What is RAID

- Redundant Array of Independent (Inexpensive) Disks
- A set of disk stations treated as one logical station
- Data are distributed over the stations
- Redundant capacity is used for parity allowing for data repair

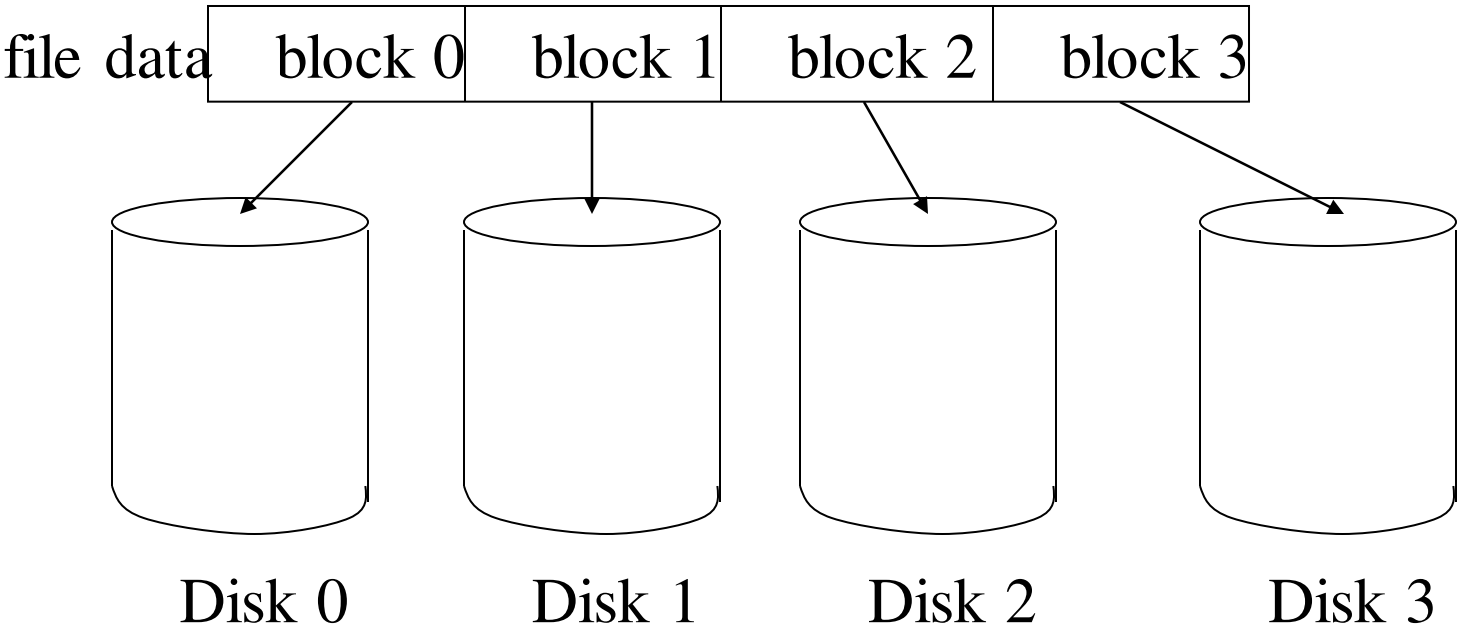
RAID (Redundant Array of Independent Disks)

- Redundant array of inexpensive disks
- Multiple disk database design
- Set of physical disk drives viewed by the OS as a single logical drive
- Data are distributed across the physical drives of an array
- Improve access time and improve reliability
  - large storage capacity
  - redundant data
  - 7 levels (6 levels in common use)

  - differing levels of redundancy, error checking, capacity, and cost

Striping

- Take file data and map it to different disks
- Allows for reading data in parallel



Parity

- Way to do error checking and correction
- Add up all the bits that are 1
  - if even number, set parity bit to 0
  - if odd number, set parity bit to 1
- To actually implement this, do an exclusive OR of all the bits being considered
- Consider the following 2 bytes

byte	parity
10110011	1
01101010	0

- If a single bit is bad, it is possible to correct it

Mirroring

- Keep to copies of data on two separate disks
- Gives good error recovery
  - if some data is lost, get it from the other source
- Expensive
  - requires twice as many disks
- Write performance can be slow
  - have to write data to two different spots
- Read performance is enhanced
  - can read data from file in parallel

