

COM/BLM 376

Computer

Architecture

Chapter 6 External Memory

Asst. Prof. Dr. Gazi Erkan BOSTANCI

ebostanci@ankara.edu.tr

Slides are mainly based on

Computer Organization and Architecture: Designing for Performance by William
Stallings, 9th Edition, Prentice Hall

Outline

1. Magnetic Disk
2. RAID
3. Solid State Drives
4. Optical Memory

MAGNETIC DISK

A disk is a circular **platter** constructed of nonmagnetic material, called the **substrate**, coated with a magnetizable material. Traditionally, the substrate has been an aluminum or aluminum alloy material. More recently, glass substrates have been introduced.

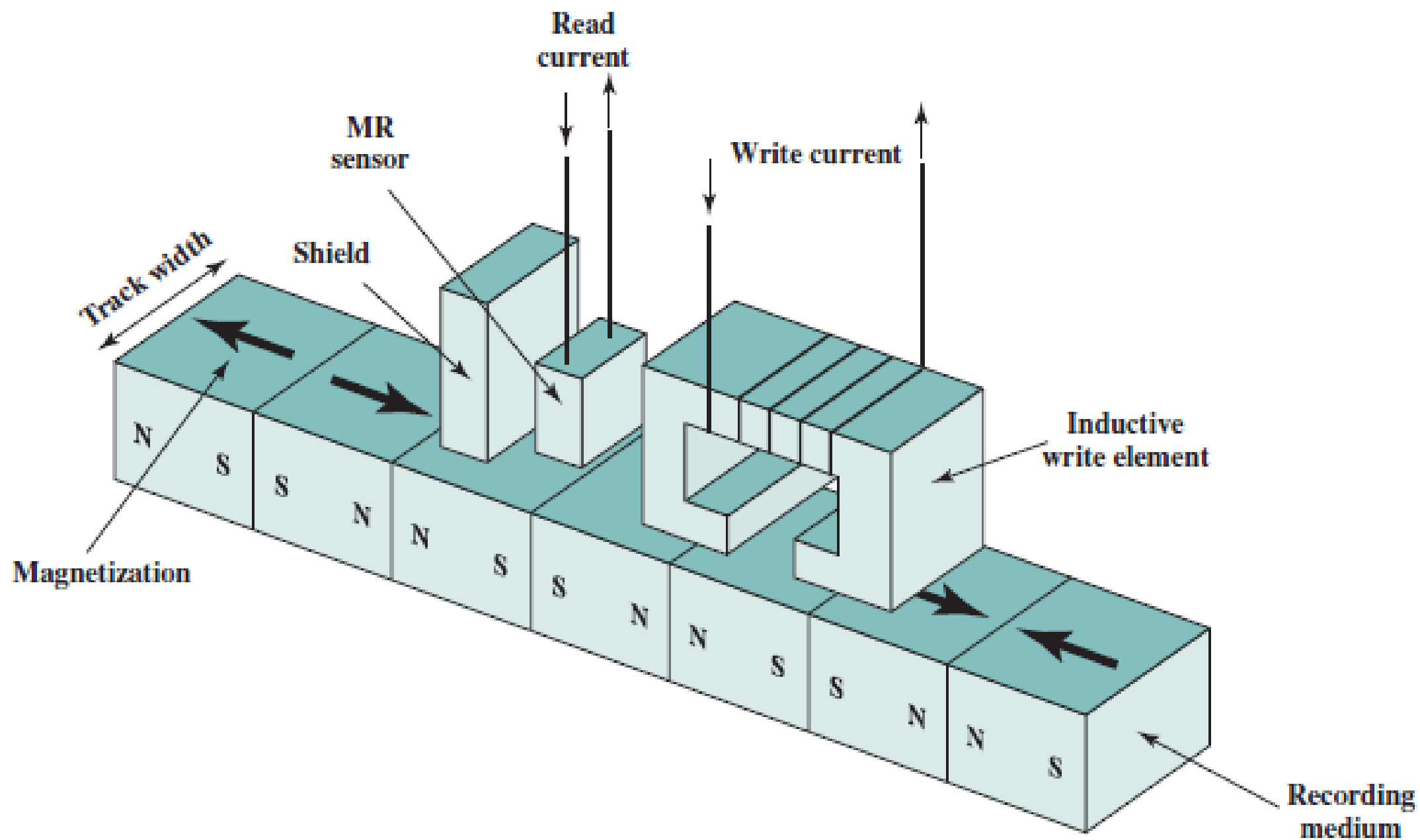
The glass substrate has a number of benefits, including the following:

- Improvement in the uniformity of the magnetic film surface to increase disk reliability
- A significant reduction in overall surface defects to help reduce read-write errors
- Ability to support lower fly heights (described subsequently)
- Better stiffness to reduce disk dynamics
- Greater ability to withstand shock and damage

Magnetic Read and Write Mechanisms

Data are recorded on and later retrieved from the disk via a conducting coil named the **head**; in many systems, there are two heads, a read head and a write head. During a read or write operation, the head is stationary while the platter rotates beneath it.

The write mechanism exploits the fact that electricity flowing through a coil produces a magnetic field. Electric pulses are sent to the write head, and the resulting magnetic patterns are recorded on the surface below, with different patterns for positive and negative currents.



An electric current in the wire induces a magnetic field across the gap, which in turn magnetizes a small area of the recording medium. Reversing the direction of the current reverses the direction of the magnetization on the recording medium.

The traditional read mechanism exploits the fact that a magnetic field moving relative to a coil produces an electrical current in the coil. When the surface of the disk passes under the head, it generates a current of the same polarity as the one already recorded.

Contemporary rigid disk systems use a different read mechanism, requiring a separate read head, positioned for convenience close to the write head. The read head consists of a partially shielded **magnetoresistive** (MR) sensor.

The MR material has an electrical resistance that depends on the direction of the magnetization of the medium moving under it. By passing a current through the MR sensor, resistance changes are detected as voltage signals.

The MR design allows higher-frequency operation, which equates to greater storage densities and operating speeds.

Data Organization and Formatting

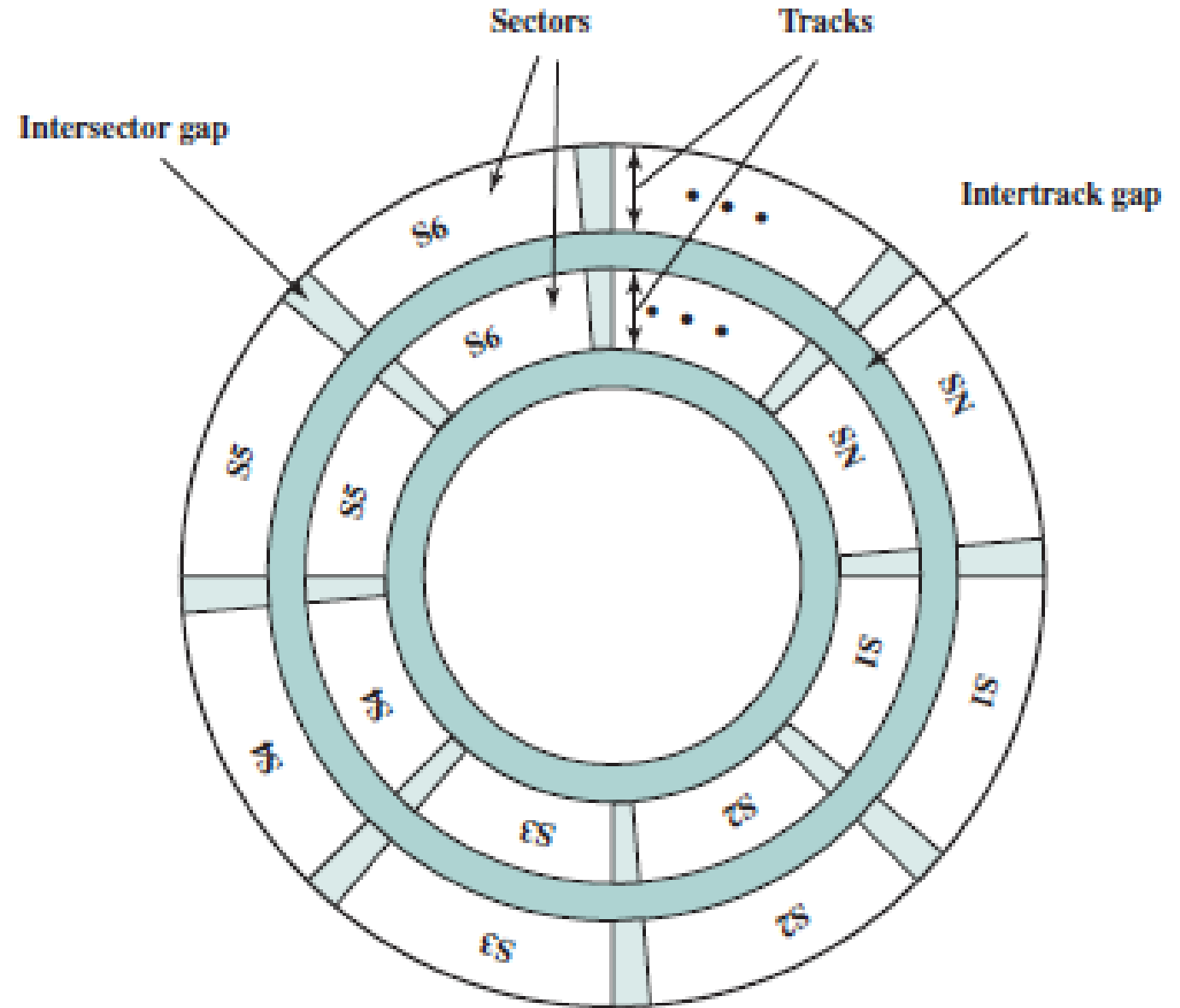
The head is a relatively small device capable of reading from or writing to a portion of the platter rotating beneath it. This gives rise to the organization of data on the platter in a concentric set of rings, called **tracks**.

