# Assignment # 4

Name : Sana Urooj

ID # 11575

Subject : Computer Architecture

Submitted To : Sir Amin

# 1

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

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

(ii) Discuss different Memory access methods in detail?

Ans) Another distinction among memory types is the method of accessing units of data. These include 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 separate records and assist in the retrieva 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 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 access 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 ~~me~~ 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 addressing mechanism, and retrieval time is constant independent of location or prior access patterns. Cache memories may employ associative access.

(iii) Discuss the importance of memory hierarchy?

Ans) 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 lookup caches also play a significant role.

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

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

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

Ans) 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 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 strength of both the direct and associative approaches while reducing their disadvantages.

## Question # 2

(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. In case of external memory, unit of

Transfer is not limited to the word size; it is often larger and is refferred to as blocks.

(ii) Memory performance parameters?

(Ans) The two most important Characteristic 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 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 consist 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) ~~Da~~ Disk Cache ?

Ans) **DISK CACHE** 8

- 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.

(v) Logical Cache & Physical Cache ?

(Ans) A logical cache, also know as a virtual cache, stores data using virtual address. The processor accesses the cache directly, without going through the MMU. A physical cache store 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 an address translation. The disadvantage has to do with the fact that most virtual memory systems ~~fy~~ 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 ~~mat~~ 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 TRANPARENCY :
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-CACHABLE MEMORY: Only a portion of main memory is shared by more than one processor, and this is ~~designed~~ 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.

## Question # 3

(i) Sequential, direct, and random access method ?

(Ans) SEQUENTIAL ACCESS: Memory is organized into units of data, called records. Access must be made in 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 having 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~~ are random access.

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

Ans) DIRECT MAPPING :
The direct mapping technique is simple and inexpensive to implement. Its main disadvantage 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 (a phenomenon known as thrashing).

- **ASSOCIATIVE MAPPING:**
  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 principle 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.

(Q.9) Split cache and Unified cache?

Ans) **SPLIT CACHE:**
Has become common to split cache;
- one dedicated to instruction.
- one dedicated to data.
- Both exist at the same level,

typically as two L1 caches.
- Trend is toward split caches at the L1 and unified caches for higher levels.
- Advantages of split cache;
- Eliminates cache contention between instruction fetch/decode unit and execution unit.
- Important in pipelining.

UNIFIED CACHE:
- Trend is toward unified caches for ~~hig~~ higher levels.
- Advantages of unified cache;
- Higher hit rate
- Balances load of instruction and data fetches automatically.
- Only one cache needs to be designed and implemented.

(iv) Write through and write back?

— WRITE THROUGH:
- Simplest technique
- All write operations are made to main memory as well as to cache.
- The main disadvantage of this 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. This makes for complex circuitry and a potential bottleneck.

## Question # 4

(P) 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

$(0.95)(0.01\text{ ms}) + (0.05)(0.01\text{ ms} + 0.1\text{ ms})$
$= 0.0095 + 0.0055 = 0.015\text{ ms}$

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

# 17

(ii) There are 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 / 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 |