Deliverable: Magnetic discs

Delivery deadline: 29 April 2022

Goals

- To understand the working principles of hard disks
- To be able to determine the capacity and performance of a hard disk from its geometry and rotation speed

1 Introduction

Magnetic devices

A magnetic dipole can be oriented at will using electrical current, and it needs no further energy supply to preserve such orientation. This fact has inspired the design of several information storage devices, both analog and digital. In digital computers, magnetic recording has been fundamental for several long-lasting, widely used storage technologies, such as magnetic core memories and many other devices that use a ferromagnetic layer on top of different mechanical supports such as drums, open reel tapes, streamers, cartridges, cassettes, floppy and hard disks.

A supporting surface (or *medium*) is coated with a layer of ferromagnetic material that contains a mosaic of tiny magnetic domains. These domains behave as independent dipoles that can be oriented using electrical current (see figure 1). The particular orientation of the magnetic domains encodes binary data. The *head* is a device located close to the magnetic substrate, and it is in charge of causing or detecting the orientation of the magnetic domains, i.e., to write or read data, respectively.

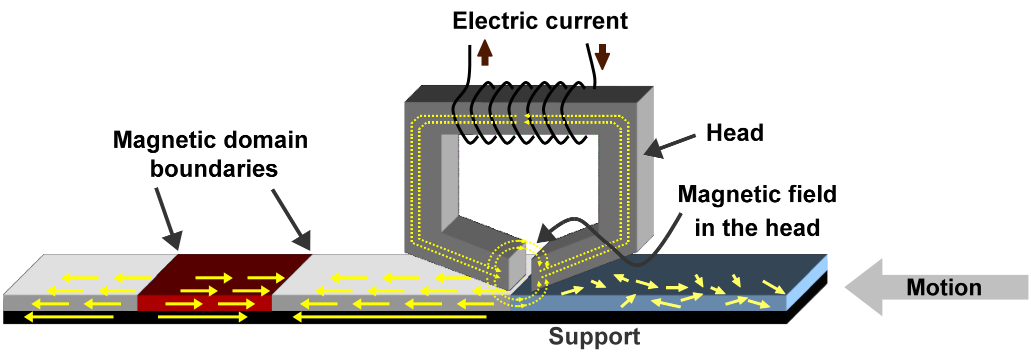

Figure 1: Mechanism for magnetic storage. The ferromagnetic layer sits on a supporting surface, either rigid or flexible. The head can create a magnetic field from electrical current or, conversely, produce electrical current when the magnetic field changes. When the medium is moving, a write head can write binary data by orienting the magnetic domains in one or the opposite direction, depending on the direction of the electrical current. Conversely, a read head can detect changes in the orientation of magnetic domains and recover the previously recorded data in the form of electrical current of one or another sign.

Depending on the type of supporting surface, magnetic storage devices enable direct or sequential access. Historically, magnetic memories have played different roles in the memory hierarchy. They have been used as main memory (ferrite core memories and magnetic drums used to store running programs); as secondary memory (floppy, and especially, hard disks support a file system, boot OS code and the swap area of virtual memory); and also off-line memory (disks and tapes have been used for backup copies, general data change and software distribution).

Hard disks are (to date) the only survivors of magnetic storage technologies. They are mainly used as secondary storage. Magnetic storage is slowly but steadily being replaced, and a future without magnetic disks can be foreseen. Flash memories will play the same role at higher performance, but their price per bit (compared to magnetic disks) is still postponing the replacement.

Capacity units

Following the IEC convention, (see Wikipedia article) memory capacity can be measured with both decimal and binary prefixes (see Figure 2). The IEC standard wants to eliminate ambiguities when expressing the capacity and bandwidth of memory devices.

Decimal prefixes

symbol	name	value
KB	kilo	10^3
MB	mega	10^6
GB	giga	10^9
TB	tera	10^{12}

Binary prefixes

symbol	name	value
KiB	kibi	2^{10}
MiB	mebi	2^{20}
GiB	gibi	2^{30}
TiB	tebi	2^{40}

Figure 2: Prefixes used for expressing memory capacity, according to the IEC convention.

Despite the existence of the IEC convention, it has not been fully adopted. For example, RAM manufacturers traditionally use decimal prefixes, but they are interpreted as binary: a RAM module marked "1 GB" has 2^{30} bytes rather than 10^9.