Each track is the same width as the head. There are thousands of tracks per surface.

Adjacent tracks are separated by **gaps**. This prevents, or at least minimizes, errors due to misalignment of the head or simply interference of magnetic fields.

Data are transferred to and from the disk in **sectors**. There are typically hundreds of sectors per track, and these may be of either fixed or variable length.

In most contemporary systems, fixed-length sectors are used, with 512 bytes being the nearly universal sector size. To avoid imposing unreasonable precision requirements on the system, adjacent sectors are separated by intratrack (intersector) gaps.



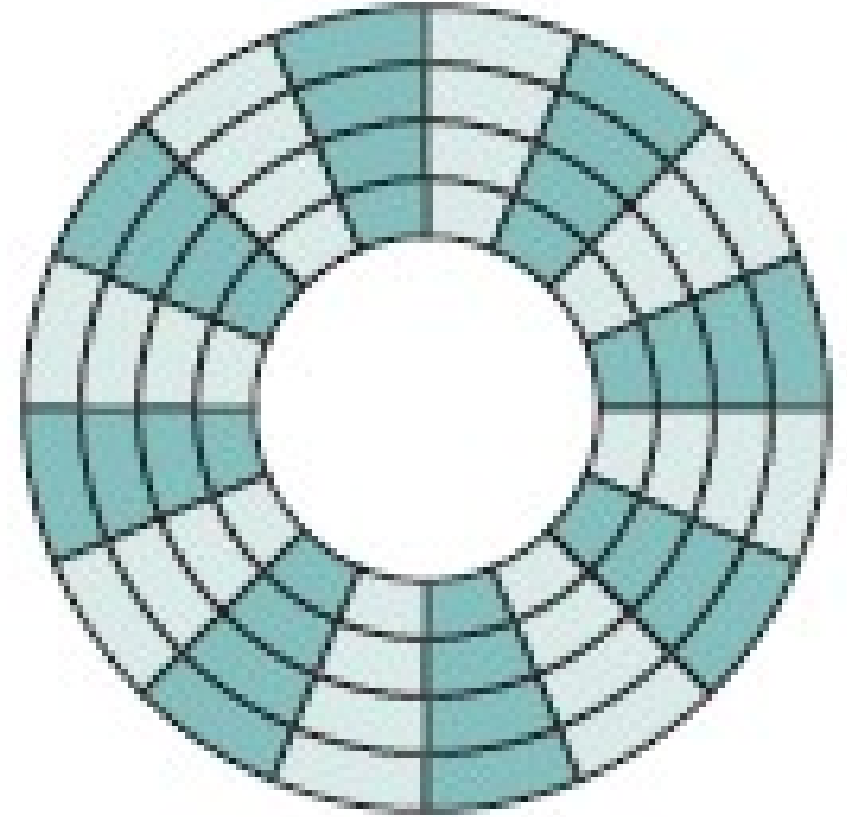
A bit near the center of a rotating disk travels past a fixed point (such as a read– write head) slower than a bit on the outside. Therefore, some way must be found to compensate for the variation in speed so that the head can read all the bits at the same rate.

This can be done by increasing the spacing between bits of information recorded in segments of the disk. The information can then be scanned at the same rate by rotating the disk at a fixed speed, known as the **constant angular velocity** (CAV).

Figure shows the layout of a disk using CAV. The disk is divided into a number of pie-shaped sectors and into a series of concentric tracks.

The advantage of using CAV is that individual blocks of data can be directly addressed by track and sector. To move the head from its current location to a specific address, it only takes a short movement of the head to a specific track and a short wait for the proper sector to spin under the head.

The disadvantage of CAV is that the amount of data that can be stored on the long outer tracks is the only same as what can be stored on the short inner tracks.

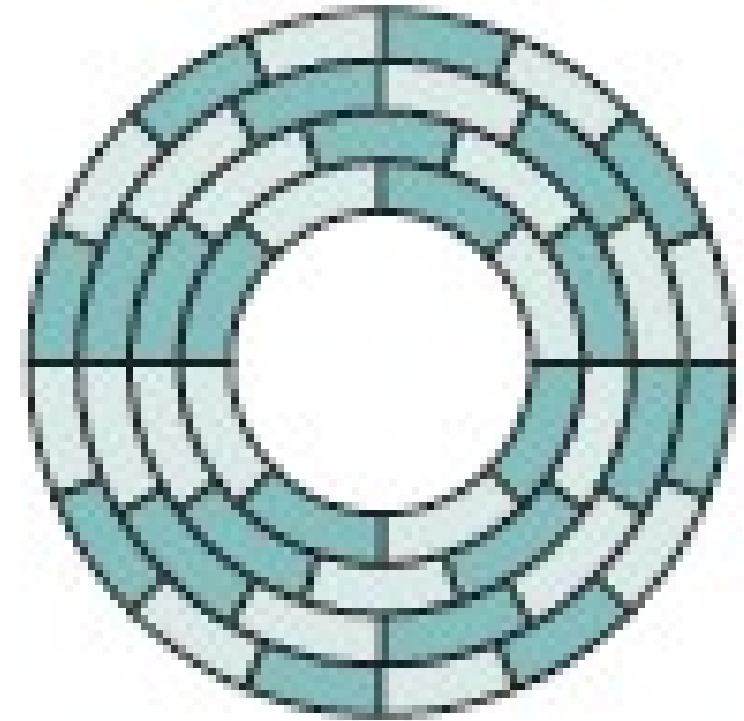


Because the **density**, in bits per linear inch, increases in moving from the outermost track to the innermost track, disk storage capacity in a straightforward CAV system is limited by the maximum recording density that can be achieved on the innermost track.

To increase density, modern hard disk systems use a technique known as **multiple zone recording**, in which the surface is divided into a number of concentric zones (16 is typical). Within a zone, the number of bits per track is constant. Zones farther from the center contain more bits (more sectors) than zones closer to the center.

This allows for greater overall storage capacity at the expense of somewhat more complex circuitry. As the disk head moves from one zone to another, the length (along the track) of individual bits changes, causing a change in the timing for reads and writes.

Figure suggests the nature of multiple zone recording; in this illustration, each zone is only a single track wide.

