



하드디스크 구조 +

주소 지정 방식 (CHS, LBA)

김우종

1. 하드디스크의 구조와 원리

1. 하드디스크의 구성 틀
2. 하드디스크의 구조
3. 하드디스크의 저장 원리

2. 섹터 주소 지정 방식

1. CHS 주소 지정 방식
2. LBA 주소 지정 방식

1-1. 하드디스크의 구성 틀



1-2. 하드디스크의 구조



플래터

오른쪽에 보이는 저 원판이며
실질적인 정보들이 기록되는 역할 장치



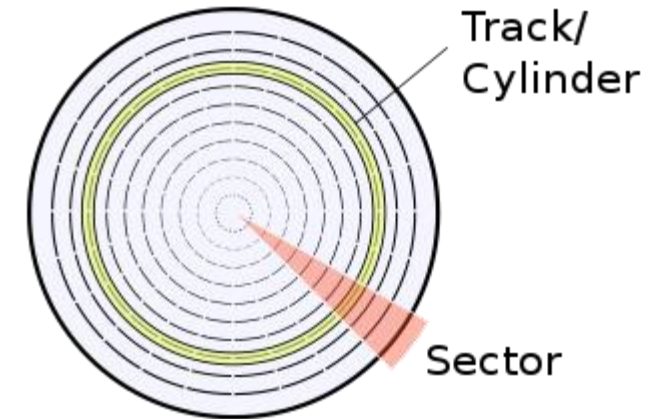
1-2. 하드디스크의 구조



플래터의 구조

섹터 : 디스크의 최소 저장 단위 (512byte = 0.5KB)

실린더 : 트랙의 모음



1-2. 하드디스크의 구조



액츄에이터 헤드

오른쪽에 보이는
저 수술칼 처럼 보이는 친구에 끝부분
데이터를 쓰는 역할 장치



1-2. 하드디스크의 구조



액츄에이터 암

오른쪽에 보이는

저 수술칼 처럼 보이는 친구에 끝부분을 제외한 부분
데이터를 쓰기 위한 헤드를 움직여주는 역할



1-2. 하드디스크의 구조



스핀들 (스핀들 모터)