It is convenient to express the capacity of hard disks and the size of stored files, using decimal prefixes coherently. The Windows OS, for example, does not follow this convention (see Figure 3) and uses decimal prefixes with a binary meaning. Linux and Mac OS X, on the other hand (see Figure 4), allow users to choose one particular convention among both possibilities.

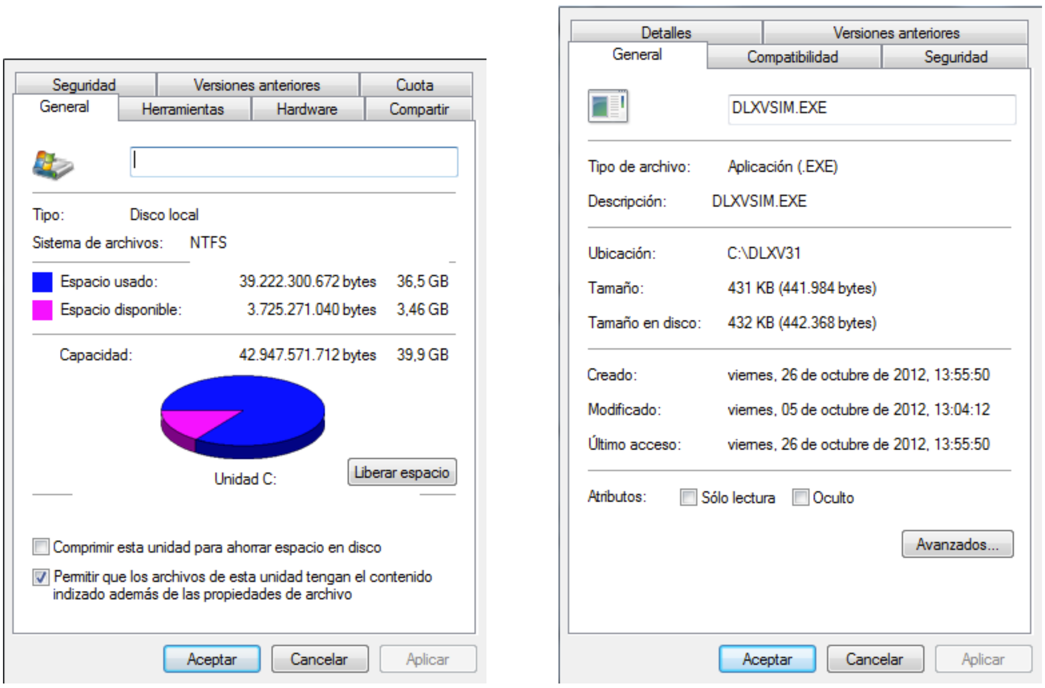

Figure 3: The *Properties* window in Windows shows disk and files capacities without any prefix and with decimal prefix with binary interpretation. Note that 39.9 GB $\simeq 42,947,571,712$ B $/2^{30}$.

```
...$ df
Filesystem            512-blocks       Used Available Capacity  Mounted on
/dev/disk0s2           975093952 929132584  45449368     96%    /
devfs                        376       376         0   100%    /dev
localhost:/kUcYdqkvr   975093952 975093952         0   100%    /Volumes/MobileBackups
...$ df -h
Filesystem              Size   Used  Avail Capacity  Mounted on
/dev/disk0s2           465Gi  443Gi  22Gil    96%    /
devfs                  188Ki  188Ki    0Bi   100%    /dev
localhost:/kUcYdqkvr   465Gi  465Gi    0Bi   100%    /Volumes/MobileBackups
...$ df -H
Filesystem             Size   Used  Avail Capacity  Mounted on
/dev/disk0s2           499G   476G    23G    96%    /
devfs                  193k   193k     0B   100%    /dev
localhost:/kUcYdqkvr   499G   499G     0B   100%    /Volumes/MobileBackups
...$
```

Figure 4: Linux and MacOS X follow the IEC convention when listing the capacities of the installed disks. This figure shows the response of the command `df` in Mac OS X. By default, the command `df` shows capacities in number of 512-byte blocks; the `-h` modifier specifies number of bytes with decimal prefixes; the `-H` modifier selects number of bytes with binary prefixes. In Linux, the block size used in the listing is 1024 bytes and the effect of the `-h` and `-H` modifiers is the opposite.

