Chapter 15

A Few Processor Cache Architectures

Srinivasan Subha

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.77233

Abstract

Data from main memory are processed in the CPU of computer in any application like database-management system (DBMS), numerical applications. The computation time can be improved by the addition of processor caches. The cache takes advantage of the locality of reference in any application/calculation. This chapter discusses the two prevalent cache architectures, namely inclusive and exclusive. A new cache architecture for data only called two-type data cache proposed in the literature is presented in the following section. The performance of two-type data cache model is compared with inclusive and exclusive architectures. The energy consumed by inclusive and exclusive caches is mentioned. Methods to reduce the energy consumption are proposed for inclusive and exclusive cache architectures. The hardware and software methods for energy saving are proposed. The proposed models are simulated using SPEC2000 benchmarks. The benchmarks are for compression, combinatorial optimization, word processing, place and route simulator, object oriented database, field-programmable gate array (FPGA) circuit placement and routing. The results are presented.

Keywords: average memory access time, exclusive cache, inclusive cache, two-type data cache

1. Introduction

The computer has three major parts, namely, processor, memory and input–output. The modern computer is based on Von Neumann architecture of stored program concept. Any application/program is fetched into main memory and executed. The program has instructions. Any instruction is fetched into the processor (CPU) and executed based on pipeline concepts. The instructions act on data. Any program has 80% of its instructions in loops. Repeated access of data gives the concept of locality of reference. If any data is accessed repeatedly it has temporal locality. Data that are in vicinity of access are said to have spatial locality. Taking advantage of

locality of reference, the concept of caches was introduced in computers. The data/instruction from main memory is fetched into cache before accessed by the CPU. This improves the performance. The memory hierarchy thus runs from registers, caches, main memory, secondary memory, tertiary memory, etc. The performance of memory system is measured as average memory access time (AMAT). Various cache levels are prevalent in modern processors.

A processor cache is denoted by the tuple (C, k, L) where C is the capacity, k the associativity and L the line size. Based on the various values of k, three types of caches are known. These are direct mapped cache with k = 1, set associative cache with k > 1, fully associative cache with one set and n blocks. An address a is mapped to set given by a mod S, with tag value a div S for S sets. If a line is present in cache, it is cache hit, else it is cache miss. Cache misses are of three kinds: cold, capacity and conflict. A cache miss for first occurrence is called cold miss. The difference between misses in cache and fully associative cache of same capacity is the capacity miss. If a line maps to occupied set or way, it is called conflict miss. A computer system usually has many cache levels. A line can reside in cache level and higher cache levels in inclusive caches. A line resides in only one cache level in exclusive caches. Usually processors have caches dedicated to instructions and data separately. These are called instruction cache and data cache, respectively. Certain systems have same cache for both instruction and data. These are called unified caches. A system with caches of two or more kinds (direct mapped, set associative, fully associative) is called multilateral or hybrid cache. As the address mapping may vary with each cache level, the average memory access time (AMAT) is used to measure the cache performance.

As the number of computer components active increases, the energy consumed also increases. The power consumed by cache depends on number of active components. The energy is given as E = power $\times$ time. The energy is given by the formula $E = \frac{1}{2}cv^2f$ for electronic component where c is the capacity, v the voltage and f the frequency.

The performance of CPU caches is measured by execution time for various applications. Benchmarks are used to measure the CPU performance. The SPEC2000 benchmarks are one of standard benchmarks. The integer benchmark suites of SPEC2K are given as follows:

Name	**Description**
256.bzip2	Compression
181.mcf	Combinatorial optimization
197.parser	Word processing
300.twolf	Place and route simulator
255.vortex	Object-oriented database
175.vpr	FPGA circuit placement and routing

The memory performance is improved by adding caches. The inclusive, exclusive and two-type data cache models are presented in this chapter. The proposed models are simulated using SPEC2000 benchmarks. The benchmarks are run using Simplescalar Toolkit for simulations.

http://dx.doi.org/10.5772/intechopen.77233

2. Inclusive caches

Consider cache system of n cache levels, main memory. Let the cache levels are $L_1, L_2, .., L_n$. Let the cache be inclusive. Then, $L_1 \subseteq L_2 \subseteq L_3 \subseteq \ldots \subseteq L_n$. Denote this system as C_{incl}. This is shown in **Figure 1**. The cache sizes grow with the level number [1, 2]. Consider three-level cache system.

Let the levels be L_1, L_2, L_3. Let an address trace have R references. Let h_1, h_2, h_3 be number of hits in level one, level two, level three, respectively. Let $t_1, t_2, t_3, t_{21}, t_{32}$ be the access time to level one, level two, level three, transfer time between level two and level one and transfer time between level three and level two cache levels, respectively. Let M be the miss penalty. The average memory access time is given by

$$AMAT(C_{incl}) = \frac{1}{R}(h_1 t_1 + h_2(t_1 + t_2 + t_{21}) + h_3(t_1 + t_2 + t_3 + t_{32} + t_{21}) + (R - h_1 - h_2 - h_3)M) \tag{1}$$

The first three terms in Eq. (1) are the access time of level one, level two and level three cache hits. The last term is the miss penalty. This expression can be extended to any number of cache levels.

Energy consumed in cache depends on number of active components. The individual lines can be selectively switched on in caches using certain software techniques or hardware circuits. The total power consumed can be reduced through this technique. Consider w-way set associative cache of S sets. Let the power consumed per line be p watts. The total power consumed is wpS. Consider a circuit which enables lines if occupied. This is shown in **Figure 2**. If the power consumed by the circuit is q watts, the number of occupied lines is y, the total power consumed is q + yp. An improvement in power consumption is observed if q + yp < wpS.

Power saving using software techniques involves mapping lines to fixed ways by address mapping techniques [4].