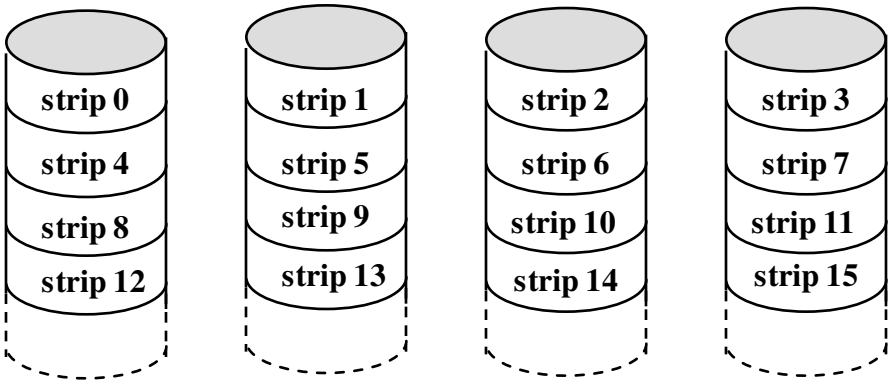
Levels of RAID

- 6 levels of RAID (0-5) have been accepted by industry
- Other kinds have been proposed in literature
- Level 2 and 4 are not commercially available, they are included for clarity

RAID 0

- All data (user and system) are distributed over the disks so that there is a reasonable chance for parallelism
- Disk is logically a set of strips (blocks, sectors,…). Strips are numbered and assigned consecutively to the disks (see picture.)

Raid 0 (No redundancy)



