Welcome to CS245A

The menu for this quarter:

1. Storage Systems:
    a) Physical devices and characteristics, especially disks.
    b) Logical layout of data; data structures, especially, indexed-sequential files, B-tree, hashing.
2. Query optimization:
    a) Query-plan generation; algebraic transformations.
    b) Join methods.
3. Resilience:
    a) Logging.
    b) Authorization and encryption.
4. Transaction processing:
    a) Serialization, deadlocks, locking, timestamping.
    b) Distributed DB's.

H. G.-M.'s Strawman DBMS

The MEGATRON 2000 DBMS

- The latest from Megatron Labs.
- Incorporates the latest relational technology.
- UNIX compatible.

Implementation Details

- Relations stored in Files, e.g., relation *Students* in `/usr/db/Students`:

```
Smith#123#CS
Johnson#522#EE
...
```

- DB schema stored in `/usr/db/schema`:

```
Students#name#STR#id#INT#dept#STR
Depts#name#STR#office#STR
...
```

Sample Session

```
Elaine31> megatron2000

    WELCOME TO MEGATRON 2000!
&
...
& quit
Elaine31>
```

```
&   SELECT *
    FROM Students #
```

name	id	dept
Smith	123	CS
Johnson	522	EE

```
&   SELECT *
    FROM Students | lpr #
```

(result sent to printer)

```
&   SELECT *
    FROM Students
    WHERE id > 500 | R #
```

(new relation R created)

Query Execution

For `SELECT * FROM R WHERE condition`:

1. Read schema file to determine attributes of R and their types.
2. Read R file and for each line:
    a) Check condition.
    b) Display if true.

To execute `SELECT * FROM R WHERE condition | T`:

1. Process query as before.
2. Write result to a new file T.
3. Add entry for T to the `schema` file.

To execute:

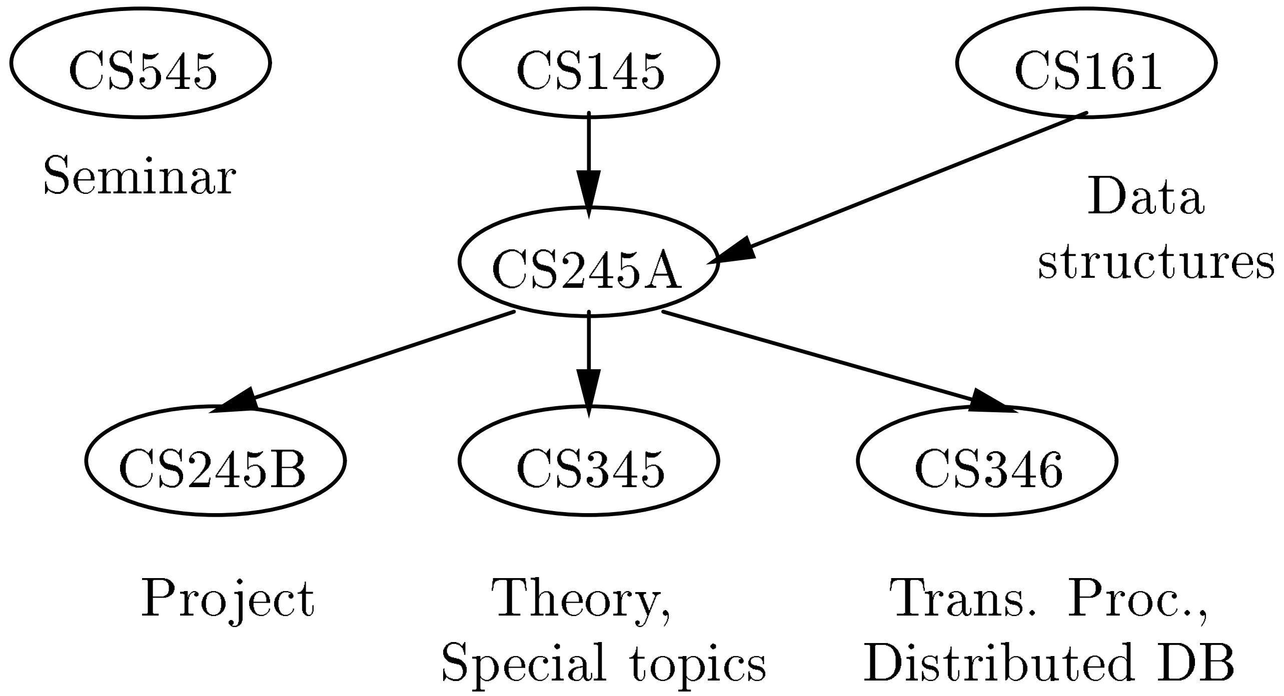
```
SELECT office
FROM Students, Depts
WHERE Students.name = 'Smith' AND
    Students.dept = Depts.name #
```

1. Join `Students`, `Depts` in nested loops.
2. Apply condition to each joined tuple.
3. Display if true.

What is Wrong with MEGATRON 2000?