☞ **EXERCISE 1** A file takes 2,147,483,648 bytes. Give its size using both decimal and binary prefixes. Make sure you apply the recommendations given in Annex *Style matters* at the end of the task instructions.

2 Organisation of hard disks

Current hard disks are of a type named *Winchester* and were invented by engineers at IBM. Their design relies on the physics of air, which makes a rotating disk keep the magnetic head suspended, floating at the proper distance from the medium. When the disk rotation is stopped (e.g. disk suspension for energy saving), the electronics of the hard disk makes the head *park* in the *landing zone* of the disk.

A typical hard disk (see Figure 5) is formed by one or more parallel (*platters*) that spin together around their axis. The set of n platters offers up to $2n$ useful surfaces or (*sides*). A mechanical arm holds and moves as many heads as usable sides on the disk.

A motor operates the arm. By moving the arm, all the heads are positioned at a particular point of the disk surface. Each positioning point defines a disk *track* on each surface. The set of tracks accessible at every positioning point of the arm is called a *cylinder*.

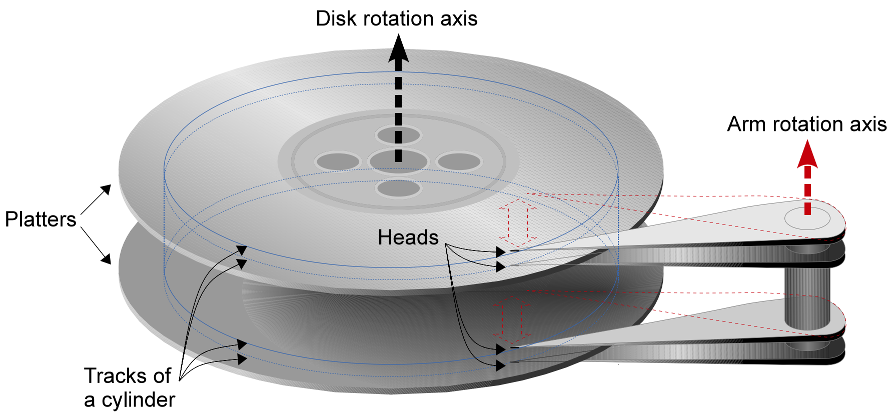

Figure 5: Scheme of a two-platter hard disk with four magnetic heads. When the arm positions the four heads at a particular point, the four accessible tracks form a *cylinder*.

Each track is further subdivided into many *sectors*. A sector is the smallest addressable information unit for reading and writing a disk. Each sector contains a data structure that includes separation marks between neighbouring sectors, a sector identifier, the data contents, and a redundant code that enables error detection. For further information about the structure of sectors, you can find the details in books such as Scott Mueller's *"Upgrading and Repairing PCs"*, available at the UPV library and in Google books. Since the early hard-disk designs, the data contents of a single sector is typically 512 bytes. Operating systems, however, tend to define *logical* sectors formed by 2^n physical sectors, thereby taking advantage of spatial locality.

Sectors can be identified by using their CHS coordinates, formed by a **C**ylinder, a **H**ead, and a **S**ector number within the track. This numbering convention was more visible in the older days. Today, the electronics of hard disks hide this information and provide a linear numbering called LBA (*Logical Block Address*), with sectors numbered from 0 onwards. The disk unit does the translation to CHS coordinates internally and transparently. Note that only operating system device drivers need to address individual sectors: user programs do not deal with such low-level details.

3 Hard disk capacity

Physical dimensions strongly affect the capacity of a disk, given that they place a bound on the maximum number of platters and the usable surface of each platter side. Manufacturers use the *"form factor"*, in inches, that refers to the needed space in the computer's chassis. Typical form factors are 3.5″ for desktop computers and 2.5″ for laptops. You can check these details in this Wikipedia article.

Besides the size of the hard disk drive unit, we need to consider the minimum space required to store a single bit. This depends on the characteristics of the medium, the head size, and the distance between the head and the surface of the medium. The inverse of this minimum space is the maximum density, given in bits per surface unit.

