

## 2. 파일 저장장치



© DBLAB, SNU

### ❖ 파일 저장장치의 특성

- ◆ 저장장치 (storage device)
  - 저장 매체
  - 매체에 데이터를 저장하고 검색하기 위한 장치
- ◆ 저장 매체 (storage medium)
  - 소멸성(volatile) vs 비소멸성(nonvolatile)
- ◆ 접근 장치 (access device)
  - 데이터를 판독하거나 기록하는 장치

## ▶ 저장장치 (memory, storage)

### i) 1차 저장장치(primary storage)

#### 1. 메인 메모리(main memory)

- 내용 접근 시간은 일정하고 빠름
- 프로그램/데이타를 처리하기 위한 작업 공간

#### 2. 캐시 메모리(cache memory)

- 메인 메모리 성능 향상 목적



© DBLAB, SNU

## ▶ 저장장치 (main memory, storage)

### i) 2차 저장장치(secondary storage)

#### 1. 자기 디스크(main memory)

- 내용 접근 시간이 느림
- 용량이 크고 싸서 주로 화일 저장에 쓰임
- 저장된 데이타는 메인 메모리를 거쳐 CPU에 의해 처리

#### 2. 광 디스크(optical disk), 자기 테이프(magnetic tape) 등



© DBLAB, SNU

## ❖ 저장장치의 유형

### ◆ 캐시 메모리(cache memory)

- 가장 빠르고 가장 비싼 저장장치
- SRAM(static random access memory)
  - ◆ 데이터를 유지하기 위해 자주 재생(refresh)할 필요가 없음

### ◆ 메인 메모리(main memory)

- 프로그램 실행과 이에 필요한 데이터 유지 공간
- DRAM(dynamic random access memory)
- 용량이 적고, 소멸성 (데이터 저장에는 부적합)



© DBLAB, SNU

## ❖ 저장장치의 유형

### ◆ 플래시 메모리(flash memory)

- 고밀도, 고성능 메모리로서 비소멸성
- 메인 메모리와 비슷한 접근 속도
- 재기록 시 한 블록 전체를 동시에 지우고 기록해야 함

### ◆ 자기 디스크(magnetic disk)

- 데이터 저장 장치의 주 매체
- 데이터 처리와 기록은 메인 메모리를 거쳐야 함
- 대용량이고 비 소멸성



© DBLAB, SNU

## ❖ 저장장치의 유형

### ◆ 광 디스크(optical disk)

- 광학적으로 저장, 레이저로 판독
- 용량이 크고 보존 기간이 길다
- DVD(digital video disk): 4.5GB - 15GB

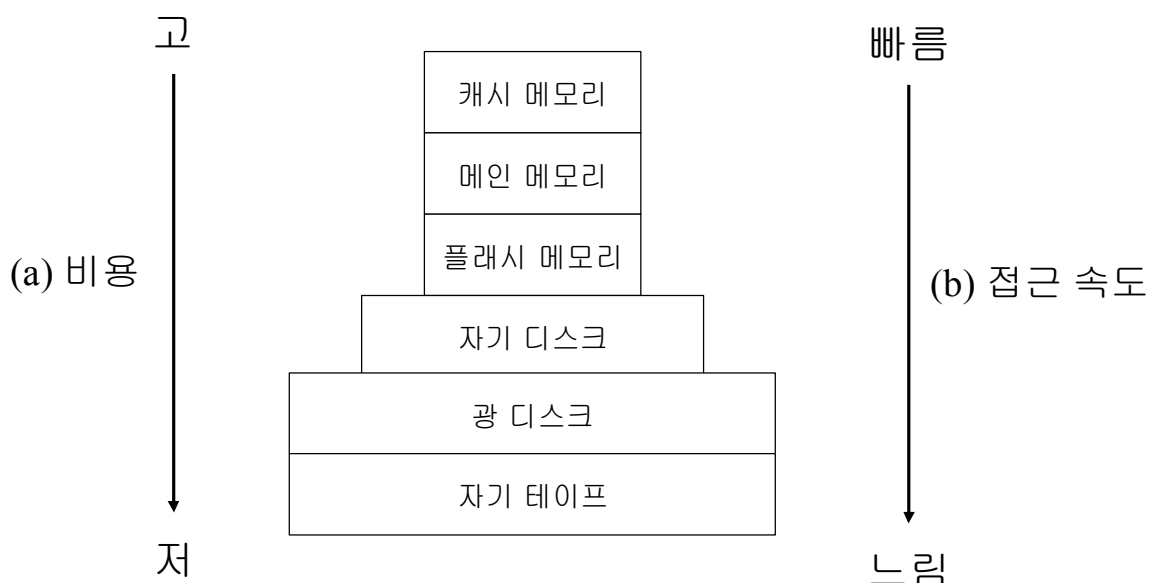
### ◆ 자기 테이프(magnetic tape)

- 데이터의 백업과 보존을 위한 저장매체
- 순차 접근 저장 장치
- 테이프 쥬크 박스 (대용량 데이터 저장 가능): TB 규모



© DBLAB, SNU

## ❖ 저장장치의 계층



© DBLAB, SNU

# 1. 하드 디스크 (Magnetic Disk)

- 직접 접근 저장 장치 (DASD; direct access storage device)  
중 가장 많이 쓰이는 장치

## ◆ 종류

- 하드 디스크(hard disk) : 1955년 IBM 개발-초기 5MB
- 유연한 디스크(flexible disk) : floppy disk, diskette

## ◆ 분류 기준

- 기록 표면의 수 (recording surfaces)
- 데이터 전송률 (data transfer rate)
- 기록/판독 헤드 이동 시간 (R/W head movement times)
- 접근 방법 (access mechanism technology)
- 회전 지연 (rotational latency)
- 밀도 (density)