Data mapping Level 0

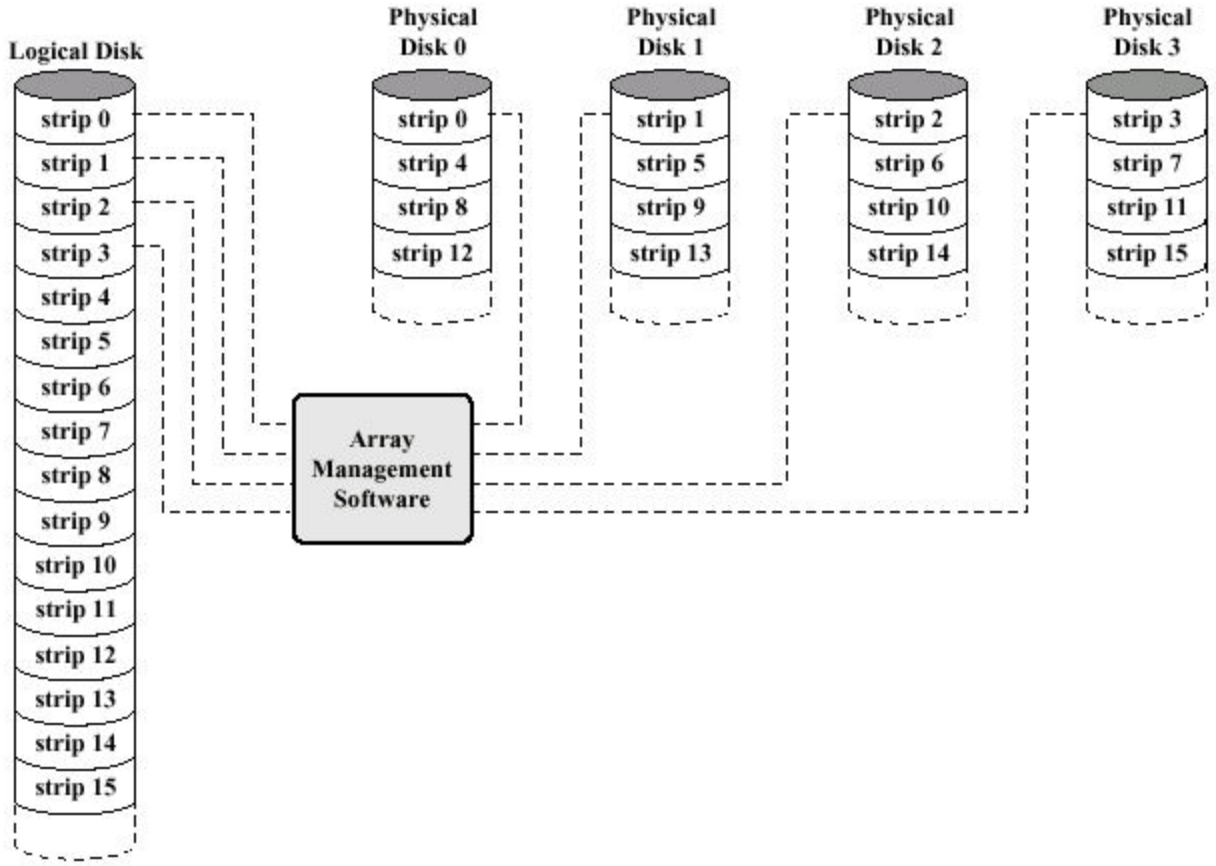


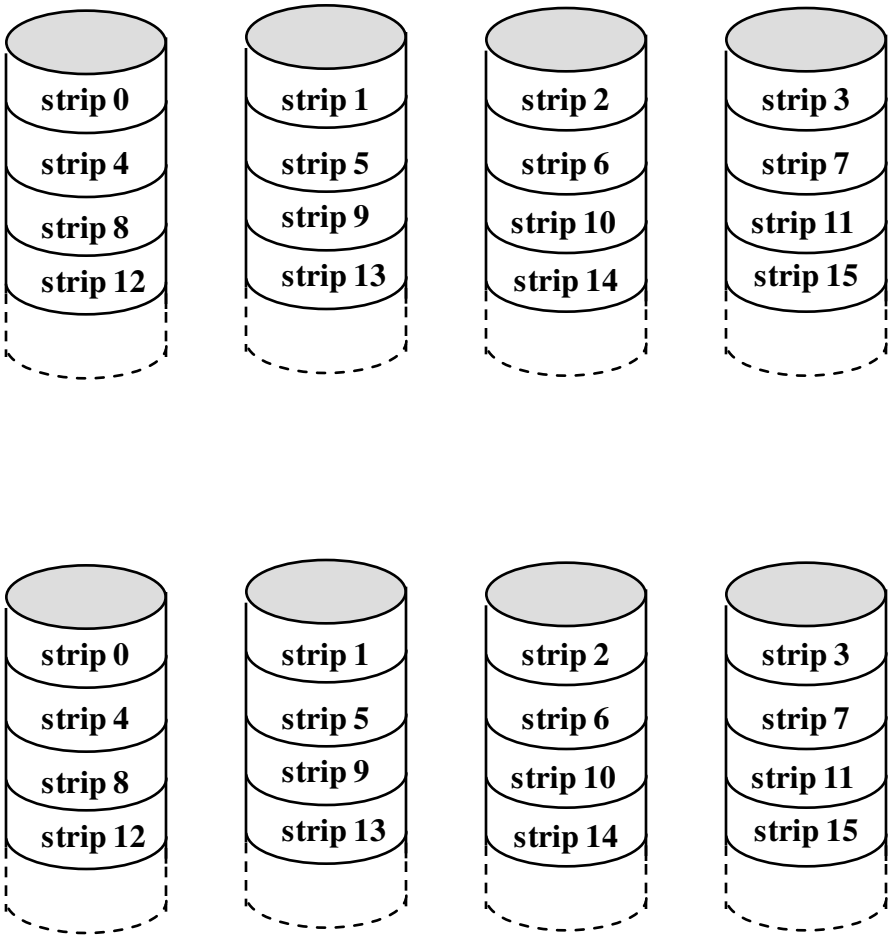
Figure 11.9 Data Mapping for a RAID Level 0 Array [MASS94]