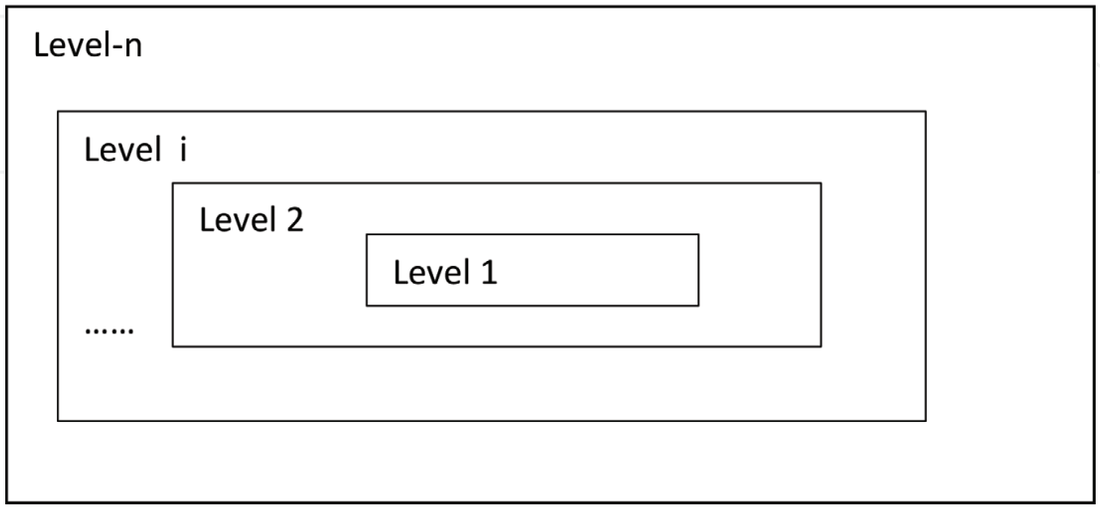

Figure 1. Inclusive cache of n levels.

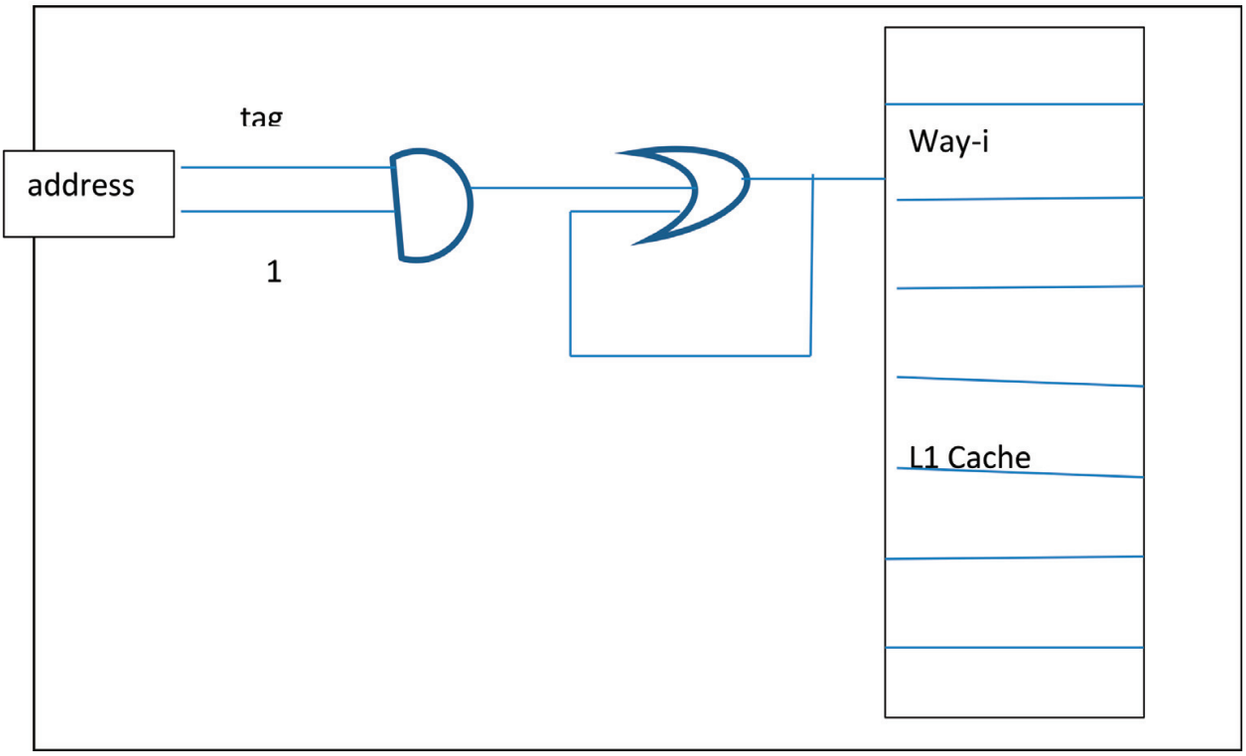

Figure 2. Sequential circuit in cache way to save power consumption. For details, refer [3].

3. Exclusive caches

Exclusive caches have a line in one cache level only. There is no containment property of the inclusive caches. Let the cache levels be $L_1, L_2, .., L_n$.The exclusive cache has the property that $L_i \cap L_j = \phi$, for i not equal to j. The exclusive cache is depicted in **Figure 3**.

Consider cache with two cache levels. The placement, replacement algorithm as proposed by Jouppi and Wilton [5] is given as follows:

1. Check if line is present in level one. If present, access the line and stop.
2. Check if line is present in level two. If present, swap with level one cache line, access and stop.
3. If line is not present in cache, put the line in level one cache evicting the victim in first level cache to second level cache.

The number of sets in both levels has to be equal in the abovementioned design. Let h_1, h_2 be level one hits and level two hits in trace of R references. Let t_1, t_2, t_{12}, t_{1m} be access time to level one, level two, transfer time between level one and level two caches, transfer time between level one and main memory. Let M be miss penalty. Denote this exclusive cache system as C_{excl}. The average memory access time is given by Eq. (2).

Figure 3. Exclusive cache of n levels.

$$AMAT(C_{excl}) = \frac{1}{R}(h_1 t_1 + h_2(2t_1 + t_2 + 2t_{12}) + (R - h_1 - h_2)(2t_1 + t_2 + t_{1m} + t_{12})M) \quad (2)$$

The first term in Eq. (2) is level one hit time. The second term is level two hit time. The factor of 2 in this expression is because of swapping of the lines. The third term is the miss penalty.