© DBLAB, SNU

## (1) 자기 디스크의 물리적 특성

### ◆ 디스크 팩(disk pack)

- 디스크 원반(platter)의 모음(보통 6~20개의 원반)
- 원반 직경 : 10.5인치, 14인치(소형의 경우 1~3.5인치)
- 면당 : 수천 ~ 14,000개 이상의 트랙
- 기록 표면(surface) : 양면 사용(맨 위/아래 면 제외)
  - ◆ 11개의 디스크는 20면
- 회전 속도 : 3,600 ~ 7,200 rpm (revolutions per minute)

### ◆ 디스크 구동기(disk driver)

- 제어기, 접근 암, 판독/기록 헤드, 팩 회전 장치
- 고정식, 탈착식

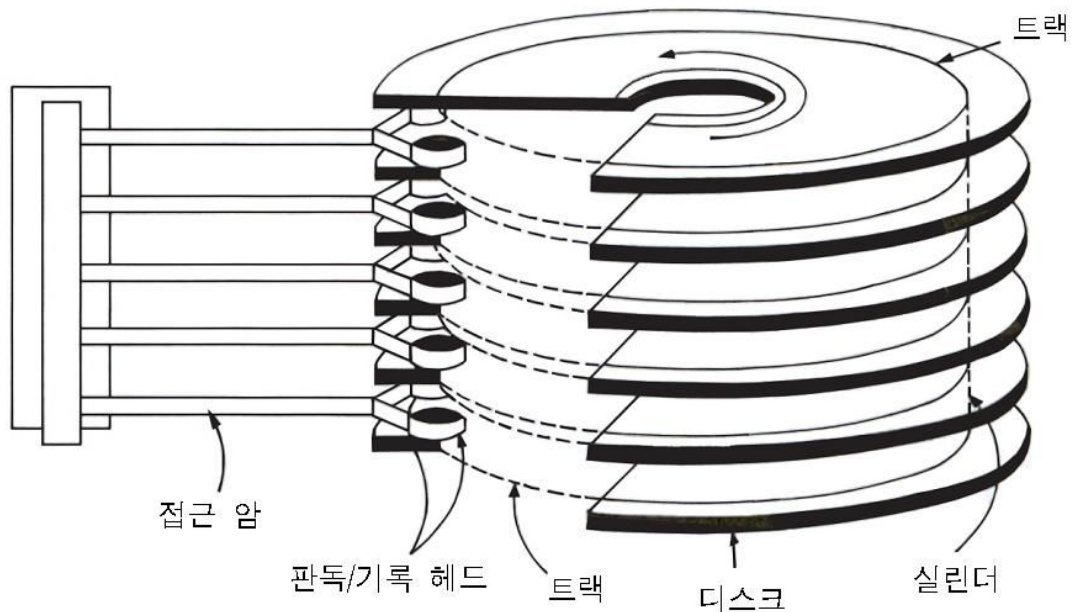
### ◆ 디스크 제어기(controller)

- 원하는 데이터가 어느 구동기, 어느 면, 어느 주소에 있는지 판독
- 버퍼 관리
- 오류 발견/수정
- 판독/기록 관리



© DBLAB, SNU

## (1) 자기 디스크의 물리적 특성



© DBLAB, SNU

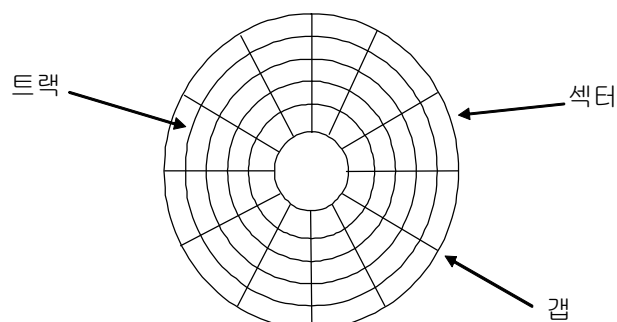
## (2) 데이터 저장

### ◆ 디스크의 구성

- 트랙(track): 갭(gap)으로 분리된 섹터(sector)들로 구성
- 섹터(sector): 기록과 판독 작업의 최소 단위
- 실린더(cylinder) : 지름이 같은 모든 트랙

### ◆ 블록(block)

- 디스크와 메인 메모리 사이에 전송되는 데이터의 논리적 단위
- 블록은 하나 이상의 섹터로 구성



© DBLAB, SNU

### (3) 유동 헤드 디스크와 고정 헤드 디스크

#### ◆ 유동 헤드 디스크(movable-head disk)

- 각 기록 면마다 판독/기록 헤드가 있음
- 판독/기록시 헤드가 원하는 트랙에 위치하도록 액세스 암을 이동
- 전송시간 : 데이터를 동일 면 보다는 동일 실린더에 저장하는 것이 더 효율적

#### ◆ 고정 헤드 디스크(fixed-head disk)

- 기록면의 각 트랙마다 하나의 판독/기록 헤드가 있음
- 판독/기록을 위해 헤드를 이동시킬 필요가 없음



© DBLAB, SNU

### (4) 윈체스터 디스크

#### ◆ 윈체스터 디스크(Winchester disk)

- 기록면, 접근 장치, R/W 헤드가 함께 밀봉
- 기록밀도가 높음 - 헤드가 디스크에 근접
- 보편적인 하드디스크는 윈체스터 디스크임
- 특성
  - ◆ 용량: 수 GB - 수십 GB
  - ◆ 회전속도: 5400 - 10000 rpm
  - ◆ 전송률: 5 - 50MB/sec



