

Memory Systems



School of Computer Science G51CSA

1

Computer Memory System Overview

- └ Historically, the limiting factor in a computer's performance has been memory access time
- └ Memory speed has been slow compared to the speed of the processor
- └ A process could be bottlenecked by the memory system's inability to "keep up" with the processor



School of Computer Science G51CSA

2

Computer Memory System Overview

Terminology

- └ **Capacity:** (For internal memory) Total number of words or bytes. (For external memory) Total number of bytes.
- └ **Word:** the natural unit of organization in the memory, typically the number of bits used to represent a number - typically 8, 16, 32
- └ **Addressable unit:** the fundamental data element size that can be addressed in the memory -- typically either the word size or individual bytes
- └ **Access time:** the time to address the unit and perform the transfer
- └ **Memory cycle time:** Access time plus any other time required before a second access can be started



School of Computer Science G51CSA

3

Memory Hierarchy

- └ **Major design objective of any memory system**
 - └ To provide adequate storage capacity at
 - └ An acceptable level of performance
 - └ At a reasonable cost
- └ **Four interrelated ways to meet this goal**
 - └ Use a hierarchy of storage devices
 - └ Develop automatic space allocation methods for efficient use of the memory
 - └ Through the use of virtual memory techniques, free the user from memory management tasks
 - └ Design the memory and its related interconnection structure so that the processor can operate at or near its maximum rate



School of Computer Science G51CSA

4

Basis of the memory hierarchy

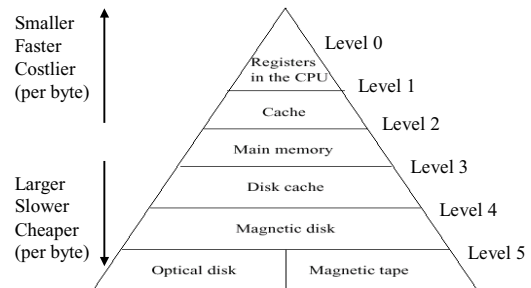
- └ Registers internal to the CPU for temporary data storage (small in number but very fast)
- └ External storage for data and programs (relatively large and fast)
- └ External permanent storage (much larger and much slower)
- └ Remote Secondary Storage (Distributed File Systems, Web Servers)



School of Computer Science G51CSA

5

The Memory Hierarchy



School of Computer Science G51CSA

6

Typical Memory Parameters

Memory Type	Technology	Size	Access Time
Cache	Semiconductor RAM	128-512 KB	10 ns
Main Memory	Semiconductor RAM	4-128 MB	50 ns
Magnetic Disk	Hard Disk	Gigabyte	10 ms, 10 MB/sec
Optical Disk	CD-ROM	Gigabyte	300 ms, 600 KB/sec
Magnetic Tape	Tape	100s MB	Sec-min., 10MB/min



School of Computer Science G51CSA

7

Typical Memory Parameters

Suppose that the processor has access to two levels of memory. Level 1 contains 1000 words and has an access time of $0.01 \mu\text{s}$; level 2 contains 100,000 words and has an access time of $0.1 \mu\text{s}$. Assuming that

if a word to be accessed is in level 1, then the processor access it directly.

If it is in level 2, then the word is first transferred to level 1 and then accessed by the processor.

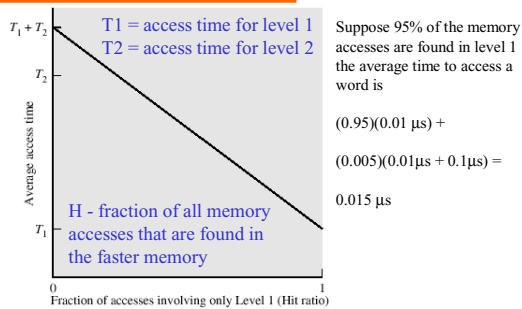
For simplicity, ignore the time required for the processor to determined whether is in level 1 or level 2. A typical performance of a simple two level memory has this shape:



School of Computer Science G51CSA

8

Typical Memory Parameters



School of Computer Science G51CSA

9

The Locality Principle

The memory hierarchy works because of *locality of reference*

- Well written computer programs tend to exhibit good locality. That is, they tend to reference data items that are near other recently referenced data items, or that were recently referenced themselves. This tendency is known as the locality principle.
- All levels of modern computer systems, from the hardware, to the operating system, to the application programs, are designed to exploit locality.