Two factors determine how much information can be stored in a disk: the **linear track density** and the number of sectors per track. Track density is expressed in tracks per inch of surface radius (or *tpi*). It is limited by the head size and the precision of the electronics controlling the arm. Currently, track widths are tens of nanometers, and track density reaches hundreds of thousand tracks per inch. The total number of cylinders in a disk is thus determined by the diameters corresponding to the outermost track (*OD*, *Outermost Diameter*) and innermost track (*ID*, *Innermost Diameter*), and the linear track density.

☞ **EXERCISE 2** A given hard disk has 6 sides and a linear track density of 52000 tpi. The innermost diameter is 1.2 ″ and the outermost diameter is 2.5 ″.

1. What is the amount of useful surface in the disk? Give the result in square inches (in^2).
2. How many cylinders and tracks does the disk contain?

The **number of sectors per track** depends on how sectors are distributed across tracks. Sector capacity is a constant of the disk. It affects several design levels of the disk drive, from the control electronics to the management of the file system by the operating system. There is a minimum surface of the medium required per sector for a given technology. Knowing the distribution of tracks and sectors in the medium, we can calculate an *areal density* of bits, which currently stands at some 500 Gb per square inch.

Three strategies exist for distributing sectors across the circular tracks (see figures 6 and 7).

CAV (*Constant Angular Velocity*): The number of sectors per track is constant throughout the platter surface, and the disk rotates at a constant angular velocity (measured in *rpm*). Sectors in the outer cylinders are larger than those in the innermost cylinders. Still, it takes the same amount of time for them to pass below the heads, precisely because the disk rotates at a constant angular velocity. Thus the innermost cylinders are denser, and they place an upper bound to the maximum number of sectors per track.

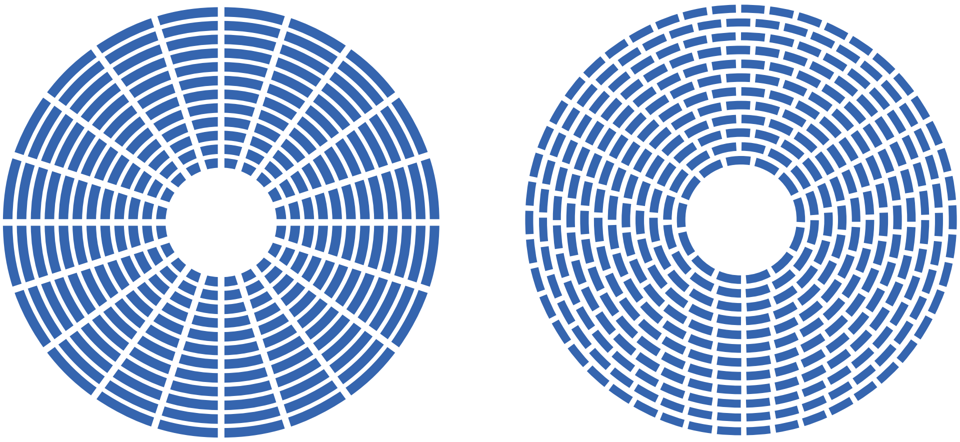

Figure 6: Two distribution strategies for sectors on each side in a 12-cylinder disk. Left, a CAV organisation with 20 sectors per track and 240 sectors per side. In the outermost track, sectors are three times larger than they are in the innermost track. Right, a CLV distribution with 46 sectors on the outermost track. Each subsequent track holds three fewer sectors than the outer neighbour. The innermost track contains 13 sectors. There are a total of 354 sectors per side.

Let H be the number of heads, C the number of cylinders, S the number of sectors per track, and B the sector capacity. The total capacity of a disk using CAV can thus be calculated as:

$$Capacity\text{ (CAV)} = H \times C \times S \times B$$

CAV was most used with the now obsolete floppy disks. For example, DS, DD (*Double Sided*, *Double Density*) floppy disks contained two sides and 80 cylinders and their nominal capacity was marked as "720 KB". The old MS-DOS operating system formatted these disks with nine 512-byte sectors per track. Hence the resulting capacity was:

$$\begin{aligned} Capacity\text{ (DS,DD)} &= \text{2 sides} \times \text{80 cylinders} \times \text{9 sectors/track} \times \text{512 bytes/sector} \\ &= \text{1440 sectors} \times \text{512 bytes/sector} \\ &= \text{720 KiB} \\ &= 737,280\text{ B} \simeq \text{737 KB} \end{aligned}$$

With CAV, the number of sectors in a track is limited by the medium density and the linear length of the innermost track. Hence the medium density is wasted in the outermost tracks. The larger the track radius, the lower the linear density of sectors.