© DBLAB, SNU

## 2. 플로피 디스크 (Floppy Disk)

- 유연한 디스크(flexible disk) 저장장치
  - ♦ floppy, diskette(1970년경 IBM에서 소개)
- 직경 :  $3\frac{1}{2}$ ( $5\frac{1}{4}$ )인치,
- 회전속도 : 360 rpm
- 3.5인치
  - ♦ 80트랙, 18섹터/트랙, 512B/섹터: 1.44MB
- 5.25인치
  - ♦ 80트랙, 15섹터/트랙, 512B/섹터: 1.2MB
- 수직축 기록(vertical axis recording) :  
저장 용량을 늘리기 위한 신기술 - 표면과 수직방향으로  
자화시켜 각 자화점의 표면 폭을 축소시킴. 밀도 50배  
이상 향상



© DBLAB, SNU

## ❖ 디스크 저장장치의 특성

### ◆ 디스크의 특성 요소

- 회전 속도
- 디스크 드라이브의 원반 수
- 기록 면당 트랙 수
- 트랙당 섹터 수
- 섹터(트랙)당 바이트 수

### ▶ Giga 디스크(예)

- 디스크 드라이브- 원반 수: 8개, 기록 면: 16개
- 16,384개의 트랙/기록면
- 128섹터/트랙
- 4096바이트/섹터
- 용량: 16면x16384트랙x128섹터x4096바이트=128GB



© DBLAB, SNU



## ❖ 디스크 데이터 접근

### ◆ 디스크에 저장된 데이터 처리

- 메인 메모리에서만 연산 작업이 가능
- 디스크와 메인 메모리 사이에 데이터 블록 전송이 필요
- 블록은 하나 이상의 섹터로 구성

- a) 블록이 위치하는 트랙이 포함된 실린더에 헤드가 위치
- b) 디스크가 회전하면서 블록이 포함된 섹터들을 헤드가 인식, 판독/기록

### ◆ 데이터 접근 시간(access time)

- 탐구 시간(seek time)
- 회전 지연 시간(rotational latency)
- 전송 시간(transfer time)



© DBLAB, SNU

## 1. 탐구시간 (seek time) : s

- 원하는 데이터가 있는 실린더(또는 트랙)에 R/W 헤드를 위치시키는데 걸리는 시간
- 헤드 이동 거리와 시간 관계는 비선형적

$$s = c + \delta \cdot i$$

$c$  : 접근장치가 처음 가동하는데 걸리는 일정한 시간  
 $\delta$  : 단위 거리 이동 시간  
 $i$  : 트랙간의 간격을 단위로 이동한 거리  
 $c + \delta < 1$  회전

- 속도 평가로 평균 탐구 시간(average seek time : s)을 사용



© DBLAB, SNU

## 2. 회전 지연 시간 (rotational latency) : r

- 탐구완료에서 데이터 전송 시작까지의 지연
- 목표 트랙에서 판독/기록 헤드 밑에 블록의 첫 번째 섹터가 위치할 때까지의 시간
- 분당 디스크 회전 수: rpm(revolutions per minute)
- 디스크 회전속도(ms): rpm/(60x1000)

$$r = \frac{1}{2} \cdot (1 \text{회전 시간})$$

$$1 \text{회전 소요 시간(ms)} : 2r = \frac{60 \times 1000}{rpm}$$

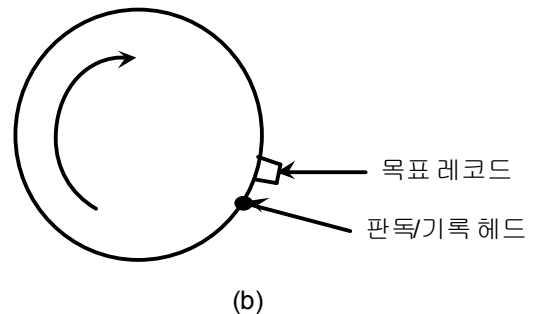
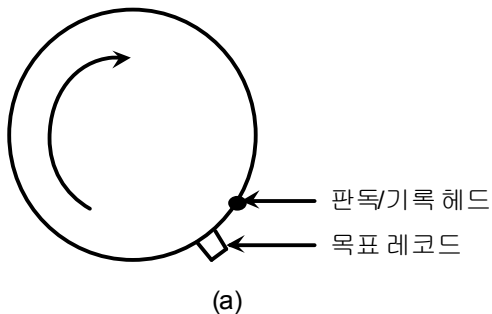
3,600 rpm : 8.33 ms

5,400 rpm : 5.56 ms



© DBLAB, SNU

### ▶ 회전 지연 시간



© DBLAB, SNU

### 3. 전송 시간 (transfer time)

#### ◆ 전송 시간(transfer time; t)

- 블록의 섹터들과 이들 사이의 갭들이 헤드 밑을 회전하며 통과하는데 걸리는 시간
- 트랙 크기가 250000B이고 1회전 소요시간이 10ms일 때 25MB/sec 판독 가능
  - ◆ 16384B 크기의 블록 전송 시간: 2/3ms

#### ◆ 전송률 (transfer rate)

- 초당 데이터가 전송되는 속도(MBps)
- 25MBps



© DBLAB, SNU

### 4. 블록의 판독

#### ◆ Giga 디스크로부터 16,384B 크기의 블록 판독

- 분당 회전수 7200rpm (8.33ms/rotation)
- 헤드의 트랙간 이동 시간(1.001ms)
- 헤드의 최대 이동시간-  
(16,384 트랙을 이동하는 시간: 17.38ms)

#### ◆ 블록 전송 시간

- 16,384B는 4개의 섹터(4096B)와 3개의 갭으로 구성
- 트랙의 10%는 갭, 90%가 섹터
- 트랙은 128개의 섹터와 128개의 갭으로 구성
- 블록의 아크 각도 :  $36^\circ \times (3/128) + 324^\circ \times (4/128) = 10.97^\circ$
- 전송 시간  
 $(10.97/360) \times 8.33 = 0.25\text{ms}$



