Secondary memory technologies

- Electro-magnetic storage
  - Existing for over 50 years
  - Excellent performance for sequential work load
- Solid state storage
  - Upcoming & gaining popularity
  - Gives excellent performance for random read workload

Storage Devices

- Disk storage :
- Solid State storage
- Tape storage

IBM RAMAC 350 disk

- Developed in 1956
- Height 66 inches
- 50 platters of size 24 inches
- Weight : around 1 ton
- Storage capacity : 4MB

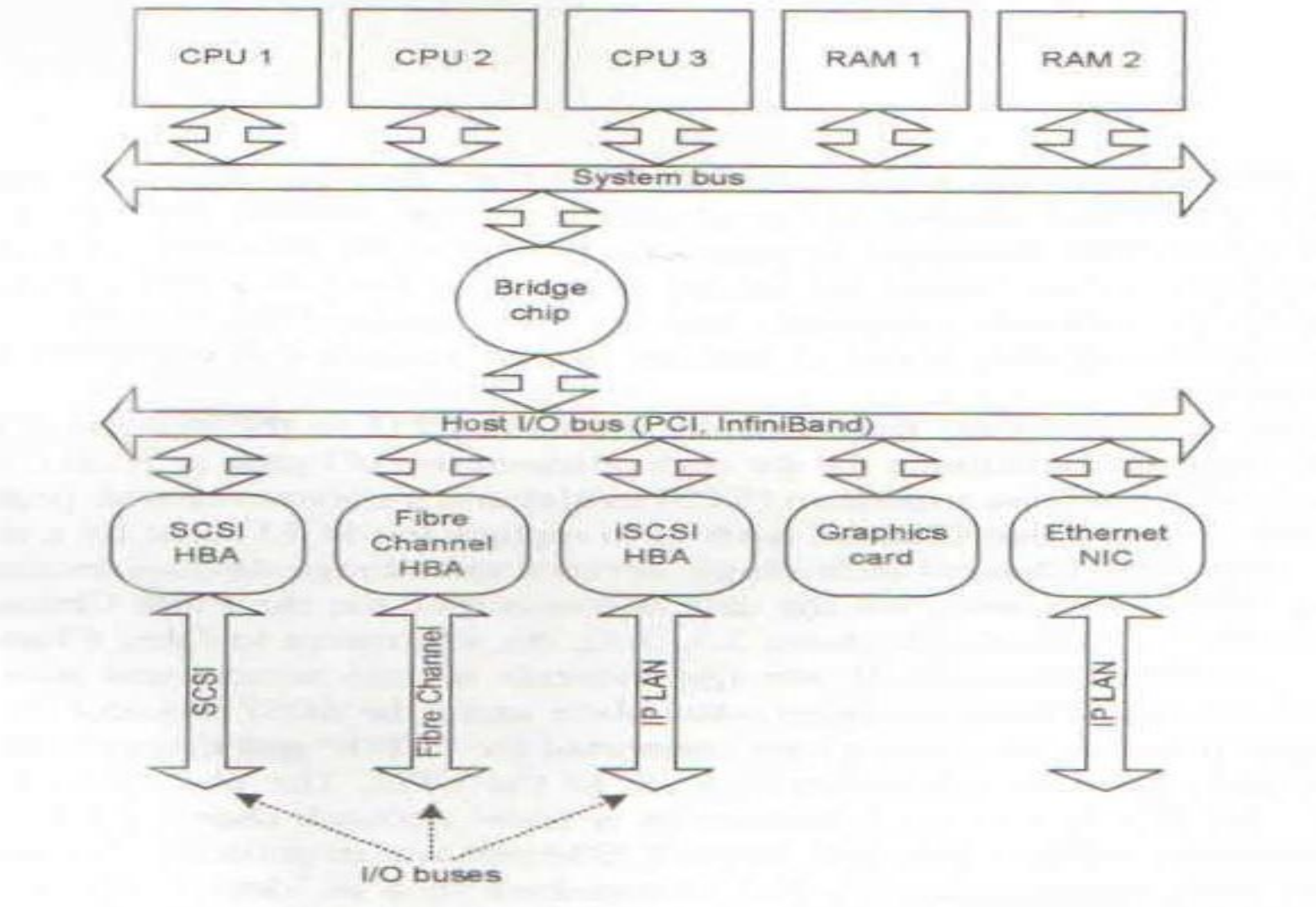
Modern Disk Drives

- Mechanism
  - Recording Components
    - Rotating Disk
    - Heads
  - Positioning Components
    - Arm Assembly
    - Track-following System
- Controller
  - Microprocessor
  - Buffer Memory
  - Interface to SCSI bus/SATA

Magnetic Tape

- Relatively permanent and holds large quantities of data
- 1000 times slower than disk
- Mainly used for backup, storage of infrequently-used data, transfer medium between systems
- 20GB-1.5TB typical storage
- Common technologies are 4mm, 8mm, 19mm, LTO-2 and SDLT
- the original version of which was released in 2000 and can hold 100 GB of data
- The seventh generation of LTO Ultrium was released in 2015 and can hold 6.0 TB

I/O path from CPU to storage

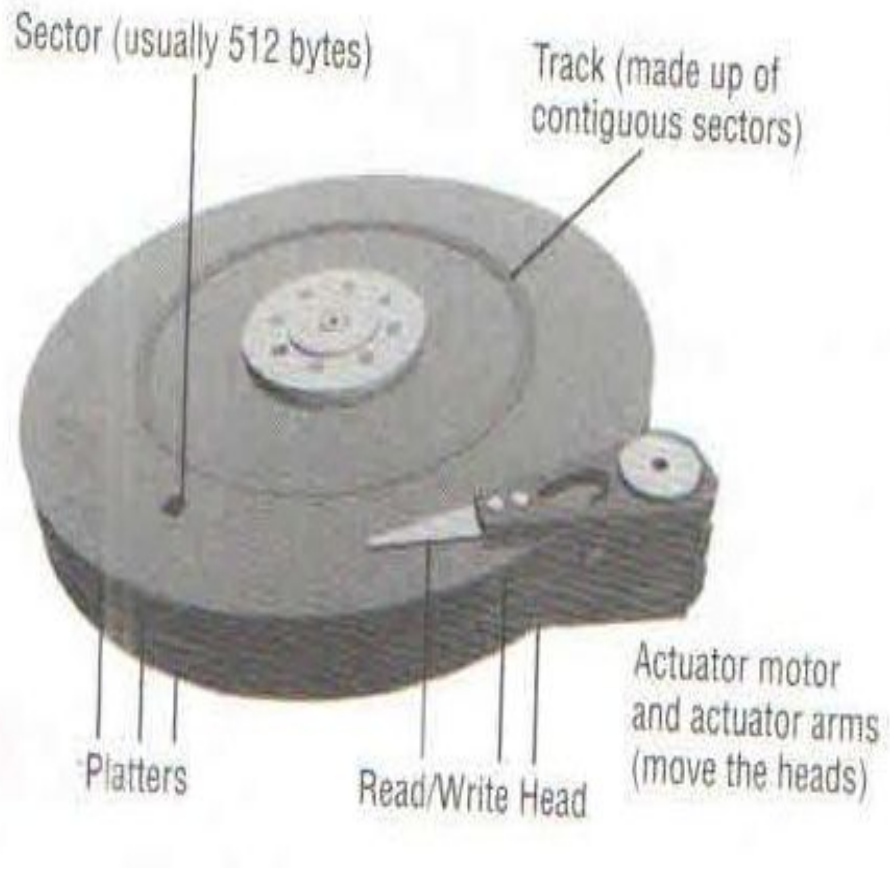


Disk Attachment

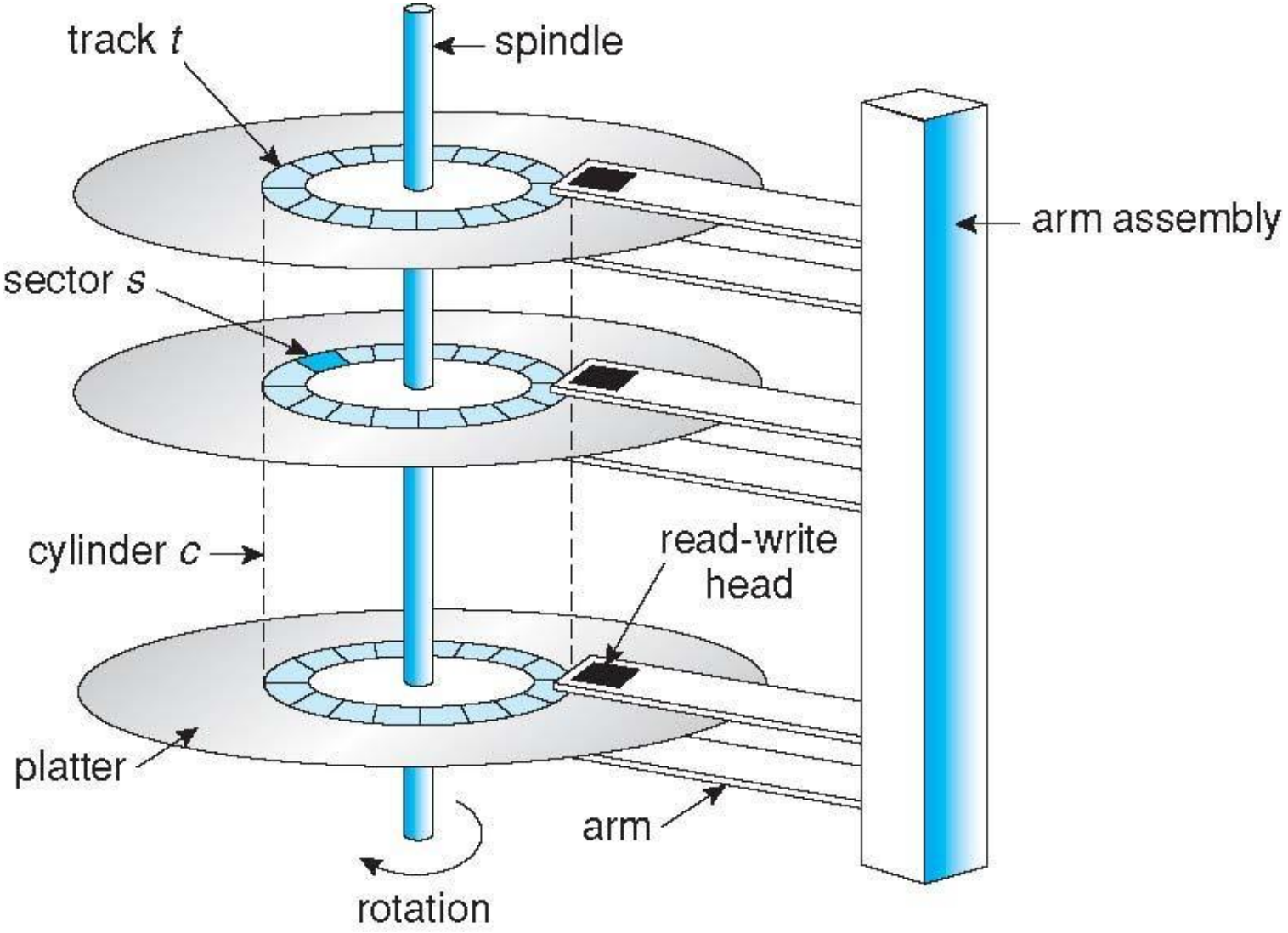
- Drives are attached to computer via **I/O bus**
  - USB
  - SATA (replacing ATA, PATA, EIDE)
  - SCSI
    - itself is a bus, up to 16 devices on one cable, **SCSI initiator** requests operation and **SCSI targets** perform tasks
- FC (Fiber Channel) is high-speed serial architecture

Anatomy of Disk

- Major components of disk drives
  - Platters
  - Read write head
  - Actuator assembly
  - Spindle motor

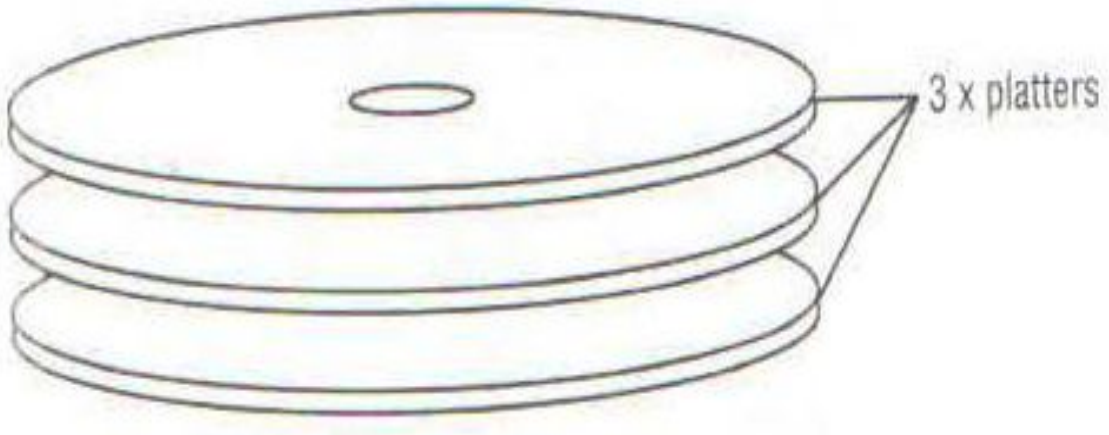


Moving-head Disk Mechanism



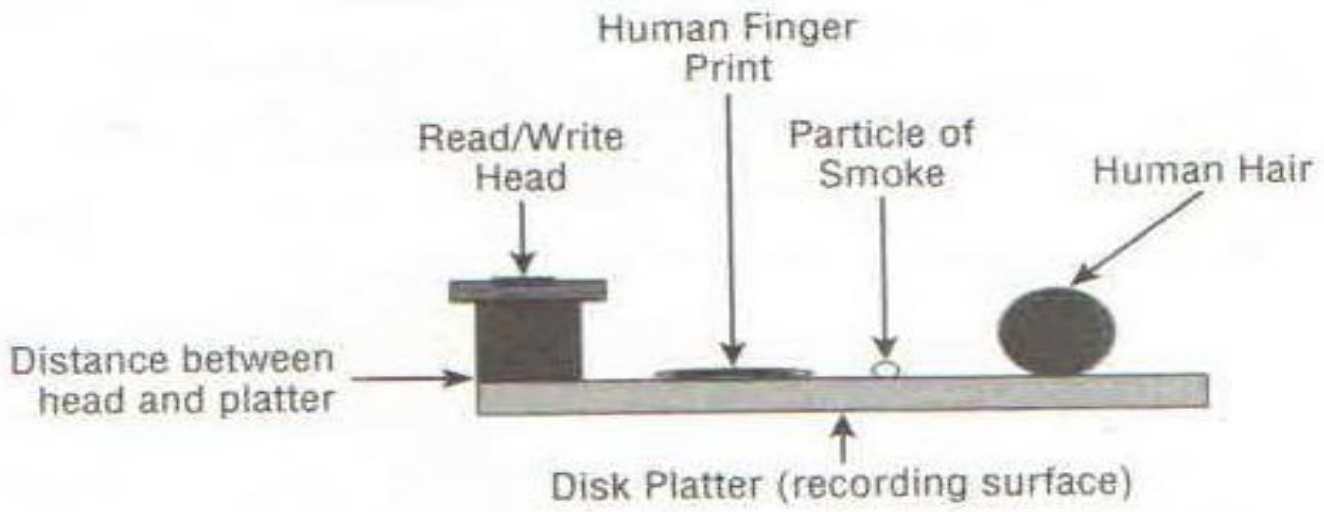
Platter

- Are made up of glass or aluminum substrate
- Is coated with magnetic material
- It is rigid, thin, flat , smooth
- All platters are attached to common shaft (spindle)
- Rigidity & smoothness is very important . Any defect could result in head crash