모터를 사용하여 플래터를 돌리는 역할을 하는 장치



1-3. 하드디스크의 저장 원리



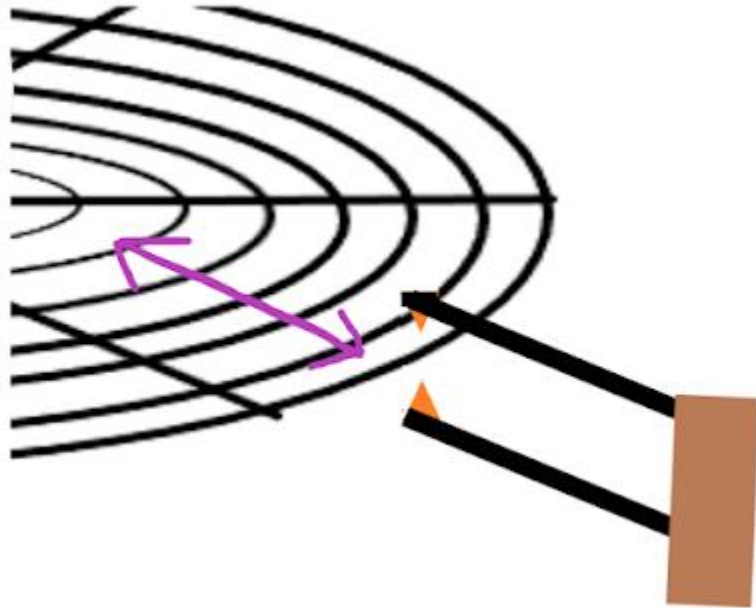
1-3. 하드디스크의 저장 원리



1-3. 하드디스크의 저장 원리



1-3. 하드디스크의 저장 원리



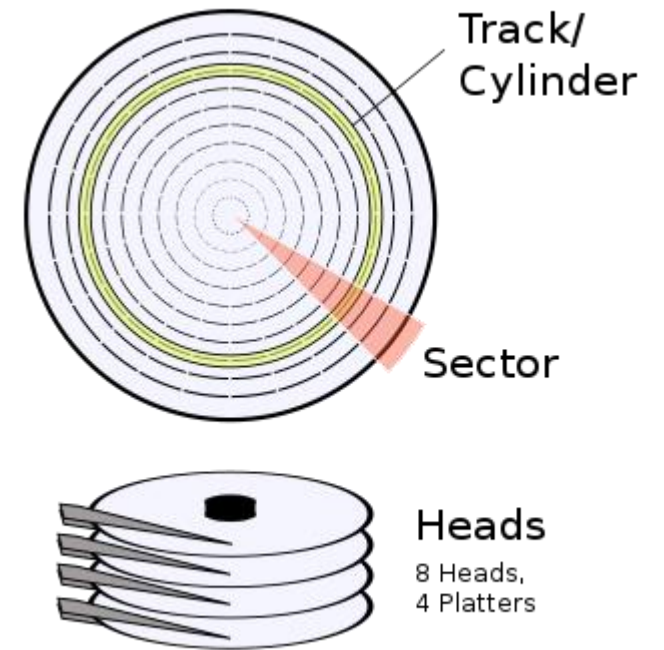
2-1 CHS 주소 지정 방식



CHS 주소 지정 방식

CHS(Cylinder-Head-Sector)주소 지정 방식은 이름과 같이 실린더,헤드,섹터에 번호를 할당해 이를 주소로 데이터를 찾아 읽고 쓰는 방식

표기 방식 : CHS(1,2,3)



2-1 CHS 주소 지정 방식



CHS 주소 지정 방식

**초기의 ATA 표준은 28bit 블록 주소 방식
BIOS의 조건에 부합하지 않아 후에
ATA-1로 정식 소개때는 24bit로 소개**

	최대 실린더 수	최대 헤드 수	최대 섹터 수	표현 가능한 최대 용량
IDE/ATA Limits	16(65,536)	4(16)	8(255)	128 GB
BIOS	10(1,024)	8(256)	6(63)	7.88 GB
최소 가능 비트	10(1,024)	4(16)	6(63)	504 MB

2-1 CHS 주소 지정 방식



CHS 주소 지정 방식

이 때 보면 알 수 있듯이 표현 가능한 최대 용량은 504MB이므로 한계가 존재...

이 한계점이 들어나며 LBA가 각광받기 시작

	최대 실린더 수	최대 헤드 수	최대 섹터 수	표현 가능한 최대 용량
IDE/ATA Limits	16(65,536)	4(16)	8(255)	128 GB
BIOS	10(1,024)	8(256)	6(63)	7.88 GB
최소 가능 비트	10(1,024)	4(16)	6(63)	504 MB