© DBLAB, SNU

## 4. 블록의 판독

### ◆ 블록의 판독 시간

- 최소 블록 판독 시간 : 블록의 위치가 헤더의 위치와 일치한 경우:  
 $0.25\text{ms}$
- 최대 블록 판독 시간 : 헤드의 최대 이동 시간(블록이 가장 바깥쪽 실린더에 위치하고 헤드는 가장 안쪽에 위치)과 최대 회전 지연 시간이 필요한 경우:  
 $17.38 + 8.33 + 0.25 = 25.96\text{ms}$
- 평균 블록 판독 시간
  - (a) 블록 전송 시간 :  $0.25\text{ms}$
  - (b) 평균 회전 지연 시간 :  $4.17\text{ms} = (8.33/2)$
  - (c) 평균 탐구 시간 : 전체 트랙 수(16384)의 1/3 을 이동하는 시간:  
 $6.46\text{ms} = (1+5461/1000=6.46\text{ms})$ 
    - 출발과 정지를 위해  $1\text{ms}$
    - 1000실린더 이동에  $1\text{ms}$  걸림

평균 블록 판독 시간 =  $6.46 + 4.17 + 0.25 = 10.88\text{ms}$



© DBLAB, SNU

## 5. 블록의 기록과 갱신

### ◆ 블록의 기록

- 블록의 기록은 블록을 디스크로부터 메인 메모리로 전송하는 대신 메인 메모리로부터 디스크 블록으로 전송하는 것만 다르고 다른 과정은 블록의 판독과 동일

### ◆ 블록의 갱신

- 디스크에서의 직접 갱신은 불가능
  - (1) 메인 메모리로 블록 이동
  - (2) 메인 메모리에서 블록 사본을 갱신
  - (3) 갱신된 블록 사본을 디스크에 재기록

### ◆ 블록 갱신 시간

- 블록 판독 시간 + 블록 사본 갱신 시간 + 블록 재기록 시간
- 메인 메모리 내에서의 블록 갱신 시간은 보통 무시



© DBLAB, SNU

## 6. 블로킹

### ◆ 블록(block)

- 디스크와 메인 메모리 사이의 데이터 전송의 단위 : 물리적 레코드(physical record)
- 트랙 길이 =  $b(\text{\# of blocks in a track}) \times B(\text{block size})$   
(블록의 크기  $\leq$  트랙의 길이)

### ◆ 블록의 크기

- 보통 512byte, 1KB, 4KB
- 너무 크면 불필요한 데이터 전송을 야기
- 과다한 버퍼 할당으로 메모리의 효율성 저하



© DBLAB, SNU

## (1) 블로킹 인수 (blocking factor) : $Bf$

### (1) 블로킹(blocking)

- 메인 메모리와 I/O 효율을 위해 몇 개의 논리적 레코드를 하나의 물리적 레코드(블록)에 저장시키는 것
- 논리적 레코드(logical record)는 사용자가 데이터를 저장하기 위한 논리적 단위
- 블로킹 인수 ( $Bf$ ): 한 블록에 포함시킬 수 있는 논리적 레코드 수

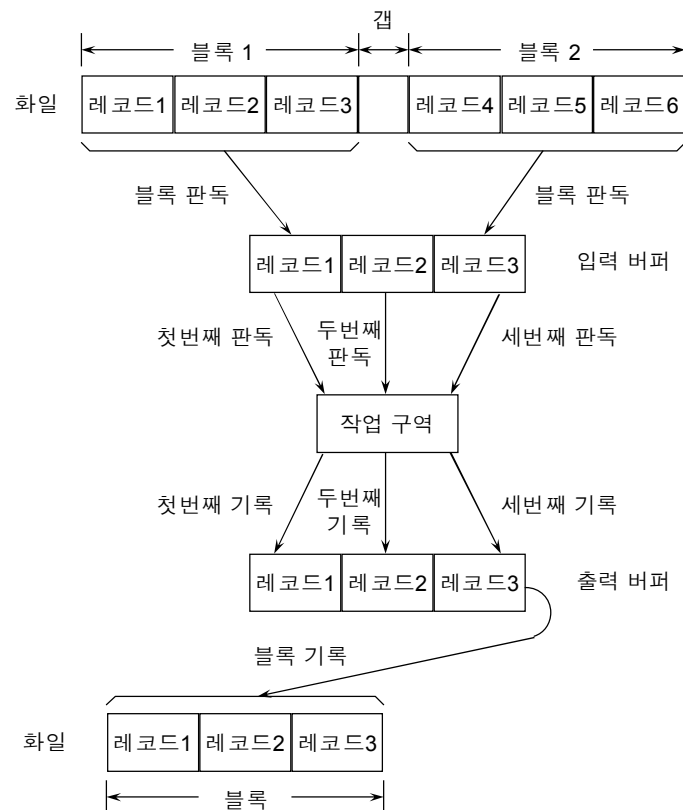
$$Bf = \lfloor B/R \rfloor$$

$R$  : 레코드 크기 (fixed or variable)  
 $B$  : 블록 크기



© DBLAB, SNU

## ▶ 블로킹을 이용한 레코드 판독과 기록



© DBLAB, SNU

## ▶ 블로킹 방법

### i) 고정 길이 블로킹

- 고정 길이 레코드(fixed length records)를 블록에 저장
- fixed length block

### ii) 신장된 가변 길이 블로킹

- 신장된 가변 길이 레코드(spanned variable length records)를 블록에 저장
- fixed length block (with spanning)

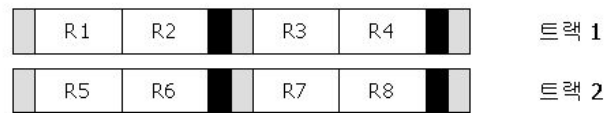
★ 신장(spanned) : 한 레코드가 인접한 몇 개의 블록에 걸쳐 저장