☞ **EXERCISE 3** Calculate the capacity of the disk of exercise 2 assuming CAV format with 3500 sectors/track and a sector size of 1024 bytes. What is the areal density of the disk? Give it in both kbit/in^2 and Mbit/in^2 units.

CLV (*Constant Linear Velocity*): The linear length of sectors is constant throughout the disk surfaces, independently of the cylinder radius. The disk?s rotation speed is adjusted for each read/write operation so that the linear velocity remains constant for all tracks. Hence rotation is slower in the outer cylinders and faster in the innermost cylinders.

This distribution takes advantage of medium density throughout the available disk area, but it complicates the control electronics of the drive unit. CLV is not used on magnetic disks but is commonly applied to optical devices such as compact disks (CD). You can check by yourself how a CD drive makes a different sound depending on the diameter corresponding to the sector being read. This is caused by the varying rotation speed needed to achieve a constant linear velocity.

ZCAV/ZBR (*Zone Constant Angular Velocity/Zone Bit Recording*): Cylinders in the disk are grouped in several concentric, ring-shaped zones. Within each zone, tracks have a constant number of sectors.

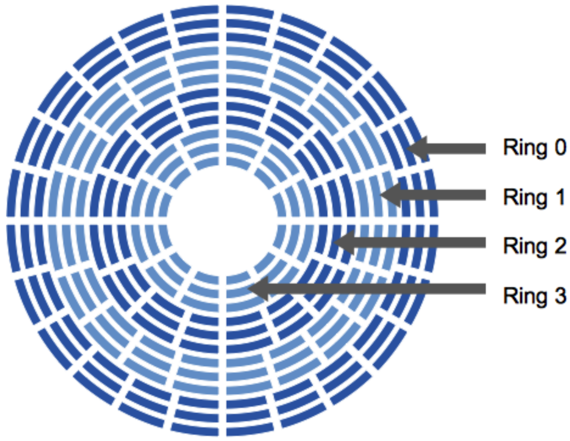

Figure 7: ZCAV distribution example. There are four zones, or rings, numbered 0 to 3, from the disk platter?s outer to the inner diameter. Within each ring, all tracks have the same number of sectors. All rings have three cylinders; outer rings, however, contain more sectors than inner rings.

If a zone z contains C_z cylinders, and there are S_z sectors/track in that zone, the disk capacity will be:

$$Capacity\ \text{(ZCAV)} = H \times \sum_z (C_z \times S_z) \times B$$

If C_z is constant, as is usually the case:

$$Capacity\ \text{(ZCAV)} = H \times C_z \times \sum_z S_z \times B$$

For example, Apple Macintosh computers formatted DS-DD floppy disks in five zones (or rings) of 16 cylinders each and defined 12, 11, 10, 9 and 8 sectors/track on them. The floppy disk capacity was:

$$\begin{aligned} Capacity\ \text{(DS,DD)} &= 2 \text{ sides} \\ &\times 16 \text{ cylinders/zone} \\ &\times (12 + 11 + 10 + 9 + 8) \text{ sectors/track per zone} \\ &\times 512 \text{ bytes/sector} \\ &= 1600 \text{ sectors} \times 512 \text{ bytes/sector} = 800 \text{ KiB} \\ &= 1600 \text{ sectors} \times 512 \text{ bytes/sector} = 819,2 \text{ KB} \end{aligned}$$

The ZCAV/ZBR geometry is a compromise between CAV and CLV. It improves the areal density without imposing too complicated control electronics. Disk drive units can take two approaches to handle a ZCAV disk. These approaches differ in whether they use constant linear velocity or constant angular velocity:

- The old Macintosh floppy disk drives adjusted the rotation speed depending on the zone to achieve a constant linear velocity that took maximum advantage of the medium density.
- Current hard disks favour a simpler control circuitry and apply a **constant rotation velocity**. Hence the linear speed (in bits/s) is always higher in the outermost rings than in the innermost rings.