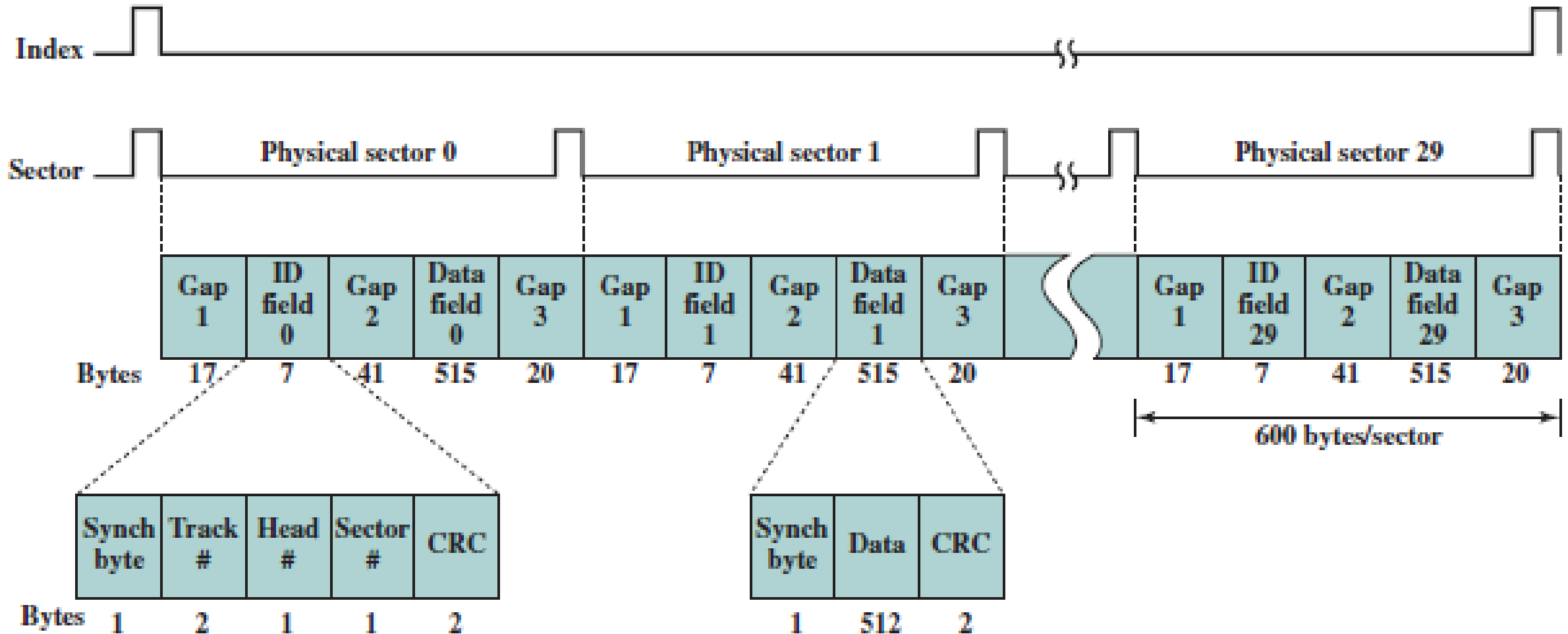


Some means is needed to locate sector positions within a track. Clearly, there must be some starting point on the track and a way of identifying the start and end of each sector.

These requirements are handled by means of control data recorded on the disk. Thus, the disk is formatted with some extra data used only by the disk drive and not accessible to the user.

An example of disk formatting is shown in figure below.

Winchester Disk Format (Seagate ST506)



In this case, each track contains 30 fixed-length sectors of 600 bytes each. Each sector holds 512 bytes of data plus control information useful to the disk controller.

The ID field is a unique identifier or address used to locate a particular sector. The SYNCH byte is a special bit pattern that delimits the beginning of the field.

The track number identifies a track on a surface. The head number identifies a head, because this disk has multiple surfaces. The ID and data fields each contain an error detecting code.

Physical Characteristics

Table lists the major characteristics that differentiate among the various types of magnetic disks.

Head Motion	Platters
Fixed head (one per track)	Single platter
Movable head (one per surface)	Multiple platter
Disk Portability	Head Mechanism
Nonremovable disk	Contact (floppy)
Removable disk	Fixed gap
	Aerodynamic gap (Winchester)
Sides	
Single sided	
Double sided	

First, the head may either be fixed or movable with respect to the radial direction of the platter. In a **fixed-head disk**, there is one read-write head per track. All of the heads are mounted on a rigid arm that extends across all tracks; such systems are rare today.

In a **movable-head disk**, there is only one read-write head. Again, the head is mounted on an arm. Because the head must be able to be positioned above any track, the arm can be extended or retracted for this purpose.

The disk itself is mounted in a disk drive, which consists of the arm, a spindle that rotates the disk, and the electronics needed for input and output of binary data. A **nonremovable disk** is permanently mounted in the disk drive; the hard disk in a personal computer is a nonremovable disk. A **removable disk** can be removed and replaced with another disk.

The advantage of the latter type is that unlimited amounts of data are available with a limited number of disk systems.

Furthermore, such a disk may be moved from one computer system to another. Floppy disks and ZIP cartridge disks are examples of removable disks.

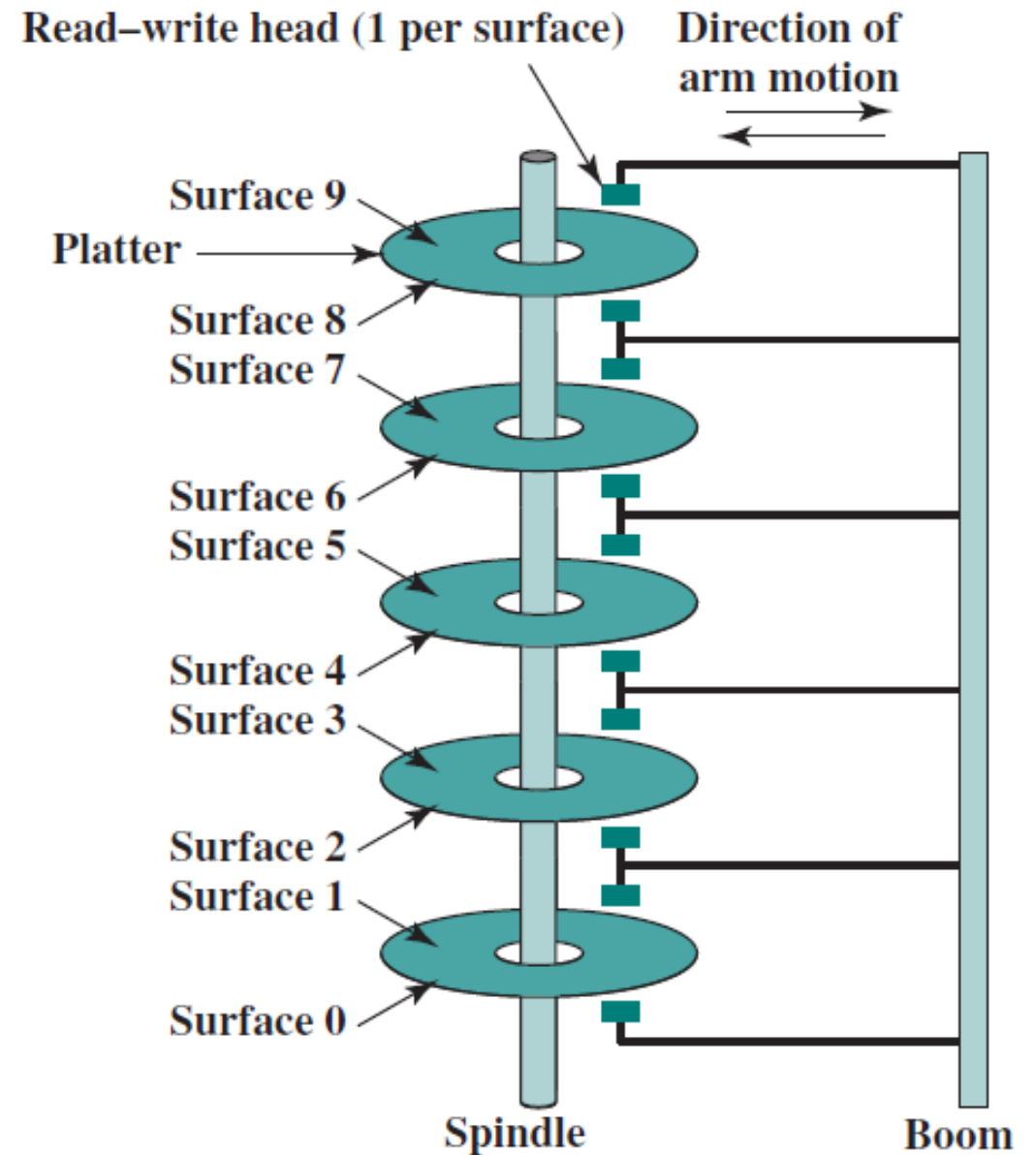
For most disks, the magnetizable coating is applied to both sides of the platter, which is then referred to as **double sided**.

Some less expensive disk systems use **single-sided** disks.

Some disk drives accommodate **multiple platters** stacked vertically a fraction of an inch apart. Multiple arms are provided (Figure).

