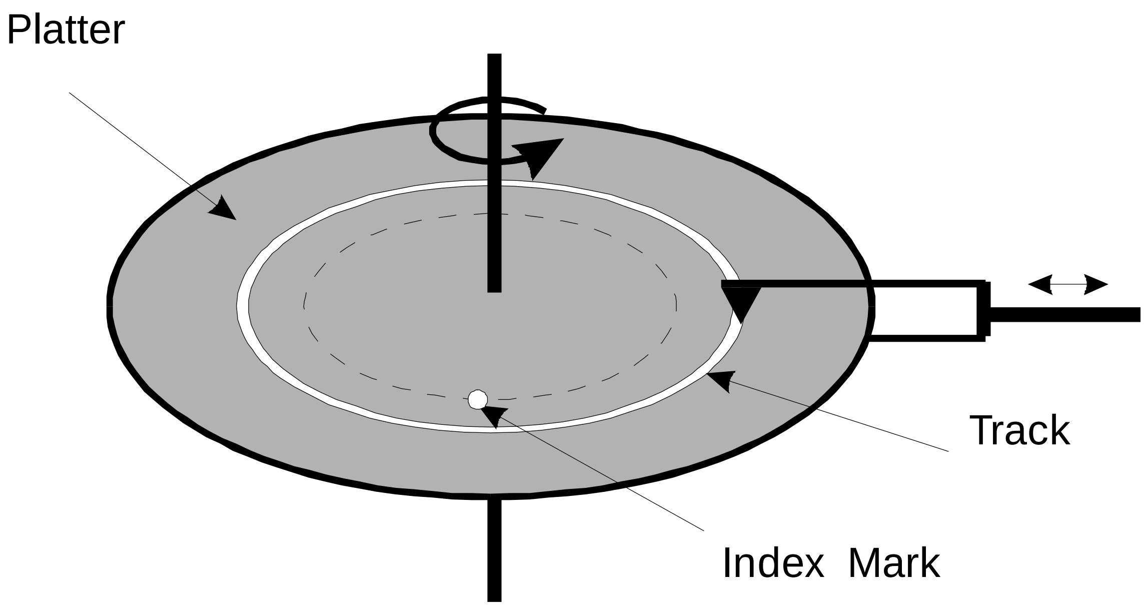
Disk Storage

The major "permanent" storage medium for computers is, at present, generally magnetic media in the form of either magnetic tape or disks. For microcomputers, tape storage is somewhat inconvenient as it is a serial access medium with very little parallelism and therefore disk storage is to be preferred in many, but not all applications.