☞ **Exercise 4** Calculate the capacity of the disk described in exercise 2 assuming a linear density of 52000 *tpi and that it receives ZCAV format with the following distribution of sectors of 1024 bytes.*

Zone	*Limits (ID - OD)*	*sectors/track*
0	*2.18 ″– 2.50 ″*	*5500*
1	*1.85 ″– 2.18 ″*	*4400*
2	*1.525 ″– 1.85 ″*	*3500*
3	*1.20 ″– 1.53 ″*	*2800*

What is the areal density of this disk? Give it in both kbit/in^2 and Mbit/in^2 units.

4 Hard disk performance

Current hard disks rotate at a constant speed, in *rpm*, in a range between 4,500 and 15,000 rpm.

Basic timing

A disk operates when it receives proper commands via its adapter. Read and write commands always target particular sectors. The timing of disk access involves several components:

1. First, the arm has to be positioned so that the heads reach the target cylinder (i.e., the one containing the track holding the target sector). The time needed by this action is called **seek time** and is measured in milliseconds. The exact seek time depends on how far the arm has to travel from its current position to the target cylinder and how fast it can move. This is why disk manufacturers usually give several timing parameters such as the *average seek time* (obtained statistically) and the time to travel from one cylinder to its immediate neighbour, known as the *track-to-track seek time*.

2. When the head arrives at the correct cylinder, the unit must wait for the addressed sector to pass beneath the head. This time is known as the **rotational latency** and is also measured in milliseconds. It depends on the rotation speed. Its average value is half the rotation period, i.e., the time needed for the disk to rotate 180^o. The faster the rotation velocity (also known as *spindle speed*), the shorter the rotational latency.

3. Then we need to count the **transfer time**, or time involved in the actual data transfer (sector read or write). This time depends on the disk's rotation speed and the medium density (because it impacts sector length).

Note that disk access follows a similar pattern to most kinds of memory:

- There is a latency between the start of the operation and the beginning of the actual data transfer (like DRAM's access time). This latency is the sum of the seek time and the rotational latency, which are variable. The seek time depends on how far the arm is from the destination cylinder, and the rotational latency depends on the location of the sector on the track. In the best case, the access latency will be almost null if the arm is already positioned in the target cylinder and the addressed sector is just about to pass below the head. But it is not difficult to find worse scenarios. In general, latencies fall in the range of 100 to 20 ms.

- There is an **internal transfer time** that depends on the linear velocity of the sector below the head. Under ZCAV, the linear speed in the outer rings is higher than in the inner rings because more sectors are transferred per disk rotation. Current disks have a maximum transfer speed of around 200 MB/s.

Optimal file storage

Most frequently, disk operations involve not one but a set of sectors. If all sectors of a file were spread randomly over the whole disk surface, accessing the entire file would incur seek and rotational latency for each sector of the file. To improve hard-disk performance, OS's and disk adapters typically apply **optimisations** when it comes to allocating sectors to files. In particular:

- A small file that fits the size of a track is optimally stored in *correlative* sectors of the track. Once the head is positioned on the first sector, access to subsequent sectors will not incur additional seek or rotational latency, as sectors just travel sequentially below the head.

- If the file takes several tracks, then it will be optimally stored using tracks of the same cylinder, if they fit (otherwise, see next case). As in the previous case, there will be no seek latency or rotational latency in between sectors either. Once positioned on the first sector, there is no need to move the head. All that is needed is to switch between them. Even the relatively small delay of switching heads is taken into account to achieve an optimal arrangement of sectors so that when the last sector of one track is accessed, the start of the first sector of the next track is just at its corresponding head.

- If the file takes more than one cylinder, then it will be optimally stored in adjacent cylinders. During sequential access to the whole file, when all sectors of a cylinder have been accessed, all that is needed is to move the head to the next cylinder. This is the best seek time you can get, the (*track-to-track time*). You may think that after moving the head to the next cylinder, you would incur the average rotational latency; but the rotational latency can also be eliminated by clever relative shifting of sectors among tracks. In this document, we assume this feature when we refer to an optimally stored file.

In summary, an optimally stored file takes correlative sectors of the same track; if it does not fit one track, it grows to other tracks of the same cylinder; and if it takes more than one cylinder, then it uses adjacent cylinders.

Operating systems try to assign sectors to files so that access to them is as fast as possible (see this Wikipedia article).

Another relevant aspect to consider in the overall disk performance is how the disk is connected to the computer. Today, the most frequently used connection (the Serial ATA bus or SATA) is way faster than the disk's internal transfer speed. However, the external transfer speed of the disk may be slower than SATA. Such is the case with some portable disks. It can also occur that a disk transfer conflicts with another transfer sharing the same computer bus. For these reasons, disks include a *buffer* that facilitates the synchronisation between internal and external transfers (see Figure 8).

