithm for Variable Cache Ways

S. Subha

***Abstract*—Cache ways are fixed in traditional caches. This paper proposes an algorithm to have variable number of ways in set associative caches. The cache is assumed to be fast registers. A free register to a set is allocated on cold miss. The least recently used policy is used to replace a way in case of no free registers. This algorithm results in variable number of ways for mapped sets. Simulations were performed with SPEC2K benchmarks on the proposed model. An improvement of 3% is seen in average memory access time.**

***Index Terms*—Average memory access time, set associative cache, variable ways.**

I. Introduction

Caches are of three kinds-direct mapped, set associative, fully associative [1]. The number of sets in direct mapped and set associative caches is fixed. The number of ways in set associative cache is fixed. The authors in [2] propose a model to have variable number of ways in set associative cache. This method is based on static profiling. The authors in [3] describe set associative cache having fixed number of ways per set. The authors in [4] propose a method to choose certain cache ways. The authors in [5] propose a cache with variable ways in secondary caches.

This paper proposes a set associative cache with variable ways in level one cache. The model assumes cache to be set of fast registers. On mapping to a set, the registers are searched for a match. On match the register is accessed. On miss, a free register is allocated for this line. The set number is stored as part of book keeping of the registers. If there are no free registers, the least recently used register among the registers allocated to this set is replaced. In case of new set and all registers allocated, the least recently used register among all registers is allocated for this line. This method results in variable number of registers allocated to each set. Each register can be thought of as a way in set associative cache. The proposed model is simulated with SPEC2K benchmarks. A performance improvement of 3% is seen.

The rest of the paper is organized as follows. Section II gives the motivation, Section III the proposed model, Section IV mathematical analysis of the proposed model, Section V simulation, Section VI conclusion, acknowledgement and references.

II. Motivation

Consider the address trace 2, 4, 6, 3, 8, 5, 4, 6, 2. Consider two way set associative cache of two sets. The addresses are mapped to sets 0, 0, 0,1,0,1,0,0,0. The address 2 is missing. It is placed in set zero, way zero. The address 4 is miss. It is placed in set zero, way one. The address 6 is missing. It is placed in set zero, way zero. The address 3 is missing and placed in set one, way zero. The address 8 is missing and placed in set zero, way one. The address 5 is missing. It is placed in set one, way one. The address 4 is missing. It is placed in set zero, way zero. The address 6 is missing and placed in set zero, way one. Thus there is 100% miss in the address trace. Consider the following algorithm.

Assume the cache is set of fast registers. Let there be four registers. On access

1. If there is a free register, place the cache line in free register
2. In case of no free registers, choose the LRU register among the registers allocated to this set. Place the cache line in it.
3. In case of mapping to new set, choose the LRU register among the whole register set. Place the cache line in it.
4. Stop.

According to this algorithm, new registers are allocated for addresses 2, 4, 6, 3. Address 8 replaces the register allocated to address 2. Address 5 replaces the register allocated to address 3. Address 4 is hit. Address 6 is hit. Address 2 replaces the register allocated to 8. There are two hits in this case. An improvement of hits in seen. This is the motivation of this paper.

III. Proposed Model

Consider cache to be set of fast registers. Let there be C registers at level one cache. Assume there are S sets. Consider an address a. Consider the following address mapping algorithm.

Fig. 1. Architecture of the proposed system

1. Start
2. Compute a % S.
3. Check among the allocated registers if there is match for this address. If found, it is cache hit. Access the line and stop.
4. Allocate a free register to this line. Mark the allocated register to represent this set.
5. Stop.

This is the case of no free registers. Allocate the LRU

Manuscript received March 17, 2013; revised May 20, 2013.

S. Subha is with Vellore Institute of Technology, Vellore, T.N, 632014 India(email:ssubha@rocketmail.com).

register among the registers allocated to this set. Stop. In case it is the first access to a set, allocate the LRU register among all the allocated registers to this line. Stop.

The proposed model is depicted in Fig. 1.

The proposed model should not be confused with the concept of fast registers as caches. In the case of fast registers as caches, there are fixed number of ways in all sets. In the proposed model, the registers are allocated as cache ways on demand. There could be a case where certain sets are absent in the cache.

IV. Mathematical Analysis of Proposed Model

Consider w-way set associative cache of S sets. Let this system be denoted by C_{trad} . Consider an address trace of R references. Let H be the number of hits. Let t be the hit time and M miss penalty. The average memory access time (AMAT) is given by

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

Next consider the following model. Consider a cache of wS fast registers with S sets.

