DISK MANAGEMENT

Chapter 5.3

SYLLABUS

- Disk Management
  - Disk Structure
  - Disk Scheduling Algorithm
  - Error Handling and Formating
  - Stable Storage Management

DISK STRUCTURE

- Disks come in a variety of types.
- **Magnetic Disks**(Hard Disk, Floppy Disk)
  - reads and writes are equally fast
  - ideal as secondary memory (paging, file systems, etc.)
  - Arrays of these disks are sometimes used to provide highly-reliable storage
- **Optical Disks**(CD ROM, DVD ROM)
  - For distribution of programs, data, and movies

Magnetic Disks

- How does a hard disk work?

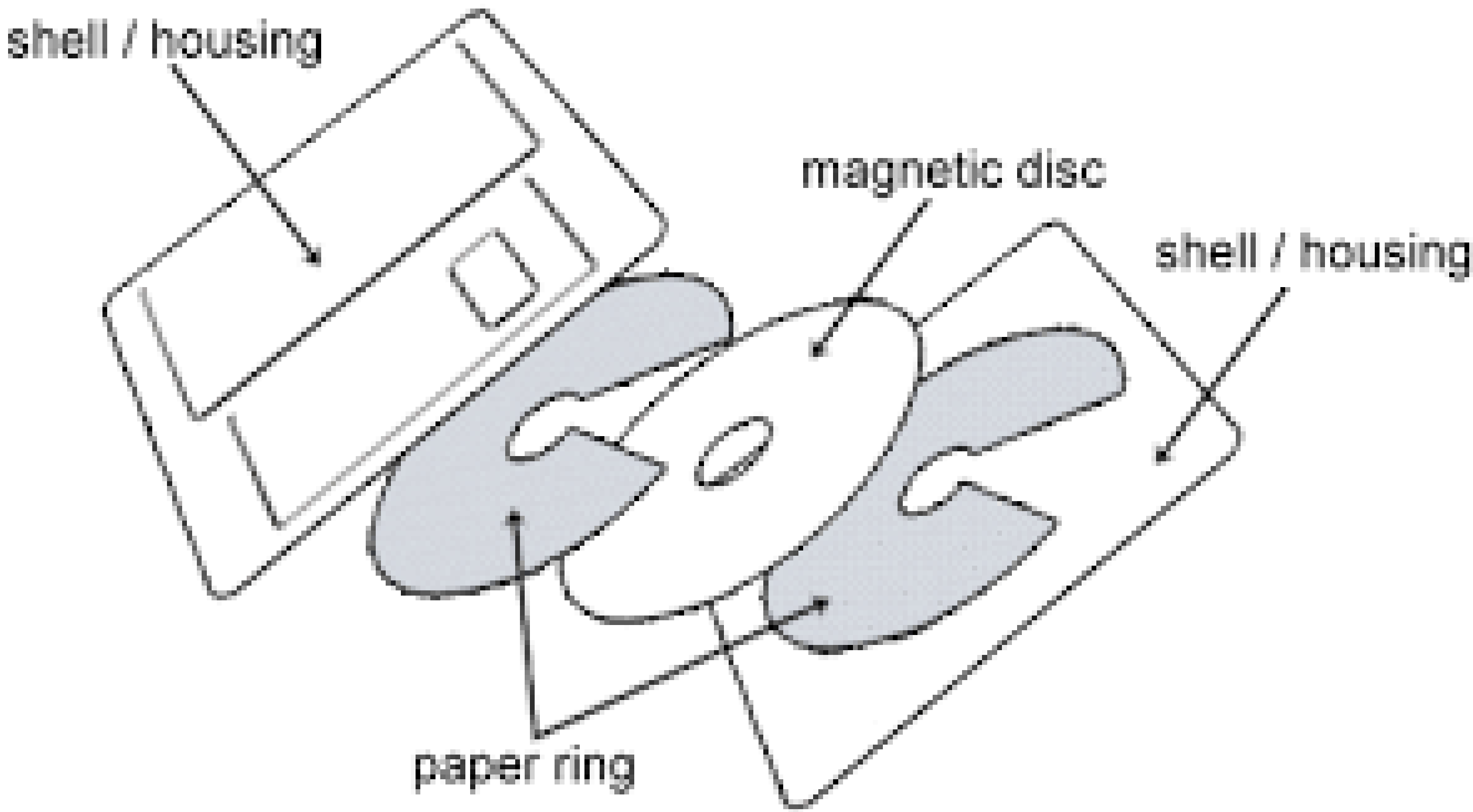
Step 1.
Circuit board controls movement of head actuator and a small motor.

Step 2.
Small motor spins platters while computer is running.

Step 3.
When software requests a disk access, read/write heads determine current or new location of data.

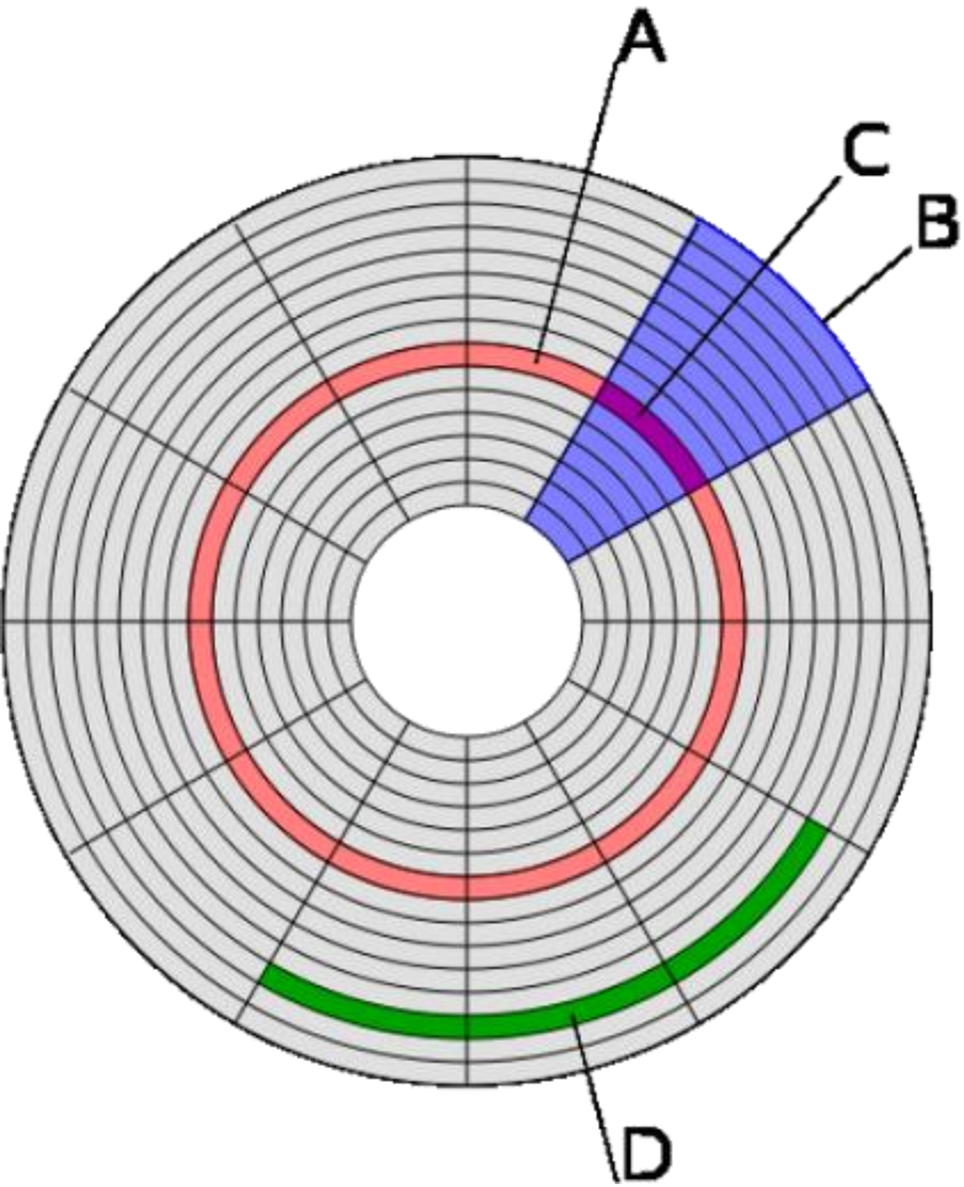
Step 4.
Head actuator positions read/write head arms over correct location on platters to read or write data.