School of Computer Science G51CSA

10

The Locality Principle

- At hardware level, the principle of locality allows computer designers to speed up main memory accesses by introducing small fast memories known as the *cache memories*.
- At operating system level, main memory is used to cache the most recently referenced chunks of virtual address space and the most recently used disk blocks in a disk file system.
- At application level, Web browsers cache recently referenced documents in local disk

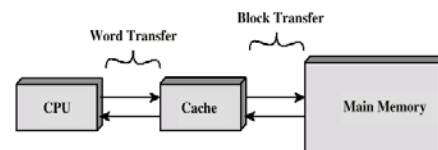


School of Computer Science G51CSA

11

Cache Memory

- Small amount of fast memory
- Sits between normal main memory and CPU
- May be located on CPU chip or module
- Intended to achieve high speed at low cost



School of Computer Science G51CSA

12

Cache Memory

- Cache retains copies of recently used information from main memory, it operates transparently from the programmer, automatically decides which values to keep and which to overwrite.
- An access to an item which is in the cache: **hit**
- An access to an item which is not in the cache: **miss**
- The proportion of all memory accesses that are found in cache: **hit rate**



School of Computer Science G51CSA

13

Cache operation - overview

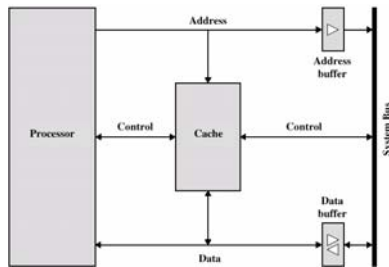
- CPU requests contents of memory location
- Check cache for this data
- If present, get from cache (fast)
- If not present, read required block from main memory to cache
- Then deliver from cache to CPU



School of Computer Science G51CSA

14

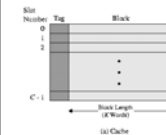
Typical Cache Organization



School of Computer Science G51CSA

15

Cache/Main Memory Structure



- Main memory consists of fixed length blocks of K words ($M = 2^s/K$ blocks)

- Cache consists of C Lines of K words each

- The number of lines is much less than the number of blocks ($C \ll M$)

- Block size = Line Size

Cache includes **tags** to identify which block of main memory is in each cache slot



School of Computer Science G51CSA

16

Mapping Function

- Fewer cache line than main memory block
- Need to determine which memory block currently occupies a cache line
- Need an algorithm to map memory block to cache line

- Three Mapping Techniques:

- Direct**

- Associative**

- Set associative**



School of Computer Science G51CSA

17

Direct Mapping

- Each main memory address can be viewed as consisting 3 fields:

2 The least significant w bits identify a unique word or byte within a block of main memory

2 The remaining s bits specify one of 2^s blocks of main memory

2 The cache logic interprets these s bits as:

2 a tag field of $s - r$ bits (most significant portion)