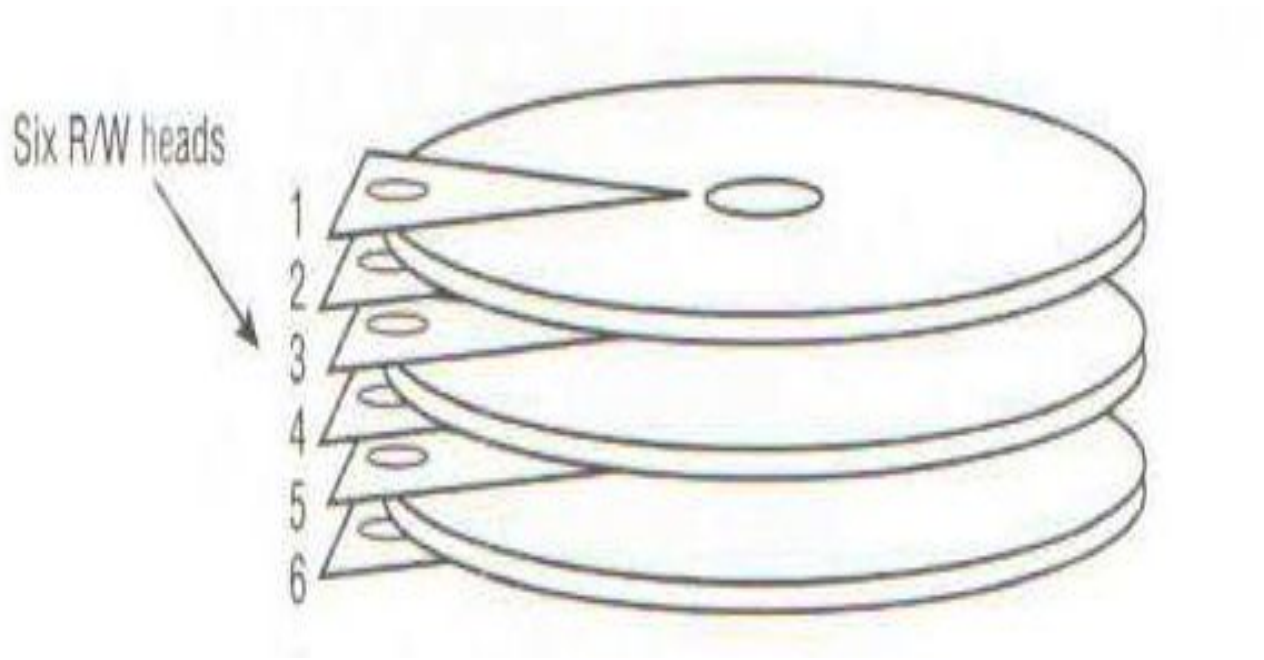
Read Write head

- Head flies above the platter surface
- Attached to actuator assembly
- OS does not know any thing about read write head
- Flying height is measured in micrometers



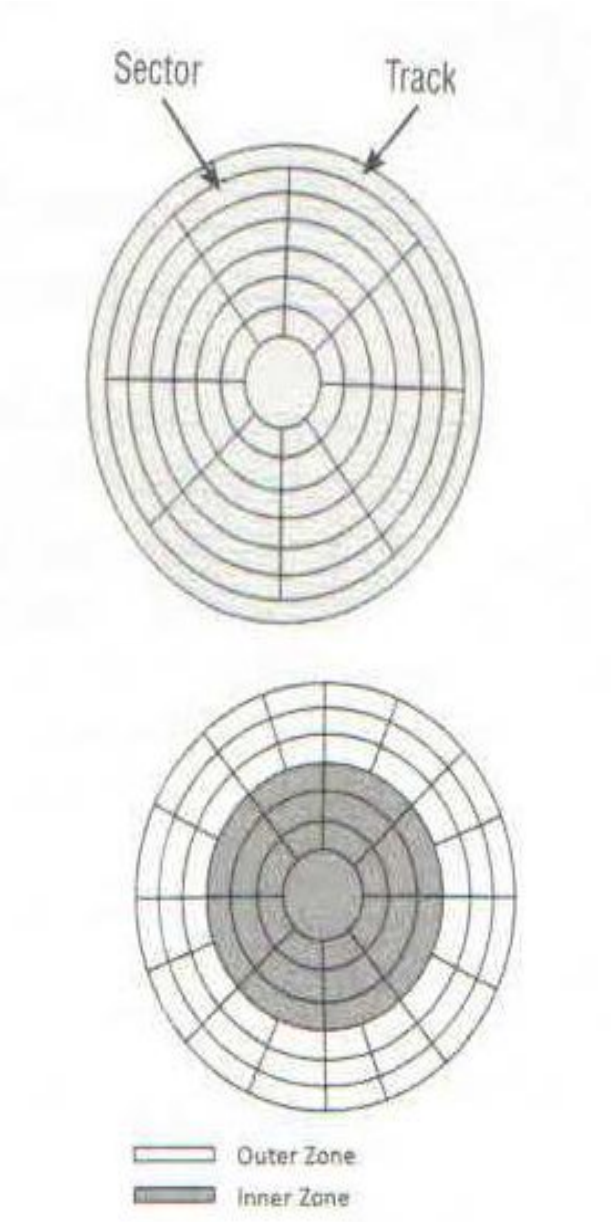
Head crash

- Read /write head never touch platter. If they touch , it is known as head crash.
- Head crash would always result in complete loss of data .
- Each platter surface has its own read/write head
- Concept of head and recording surface gave rise to CSH addressing



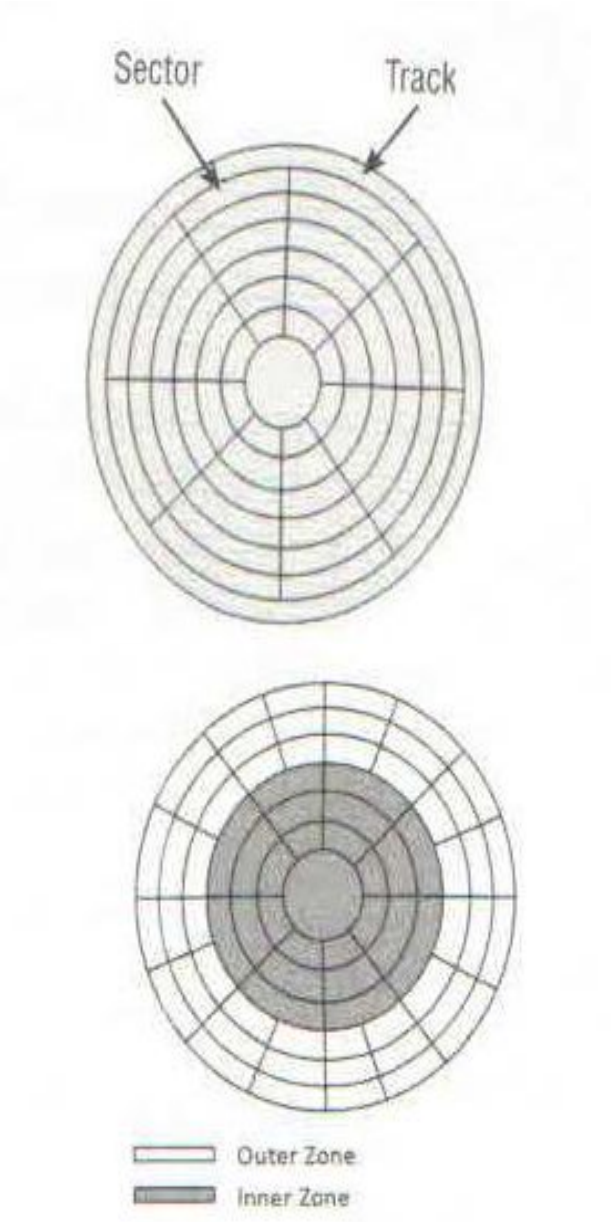
Tracks & Sectors

- Surface of every platter is microscopically divided into tracks & sectors
- Sector is the smallest addressable unit of a disk drive
- Sector size is typically 512 bytes or 520 bytes
- SATA (Serial Advance Technology Attachment) have fixed sector size of 512 bytes
- FC (Fiber channel) and serial attached SCSI (SAS) drives can be arbitrarily formatted to different size
  - This is important for implementation of data integrity technology for End to end data protection EDP
- 8 bytes of data added to end of every 512 byte sector
  - This allows errors to be detected



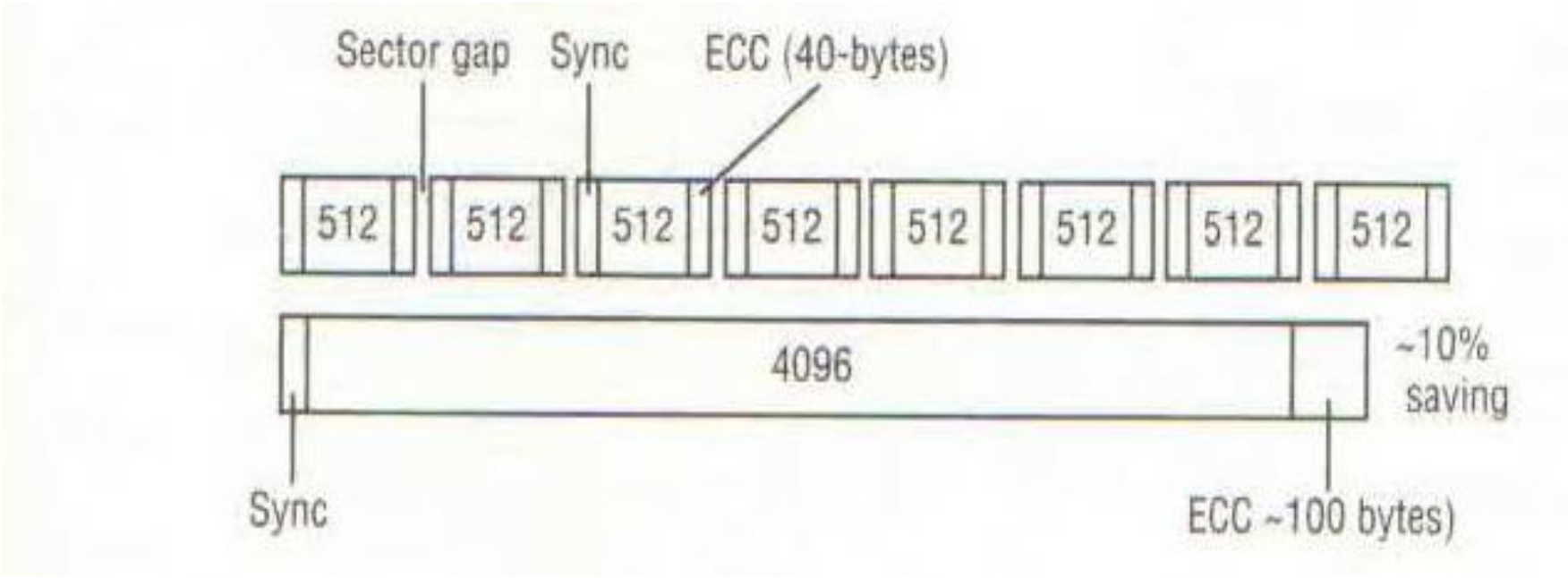
Tracks & Sectors

- Size of each sector is getting smaller and smaller towards the center .
- The recording density being same , outer sectors waste space
- Most modern disk implement Zoned Data Recording (ZDR)
- The outer track store and retrieve more data per spin
- If data is read or written to contiguous sector of same or adjacent track we get better performance
- Short stroking to achieve performance
  - Data could be stored only on outermost tracks
  - Reduced capacity , reduced load



Larger sector size trend

- 512 byte sector requires 320 bytes for ECC
- 4K byte requires only 100 byte
- Type of disk where size of sector is fixed and can not be changed is known as Fixed Block Architecture (FBA)



Logical block addressing

- OS treats disk as array of blocks
- Hides complexity of CSH based addressing
  - Is complex due to ZDR
  - Bad sector handling mechanism
- LBA is implemented in drive controller
- LBA to PBA mapping is required
- OS or volume managers never know the precise location of the data on disk

Controller

- Each disk drive comes with its own controller
- Controller has a processor, memory and it runs firmware
- Controller hides the disk operation complexities
- It maps Physical address to logical address and exposes LBA to OS/ Bios
- Controller also monitors the health of disk
- It also maintains the bad block map
  - Each drive has a number of hidden / spare block which are used to remap bad blocks

Common protocol & interfaces

- Serial advance technology attachment (SATA)
- Serial attached SCSI (SAS)
- Nearline SAS (NLSAS)
- Fibre channel (FC)

HDD Form factor

Form factor	Hight	Width	Depth
3.5 inch	1 inch	4 inch	5.75 inch
2.5 inch	0.6 inch	2.75 inch	3.94 inch

Drive speed

- 5400 RPM
- 7200 RPM
- 10000RPM
- 15000 RPM (240 KM / hr)
- Higher RPM better performance
- Higher RPM lower the capacity
  - 2.5 inch , 900Gb 10K
  - 3.5 inch , 4 TB 7500RPM

Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth.
- Access time has two major components
  - *Seek time* is the time for the disk arm to move the heads to the cylinder containing the desired sector.
  - *Rotational latency* is the additional time waiting for the disk to rotate the desired sector to the disk head.
- Minimize seek time
- Seek time ≈ seek distance
- Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.

Disk performance

- Seek time (15k 3.4 to 3.9ms) (7.2K 8.5 to 9.5 ms)
- Rotational latency 15k 2ms 7.2k 4.16ms
- Transfer time
- Access time = seek time+ rot delay + transfer time
- IOPS = 1 /(AST+ART) *1000
- eg. 1/(3.2+2)* 1000 = 181

Disk Scheduling (Cont.)

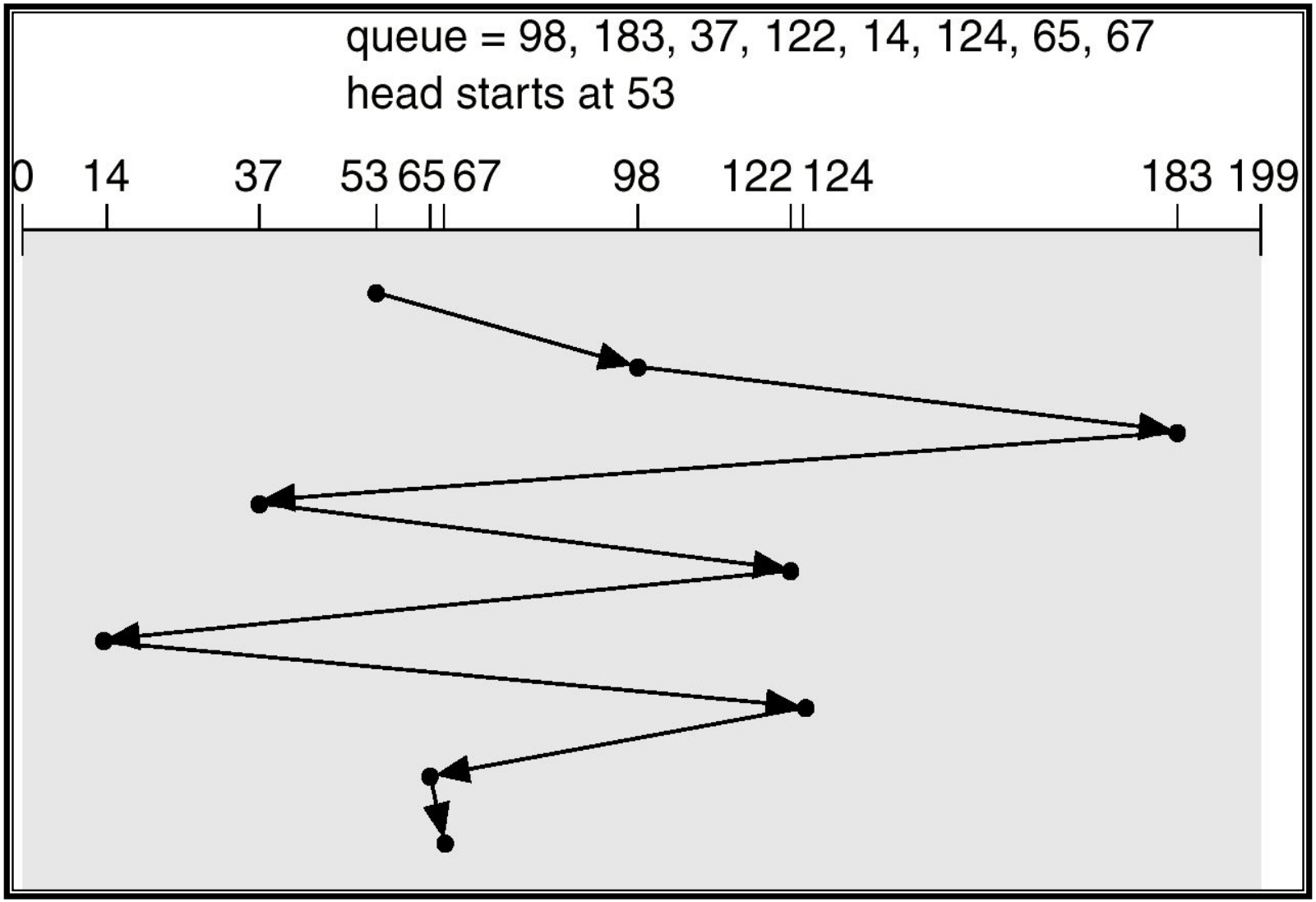
- Several algorithms exist to schedule the servicing of disk I/O requests.
- Example: request queue (0-199).