MAGNETIC DISKS



MAGNETIC DISKS

- Disk surface is divided into number of logical block called ***sectors*** and ***tracks***.
- The term ***cylinder*** refers to all the tracks at particular head position in hard disk.



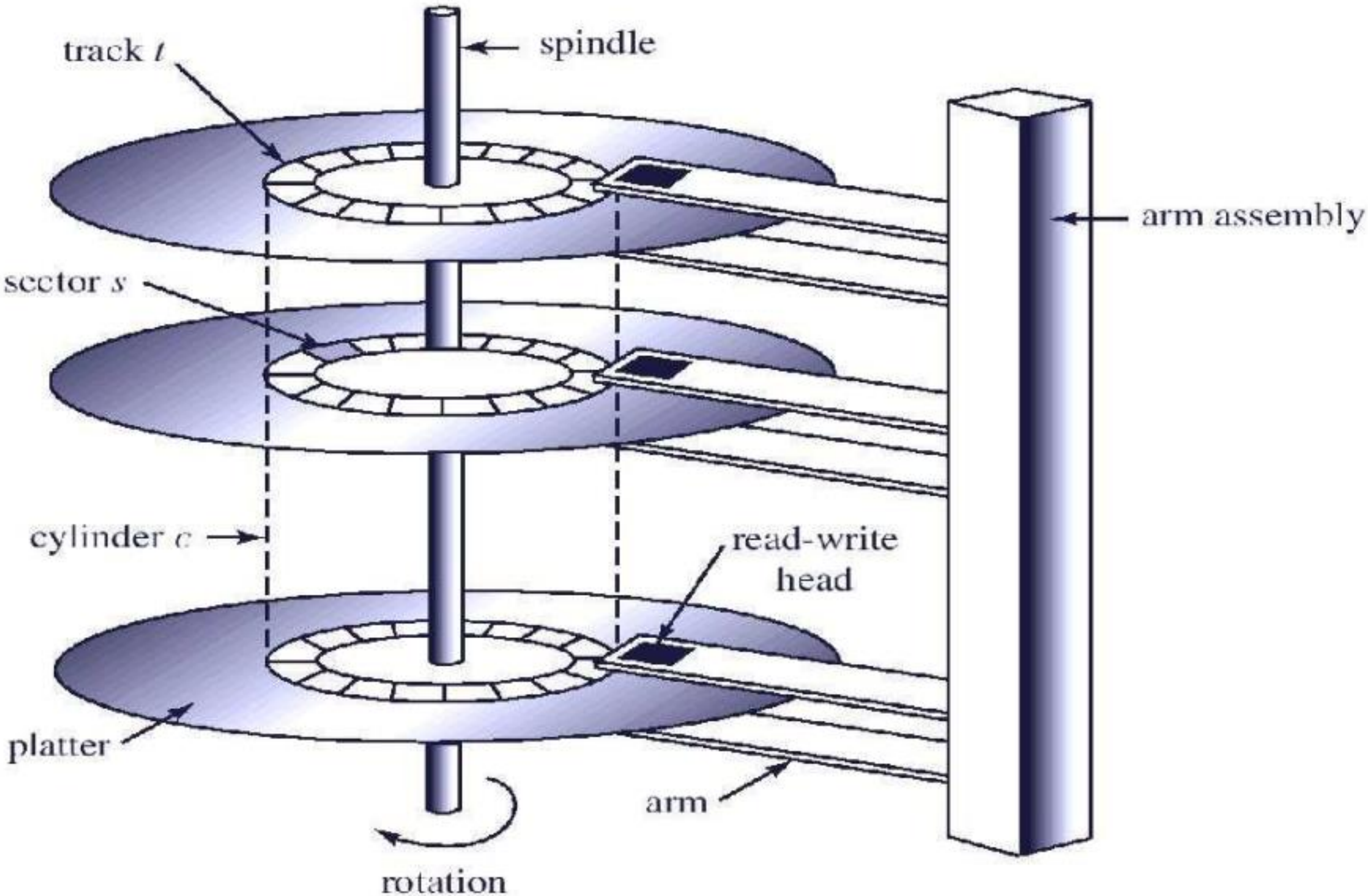
Hard Drive Structure:

A = track

B = sector

C = sector of a track

D = cluster


track t
spindle
arm assembly
sector s
cylinder c
read-write
head
platter
arm
rotation

