Batch: D2 **Roll No.:16010122323**

Experiment / assignment / tutorial No. 8

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE : Implementation of Cache Mapping Techniques.

AIM: To study and implement concept of various mapping techniques designed for cache memory.

Expected OUTCOME of Experiment: (Mention CO/CO's attained here)

Books/ Journals/ Websites referred:

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", Fifth Edition, TataMcGraw-Hill.
2. Dr. M. Usha, T. S. Srikanth, "Computer System Architecture and Organization", First Edition, Wiley-India.

Pre Lab/ Prior Concepts:

Cache memory: The cache is a smaller, faster memory which stores copies of the data from the most frequently used main memory locations. As long as most memory accesses are cached memory locations, the average latency of memory accesses will be closer to the cache latency than to the latency of main memory.

2. Hit Ratio: You want to increase as much as possible the likelihood of the cache containing the memory addresses that the processor wants.

Hit Ratio= No. of hits/ (No. of hits + No. of misses)

There are only fewer cache lines than the main memory blocks, an algorithm is needed for mapping main memory blocks into cache lines. Further a means is needed for determining which main memory block currently occupies in a cache line. The choice of cache function dictates how the cache is organized. Three techniques can be used.

1. Direct mapping.

2. Associative mapping.
3. Set Associative mapping.

Direct Mapped Cache: The direct mapped cache is the simplest form of cache and the easiest to check for a hit. Since there is only one possible place that any memory location can be cached, there is nothing to search; the line either contains the memory information we are looking for, or it doesn't.
Unfortunately, the direct mapped cache also has the worst performance, because again there is only one place that any address can be stored. Let's look again at our 512 KB level 2 cache and 64 MB of system memory. As you recall this cache has 16,384 lines (assuming 32-byte cache lines) and so each one is shared by 4,096 memory addresses. In the absolute worst case, imagine that the processor needs 2 different addresses (call them X and Y) that both map to the same cache line, in alternating sequence (X, Y, X, Y). This could happen in a small loop if you were unlucky. The processor will load X from memory and store it in cache. Then it will look in the cache for Y, but Y uses the same cache line as X, so it won't be there. So Y is loaded from memory, and stored in the cache for future use. But then the processor requests X, and looks in the cache only to find Y. This conflict repeats over and over. The net result is that the hit ratio here is 0%. This is a worst case scenario, but in general the performance is worst for this type of mapping.

Fully Associative Cache: The fully associative cache has the best hit ratio because any line in the cache can hold any address that needs to be cached. This means the problem seen in the direct mapped cache disappears, because there is no dedicated single line that an address must use.However (you knew it was coming), this cache suffers from problems involving searching the cache. If a given address can be stored in any of 16,384 lines, how do you know where it is? Even with specialized hardware to do the searching, a performance penalty is incurred. And this penalty occurs for all accesses to memory, whether a cache hit occurs or not, because it is part of searching the cache to determine a hit. In addition, more logic must be added to determine which of the various lines to use when a new entry must be added (usually some form of a "least recently used" algorithm is employed to decide which cache line to use next). All this overhead adds cost, complexity and execution time.

Set Associative Cache (To be filled in by students)

Set-associative cache is a trade-off between direct-mapped cache and fully associative cache.

A set-associative cache can be imagined as a (n*m) matrix. The cache is divided into ‘n’ sets and each set contains ‘m’ cache lines. A memory block is first mapped onto a set and then placed into any cache line of the set.

The range of caches from direct-mapped to fully associative is a continuum of levels of set associativity. (A direct-mapped cache is one-way set-associative and a fully associative cache with *m* cache lines is *m*-way set-associative.)

Many processor caches in today's designs are either direct-mapped, two-way set-associative, or four-way set-associative.

Direct Mapping Implementation:

