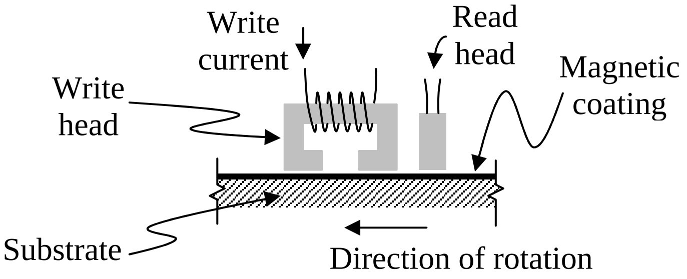
HARD DRIVE CHARACTERISTICS REFRESHER

The read/write head of a hard drive only detects ***changes*** in the magnetic polarity of the material passing beneath it, not the direction of the polarity.

Writes are performed by sending current either one way or the other through the write head coil.

Reads are performed through a separate read head

- Partially shielded magneto resistive (MR) sensor
- Electrical resistance depends on direction of magnetic field – Passing current through it results in different voltage levels for different resistances
- MR head has higher frequency operation allowing better storage density and speed

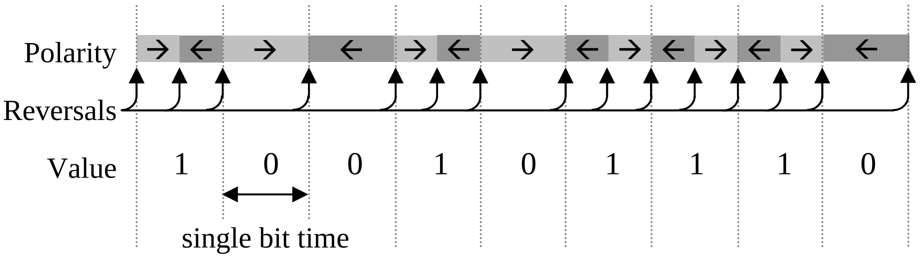


Data Encoding: It might seem natural to use the two directions of magnetic polarization to represent 1's and 0's. This is not the case for two reasons.

- The controllers only detect changes in magnetic direction, not the direction of the field itself.
- Large blocks of data that are all 1's or all 0's would be difficult to read because eventually the controller might lose track of where one bit ended and the next began.

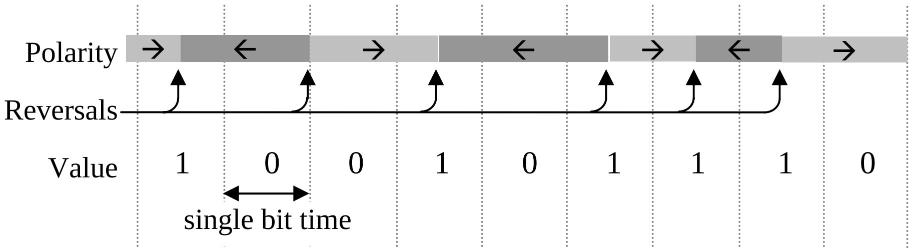
Frequency Modulation (FM)

- a magnetic field change at the beginning of every "bit time"
- a magnetic field change in the middle of a bit time for a logic 1



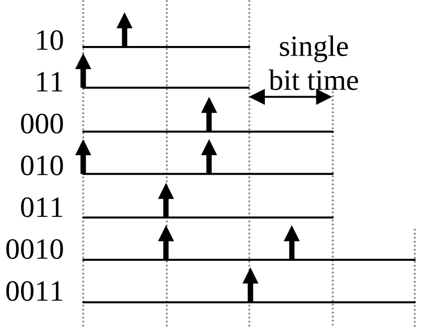
Modified FM (MFM)

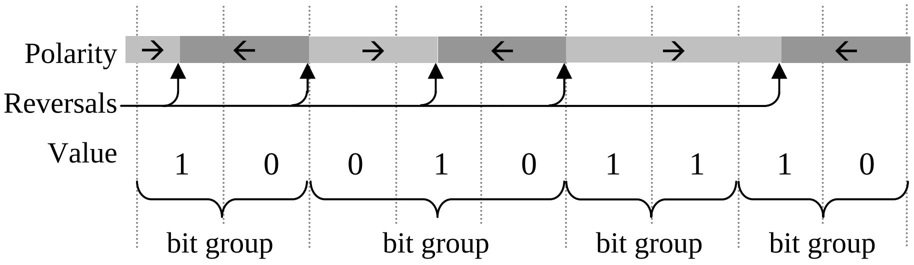
- a magnetic field change between two or more zeros
- a magnetic field change in the middle of a bit time for a logic 1