MAGNETIC DISKS

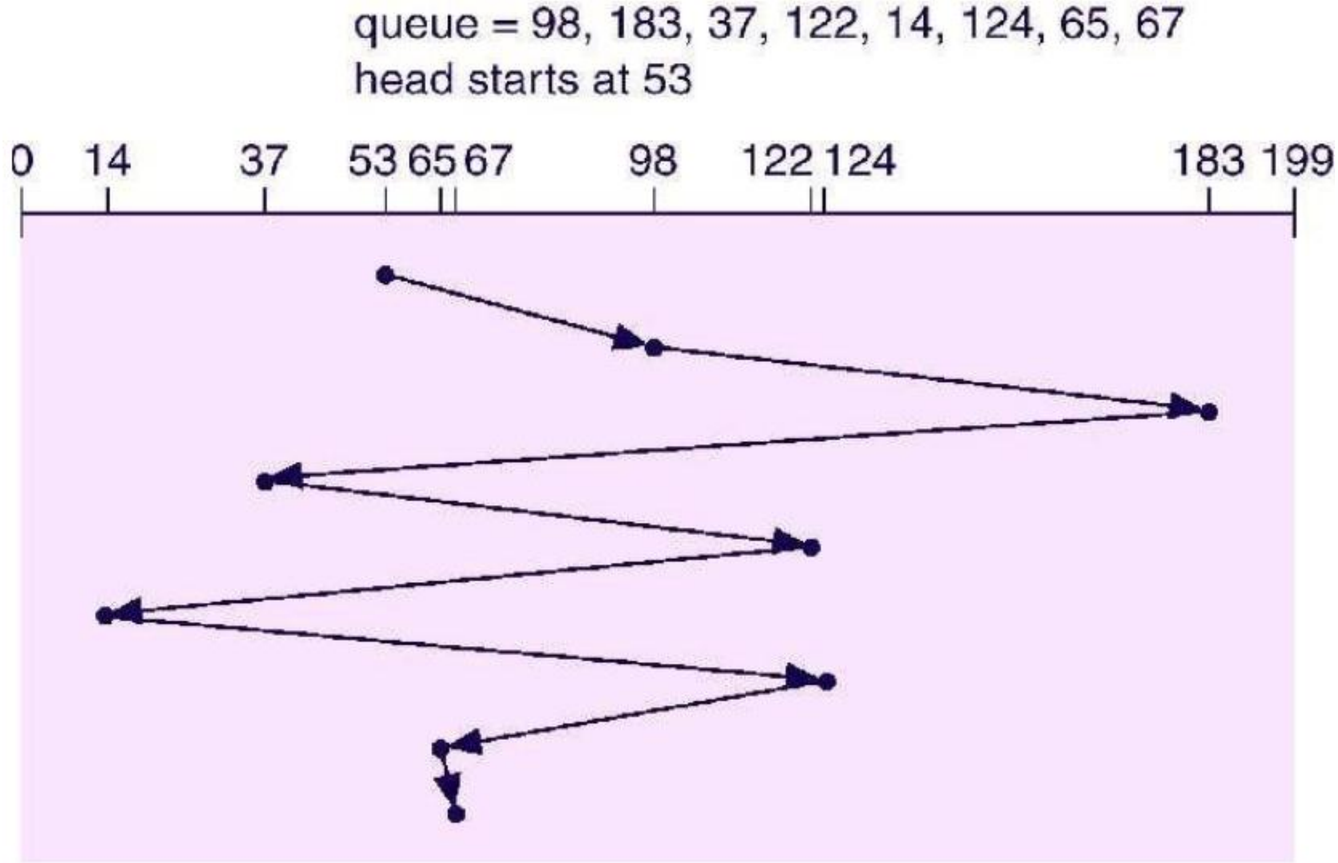
- ***Latency Time:*** The time taken to rotate from its current position to a position adjacent to the readwrite head.
- ***Seek:*** The process of moving the arm assembly to new cylinder.
- ***To access a particular record****, first the arm assembly must be moved to the appropriate cylinder, and then rotate the disk until it is immediately under the read-write head.*
- The time taken to access the whole record is called ***transmission time***.

DISK ARM SCHEDULING ALGORITHM

- *OS is responsible to use the hardware efficiently –*
  - *for the disk drive this means fast seek, latency and transmission time*
- For most disks, the ***seek time*** **dominates the other two** times, so reducing the *mean seek time* can improve system performance substantially

FIRST-COME FIRST-SERVED (FCFS)

- If the disk driver accepts requests one at a time and carries them out in that order, that is, First-Come, First-Served (FCFS).

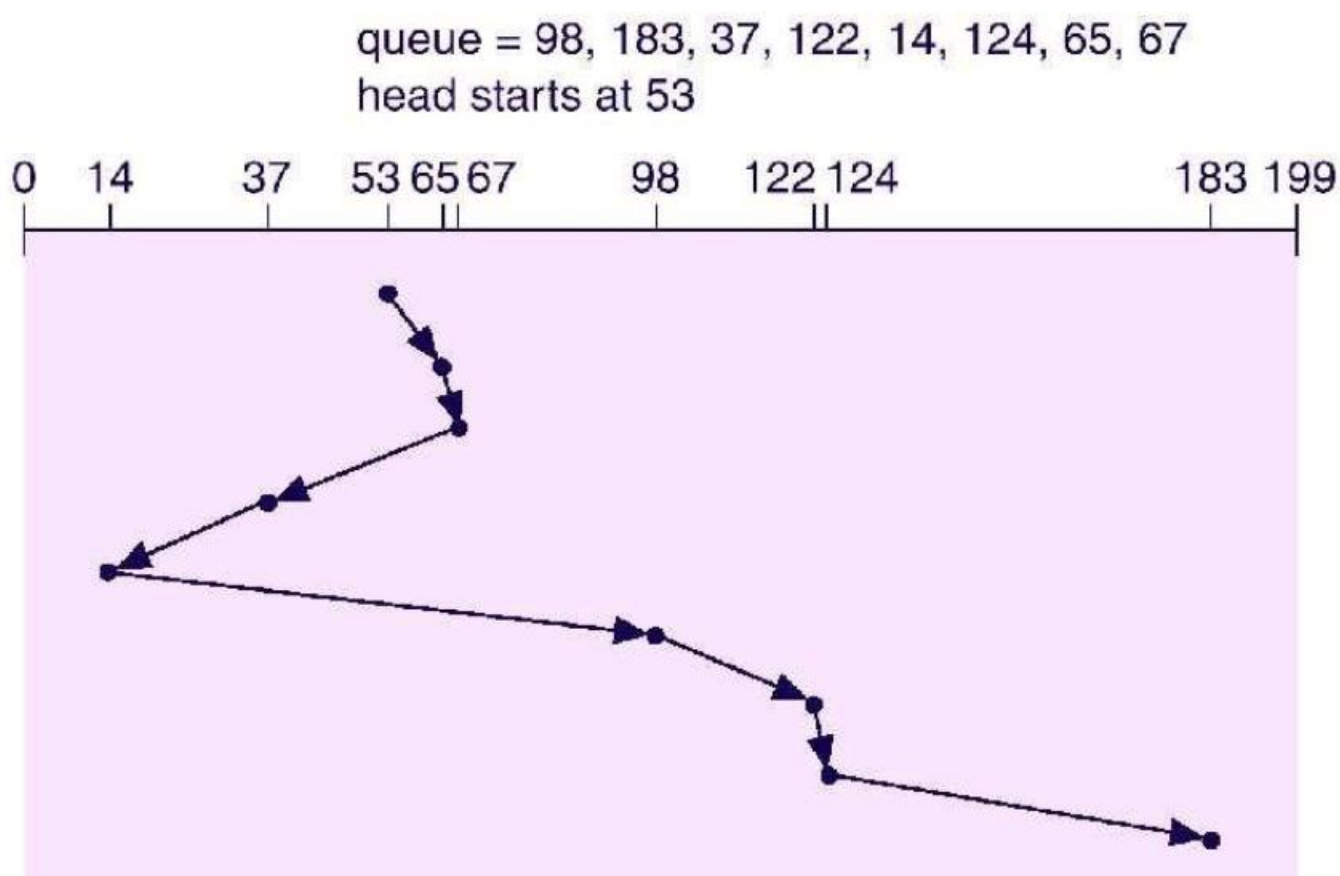


- Total Head movement 640 cylinders
- Advantage:
  - Simple and Fair
- Problems:
  - Does not provide fastest service