2-1 CHS 주소 지정 방식



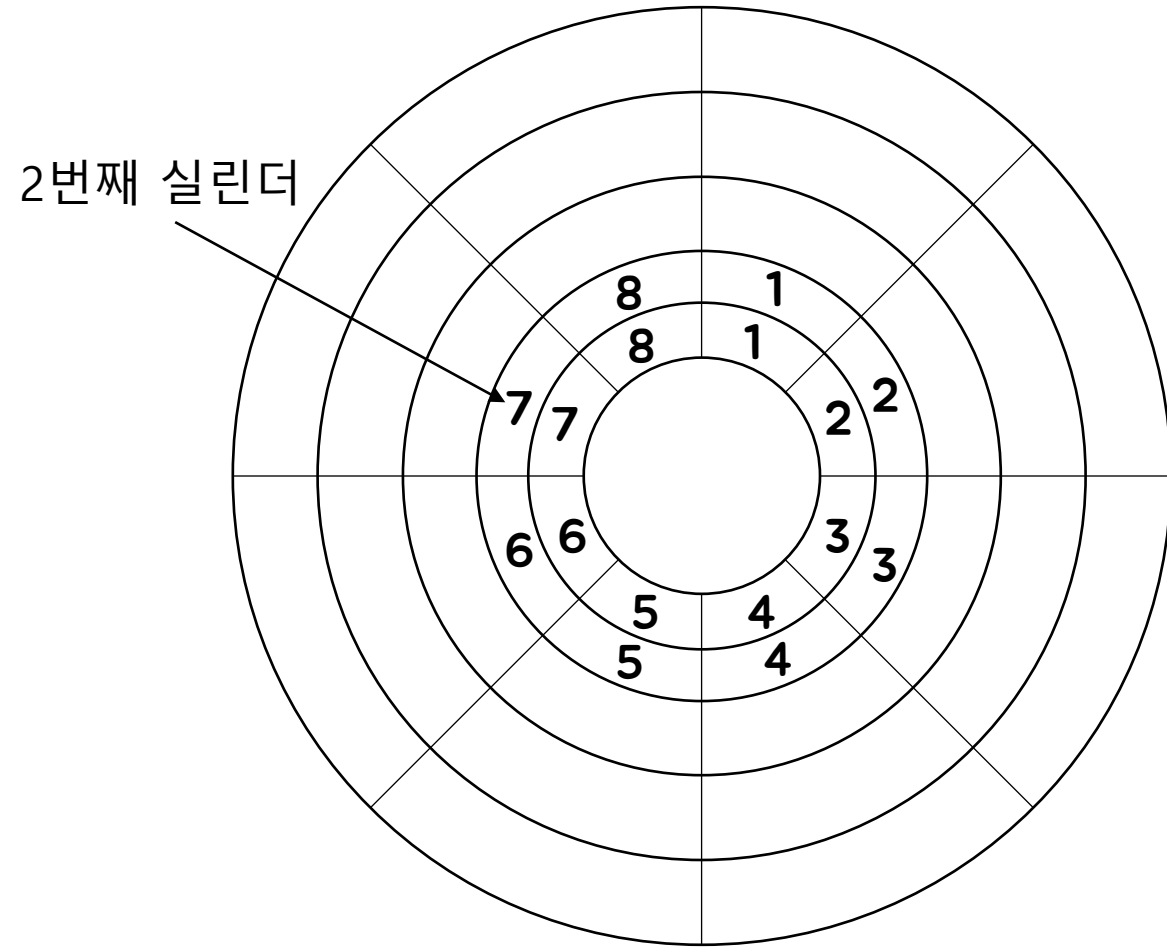
LBA 주소 지정 방식

LBA(Logical-Block-Addressing)는 논리적 주소를 가짐

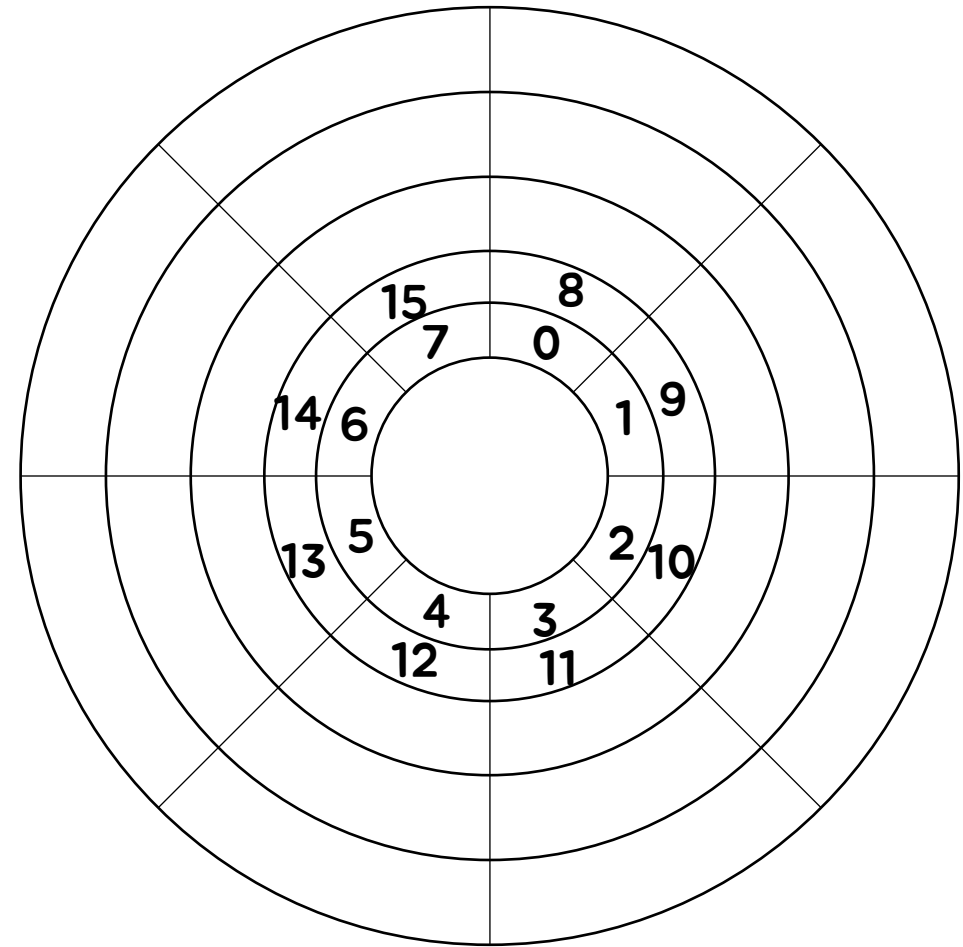
LBA는 CHS와 별개로 생성된 것이 아님 다만 CHS의 한계점이 들어나며
각광받았을 뿐