RAID 0:

- Performance depends highly on the the request patterns
- High data transfer rates are reached if
  - Integral data path is fast (internal controllers, I/O bus of host system, I/O adapters and host memory busses)
  - Application generates efficient usage of the disk array by requests that span many consecutive strips
- If response time is important (transactions) more I/O requests can be handled in parallel

- Block level stripping
- Without parity or mirroring
- No redundancy, No backup, No fault tolerance
- Improved performance, faster, as it is uses block level stripping
- Maximum use of storage space as there is no backup
- Any drive failure destroys the array

Raid 1 (mirrored)



RAID 1

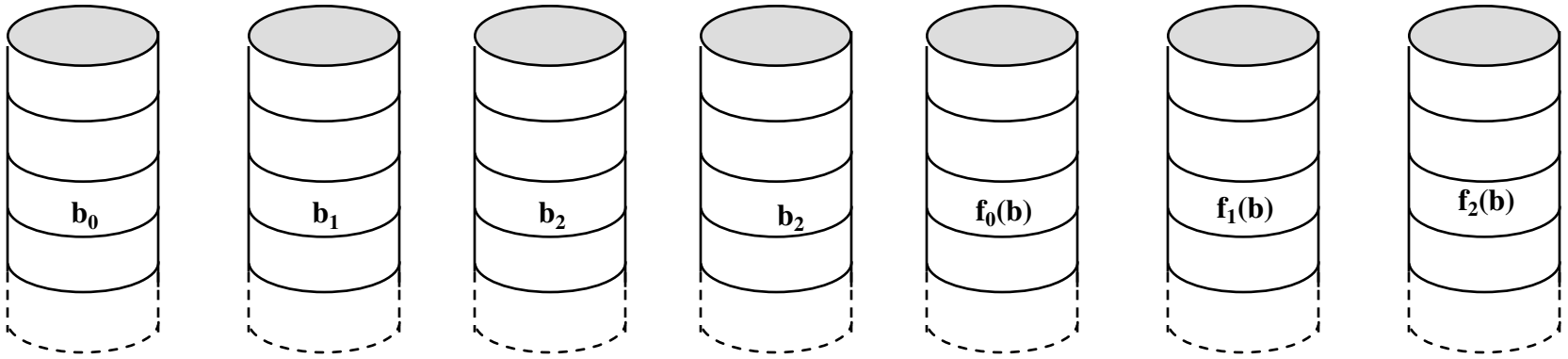
- RAID 1 does not use parity, it simply mirrors the data to obtain reliability
- Plus:
  - Reading request can be served by any of the two disks containing the requested data (minimum search time)
  - Writing request can be performed in parallel to the two disks: no "writing penalty"
  - Recovery from error is easy, just copy the data from the correct disk

RAID 1

- Minus:
  - Price for disks is doubled
  - Will only be used for system critical data that must be available at all times
- RAID 1 can reach high transfer rates and fast response times (~2*RAID 0) if most of the requests are reading requests. In case most requests are writing requests, RAID 1 is not much faster than RAID 0.

- Mirroring
- Without parity
- Data is written identically to two drives (parallel write)
- Not slower, not faster (Write)
- Read can be faster by parallel access
- Fault tolerance
- 50% storage space can be used

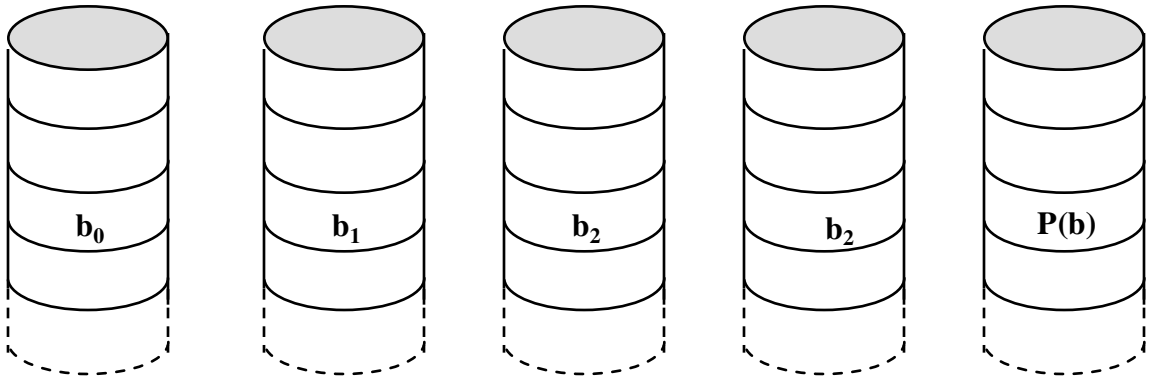
Raid 2 (redundancy through Hamming code)