Run Length Limited (RLL)

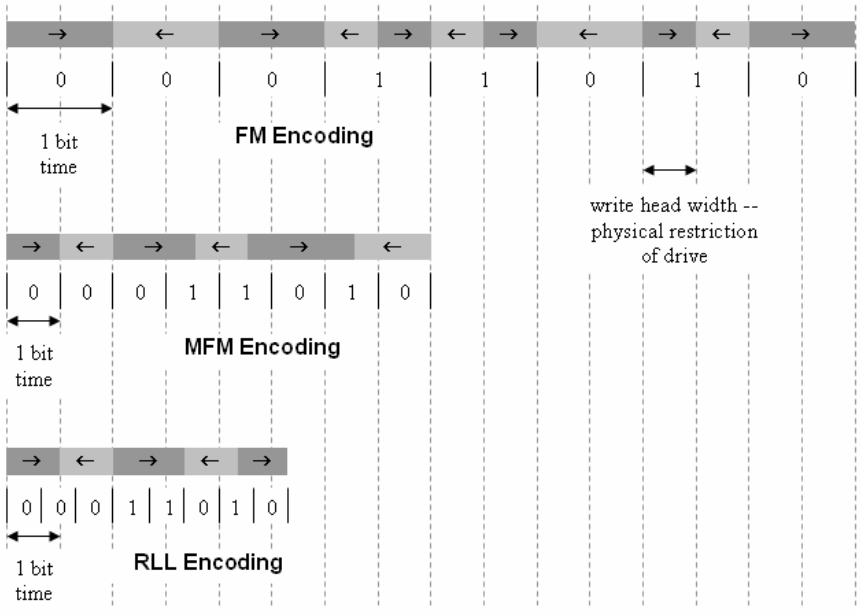
Any pattern of ones and zeros can be represented using a combination of this set of sequences.





Goals of encoding:
- to ensure enough polarity changes to maintain bit synchronization;
- to ensure enough bit sequences are defined so that any sequence of ones and zeros can be handled; and
- to allow for the highest number of bits to be represented with the fewest number of polarity changes

SIDE-BY-SIDE COMPARISON OF ENCODING METHODS



LATEST ENCODING TECHNOLOGIES

- Improved encoding methods have been introduced since the development of RLL
- Use digital signal processing and other methods to realize better data densities.
- These methods include Partial Response, Maximum Likelihood (PRML) and Extended PRML (EPRML) encoding.

SELF-MONITORING, ANALYSIS & REPORTING TECHNOLOGY SYSTEM (S.M.A.R.T.)

- Self-Monitoring, Analysis & Reporting Technology System (S.M.A.R.T.) is a method used to predict hard drive failures
- Controller monitors hard drive functional parameters
- For example, longer spin-up times may indicate that the bearings are going bad
- S.M.A.R.T. enabled drives can provide an alert to the computer's BIOS warning of a parameter that is functioning outside of its normal range
- Attribute values are stored in the hard drive as an integer in the range from 1 to 253. The lower the value, the worse the condition is.
- Depending on the parameter and the manufacturer, different failure thresholds are set for each of the parameters.

Sample S.M.A.R.T. Parameters

- ***Power On Hours:*** This indicates the age of the drive.
- ***Spin Up Time:*** A longer spin up time may indicate a problem with the assembly that spins the platters.
- ***Temperature:*** Higher temperatures also might indicate a problem with the assembly that spins the platters.
- ***Head Flying Height:*** A reduction in the flying height of a Winchester head may indicate it is about to crash into the platters.
- S.M.A.R.T. doesn’t cover all failures, e.g., IC failure or a failure from a catastrophic event

HOW LONG WILL I HAVE TO WAIT FOR MY DATA?

Queuing time: waiting for I/O device to be useable

- Waiting for device – The device may be busy serving another request.
- Waiting for channel – If device shares I/O channel with other devices, the channel may be busy.
- Sleep mode – Energy saving feature may turn off disk forcing O/S to wait for spin-up.

Seek time: the time it takes for the head to move to the correct position

- Find correct track by moving head (moveable head)
- Some details cannot be pinned down and are random
  - Ramping functions for any mechanical movement
  - Distance between current track and desired track
  - Shorter distances and lighter components have reduced seek time
- Typical seek times:
  - Typical seek times for a hard drive are between 4 and 8 ms.
  - CDROMS are slower because of heavier heads.
  - Solid state drives (1 head per track) have little or no seek time – they simply switch heads.

