Name : Sajawal Khan

Department: Computer Science

ID : 14756

Subject : Computer Architetes

Semester : 4th

Submitted to: Sir Amin

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

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

(ii) Discuss different memory access methods in detail.

Another distinction among memory type is the method of accessing units of data. These includes the following:

- **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 seperate records and assist in the retrieval process. A shared read-write mechanism is used,

this must be moved from its current location to the desired location, passing and rejecting each intermediate record. Thus, the time to access an arbitary 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 and 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.

(iii) Discuss the importance of memory hierarchy:

Memory hierarchy is particularly important for understanding optimizations and performance costs that happen at the hardware level. Storing data on disk versus main memory can impact running time. The structure of page tables, virtual memory and look up caches also play a significant role.

(iv) How does the principle of locality relate to the use of multiple memory levels?

Slower and less expansive memory is used n higher stages with the most expansive eing the registers in the processer as well cache. Main memory is slower i ower and less expensive and is tside of the processor.

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

Direct mapping:

- It is 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 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 and associative approaches while reducing their disadvantages.

Q.3 Write note on each of the following:

(i) Memory unit of transfer:

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. In case of external memory unit of transfer is not limited to the word size, it is often larger and is referred to as blocks.

(ii) Memory performance parameters:

The two most important characteristics of memory are capacity 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 performed to the memory to the instant that data have been stored or made available to use.

- Memory Cycle time:

This concept is highly 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) **Disk Cache**

- Disk cache
  - 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 transfer of data can be used as a buffer instead of many small transfer of data.

- Data can be retrieved rapidly from the software cache rather than slowly from the disk.

(iv) Principle of locality:

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 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 physical caches because the logical cache is that cache can respond before ~~access~~ the MMU performs an address translation. The 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 as two different physical addresses.

(vi) Replacement algorithms:

Once the cache has been filled, when a new block is bought into the cache, one of the existing block 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 as algorithm must be implemented in hardware.

(vii) Possible approaches to cache coherency:

Possible approaches to cache coherency includes the following:

- Bus watching with write through. Each cache 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 policy 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-address bits.

Qno3:- Differentiate each of the following:

(i) 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 seperate records and assist in the

Strieval process. A shared read-write mechanism is used, and this must be moved from its current locations 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 block 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 independant 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.

(ii) Direct, associative and set-associative mapping:

Direct Mapping:

The direct mapping technique is simple and inexpensive to implement, its main disadvantage is that there is a field 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 Mapping:

With associative mapping there is a flexibility as to which block to replace when a new block is read into the cache, Replacement algorithms, discussed later in the 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 Mapping:

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

(iii) Split cache and unified cache:

Split cache:

- Has become common to split cache:
  - One dedicated to instructions.
  - One dedicated to data.
  - Both exist at the same level
- Trend is toward split caches at the L_1 and unified caches for higher levels.
- Advantages of split cache:
  - Eliminates cache contention between instruction fetch/decode unit of execution unit.
  - Important in pipelining.

Unified Cache:

- Trend is towards unified caches of higher levels.
- Advantages:
  - Higher hit rate
  - Balances load of instruction
  - Only one cache needs to be designed and implemented.

(iv) Written through and write back:

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

- Write Back
  - 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.

Q.4 Solve each of the following.

) Suppose that the processor has access two levels of memory. Level 1 contains oo words and has an access time 0.01 μs; level 2 contains 100,000 ords and has an access time of 1 μs. Assume that if a word to accessed is in level 1 and the essor asseses it directly. If it is level 2, then the word is first nsferred to level 1 and then

assesed by the processor. Suppose 95% of the memory accesses are found in level 1. Then find the everage time to access a word.

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

$(0.95)(0.01\,ms) + (0.05)(0.01\,ms + 0.1\,ms)$

$= 0.0095 + 0.0055 = 00.15\,ms$

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

(21)

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

There are a total of 8 kbytes/16 bytes = 512 lines in the cache. Thus the 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-Mbytes = 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.

(22)

Main memory Address =

Tag	Set	Word
14	8	4

Q NO #4

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) 177H/0EEH/3H