RAID 2

- Small strips, one byte or one word
- Synchronized disks, each I/O operation is performed in a parallel way
- Error correction code (Hamming code) allows for correction of a single bit error
- Controller can correct without additional delay
- Is still expensive

RAID 3 (bit-interleaved parity)



RAID 3

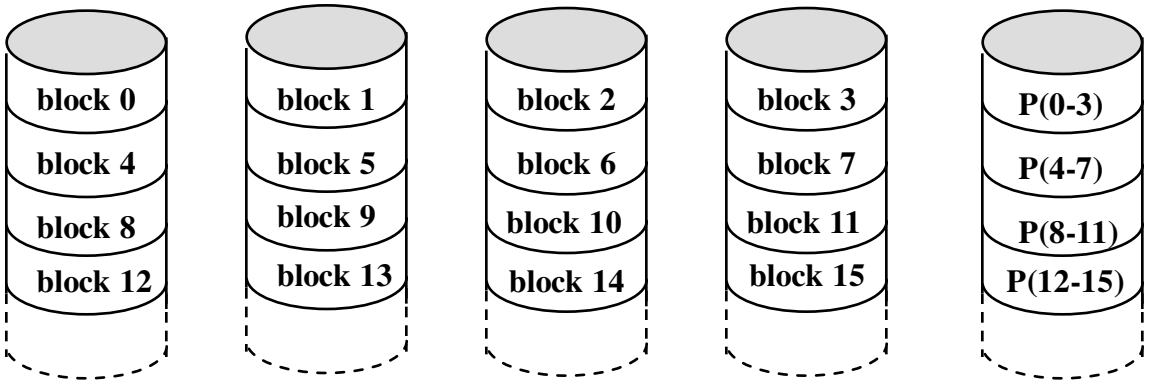
- Level 2 needs $\log_2$(number of disks) parity disks
- Level 3 needs only one, for one parity bit
- In case one disk crashes, the data can still be reconstructed even on line ("reduced mode") and be written (X1-4 data, P parity):

$$P = X1+X2+X3+X4$$

$$X1=P+X2+X3+X4$$

- RAID 2-3 have high data transfer times, but perform only one I/O at the time so that response times in transaction oriented environments are not so good

RAID 4 (block-level parity)



RAID 4

- Larger strips and one parity disk
- Blocks are kept on one disk, allowing for parallel access by multiple I/O requests
- Writing penalty: when a block is written, the parity disk must be adjusted (e.g. writing on X1):