Another logic to realize exclusive caches is proposed by Subha [6]. Consider two-level cache system. The placement, replacement logic is given as follows:

1. Initialize all lines to be in level zero.
2. If the line is present in level one or level two cache (cache hit), let the line be in logical level one and stop.
3. Check if level one cache line is free. If so, place the line in level one cache, consider it as logical level one cache and stop.
4. Check if level two cache line is free. If so, place the line in level two cache; consider it as logical level one cache and stop.
5. Check the status of physical level one and physical level two caches. If physical level one cache has status of logical level two and vice-versa, place the block in level one cache and change its status to logical level one, change the status of physical level two cache to logical level two cache and stop. Else, place the block in physical level two, treat physical level one as logical level two, physical level two as logical level one and stop. [Put in logical level two, flip the level indices].

The abovementioned model does not require the two cache levels to have equal sets. There is a path between main memory and level two. Let β, γ, t_{2m} be the decrease in level one hits from model proposed in [5], increase in level two hits from model proposed in [5], transfer time between level two and main memory. Let this exclusive cache system be denoted as C_{excl2}. The average memory access time of this system is given by Eq. (3).

$$AMAT(C_{excl2}) = \frac{1}{R}(H_1 - \beta) + (H_2 + \gamma)(t_1 + t_2) + 2xt_{1m} + 2yt_{2_m} \quad (3)$$

where H_1, H_2 are level one and level two hits and $x + y = R - H_1 - H_2 - \beta - \gamma$. An improvement in AMAT is observed given by Eq. (4).

$$\frac{1}{R}(h_1 t_1 + h_2(2t_1 + t_2 + 2t_{12}) + (R - h_1 - h_2)(2t_1 + t_2 + t_{1m} + t_{12})M) > \frac{1}{R}(H_1 - \beta) + (H_2 + \gamma)(t_1 + t_2) + 2xt_{1m} + 2yt_{2m} \tag{4}$$

The simulations of exclusive cache proposed in [6] using SPEC2K benchmarks are presented in the following section. **Table 1** gives the configurations and **Table 2** gives the AMAT.

The AMAT is depicted in **Figure 4**.

The power consumed by exclusive depends on the number of active cache lines. One method to reduce the number of active lines is to have separate cache called tag cache [7]. The tag cache contains the tag values of all cache levels. The address mapping proceeds as follows in the tag cache.

1. Compute the following

   Set1 = a mod S_1.

   Tag1 = a div S_1.

   Set2 = a mod S_2.

   Tag2 = a div S_2.

2. Check in tag cache for the matching of Tag1. If match is found, the line is present in level one cache as it is level one hit condition. Access the line in level one cache and stop. If there is level one cache miss, check for level two cache hit by inspecting match of Tag2. If there is level two cache hit, access the line in level two cache and stop.

3. This step is for cache miss condition. Place the least recently used line in Set2 of level two in main memory. Transfer the least recently used line of Set1 in level one cache in the evicted level two line. Place the line with address a in level one cache. Update the entries in the tag cache.

S.No	Configuration	L1 Size(sets)	L2 size (sets)	L1 access time(cycles)	L2 access time(cycles)	L1-memory access time(cycles)	L2-memory access time(cycles)	L1-L2 access time(cycles)
1	SIM1(direct-direct)	1024	4096	3	12	50	65	NA
2	SIM2(direct-set assoc)	1024	1024x4	3	12	50	65	NA
3	SIM3(set assoc-set assoc)	256x4	1024x4	3	12	50	65	NA
4	SIM4(direct-set assoc)	1024	1024x4	3	12	50	65	20

Table 1. Simulation configurations for two-level exclusive cache.

Name	SIM1	SIM2	SIM3	SIM4
256.bzip2	12.30354166	89.87282913	11.8696322	86.28267447
181.mcf	35.56382442	77.60141271	35.83299697	86.49583649
197.parser	15.17529205	88.39877944	16.93835532	96.49734912
300.twolf	33.12386439	74.00487373	35.36552946	80.44703014
255.vortex	24.04549228	63.56599701	30.74182301	50.67853881
175.vpr	26.9259936	71.23640932	33.29396985	59.29468875
Average	**24.5230014**	**77.44671689**	**27.34038447**	**76.61601963**

Table 2. AMAT values of two-level exclusive caches.

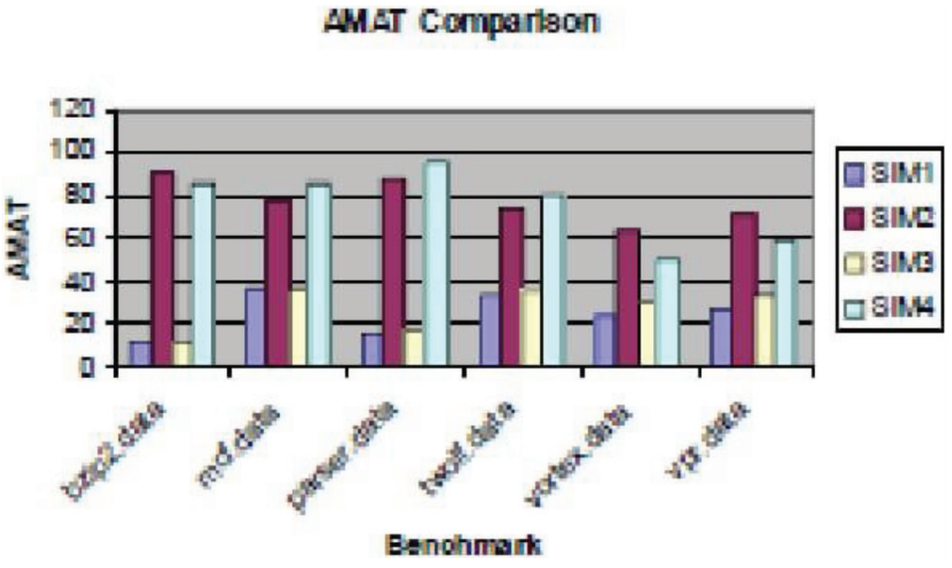

Figure 4. AMAT comparisons of two-level exclusive caches.

