

# A Varying Processor Cache Sets Architecture



S Subha

**Abstract:** Any processor cache has three parameters capacity, line size and associativity. Usually all three are fixed at design time. Algorithms to have variable cache sets are proposed in literature. This paper proposes a method to have variable cache sets logically. The cache comes with fixed sets. The cache is visualized to have logically any number of sets greater than or equal to one. An algorithm for line placement/replacement is proposed in this paper for this model. The proposed model is simulated with SPEC2K benchmarks using SimpleScalar Toolkit for two level inclusive set associative cache system. A power saving of 8.4% for L1 cache size 512x4, 17.58% for 1024x4 and 31.3% for 2048x4 is observed compared with traditional set associative cache of same size. A power saving of 7.53% compared with model proposed in literature for L1 size 512x4, 7.64% for 1024x4 and 7.645% for 2048x4 is observed. The L2 cache size is fixed at 2048x8. The average memory access time (AMAT) is found to degrade compared with conventional set associative cache by 19.63% for L1 size of 512x4, 24.68% for 1024x4 and 2048x4. (Abstract)

**Keywords :** Average memory access time, power, set associative cache, variable processor cache sets

## I. INTRODUCTION

Processor caches are of three types direct mapped, set associative and fully associative. A cache is identified by its capacity, associativity and line size. This is denoted by tuple (C,k,L) where C is the capacity, k the associativity and L line size. A line is placed in fixed location in direct mapped cache. It can occupy any of the k ways of mapped set in k-way set associative cache. It can occupy in any of n blocks in fully associative cache of n blocks. Direct mapped and set associative caches have fixed number of sets. A fully associative cache has one set. However the number of occupied cache sets is less than the cache capacity [1,2].

Algorithms to enable cache sets selectively is proposed in literature . The authors in [4] propose Set Balancing Cache. In this lines are moved from stressed sets to underutilized sets based on saturation counter value. The move can be static or dynamic. However all sets are enabled in this model. The authors in [3] propose Smart Cache. This model puts off unused sets. There is remapping of dirty lines in this model. The authors in [5] propose reconfigurable cache. However the number of sets is power of two in this case. The author in

[6] proposes an algorithm to have variable cache sets. However the number of cache sets is doubled on conflicts. A set can be put off using sequential circuit in this model. The author in [7] proposed variable cache architecture. In this model the cache starts with one set and increases number of active sets as needed. A register per set has entries one bit per cache set. The occupied set entry in the register is set to one. This is AND'd with logic one pointing to the way containing the line. The least recently used (LRU) algorithm is used to replace lines if cache is full.

A varying cache set architecture is proposed in this paper. The proposed model assumes that cache to have logically infinite sets limited by max value practically. The maximum number of sets, varset, is input for any address trace. The varset is less than or equal to max. Each set has register R with max one bit entries. An address is mapped taking varset number of sets. The cache starts physically with one set called numset. The index corresponding to the calculated value entry in R is set to one. This entry is AND'd with logic 1 to point to the way the line is placed. When the set is full, new set is added to the list of active sets by incrementing numset. If all sets are active and numset is greater than the physical cache size, the least recently used way is used for line placement. The proposed model is simulated with SPEC2K benchmarks with SimpleScalar Toolkit for two level set associative inclusive cache system. The L1 cache sizes of 512x4, 1024x4 and 2048x4 and L2 cache size of 2048 x 8 set with line size of 32B is used for simulation. The proposed model is compared with traditional set associative cache of same size called Trad and the model proposed in [7] called pprop1 for same cache sizes. A power saving of 8.4% for L1 cache size 512x4, 17.58% for 1024x4 and 31.3% for 2048x4 is observed compared with traditional set associative cache of same size. A power saving of 7.53% over model proposed in [7] for L1 size 512x4, 7.64% for 1024x4 and 7.645% for 2048x4 is observed. The average memory access time (AMAT) is found to degrade compared with conventional set associative cache by 19.63% for L1 size of 512x4, 24.68% for 1024x4 and 2048x4. The AMAT remained same for model proposed in [7].

The rest of paper has the following. Section 2 gives motivation, section 3 proposed model, section 4 mathematical model, section 5 simulations, section 6 conclusion followed to references.

## II. MOTIVATION

Consider two level inclusive set associative cache. Let level 1 be 4-way set associative cache, level 2 be 8-way set associative cache of 2048 sets with line size of 32B. A set is defined active if atleast one way is occupied.