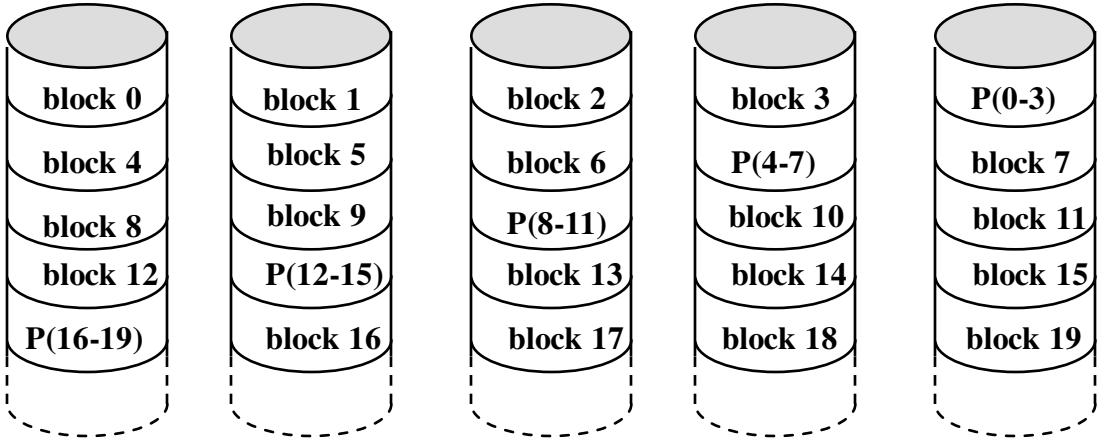
  P =X4+X3+X2+X1

  P'=X4+X3+X2+X1'

  =X4+X3+X2+X1'+X1+X1

  =P+X1+X1'
- Parity disk may be a bottleneck
- Good response times, less transfer rates

RAID 5 (block-level distributed parity)

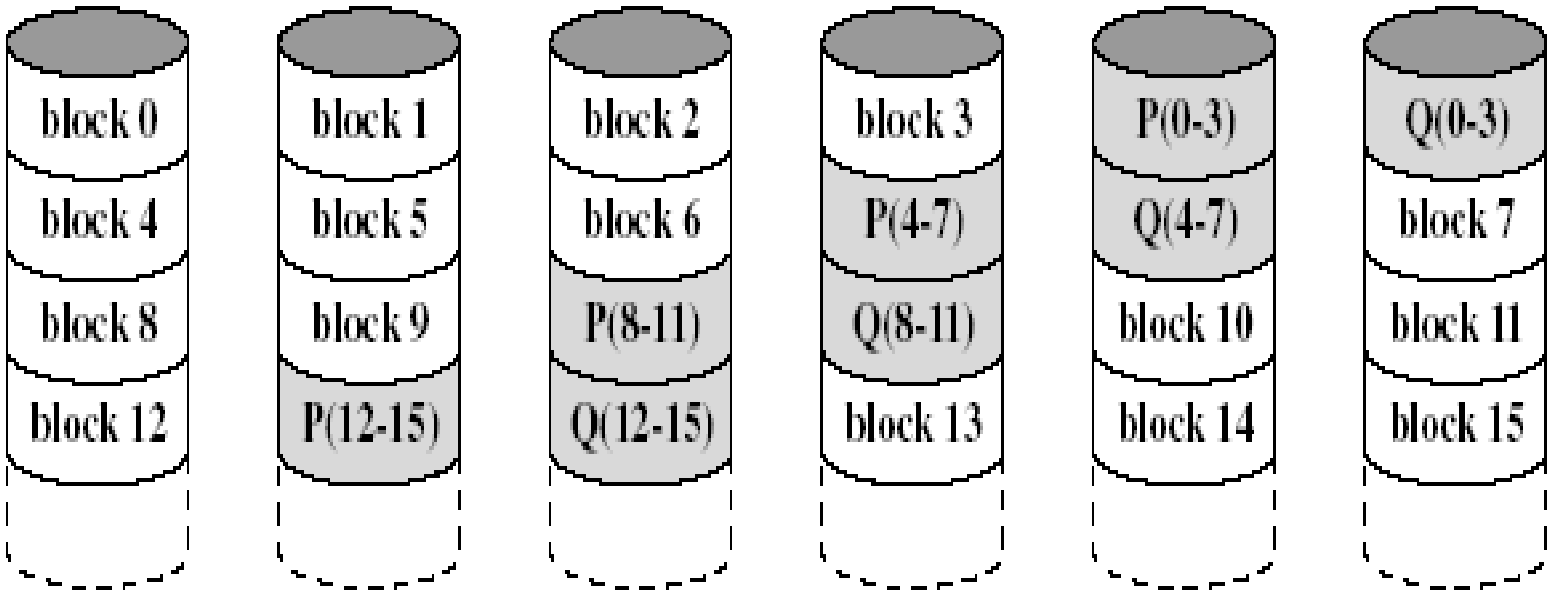


RAID 5

- Distribution of the parity strip to avoid the bottle neck.