4. Stop

In the abovementioned algorithm, a level one cache line or higher level cache line is enabled only on cache hit or cache miss in all levels. This saves the energy consumed by the cache system. The cache is exclusive in nature and the exclusive algorithm proposed by Jouppi and Wilton [5] is used. The proposed model has scalability. The architecture is shown in **Figure 5**.

Let trace be R references. Let h_1, h_2, *cmiss*1, *cmiss*2, *miss* be he hits in level one cache, hits in level two cache, misses filled in vacant level one cache, misses filled in vacant level two cache, conflict misses in level one and level two caches, respectively. Let t_0, t_1, t_2, t_{1m}, t_{2m}, t_{12} be tag cache access time, level one cache access time, level two cache access time, transfer time between level one and main memory, transfer time between level two and main memory,

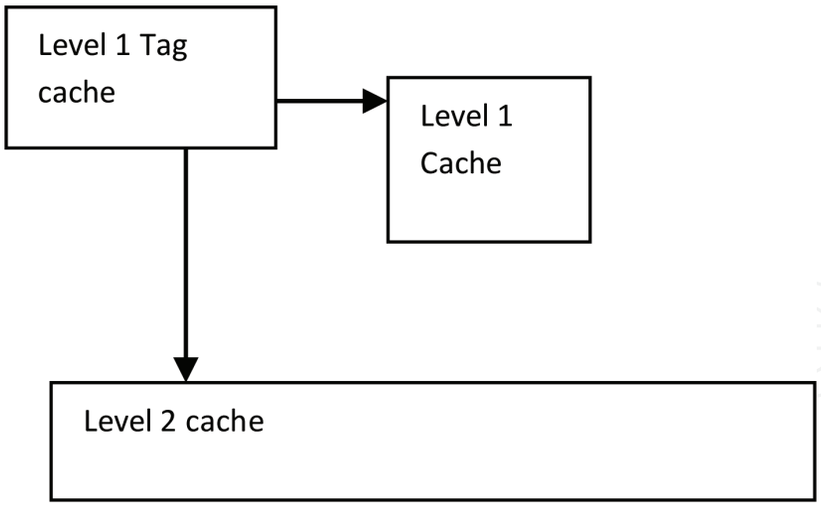

Figure 5. Exclusive tag cache architecture [6].

transfer time between level one and level caches, respectively. Let the tag cache system be denoted as $C_{excltag}$. The average memory access time is given by Eq. (5).

$$AMAT\left(C_{excltag}\right) = \frac{1}{R}\begin{pmatrix} Rt_0 + h_1t_1 + h_2t_2 + \\ cmiss1(t_1 + 2t_0) + \\ cmiss2(t_2 + 3t_0) + \\ miss\begin{pmatrix} 4t_0 + t_1 + \\ t_2 + t_{12} + t_{2m} + t_{1m} \end{pmatrix} \end{pmatrix} \tag{5}$$

The tag cache access time is the first term in Eq. (5). The hit time accesses to level one and level two caches are given by second and third terms, respectively. The time taken to fill vacant way in level one cache is given by the fourth term. This involves accessing tag cache in level one, fetching the line to level one cache and updating the tag cache entry. The time taken to fill vacant second level cache line is given by the fifth term. This includes accessing the tag cache to check for match in level one cache and level two cache and placing the line in level two cache, updating the tag cache entry. The time taken to replace existing line is given by the sixth term. This involves checking for tag match in level one cache, level two cache, replacing the level one cache line and updating the tag cache entries. As the tag cache contains the tags in consecutive locations, it may be the case that the tag entries in level one and level two are in two different cache blocks.

The energy consumed in exclusive tag cache is calculated in the following section. Let us assume that the cache operates in two modes: high-power mode and low-power mode. On accessing cache way, its corresponding set is placed in high-power mode from low-power mode. Let E_{high}, E_{low} be the energy consumed by the cache way in the proposed tag cache model in high-power mode and low-power mode, respectively. Let W_{high}, W_{low} be the energy consumed by one cache line in tag cache in high-power mode and low-power mode, respectively. Let $E_{delta,}$ W_{delta} be the difference in energy level for cache way and tag cache way between the two modes of operation, respectively. When no cache operation is performed, the

energy consumed in the cache system is $(w_1S_1 + w_2S_2 + T)E_{low}$ for w_1-way set associative cache at level one and w_2-set associative cache in level two. For level one cache hit, the energy consumed is $W_{delta} + w_1E_{delts}$. This is because the tag cache entry and the set in level one cache are enabled in high energy mode. For level two cache hit, the energy consumed is $2W_{delta} + w_2E_{delta}$. This is because the tag cache is searched for match in level one cache and level two cache and the level two cache set containing the line is enabled. For the unfilled level one cache situation, the energy consumed is $W_{delta} + w_1E_{delta}$ as the tag cache is searched to confirm that it is a free level one cache and the way in the level one cache is filled with the block. For the free level two cache way, the energy consumed is $2W_{delta} + w_2E_{delta}$. This is because the tag cache entry in level one and level two caches are searched to confirm that there is free level two cache way, and the corresponding level two cache set is enabled to fill the way. For a miss, the total energy consumed is given by $2W_{delta} + w_1E_{delta} + w_2E_{delta}$. This is because the tag entries for level one and level two caches are enabled, and the line replaces a filled level one cache set and level two cache set. Consider a program with R references. The total energy consumed for this address trace is given by Eq. (6).

$$\begin{aligned} &R(S_1 + S_2 + T)E_{low} + h_1(W_{delta} + w_1E_{delta})+ \\ &h_2(2W_{delta} + w_2E_{delta})+ \\ &cmiss1(W_{delta} + w_1E_{delta})+ \\ &cmiss2(2W_{delta} + w_2E_{delta})+ \\ &miss(2W_{delta} + w_1E_{delta} + w_2E_{delta}) \end{aligned} \tag{6}$$