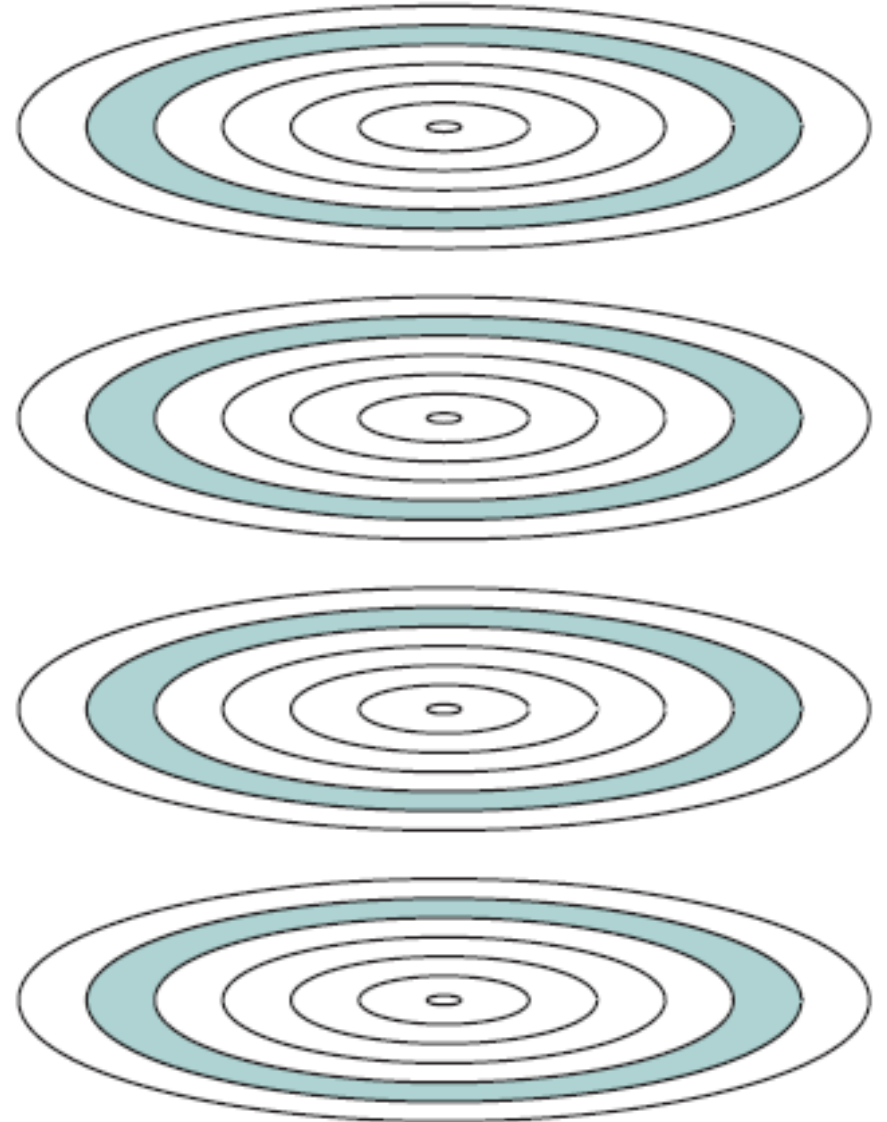
Multiple-platter disks employ a movable head, with one read-write head per platter surface.

All of the heads are mechanically fixed so that all are at the same distance from the center of the disk and move together. Thus, at any time, all of the heads are positioned over tracks that are of equal distance from the center of the disk.



The set of all the tracks in the same relative position on the platter is referred to as a **cylinder**.

For example, all of the shaded tracks in figure are part of one cylinder.



Finally, the head mechanism provides a classification of disks into three types. Traditionally, the read-write head has been positioned a fixed distance above the platter, allowing an air gap.

At the other extreme is a head mechanism that actually comes into physical contact with the medium during a read or write operation. This mechanism is used with the **floppy disk**, which is a small, flexible platter and the least expensive type of disk.

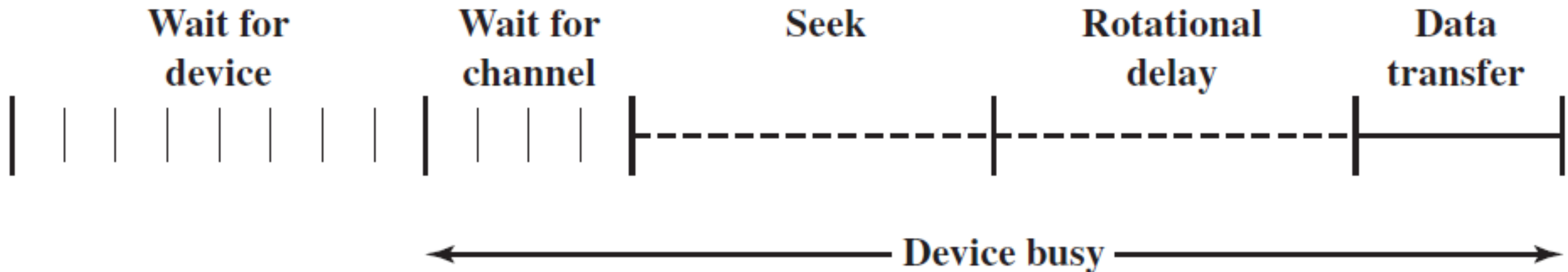
To understand the third type of disk, we need to comment on the relationship between data density and the size of the air gap. The head must generate or sense an electromagnetic field of sufficient magnitude to write and read properly. The narrower the head is, the closer it must be to the platter surface to function.

A narrower head means narrower tracks and therefore greater data density, which is desirable. However, the closer the head is to the disk, the greater the risk of error from impurities or imperfections.

To push the technology further, the Winchester disk was developed. Winchester heads are used in sealed drive assemblies that are almost free of contaminants. They are designed to operate closer to the disk's surface than conventional rigid disk heads, thus allowing greater data density.

Disk Performance Parameters

The actual details of disk I/O operation depend on the computer system, the operating system, and the nature of the I/O channel and disk controller hardware. A general timing diagram of disk I/O transfer is shown in figure.



When the disk drive is operating, the disk is rotating at constant speed. To read or write, the head must be positioned at the desired track and at the beginning of the desired sector on that track. Track selection involves moving the head in a movable-head system or electronically selecting one head on a fixed-head system.

On a movable-head system, the time it takes to position the head at the track is known as **seek time**. In either case, once the track is selected, the disk controller waits until the appropriate sector rotates to line up with the head.

The time it takes for the beginning of the sector to reach the head is known as **rotational delay**, or *rotational latency*.

The sum of the seek time, if any, and the rotational delay equals the **access time**, which is the time it takes to get into position to read or write.

Once the head is in position, the read or write operation is then performed as the sector moves under the head; this is the data transfer portion of the operation; the time required for the transfer is the **transfer time**.

In addition to the access time and transfer time, there are several queuing delays normally associated with a disk I/O operation. When a process issues an I/O request, it must first wait in a queue for the device to be available. At that time, the device is assigned to the process.

If the device shares a single I/O channel or a set of I/O channels with other disk drives, then there may be an additional wait for the channel to be available. At that point, the seek is performed to begin disk access.

In some high-end systems for servers, a technique known as rotational positional sensing (RPS) is used. This works as follows: When the seek command has been issued, the channel is released to handle other I/O operations.

When the seek is completed, the device determines when the data will rotate under the head. As that sector approaches the head, the device tries to reestablish the communication path back to the host.

If either the control unit or the channel is busy with another I/O, then the reconnection attempt fails and the device must rotate one whole revolution before it can attempt to reconnect, which is called an RPS miss.

SEEK TIME Seek time is the time required to move the disk arm to the required track. It turns out that this is a difficult quantity to pin down.

The seek time consists of two key components: the initial startup time, and the time taken to traverse the tracks that have to be crossed once the access arm is up to speed.

Unfortunately, the traversal time is not a linear function of the number of tracks, but includes a settling time (time after positioning the head over the target track until track identification is confirmed).

A typical average seek time on contemporary hard disks is under 10 ms.

ROTATIONAL DELAY Disks, other than floppy disks, rotate at speeds ranging from 3600 rpm (for handheld devices such as digital cameras) up to 20,000 rpm; at this latter speed, there is one revolution per 3 ms.

Thus, on the average, the rotational delay will be 1.5 ms.

TRANSFER TIME The transfer time to or from the disk depends on the rotation speed of the disk in the following fashion:

$$T = \frac{b}{rN}$$

where

T = transfer time

b = number of bytes to be transferred

N = number of bytes on a track

r = rotation speed, in revolutions per second

Thus the total average access time can be expressed as

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

where T_s is the average seek time. Note that on a zoned drive, the number of bytes per track is variable, complicating the calculation.

A TIMING COMPARISON

With the foregoing parameters defined, let us look at two different I/O operations that illustrate the danger of relying on average values.

Consider a disk with an advertised average seek time of 4 ms, rotation speed of 15,000 rpm, and 512-byte sectors with 500 sectors per track.

Suppose that we wish to read a file consisting of 2500 sectors for a total of 1.28 Mbytes. We would like to estimate the total time for the transfer.

First, let us assume that the file is stored as compactly as possible on the disk. That is, the file occupies all of the sectors on 5 adjacent tracks (5 tracks * 500 sectors/ track = 2500 sectors). This is known as *sequential organization*. Now, the time to read the first track is as follows:

Average seek	4 ms
Average rotational delay	2 ms
Read 500 sectors	4 ms
	<hr/>
	10 ms

Suppose that the remaining tracks can now be read with essentially no seek time. That is, the I/O operation can keep up with the flow from the disk. Then, at most, we need to deal with rotational delay for each succeeding track. Thus each successive track is read in $2 + 4 = 6$ ms. To read the entire file,

$$\text{Total time} = 10 + (4 * 6) = 34 \text{ ms} = 0.034 \text{ seconds}$$

Now let us calculate the time required to read the same data using random access rather than sequential access; that is, accesses to the sectors are distributed randomly over the disk. For each sector, we have

Average seek	4	ms
Rotational delay	2	ms
Read 1 sectors	0.008	ms
	<hr/>	
	6.008	ms

$$\text{Total time} = 2500 * 6.008 = 15,020 \text{ ms} = 15.02 \text{ seconds}$$

It is clear that the order in which sectors are read from the disk has a tremendous effect on I/O performance. In the case of file access in which multiple sectors are read or written, we have some control over the way in which sectors of data are deployed.