  98, 183, 37, 122, 14, 124, 65, 67

  Head pointer 53

FCFS

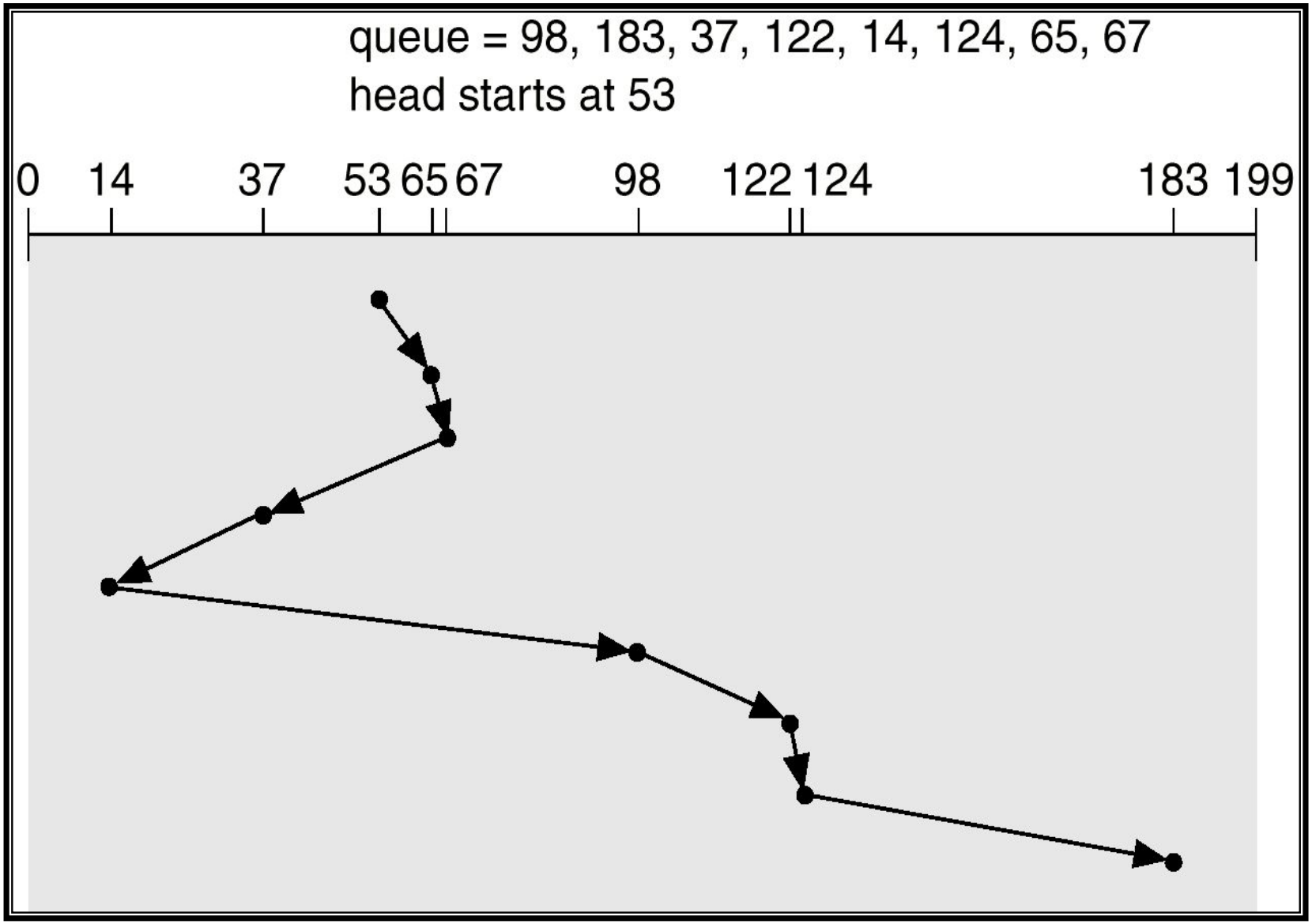
Illustration shows total head movement of 640 cylinders.



SSTF

- Selects the request with the minimum seek time from the current head position.
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests.
- Illustration shows total head movement of 236 cylinders.

SSTF (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

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*.
- Illustration shows total head movement of 208 cylinders.

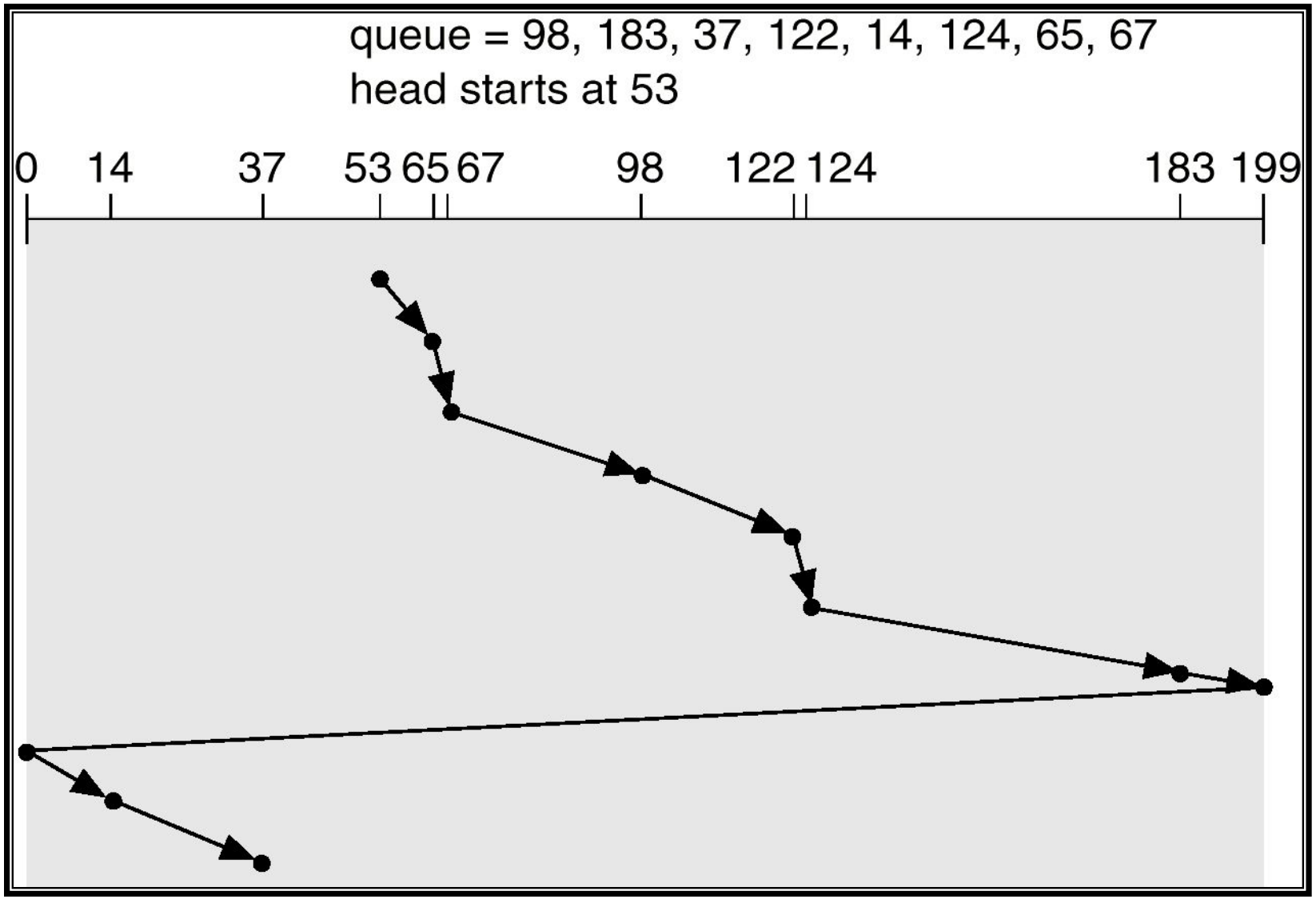
SCAN (Cont.)

0
14
37
53
65
67
98
122
124
183
199
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53

C-SCAN

- Provides a more uniform wait time than SCAN.
- The head moves from one end of the disk to the other servicing requests.
- When it reaches the other end, 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.

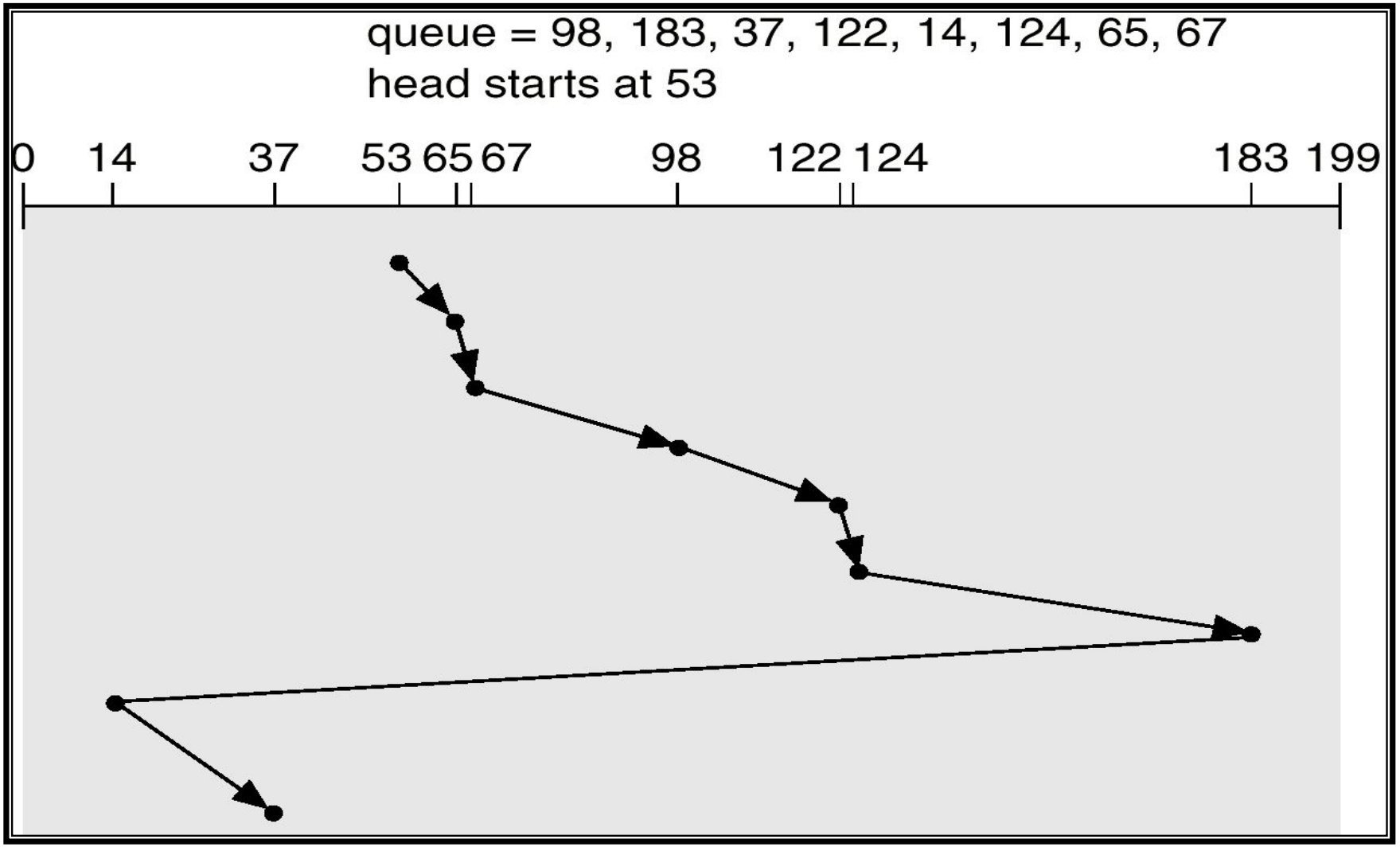
C-SCAN (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

C-LOOK

- Version of C-SCAN
- Arm only goes as far as the last request in each direction, then reverses direction immediately.

C-LOOK (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk.
- Performance depends on the number and types of requests.
- Requests for disk service can be influenced by the file-allocation method.
- Either SSTF or LOOK is a reasonable choice for the default algorithm.

RAID Structure

- **RAID** – multiple disk drives provides **reliability** via **redundancy**

- RAID is arranged into seven different levels.

Reliablity

- It is the measure of continuous service accomplishment Or time to fail
- MTTF ; mean time to failure (specified in no of hours0 is a measure of reliability
- Service interruption is measured as Mean time to repair (MTTR)
- Mean time between failure(MTBF)
- MtbF = MTTF+ MTTR
- Availability = MTTF / (MTTF + MTTR)

RAID (cont)

- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively.
- Disk striping uses a group of disks as one storage unit.
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data.

- RAID is a set of physical drives viewed by OS as a single Logical drive
- Data is distributed across the physical drives of a array
- Redundant disk capacity is used for parity for data recovery
- increase I/O handling capacity
- increase data handling capacity

RAID 0

- Does not include any redundancy
- Two different I/O requests can be issued in parallel
- The logical disk is divided into strips
- These strips may be physical blocks, sectors or some other unit
- Increases I/O handling capacity
- Increases data handling capacity

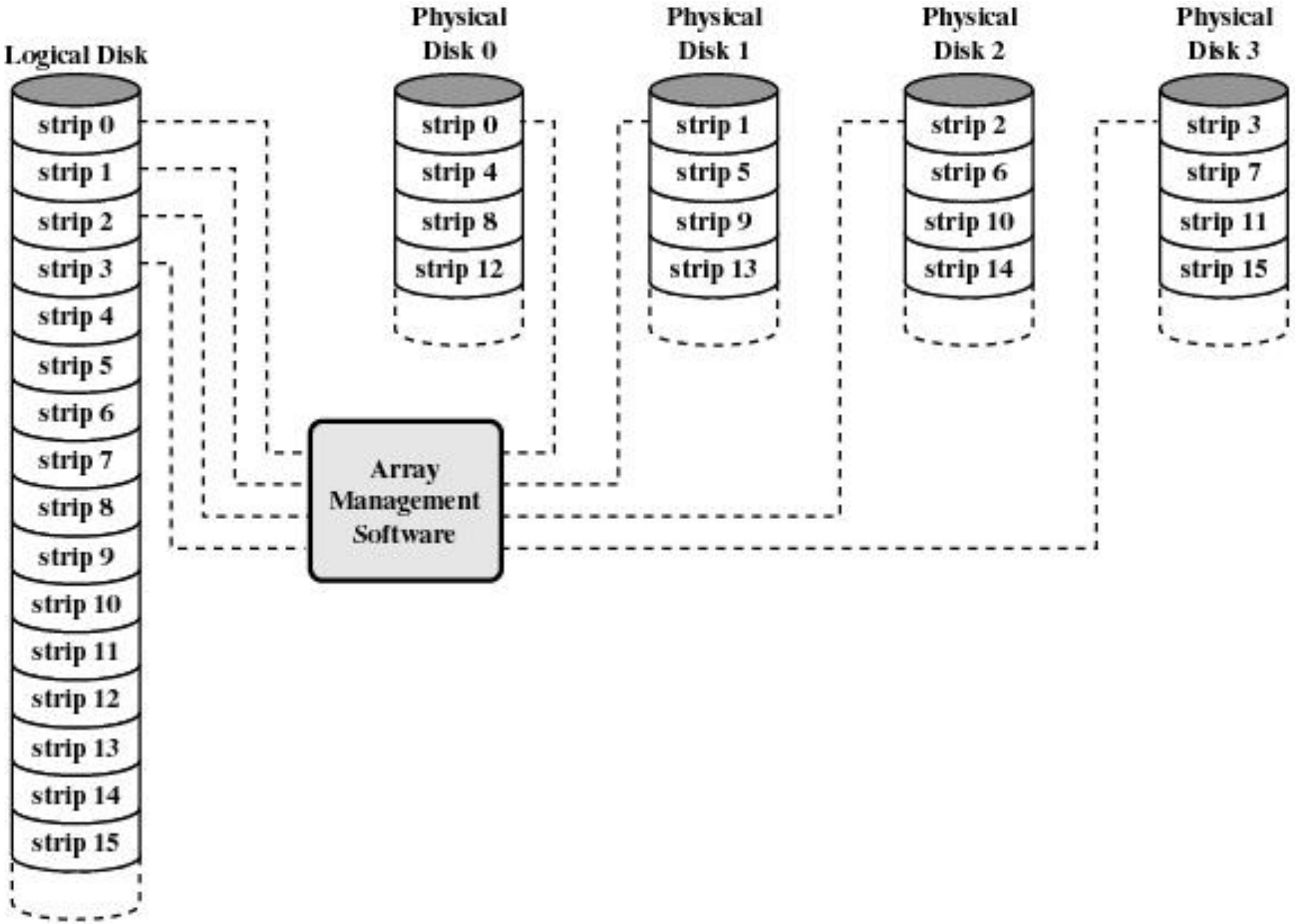
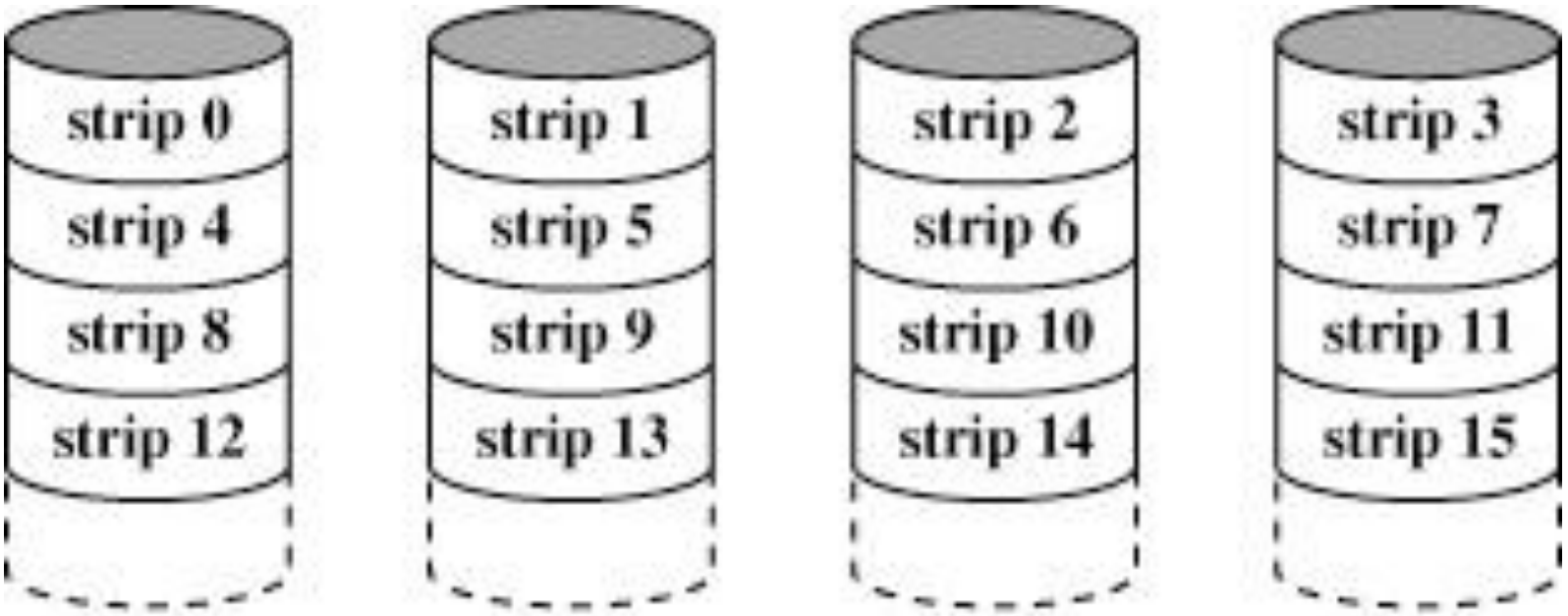