The first term in Eq. (6) is the energy consumed when the cache is not accessed. The second term is the energy consumed in level one hits. This is equal to enabling the line in tag cache and accessing the ways in the level one cache for the set. The third term gives the energy consumed for level two cache hits. The fourth term is the energy consumed for filling a vacant level one cache line. The fifth term is the energy consumed to fill a vacant level two cache line. The sixth term is the energy consumed to replace an existing line in level one cache on a conflict miss. Consider a traditional exclusive cache with the same algorithm given in Section 3. The energy consumed for R references is calculated as follows. Let E_{high} be the energy consumed per way in the cache system. All the sets in both the cache levels are enabled in high energy power mode for all references. The energy consumed is given by Eq. (7).

$$(S_1 + S_2)E_{high} \tag{7}$$

A saving in energy consumption is observed as given by Eq. (8).

$$\begin{aligned} &R(S_1 + S_2 + T)E_{low} + h_1(W_{delta} + w_1E_{delta})+ \\ &h_2(2W_{delta} + w_2E_{delta})+ \\ &cmiss1(W_{delta} + w_1E_{delta})+ \\ &cmiss2(2W_{delta} + w_2E_{delta})+ \\ &miss(2W_{delta} + w_1E_{delta} + w_2E_{delta}) \\ &\Leftarrow \\ &(S_1 + S_2)E_{high} \end{aligned} \tag{8}$$

S.No	Parameter	Value
1	Cache Level one size	128KB
2	Associativity of Level one cache	4
3	Cache Level two size	2048KB
4	Associativity of Level two cache	64
5	Access time of Level one cache	3 cycles
6	Access time of Level two cache	18 cycles
7	Transfer time between Level one and level two caches	18 cycles
8	Transfer time between Level one and main memory	60 cycles
9	Transfer time between Level two and memory access time	90 cycles
10	Cache block size	32 bytes

Table 3. Simulation parameters for Exclusive Tag cache model.

The simulations for the above proposed tag cache model of exclusive cache is presented in the following section [7]. The simulation parameters are given in **Table 3**. The energy consumed in low-power mode is 5 J and high-power mode is 15 J. The AMAT was calculated using C routines. **Figure 6** gives the AMAT and **Figure 7** gives the energy consumed. As seen from **Figure 6**, the AMAT performance is comparable with traditional exclusive cache. The energy is saved by 23% as seen from **Figure 7** when compared with traditional exclusive cache.

A hardware method [8] to improve the power consumption is to enable level one and level two cache lines based on the contents of the tag cache. This is depicted in **Figure 8**.

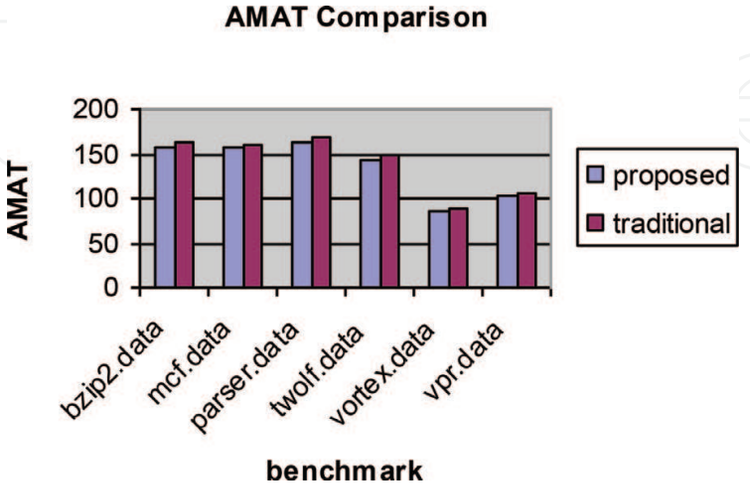

Figure 6. AMAT comparison for tag cache model of exclusive caches.

http://dx.doi.org/10.5772/intechopen.77233

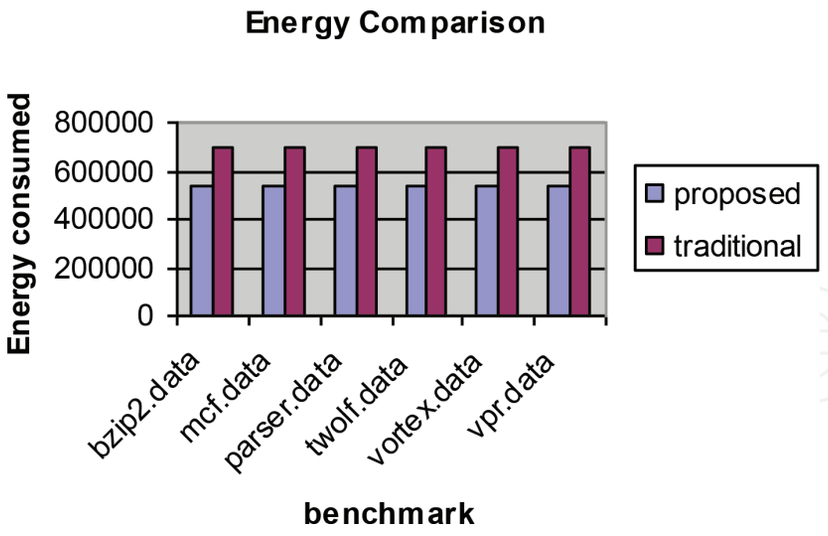

Figure 7. Energy comparison of tag cache model of exclusive caches.