### iii) 비신장된 가변 길이 블로킹

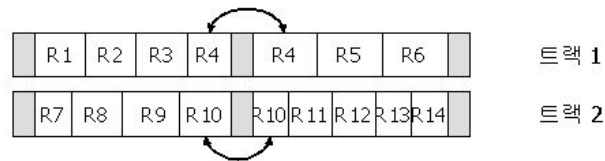
- 고정 또는 가변 길이 레코드(fixed or variable length records)를 블록에 저장
- 레코드는 블록들에 분할 저장될 수 있음
- fixed length block (with no spanning)



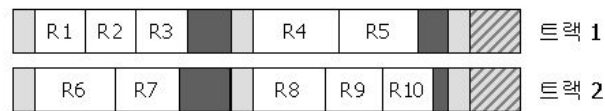
© DBLAB, SNU



고정 길이 블로킹



신장된 가변 길이 블로킹



비신장 가변 길이 블로킹

-  데이터
-  하드웨어 갭
-  트랙의 크기가 블록 크기의 정수배가 아닌 데서 생기는 낭비
-  블록의 크기가 레코드 크기의 정수배가 아닌 데서 생기는 낭비
-  블록의 크기가 고정 길이 레코드 크기의 정수배가 아닌 데서 생기는 낭비

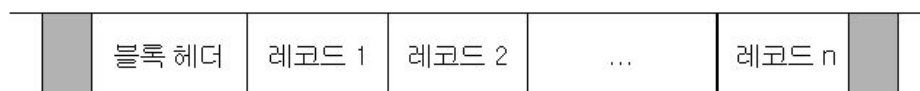


© DBLAB, SNU

## (2) 레코드와 블록

### ◆ 블록 헤더(block header)

- 블록 내의 레코드 수, 블록 ID, 블록 수정 날짜, 블록 내에서의 레코드 시작점과 끝점 식별 등 정보 저장



블록 헤더가 추가된 블록 구조

### ◆ 고정 길이 블로킹

- 레코드 길이만 알면 레코드 구분 가능

### ◆ 가변 길이 블로킹

- i) 레코드 사이에 분리 표시(레코드 끝 마크; end-of-record marker)
- ii) 각 레코드 앞에 길이 지시자(length indicator)
- iii) 블록 헤더에 위치 테이블(position table)

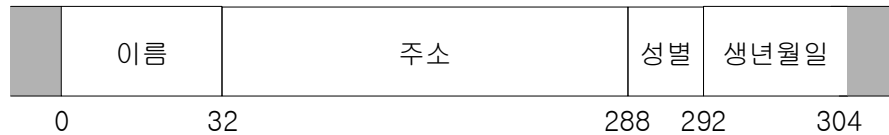


© DBLAB, SNU

## (2) 레코드와 블록

### ◆ 레코드 설계시 고려(제약) 사항

- 메인 메모리 바이트 주소(4의 배수) 특성
- 레코드 헤더
- 각 필드는 레코드의 오프셋(offset)으로 표현 가능



(a) 학생 레코드 레이아웃



(b) 레코드 헤더가 추가된 학생 레코드 레이아웃



© DBLAB, SNU

## (3) 블로킹의 고려 사항

### ◆ 적재 밀도(loading density)

- 갱신(삽입)을 위한 블록 내에 자유 공간의 할당
- 실제 데이터가 저장된 공간과 자유 공간을 포함한 총 공간과의 비율

### ◆ 균형 밀도(equilibrium density)

- 레코드 자체의 확장과 축소
- 상당히 긴 기간 동안 시스템을 운용하고 안정시킨 뒤에 예상되는 저장 밀도

### ◆ 집약성(locality)

- 상호 연관된 레코드들의 물리적 근접성
- 같은 블록, 트랙, 실린더



© DBLAB, SNU



## ❖ 자기 테이프

### ◆ 자기 테이프 (Magnetic Tape)

- 개방 릴 테이프(open reel tape), 테이프 카트리지(tape cartridge)
- 산화 제2철(ferric-oxide)이 입혀진 1.5/1000인치 정도의 얇은 테이프
- 길이 : 300~3,600ft (보통 2,400ft)
- 폭 : 1/2인치, 1/4인치(주로 1/2인치)
- 7-트랙, 9-트랙
- 릴(reel) : 10 1/2"



© DBLAB, SNU

## (1) 자기 테이프 저장장치 구조

### ◆ 자기 테이프 구조



### ◆ 테이프 구동기(tape drive)

- 판독/기록 헤드
- 판독/기록 : 테이프의 자화 표면과 헤드의 자속 (magnetic flux) 사이의 상호 작용



© DBLAB, SNU

## (2) 데이터 표현과 기록 밀도

### ◆ 데이터 표현

- 자화된 방향에 따라 0/1의 값
- 테이프가 기록/판독 헤드를 통과하는 속도 :  
200ips(인치/초)
- 기록 : 충전된 코일에 의해 자화
- 판독 : 자화가 코일에 전류를 유도하여 감지

### ◆ 기록 밀도

- 1인치당 비트수(bpi : bits per inch)
- 1바이트 : 한 컬럼에 수직으로 기록할 수 있는 비트로 이루어진 단위, 보통 8비트로 하나의 문자를 표현
- 9트랙 테이프 : 8트랙의 데이터 + 1트랙의 오류 제어 비트(parity bit)
- 1,600bpi, 6,250bpi, 30,000bpi(최근)



© DBLAB, SNU

## (3) 블록

### ◆ 블록 또는 물리적 레코드(physical record)

- 메모리와 테이프 사이의 데이터 입출력 단위
- 크기: 4000B – 20000B
- 버퍼의 크기를 감안하여 결정
- 블로킹을 이용
  - ◆ 블로킹 인수는 테이프의 활용도와 접근 시간에 영향