- Faster- Block level stripping
- Fault tolerance by Distributed parity (backup of data)
- Distribution of the parity strip to avoid the bottle neck.
- The array is not destroyed by the failure of single hard drive
- Minimum three disks
- Loose one hard disk space for parity
- 75% space

RAID 6


block 0
block 4
block 8
block 12
block 1
block 5
block 9
P(12-15)
block 2
block 6
P(8-11)
Q(12-15)
block 3
P(4-7)
Q(8-11)
block 13
P(0-3)
Q(4-7)
block 10
block 14
Q(0-3)
block 7
block 11
block 15

RAID Level 6

- Two different parity calculations are carried out and stored in separate blocks on different disks.
  - Example: XOR and an independent data check algorithm => makes it possible to regenerate data even if two disks containing user data fail.
- No. of disks required = N + 2 (where N = number of disks required for data).
- Provides HIGH data availability.
- Incurs substantial write penalty as each write affects two parity blocks.
- Three disks would have to fail within MTTR (mean time to repair) interval to cause data to be lost

- Faster-Block level stripping
- Provides fault tolerance up to two failed drives Fault tolerance- double distributed parity (backup of backup of data)
- High availability systems
- Failure of hard drive, slow down the performance of the system
- Loose two hard disk space for parity
- Incurs substantial write penalty