The simulations for the model presented in [8] are presented in the following section. The simulation parameters are shown in **Table 4**. The power consumed is shown in **Table 5**. The AMAT is shown in **Figure 9**. The t is traditional model and p is the proposed model. There is 49% improvement in power consumption with no change in AMAT for this model compared

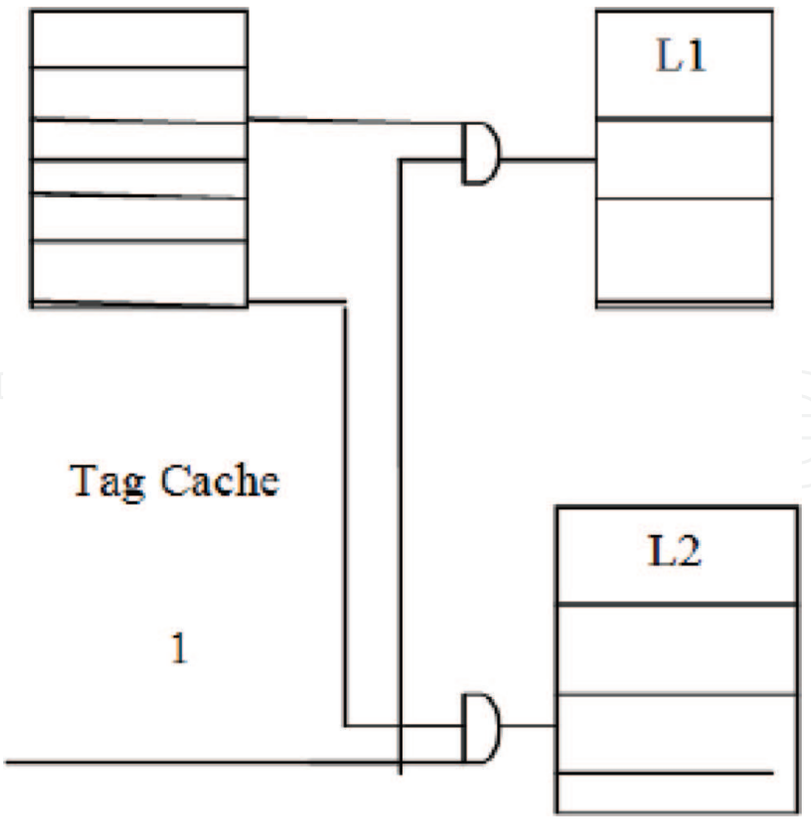

Figure 8. Proposed hardware for power saving in exclusive tag cache architecture of two levels [9].

S.No	Parameter	Value
1	Level one cache size	128KB
2	Level one associativity	4
3	Level two cache size	256KB
4	Level two associativity	8
5	Level one access time	3 cycles
6	Level two access time	18 cycles
7	Level one to level two transfer time	18cycles
8	Level one to memory access time	60 cycles
9	Level two to memory access time	90 cycles
10	Line size	32 bytes
11	Power per cache way	$8332.8\mu W$ 8334.8

Table 4. Simulation parameters of sequential circuit for exclusive caches.

Name	Power(t)KW	Power(p)KW	%improve
256.bzip2	73271.56	73271.56	0
181.mcf	621.5154	229.5009	63.07398015
197.parser	5086.737	2716.116	46.60396242
300.twolf	707.5931	262.1133	62.95705823
255.vortex	1888.692	710.3674	62.38839366
175.vpr	1372.853	516.3521	62.38839118
Average			**49.56863094**

Table 5. Power consumed in sequential circuit of exclusive cache.

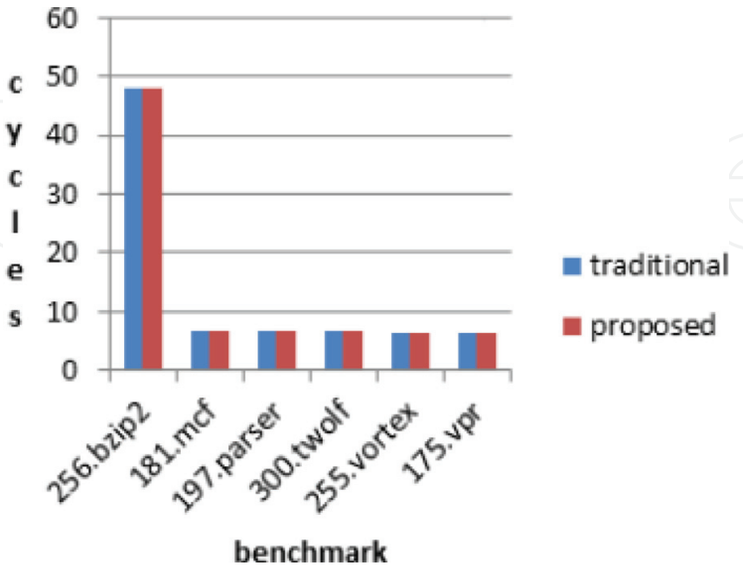

Figure 9. AMAT comparison of sequential circuit model for exclusive caches.

with the exclusive model proposed in tag cache model of exclusive cache (t in this discussion) as proposed in [7].

4. Two-type data cache model

The caches discussed so far in this chapter are inclusive and exclusive. A two-level data cache model [9] which selectively makes the cache ways in various levels as inclusive or exclusive is presented in the following section. This model makes the cache ways inclusive based on access. A two-level data cache is chosen for discussion. Initially, both cache levels are exclusive. The first occurrence of address places the data in one of cache levels making the occupied way exclusive. Preference is given to place the data in level one in this case. On consecutive access to data in level one cache, cache way is made inclusive with level two cache. During this process, data in level two cache may be replaced. The least recently used algorithm (LRU) is used for data replacement in level two cache. If the data have temporal access in level two cache, the way is made exclusive. On a cold miss with both the levels occupied, the block is placed in level one cache making it exclusive. As the number of ways to place the data increases, the performance increases in terms of the average memory access time (AMAT). **Figure 10** shows this architecture.

The algorithm for two-type data cache model is given in the following section.

Algorithm for two-type data cache: Given two-level data cache, this algorithm places the line in the cache system. Input is the address. An index array is maintained per cache for each way. It is zero to indicate inclusive and one to indicate exclusive.

