Name Farkhanda Jabeen

Id 14725

Dept Bs(cs)

Semester 4th

Submitted to Sir Amin

Assigment No 4

Give answers to the following questions.

a) What is the general relationship among access time, memory cost and capacity?

Ans) As access time becomes faster the cost per bit increases. As memory size increase, the cost per bit is smaller. Also, with greater capacity, the access time becomes slower.

b) Discuss different memory access methods in detail.

Ans Different memory access methods are as follows.

- Sequential access:-

Memory is organized into units of data, called records. Access must be made in a specific linear sequence. stored addressing information is used to separate records and assist in the retrieval process. A shared read-write mechanism is used, and this must be moved from its current location to the desired location, passing and rejecting each intermediate record. Thus the time to access an arbitrary record is highly variable.

- Direct access:-

As with sequential access, direct access involves a shared read-write mechanism. However, individual blocks or records have a unique address based on physical location. Access is #

accomplished by direct access to reach a general vicinity plus sequential searching counting or waiting to reach the final location. Again, access time is variable.

- Random access:-

Each addressable location in memory has a unique, physically wired-in addressing mechanism. The time to access a given location is independent of the sequence of prior accesses and is constant. Thus, any location can be selected at random & directly addressed and accessed. Main memory and some cache systems are random access.

- Associative:-

This is a random access type of memory that enables one to make a comparison of desired bit locations within a word for a specified match, and to do this for all words simultaneously. Thus, a word is retrieved based on a portion of its contents rather than its address. As with ordinary random access memory, each location has its own adressing mechanism, and retreival time is constant independent of location or prior access patterns. Cache memories may employ associative access.

c) Discuss the importance of memory hierarchy.

Ans An efficiently used memory hierarchy is of primary importance in optimizing data transfer and storage. To exploit such a memory hierarchy, the code to be mapped should expose maximal data reuse possibilities. Code rewriting techniques, consisting of loop and data flow transformations, are essential to achieve this.

d) How does the principle of locality relate to the use of multiple memory levels.?

Ans Slower and less expensive memory is used in higher stages, with the most expensive being the registers in the processor as well as cache. Main memory is slower and less expensive, and is outside of the processor.

e) How main memory address is interpreted in direct, associative, and set-associative mapping?

Ans. Direct mapping:

* The simplest technique.
* Maps each block of main memory into only one possible cache line.

Associative mapping:-

* Permits each main memory block to be loaded into any line of the cache.

* The cache control logic interprets a memory address simply as a Tag and a word field.
* To determine whether a block is in the cache, the cache control logic must simultaneously examine every line's Tag for a match.

Set - associative mapping:

* A compromise that exhibits the strengths of both the direct & associative approaches while reducing their disadvantages.

Q) Write note on each of the following:

i) Memory unit of transfer.

Ans Unit of transfer :-

It is the maximum number of bits that can be read or written into the memory at a time. In case of main memory, it is mostly equal to word size: it is often larger and is refered to as blocks.

ii) Memory Performance parameters:

Ans The two most important characteristics of memory are capicity and performance. Three performance parameters are used.

- Access time (latency) :-

For random-access memory, this is the time it takes to perform a read or write operation, that is, the time from the instant that

an address is presented to the memory to the instant that data have been stored or made available for use. For non-random-access memory, access time is the time it takes to position the read-write mechanism at the desired location.

- Memory cycle time:-

This concept is primarily applied to random-access memory and consists of the access time plus any additional time required before a second access can commence. This additional time may be required for transients to die out on signal lines or to regenerate data if they are read destructively. Note that memory cycle time is concerned with the system bus, not the processor.

- Transfer rate:-

This is the rate at which data can be transferred into or out of a memory unit, for random-access memory, it is equal to 1/(cycle time)

(iii) Disc Cache

Ans A portion of main memory can be used as a buffer to hold data temporarily that is to be read out to disk.

* A few large transfers of data can be used instead of many small transfers of data.
* Data can be retrieved rapidly from the software cache rather than slowly from the disk.

iv) Principle of locality

Ans The principle of locality states that data in the vicinity of a referenced word are likely to be referenced in the near future.

OR

An implication of locality is that we can predict with reasonable accuracy what instructions and data a program will use in the near future based on its accesses in the recent past.

v) Logical cache and physical cache

A logical cache, also known as a virtual cache, stores data using virtual addresses. The processor accesses the cache directly, without going through the MMU. A physical cache stores data using main memory physical addresses. One obvious advantage of the logical cache is that cache access speed is faster than for a physical cache, because the cache can respond before the MMU performs any address translation. The disadvantage has to do with the fact that most virtual memory systems supply each application with the same virtual memory address space. That is, each application sees a virtual memory that starts at address 0. Thus, the same virtual address in two different applications refers to two different physical addresses.

vi) Replacement algorithms.

Ans Once the cache has been filled, when a new block is brought into the cache, one of the existing blocks must be replaced. For direct mapping, there is only one possible line for any particular block, and no choice is possible. For the associative and set-associative techniques, a replacement algorithm is needed. To achieve high speed, such an algorithm must be implemented in hardware.

vii) Possible approaches to cache Coherency

Ans Possible approaches to cache Coherency include the following.

* Bus watching with write through:-

Each cache controller monitors the address lines to detect write operations to memory by other bus masters. If another master writes to a location in shared memory that also resides in the cache memory, the cache controller invalidates that cache entry. This strategy depends on the use of a write-through police by all cache controllers.