1. If there is a free register, place the cache line in free register
2. In case of no free registers, choose the LRU register among the registers allocated to this set. Place the cache line in it.
3. In case of mapping to new set, choose the LRU register among the whole register file. Place the cache line in it.
4. Stop.

According to the above algorithm, let h be the number of hits. Let T be the cache hit time, M the miss penalty. The circuitry needed to map to the registers allocated to a set is assumed to be present. The AMAT of this system is given by

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

A performance improvement in AMAT is seen if

$$\frac{1}{R}(hT + (R-h)M) <= \frac{1}{R}(Ht + (R-H)M) \quad (3)$$

Consider the energy consumption. Let E be the energy consumed per cache line. Assume the cache operates in two modes – high power mode and low power mode. The cache is in low power mode by default. In traditional w-way set associative cache, one set is enabled to high power mode per access. The total energy consumed during cache operation is wRE. In the proposed system the set number is searched for per reference. Let the number of registers searched for each of R references be $w_1, w_2, .., w_R$ respectively. The total energy consumed by the cache system during operation is given by $E(w_1 + w_2 + .. + w_R)$. An improvement in energy consumption is seen if

$$E(w_1 + w_2 + .. + w_R) <= \text{wRE} \quad (4)$$

V. Simulation

The proposed model is simulated with the parameters shown in Table I. Routines in C language were written to simulate the model. The level one cache is simulated for the proposed design. The cache system is assumed to have one level. The model is compared with set associative cache of same size. The AMAT is shown in Fig. 2. As seen from Fig. 2 there is 3% improvement in AMAT.

TABLE I: List of Parameters

S.No	Parameter	Value
1	Register file size	8192
2	Line size	32 bytes
3	L1 access time/Register access time	3 cycles
4	Miss penalty	50 cycles
5	L1 cache size in set associative cache	1024
6	Associativity in set associative cache	8

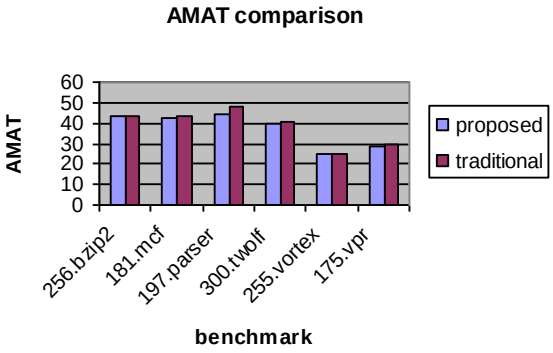

Fig. 2. AMAT comparison

VI. Conclusion

A variable way cache model is proposed in this paper. The cache is assumed to be set of fast registers and of fixed number of sets. An address is mapped to a set and allocated a free register on cold miss. The LRU policy is adapted in case of no free registers among the registers in the set for allocated set and all registers in case of new set access. The proposed model is simulated with SPEC2K benchmarks. A performance improvement of 3% is seen in AMAT.

Acknowledgment

The author expresses thanks to Santa Clara University, Santa Clara, CA, USA for providing SPEC2000 benchmarks and Simplescalar Toolkit.

References

[1] A. J. Smith, "Cache memories," *Computing Surveys*, vol. 14, no. 3, pp. 473-530, September1982

[2] R. E. Aly, B. R. Nallamilli, and M. A. Bayoumi, "Variable-way set associative cache design for embedded system applications," in *Proc.*

46th Midwest symposium on circuits and systems, 2003, pp.1435-1438.

[3] D. A. Patterson and J. L. Hennessey, *Computer System Architecture : A Quantitative Approach*, 3rd edition, Morgan Kaufmann Publishers Inc., 2003.

[4] D. H. Albonesi, "Selective Cache ways: on demand cache resource allocation," in *Proc. 32nd International Symposium on Microarchitecture*, 1999, pp. 248-259.

[5] M. K. Qureshi, D. Thompson, and Y. N. Patt, "The v-way cache:demand based associativity via global replacement," in *Proc. 32nd International Symposium on Computer Architecture*, 2005, pp. 544-555.

S. Subha graduated with Ph.D in computer engineering from Santa Clara University, Santa Clara, CA, USA in 2010.

She has worked in software industry in USA. She is currently working at Vellore Institute of Technology, Vellore, India. She does research in computer architecture.