Overview Raid 0-2

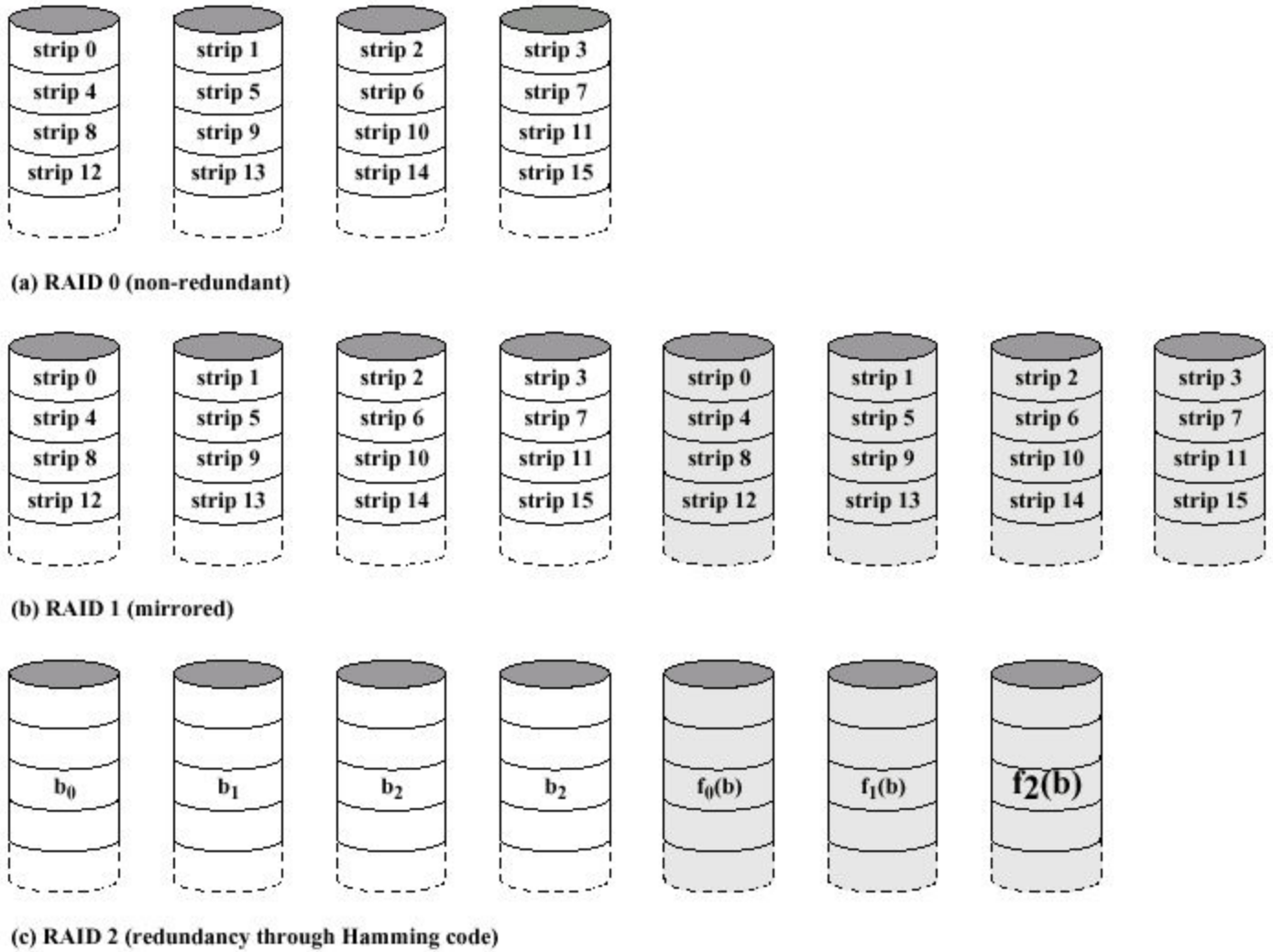
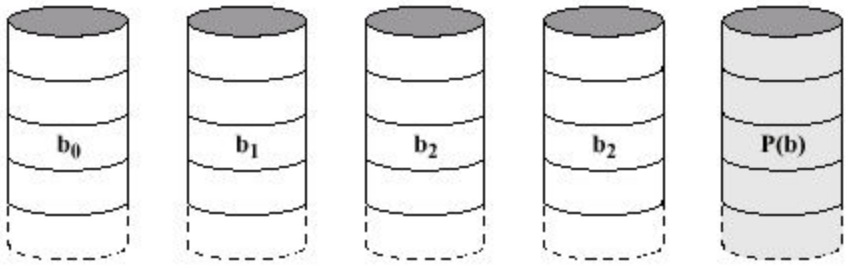
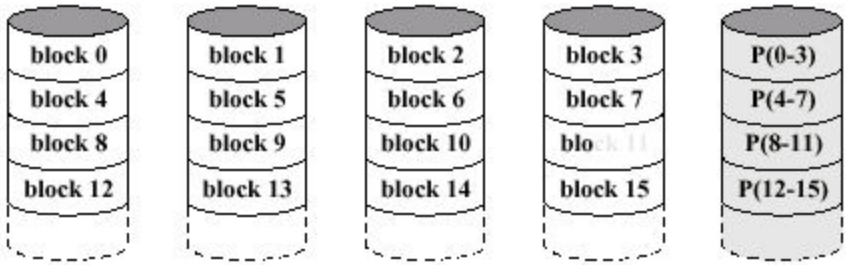


Figure 11.8 RAID Levels (page 1 of 2)

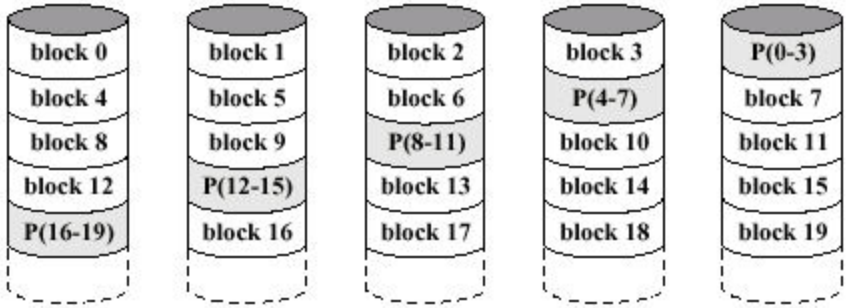
Overview Raid 3-5



(d) RAID 3 (bit-interleaved parity)



(e) RAID 4 (block-level parity)



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

Figure 11.8 RAID Levels (page 2 of 2)