* Hardware transparency:

Additional hardware is used to ensure that all updates to main memory via cache are reflected in all caches. Thus, if one processor modifies a word in its cache, this update is written to main memory. In addition, any matching words in other caches are similarly updated.

* Non cacheable memory:-
Only a portion of main memory is shared by more than one processor, and this is designated as non-cacheable. In such a system, all accesses to shared memory are cache misses, because the shared memory is never copied into the cache. The non-cacheable memory can be identified using chip-select logic or high-adress bits.

Q3 Differentiate each of the following.

i) Sequential, direct and random access.

Ans Sequential:-
Memory is ~~accessed~~ organized into units of data, called records. Access must be made in a specific linear sequence. Stored addressing information is used to separate records and assist in the retrieval process. A shared read-write mechanism is used, and this must be moved from its current location to the desired location, passing & rejecting each intermediate record. Thus, time to access an arbitrary record is highly variable.

Direct access:-
As with sequential access, direct access involves a shared read-write mechanism. However, individual blocks or records have a unique address based on physical location

Index Register

Access is accomplished by direct access to reach a general vicinity plus sequential searching, counting or waiting to reach the final location. Again, access time is variable.

Random access:-

Each addressable location in memory has a unique, physically wired-in addressing mechanism. The time to access a given location is independent of the sequence of prior accesses and is constant. Thus, any location can be selected at random and directly addressed and accessed. Main memory and some cache systems are random access.

7) Direct, associative, and set-associative mapping.

Ans Direct:-

The direct mapping technique is simple and inexpensive to implement. Its main disadvantages is that there is a fixed cache location for any given block. Thus, if a program happens to reference words repeatedly from two different blocks that map into the same line, then the blocks will be continually swapped in the cache, and the hit ratio will be low.

Associative:-

With associative mapping, there is flexibility as to which block to replace when a new block

is read into the cache. Replacement algorithms, discussed later in this section are designed to maximize the hit ratio. The principal disadvantage of associative mapping is the complex circuitry required to examine the tags of all cache lines in parallel.

Set-associative

Set-associative mapping is a compromise that exhibits the strengths of both the direct and associative approaches while reducing their disadvantages.

In this case the cache consists of a number sets, each of which consists of a number of lines. The relationships are

$m = v * k$

$i = j$ modulo v

where

i = cache set number

j = main memory block number

m = number of line in the cache

v = number of sets

k = number of lines in each set

This is referred to as k-way set-associative mapping. With set-associative mapping, ~~with~~ block B_j can be mapped into any of the lines of set j.

iii) Split cache and unified cache

Ans Has become common to split cach
* One dedicated to instructions.
* One dedicated to data
* Both exist at same level, typically as two L1 caches.

Advantages of unified cache:
* Higher hit rate
* Balances load of instruction and data fetches automatically.
* Only one cache needs to be designed and implemented.

Trend is toward split caches at the L1 and a unified caches for higher levels.

Advantages of split cache:
* Eliminates cache contention between instruction fetch/decode unit and execution unit
* Important in pipelining.

iv) Write through and write back

* Simplest technique.
* All write operations are made to main memory as well as to the cache.
* The main disadvantage of this technique is that it generates substantial memory traffic and may create a bottleneck.

* Minimizes memory writes.
* Updates are made only in the cache.

* Portions of main memory are invalid and hence accesses by I/O modules can be allowed only through the cache.

Q4 Solve each of the following:

i) Suppose that the processor has access to two levels of memory. Level-1 contains 1000 words and has an access time of 0.01 μs; L-2 contains 100000 words and has an access time of 0.1 μs. Assume that if a word to be accessed in level 1, then the processor accesses it directly. If it is in L-2 then the word is first transfered to level 1 and then accessed by the processor. Suppose 95% of the memory accesses are found in level 1, Then find the average time to access a word.

Ans) In our example, suppose 95% of the memory accesses are found in level 1. Then the average time to access a word can expressed as.

$(0.95)(0.01ms) + (0.05)(0.01ms + 0.1ms) =$
$0.0095 + 0.0055 = 0.015ms.$

The average access time is much closer to 0.01ms than to 0.1 ms, as desired.

ii) A two-way set-associative cache has lines of 16 bytes and total size of 8-Kbytes. The 64-Mbyte main memory is byte addressable. Show the format of main memory address.

Ans There are a total of 8 Kbytes / 16 bytes = 512 lines in the ~~cache~~ cache consists of 256 sets of 2 lines each. Therefore 8 bits are needed to identify the set number. For the 64-Mbyte main memory, a 26-bit address is needed. Main memory consists of 64 Mbyte / 16 bytes = 2^{22} blocks. Therefore, the set plus tag lengths must be 22 bits, so the tag length is 14 bits and the word field length is 4 bits.

Main memory Address =

TAG	Set	word.
14	8	4

iii) For the main memory address BBBBBBB1H, Show the following information in hexadecimal format.

a) Tag, line & word values for a direct-mapped cache.

b) Tag and word values for an associative cache.

c) Tag, set & word values for a two way set-associative cache.

Date: / /

(14)

Day

Address (H) BBBBBB

Address (binary) 101110111011101110111011

(a) Tag(8)/line(14)/word(2) BBH/2EEH/3H

(b) Tag(22)/word(2) 2EEEEEH/3H

(c) Tag(9)/set(13)/word(2) 17DH/0EEEH/3H