### ◆ 블록 간 갭(IBG; inter-block gap)

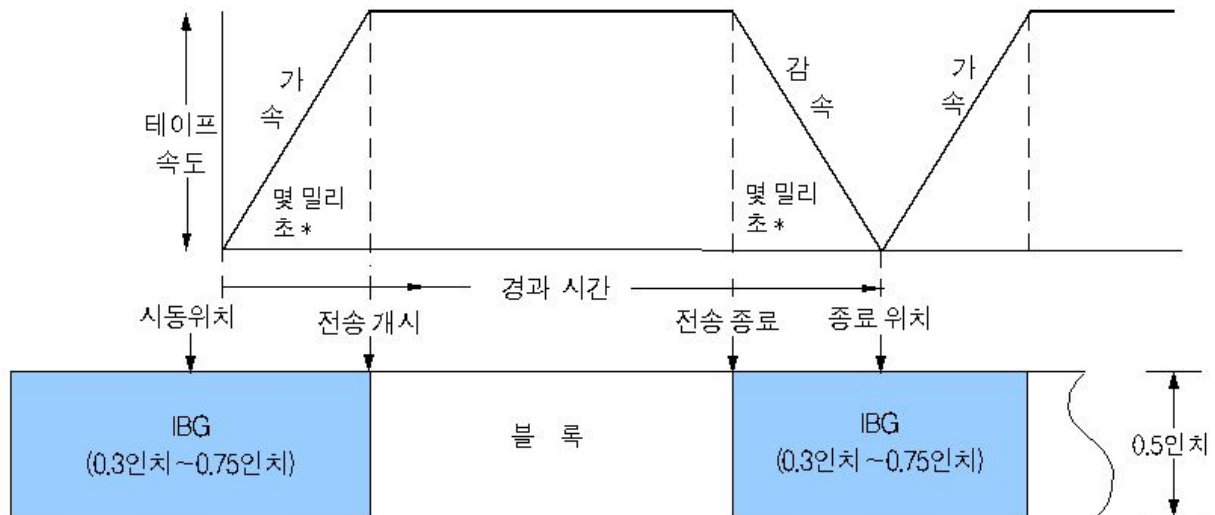
- 테이프 이동 속도 조절 (0.3~0.75inch)
- 가속 시간과 감속 시간 제공



© DBLAB, SNU

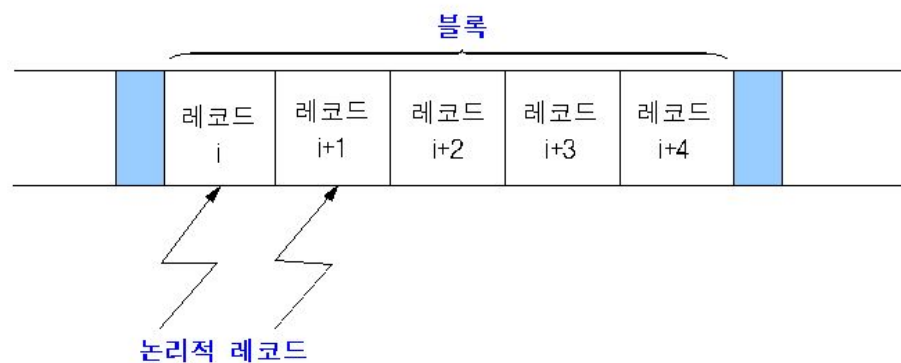
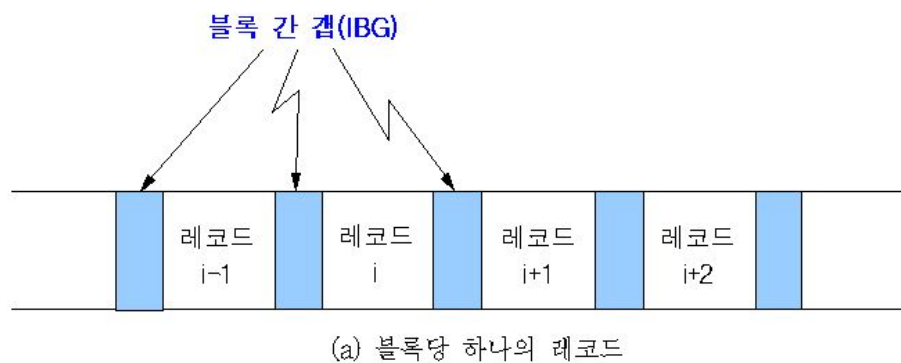
## • 블록 간 갭(IBG)

◆ 가속 시간과 감속 시간을 합한 것이 IBG 시간



© DBLAB, SNU

## • 블로킹



© DBLAB, SNU

## • 테이프에서의 블로킹

### ◆ 6250bpi 테이프에 100B 크기의 레코드 24000개 기록

- 하나의 레코드를 하나의 블록으로 기록
  - ◆  $((100/6250) + 0.3) \times 24000 / 12(\text{인치}) = 632 \text{ ft}$
  - ◆ 632 ft 중에 5%인 32ft( $100/6250 \times 24000 / 12$ )만 실제로 데이터 기록용으로 사용되고 95%인 600ft는 IBG로 사용
- 5개의 레코드를 하나의 블록으로 저장
  - ◆  $(5 \times (100/6250) + 0.3) \times (24000/5)/12 = 152 \text{ ft}$
  - ◆ 152ft 중에 21%만 데이터 저장에 사용되고 79%가 IBG로 사용

### ◆ 테이프의 판독/기록 시간

- 200 inch/sec, 갭을 통해 출발 정지 시간은 0.15초
- 하나의 레코드를 하나의 블록으로 기록한 경우
  - ◆  $((100/6250) \times 24000)/200 + 24000 \times 0.15 = 3601.92\text{sec}$
- 5개의 레코드를 하나의 블록으로 저장한 경우
  - ◆  $((5 \times (100/6250) \times (24000/5))/200 + (24000/5) \times 0.15 = 721.92\text{sec}$