The mapping is expressed as

i=j modulo m

i=cache line number

j= main memory block number

m= number of lines in the cache

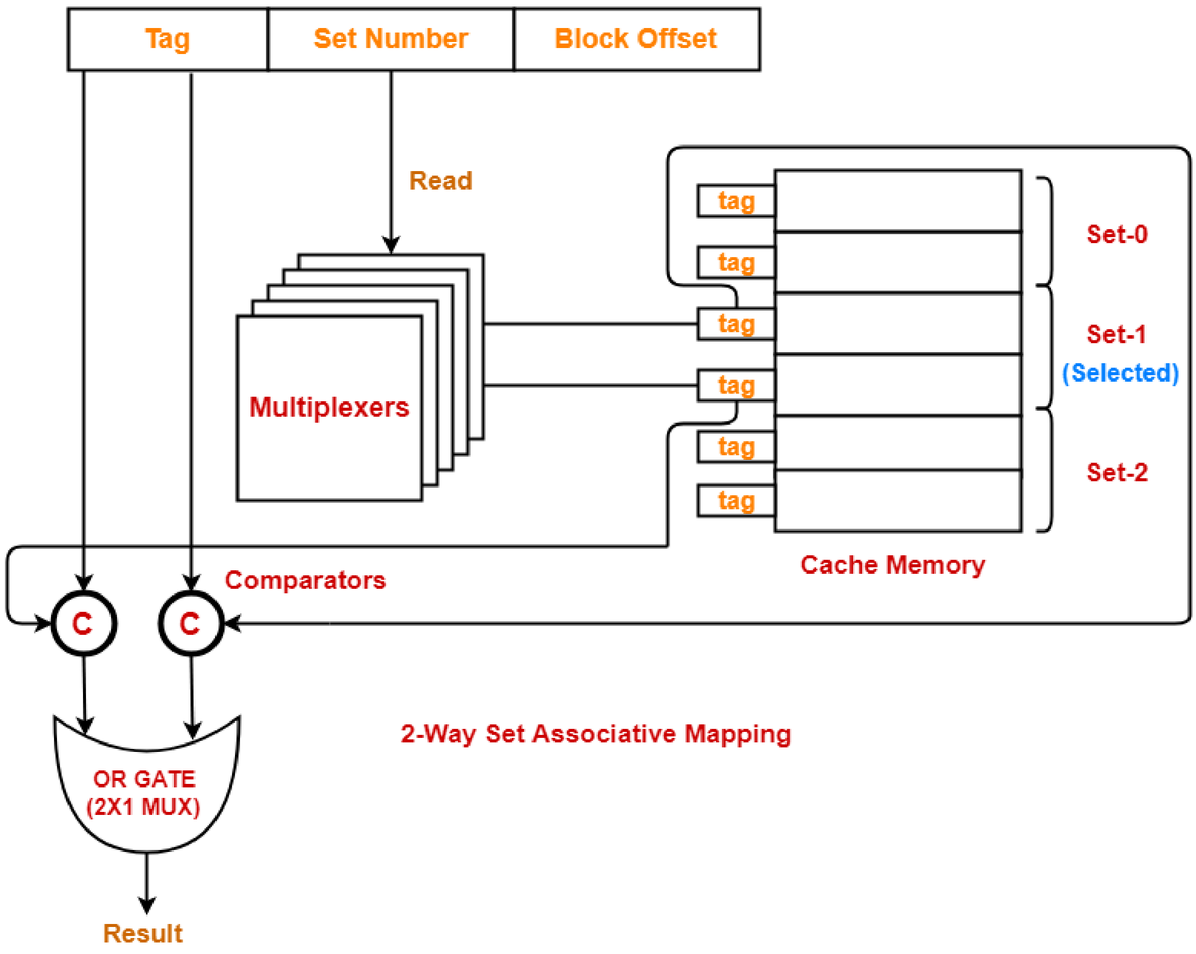
- Address length = (s+w) bits
- Number of addressable units = 2^{s+w} words or bytes
- Block size = line size = 2^{w} words or bytes
- Number of blocks in main memory = $2^{s+w} / 2^{w} = 2^{s}$
- Number of lines in cache = m = 2^{r}
- Size of tag = (s-r) tags

Associative Mapping Implementation: (To be filled in by students)

In fully associative mapping,

- A block of main memory can be mapped to any freely available cache line.
- This makes fully associative mapping more flexible than direct mapping.
- A replacement algorithm is needed to replace a block if the cache is full.

Set Associative Mapping Implementation :



Step-01:

- Each multiplexer reads the set number from the generated physical address using its select lines in parallel.
- To read the set number of S bits, number of select lines each multiplexer must have = S.

Step-02:

- After reading the set number, each multiplexer goes to the corresponding set in the cache memory.
- Then, each multiplexer goes to the lines of that set using its input lines in parallel.
- Number of input lines each multiplexer must have = Number of lines in one set

Step-03:

- Each multiplexer outputs the tag bit it has selected from the lines of selected set to the comparators using its output line.
- Number of output line in each multiplexer = 1.

<u>Post Lab Descriptive Questions</u>

**1. For a direct mapped cache, a m
ain memory is viewed as consisting of 3 fields. List and define 3 fields.**

I is the cache line number
j is the main memory block number
m is the number of lines in the cache

One field on the direct-mapped cache memory identifies a unique word or byte within a block of main memory. The remaining two fields specify one of the blocks of main memory. These two fields are a line field, which identifies one of the lines of the cache, and a tag field, which identifies one of the blocks that can fit into that line.

3. **What is the general relationship among access time, memory cost, and capacity?**

There is a relationship among Access time, Memory cost per bit and the storage capacity for that access times are given below-

Faster access time,

Greater cost per bit;

Greater capacity,

And in opposite sense,

Slower access time,

Smaller cost per bit,

Greater capacity.

Conclusion:

Implementation of Cache Mapping Techniques is understood.