However, even in the case of a file access, in a multiprogramming environment, there will be I/O requests competing for the same disk. Thus, it is worthwhile to examine ways in which the performance of disk I/O can be improved over that achieved with purely random access to the disk.

RAID

With the use of multiple disks, there is a wide variety of ways in which the data can be organized and in which redundancy can be added to improve reliability.

This could make it difficult to develop database schemes that are usable on a number of platforms and operating systems. Fortunately, industry has agreed on a standardized scheme for multiple-disk database design, known as RAID (Redundant Array of Independent Disks).

The RAID scheme consists of seven levels, zero through six.

These levels do not imply a hierarchical relationship but designate different design architectures that share three common characteristics:

1. RAID is a set of physical disk drives viewed by the operating system as a single logical drive.
2. Data are distributed across the physical drives of an array in a scheme known as striping, described subsequently.
3. Redundant disk capacity is used to store parity information, which guarantees data recoverability in case of a disk failure.

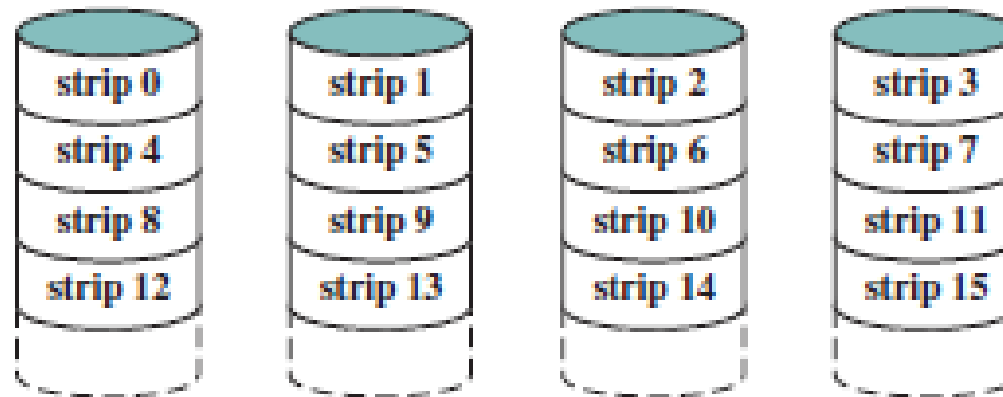
The details of the second and third characteristics differ for the different RAID levels. RAID 0 and RAID 1 do not support the third characteristic.

RAID Levels

Category	Level	Description	Disks Required	Data Availability	Large I/O Data Transfer Capacity	Small I/O Request Rate
Striping	0	Nonredundant	N	Lower than single disk	Very high	Very high for both read and write
Mirroring	1	Mirrored	$2N$	Higher than RAID 2, 3, 4, or 5; lower than RAID 6	Higher than single disk for read; similar to single disk for write	Up to twice that of a single disk for read; similar to single disk for write
Parallel access	2	Redundant via Hamming code	$N + m$	Much higher than single disk; comparable to RAID 3, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
	3	Bit-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
Independent access	4	Block-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 5	Similar to RAID 0 for read; significantly lower than single disk for write	Similar to RAID 0 for read; significantly lower than single disk for write
	5	Block-interleaved distributed parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 4	Similar to RAID 0 for read; lower than single disk for write	Similar to RAID 0 for read; generally lower than single disk for write
	6	Block-interleaved dual distributed parity	$N + 2$	Highest of all listed alternatives	Similar to RAID 0 for read; lower than RAID 5 for write	Similar to RAID 0 for read; significantly lower than RAID 5 for write

RAID Level 0

RAID level 0 is not a true member of the RAID family because it does not include redundancy to improve performance. However, there are a few applications, such as some on supercomputers in which performance and capacity are primary concerns and low cost is more important than improved reliability.



Nonredundant

For RAID 0, the user and system data are distributed across all of the disks in the array. This has a notable advantage over the use of a single large disk:

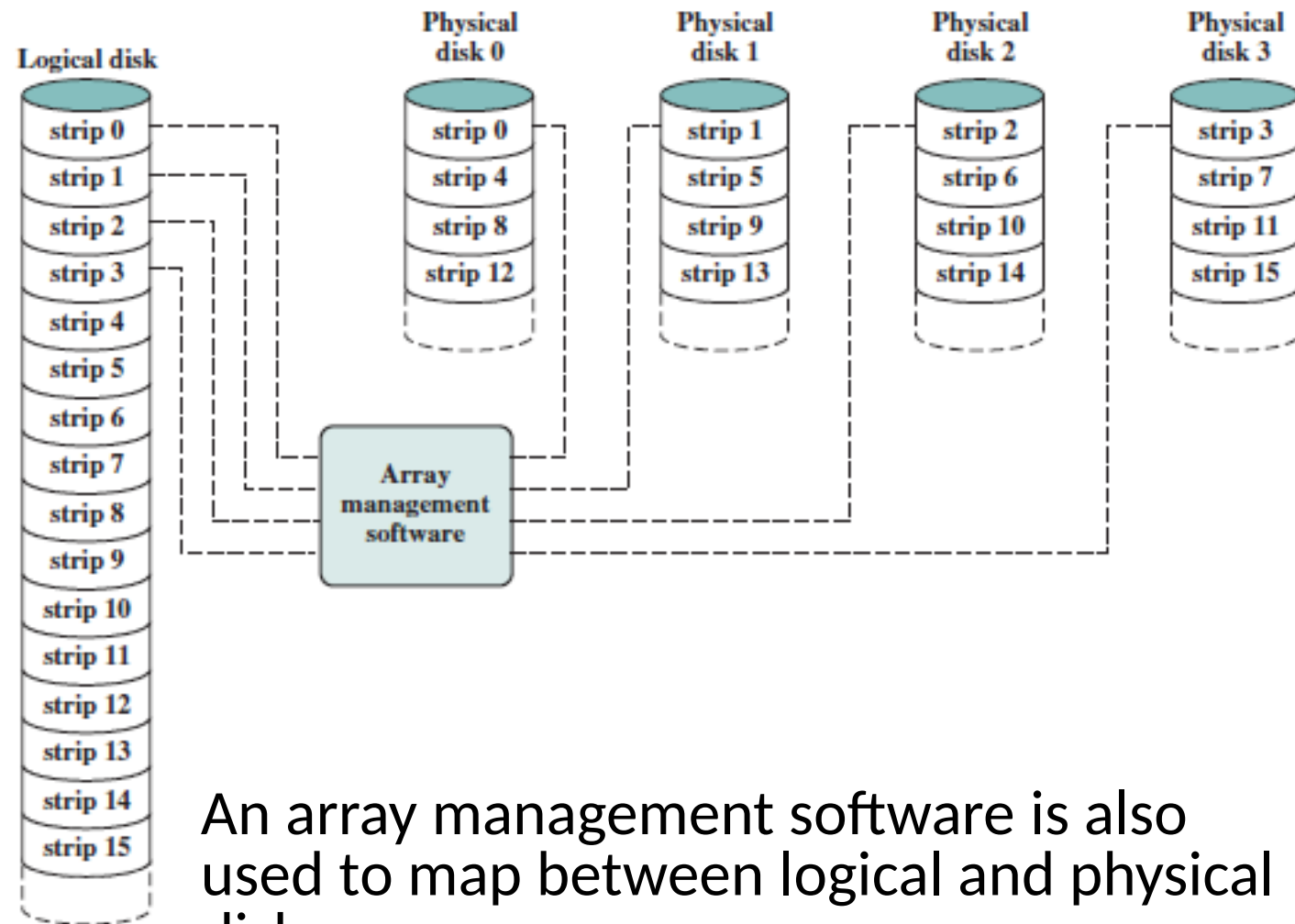
If two -different I/O requests are pending for two different blocks of data, then there is a good chance that the requested blocks are on different disks.

Thus, the two requests can be issued in parallel, reducing the I/O queuing time. But RAID 0, as with all of the RAID levels, goes further than simply distributing the data across a disk array: The data are *striped* across the available disks.

All of the user and system data are viewed as being stored on a logical disk. The logical disk is divided into strips; these strips may be physical blocks, sectors, or some other unit. The strips are mapped round robin to consecutive physical disks in the RAID array.

A set of logically consecutive strips that maps exactly one strip to each array member is referred to as a **stripe**.

In an n -disk array, the first n logical strips are physically stored as the first strip on each of the n disks, forming the first stripe; the second n strips are distributed as the second strips on each disk; and so on.



An array management software is also used to map between logical and physical disk space.