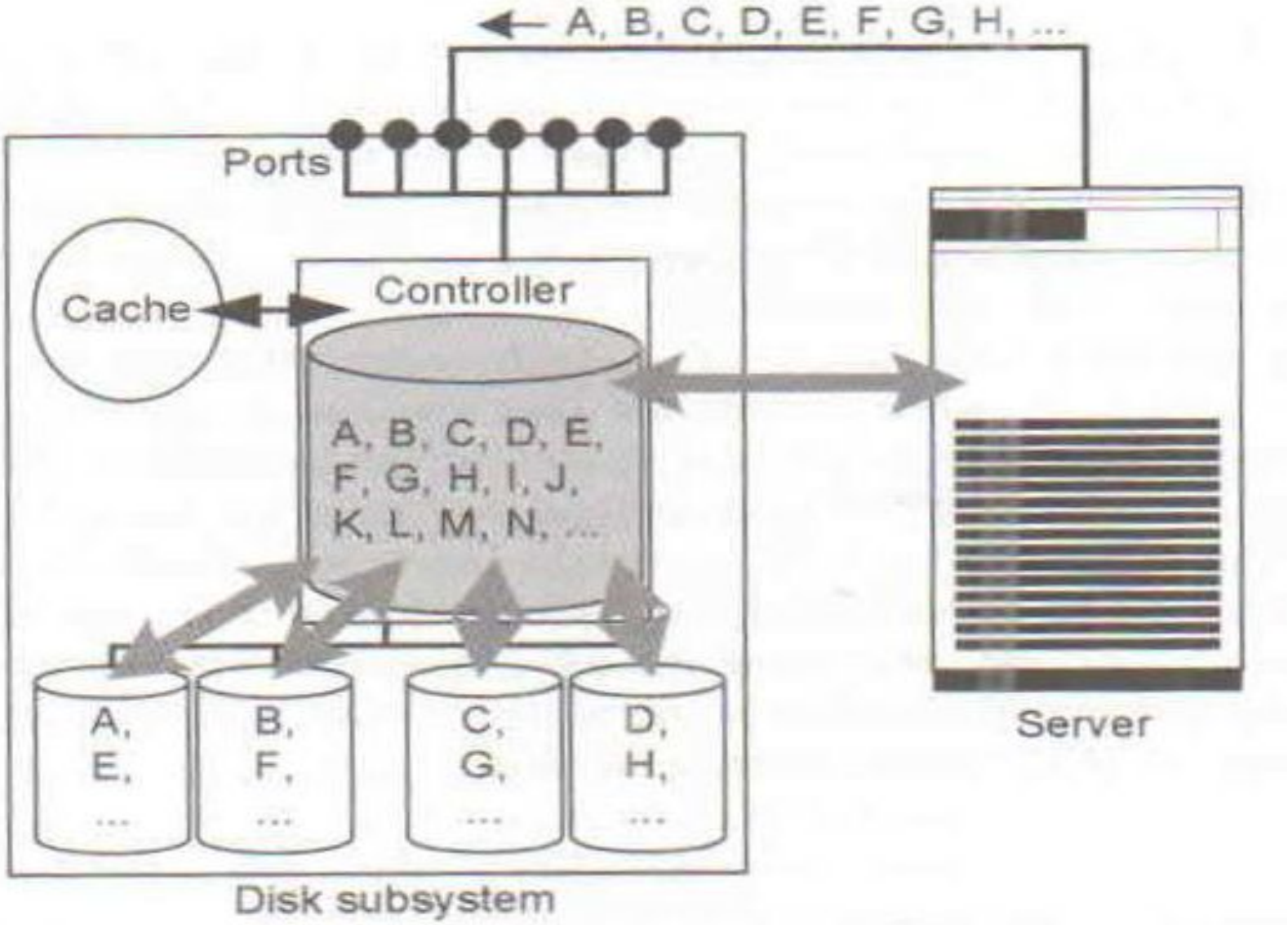
Figure 11.10 Data Mapping for a RAID Level 0 Array [MASS97]



(a) RAID 0 (non-redundant)

Figure 11.9 RAID Levels (page 1 of 2)

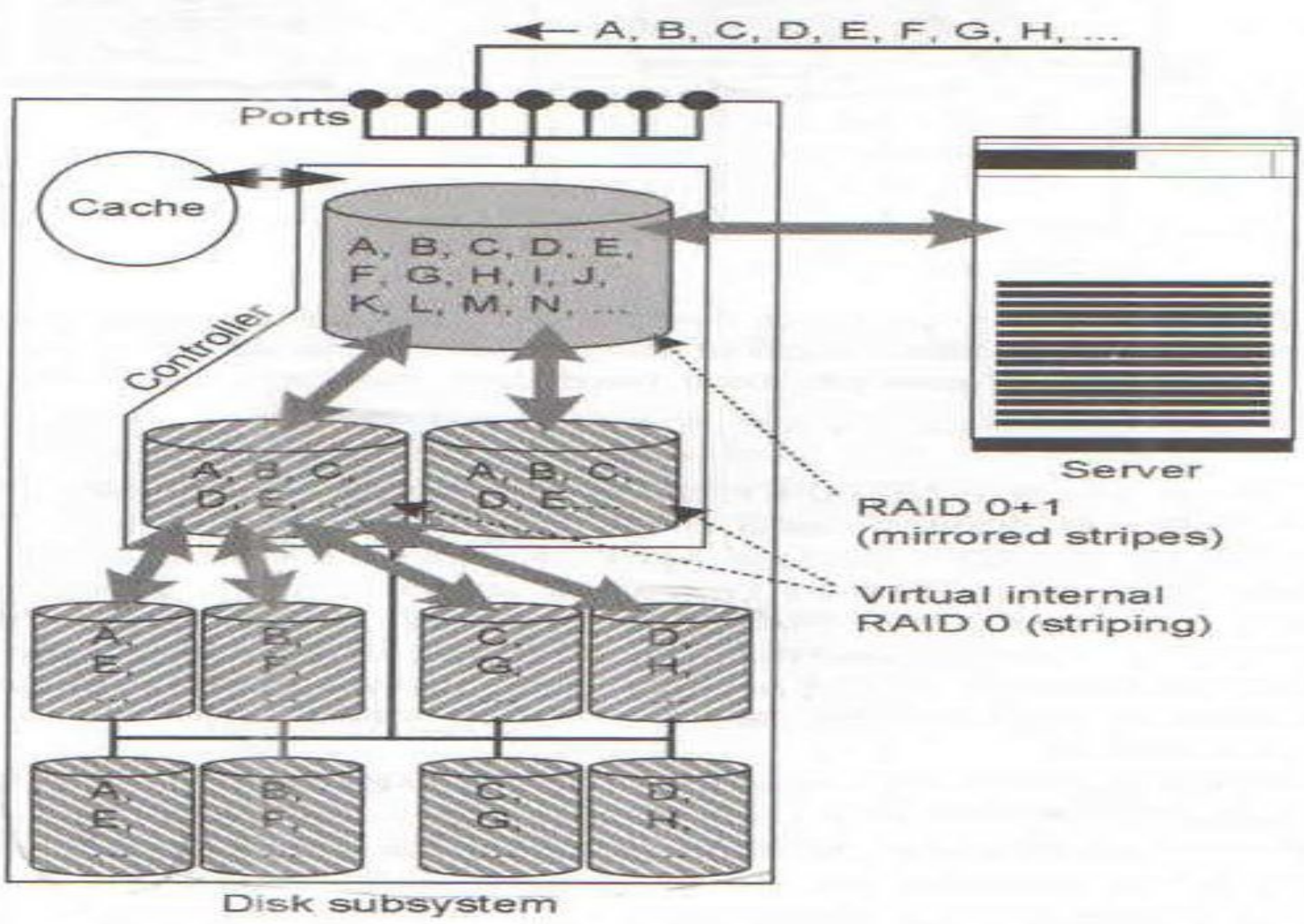
RAID 0

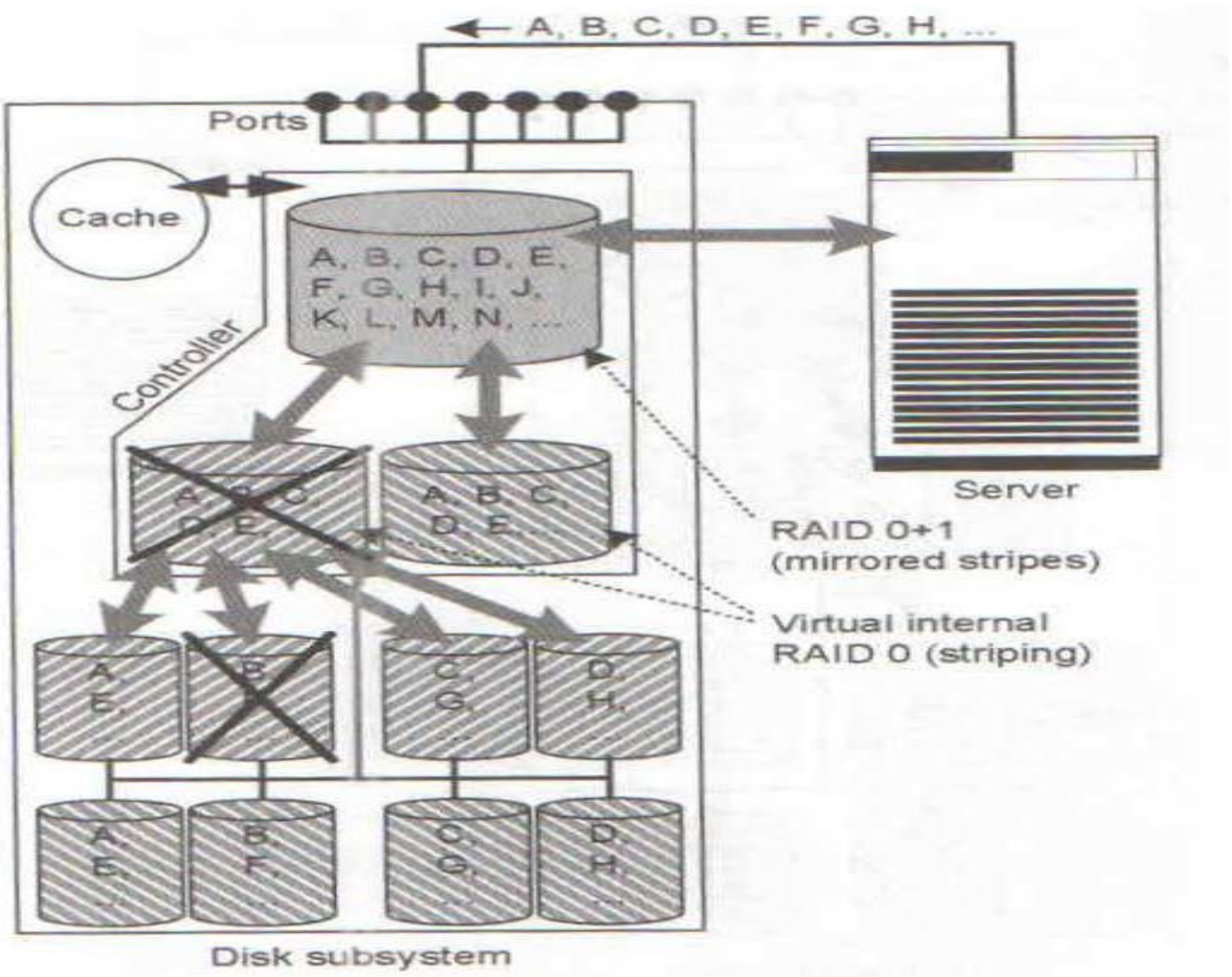

A, B, C, D, E, F, G, H, ...
Ports
Cache
Controller
A, B, C, D, E,
F, G, H, I, J,
K, L, M, N, ...
A,
E,
...
B,
F,
...
C,
G,
...
D,
H,
...
Server
Disk subsystem

RAID 1

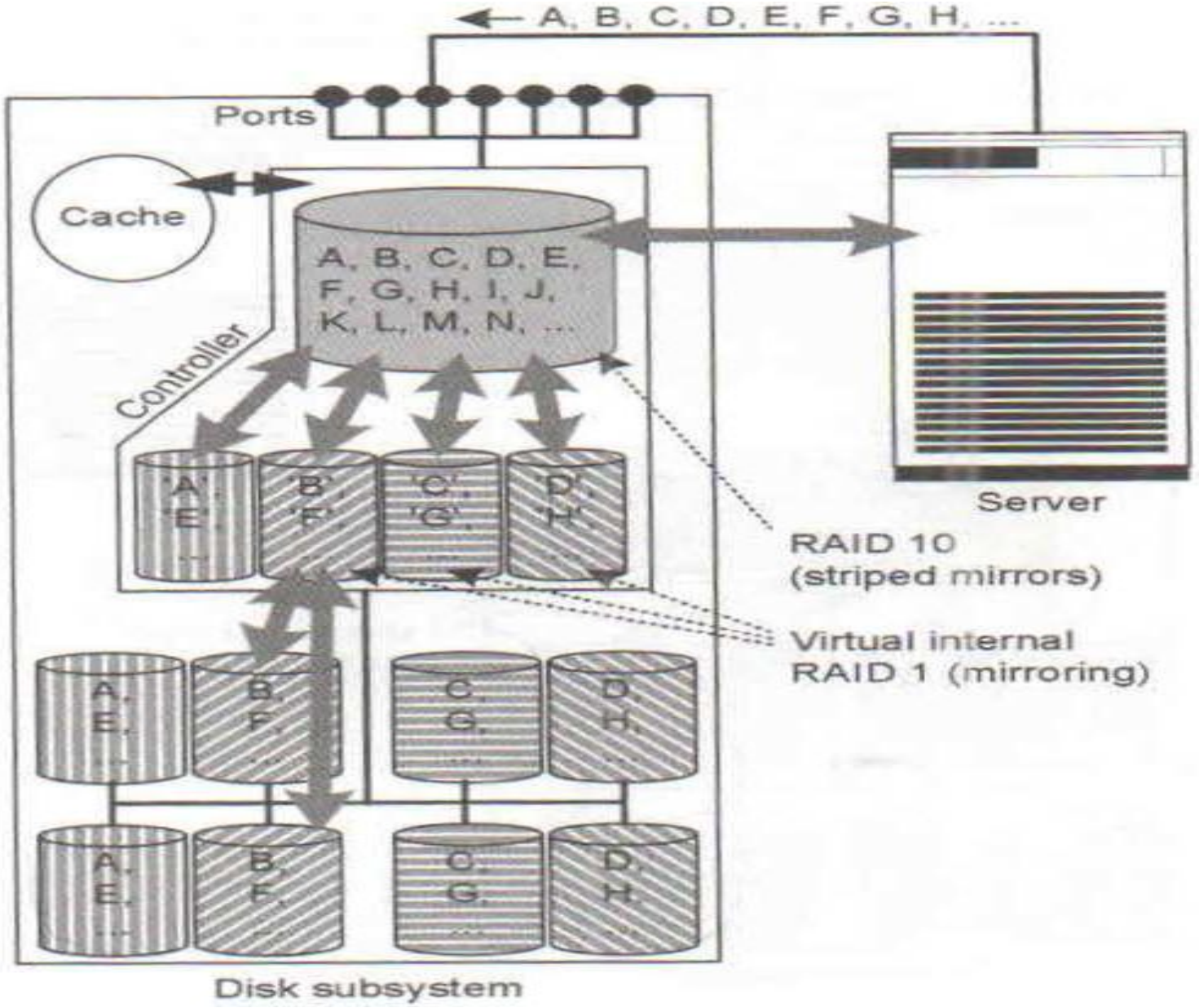
- Uses Mirroring
- A read request can be serviced by either of two disks whichever has lesser service time
- A write request requires that both corresponding strips be updated. This can be done in parallel.
- Recovery from failure is simple.
- High data transfer and I/O handling rate.