Rotational Latency: the time it takes for the data to rotate to a position beneath the head

- Floppies – 3600 RPM
- Hard Drives – up to 15,000 RMP
- Average rotational delay is 1/2 time for full rotation: ½ × 60 sec/min ÷ rotational speed in RPM
- Maximum rotational delay is 1 full rotation
- Example: For a 7200 RPM disk, average rotational delay = ½ × 60 sec/min 7200 rot/min = 4.2 ms

Transfer time: the time it takes to send the data from the hard drive to the requesting device

Transfer time (T) = b/(rN)

- b = number of bytes to transfer
- N = number of bytes on a track (i.e., bytes per full revolution)
- r = rotation speed in RPS (i.e., tracks per second)

Total access time = queuing + seek + rotational + transfer

ROTATIONAL POSITION SENSING (RPS)

- Allows other devices to use I/O channel while seek is in process.
- When seek is complete, device predicts when data will pass under heads
- At a fixed time before data is expected to come, tries to re-establish communications with requesting processor – if fails to reconnect, must wait full disk turn before new attempt is made – called an RPS miss

RANDOM ACCESS AFFECTS SEEK TIME

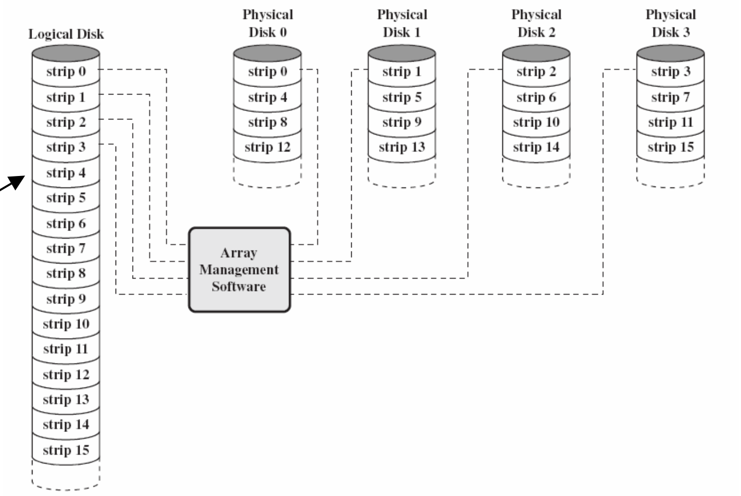
- File is arranged in contiguous sectors – only one seek time per track
- File is scattered to different sectors or device is shared with multiple processes – seek time increased to once per sector

REDUNDANT ARRAY OF INDEPENDENT DISKS (RAID)

- Rate of improvement in secondary storage has not kept up with that of processors or main memory
- In many system, gains can be had through parallel systems
- In disk systems, multiple requests can be serviced concurrently if there are multiple disks and the data for parallel requests is stored in a logical way on different disks
- 7 levels (0 through 6)
- Not a hierarchy
- Set of physical disks viewed as single logical drive by O/S
- Data distributed across multiple physical drives of array
- Can use redundant capacity to store parity information to aid in error correction/detection – ***This is important because the increased number of disks means an increased probability of failures.***

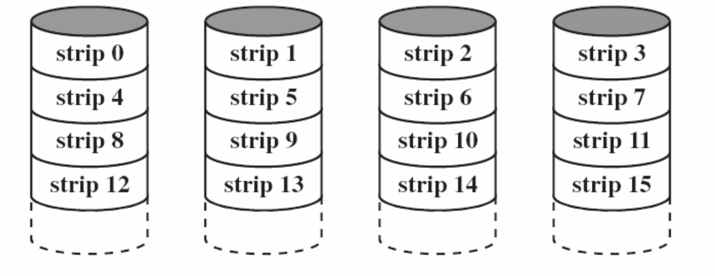
STRIPING

- User's data and applications see one logical drive
- Data is divided into strips
  - o Could be physical blocks, sectors, or some other unit
  - o The strips are then mapped to the different physical drives



RAID 0:

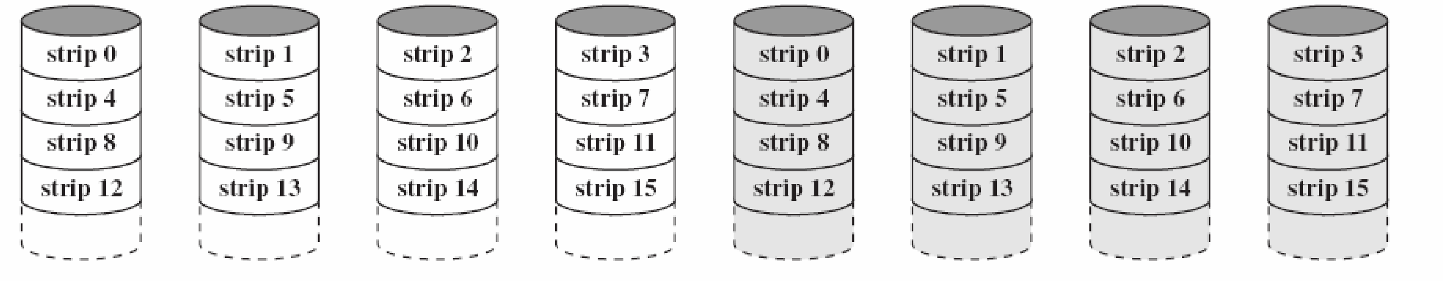
- May not be considered RAID officially as it doesn't support error correction/detection, i.e., there is no redundancy
- Data striped across all disks in a round robin fashion
- Performance characteristics: Increases speed since multiple data requests are probably in sequence of strips and therefore can be done in parallel (High I/O request rate)



RAID 0 (non-redundant)

RAID 1:

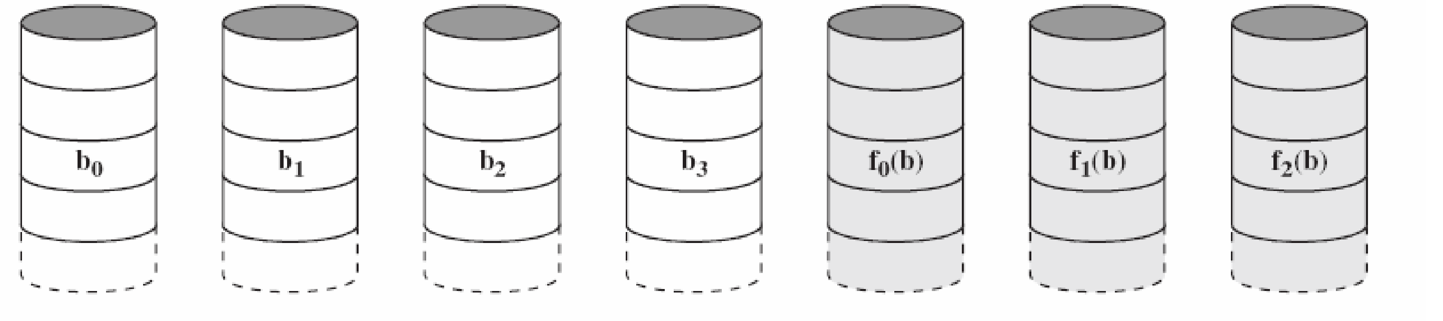
- Mirrored Disks – 2 copies of each strip on separate disks
- Data is striped across disks just like RAID 0
- Read from either –slight performance increase; 1 disk has shorter seek time
- Write to both – slight performance drop; one disk will have longer seek time
- Recovery is simple – swap faulty disk & re-mirror; no down time
- Performance characteristics: Same as for RAID 0
- Expensive since twice capacity is required – likely to be limited to critical system software and data files



RAID 1 (mirrored)

RAID 2:

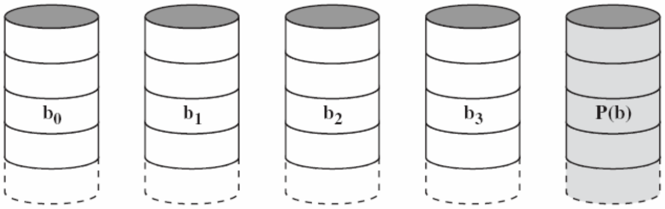
- Disks are synchronized to the point where each head is in same position on each disk
- On a single read or write, all disks are accessed simultaneously
- ***Striped at the bit level*** – Error correction calculated across corresponding bits on disks
- Multiple parity disks store Hamming code w/parity (SEC-DED) error correction in corresponding position
- Only effective in environment where many errors occur
- Not as expensive as RAID 1, but still rather costly
- Not commercially accepted
- Performance characteristics: Only one I/O request at a time (non-parallel)