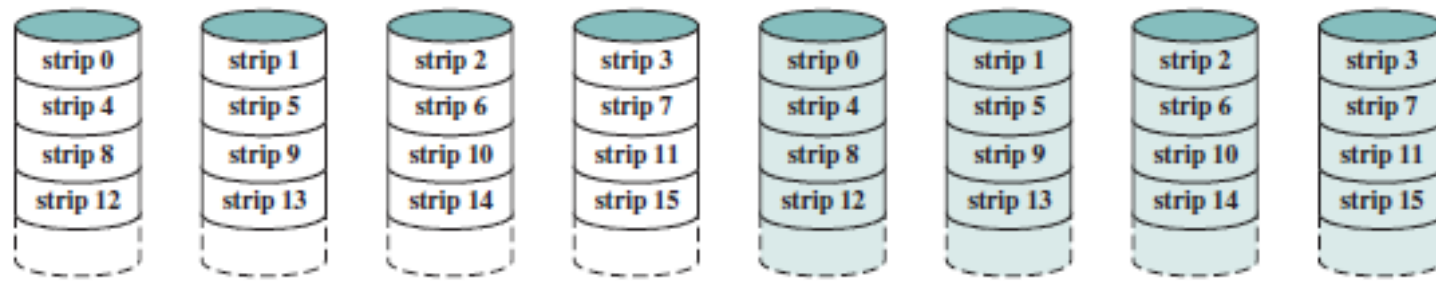
This software may execute either in the disk subsystem or in a host computer.

The advantage of this layout is that if a single I/O request consists of multiple logically contiguous strips, then up to n strips for that request can be handled in parallel, greatly reducing the I/O transfer time.

RAID Level 1

RAID 1 differs from RAID levels 2 through 6 in the way in which redundancy is achieved. In these other RAID schemes, some form of parity calculation is used to introduce redundancy, whereas in RAID 1, redundancy is achieved by the simple expedient of duplicating all the data.

Data striping is used, as in RAID 0. But in this case, each logical strip is mapped to two separate physical disks so that every disk in the array has a mirror disk that contains the same data. RAID 1 can also be implemented without data striping, though this is less common.



Mirrored

There are a number of positive aspects to the RAID 1 organization:

1. A read request can be serviced by either of the two disks that contains the requested data, whichever one involves the minimum seek time plus rotational latency.
2. A write request requires that both corresponding strips be updated, but this can be done in parallel. Thus, the write performance is dictated by the slower of the two writes (i.e., the one that involves the larger seek time plus rotational latency). However, there is no “write penalty” with RAID 1. RAID levels 2 through 6 involve the use of parity bits. Therefore, when a single strip is updated, the array management software must first compute and update the parity bits as well as updating the actual strip in question.
3. Recovery from a failure is simple. When a drive fails, the data may still be accessed from the second drive.

The principal disadvantage of RAID 1 is the cost; it requires twice the disk space of the logical disk that it supports. Because of that, a RAID 1 configuration is likely to be limited to drives that store system software and data and other highly critical files. In these cases, RAID 1 provides real-time copy of all data so that in the event of a disk failure, all of the critical data are still immediately available.

In a transaction-oriented environment, RAID 1 can achieve high I/O request rates if the bulk of the requests are reads. In this situation, the performance of RAID 1 can approach double of that of RAID 0. However, if a substantial fraction of the I/O requests are write requests, then there may be no significant performance gain over RAID 0.

RAID Level 2

RAID levels 2 and 3 make use of a parallel access technique. In a parallel access array, all member disks participate in the execution of every I/O request. Typically, the spindles of the individual drives are synchronized so that each disk head is in the same position on each disk at any given time.



Redundancy using
Hamming code

As in the other RAID schemes, data striping is used. In the case of RAID 2 and 3, the strips are very small, often as small as a single byte or word.

With RAID 2, an error-correcting code is calculated across corresponding bits on each data disk, and the bits of the code are stored in the corresponding bit positions on multiple parity disks. Typically, a Hamming code is used, which is able to correct single-bit errors and detect double-bit errors.

Although RAID 2 requires fewer disks than RAID 1, it is still rather costly. The number of redundant disks is proportional to the log of the number of data disks.

On a single read, all disks are simultaneously accessed. The requested data and the associated error-correcting code are delivered to the array controller. If there is a single-bit error, the controller can recognize and correct the error instantly, so that the read access time is not slowed.

On a single write, all data disks and parity disks must be accessed for the write operation.

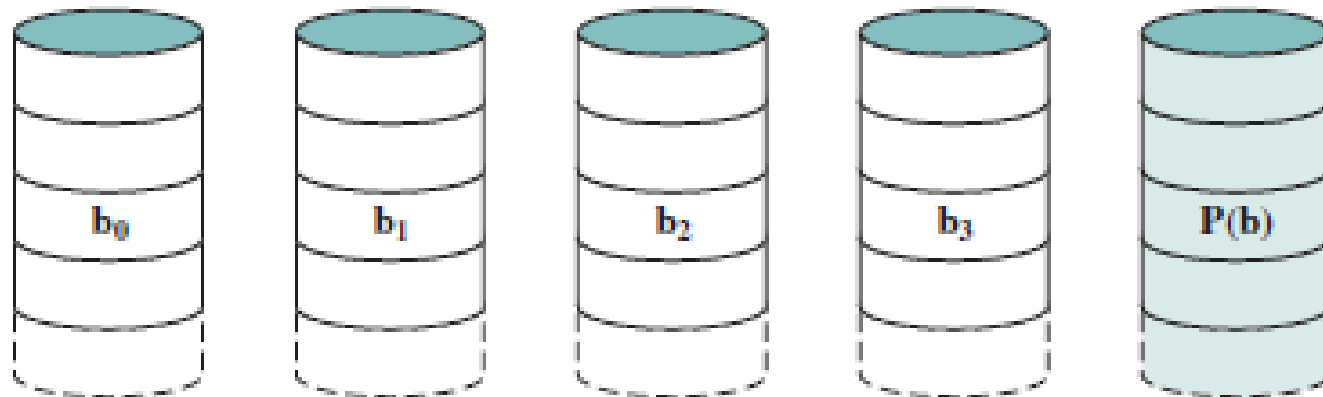
RAID 2 would only be an effective choice in an environment in which many disk errors occur.

Given the high reliability of individual disks and disk drives, RAID 2 is overkill and is not implemented.

RAID Level 3

RAID 3 is organized in a similar fashion to RAID 2. The difference is that RAID 3 requires only a single redundant disk, no matter how large the disk array.

RAID 3 employs parallel access, with data distributed in small strips. Instead of an error-correcting code, a simple parity bit is computed for the set of individual bits in the same position on all of the data disks.



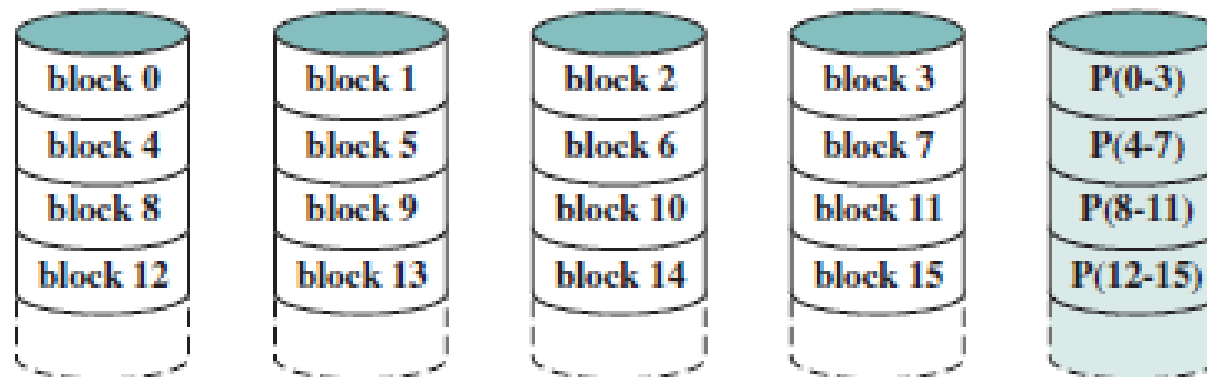
Bit-interleaved
parity

Because data are striped in very small strips, RAID 3 can achieve very high data transfer rates. Any I/O request will involve the parallel transfer of data from all of the data disks. For large transfers, the performance improvement is especially noticeable.

On the other hand, only one I/O request can be executed at a time. Thus, in a transaction-oriented environment, performance suffers.

RAID Level 4

RAID levels 4 through 6 make use of an independent access technique. In an independent access array, each member disk operates independently, so that separate I/O requests can be satisfied in parallel. Because of this, independent access arrays are more suitable for applications that require high I/O request rates and are relatively less suited for applications that require high data transfer rates.