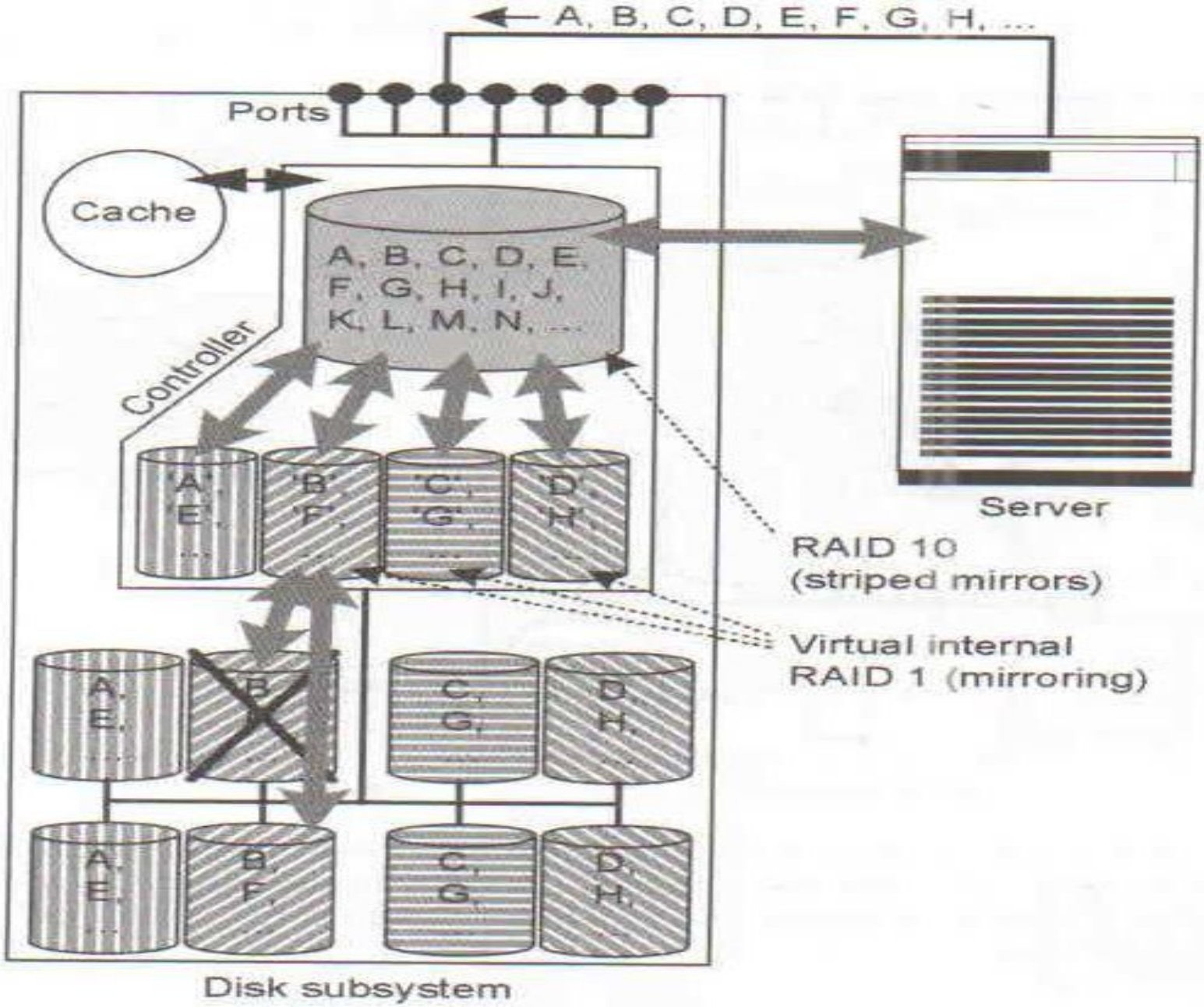
Composite RAID (0+1)




A, B, C, D, E, F, G, H, ...
Ports
Cache
A, B, C, D, E,
F, G, H, I, J,
K, L, M, N, ...
Controller
Server
RAID 0+1
(mirrored stripes)
Virtual internal
RAID 0 (striping)
Disk subsystem

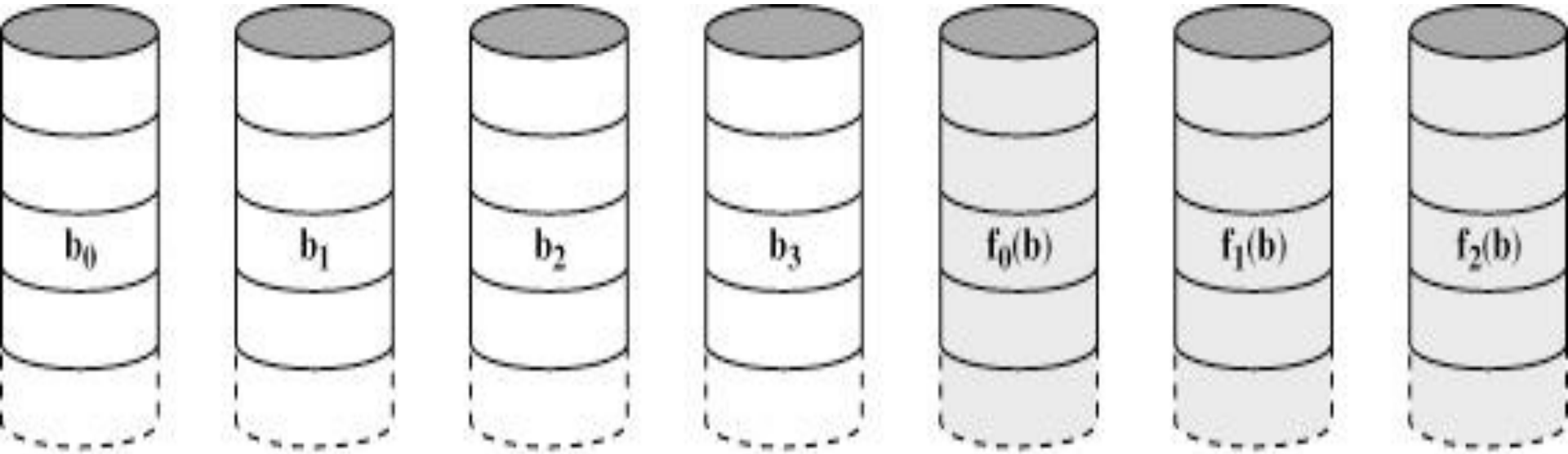
RAID 10


A, B, C, D, E, F, G, H, ...
Ports
Cache
A, B, C, D, E,
F, G, H, I, J,
K, L, M, N, ...
Controller
A'
E'
B'
F'
C'
G'
D'
H'
Server
RAID 10
(striped mirrors)
Virtual internal
RAID 1 (mirroring)
A
E
B
F
C
G
D
H
A
E
B
F
C
G
D
H
Disk subsystem


A, B, C, D, E, F, G, H, ...
Ports
Cache
A, B, C, D, E,
F, G, H, I, J,
K, L, M, N, ...
Controller
'A'
'E'
'B'
'F'
'C'
'G'
'D'
'H'
Server
RAID 10
(striped mirrors)
Virtual internal
RAID 1 (mirroring)
A
E
B
C
G
D
H
Disk subsystem

RAID 2

- RAID 2 and 3 make use of parallel accessing technique.
- All member disks participate in every I/O request
- The disk heads are synchronized
- In RAID 2 and 3 the strips are very small (byte or word)
- Multi bit parity is calculated using Hamming code
- Number of disks used is less than RAID 1 but still costly
- All disks are simultaneously accessed
- RAID 2 is not used

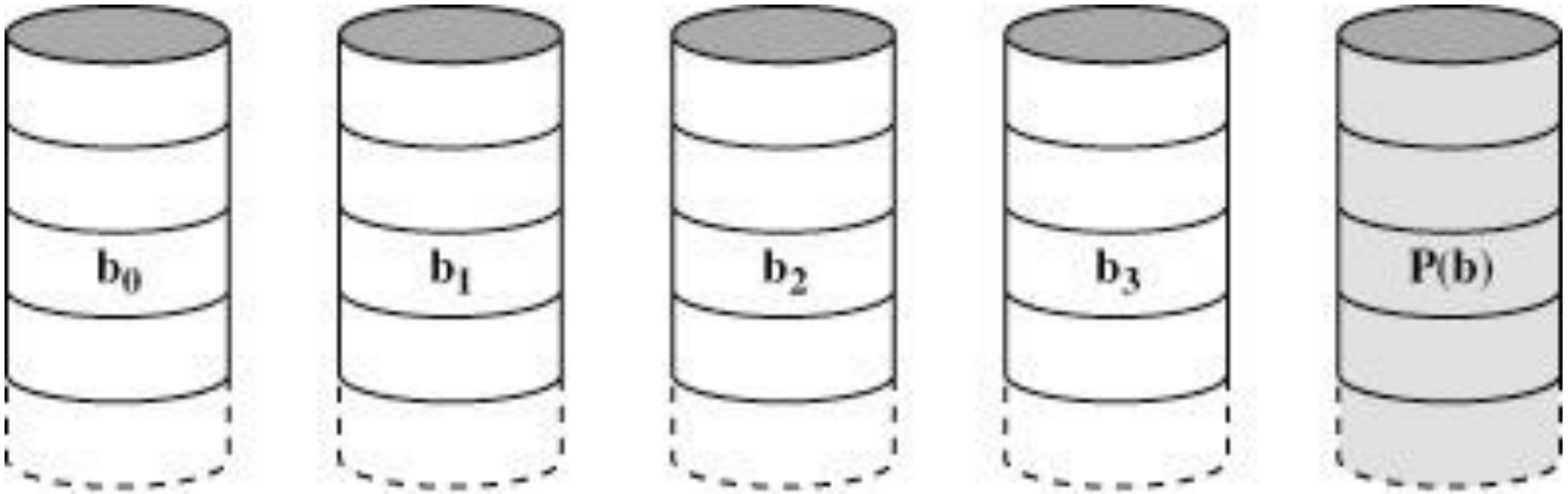


(c) RAID 2 (redundancy through Hamming code)

Figure 11.9 RAID Levels (page 1 of 2)

RAID 3

- Require only 1 redundant disk
- Simple parity bit is used
- High data transfer rate
- Only 1 I/O request can be handled at a time



(d) RAID 3 (bit-interleaved parity)

Figure 11.9 RAID Levels (page 2 of 2)

$X4(i) = X3(i) \oplus X2(i) \oplus X1(i) \oplus X0(i)$

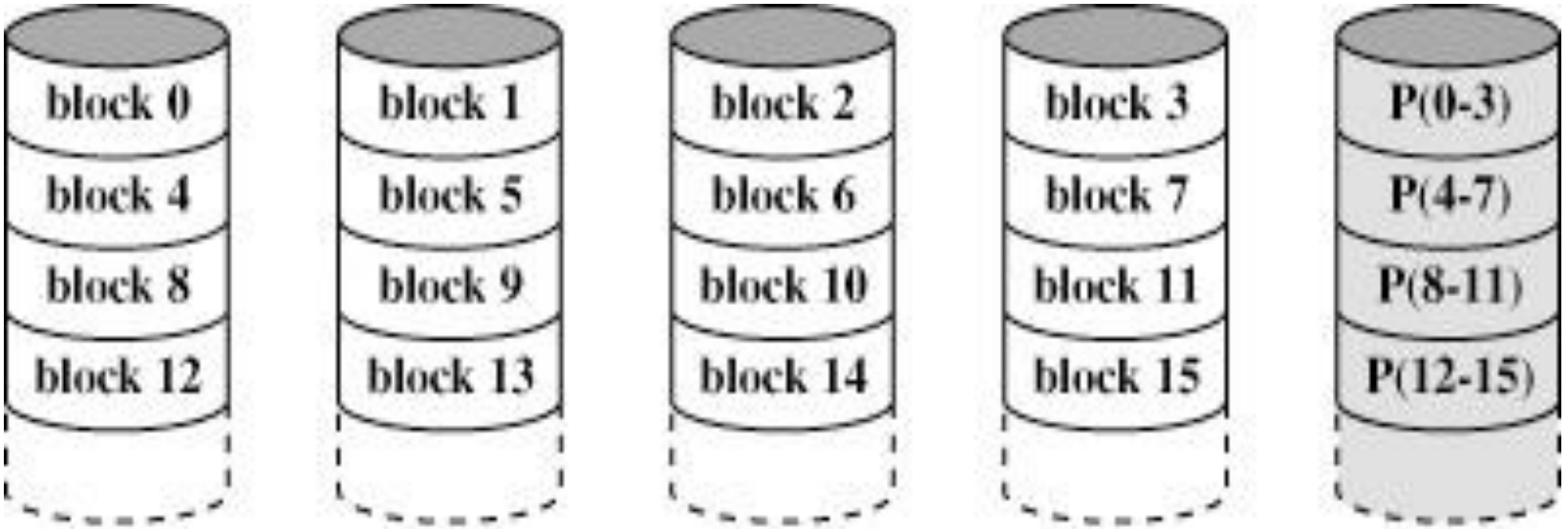
Suppose Disk X1 fails

By adding $X4(i) \oplus X1(i)$ on both sides we get

$X1(i) = X4(i) \oplus X3(i) \oplus X2(i) \oplus X0(i)$

RAID 4

- Each member disk operates independently
- Can handle high I/O request rates
- Each WRITE requires two READS and 2 WRITES, which is a bottleneck



(e) RAID 4 (block-level parity)

Figure 11.9 RAID Levels (page 2 of 2)

$X4(i) = X3(i) \oplus X2(i) \oplus X1(i) \oplus X0(i)$

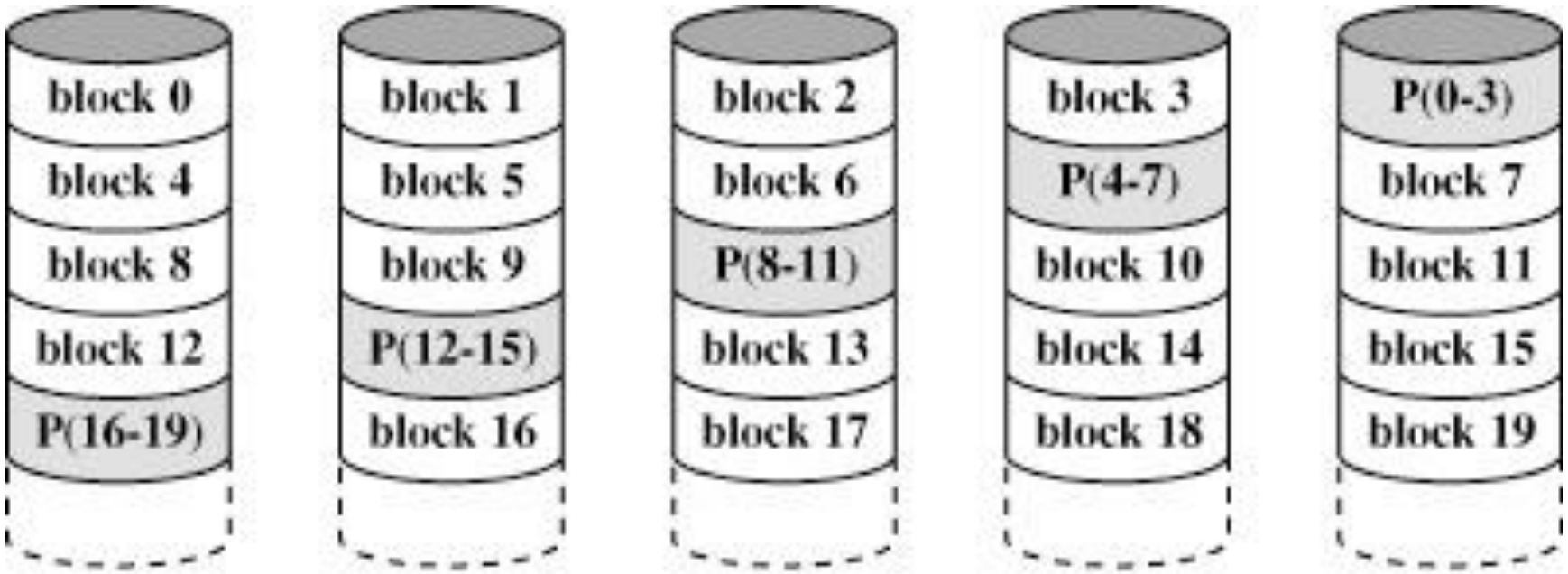
Suppose a write is performed which only involves a strip on disk X1.