2 a line field of r bits



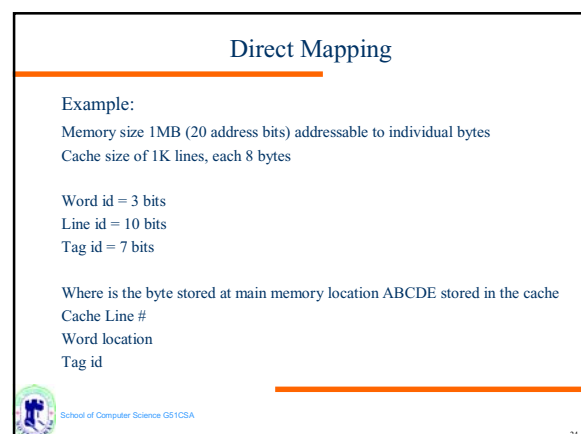
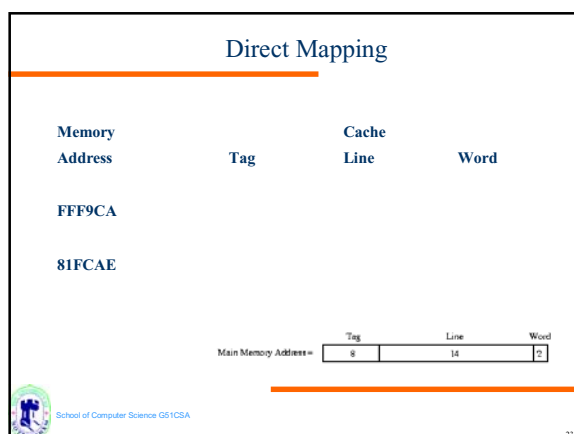
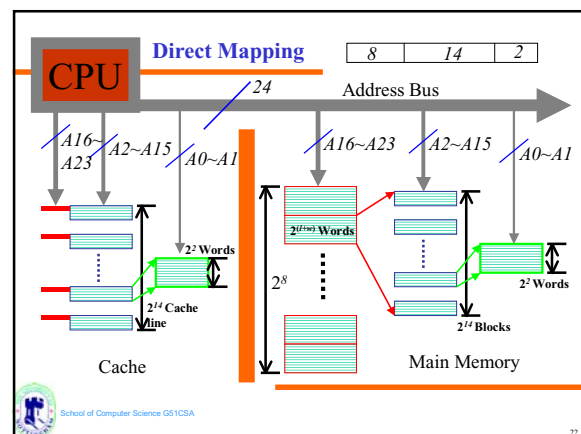
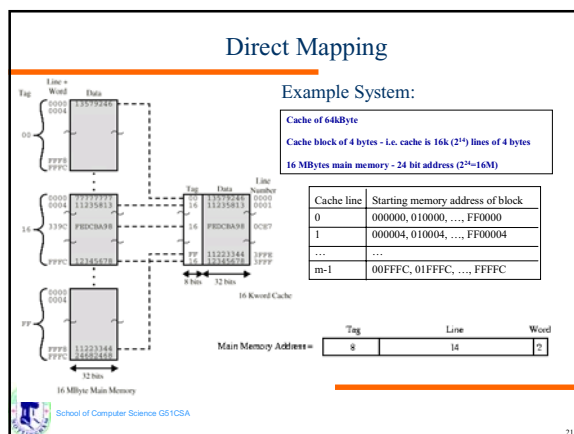
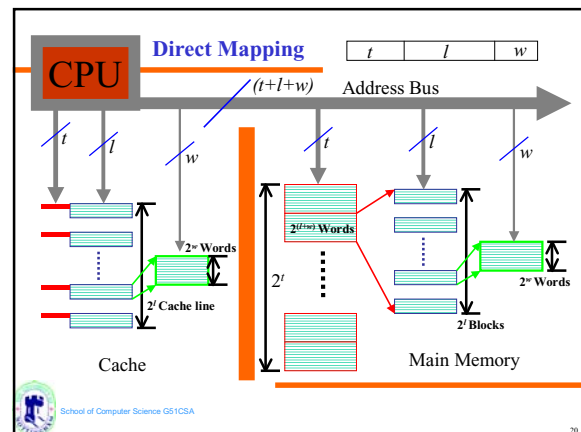
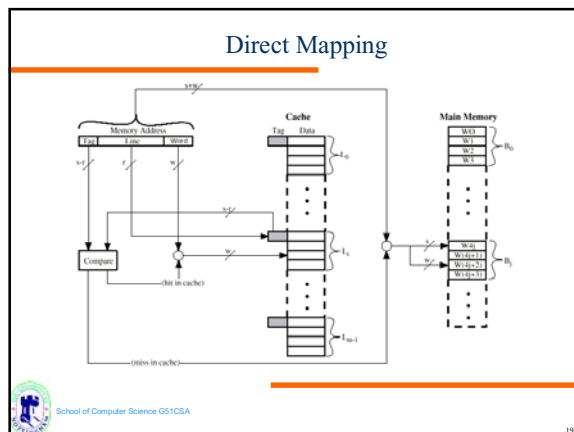
Cache line	Main Memory blocks held
0	0, m, 2m, 3m... 2^s-m
1	1, m+1, 2m+1... 2^s-m+1
...	...
m-1	m-1, 2m-1, 3m-1... 2^s-1

