ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(6), DOI: 10.17485/ijst/2016/v9i6/80229, February 2016* ISSN (Online) : 0974-5645

A Power Saving Set Associative Cache Model

Srinivasan Subha*

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

Abstract

Background/Objectives: Placing conflict cache lines in alternate set is proposed for set associative caches in literature. The conflicting lines are placed in one alternate set for each set. **Methods/Statistical Analysis:** This paper proposes architecture to place conflicting lines in set with maximum vacant ways in set associative caches. The model introduces one register per cache way with number of bits equal to number of cache sets. A sequential circuit using the register enables the cache ways of any set. The unoccupied ways are disabled. The proposed model is simulated with SPEC2K benchmarks. The power consumption is calculated using Quartus 2 tool and verilog code. **Findings:** An average improvement in power consumption of 68% for benchmarks with partial filled cache ways is observed with average memory access degradation of about 4.86%. A performance degradation in average memory access time of about 34% compared with direct mapped cache system and 3% compared with random placement algorithm is observed for the proposed system. The proposed model is for power consumption as is validated from the findings. The proposed model can be adapted for set associative caches to improve power consumption. **Applications/Improvements:** A model with optimal operational cache sets with methods to decrease power dissipation of sequential circuits, low power techniques for digital circuits can be used to improve the power consumption.

Keywords: Average Memory Access Time, Conflict Miss, Power Saving, Set Associative Caches, Sequential Circuit

1. Introduction

A cache is denoted by tuple (C, k, L) where C is the capacity, L the line size and k associativity. Caches are of three kinds-direct mapped, set associative and fully associative. A line can be placed in any place in fully associative cache, it can occupy fixed place in direct mapped cache while it can be placed in any of k ways in k-way set associative cache[1,2]. In set associative cache, all the sets are enabled during cache operation. A line if present in cache is called cache hit else it is cache miss. Cache misses are of three kinds - cold, capacity and conflict[2]. A cache miss for first time access is called cold miss. The capacity miss the difference in miss caused between the cache and fully associative cache of same size. Conflict misses are misses caused by line mapping to filled set in set associative cache. Selective enabling of cache sets is proposed in literature[3,4]. In[3] cache ways are selectively enabled. In[4] sequential circuit is introduced in the cache system to enable only occupied cache ways. Algorithms and models are proposed to extend the cache ways to other sets during conflict misses[5–7]. Assuming the cache operates in two modes - high power mode and low power mode the number of cache ways in high power mode increases in this case. Algorithms to reduce the power consumption are proposed in literature[8]. In[7] the cache ways are extended to another set during cache set conflict. The other set is chosen as the set with maximum vacant ways. Technique to selectively enable cache ways and sets is proposed in[9]. A technique to utilize less stressed sets to place conflicting lines is proposed in[10]. A technique to divide the cache into modules, selectively switching off ways in each module saving energy with small performance degradation is proposed in[11]. The authors in[12] propose an adaptive hybrid cache to dynamically remap scratchpad memory blocks from high-demand cache sets to low-demand cache sets.

This paper proposes an algorithm to extend the cache ways to number of other sets during conflict. The other

* *Author for correspondence*

set is chosen as the set with maximum vacant ways. A vector is maintained per set to indicate the associated sets. A vector is present for each cache way denoting the set the way belongs. The occupied cache ways are enabled using sequential circuit. All other ways are disabled. The proposed model is simulated with SPEC2K benchmarks in uniprocessor environment. A power saving of 68% is observed with Average Memory Access Time (AMAT) degradation of about 4.86% for benchmarks with partial cache way occupancy. A performance degradation in average memory about 34% compared with direct mapped cache system and 3% compared with random placement algorithm is observed for the proposed system.

The proposed system can be implemented in any processor using set associative cache. It finds applications as suggested in[13–15]. The power consumption strategies proposed in[16] can be studied for power improvement. The method suggested in[17] to reduce sequential circuit power dissipation can be considered as part of future work.