Thus

$$X4'(i) = X3(i) \oplus X2(i) \oplus X1'(i) \oplus X0(i)$$
$$= X3(i) \oplus X2(i) \oplus X1'(i) \oplus X0(i) \oplus X1(i) \oplus X1(i)$$
$$= X4(i) \oplus X1(i) \oplus X1'(i)$$

RAID 5

- Data distribution similar to RAID 4
- Parity is Distributed on all disk
- I/O bottleneck of single parity disk is avoided

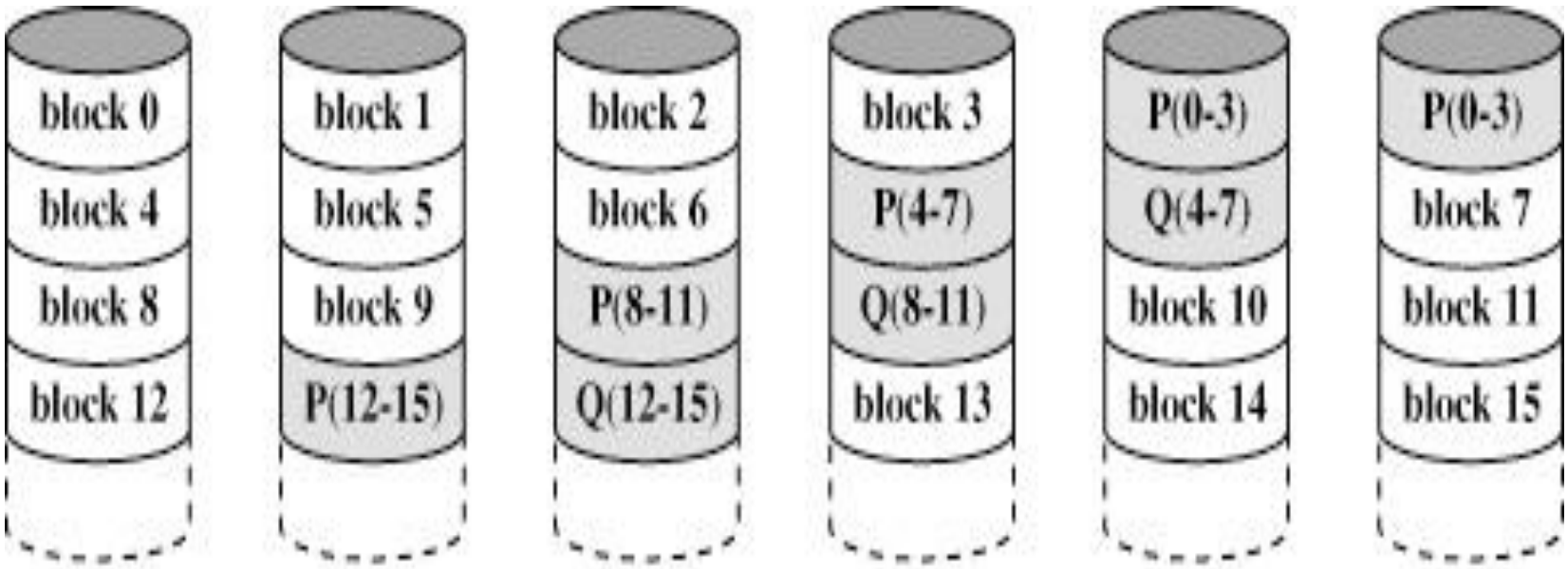


(f) RAID 5 (block-level distributed parity)

Figure 11.9 RAID Levels (page 2 of 2)

RAID 6

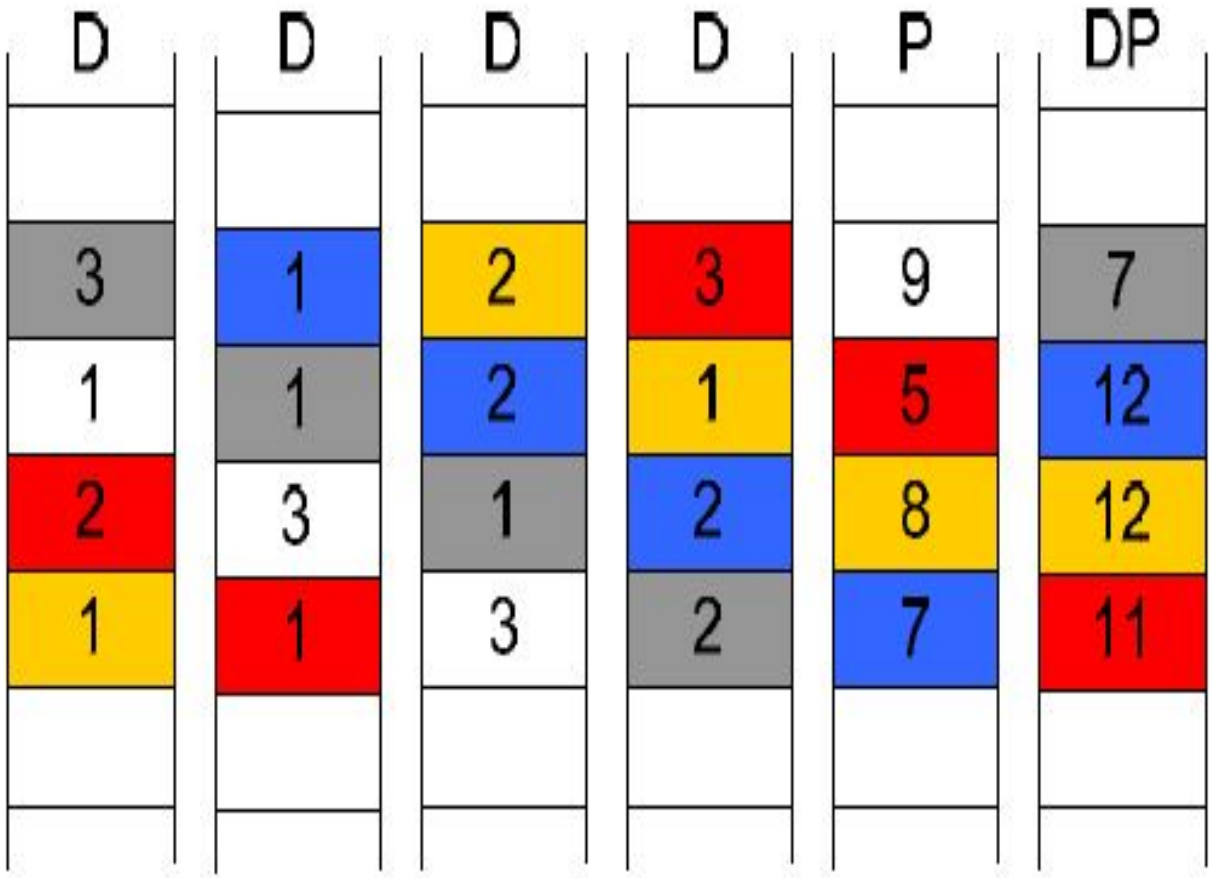
- Two different parity calculations are carried out and stored in separate blocks on different disks
- N+2 disks are required
- Can take care of two disk failures
- Provides extremely high data availability
- Incurs a substantial write penalty



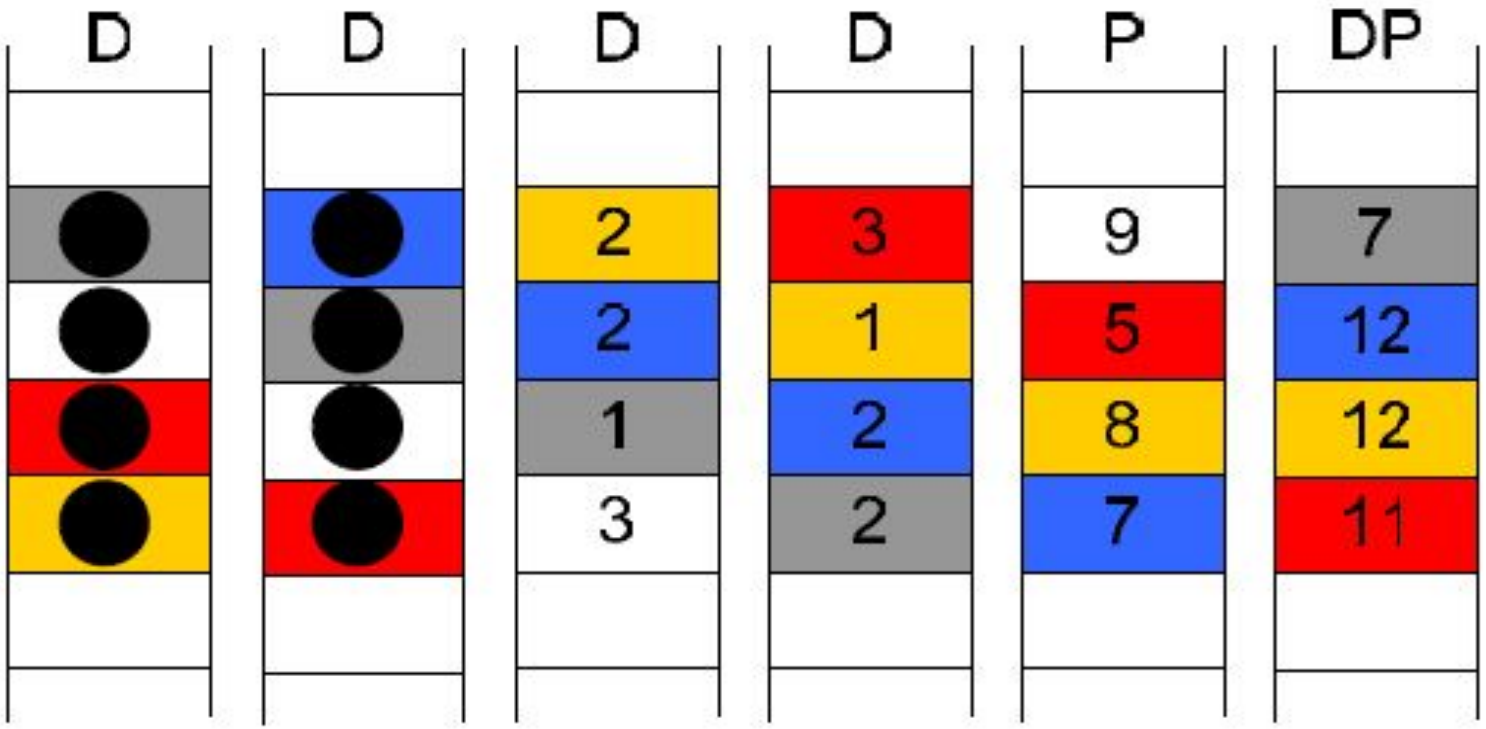
(g) RAID 6 (dual redundancy)

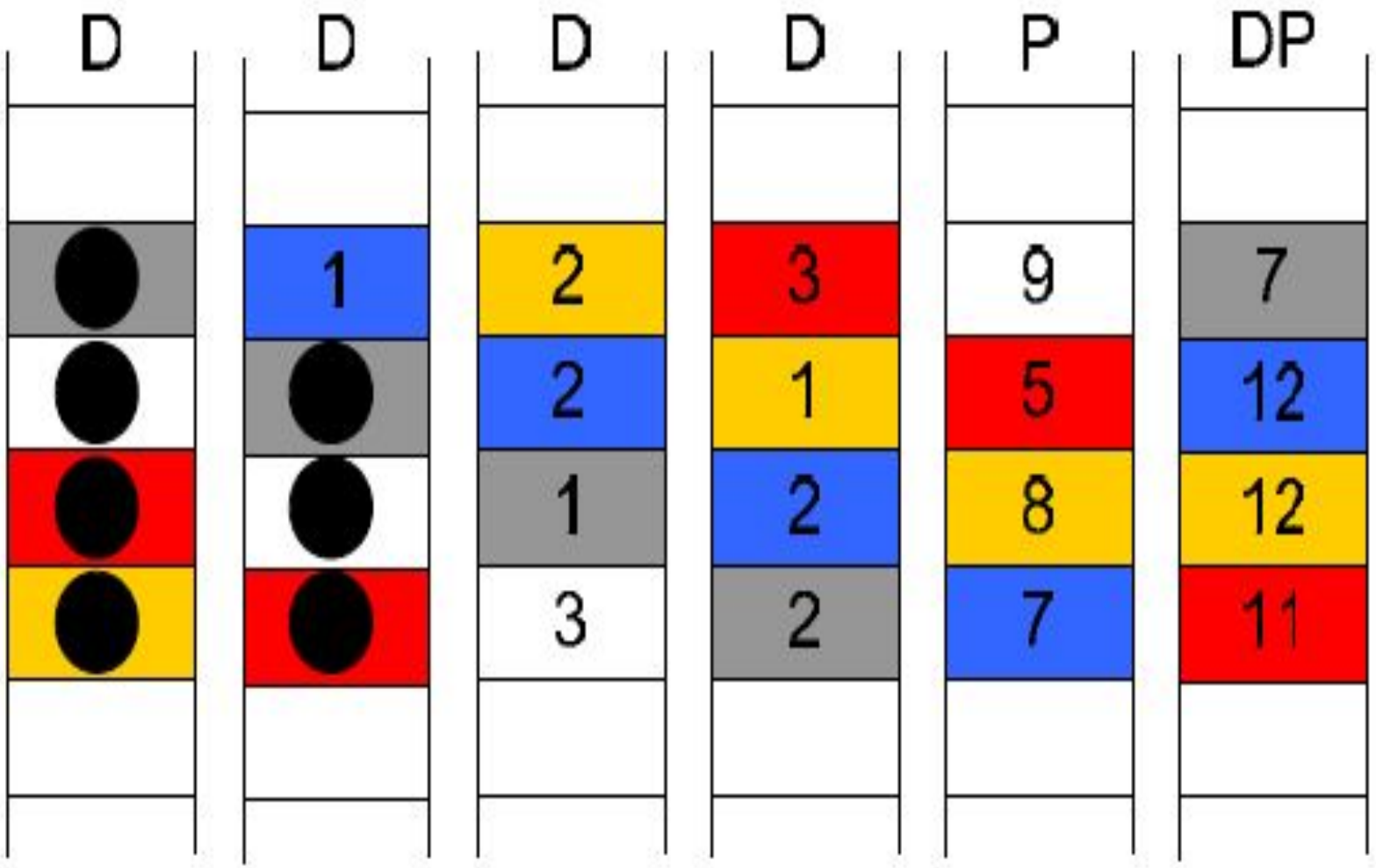
Figure 11.9 RAID Levels (page 2 of 2)