추가로 LBA가 도입되었을 때 BIOS와 OS또한 28bit-LBA를 지원하도록
변경

2-1 CHS 주소 지정 방식



CHS



LBA

2-1 CHS 주소 지정 방식



LBA 주소 지정 방식

LBA는 28bit로 2^{28} 섹터가 지정 가능

이는 용량으로 섹터가 512byte일 때 128GB의 용량이 됨

그럼 현재 우리가 쓰는 1TB, 2TB 짜리 하드는??

2-1 CHS 주소 지정 방식



LBA 주소 지정 방식

28bit로 더 큰 용량을 만들 수 없게 되자 48bit LBA를 고안

48bit를 사용하면 144PB(PeraByte)의 용량이 됨

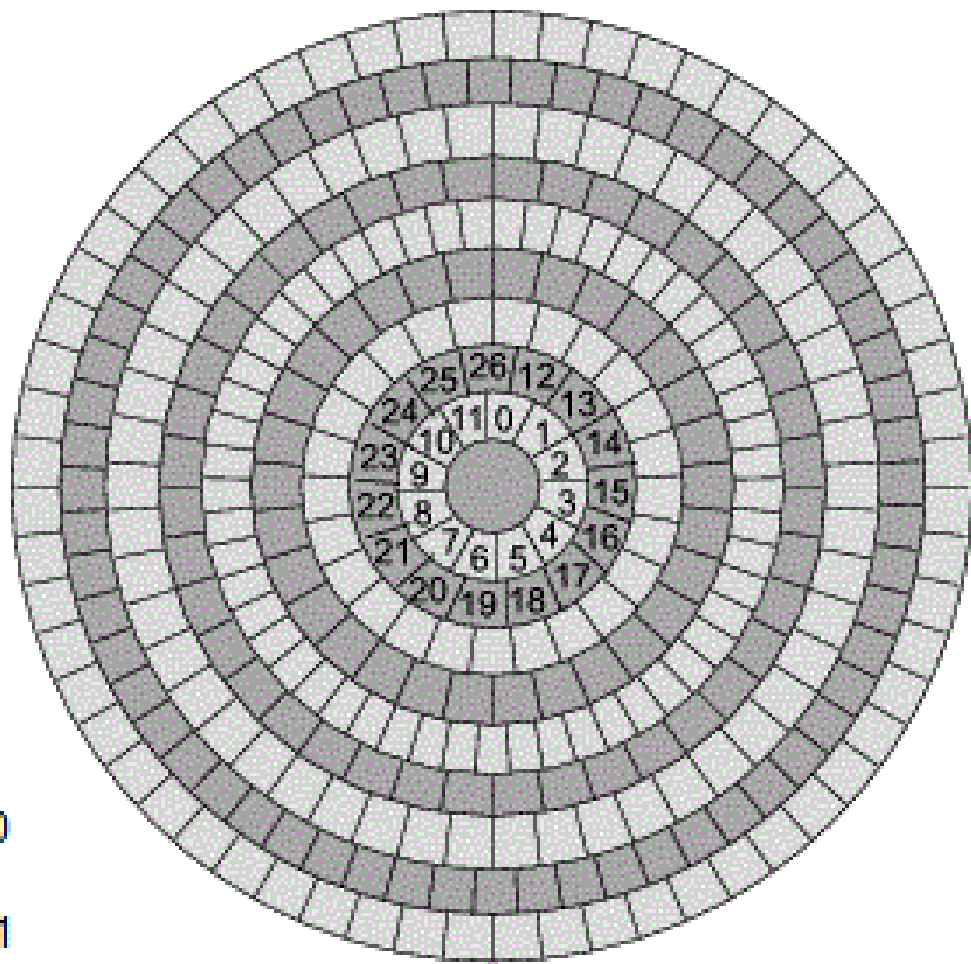
이는 GB로 환산하면 144,000,000GB임

ZBR 방식

최근 디스크는 이 ZBR(Zone Bit Recording) 방식을 이용

물리적으로 트랙의 길이가 내부보다 외부가 길다는 점에서 착안하여 트랙의 길이가 길 수록 더 많은 섹터를 할당

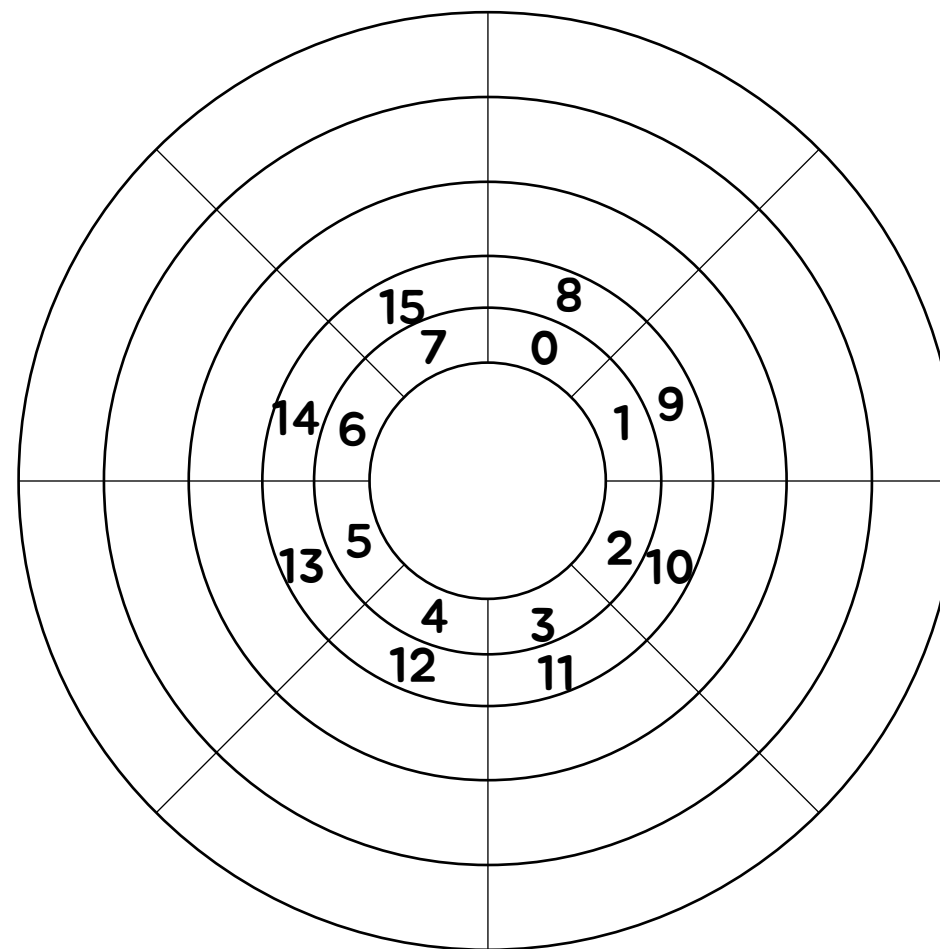
+ α



r0

r1

LBA- addressing



LBA



Q&A