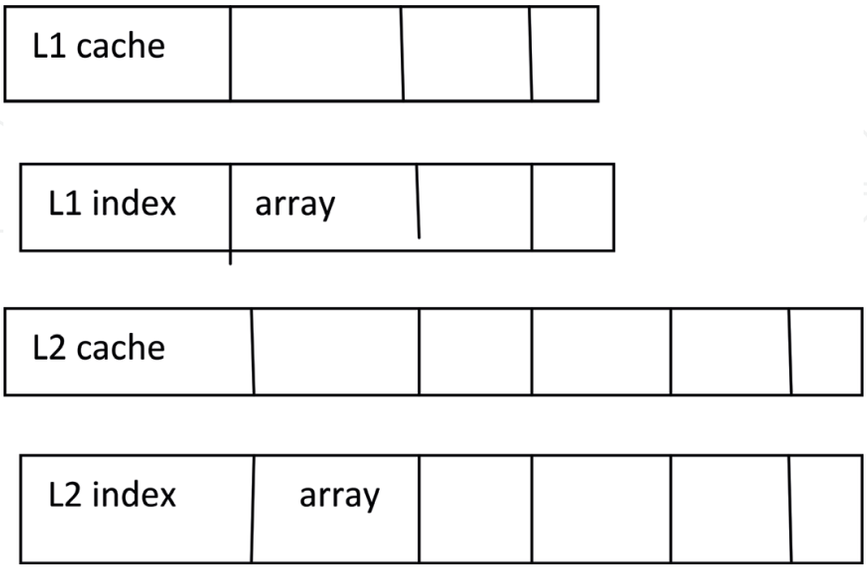

Figure 10. Two-type data cache model [5].

1. A level one hit occurs if the address is found in level one. The block is made inclusive by placing a copy of it in level two cache. The index array entry is set to 00 to indicate this in both the levels. The block is accessed and the process stops.

2. A level one miss and level two hit occurs if the address is not found in level one but is found in level two cache. The block is made exclusive in this case. This is indicated by setting index array entry of block to 11. The block is accessed and the process stops.

3. A cache miss occurs if the block is not present in level one and level two cache. If level one cache is vacant, it is placed in level one cache. Else, if level two cache way is vacant, it is placed in level two cache. The block is made exclusive. The block is accessed and process stops. If the level one set is full, the block replaces an existing block based on least recently used algorithm. The corresponding index array entry is made exclusive by setting it to 11. The mapping process stops. **Table 6** gives the algorithm.

Let R, H_1, H_2, x_1, x_2, y be the number of references, number of hits in level one, number of hits in level two, number of hits in level one after first hit, number of new level one hits, misses that are filled in vacant level two ways, respectively. Let t_1, t_2, t_{12}, t_{2m}, t_{1m} be level one access time, level two access time, transfer time between level one and level two, transfer time between level two and main memory, transfer time between level one and main memory, respectively. Let the proposed system be denoted as $C_{twotype}$. Let k be the time taken to update index array. The AMAT is given by Eq. (9).

$$AMAT(C_{twotype}) = \frac{1}{R}\begin{pmatrix} x_1(t_1+2k)+x_2(t_1+t_{12}+2k)+ \\ H_2(t_1+t_2+2k)+ \\ y(t_1+t_2+t_{2m}+2k)+ \\ (R-x_1-x_2-H_2-y) \\ (t_1+t_2+t_{1m}+2k) \end{pmatrix} \tag{9}$$

where $H_1 = x_1 + x_2$.

Level one	Level two	Similarity with	Action
Miss	Miss	Exclusive	Put in level one, make level one exclusive
Hit	Miss	Inclusive	Put in level two, make the way inclusive
Hit	Hit	Inclusive	Do nothing as it is inclusive already
Miss	Hit	None	Make level two exclusive

Table 6. Two-type data cache algorithm.

In Eq. (9) the first term gives inclusive cache type hits, the second term indicates first time level one hits. The third term indicates the level two hits. The fourth term indicates placing a block in vacant level two way/set. The fifth term indicates either placing a block in free level one cache set/way or replacing a level one cache set/way on cache full scenario. It is assumed that exchange of a block from level one to memory can be done in parallel. The term 2 k is to update the index arrays of both the cache levels. The first term gives the hit time in level one cache. The second term gives the hit time in level two cache. This involves accessing level one and level two caches. The third term gives the time to service the misses. This involves accessing level one and level two caches, determining if it is missed in both levels, accessing main memory and fetching the block into level one. The level one block is written to the memory before the new block is fetched. Simultaneously, the requested block is sent to the processor.

Denote the inclusive cache as C_i and exclusive cache as C_e. Let $H_{i1}, H_{i2}, H_{e1}, H_{e2}$ be the number of level one hits and level two hits in inclusive cache, number of level one hits and number of level two hits in exclusive cache, respectively.

The AMAT for inclusive cache is given by Eq. (10).

$$AMAT(C_i) = \frac{1}{R}\begin{pmatrix} H_{i1}t_1 + H_{i2}(2t_1 + t_2 + t_{12}) + \\ (R - H_{i1} - H_{i2}) \\ (2t_1 + 2t_2 + t_{1m} + t_{12}) \end{pmatrix} \tag{10}$$

The first term is the level one hit time. The level two hit time is given by second term. This involves accessing level one, level two and transferring data from level one to memory and data from level two to level one cache. The miss time is given by third term. This involves accessing level one, level two to determine it is a miss in both levels, write from level one to memory and level two to memory the existing blocks. The new block from main memory is fetched into level two cache and from there to level one cache. Simultaneously, it is given for processing. An improvement in performance is observed as given by Eq. (11).

$$\frac{1}{R}\begin{pmatrix} x_1(t_1 + 2k) + x_2(t_1 + t_{12} + 2k) + H_2(t_1 + t_2 + 2k) \\ +y(t_1 + t_2 + t_{2m} + 2k) + \\ (R - x_1 - x_2 - H_2 - y) \\ (t_1 + t_2 + t_{1m} + 2k) \end{pmatrix} \Leftarrow \frac{1}{R}\begin{pmatrix} H_{i1}t_1 + H_{i2}(2t_1 + t_2 + t_{12}) + \\ (R - H_{i1} - H_{i2}) \\ (2t_1 + 2t_2 + t_{1m} + t_{12}) \end{pmatrix} \tag{11}$$

The AMAT for the exclusive system is given by Eq. (12).