SHORTEST SEEK TIME FIRST (SSTF)

- Selects the request with the minimum seek time from the current head position

- Total head movement =236 cylinders



SHORTEST SEEK TIME FIRST (SSTF)

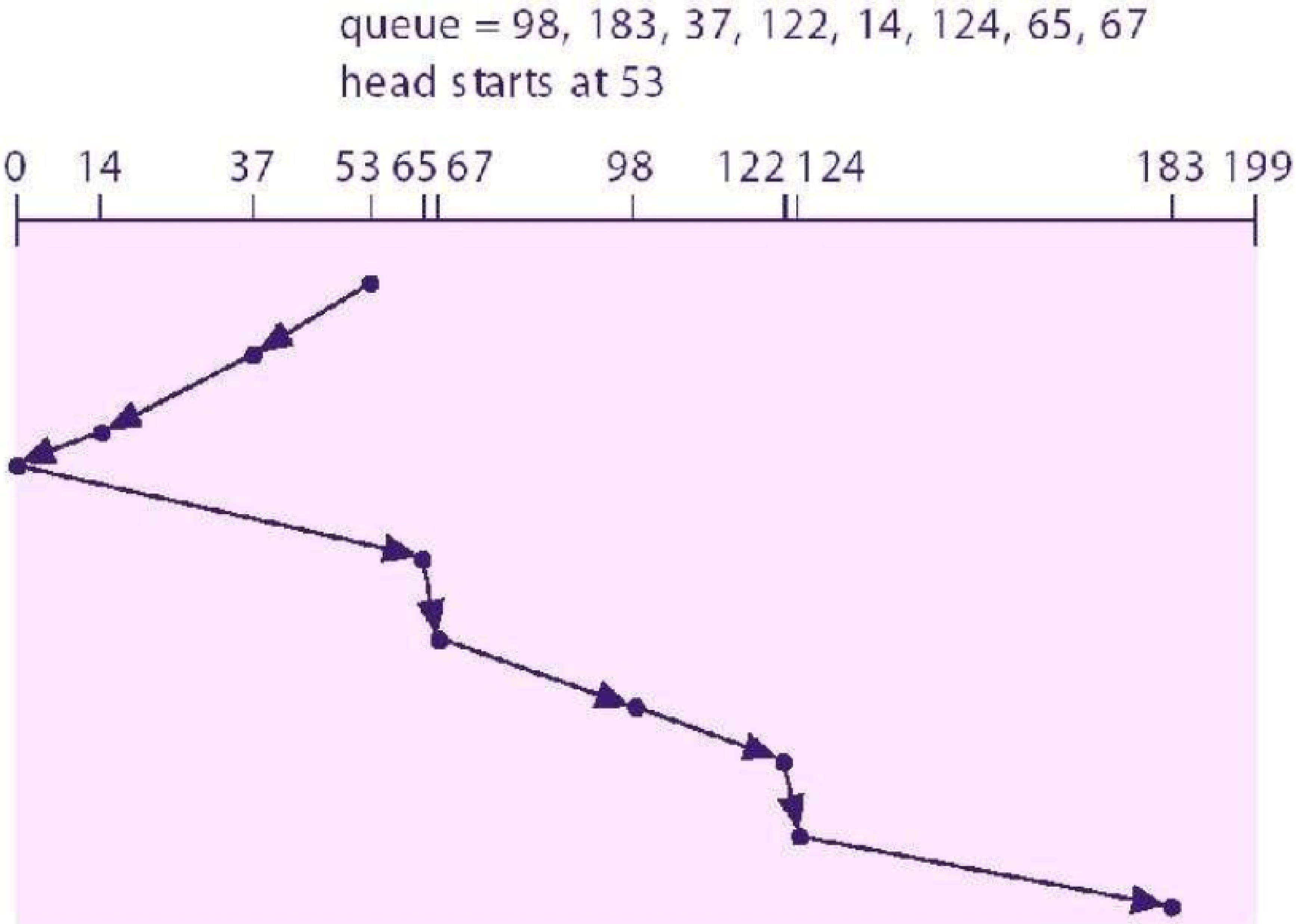
- Advantages:
  - Gives a substantial improvement in performance.

- Problems:
  - SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests.
  - Not optimal.

- *Used in batch system where throughput is the major consideration but unacceptable in interactive system.*

SCAN

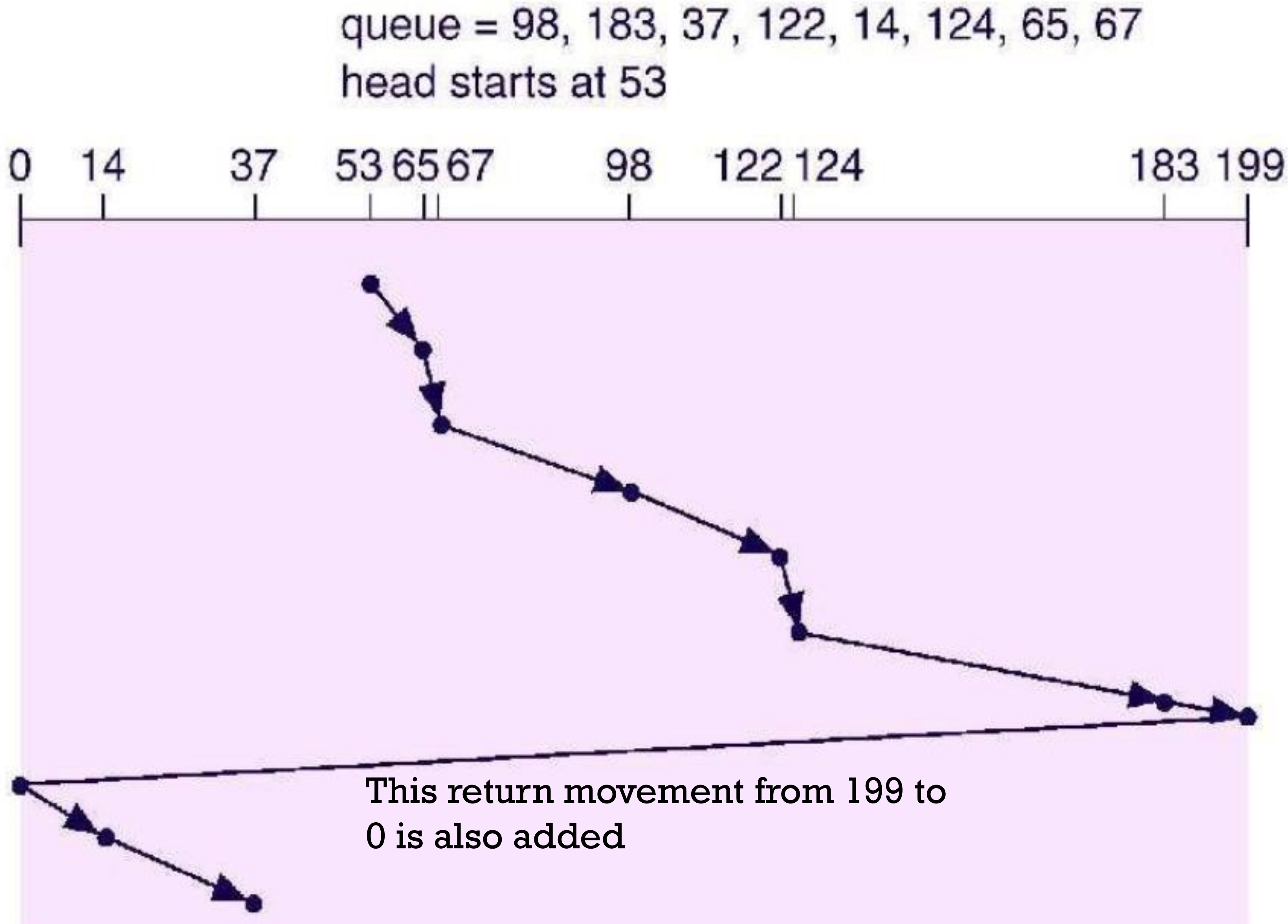
- *The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.*
- Sometimes called the ***elevator algorithm*.**
- Advantages:
  - Decreases variances in seek and improve response time.
- Problem:
  - Starvation is possible if there are repeated request in current track.