$m=2^r$ line of cache



School of Computer Science G51CSA

18



Direct Mapping

Simple

Inexpensive

Fixed location for given block

If a program accesses 2 blocks that map to the same line repeatedly, cache misses are very high



School of Computer Science G51CSA

25

Associative Mapping

m A main memory block can load into any line of cache

m Memory address is interpreted as tag and word

m Tag uniquely identifies block of memory

m Every line's tag is examined for a match

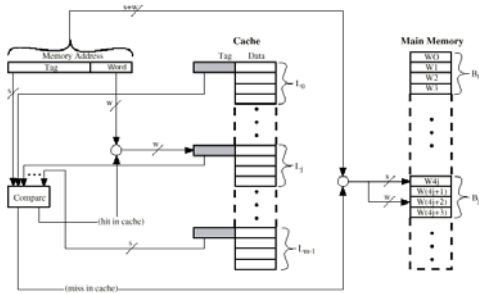
m Cache searching gets expensive



School of Computer Science G51CSA

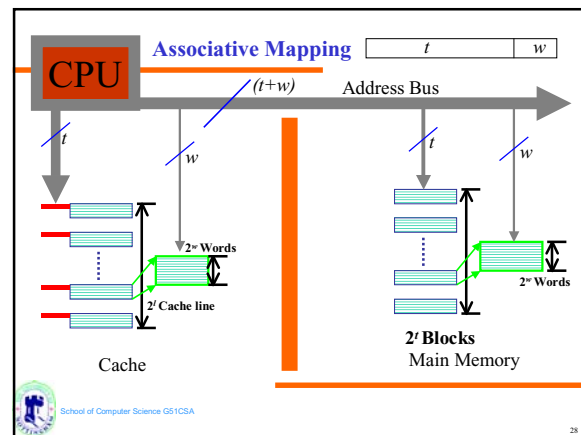
26

Associative Mapping



School of Computer Science G51CSA

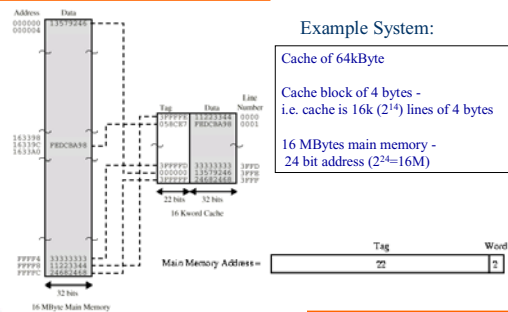
27



School of Computer Science G51CSA

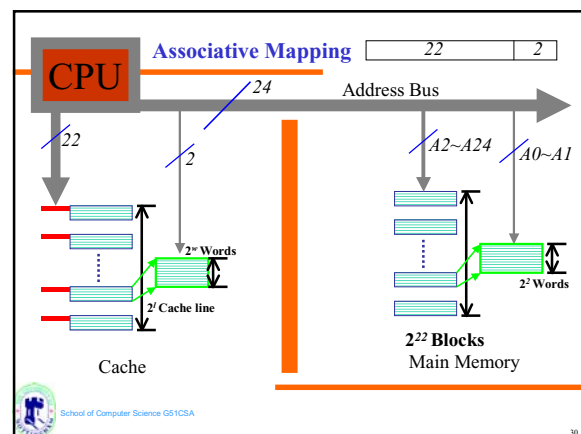
28

Associative Mapping



School of Computer Science G51CSA

29



School of Computer Science G51CSA

30

Associative Mapping

Memory

Address

FFF9CA

81FCAE

Cache

Tag

Word



School of Computer Science G51CSA

31

Associative Mapping

Example:

Memory size 1MB (20 address bits) addressable to individual bytes

Cache size of 1K lines, each 8 bytes

Word id = 3 bits

Tag id = 17 bits

Where is the byte stored at main memory location ABCDE stored in the cache

Word location
Tag id



School of Computer Science G51CSA

32

Set Associative Mapping

- Cache is divided into a number of sets
- Each set contains a number of lines
- A given block maps to any line in a given set