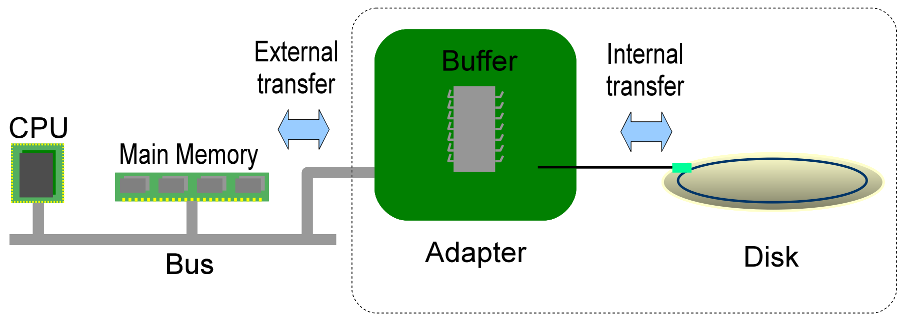

Figure 8: The disk buffer is part of the adapter's electronics. It solves general synchronisation issues among devices in the computer.

Examples and further exercises

As an example, consider a hard disk with a CAV organisation. Sectors are 512-byte long. For this example, consider a track containing 100 sectors. The average seek time is 10 ms, and the rotation speed is 6,000 rpm. Let's see how to calculate several relevant performance parameters of this disk.

- Average rotational latency:

$$\text{Time of a single rotation} = \frac{60 \text{ s/min}}{6000 \text{ rpm}} = 0.01 \text{ s} = 10 \text{ ms}$$

$$\text{Average rotational latency} = \frac{10}{2} = 5 \text{ ms}$$

- Average access time:

$$\begin{aligned}\text{Average access time} &= \text{Average seek time} + \text{Average rotational latency}\\ &= 10 + 5 = 15 \text{ ms}\end{aligned}$$

- Transfer time of a single sector (equivalent to the time it takes for a sector to pass below the head, once it is positioned in the target track):

$$\begin{aligned}\text{Sector transfer time} &= \frac{\text{Time of a single rotation}}{\text{Number of sectors in the track}}\\ &= \frac{10 \text{ ms/rotation}}{100 \text{ sectors/rotation}} = 0.1 \text{ ms}\end{aligned}$$

- Internal transfer speed:

$$\begin{aligned}\text{Internal transfer speed} &= \frac{\text{Track capacity}}{\text{Time of a single rotation}}\\ &= \frac{100 \text{ sectors} \times 512 \text{ bytes/sector}}{10 \text{ ms}} = 5120 \text{ KB/s}\end{aligned}$$

  or alternatively:

$$\begin{aligned}\text{Internal transfer speed} &= \frac{\text{Sector capacity}}{\text{Sector transfer time}}\\ &= \frac{512 \text{ bytes}}{0.1 \text{ ms}} = 5120 \text{ KB/s}\end{aligned}$$

- Average sector read time:

$$\begin{aligned}\text{Average sector read time} &= \text{Average access time} + \text{Internal transfer speed}\\ &= 15 + 0.1 = 15.1 \text{ ms}\end{aligned}$$

- Internal transfer time for 50 correlative sectors in the same track:

$$\begin{aligned}\text{Transfer time} &= \text{Sector transfer time} \times \text{number of sectors}\\ &= 0.1 \times 50 = 5 \text{ ms}\end{aligned}$$

- Average time for reading 50 correlative sectors in the same track:

$$\begin{aligned}\text{Average read time} &= \text{Average access time} + \text{Transfer time}\\ &= 15 + 5 = 20 \text{ ms}\end{aligned}$$

☞ **EXERCISE 5** Consider the disk described in exercise 4 rotates at 9000 rpm. The average seek time is 9 ms, and the track-to-track seek time is 0.8 ms. Calculate:

1. The average access time.
2. The internal transfer speed for each zone.
3. The average time it takes to read a 120 KB file, assuming it is stored in correlative sectors of the same track. Consider two cases: when the track is in zone 0 and when it is in zone 3.
4. The average time it takes to read a 120 KB file stored in randomly distributed sectors of cylinders located in zone 0. Assume the average seek time within a given zone is the average seek time divided by the number of zones, i.e., 9 / 4 = 2.25 ms.
5. The time for reading a 2000 MB file, assuming it is **optimally** stored in zone 0 (with all the optimisations described in Section 4).