2. Motivation

Consider two level inclusive cache system. Let level one be four way set associative cache of 1024 sets and level two be eight way set associative cache of 2048 sets with line size of 32 bytes. The SPEC2K benchmarks were run on this configuration using Simplescalar Toolkit. The entire cache is high power mode during operation. Consider the following algorithm for line placement extending the algorithm proposed in[7].

- Start.
- If there is vacant position in mapped set, place the line in the set and stop.
- Choose the set with maximum vacant ways as the set to place the line. Place the line in this set and stop.
- Place the line in least recently used way of mapped set and stop.

The above cache system has owner vector per cache way indicating the way ownership, other_set vector per cache set indicating the associated sets having the conflict cache lines. This is shown in Figure 1. The number of enabled cache ways in level one cache is shown in Table 1.

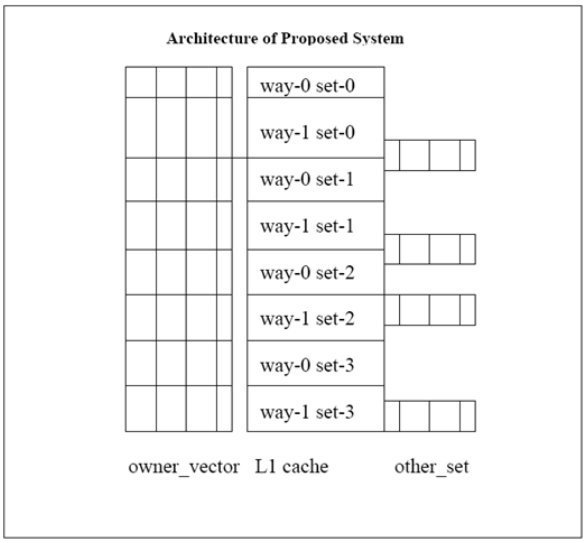

Figure 1. 2-way set associative cache of four sets. There are four other_set vectors each four bits. There are eight owner vector entries each four bits.

As seen from Table 1 only 38% of cache ways are occupied in the simulation. The entire cache system is in high power mode during operation. Next consider the following algorithm. The cache lines are initially in off mode. On occupancy they are enabled. This is achieved by introducing sequential circuit at way level. The circuit in Figure 2 gives the sequential circuit for one owner_ vector entry. One sequential circuit is introduced for

Table 1. Way utilization in L1 cache

Name	eways	Way utilization	Power(prop W)	Power (trad W)	%improve
256.bzip2	4096	100	523.8423632	199.591332	-162.457
181.mcf	420	10.2539063	54.6311064	199.591332	72.62852
197.parser	3513	85.7666016	449.4271938	199.591332	-125.174
300.twolf	437	10.6689453	56.801017	199.591332	71.54134
255.vortex	549	13.4033203	71.0968986	199.591332	64.37876
175.vpr	549	13.4033203	71.0968986	199.591332	64.37876
	average	38.9160156			
	Avg. Power	partial filled	cache ways		68.23185

each owner_vector. In the simulated example thus there are 4*1024 such sequential circuits. The number of bits in each owner_vector is 1024.

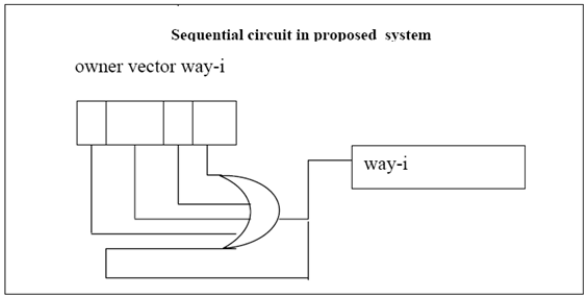

Figure 2. Sequential circuit enabling way-i depending on owner vector and way-istatus.

The power consumed by the cache system is calculated from the parameters given in Table 2. These values were obtained by simulating the circuit in Quartus 2 software. The power consumed in traditional cache system and proposed cache system is shown in Table 1. As seen from Table 1 there is 68% improvement in power consumption for the partially filled cache ways. This is the motivation of the paper.