Address length = $(s + w)$ bits

Number of addressable units = 2^{s+w} bytes or words

Block size = line size = 2^w bytes or words

Number of blocks in main memory = $(2^{s+w})/2^w = 2^s$

Number of lines in set = k

Number of sets $v = 2^d$

Number of lines in cache = $k v = k \times 2^d$

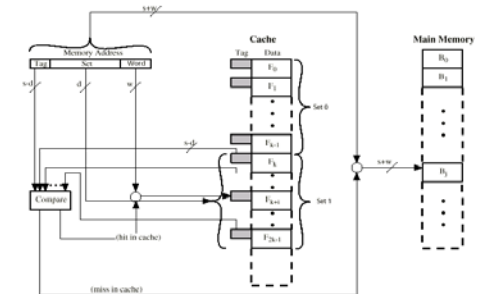
Size of tag = $(s - d)$ bits



School of Computer Science G51CSA

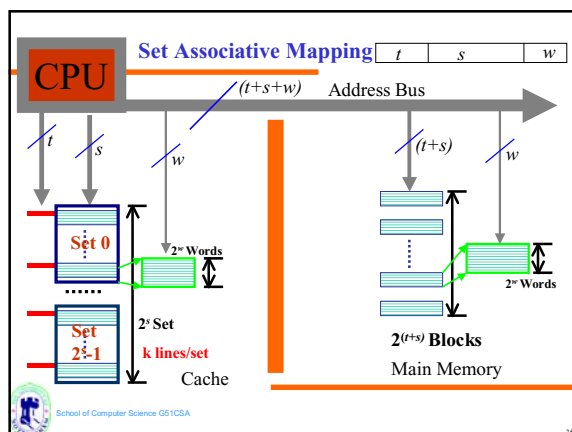
33

Set Associative Mapping



School of Computer Science G51CSA

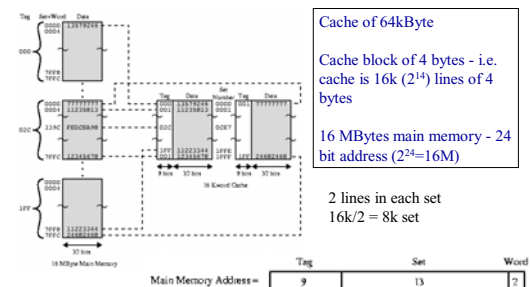
34



School of Computer Science G51CSA

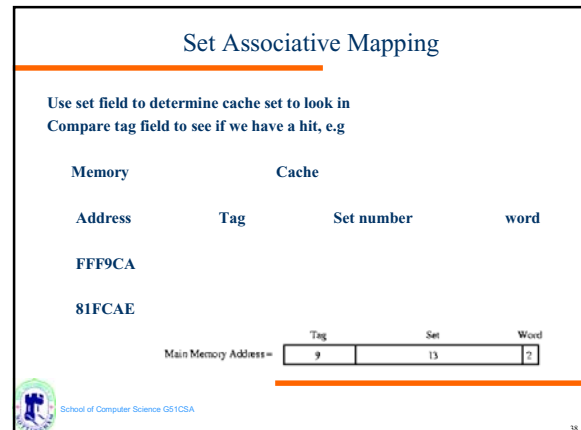
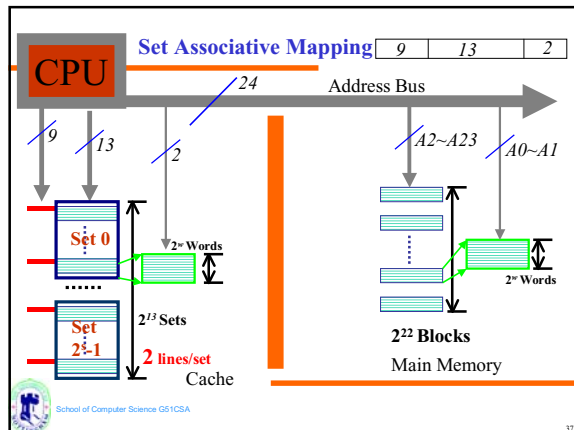
35

Set Associative Mapping



School of Computer Science G51CSA

36



Set Associative Mapping

Example:
Memory size 1MB (20 address bits) addressable to individual bytes
Cache size of 1K lines, each 8 bytes