Total Head Movement = 208 cylinders

CIRCULAR SCAN (C-SCAN)

- *Circular SCAN is **a variant** of SCAN designed to **provide a** more **uniform wait time**.*
- The head moves from one end of the disk to the other. Servicing requests as it goes. When it reaches the other end, however, it immediately returns to the beginning of the disk, **without servicing any requests on the return trip**.
- Treats the cylinders as a **circular list** that wraps around from the last cylinder to the first one.

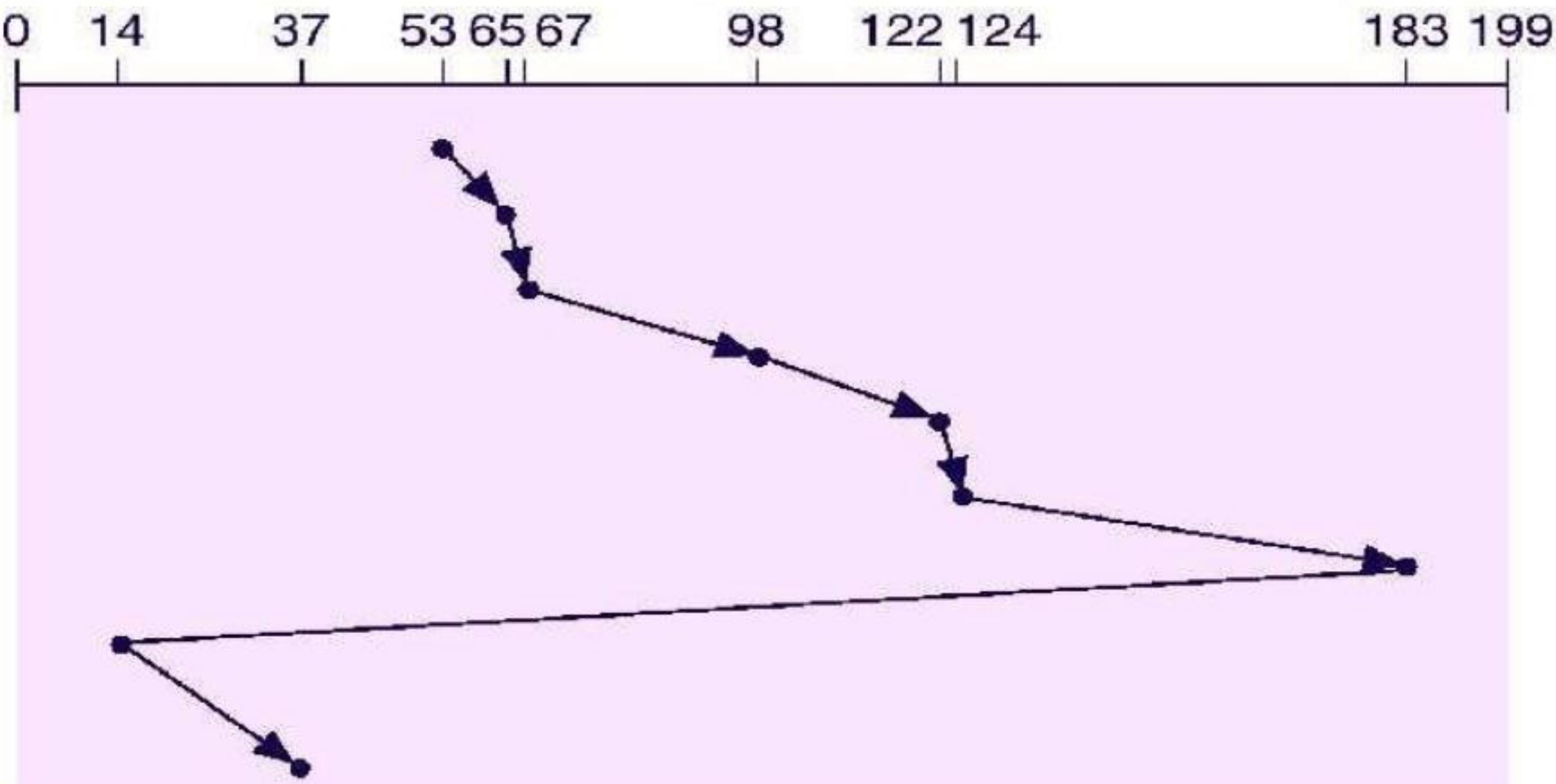

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199
This return movement from 199 to 0 is also added

C-LOOK

- Version of C-Scan
- Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk.