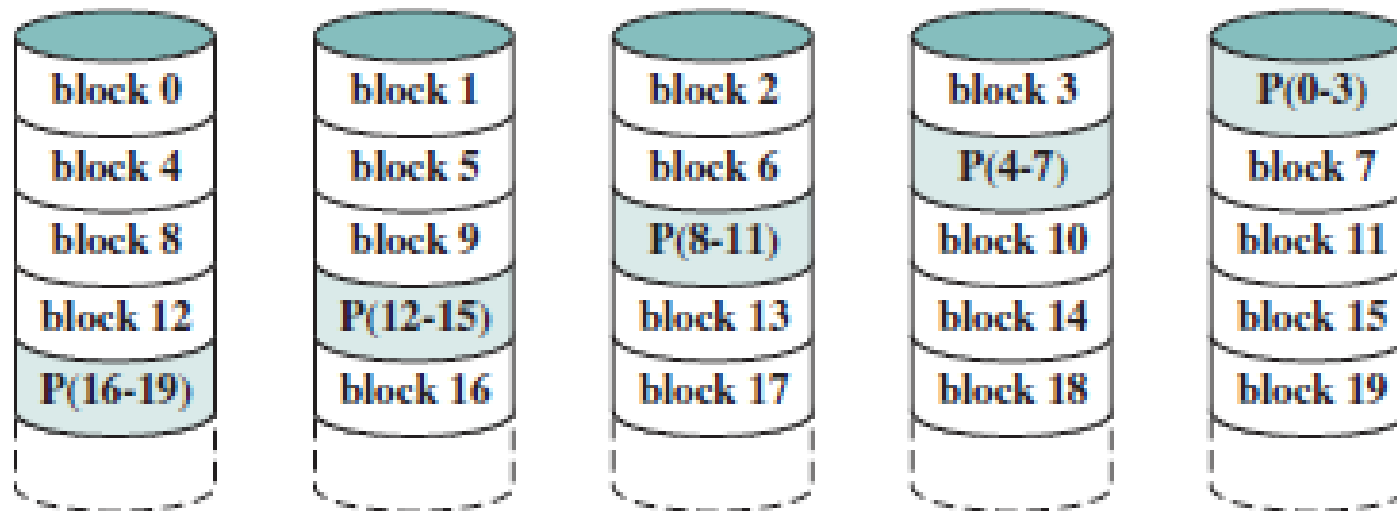
Block-level
parity

In the case of a larger size I/O write that involves strips on all disk drives, parity is easily computed by calculation using only the new data bits. Thus, the parity drive can be updated in parallel with the data drives and there are no extra reads or writes.

In any case, every write operation must involve the parity disk, which therefore can become a bottleneck.

RAID Level 5

RAID 5 is organized in a similar fashion to RAID 4. The difference is that RAID 5 distributes the parity strips across all disks. A typical allocation is a round-robin scheme.

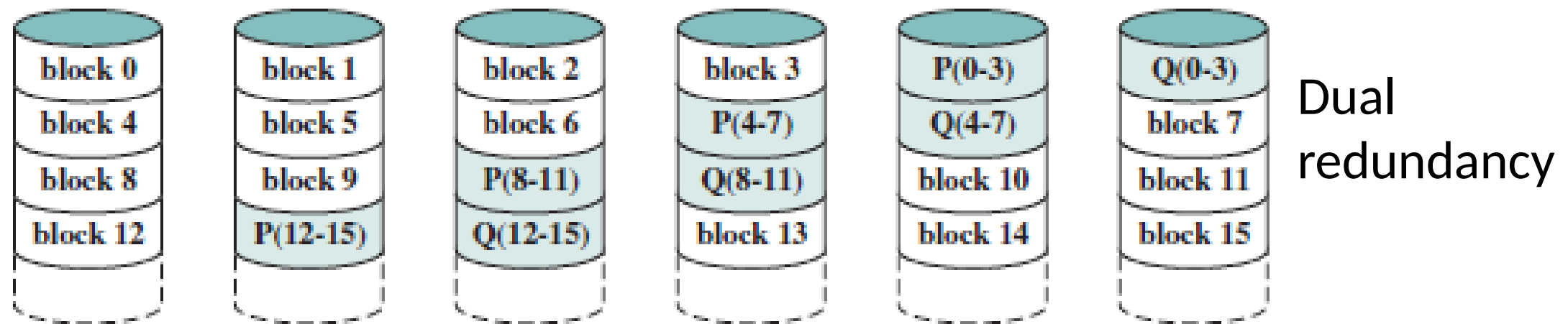


Block-level
distributed parity

For an n -disk array, the parity strip is on a different disk for the next n stripes, and the pattern then repeats. The distribution of parity strips across all drives avoids the potential I/O bottle-neck found in RAID 4.

RAID Level 6

RAID 6 was introduced by the Berkeley researchers. In the RAID 6 scheme, two different parity calculations are carried out and stored in separate blocks on different disks. Thus, a RAID 6 array whose user data require N disks consists of $N + 2$ disks.



P and Q are two different data check algorithms. One of the two is the exclusive-OR calculation used in RAID 4 and 5. But the other is an independent data check algorithm. This makes it possible to regenerate data even if two disks containing user data fail.

The advantage of RAID 6 is that it provides extremely high data availability. Three disks would have to fail within the MTTR (mean time to repair) interval to cause data to be lost.

On the other hand, RAID 6 incurs a substantial write penalty, because each write affects two parity blocks. Performance benchmarks show a RAID 6 controller can suffer more than a 30% drop in overall write performance compared with a RAID 5 implementation. RAID 5 and RAID 6 read performance is comparable.

SOLID STATE DRIVES

One of the most significant developments in computer architecture in recent years is the increasing use of solid state drives (SSDs) to complement or even replace **hard disk drives (HDDs)**, both as internal and external secondary memory.

The term *solid state* refers to electronic circuitry built with semiconductors. A **solid state drive** is a memory device made with solid state components that can be used as a replacement to a hard disk drive. The SSDs now on the market and coming on line use a type of semiconductor memory referred to as flash memory.

Flash Memory

Flash memory is a type of semiconductor memory that has been around for a number of years and is used in many consumer electronic products, including smart phones, GPS devices, MP3 players, digital cameras, and USB devices.

In recent years, the cost and performance of flash memory has evolved to the point where it is feasible to use flash memory drives to replace HDDs.

There are two distinctive types of flash memory, designated as NOR and NAND. In **NOR flash memory**, the basic unit of access is a bit, and the logical organization resembles a NOR logic device. For **NAND flash memory**, the basic unit is 16 or 32 bits, and the logical organization resembles NAND devices.

NOR flash memory provides high-speed random access. It can read and write data to specific locations, and can reference and retrieve a single byte. NOR flash memory is used to store cell phone operating system code and on Windows computers for the BIOS program that runs at startup.

NAND reads and writes in small blocks. It is used in USB flash drives, memory cards (in digital cameras, MP3 players, etc.), and in SSDs. NAND provides higher bit density than NOR and greater write speed. NAND flash does not provide a random-access external address bus so the data must be read on a blockwise basis (also known as page access), where each block holds hundreds to thousands of bits.

SSD Compared to HDD

As the cost of flash-based SSDs has dropped and the performance and bit density increased, SSDs have become increasingly competitive with HDDs. SSDs have the following advantages over HDDs:

- **High-performance input/output operations per second (IOPS):** Significantly increases performance I/O subsystems.
- **Durability:** Less susceptible to physical shock and vibration.
- **Longer lifespan:** SSDs are not susceptible to mechanical wear.
- **Lower power consumption:** SSDs use as little as 2.1 watts of power per drive, considerably less than comparable-size HDDs.
- **Quieter and cooler running capabilities:** Less floor space required, lower energy costs, and a greener enterprise.
- **Lower access times and latency rates:** Over 10 times faster than the spinning disks in an HDD.

Currently, HDDs enjoy a cost per bit advantage and a capacity advantage, but these differences are shrinking.