4-way set associative mapping $1024/4 = 256$ sets

Word id = 3 bits Set id = 8 bit Tag id = 17 bits

Where is the byte stored at main memory location ABCDE stored in the cache

Word location
Set
Tag

School of Computer Science GS1CSA

Replacement Algorithms

- When a new block is brought into the cache, one of the existing blocks must be replaced.
- Direct Mapping: One possible line for any particular block - No choice
- Associative/Set Associative Mapping:
 - Least Recently used (LRU): Replace block that has not been referenced the longest. E.g. in 2 way set associative, Which of the 2 block is LRU?
 - First in first out (FIFO): replace block that has been in cache longest
 - Least frequently used: replace block which has had fewest hits
 - Random

School of Computer Science GS1CSA

Write Policy

- Before a block that is resident in the cache can be replaced, it is necessary to consider whether it has been altered in the cache but not in the main memory.
- If it has not (been altered in cache), then the old block in the cache can be overwritten.
- If it has (been altered in cache), it means at least one write operation has been performed on a word in that cache line and main memory must be updated accordingly.

School of Computer Science GS1CSA

Write Policy

Write Through:
All writes go to main memory as well as cache
Multiple CPUs can monitor main memory traffic to keep local (to CPU) cache up to date
Lots of traffic
Slows down writes

Write Back
Updates initially made in cache only
Update bit for cache slot is set when update occurs
If block is to be replaced, write to main memory only if update bit is set
Other caches get out of sync
I/O must access main memory through cache
N.B. 15% of memory references are writes

School of Computer Science GS1CSA

Line Size

Larger blocks reduce the number of blocks that fit into the cache.

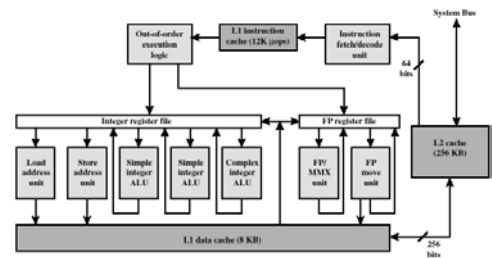
As block becomes larger, each additional word is farther from the requested word, therefore less likely to be needed in the near future



School of Computer Science G51CSA

43

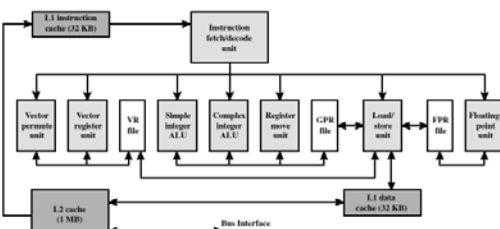
Pentium 4 Cache



School of Computer Science G51CSA

44

PowerPC Cache



School of Computer Science G51CSA

45

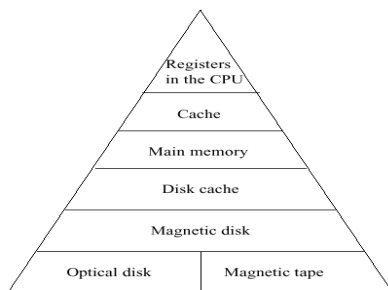
External Memory



School of Computer Science G51CSA

46

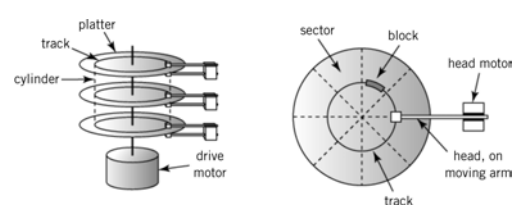
Memory Hierarchy



School of Computer Science G51CSA

47

Magnetic Disks



Engelen: The Architecture of Computer Hardware and Systems Software, 2nd edition
Chapter 9, Figure 09-02

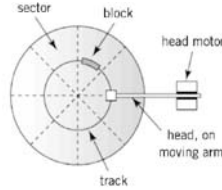


School of Computer Science G51CSA

48

Magnetic Disks

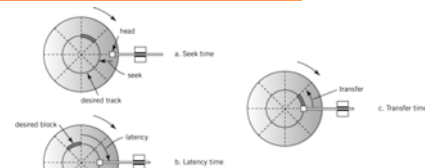
- Each sector on a single track contains one block of data, typically 512 bytes, and represents the smallest unit that can be independently read or written.
- Regardless of the track, the same angle is swept out when a sector is accessed, thus the transfer time is kept constant when the motor rotating at a fixed speed. This technique is known as CAV - Constant Angular Velocity.