Table 2. Power parameter values

Total Thermal Power per cache way in L1 cache	0.0357mW
Total Thermal Power per cache way in L2 cache	0.0356mW
Total Thermal Power per sequential owner circuit of three entries	146.35mW
Total Thermal Power for sequential owner circuit for additional bit	0.12mW
Total Thermal Power for other_set vector	146.13mW
Total Thermal Power for additional two input OR gate	0.33mW
Total Thermal Power for other_set vector	146.13mW

3. Proposed Model

Consider two level inclusive cache system in uniprocessor environment. Let level one cache be $w_1$way set associative cache of sets. Let level two cache be w_2 way set associative cache of sets. Let the cache line size be L bytes. Consider address trace of size R. Consider level one address mapping. Each set has vector called other_set. The number of entries in other_set entry is equal to number of cache sets. If a line mapped to set is present in set j, the jth entry of other_set of set-i is set to one. Else it is set to zero. If cache has S sets there are S other_set vectors one per set. Each vector has S entries of one bit. Each cache way has vector called owner_vector. The number of entries is equal to number of cache sets. Each cache way has owner_vector. For level one cache of S sets, the total number of owner_vectors is w_1S one per way. Each owner vector has S bits one per set. If line in cache way i belongs to set j, the jth entry of the owner_vector at index i is set to one. Else it is set to zero. The mapped cache lines of set i can be placed either set i or any other set in this architecture. The following algorithm is used to place line in level one cache.

- Start.
- Determine the set and tag values of the line s, t at level one.
- If the line is found in set s, increment number of hits, access line and stop.
- If the line is found in any other set, increment number of hits, access line, stop.
- This is miss condition. Increment number of misses.
- If there is vacant place in set s, place the line in set s, access line, and stop.
- Determine the set with maximum vacant ways. Choose this set to place the line. Access the line. Update other_set vector of s, owner vector of mapped way and stop.
- Place the line in least recently used way of mapped set s. Update other_set vector of s, owner vector of mapped way, access the line and stop.

The above algorithm is adopted for traditional cache denoted as C_{trad}. In step 7 of the above algorithm the cache with maximum vacant ways is chosen to place the line in case of miss. The intuition behind this is to accommodate lines in sets that are having less vacant ways. It is observed that only subset of all sets in level one cache is occupied as seen from Table 1. In order to accommodate maximum number of lines, this step is executed. In the above model, all the cache ways are enabled during cache operation. Next consider the proposed model denotes as C_{prop}. The address mapping is as listed above in this model. All cache ways are disabled initially. On occupancy, the cache way is enabled. This is achieved by introducing sequential circuit as shown in Figure 2. A cache line is enabled if it belongs to any set or already enabled. This is achieved by OR'ing the owner vector entries to determine the occupancy and

feedback for retaining power. The other_set vector entry i is set to one if set-i contains the conflict line for any set j. The owner vector entry i is set to one if the line belongs to set i.

The architecture of proposed model is shown in Figure 1.

The proposed model may not give fair chance to sets with same number of vacant ways during line placement.

The proposed model differs from fully associative cache. The whole address is not stored in the proposed model. Instead the tag value along with owner vector and other_set vector is stored in the proposed architecture. The owner vector has one bit per set. One owner vector has S bits for cache of S sets. The size of other_set vector is S bits per other_set vector entry.