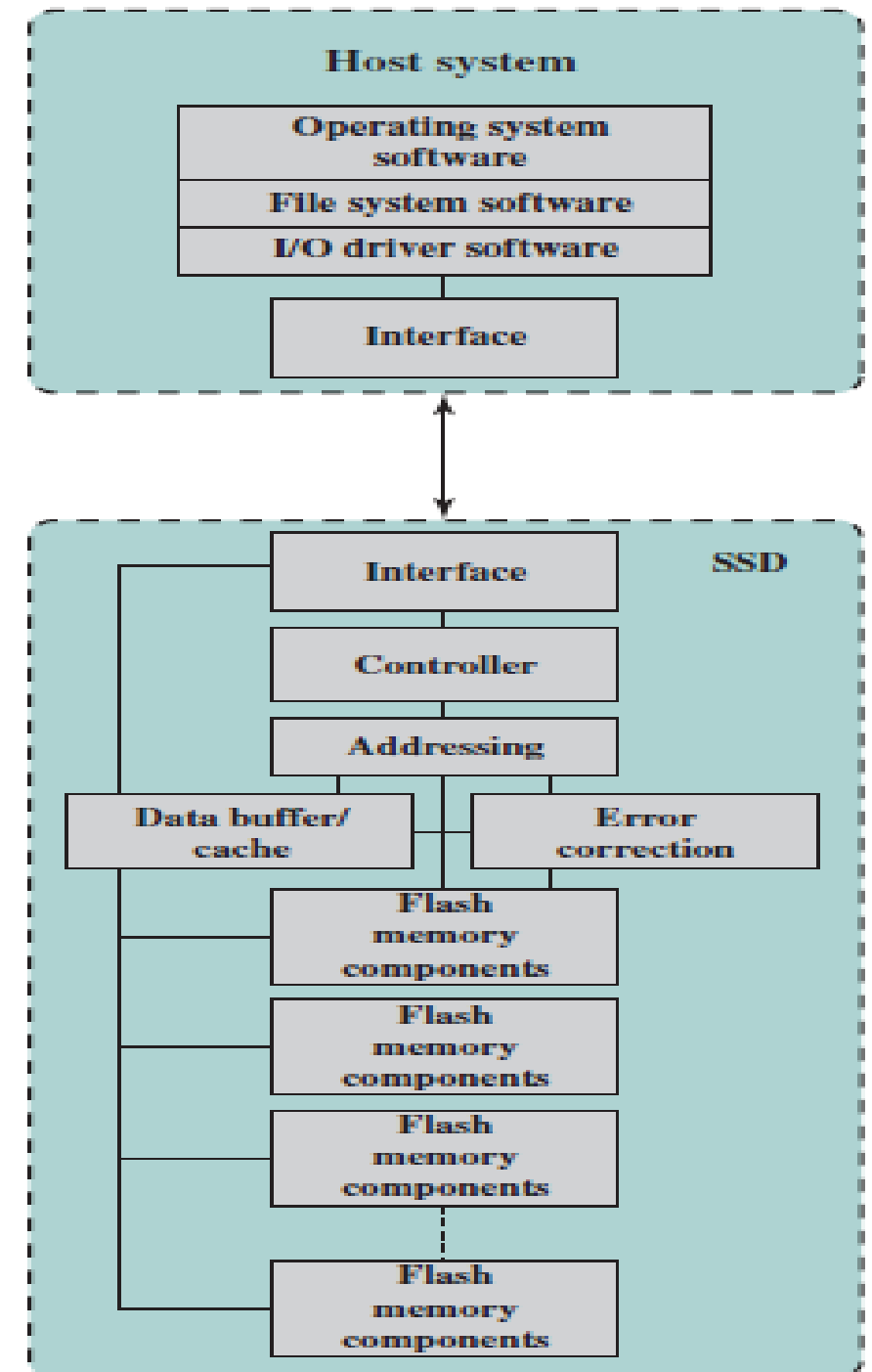
SSD Organization

Figure illustrates a general view of the common architectural system component associated with any SSD system.

On the host system, the operating system invokes file system software to access data on the disk.

The file system, in turn, invokes I/O driver software. The I/O driver software provides host access to the particular SSD product.

The interface component in Figure refers to the physical and electrical interface between the host processor and the SSD peripheral device. If the device is an internal hard drive, a common interface is PCIe. For external devices, one common interface is USB.



In addition to the interface to the host system, the SSD contains the following components:

- **Controller:** Provides SSD device level interfacing and firmware execution.
- **Addressing:** Logic that performs the selection function across the flash memory components.
- **Data buffer/cache:** High speed RAM memory components used for speed matching and to increased data throughput.
- **Error correction:** Logic for error detection and correction.
- **Flash memory components:** Individual NAND flash chips.

OPTICAL MEMORY

Compact Disk Read Only Memory (CD-ROM)

In 1983, one of the most successful consumer products of all time was introduced: the compact disk (CD) digital audio system. The CD is a nonerasable disk that can store more than 60 minutes of audio information on one side.

Both the audio CD and the **CD-ROM** (compact disk read-only memory) share a similar technology. The main difference is that CD-ROM players are more rugged and have error correction devices to ensure that data are properly transferred from disk to computer.

The disk is formed from a resin, such as polycarbonate. Digitally recorded information (either music or computer data) is imprinted as a series of microscopic pits on the surface of the polycarbonate.

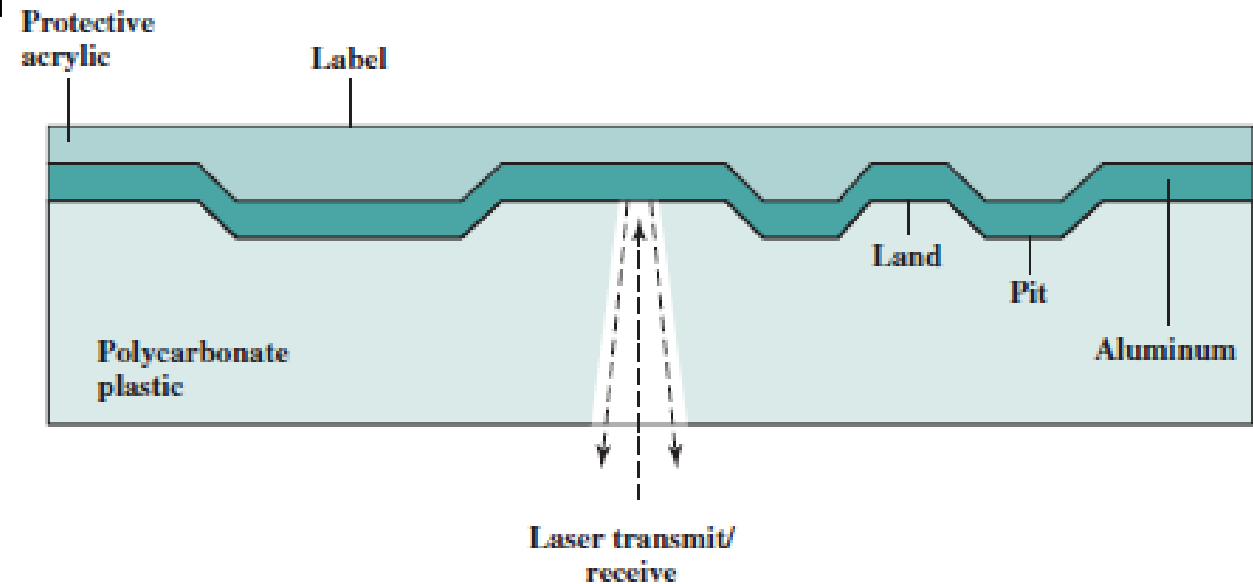
This is done, first of all, with a finely focused, high intensity laser to create a master disk. The master is used, in turn, to make a die to stamp out copies onto polycarbonate. The pitted surface is then coated with a highly reflective surface, usually aluminum or gold. This shiny surface is protected against dust and scratches by a top coat of clear acrylic.

Finally, a label can be silkscreened onto the acrylic.

Information is retrieved from a CD or CD-ROM by a low-powered laser housed in an optical-disk player, or drive unit. The laser shines through the clear polycarbonate while a motor spins the disk past it. The intensity of the reflected light of the laser changes as it encounters a **pit**. Specifically, if the laser beam falls on a pit, which has a somewhat rough surface, the light scatters and a low intensity is reflected back to the source.

The areas between pits are called **lands**. A land is a smooth surface, which reflects back at higher intensity. The change between pits and lands is detected by a photosensor and converted into a digital signal. The sensor tests the surface at regular intervals.

The beginning or end of a pit represents a 1; when no change in elevation occurs between intervals, a 0 is recorded.



Recall that on a magnetic disk, information is recorded in concentric tracks. With the simplest constant angular velocity (CAV) system, the number of bits per track is constant. An increase in density is achieved with **multiple zoned recording**, in which the surface is divided into a number of zones, with zones farther from the center containing more bits than zones closer to the center. Although this technique increases capacity, it is still not optimal.

To achieve greater capacity, CDs and CD-ROMs do not organize information on concentric tracks. Instead, the disk contains a single spiral track, beginning near the center and spiraling out to the outer edge of the disk. Sectors near the outside of the disk are the same length as those near the inside.

Thus, information is packed evenly across the disk in segments of the same size and these are scanned at the same rate by rotating the disk at a variable speed. The pits are then read by the laser at a **constant linear velocity (CLV)**. The disk rotates more slowly for accesses near the outer edge than for those near the center. Thus, the capacity of a track and the rotational delay both increase for positions nearer the outer edge of the disk.

Digital Versatile Disk (DVD)

Vast volumes of data can be crammed onto the disk, currently seven times as much as a CD-ROM. With DVD's huge storage capacity and vivid quality, PC games have become more realistic and educational software incorporates more video.

Following in the wake of these developments has been a new crest of traffic over the Internet and corporate intranets, as this material is incorporated into Web sites.

The DVD's greater capacity is due to three differences from CDs (680MB):

1. Bits are packed more closely on a DVD. The spacing between loops of a spiral on a CD is $1.6\text{ }\mu\text{m}$ and the minimum distance between pits along the spiral is $0.834\text{ }\mu\text{m}$. The DVD uses a laser with shorter wavelength and achieves a loop spacing of $0.74\text{ }\mu\text{m}$ and a minimum distance between pits of $0.4\text{ }\mu\text{m}$. The result of these two improvements is about a seven-fold increase in capacity, to about 4.7 GB.
2. The DVD employs a second layer of pits and lands on top of the first layer. A dual-layer DVD has a semireflective layer on top of the reflective layer, and by adjusting focus, the lasers in DVD drives can read each layer separately. This technique almost doubles the capacity of the disk, to about 8.5 GB. The lower reflectivity of the second layer limits its storage capacity so that a full doubling is not achieved.
3. The **DVD-ROM** can be two sided, whereas data are recorded on only one side of a CD. This brings total capacity up to 17 GB.