Manuscript published on 30 September 2019

\* Correspondence Author

S Subha\*, Department of IT, School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, T.Nadu, India. Email: [ssubha@rocketmail.com](mailto:ssubha@rocketmail.com)

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

## A Varying Processor Cache Sets Architecture

Consider the following algorithm for placement/replacement in this cache system. Assume the L1 cache can be of any size till 2048 sets. It is assumed that logically the cache is of infinite size limited by physical capacity. Let  $a$  be the address for placement in the cache system.

1. Start
2. Input the cache size in number of set say varset2
3. Start with one set say numsets.
4. Calculate  $index1 = a \% varset2$  tagval =  $a \div varset2$ .
5. Put it in the active set. Assume there is one register per set with maximum allowed number of sets. In this case this register is of 2048 bits. Make the index1 bit of the register as one and AND it with logic 1 to point to the way containing the line.
6. If the set is full, include another set by incrementing numsets in the list of active sets and proceed as before for the rest of the addresses.
7. If the numsets is greater than the physical number of sets in the cache, assume there is varset2 number of sets, perform the address translation. Place the line in the least recently used line of the cache.
8. Stop

The above algorithm assumes that if the numsets variable exceeds the L1 cache limit, the line is placed in the LRU way of the entire cache with address mapping assumed to have varset2 number of sets. This model differs from the model in [7] in that the number of active sets can be lesser or greater than the number of physical sets. The model is limited by assuming that the maximum allowed number of sets is set to fixed value. The above algorithm is simulated with SPEC2K benchmarks. The number of active sets is found to decrease compared with the traditional set associative cache model called as Trad in this paper. The improvement in power because of this is shown in Table 1. The AMAT increase is shown in Table 2. The label pprop1 refers to the variable cache set architecture proposed in [7].

As seen from Table 1 there is average improvement in power consumption by 7% over model proposed in [7]. From Table 2 it is concluded that AMAT increases by 24% compared with Trad model. The AMAT is same as in pprop1 model proposed in [7].

Table I Power comparison chart for 4-way set associative 11 cache with 1024 sets

benchmark	%power trad	%power pprop1
181.mcf	17.9329	6.74680
300.twolf	17.8352	6
255.vortex	4	7.77466
	17.2883	8

	17.2883	8.84663
175.vpr	6	2
	17.5862	7.64001
average	2	8

Table II AMAT comparison chart for 4-way set associative cache with 1024 sets

benchmark	%amat trad	%amat pprop1amat
	18.6860	
181.mcf	7	0
300.twolf	1	0
255.vortex	9	0
175.vpr	6	0
	24.6831	
average	6	0

As seen from Table 1 there is average improvement in power consumption by 7% over model proposed in [7]. From Table 2 it is concluded that AMAT increases by 24% compared with Trad model. The AMAT is same as in pprop1 model proposed in [7]. The decrease in power consumption is the motivation of this paper.

### III. PROPOSED MODEL

Consider two level inclusive cache system. Both levels are set associative caches. Let the line size be  $L$  bytes. Assume L1 is  $w_1$  way set associative cache of  $S_1$  sets and L2 be  $w_2$  set associative cache of  $S_2$  sets. Consider the address trace of  $R$  references. Consider the following algorithm for line placement/replacement. Assume the L1 cache can have logically infinite number of sets. For practical purposes assume the maximum number of sets is  $r$ . A  $r$ -bit register  $R$  in any set indicates the enabled sets in the set. The fixed length of the register limits the model. Let  $a$  be the address to be searched in the cache.

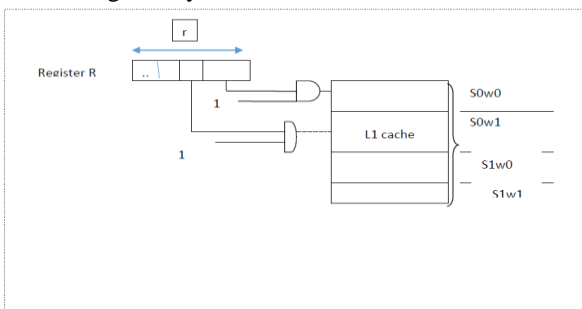
1. Start
2. Input the cache size in number of set say varset2  $\geq 1$
3. Start with one set. Let numset indicate number of active sets.
4. Calculate  $index1 = a \% varset2$  tagval =  $a \div varset2$ .
5. Search for line in active sets. If present, it is L1 hit. Access the line, update the LRU counter and stop.
6. Search for the line in L2. If present it is L2 hit. Access the line, put the line in L1 by steps 8-10. Update the LRU counter of L2 and stop.