4. Mathematical Analysis of Proposed Model

Consider two level cache system described in Section 3. Let level one cache be w_1 way set associative cache of S_1 sets. Let level two cache be w_2 way set associative cache of S_2 sets. Let the cache line size be L bytes. Consider address trace of size R. Let the traditional cache be denoted as C_{trad}. Let the proposed model be denoted as C_{prop}. Let h_1, h_2, t_1, t_2, t_{12}, m be number of level one hits, number of level two hits, level one access time, level two access time, transfer time between level one and level two, miss penalty respectively in traditional cache. The Average Memory Access Time (AMAT) in traditional cache is given by

$$AMAT(C_{trad}) = \frac{1}{R}\left(h_1 t_1 + h_2(t_1 + t_2 + t_{12}) + (R - h_1 - h_2)m\right) \quad (1)$$

Consider the proposed model. Let H_1, H_0, H_2 be the number of level one cache hits in the mapped set, level one cache hits in other set, level two cache hits. Let p number of other sets be accessed for q references. Let T_1, T_{12}, T_2, M be level one access time, transfer time between level one and level two, level two access time and miss penalty in the proposed model. The AMAT is given by

$$AMAT(C_{prop}) = \frac{1}{R}\left(H_1 T_1 + H_o(T_1 + T_o) + H_2(T_1 + T_o + T_2 + T_{12}) + (R - H_1 - H_o - H_2)M\right) \quad (2)$$

where $T_o = \frac{p}{q}$

The first term in Equation (2) is the time for level one hits in mapped set. The second term is the time for level one hits in other set than the mapped set. The third term is the time for level two hits. This involves accessing level one in the mapped set, in other sets, access level two cache, transfer between level one and level two. The transfer time between level two and level one cache involves the time to choose the level one cache way for placing the line. The fourth term is the time taken in case of cache miss. An improvement in AMAT is observed if

$$\frac{1}{R}\left(h_1 t_1 + h_2(t_1 + t_2 + t_{12}) + (R - h_1 - h_2)m\right) >= \frac{1}{R}\left(H_1 T_1 + H_o(T_1 + T_o) + H_2(T_1 + T_o + T_2 + T_{12}) + (R - H_1 - H_o - H_2)M\right) \quad (3)$$

In both the cases the level one cache access time is the time taken to access the mapped set and other_set as the case may be.

Next consider the power consumption. In the traditional cache the entire cache is in high power mode during operation. Let W be the power consumed per cache way during cache operation. The total power consumed in the traditional cache is given by

$$P(C_{trad}) = W(w_1 S_1 + w_2 S_2) \quad (4)$$

Consider the proposed model. A line is occupied if it belongs to the mapped set or it is from other set. This is given by the corresponding bit set to one in the owner vector. For owner vector i, the power consumed is given by

$$P(owner_i) = W\sum_{j=1}^{S_1} o_{ij} \quad (5)$$

Where o_{ij} is the value of bit j in owner vector of i-th cache way and summation is the logical OR of the bits. The total power consumed by the entire cache system is given by

$$P(C_{prop}) = W\sum_{i=1}^{w_1 S_1} P(owner_i) + w_2 S_2 W + E \quad (6)$$

Where E is the power consumed by additional circuitry. The first term in Equation (6) is the power consumed by the occupied cache ways in level one cache. The second term is the power consumed by level two cache. An improvement in power consumption is observed if

$$W(w_1 S_1 + w_2 S_2) >= W\sum_{i=1}^{w_1 S_1} P(owner_i) + w_2 S_2 W + E \quad (7)$$

Consider the additional space requirements of the proposed model at level one. For fully associative cache the address along with data is stored in cache blocks. Consider w_1 -way set associative cache of S_1 sets. Assume fully associative cache of w_1S_1 blocks. Let the address be a bytes and data be b bytes. The total space in fully associative cache is $w_1S_1(a+b)$ bytes. In the proposed model for t tag bytes (t<a) the total number of owner vector bits is $w_1 S_1^2$. The total number of other_set vector bits is S_1^2. The total number of additional bytes is $\frac{1}{8}(w_1S_1^2+S_1^2)$. The total space is $w_1S_1(t+b)+\frac{1}{8}(w_1S_1^2+S_1^2)$ at level one. Assuming the tag size is two thirds of address length, an improvement in space over fully associative cache is observed if $a>\frac{3S_1(1+w_1)}{8w_1}$.