C-LOOK

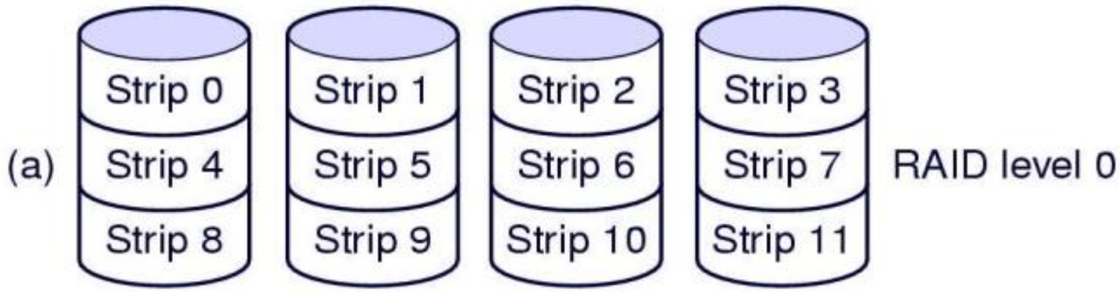
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53

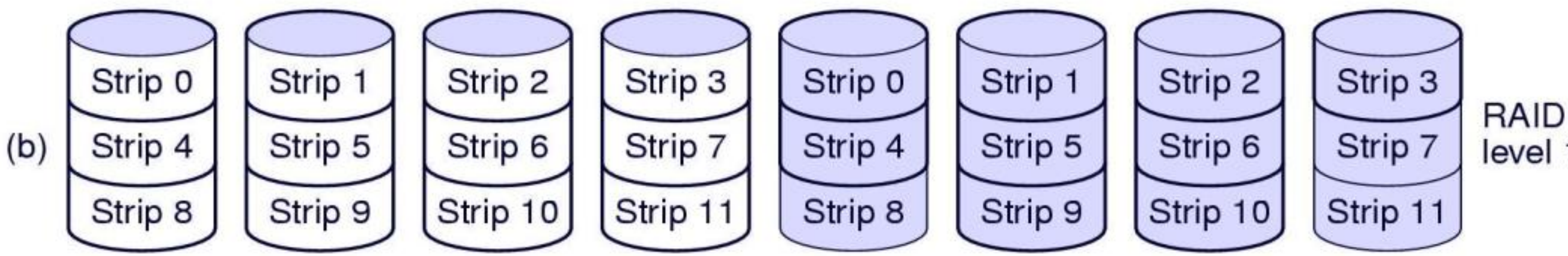


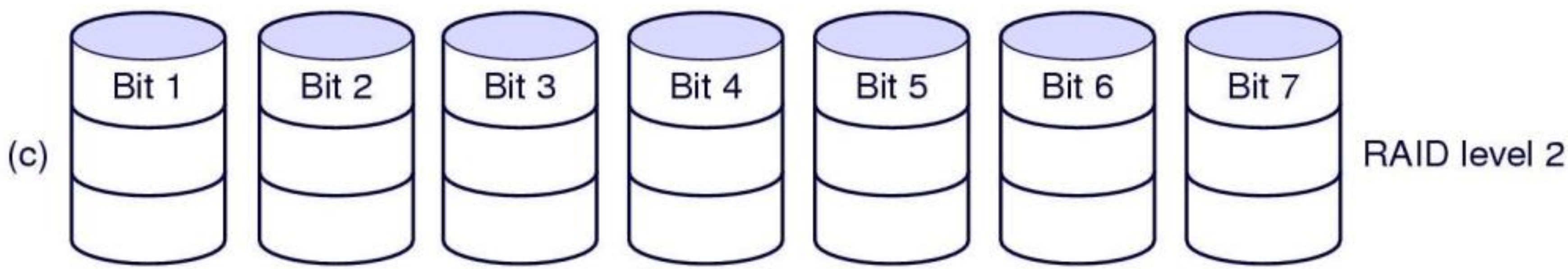
RAID

- ***Redundant Array of Independent(Originally Inexpensive) Disks.***
- **Issues:** Disk performance, Amount of storage required & Reliability
- *A technique of organizing multiple disks to* ***address above issues is RAID.***
- RAID allows more than one disk to be used for a given operation, and allows continued operation and even automatic recovery in the face of disk failure.
- Implemented in hardware or in OS.
- **Parallel processing** vs Raid
  - to speed up CPU performance.
  - Disk performance, Amount of storage required & Reliability

SIX TYPES OF RAID LEVELS







RAID ORGANIZATION LEVEL 0

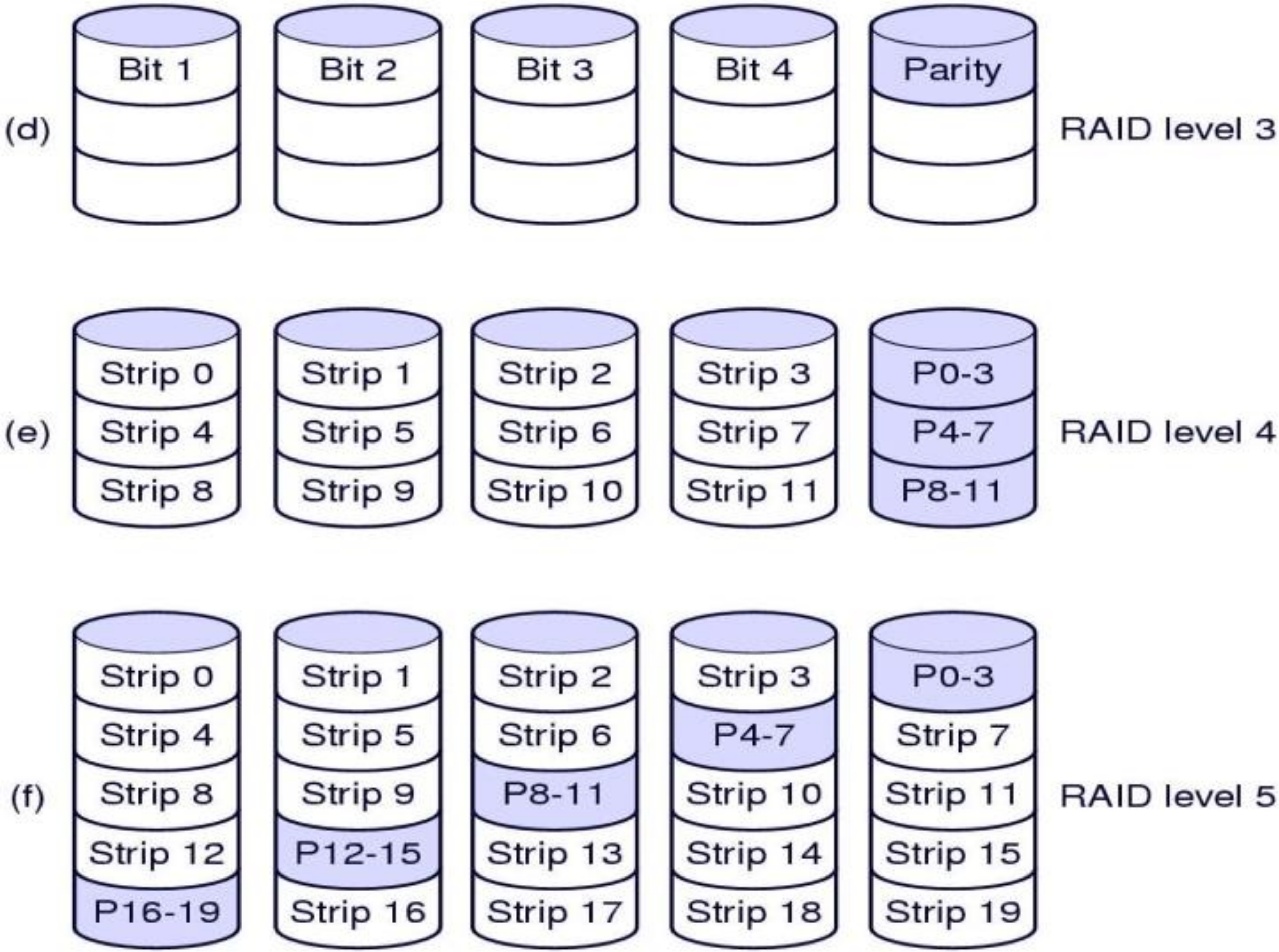
- *creates one large virtual disk from a number of smaller disks.*
- Storage is grouped into logical units called **strips,**
  - size of a strip = multiple of sector size
- The virtual storage is sequence of strips interleaved among the disks in the array.
- Worst with operating systems that **habitually ask for data one sector at a time**.
  - no parallelism and hence no performance gain.
  - best with large requests, as many Sectors at a time.
- **Advantages:**
  - Can create large disk;
  - Performance benefit can be achieved.
- **Disadvantages:**
  - Reliability decrease.