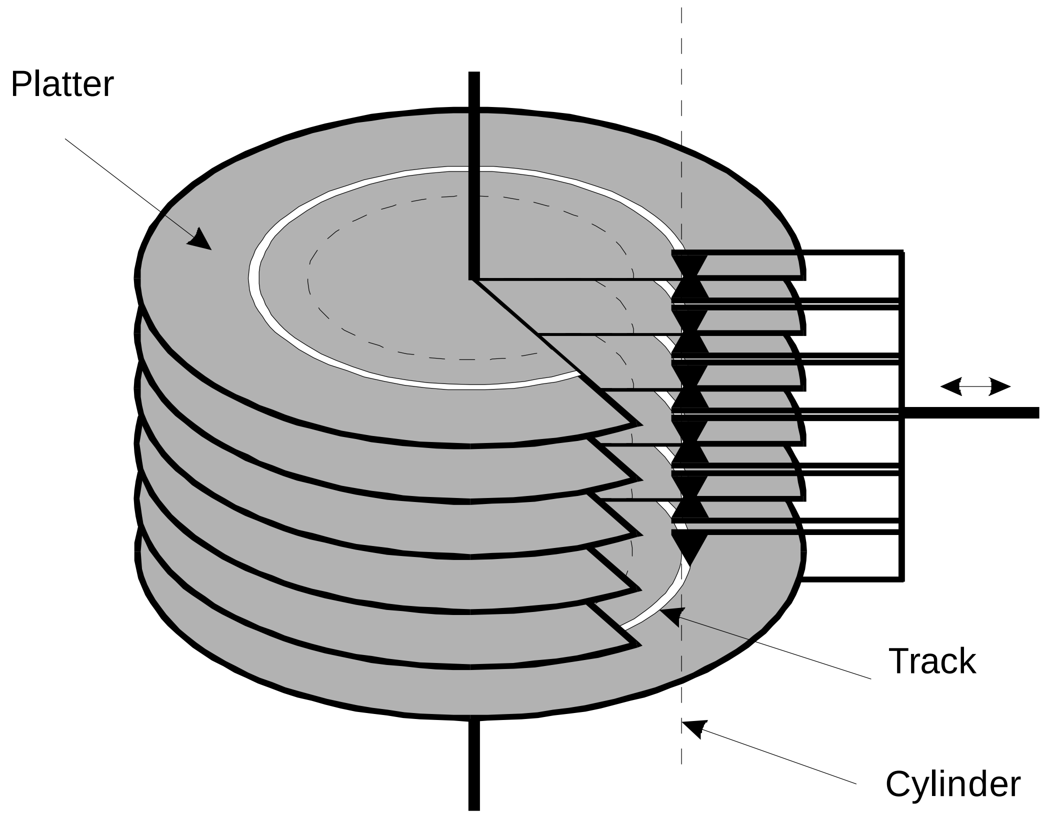


Disk Magnetic Structure

A magnetic disk consists simply of a disk coated with a similar material to that used in recording tape. As the disk spins, a head or heads moves over the surface of the disk on a radial arm in a stepwise fashion. At every step position the head scans a "track" of the disk, a track being a circle on the medium. Electrical currents in the head are used to record information on the track, or the magnetic variations induce currents in the head which are amplified to play back the recording. The tracks themselves are very narrow since the narrowness of the track determines the number of tracks which will fit on a side. The floppy disk drives, for instance, record at 135 tpi (tracks per inch) and fit 80 tracks onto a disk in less than one inch of space. Therefore disks are actually quite precise devices and should be treated as such - no dust, dirt, smoke etc. Do not drip coffee on them either.

Sides, Tracks, Cylinders and Sectors

Disks have two sides and one or both may be used. Larger disks have more sides by stacking "platters" (single disks) into larger assemblies. Each side requires a read/write head and therefore the cost goes up, but the degree of parallelism also goes up giving faster access to a random piece of data, or the same access time to a larger amount of data. On multi-platter disks the "tracks" accessible at a given head position are often referred to collectively as "cylinders".

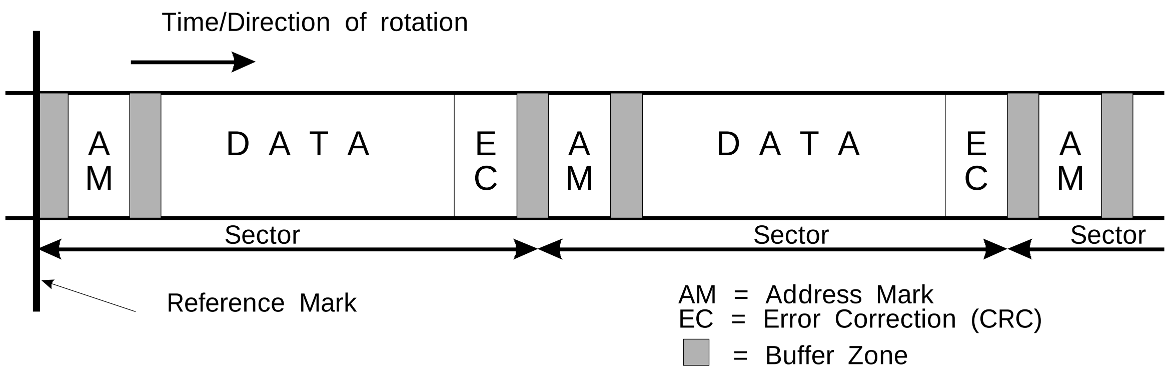


Multi-Platter Disk Structure

A track needs a "start" and "end" point and this is supplied by a mechanical marker which in the case of a floppy disk is an index hole on the inside edge which indicates the start of the track to the recording system.

A track is too large a segment of information to split the disk into. Even allowing for two sides of these small floppy disks, there are only 160 tracks on a disk and if the recording unit is a track that means that only 160 separate units of information may be stored. Furthermore if you only want to store a small amount of information you still need to take a whole track. So tracks are divided into "sectors" of information. The sectors are either delineated by more hardware markers (hard sectoring) or by software markers actually recorded on the disk (soft sectoring). The floppy disks use soft-sectoring.

A soft-sectored disk has a track structure of alternating sector markers and data areas. Each sector marker contains the number of the next sector and each data area is a sector of information. To read the disk we check the sector markers - called address marks - until we find the right one and then read the next data area after it. To write, we read the markers again and then record in the next data area. Of course we must be sure that recording is finished before the next sector marker or that will be corrupted, so there are buffer zones between the sectors sufficiently long to ensure that under all normal conditions data areas do not overlap address marks. We can therefore indicate the track structure linearly as:



Track Structure In Linear Format

The structure of Address Marks containing the sector numbers is laid down when a disk is "formatted", but for normal read/write operations only the data areas are changed.