© DBLAB, SNU

## (4) 테이프 카트리지

- 송신 릴과 수신 릴을 하나의 카트리지 내에 장착해 둔 휴대형 소형 매체

### ◆ QIC(quarter-inch cartridge)

- 가장 오래된 형태
- 용량 : 7~70GB, 전송률 : 4~6MBps

### ◆ DAT(digital audio tape)

- 음향 기기용으로 개발
- 폭 : 4mm, 용량 : 4~40GB, 전송률 : 4~6MBps

### ◆ DLT(digital linear tape)

- 대용량 : 10~80GB, 고전송률 : 12MBps



© DBLAB, SNU

## ❖ 광 디스크(1)

### 1. 광 디스크 : CD-ROM

#### (1) 저장 원리 : CD-ROM에 데이터 기록 방법

- ◆ 레이저 빔으로 금속성 표면에 구멍(pit)을 만드는 방법이 보통
- EFM(eight to fourteen modulation) : 1 바이트(8비트)를 14개의 채널비트로 변환하여 저장

#### (2) 물리적 특성

- ◆ 중심에서 바깥까지 약 3마일의 단일 나선형 트랙으로 구성
- ◆ 위치에 관계없이 저장 밀도 일정
- ◆ 중심에서 바깥 방향으로 순차 판독
- ◆ 균일 선형 속도(CLV - Constant Linear Velocity) 방식
  - 섹터가 모두 같은 시간 내에 판독
- ◆ 안쪽 섹터와 바깥쪽 섹터의 회전 속도가 다름
- ◆ 실제 순차 판독시간 77분 , 저장 용량은 650MB



© DBLAB, SNU

## ❖ 광 디스크(2)

#### (3) 섹터

- ◆ 섹터당 2,352B
- ◆ 초당 75개 의 섹터 (150KB/sec) 판독
- ◆ 주소 표현 방식
  - 시작부터 해당 섹터까지의 경과 시간 <분:초:섹터번호>

#### (4) CD-ROM의 장단점

##### ▶ 장점

- 가격이 저렴하고, 대용량(650MB)
- 대량 생산 가능
- 보존성이 좋음

##### ▶ 단점

- 자기 디스크보다 탐구 시간이 길다(회전 속도 가변)
- 임의 접근이 신속하지 못함
- 판독 전용



© DBLAB, SNU

## ❖ 광 디스크(3)

### (5) CD-WO와 CD-MO

#### ▶ CD-WO(write once)

- WORM(Write-Once-Read-Many)의 CD 버전
- 한 번만 기록, 판독은 무한정 가능

#### ▶ CD-MO(magneto optic)

- 광자기 디스크(MOD : Magneto Optic Disk)의 CD 버전
- 소거 가능한 광 디스크
- 백업 장치로 유망

### (6) DVD(digital video disk)

- 양면 사용, 밀도와 저장 방식 개선
- CD의 7배 이상: 4.7~17GByte
- 영화나 비디오 저장 매체로 개발
- 컴퓨터 저장 매체(DVD-ROM Disk)로 이용



© DBLAB, SNU

## ❖ RAID

### ◆ RAID(Redundant Arrays of Inexpensive Disks)

- 디스크의 성능과 신뢰성을 증진시키는 디스크 조직기법
- 많은 수의 저가 소형 디스크를 이용
  - ◆ 고도의 신뢰성
  - ◆ 고속의 데이터 전송
  - ◆ 대용량 데이터 저장
- 많은 수의 디스크를 병렬로 작동
  - ◆ 데이터 판독, 기록 속도 개선
- 다중 디스크에 정보를 중복 저장
  - ◆ 신뢰성 증가



© DBLAB, SNU

## (1) 디스크의 신뢰성(1)

### ◆ 디스크의 장애간 평균시간(MTTF: mean time to failure)

- 단일 디스크 : 100,000시간(11.4년)
- 100개의 디스크 어레이:  
 $100,000/100 = 1000\text{시간}(41.66\text{일})$

### ◆ 신뢰성 문제 해결을 위해 **Mirroring** 기법 (디스크 중복)이용

- 미러링된 디스크 시스템의 장애간 평균 시간은 개별 디스크의 장애간 평균 시간과 수리간 평균 시간(MTTR: mean time to repair)에 의존

- ◆ MTTR: 디스크 교체와 복구시간



© DBLAB, SNU

## (1) 디스크의 신뢰성(2)

예: 단일 디스크의 장애간 평균시간: 100,000시간

수리간 평균시간: 10시간

두 디스크의 장애는 서로 독립적이라는 것을 가정

미러링 디스크 시스템의 데이터 손실간 평균시간(mean time to data loss)은:

$$100000 \times 2 / (2 \times 10) = 500 \times 10 \times 6(\text{hrs}) = 57000(\text{years})$$

- ◆ 보통 평균 55년에서 110년 정도 데이터 손실간 평균 시간을 갖는 미러링 디스크 시스템이 지원



© DBLAB, SNU

## (2) 디스크 병렬성

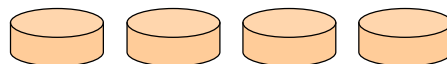
### ◆ Data Striping을 통해서 전송률을 개선

- 데이터를 똑같은 크기로 분할하여 여러 디스크에 차례로 분산 저장
- 스트라이핑 단위(데이터의 분할 크기): 비트, 블록
- 비트 레벨 스트라이핑(bit-level striping)
  - ◆ 각 바이트의 비트를 여러 디스크에 분할 저장
  - ◆ 4개의 디스크로 구성된 디스크 어레이
    - 각 바이트의 비트  $i$ 를 디스크  $(i \bmod 4)+1$ 에 기록
    - 블록이 4배 크고 전송 속도가 4배 빠른 성능의 디스크 역할
- 블록 레벨 스트라이핑(block-level striping)
  - ◆ 파일의 블록들을 여러 디스크에 차례로 분산 저장
  - ◆  $d$  디스크 어레이
    - 파일의 블록  $i$ 는 디스크  $(i \bmod d)+1$ 에 저장
  - ◆  $d$ 개의 디스크로부터  $d$ 개의 블록을 병렬로 판독할 수 있어서  $d$ 배의 전송률을 제공

