:: ASSIGNMENT # 4 ::

ID: 11533

Name: Ashir Ali Khan

Subject: Computer Architecture

Teacher: Sir Muhammad Amin

Iqra National University

Q1 Give detail answer to each of the fallaoing:

~~i What is the~~

ii Discuss different . . – –

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.

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 prioraccesses and is constant. Thus, any location can be selected at random and directly addrassed 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 the desired bits locations within a word for a specified match and to do this for all words simultaneously. Thus, a word is retriaved based on a postion of its contents rather than its address. As with ordinary ~~r~~ random-access memory each location has its own addressing mechanism and retrieval time is constant independent of location or prior patterns. Cache memories may employ associative access.

(i) What is the . . —

Answer:

An accesstime becomes faster, the cost per bit increases. As memomry size increases, the cost per bit is smaller. Also, with greater capacity, the access time becomes slower.

(iii) Discuss - - —

Importance of memory hierarchy.

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

(iv) How does . . - - ...

Answer:

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.

~~(IV) How main memory....~~
~~Direct mapping~~
~~it is the simplest technique.~~
~~Maps each block of main memory into only one possible cache line.~~

(IV) How does principle- . .

Answer:

It is possible to organize data across a memory hierarchy such that the percentage of access to each successively lower level is substantially less than that of the level above. Because memory references tend to clusters the data in the high ~~higher~~-level memory need not change very often to satisfy memory access requests.

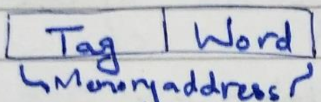
(V) How mainmemory: . ~-

Direct Mapping: In direct mapping each main mem address can be viewed as consisting of three fields. Tag, line and word: It maps each block of memory main into only one cache line

Tag	Line	Word

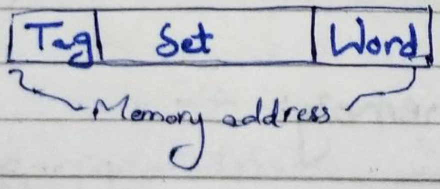
Associative Mapping:

In this case, the cache control logic interprets a memory address simply as a Tag and a word field. The Tag field uniquely identifies a block of main memory. To determine weather a block is in the cache the cache ~~[illegible]~~ control logic must simultaneously examine every line's tag for a match.



Set-Associative Mapping:

The set-Associative mapping, the cache control logic interprets a memory address as three fields: Tag, Set and Word. With set-Associative mapping the tag in a main memory address is quite large and must be compared to the tag of every line in the cache.



Q2 Write note on each of the following:

(i) Memory 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 wordsize. In case of external memory, unit of transfer is not limited to the wordsize. It is often larger and is reffered to as blocks.

(ii) Memory performance parameters:

The two most important characteristics of memory are capacity and ~~parameters~~ 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 the data have been stored or made available for use. For ~~non~~ 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 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:

A diskcache is a cache memory that is used to speedup the process of storing and accessing data from the host hard disk. It enables faster processing of reading/ writing, commands and other inputs and outputs process between the hard disks, the main memory and computing components. A disk cache is also reffered as a disk buffer and cache buffer.

- 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:

The principle of locality states that data in the vacinity of a referenced word are likely to be reffered in the nearfuture.

OR

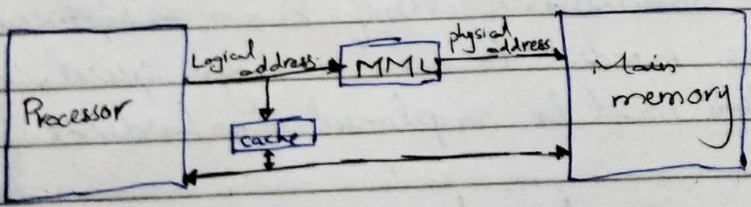
An implication of locality is that we can predict with reasonable accuracy what instructions and data a program will use in the near future based on its accesses in the recent past.

It reffers to the tendency of the computer program to access instructions whose addresses are near one another. The property of locality of reference is mainly shown by loops and subroutine calls in program.