RAID ORGANIZATION LEVEL 1

- is a true RAID.
- It duplicates all the disks, so there are
  - four primary disks and
  - four backup disks.
- On a write, every strip is written twice. On a read, either copy can be used
- **Advantages:**
  - Excellent reliability; if drive crashes, the copy is used.
  - Read performance can be achieved .
- **Disadvantages:**
  - write performance is no better than for a single drive.

RAID ORGANIZATION LEVEL 2

- *An error-correcting code is used for corresponding bits on each data disks.*
- Error-correcting scheme store two or more extra bits, and can reconstruct the data if a single bit get damaged.
- For Example, the first bit of each byte is stored in disk 1, second bit in disk 2, and until eight bit in disk 8, and error correcting bits are stored in further disks. If one of the disk fail, the remaining bits of the byte and associated error-correction bits can be read from other disks and be used to reconstruct the damage data.
- Advantages:
  - Total parallelism.
- Disadvantages:
  - Requires substantial number of drives.


(d)
Bit 1
Bit 2
Bit 3
Bit 4
Parity
RAID level 3
(e)
Strip 0
Strip 1
Strip 2
Strip 3
P0-3
Strip 4
Strip 5
Strip 6
Strip 7
P4-7
Strip 8
Strip 9
Strip 10
Strip 11
P8-11
RAID level 4
(f)
Strip 0
Strip 1
Strip 2
Strip 3
P0-3
Strip 4
Strip 5
Strip 6
P4-7
Strip 7
Strip 8
Strip 9
P8-11
Strip 10
Strip 11
Strip 12
P12-15
Strip 13
Strip 14
Strip 15
P16-19
Strip 16
Strip 17
Strip 18
Strip 19
RAID level 5

RAID ORGANIZATION LEVEL 3

- A single parity bit is used instead of error-correcting code,
  - hence required just one extra disk.
  - If any disk in the array fails, its data can be determined from the data on the remaining disks by comparing with parity bit.
- *It is as good as Level 2 but is less expensive in the number of extra disks.*

RAID ORGANIZATION LEVEL 4

- *it uses block-level striping, as in Level 0, and in addition*
- *keeps a parity block on separate disk for corresponding blocks from other disks.*
- If one of the disks fails, the parity block can be used with the corresponding blocks from other disks to restore the blocks of the fail disks.
- *The transfer rate for large read as well as large write is high since reads and writes in parallel but small read and write can not be in parallel.*
- Problem: Parity bottleneck

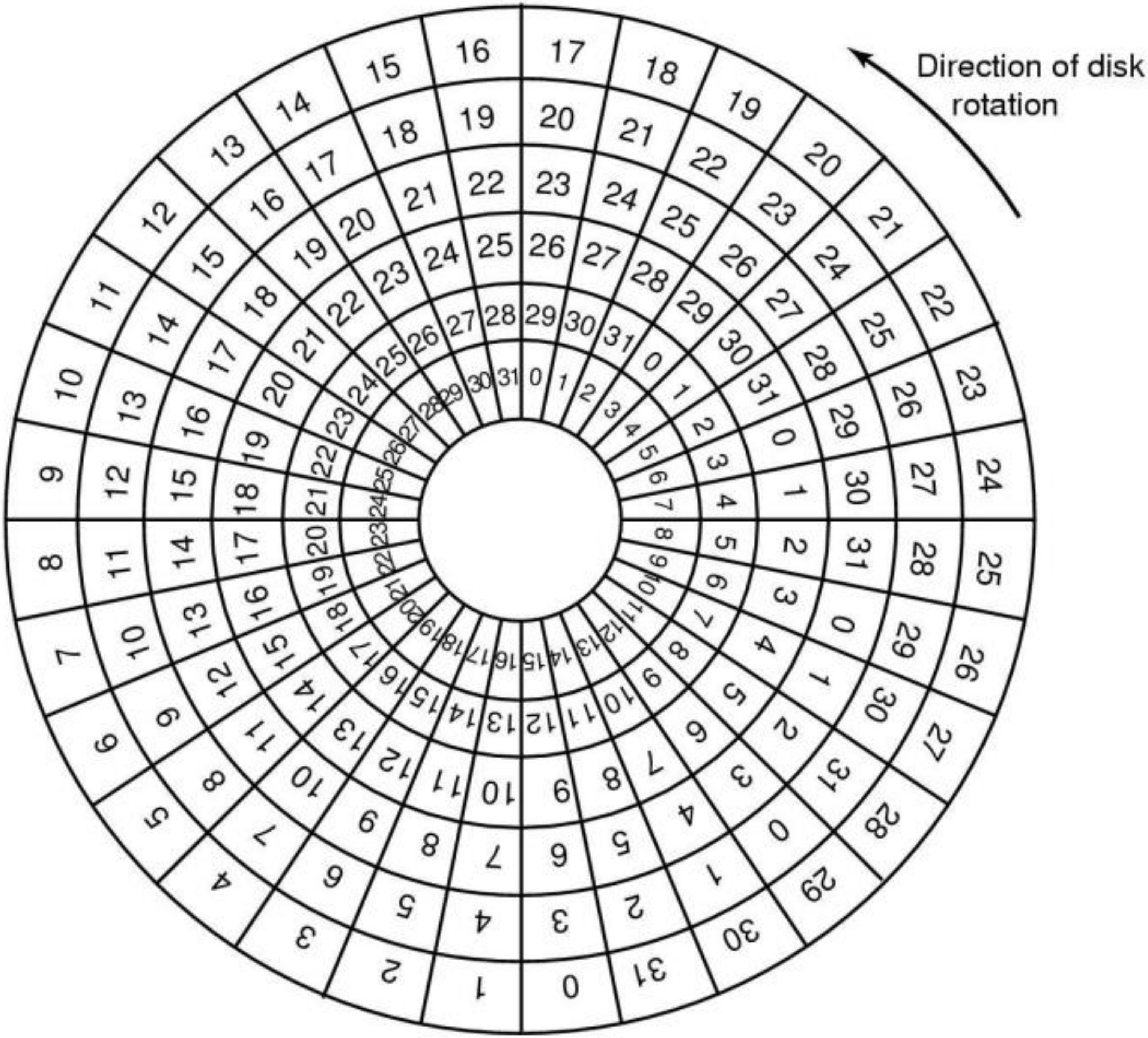
RAID ORGANIZATION LEVEL 5

- *Similar to level 4 but parity information is distributed in all disks.*
- For each block one of the disk stores parity and other stores data.