RAID 2 (redundancy through Hamming code)

RAID 3:

- Similar to RAID 2:
  - o Striped at the bit level
  - o Single read or write accesses all disks
- Only one redundant disk, no matter how large the array – simple parity bit for each set of corresponding bits
- Unable to detect failed drive, but can replace it
- Data on failed drive can be reconstructed from surviving data and parity info
- Performance characteristics: Very high transfer rates
- Problem: Only one I/O request at a time



RAID 3 (bit-interleaved parity)

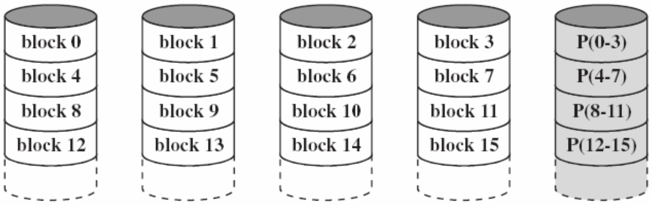
- Example, assume RAID 3 with 5 drives

  $X_4(i) = X_3(i) \oplus X_2(i) \oplus X_1(i) \oplus X_0(i)$
- Failed bit (e.g., X1(i)) can be replaced with:

  $X_1(i) = X_4(i) \oplus X_3(i) \oplus X_2(i) \oplus X_0(i)$
- Equation derived from XOR'ing $X_4(i) \oplus X_1(i)$ to both sides.

RAID 4:

- Larger strips than RAID 2 and RAID 3
- Bit-by-bit parity calculated across strips on each disk – stored on parity disk
- Performance characteristics
- High I/O request rates
- Less suited for high data transfer rates
- Problem – there is a write penalty with each write
  - o old data strip must be read
  - o old parity strip must be read
  - o a new parity strip must be calculated
  - o a new parity strip must be stored
  - o new data must be stored



RAID 4 (block-level parity)

- Original parity calculation

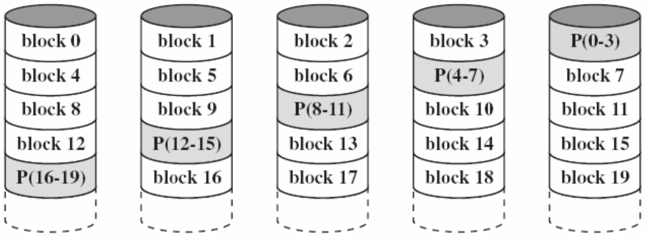
$X_4(i) = X_3(i) \oplus X_2(i) \oplus X_1(i) \oplus X_0(i)$

- New bit is stored (e.g., X1(i)) – parity recalculated:

$X_4'(i) = X_3(i) \oplus X_2(i) \oplus X_1'(i) \oplus X_0(i)$
$X_4'(i) = X_3(i) \oplus X_2(i) \oplus X_1'(i) \oplus X_0(i) \oplus X_1(i) \oplus X_1(i)$
$X_4'(i) = X_4(i) \oplus X_1(i) \oplus X_1'(i)$

RAID 5:

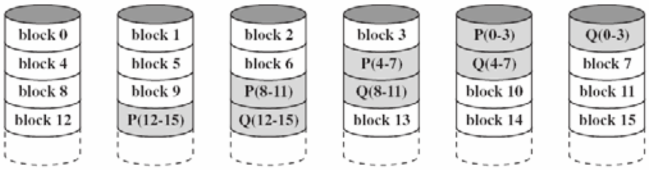
- Like RAID 4 except drops parity disk
- Parity strips are staggered across all data disks
- Round robin allocation for parity stripe
- Avoids RAID 4 bottleneck at parity disk
- Commonly used in network servers