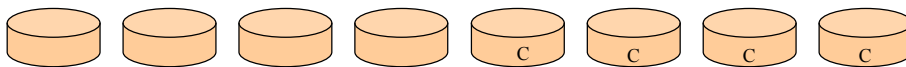


© DBLAB, SNU

## (3) RAID 레벨 (1)



(a) RAID 0 : 비중복 스트라이핑



(b) RAID 1 : 미러드 디스크(copy)



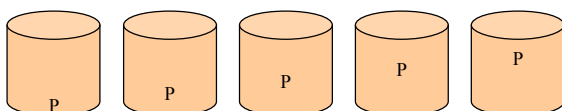
(c) RAID 2 : 메모리-스타일 에러 교정 코드(p)



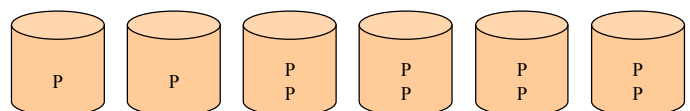
(d) RAID 3 : 비트-인터리브드 패리티



(e) RAID 4 : 블록-인터리브드 패리티



(f) RAID 5 : 블록-인터리브드 분산 패리티



(g) RAID 6 : P+Q 추가



© DBLAB, SNU



### (3) RAID 레벨 (2)

#### ◆ RAID 레벨 0: 비중복 스트라이핑

- 미러링이나 패리티 비트와 같은 추가 정보를 유지함이 없이 데이터 스트라이핑 기법을 사용
- 디스크 드라이브의 수가 증가함으로써 MTTF가 감소하게 되어 신뢰성이 약함
- 추가 정보가 없기 때문에 기록 성능은 높음

#### ◆ RAID 레벨 1: 미러드 디스크

- mirrored disk 구조
- 데이터 스트라이핑이 없이 하나의 블록은 두 개의 상이한 디스크에 저장하여 미러드 디스크 유지
- 기록은 기본 디스크에 먼저 수행한 뒤 미러 디스크에 수행
- 데이터 판독은 두 디스크에 분산시켜 수행하기 때문에 효율적임



© DBLAB, SNU

### (3) RAID 레벨 (3)

#### ◆ RAID 레벨 2: 메모리-스타일 에러 교정 코드

- 메모리-스타일(memory-style) 에러 교정 코드 (error-correcting code: ECC) 구조
- 비트 레벨 스트라이핑
- 에러 교정 비트를 저장하기 위한 추가 디스크를 사용
  - ◆ Hamming code 기법 사용 - 어떤 디스크에서 에러가 일어났는지 장애 디스크 식별 가능
- d개의 데이터 디스크로 구성된 디스크 어레이에서 최소 데이터 전송 단위는 d블록이 되기 때문에 대량 데이터 요구 작업 환경에 유리

#### ◆ RAID 레벨 3: 비트-인터리브드 패리티

- bit-interleaved parity 구조
- 하나의 패리티 비트만 저장하여 사용하기 때문에 하나의 디스크만 추가로 사용
- 디스크 제어기의 기능을 활용하여 에러 교정
- 손실 데이터 회복에 중점
- 가장 경제적인



© DBLAB, SNU

### (3) RAID 레벨 (4)

#### ◆ RAID 레벨 4: 블록-인터리브드 패리티

- block-interleaved parity 구조
- 블록 레벨 스트라이핑
- 데이터 블록에 대한 패리티 블록들은 하나의 디스크에 별도로 유지
- 대규모 데이터에 대해서는 전체적 입출력 성능이 높음
- 소규모의 여러 독립적 기록 연산은 병렬로 수행할 수 없음

#### ◆ RAID 레벨 5: 블록-인터리브드 분산 패리티

- block-interleaved distributed parity 구조
- d개의 데이터 디스크와 1개의 패리티 디스크 대신 데이터와 패리티를 d+1개의 디스크에 균등하게 분산 저장
- 데이터 블록과 이 블록의 패리티 블록은 반드시 상이한 디스크에 저장
- 기록과 판독이 병렬로 수행
- 추가 정보를 유지하는 RAID 레벨 중에서 성능이 가장 높음



© DBLAB, SNU

### (3) RAID 레벨 (5)

#### ◆ RAID 레벨 6: P+Q 추가

- 대규모 디스크 어레이에 사용
- 하나가 아니라 두 개의 디스크가 동시에 장애가 일어났을 때도 회복 가능
- Reed-Solomon code를 사용하는 P+Q redundancy scheme을 사용
- 매 4비트마다 2 비트를 추가로 저장
- 패리티 정보는 블록 레벨로 디스크에 균등 분산 저장
- 하나의 데이터 블록을 갱신하기 위해서는 패리티 정보(P와 Q)를 가진 두 개의 블록도 갱신



© DBLAB, SNU

## (4) RAID레벨의 선택

### ◆ RAID 레벨 0

- 데이터 손실이 중요하지 않은 고성능 응용에서 사용

### ◆ RAID 레벨 1

- 디스크 고장시 데이터 재구성이 용이
- 최고의 기록 성능 제공 - 데이터베이스의 로그 파일 저장에 이용
- 높은 오버헤드 때문에 레벨 3이나 5를 선호

### ◆ RAID 레벨 2와 4

- 레벨 3과 5에 포함

### ◆ RAID 레벨 3과 5

- 레벨 3: 고도의 데이터 전송이 필요할 때
- 레벨 5: 대부분의 데이터베이스 시스템에서와 같이 임의 판독이 중요할 때

### ◆ RAID 레벨 6

- 고도의 신뢰성이 요구되는 환경