$$AMAT(C_e) = \frac{1}{R}\begin{pmatrix} H_{e1}t_1 + H_{e2}(t_1 + t_2 + t_{12}) + \\ (R - H_{e1} - H_{e2}) \\ (t_1 + t_2 + t_{1m} + t_{12} + t_{2m}) \end{pmatrix} \tag{12}$$

The first term in Eq. (12) is the level one hit time. The second term is for level two cache hits. This needs access of level one cache, level two cache and exchange the contents. On a miss in

both the levels, the contents of level two are updated in the main memory, level one cache block is sent to level two and new block is fetched from main memory to level one cache. Simultaneously, the requested block is sent to the processor. The terms in Eq. (9) and Eq. (12) differ due to the architectural differences. A performance improvement in the proposed model over exclusive cache is seen as given by Eq. (13).

$$\frac{1}{R}\begin{pmatrix} x_1(t_1+2k)+x_2(t_1+t_{12}+2k)+ \\ H_2(t_1+t_2+2k)+ \\ y(t_1+t_2+t_{2m}+2k)+ \\ (R-x_1-x_2-H_2-y) \\ (t_1+t_2+t_{1m}+2k) \end{pmatrix} \Leftarrow \frac{1}{R}\begin{pmatrix} H_{e1}t_1+H_{e2}(t_1+t_2+t_{12})+ \\ (R-H_{e1}-H_{e2}) \\ (t_1+t_2+t_{1m}+t_{12}+t_{2m}) \end{pmatrix} \tag{13}$$

In all the models, it is assumed that the cache block when fetched into the cache is simultaneously sent to the processor.

The proposed two-type data cache model is simulated using SPEC2000 benchmarks. The proposed model is compared with inclusive, exclusive caches described in this chapter. The simulation parameters are given as follows (**Tables 7** and **8**):

The AMAT values are given in the graph in **Figure 11**. It is compared with inclusive and exclusive caches.

S.No	Parameter	Value
1	Cache level one size	32KB with line size 32B
2	Associativity level one	4
3	Cache level two size	128KB with line size 32B
4	Associativity level two	8
5	Cache level one access time	3 cycles
6	Cache level two access time	12 cycles
7	Transfer time between level one and memory	50 cycles

Table 7. Simulation parameters of proposed two-type data cache and inclusive cache.

S.No	Parameter	Value
1	Cache level one size	32KB with block size of 32B
2	Cache level two size	32KB with block size 32B
3	Access time of level one cache	3 cycles
4	Cache level one to memory access time	50 cycles
5	Transfer time between level one and level two	20 cycles
6	Access time of level two cache	65 cycles

Table 8. Simulation parameters for exclusive cache in two-type data cache.

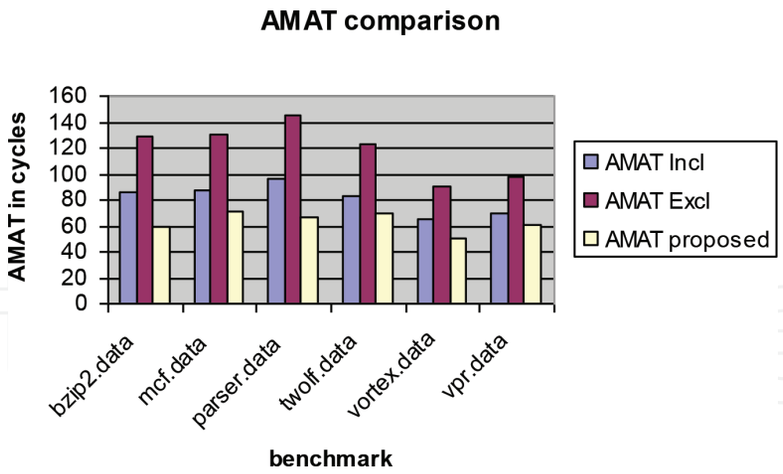

Figure 11. AMAT comparison with inclusive cache.

Name	AMAT(incl)	AMAT(excl)	AMAT(prop)	%Improvement (Incl)	%Improvement (Excl)
bzip2.data	86.285727	129.212955	59.034869	31.58211555	54.31195812
mcf.data	86.973512	129.696266	70.372351	19.08760566	45.74064993
parser.data	96.926332	145.059242	67.089295	30.7832107	53.75041667
twolf.data	83.424551	122.8655	69.034788	17.24883482	43.81271553
vortex.data	64.570065	90.905944	49.918977	22.69021721	45.08722444
vpr.data	69.496117	97.652581	60.053221	13.58765987	38.50319123
Average				22.4966073	46.86769265

Table 9. AMAT values for two-type data cache simulations.

As observed from **Figure 11**, improvement in AMAT in proposed system compared with inclusive cache is seen. There is a decrease in AMAT by 3% compared with exclusive caches. The proposed model has better performance for systems where elements of set are accessed more than two times after a sequence of other cold misses mapped to the same set such that the number of total misses is a multiple of number of elements of level one cache.

The AMAT values are shown in **Table 9**.

Acknowledgements

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

Author details

Srinivasan Subha

Address all correspondence to: ssubha@rocketmail.com

School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

References

[1] Smith AJ. Cache memories. ACM Computing Surveys. 1982:473-530

[2] Patterson DA, Hennessey JL. Computer System Architecture: A Quantitative Approach. Morgan Kaufmann Publishers Inc.; 2003

[3] Subha S. A reconfigurable cache architecture. In: Proceedings of ICHPCA. 2014

[4] Subha S. An energy saving cache algorithm. In: Proceedings of ICGCE. 2013. pp. 757-760

[5] Jouppi NP, Wilton SJE. Tradeoffs in two-level onchip caching. In: Proceedings of ISCA. 1994. pp. 34-45

[6] Subha S. An exclusive cache model. In: Proceedings of ITNG. 2009. pp. 1715-1716

[7] Subha S. An energy saving model of exclusive cache. In: Proceedings of HPCS. 2011. pp. 233-238

[8] Subha S. An exclusive cache architecture with power saving. In: IJST. 2015. pp. 1-5

[9] Subha S. A two-type data cache model. In: Proceedings of 2009 IEEE International Conference on Electro/Information Technology. 2009. pp. 476-481