7. It is L2 miss. Put the line in L2. Do steps 8-10 and stop.
8. Put the line in the active set. Make the index1 bit of the register as one and AND it with logic 1 to point to the way containing the line. Update LRU counters in L1 and L2 and stop.
9. If the set is full, include another set by incrementing numset in the list of active sets. If the numset is less than or equal to number of physical sets, do steps 8-9.
10. If the numset is greater than the number of physical sets in the cache, assume there is varset2 number of sets. Place the line in the least recently used line of the L1 cache. The value of varset2 is limited by the value of r.
11. Stop

The above algorithm assumes that the number of cache sets is variable from one to r. The model differs from [7] in that the maximum number of sets is the number of physical sets in cache. The model is depicted in Fig. 1. As shown in Fig. 1 the cache has one register called R per set. The number of bits in R is equal to r. A bit in position i is set to one if there a line corresponding to set number i in any set. An AND gate enables the ways occupied by any set in given set. The input to AND gate is the bit from R and logic one. The address mapping is same as existing cache models with the number of sets as the input. The line replacement algorithm is least recently used algorithm (LRU). The model can have multiple cache ways in different physical sets mapped to same physical set. The associativity of sets is varying because of this. Lines mapped to different sets can reside in same physical set. The number of occupied sets varies with time. The number of cache ways per set is constant. The cache line size is constant. The model differs from proposed model in [5] in that the number of sets need not be power of two. As the number of active sets varies with time, the number of active components varies with time. Hence, the energy consumed or power consumed varies with time and intuitively should decrease compared with caches which are enabled for all sets and ways.

#### IV. MATHEMATICAL ANALYSIS OF PROPOSED MODEL

Consider the model proposed in section 3. Let the traditional model be denoted  $C_{trad}$  and the proposed model  $C_{prop}$ . Let the cache system have inclusive two cache levels L1 and L2. The average memory access time (AMAT) is chosen as performance metric. This is because the L1 cache is the proposed model and L2 cache is w-way set associative cache. Let  $h_1, h_2, t_1, t_2, t_{12}, m$  be level one hits, level two hits, level one access time, level two access time, transfer time between level one and level two in the traditional cache. The AMAT is given by



**Fig.1 Architecture of proposed system. The cache has max of r sets. Register R has one bit per each cache set. An AND gate with logic one as one of the inputs enables each cache way of the set. The value Siwj indicates set-i way-j.**

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

The first term in (1) is the level one hit time, second term is the level two hit time and third term is the miss time.

Let  $H_1, H_2, T_1, T_2, T_{12}, M$  be level one hits, level two hits, level one access time, level two access time, transfer time between level one and level two in the proposed cache. The average memory access time is given by

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

The first term in (2) is the level one hit time, second term is the level two hit time and third term is the miss time.

An improvement in AMAT is observed if

$$\begin{aligned} & \frac{1}{R} (h_1 t_1 + h_2 (t_1 + t_2 + t_{12}) + (R - h_1 - h_2) m) \\ & >= \frac{1}{R} (H_1 T_1 + H_2 (T_1 + T_2 + T_{12}) + (R - H_1 - H_2) M) \end{aligned} \quad (3)$$

Next consider the power consumption. Let  $P_{high}$  be the power consumed by cache way. The power consumed in traditional cache is given by

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

The first term in (4) is the number of ways in level one and the second term is the number of ways in level two cache.

Let  $S_1^1$  number of sets be enabled in level one in proposed model. Let  $P_r, P_a$  be the additional power consumed by the R registers and AND gates in the proposed model respectively. Then

$$P(C_{prop}) = (w_1 S_1^1 + w_2 S_2) P_{high} + P_r + P_a \quad (5)$$

The first term in (5) is the number of ways in level one. The second term in (5) is the number of ways in level two cache. An improvement in power consumption is observed if

## A Varying Processor Cache Sets Architecture

$$P_{high}(w_1S_1 + w_2S_2) \geq \frac{(w_1S_1^1 + w_2S_2)}{P_r + P_a} P_{high} + \quad (6)$$

### V. SIMULATIONS

The proposed model is simulated with parameters shown in Table 2.