(V) Logical cache and physical cache:

Logical Cache:

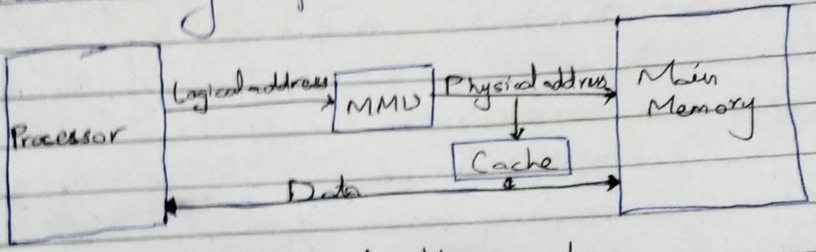
A logical cache stores data in a virtual space. A logical cache is located b/w the processor and the MMU. The processor can access data from a logical cache directly without going through the MMU. A logical cache is also known as a virtual cache.



Logical cache:

Physical Cache:

A physical cache stores memory using physical cache. A physical cache is located between the MMU and mainmemory. For the processor to access memory. The MMU must first translate the virtual address to a physical address before the cache memory can provide data to the core.



Physical cache:

(VI) Replacement Algorithm:

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:

Possible approaches to cache coherency includes 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 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 caches this update is written to main memory. In addition any matching words in other cache 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.

Q³ Differentiate each of the following.

(i) Sequential, direct and random access methods.

Ans. **Sequential access method:**

Memory is organized into units of data, called records. Access must be made in specific linear sequence.

~~**Direct access method.**~~

~~Individual blocks or records have 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.~~

Direct access method:

Individual blocks or records have a unique address based on physical location. Access is accomplished by direct access to reach a general vacinity plus sequential searching, counting or waiting to reach the final location.

Random access method.

Each addressable location in memory has a unique, physically witedin addressing mechanism. The time to access a given location is independent of the sequence of prior accesses and is constant.

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

Direct:	Associative	Set-Associative
In a cache system, direct mapping maps each block of main memory into only one possible cache line.	In associative, mapping permits each main memory block to be loaded into any line of the cache.	In set-associative mapping the cache is divided into a number of sets of cache lines, each main memory block can be mapped into any line in a particular set.

(iii) Split cache and unified cache:-

Split cache:

Has become common to splitcache:

- One dedicated to instructions.
- One dedicated to data.
- Both exist at the same level, typically as two L_1. caches

- Trend is towards split cache at the L_1 and unified caches for higher levels
- It balances load between data and instruction automatically.
- ~~Only one cache is needed to design.~~

Unified cache:

- Earlier same cache is used for data as well as instructions.
- It balances load
- If execution involves more instructions fetches the cache will tend to fill up with instructions and if execution involves more of data fetches, the cache tends to fill up the data.
- Only one cache is needed to design.

(IV) Write through and Write back.

Write through	Write back.
• Simplest technique • All write operators are made to main memory as well as to the cache • Slows down writes • Lots of traffic	• Updates initially made in cache only • Other cache get out of sync • I/O must access main memory through cache • Update bit for cache slot is set when update occurs.

Q4 Solve each of the following:

(1) Suppose . . - -- a word.

Solution:

The average time to access a word can be expressed as:

$= (0.95)(0.1\mu s) + (0.05)(0.01\mu s + 0.1\mu s)$

$= 0.0095 + 0.0055$

$= 0.015\mu s$

The average access time is much closer to 0.01µs ~~tan~~ than to 0.1µs, as desired

Note: 95% at L1 cache and 5% at L2 cache.

ii) A two way set . - - - . . .. addresses.

Solution:

Given Data:

Two way set associative mapping
Size of each cache line = 16 bytes.
Size of memory = 8 Kb
Byte addressable = 64MB.

Now:

Numbers of ~~[illegible]~~ cache lines $= \frac{Kb \times 1028}{\text{No of bytes}}$ — (1).

2020/6/10 04:23

$= \frac{8 \times 1024}{16}$

- Therefore no of cache lines is '512" since its two way associative the cache consists of 2 set each consists of 256 cache lines.
- Number of bits required for "set field" in main memory address format is 8-bits because $2^8 = 256$
- Second step is to calculate number of blocks in main memory:

Number of blocks in mainmemory $= \frac{64MB}{16 bytes}$

$$= \frac{2^6 \times 2^{20} \times 2^3}{2^4 \times 2^3}$$

$= 2^{22}$ blocks.

Therefor the set plus tag must be of 22 bits, so the tag length is 14 and the word length is 4 bits:

	TAG	SET	WORD.
Mainmemory address =	14	8	4

(iii) For the main . ―― ――

Solution:

Address (H)	BBBBBB
Address (Binary)	101110111011101110111011
a Tag(8) / Line(14) / Word(2)	BBH / 2EEEH / 3H
b Tag (22) / Word (2)	2EEEEEH / 3H
c Tag(9) / Set (13) / Word(2)	177H / 0EEEH / 3H