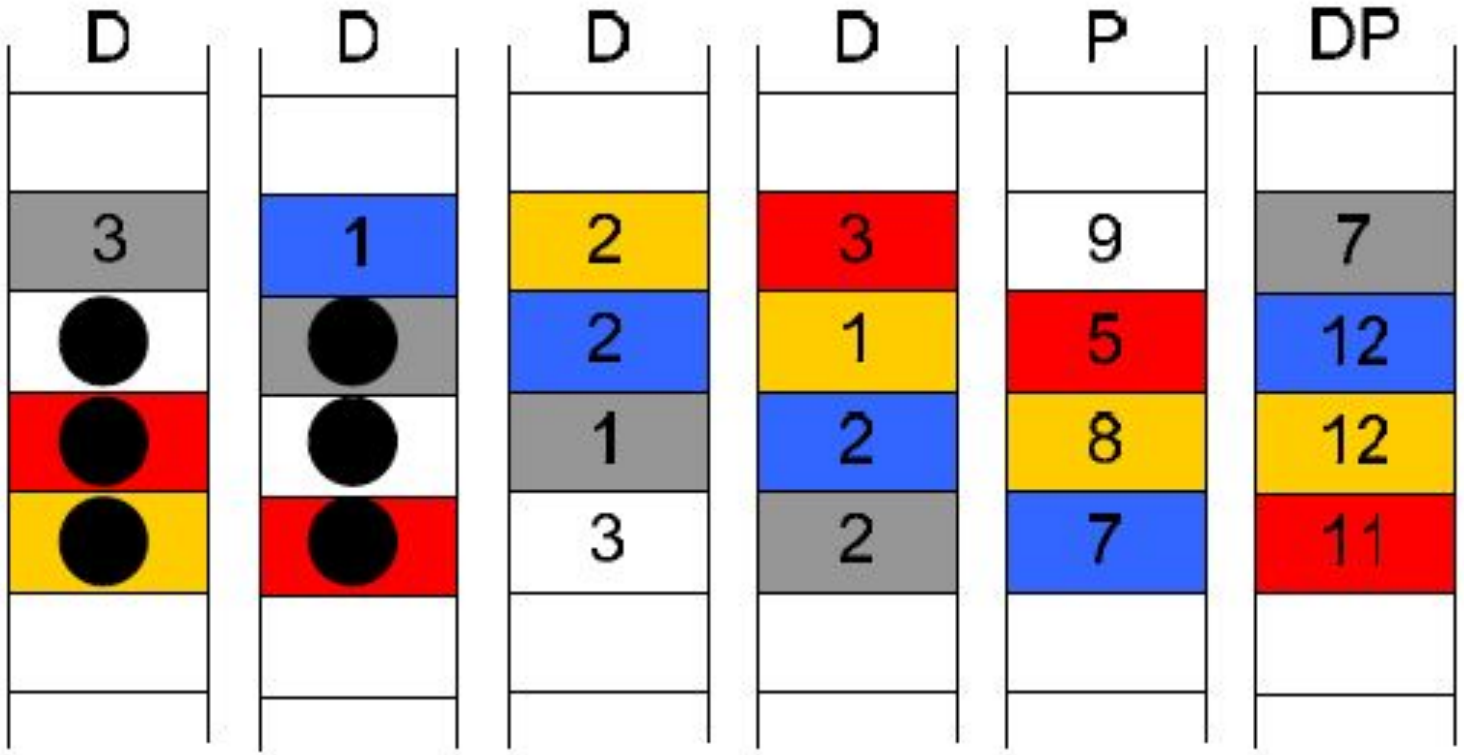
RAID DP

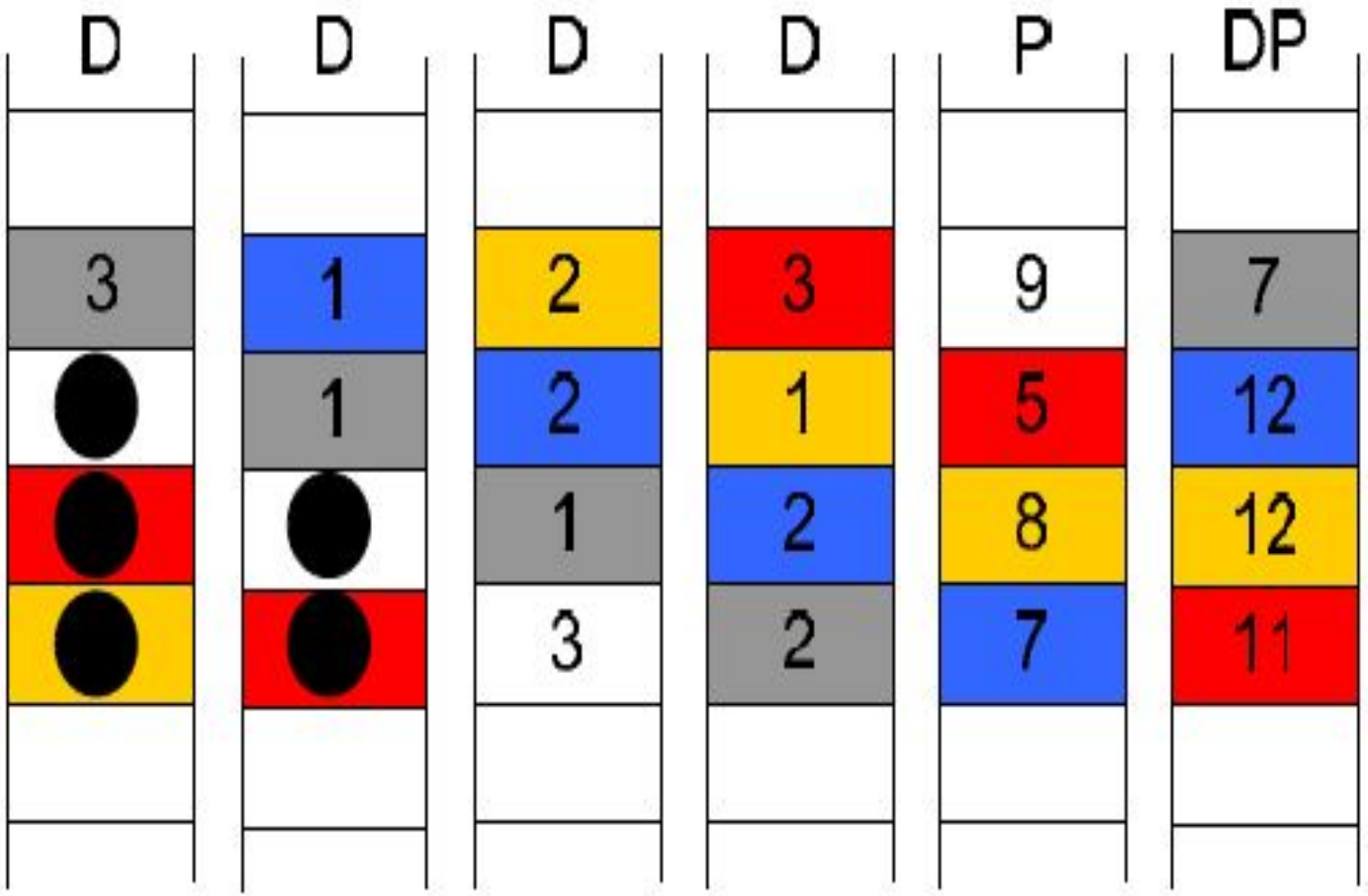


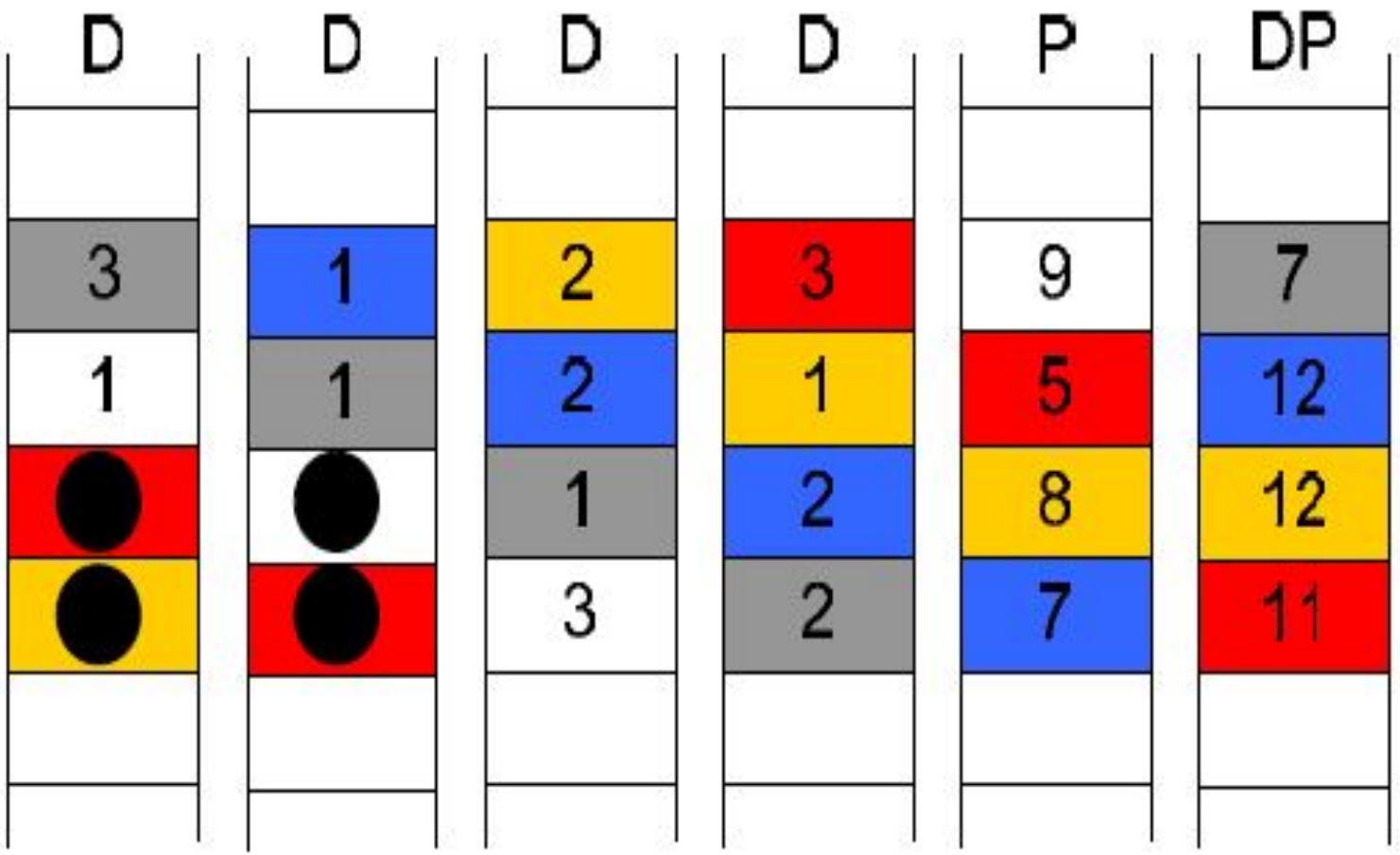
Recovery from failure

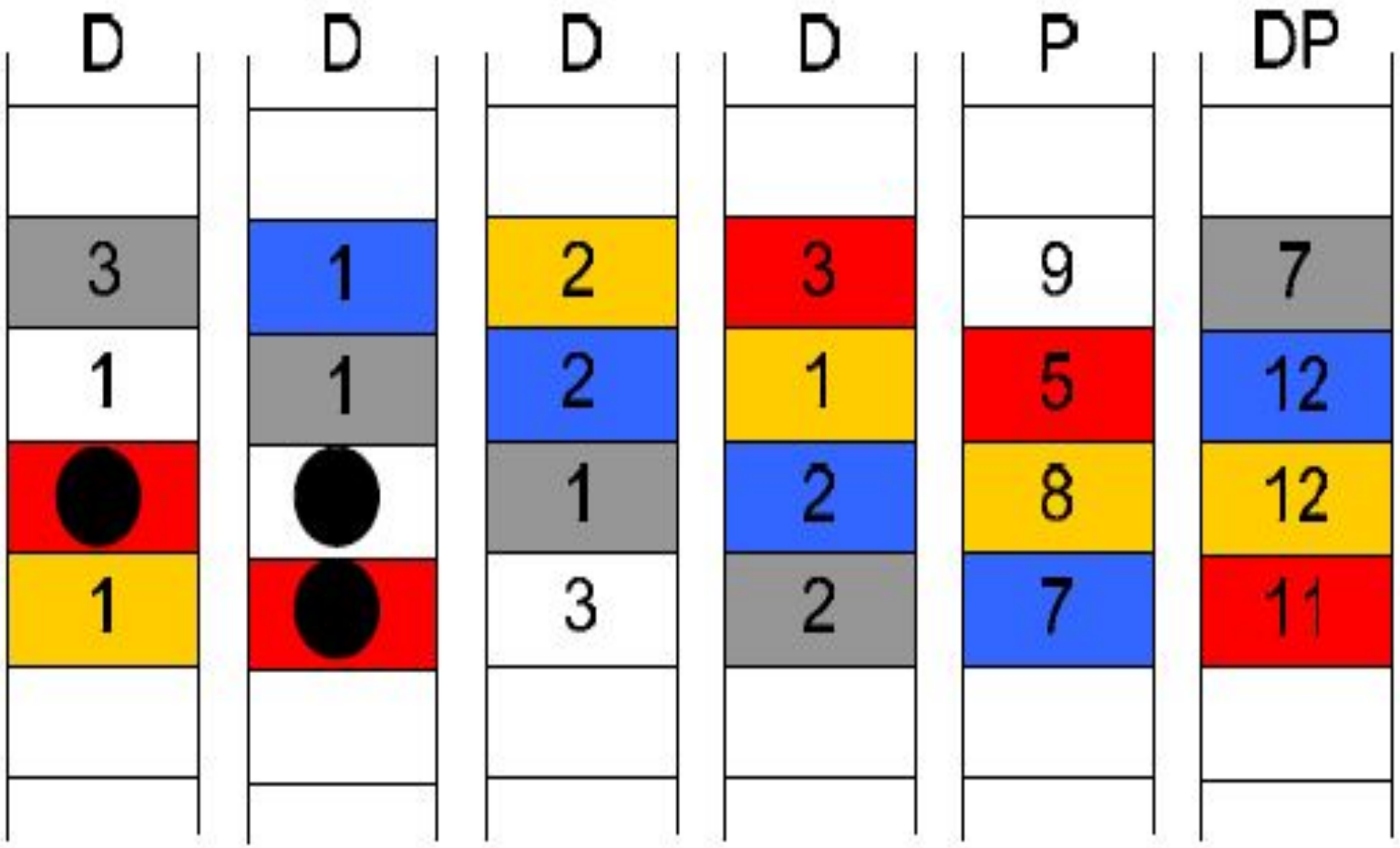


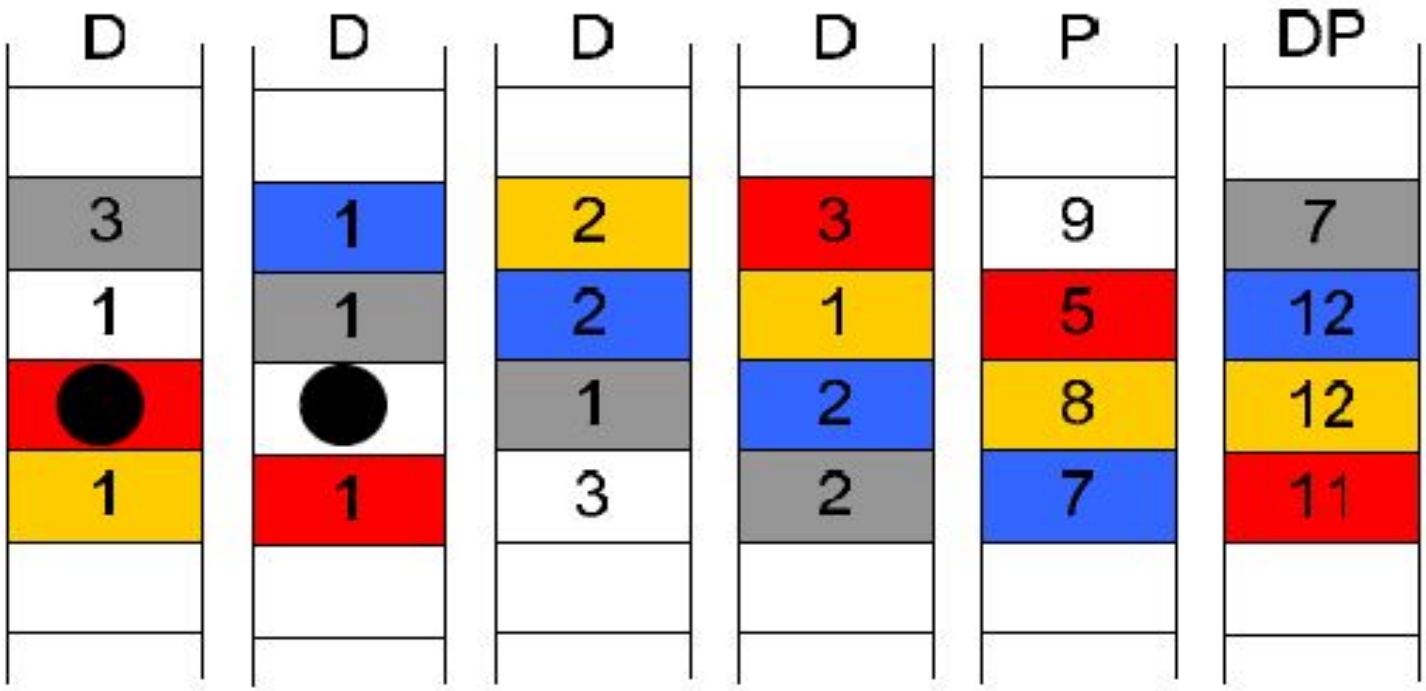

D
D
D
D
P
DP
1
2
3
9
7
2
1
5
12
1
2
8
12
3
2
7
11

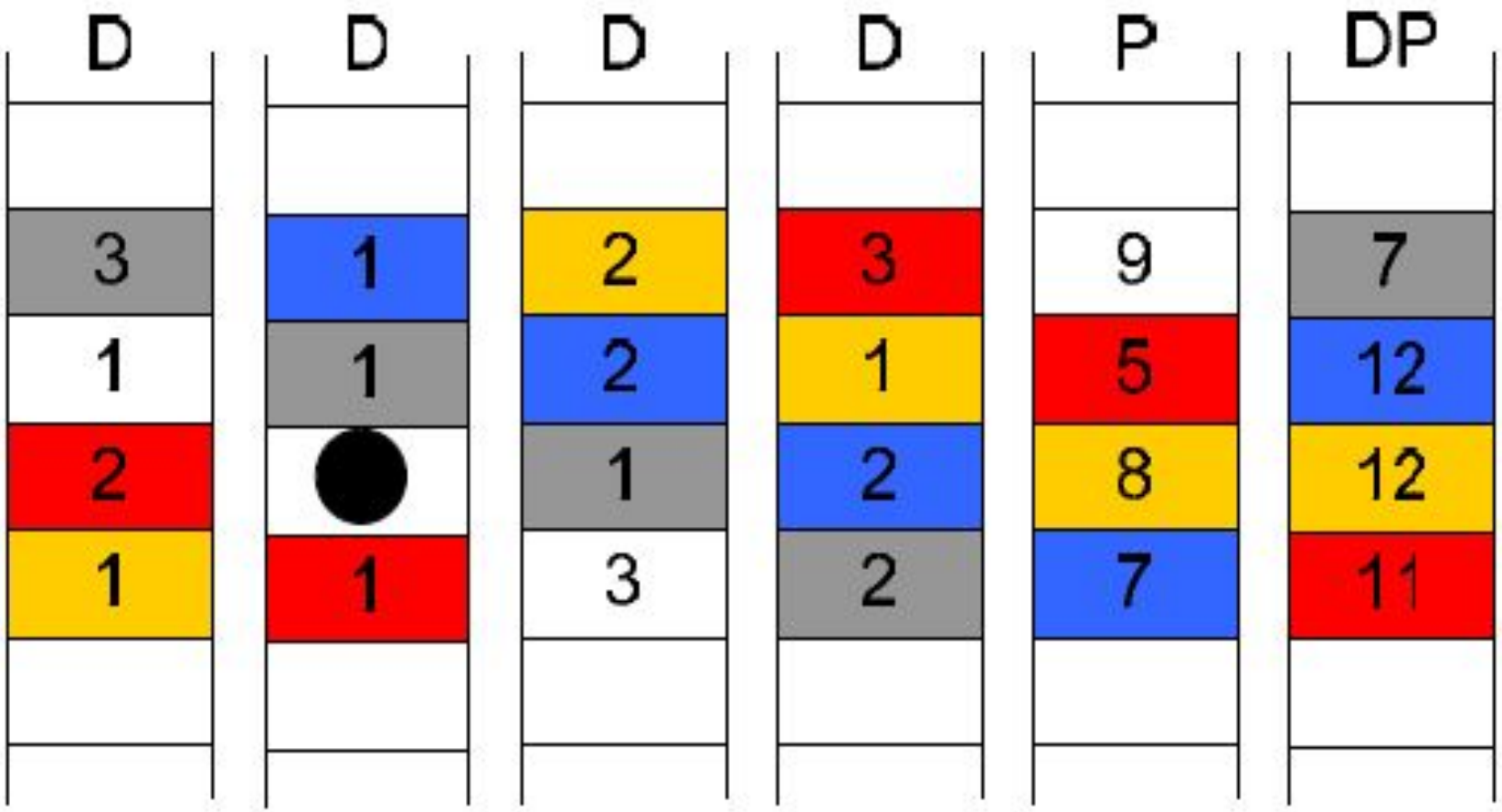

D
D
D
D
P
DP
3
1
2
3
9
7
2
2
1
5
12
1
2
8
12
3
2
7
11

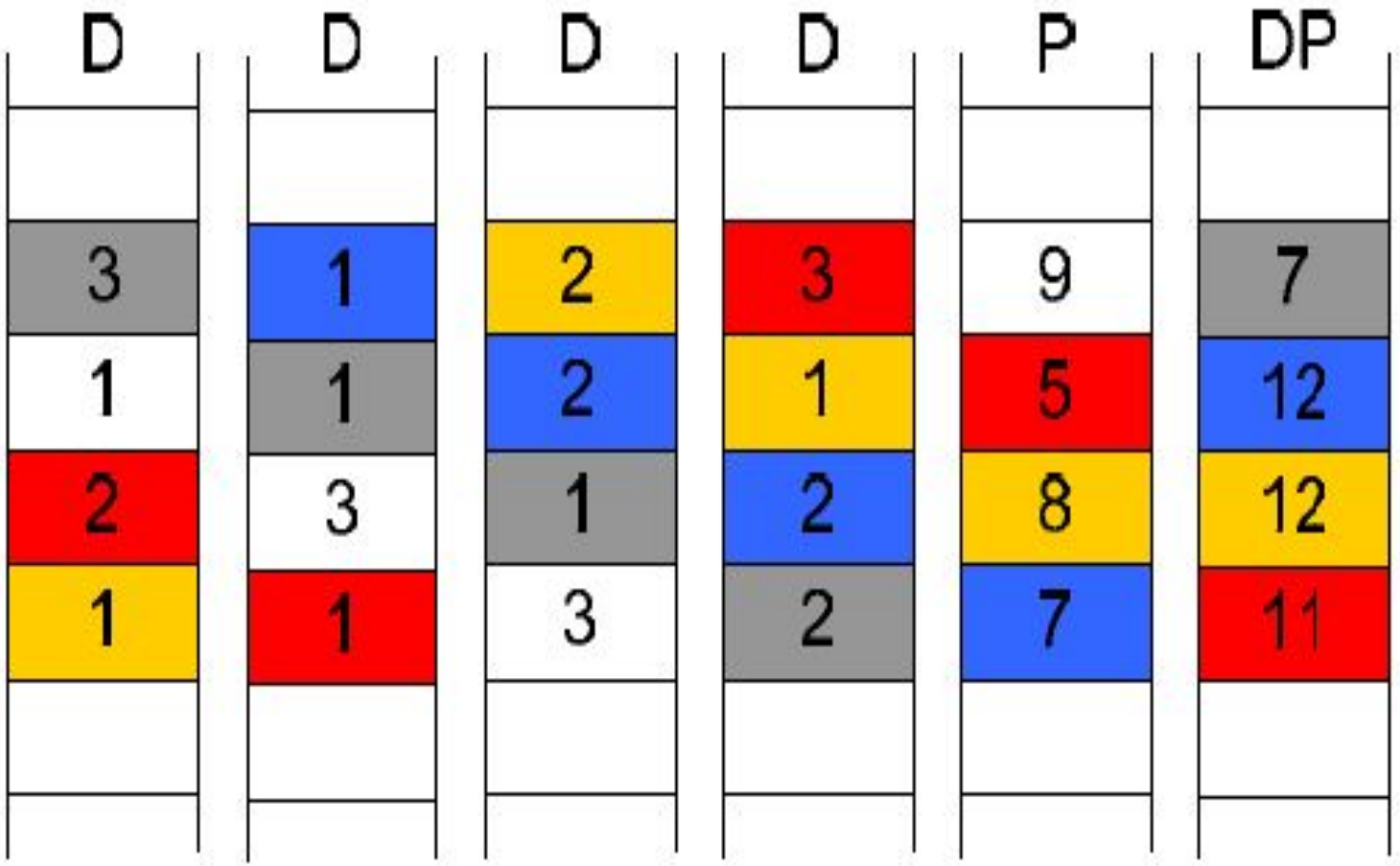

D
D
D
D
P
DP
3
1
2
3
9
7
1
2
1
5
12
1
2
8
12
3
2
7
11


D
D
D
D
P
DP
3
1
2
3
9
7
1
1
2
1
5
12
1
2
8
12
3
2
7
11


D
D
D
D
P
DP
3
1
2
3
9
7
1
1
2
1
5
12
1
2
8
12
1
3
2
7
11


D
D
D
D
P
DP
3
1
1
1
1
1
2
2
1
3
3
1
2
2
9
5
8
7
7
12
12
11


D
D
D
D
P
DP
3
1
2
1
2
1
2
1
1
1
2
1
3
2
9
5
8
7
7
12
12
11

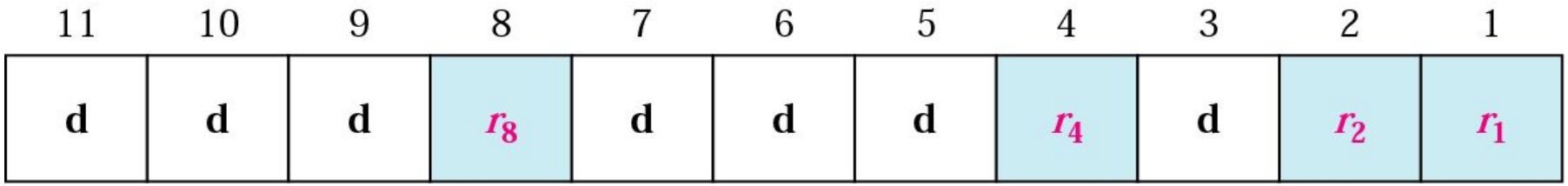

D
D
D
D
P
DP
3
1
2
3