**Table II Simulation parameters**

Parameter	Value
L2 size	2048 sets
Line size	32B
L1 sizes	512, 1024, 2048 sets
L1 associativity	4
L2 associativity	8
L1 access time in Trad cache	3 cycles
L1 access time in pprop1 cache	5 cycles
L1 access time in proposed cache	5 cycles
L1 to L2 transfer time in Trad cache	12 cycles
L1 to L2 transfer time in pprop1 cache	12 cycles
L1 to L2 transfer time in proposed model	12 cycles
Miss penalty	65 cycles
L2 access time in proposed cache	20 cycles
L2 access time in pprop1 model	20 cycles
L2 access time in trad model	20 cycles
Power per way	20 mW
Power for 2048 bit register and AND gate	66.86mW

The simulations are done for fixed cache associativity and line size. The pprop1 and proposed cache require in worst case to search all the occupied sets. The access time in level one is assumed to be five cycles for this purpose. Routines in C language were written to simulate the proposed model. The hits and misses in two cache levels were collected for SPEC2K benchmarks. The Fig. 2 to Fig. 5 gives the power saving of proposed model compared with Trad and pprop1 for various cache sizes.

Table III Power comparison for L1 size 512 x 4

L1: 512x4

benchmark	%trad	%pprop1
181.mcf	8.814328	6.331039
300.twolf	8.705821	7.191966
255.vortex	8.098183	7.774668
175.vpr	8.098183	8.846632
<b>average</b>	<b>8.429129</b>	<b>7.536076</b>

From Table 3 the power saving in proposed cache is observed for L1 cache size of 512 x 4. The L2 cache size is 2048 x 8.

Table IV Power comparison for L1 size 1024 x 4

L1: 1024x4

benchmark	%p_trad	%pprop1
181.mcf	17.9328955	6.746806
300.twolf	17.8352393	7.191966
255.vortex	17.2883643	7.774668
175.vpr	17.2883643	8.846632
<b>average</b>	<b>17.5862158</b>	<b>7.640018</b>

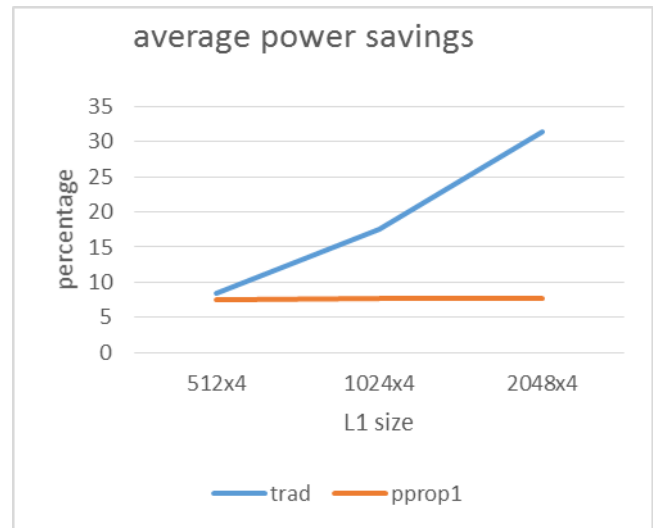
Table V Power comparison for L1 size 2048 x4

L1: 2048x4

benchmark	%trad	%pprop1
181.mcf	31.61075	6.767498
300.twolf	31.52937	7.191966
255.vortex	31.07364	7.774668
175.vpr	31.07364	8.846632
<b>average</b>	<b>31.32185</b>	<b>7.645191</b>

From Table 4 the power saving in proposed cache is observed for L1 cache size of 1024 x 4. The L2 cache size is 2048 x 8.

From Table 5 the power saving in proposed cache is observed for L1 cache size of 1024 x 4. The L2 cache size is 2048 x 8.



**Fig. 2 Average Power savings in proposed model**

From Fig. 2 it is observed that there is saving in average power consumption with increase in L1 size compared with Trad and pprop1 for the proposed model.

Tables. 6-8 show the AMAT increase for the proposed model compared with Trad. It is found that the AMAT is same w.r.t. pprop1. From the Fig. 3 it is seen that the average AMAT increases and reaches constant value for the chosen parameters compared with Trad.