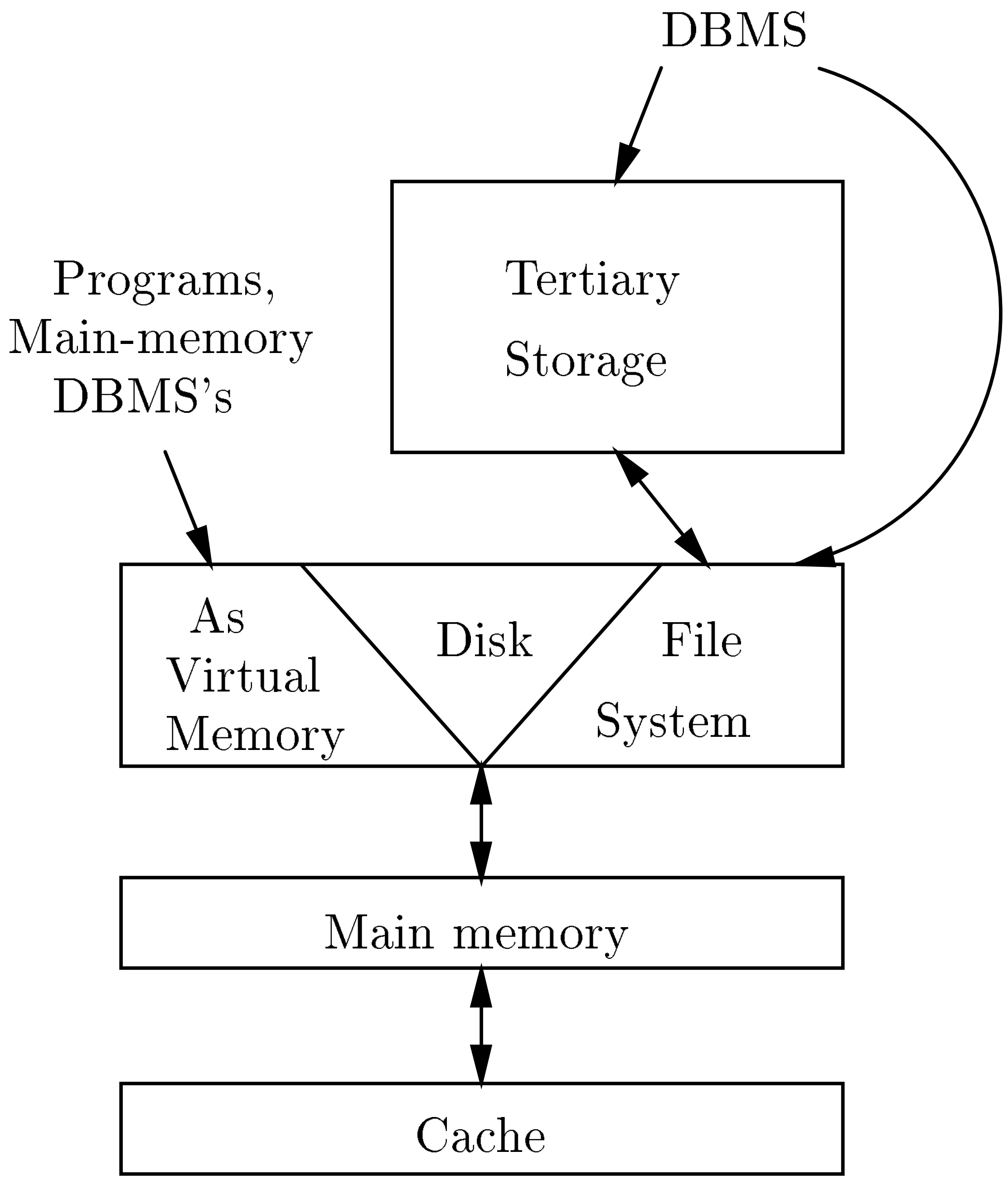
- Tuple layout on disk: no flexibility for DB modifications.
  - Change `EE` to `ECON` and the entire file has to be rewritten.
- Search Expensive: no indexes; always read entire relation.
- Brute-force query processing.
  - Did we need to look at all pairs of student-dept tuples?
- No buffer manager: everything comes off of disk all the time.
- No concurrency control: several users can modify a file at the same time with unpredictable results.
- No reliability: can lose data in a crash or leave operations half done.
- Little security: file system protection too coarse.