5. Simulation

The proposed model is simulated with SPEC2K benchmarks in uniprocessor environment. The simulation parameters are shown in Table 3. Addresses in SPEC2K benchmarks were collected using Simplescalar Toolkit. The proposed model was simulated using C language routines. The hits and misses in two level cache were collected. The AMAT values were calculated. The miss penalty of the proposed system includes the time taken to find the cache way for placing the line. This includes the time to find the vacant ways. The time to transfer between level one and level two includes the time to find the cache way to place/replace the level two cache line in level one cache. As seen from Figure 3 the AMAT degrades by about 4.86%. The AMAT for direct mapped cache was simulated for cache system of same capacity. As seen from Figure 3 there is degradation in AMAT of about 34% for the proposed system compared with direct mapped cache system. The AMAT for cache system with random choice of cache ways to place the conflict lines was simulated. The results are shown in Figure 3. As seen from Figure 3 there is degradation in AMAT in proposed system by about 3% compared with the random placement. The proposed model was simulated in hardware using Verilog in Quartus 2. The power consumed by the cache way, additional circuit was obtained from this as shown in Table 2. The power consumed is taken from the Power Play Power Analyzer tool of Quartus 2. The power values collected is the total power dissipation value. This includes the core dynamic thermal power dissipation, core static thermal power dissipation and I/O thermal power dissipation. The power consumed to set bits to one from zero or zero from one for the owner vector and other_set vectors is zero from this tool. For calculating the power for the simulated benchmarks, the values from Table 2 are taken. For n components of similar type the power consumed is calculated as below.

Power (n components) = Power (first component) + (n-1) Power (additional component)(8)

The power is the total thermal power consumed obtained from Quartus 2. For R references the power consumed is calculated as given below.

(Total number of enabled ways in level one cache + number of ways in level two cache)* power per cache way + power for the owner_vector for all ways + power for the other_set vector for all sets + power for sequential circuit (9)

The power for sequential circuit is given below

Power for first sequential circuit + (total number of cache ways -1)* incremental power for sequential circuit (10)

The total power consumed is shown in Figure 4. As seen from Figure 4 there is 68% improvement in power consumption considering partially filled cache entities. The power consumed by direct mapped cache model, random placement cache model is same as that of the traditional model as the entire cache system is enabled during cache operation. The AMAT and power consumed are shown for the SPEC2K benchmarks for which the binaries were built using Simplescalar Toolkit.

Table 3. Simulation parameters

Parameter	Value
L1 Size	1024 sets
L1 associativity	4
L2 Size	2048 sets
L2 associativity	8
Line size	32 bytes
L1 access time	3 cycles
L2 access time	12 cycles
L1 to L2 transfer time in proposed cache	22 cycles
L1 to L2 transfer time in traditional cache	20 cycles
Miss penalty in traditional cache system	65 cycles
Miss penalty in proposed cache system	70 cycles

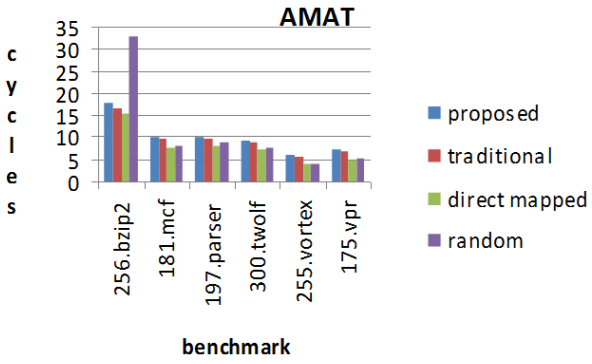

Figure 3. AMAT comparison.

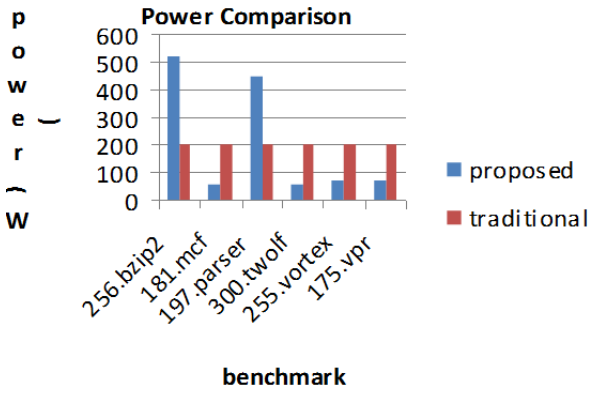

Figure 4. Power consumption comparison.