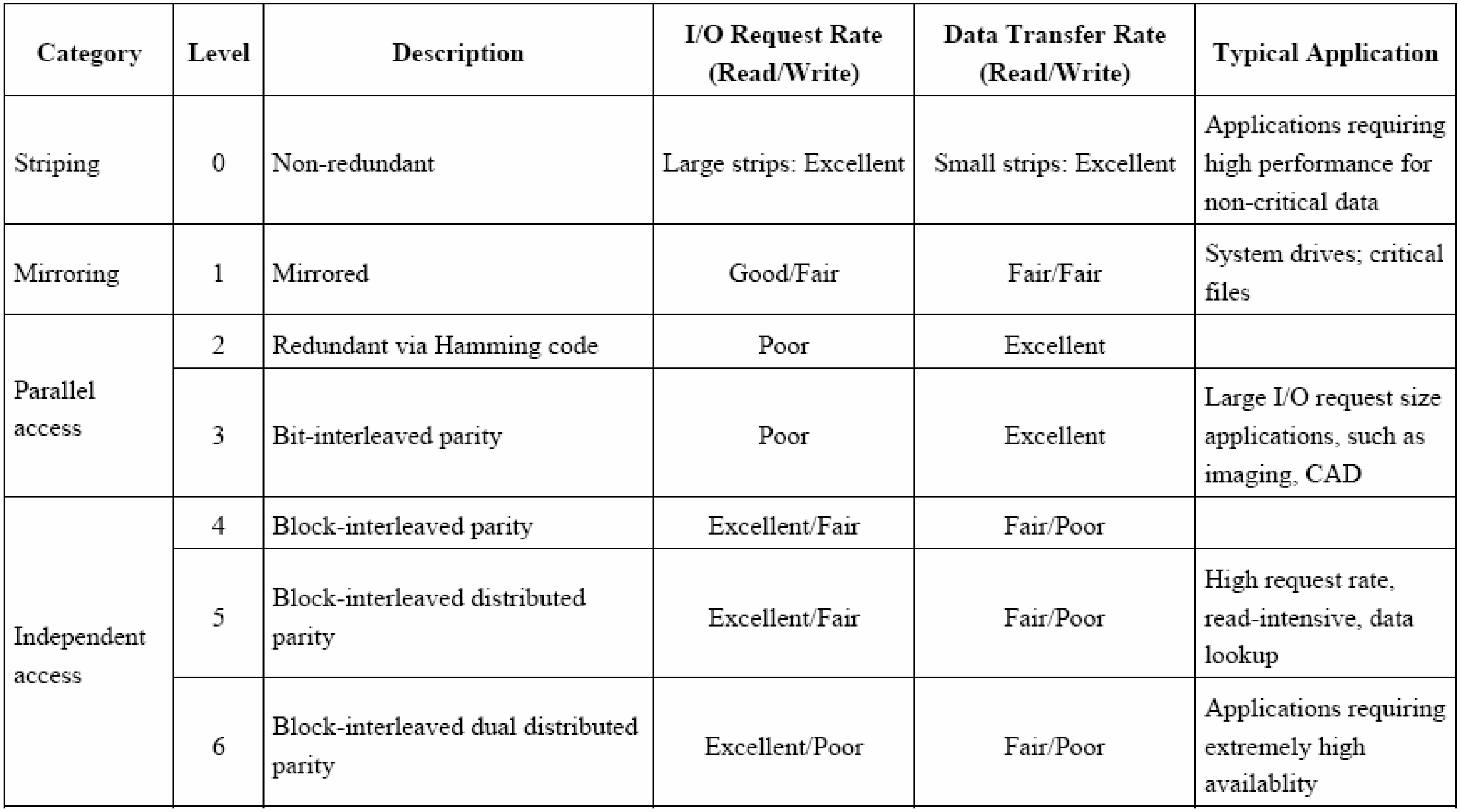
RAID 5 (block-level distributed parity)

RAID 6:

- Two parity calculations
- XOR parity is one of them
- Independent data check algorithm
- Stored in separate blocks on different disks User requirement of N disks needs N+2
- High data availability
- Three disks need to fail for data loss
- Significant write penalty



RAID 6 (dual redundancy)



Category	Level	Description	I/O Request Rate (Read/Write)	Data Transfer Rate (Read/Write)	Typical Application
Striping	0	Non-redundant	Large strips: Excellent	Small strips: Excellent	Applications requiring high performance for non-critical data
Mirroring	1	Mirrored	Good/Fair	Fair/Fair	System drives; critical files
Parallel access	2	Redundant via Hamming code	Poor	Excellent	
	3	Bit-interleaved parity	Poor	Excellent	Large I/O request size applications, such as imaging, CAD
Independent access	4	Block-interleaved parity	Excellent/Fair	Fair/Poor	
	5	Block-interleaved distributed parity	Excellent/Fair	Fair/Poor	High request rate, read-intensive, data lookup
	6	Block-interleaved dual distributed parity	Excellent/Poor	Fair/Poor	Applications requiring extremely high availablity

Source: Stallings, William, *Computer Organization & Architecture – Designing for Performance, 7ed.*, Pearson/Prentice Hall, Upper Saddle River, NJ, 2006, p. 181.