Stanford DB Course Structure



- Important prerequisites:
  - From CS145: SQL, ODL/OQL, Relational algebra, etc..
  - From CS161: linked lists, hashing, trees, balanced trees, sorting, etc.

The Memory Hierarchy



- Cache: fastest, perhaps 1Mb.
- Main memory: under a microsecond, random access, perhaps 64Mb.
- Secondary storage:
  - Typically magnetic disks, magneto-optical (erasable), CD-ROM.
  - Access times in milliseconds, great variability.
  - Unit of read/write = *block* or *page*, typically 4Kb.
  - Capacities in gigabytes.
- Tertiary storage:
  - Desired data carried to read/write port, access times in seconds.
  - Most common: racks of tapes; newer devices: CD-ROM "juke boxes," tape "silo's."
  - Capacities in terabytes.

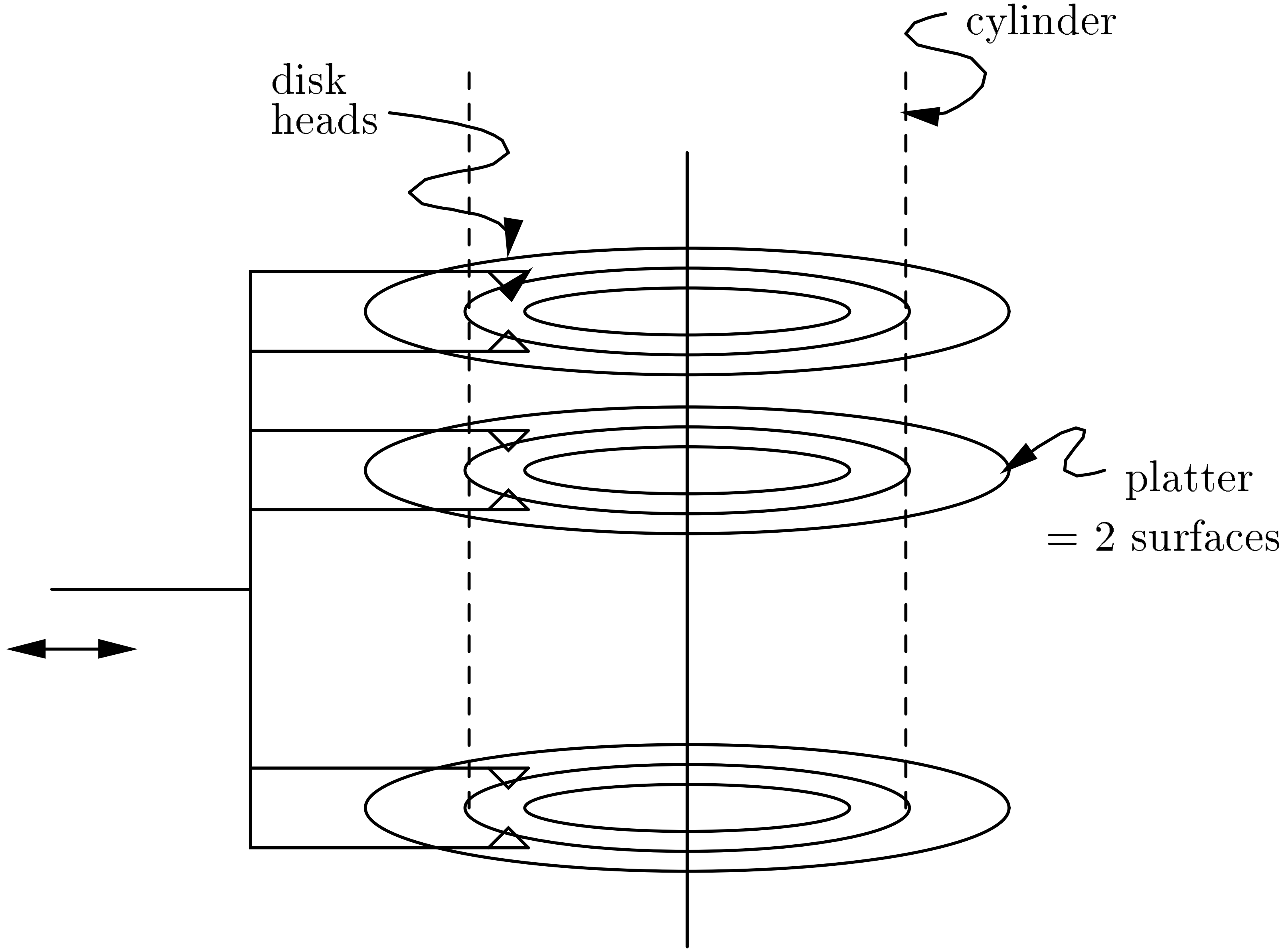
Use of Secondary Storage

- Common use is to store *pages* from the (virtual memory) address space of an application program.