DISK FORMATTING

- *Before a disk can store data, it must be divided into sectors that the disk controller can read and write, called **low-level formating.***
- Sector typically consists of **preamble, data and ECC**.
- **Preamble contains**
  - the cylinder and sector number
- **ECC contains**
  - redundant information that can be used to recover from **read error (Error Correcting Code)**
  - **ECC** size depends upon the manufacturer, depending on reliability.

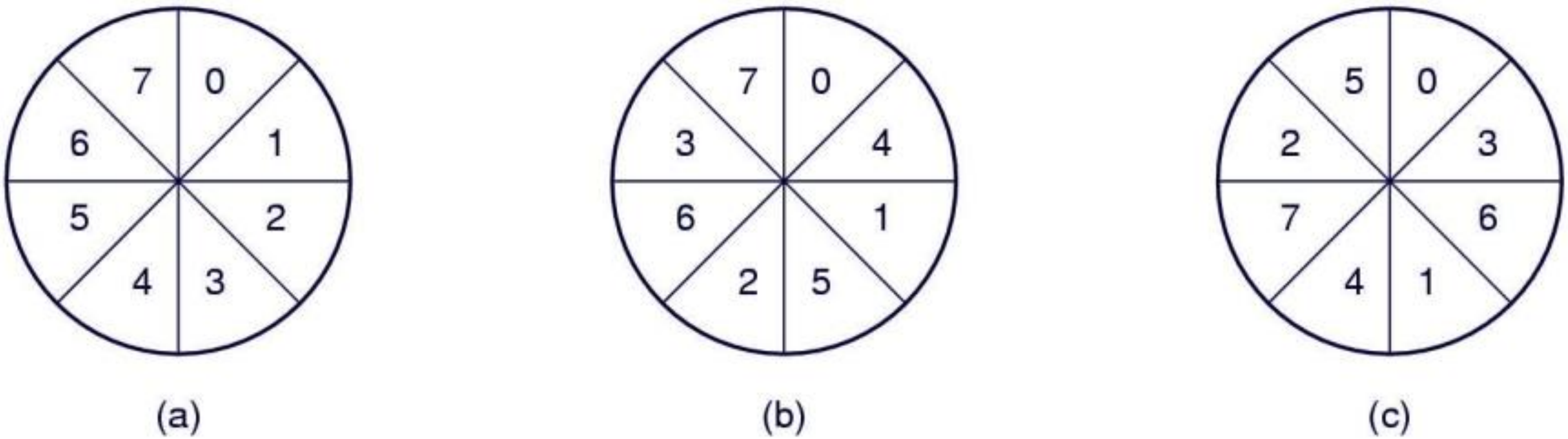
Preamble	Data	ECC


Direction of disk rotation

INTERLEAVING

- If disk I/O operations are limited to transferring a single sector at a time, it reads the first sector from the disk and **doing the ECC calculation, and transfers to main memory, during this time the next sector will fly by the head.**
- When transferring completes, the controller will have to wait almost an entire rotation for the second sector to come around again.
- *This problem can be eliminated by numbering the sectors in an interleaved fashion when formating the disk.*
- According to the copying rate, interleaving may be of **single or double.**

INTERLEAVING

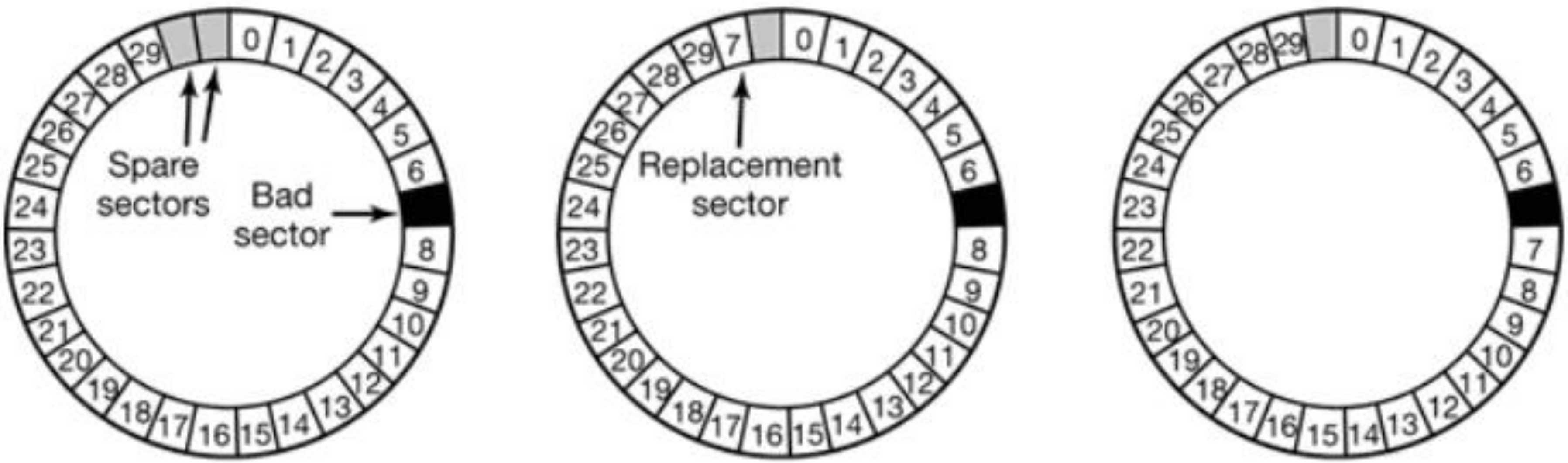


a) No interleaving b) Single interleaving c) Double interleaving

ERROR HANDLING

- Most frequently, one or more sectors becomes defective or most disks even come from factory with **bad blocks**.
- Depending on the disk and controller in use, these blocks handled in variety of ways
- **1. Bad blocks are handled manually**- For example run MSDOS chkdsk command to find bad block, and format command to create new block – data resided on bad blocks usually are lost.
- **2. Using bad block recovery-** The controller maintains the list of bad blocks on the disk and for each bad block, one of the spares is substituted.

BAD BLOCKS



(a) (b) (c)

- (a) A disk track with a bad sector. (b) Substituting a spare for the bad sector. (c) Shifting all the sectors to bypass the bad one.

RAM DISK

- ***RAM Disk** is virtual block device created from main memory.*
- Commands to read or write disks blocks are implemented by **RAM disk driver.**
- It completely **eliminates seek and rotational delays** suffered in disk devices.
- useful for storing files that are frequently accessed or temporary.
- especially used in high performance applications.
- Some OS define the RAM disks at boot time, other dynamically.
- ***Disadvantages**: cost and volatility.*
- The volatility is solved by providing battery backups.