Access Times

The time it takes to access the data depends upon how close we are to the data when the request is made and is composed of two elements:

a) The time taken to get the head to the right track

b) The time taken for the data to get under the head.

The worst case for a) is (number of tracks)*(time to step 1 track) and for b) time for 1 disk revolution. For small floppy disks the step time is about 5mS and the rotation rate 300 rpm - giving a worst-case access time of 80*5/1000 + 1/5 secs = 0.6 secs. Contrast this with the corresponding number for a 5Gbyte hard disk - 21mS (full stroke) and a rotation rate of 5400 rpm (specs for a Maxtor 85120A8 disk) - and we get 32mS over a much larger dataset. The average access time - which everybody likes to quote - is obviously less than the worst case. Note incidentally that a large computer, such as our "server" system, gets on with something else while the disk is doing a "seek" whereas ours hangs around until it's done.

Total Capacity

The number of sectors on a track depends upon the size of the sectors, the frequency of recording and the method of recording since some methods are inherently more dense than others. A standard 3.5" floppy disk squeezes 18 sectors of 512 bytes onto each track. There are two ways to increase this: either use more linear density of bits/m along the track or use more tracks per inch. Both refinements produce issues of alignment, for example the track density of a 3.5" floppy disk is 270 tpi - already about 0.004"/track!!!

By simple calculation small floppy disks hold 80*2*18*512 bytes or 1440k of information.

If you have been paying attention you will realise that because of the "gaps" and other marks which are required in the formatting process, the amount of data which you can actually store on a disk is less than the amount that you would get by multiplying the "bits/inch" figure by the total length of the tracks. This explains why an "HD" 3.5" floppy disk for example is listed as "2Mb" capacity on the label but only formats to 1,474,560bytes of information at best - and there is still some to be lost in the directory structure.

Caching

Since disks are slower than the main CPU and memory is getting cheaper all the time, many operating systems and some disk controllers “cache” data as it is transferred to and from the disk.

When reading a disk it is often the case that we can predict the next piece of information required - it is just the next piece of information from the file. Thus we can read that information and store it locally even if the user hasn’t asked for it yet. Then, when it is requested, the access time is apparently blindingly fast. The success of a caching scheme depends upon:

- Being able to accurately predict what is going to be required next.
- Large reads being much more efficient than short ones
- Having a lot of memory
- Spending less time finding out if the data is in the cache than it would take to get it from the disk.

The maximum size of a cache is a matter of some experiment. At some point the cache starts to lose efficiency because it is violating one of the points above - maybe reading too far ahead and reading a lot of useless data - and the apparent speed drops. Systems are often tuned by actually looking at the rate of processing as a function of the cache size.

When writing a disk, caching can also be used to improve the efficiency of the process. Writing large blocks is more efficient than writing small blocks. It might also be that the data is required to be read back again quickly. The same general arguments apply to size as were discussed above.

There is a hidden problem in disk-caching - it inherently slows down the rate at which you can read or write to the disk! This can be seen in its simplest form in the following situation. If you want to read the entire disk in one go from end to end. Is it easier just to read the next piece of data, or spend time trying to figure out whether it is in the cache? It must be faster just to read the data because we all know it won't be in the cache. The correct strategy would be to read the disk in as large a unit as possible - preferably a cylinder at a time - in sequence to minimise head movement time. You will find that some video systems actually require you to turn off any caching systems in your operating system for this very reason.

There is one additional point about write caching - if the computer crashes while there is data pending in the write cache, the state of the disk is uncertain. What this means is that there is a finite chance that the computer will be unable to make sense of the disk structure when it comes back up and at a minimum recent data will be lost. At worst, the entire disk could be unreadable!

To prevent this, disk write caches should be "flushed" (ie emptied) before turning the computer off. This implies that all computers with caches need to be "shut down" rather than simply "turned off". This is the case with your machines - please observe the rule to "shut down" rather than turn off.

As an additional measure, disk caches often flush if there has been no system activity for a few seconds - this helps to keep the real disk and the "apparent disk" synchronised in the case of a power outage or other problem.

Why go On about It?

For our purposes the main problem with disks is knowing reliably how fast you can read or write to them. This limits the rate at which we can take data through the system without a compression step first. With modern processors the limiting feature is often the disk storage rate which can be as high as 6Mbytes/sec and as low as 10kbytes/sec and anything in between. Limitations may come from any part of the disk and operating system as we shall see.