5 Final notes and conclusion

To increase disk capacity, manufacturers have improved head design by reducing them to sub-millimetre sizes, modifying their shape to enable vertical polarisation (hence reducing the size of magnetic domains), and lowering their flying distance to just a few nanometers.

Another improvement comes from widening the capacity of physical sectors so that they can host logical sectors of 4 KiB without imposing the need for zone-grouping (see the Wikipedia article *Advanced Format*). This way, fewer control bits are needed, and the medium density is better exploited.

The combined effect of these improvements is in the basis of Kryder's Law (see the related article published in Scientific American), which states that the growth in the storage density of magnetic disks is faster than that of chips' integration described by Moore's Law.

Hard disk manufacturing is today concentrated in just three manufacturers, namely Seagate, Toshiba, and Western Digital (see this Wikipedia diagram).

6 More bibliography

In Wikipedia you will find descriptions of magnetic drums, ferrite core memories, magnetic tapes, and floppy disks. There is also a description of robotic libraries that used to populate the most significant data processing centres.

In Tom's hardware you will find miscellaneous information about hard disks: their history, components evolution, and a description of encoding techniques.

Also, in Wikipedia, in this article you will find references to other articles about the history of hard disks, the technological aspects discussed in this document (capacity and performance), and other aspects not cited here, such as market segments and reliability measurements. You will also find an informative video showing an open disk unit at work.

Annex: Style matters

Engineers must give numerical results so that they are as clear and precise as possible and use a standard form if one exists. When it comes to representing capacities, speeds, densities, or time, you should follow the basic rules given here:

Bits or bytes? Hard disk capacity is always given in bytes, with the appropriate prefix for the case (k, M, G, T...). Their transfer speeds are given in *bytes per second*, (whereas transfer speed in networks is typically given in *bits* per second). Areal density, however, is given in bits per area unit. In the case of hard disks, the disk area is given in *square inches*, (in^2 or *sq in*).

Do not say:	**Say:**
"The disk capacity is 320 Gbit"	"The disk capacity is 40 GB"
"The HD bandwidth is 800 Mbit/s"	"The HD bandwidth is 100 MB/s"
"The areal density is 10 GB/in^2"	"The areal density is 80 Gbit/in^2"

Prefixes: binary or decimal? Beware that, with hard disks, capacities are not given with binary prefixes, as is usual with main memory or cache; instead, they use decimal prefixes. So when a disk is labelled 10 GB, it has 10,000,000,000 Bytes. Remind that prefixes for transfer speeds or bandwidth are always decimal.

Use of prefixes in numbers with integer and fractional part If you use the decimal prefixes (see Figure 2), you could either write "1.53 GB" or "1530 MB". The first of these two is preferable (see next item). When we use decimal prefixes, changing the prefix is just a matter of shifting the decimal point.

With binary prefixes, however, shifting the decimal point does not work. For example, "1024 MiB" is exactly "1 GiB", and not "1.024 GiB". Remind that the ratio between two consecutive binary prefixes is 2^{10}, not 10^3. However, historical examples break this rule, such as labelling the capacity of "1440 KiB" floppy disks as "1.44 MiB". Indeed, that was not correct.

Write readable results In this task, you will obtain large results (e.g., billions of bytes) and small results (e.g. microseconds). Your calculator will use scientific notation to represent them. However, do not just copy what you read in your calculator when you give a result. Instead, use short decimal numbers with adequate prefixes and avoid scientific notation. That will make your results more readable. For example, rather than "$3.445 \cdot 10^7$ bytes" you should write "34.45 MB". Rather than "0.0015 s" you should write "1.5 ms".

How many significant digits to use? You should not write, e.g. "85.7547805 Gb/in^2" as a final result. Writing "85.8 Gb/in^2" is just fine. In general, do not use more than three significant digits for the integer part (choose the appropriate prefix) and don't give more than one or two digits in the fractional part. Be careful, however, not to overdo: this rule applies to *final* results, but don't be using it with every intermediate calculation, or you risk losing too much precision.