School of Computer Science G51CSA

49

Magnetic Disks



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 9, Figure 09-05



School of Computer Science G51CSA

50

Seek time: the time required to move from one track to another
Latency time: After the head is on the desired track, the time taken to locate to correct sector.
Transfer time: Time taken to transfer one block of data.

Magnetic Disks

After the head is on the desired track, the time taken to locate to correct sector

Maximum Latency Time

Average Latency Time

Time taken to transfer one block of data

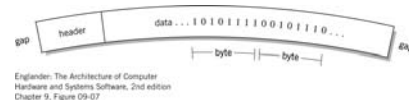
Transfer Time



School of Computer Science G51CSA

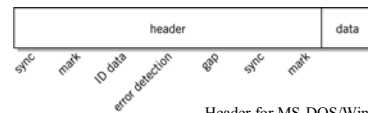
51

Magnetic Disks



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 9, Figure 09-07

A single data block



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 9, Figure 09-08

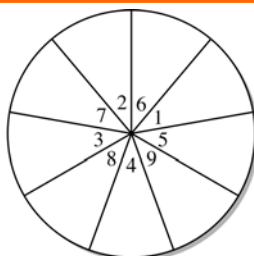
Header for MS-DOS/Windows disk



School of Computer Science G51CSA

52

Magnetic Disks



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 9, Figure 09-09

Disk interleaving



School of Computer Science G51CSA

53

Magnetic Disks

A floppy disk is rotating at 300 rpm (revolutions per minute). The disk is divided into 12 sectors, with 40 tracks on the disk. The disk is singled sided. A block consists of a single sector on a single track. Each block contains 200 bytes.

What is the disk capacity in bytes?

What is the maximum and minimum latency time for this disk?

What is the transfer time for a single block?



School of Computer Science G51CSA

54

Magnetic Disks

A multiplattered hard disk is divided into 40 sectors and 400 cylinders. There are four platter surfaces. The total capacity of the disk is 128 MB. A cluster consists of 4 blocks. The disk is rotating at a rate of 4800 rpm. The disk has an average seek time of 12 msec.

What is the capacity of a cluster for this disk?

What is the disk transfer rate in bytes per second?

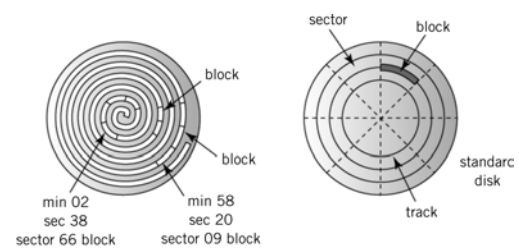
What is the average latency time for the disk?



School of Computer Science GS1CSA

55

Optical Disks



England: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 9, Figure 09-13



School of Computer Science GS1CSA

56

Optical Disks

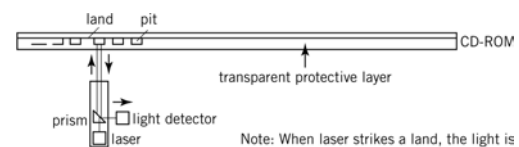
- ⌞ CD format designed for maximum capacity
- ⌞ Each block the same length along the track, regardless of locations
- ⌞ More bits per revolution at the outside of the disk than at the inside
- ⌞ A variable speed motor is used to keep transfer rate constant
- ⌞ The disk move slower when the outside tracks are be read
- ⌞ Constant Linear Velocity, CLV



School of Computer Science GS1CSA

57

Optical Disks



Note: When laser strikes a land, the light is reflected into the detector; when the light strikes a pit, it is scattered.

England: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 9, Figure 09-14



School of Computer Science GS1CSA

58

Others

- ⌞ Tape
- ⌞ RAID
- ⌞ ...



School of Computer Science GS1CSA

59