Table VI AMAT degradation for l1 size 512x4

benchmark	%trad	512x4
181.mcf	15.74411	
300.twolf	16.71005	
255.vortex	24.67962	
175.vpr	21.39468	
average	19.63212	

Table VII AMAT degradation of l1 size 1024 x 4

benchmark	%amat trad	1024x4
mcf	18.6860698	
twolf	20.0625108	
vortex	32.7661879	
vpr	27.2178556	
average	24.683156	

Table VIII AMAT degradation for l1 size 2048 x 4

benchmark	%trad	2048x4
181.mcf	18.68607	
300.twolf	20.06251	
255.vortex	32.76619	
175.vpr	27.21786	
average	24.68316	

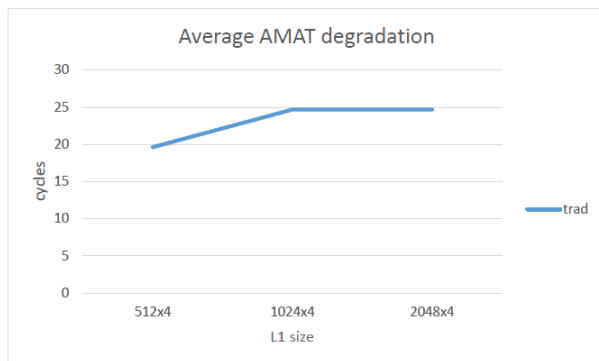


Fig. 3 Average AMAT increase

It is concluded from simulation results that there is saving in power consumption and increase in AMAT for proposed model.

## VI. CONCLUSION

A varying cache set architecture is proposed in this paper. The proposed model assumes that cache to have logically infinite sets limited by max value practically. An algorithm for placement and replacement of cache lines is proposed for this model. The proposed model is simulated with SPEC2K benchmarks with SimpleScalar Toolkit for two level inclusive set associative cache system. The level one cache is 4-way set associative of cache sizes of 512, 1024 and 2048 sets. The level two cache is 8-way set associative cache of 2048 sets. The proposed model is compared with traditional set associative cache of same size called Trad and the model pprop1 proposed in literature. A power saving of 8.4% for L1 cache size 512x4, 17.58% for 1024x4 and 31.3% for 2048x4 is observed compared with traditional set associative cache of same size. A

power saving of 7.53% compared with model proposed in literature for L1 size 512x4, 7.64% for 1024x4 and 7.645% for 2048x4 is observed. The L2 cache size is fixed at 2048x8. The average memory access time (AMAT) is found to degrade compared with conventional set associative cache by 19.63% for L1 size of 512x4, 24.68% for 1024x4 and 2048x4. Thus simulation results showed AMAT degradation in case of Trad grows linearly reaching saturation value with no change in pprop1. The power saving improved with cache sizes for Trad and pprop1 for the chosen parameters.

## ACKNOWLEDGMENT

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

## REFERENCES

1. Alan Jay Smith, "Cache Memories", Computing Surveys, Vol.14, No.3, pp. 473-530, September 1982
2. David A. Patterson, John L. Hennessy: "Computer System Architecture : A Quantitative Approach", 3rd edition, Morgan Kaufmann Publishers Inc., 2003
3. K.T.Sundararajan, Timothy M.Jones, Nigel Topham, Smart Cache: A Self Adaptive Cache Architecture for Energy Efficiency, Proceedings of International Conference on Embedded Computer Systems, pp. 41-50, July 2011
4. D.Rolan, B.B.Fraguela, R.Doallo, Adaptive Line Placement with Set Balancing Cache, Proceedings of MICRO, pp. 529-540, December 2009
5. Se-Hyun Yang, Michael D. Powell, Babak Falsafi, T. N. Vijaykumar., Exploiting Choice in Resizable Cache Design to Optimize Deep-Submicron Processor Energy-Delay. HPCA 2002: 151-161
6. S. Subha, An Energy Saving Cache Algorithm, 2014 International Conference on Computational Science and Technology (ICCST), 2014, ISBN: 978-1-4799-3241-2
7. S.Subha, A Variable Cache Set Architecture, IJAER, Vol. 10, No. 20, November 2015



10817- Prof. Subha S

## AUTHORS PROFILE

**S. Subha** has done her Ph.D in computer Engineering in processor caches from Santa Clara University, CA, USA. Her research interests are in processor cache memories, computer arithmetic. She has teaching experience of seventeen years, software industry experience of five and half years. She is presently working in Vellore Institute of Technology, Vellore, India. She has successfully guided/co-guided four research scholars in area of computer architecture, parallel processing, cloud computing at VIT, Vellore. She has authored/co-authored fifty journal papers in international journals, thirty nine international conference publications. She has worked as reviewer of international journals for past five years