9
7
1
1
2
1
5
12
2
3
1
2
8
12
1
1
3
2
7
11

Hamming Code

How much code redundancy?

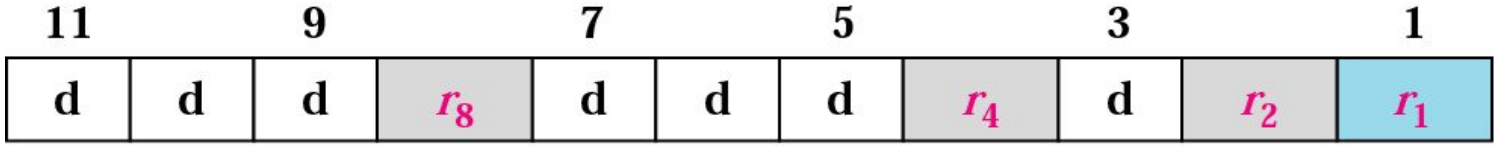
- How many check bits needed, i.e., given m data bits, how many more bits (r) are needed to allow all single-bit errors to be corrected?
  - Log (m) +1

Positions of redundancy bits in Hamming code

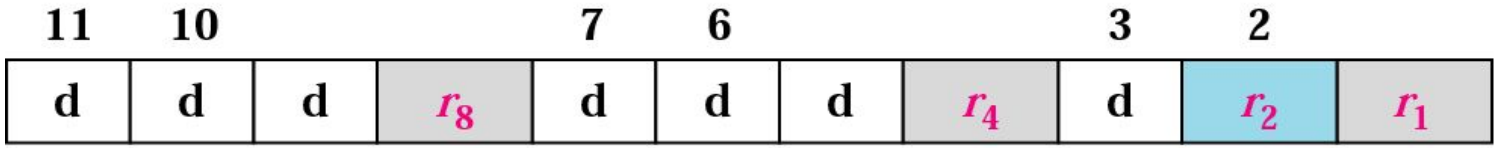


Redundancy bits calculation

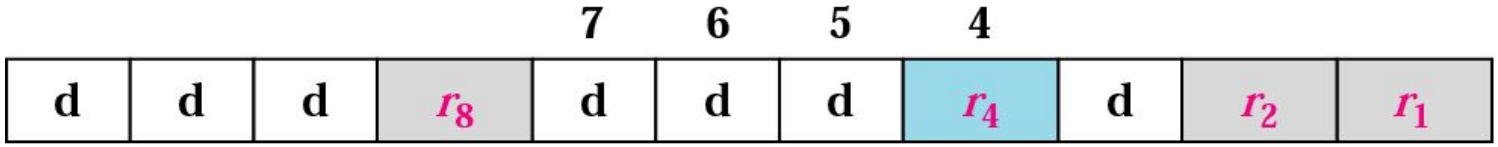
r_1 will take care of these bits.



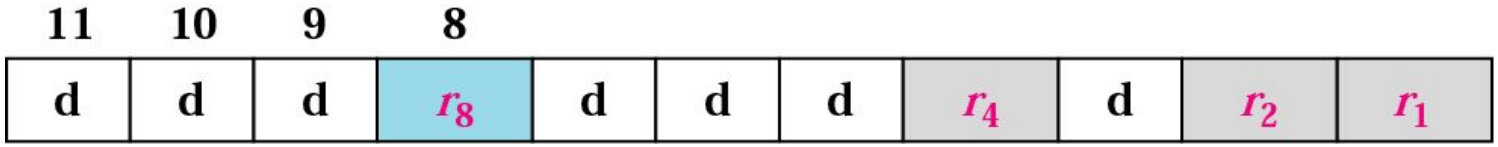
r_2 will take care of these bits.



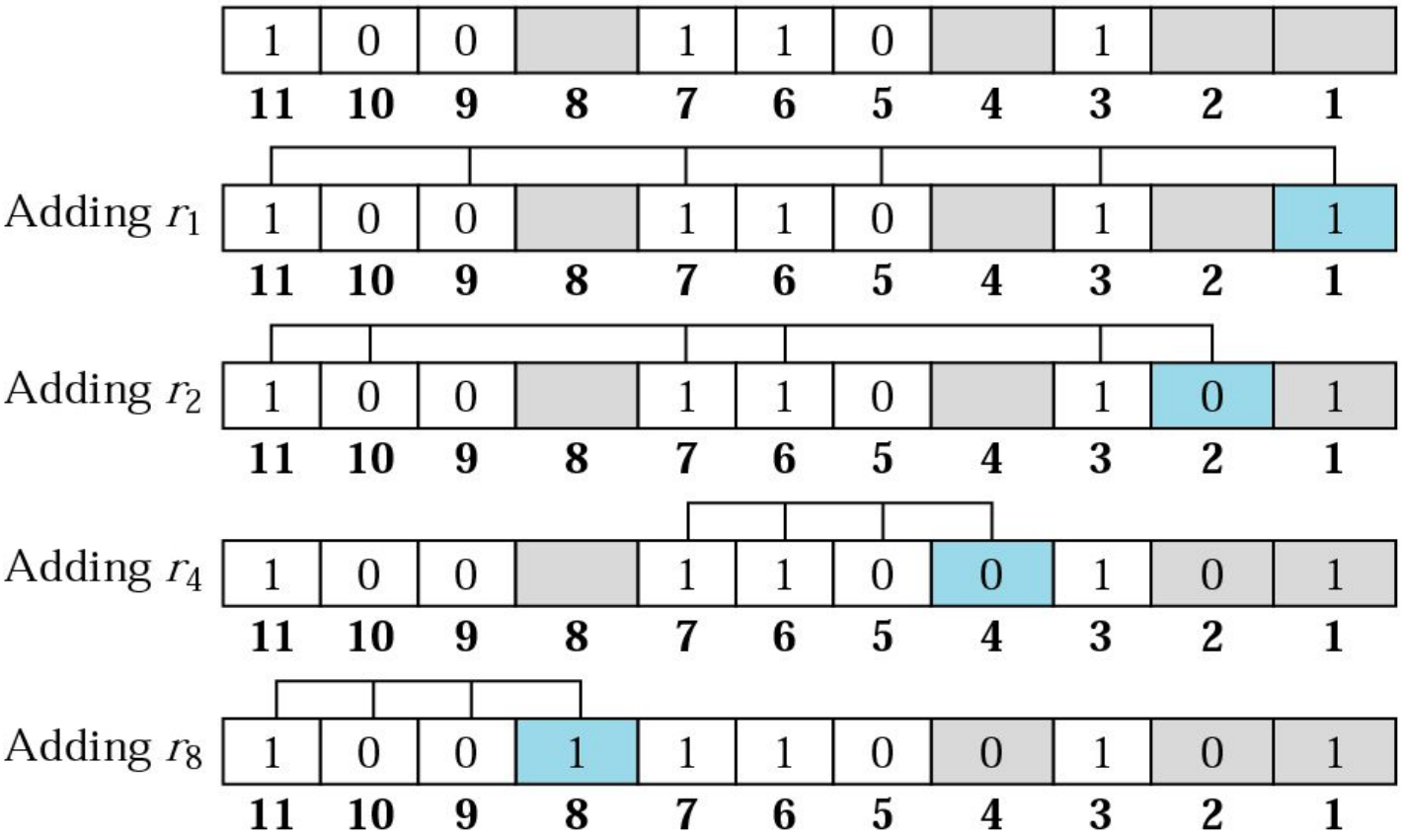
r_4 will take care of these bits.



r_8 will take care of these bits.



Example of redundancy bit calculation



Data:
1 0 0 1 1 0 1

Code:
1 0 0 1 1 1 0 0 1 0 1

Error detection using Hamming code