6. Conclusion

A model to reduce power consumption in variable ways cache architecture spanning over many sets is proposed in this paper. The proposed model is simulated with SPEC2K benchmarks in C language and Verilog language in Quartus 2 tool. An average improvement in power consumption of 68% for benchmarks with partial filled cache ways is observed with average memory access time degradation of about 4.86%. A performance degradation in average memory about 34% compared with direct mapped cache system and 3% compared with random placement algorithm is observed for the proposed system.

The proposed architecture can be improved to have simpler circuit for cache ways enabling. A model with optimal number of enabled cache sets can be developed as part of future work.

7. Acknowledgment

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

8. References

1. Smith AJ. Cache Memories. Computing Surveys; 1982. p. 473.
2. Patterson DA, Hennessey JL. Computer system architecture: A quantitative approach. 3rd ed. CA: Morgan Kaufmann Publishers Inc; 2003.
3. Albonesi DH. Selective cache ways: On demand cache resource allocation. 32nd Annual International Symposium on Microarchitecture MICRO'99; Haifa. 1999. p. 248–59.
4. Subha S. A reconfigurable cache architecture. Proceedings of ICHPCA; 2014. p. 1–5.
5. Aly RE, Nallamilli BR, Bayoumi MA. Variable-way Set associative cache design for embedded system applications. Proceedings of IEEE Symposium on Circuits and Systems; 2003. p. 1435–8.
6. Subha S. An algorithm for variable cache ways. Proceedings of International Conference on Advances in Technology and Engineering ICATE; Mumbai. 2013. p. 1–3.
7. Subha S. A cache architecture. Proceedings of 15th International Conference on Advanced Computing Technologies, ICACT; 2013. p. 1–4.
8. Subha S. A power saving varying cache ways architecture. Journal of Theoretical Applied Information Technology. 2015; 76(3):309–13.
9. Yang SH, Powell M, Falsafi B, Vijaykumar TN. Exploiting choice in resizable cache design to optimize deep-submicron processor energy-delay. Proceedings of 8th International Symposium on High-Performance Computer Architecture, HPCA; 2002. p. 151–61.
10. Rolan D, Fraguela BB, Doallo R. Adaptive line placement with set balancing cache. Proceedings of 42nd Annaul IEEE/ACM International Symposium on Micro architecture, MICRO; 2009. p. 529–40.
11. Mittal S, Zhang Z, Vetter JS. Flexiway: A cache energy saving technique using fine-grained cache reconfiguration. Proceedings of International Conference on Computer Design ICCD; Asheville, NC. 2013. p. 100–7.
12. Cong J, Gururaj K, Huang H, Liu C. An energy efficient adaptive hybrid cache. Proceedings of International Symposium on Low Power Electronics and Design, ISPLED; Fukuoka. 2011. p. 67–72.
13. Sinha N, Alex JSR. IoT based iPower saver meter. Indian Journal of Science and Technology. 2015 Aug; 8(19).
14. Sindhuja P, Balamurugan MS. Smart power monitoring and control system through internet of things using cloud date storage. Indian Journal of Science and Technology. 2015 Aug; 8(19).
15. Madhubala JS, Umamakeswari A. A vision based fall detection system for elderly people. Indian Journal of Science and Technology. 2015 May; 8S(9).
16. Verma G, Kumar M, Khare V. Low power techniques for digital system design. Indian Journal of Science and Technology; 2015 Aug; 8(17).
17. Neelgar BI, Benakop PG. Reduction of power dissipation in sequential circuits. Indian Journal of Science and Technology. 2006 Aug; 1(3).