  - 32-bit address space means 4Gb of data.
  - Enough for most purposes, but not for large DB.
  - *Main-memory DB*: specialized DBMS that operates in virtual memory.
- *File system.* File = collection of blocks holding logically related data.
  - Common: file = big character string.
  - DB: file = relation or extent of a class; units of the file are records representing tuples or objects.

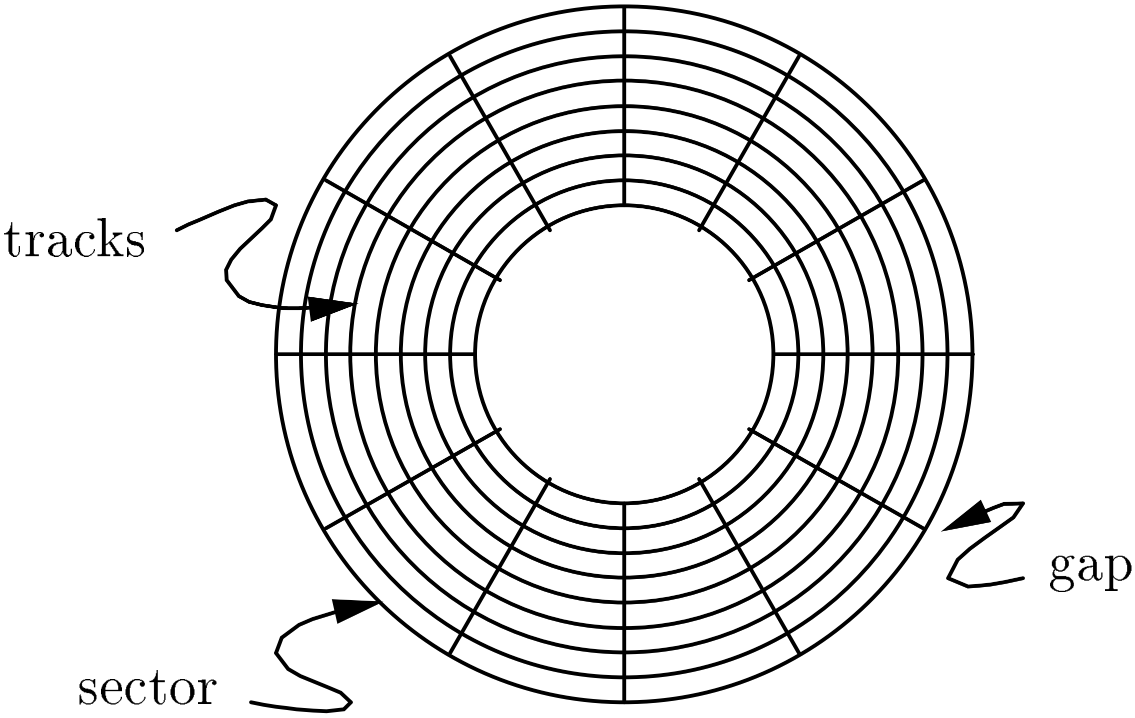
(Non)Volatile Storage

- A storage device is *nonvolatile* if it can retain its data after a power shutoff.
  - ❖ Issue is important because DBMS's cannot allow data to be lost, ever.
- Disks and tapes of all types are nonvolatile.
- Main memory is typically volatile — charge drains out of simple memory circuits.
- But there are "RAM disk" devices that can survive a power outage by virtue of a backup power supply.

Disks



- Platters with top and bottom *surfaces* rotate around a spindle.
  - Diameters 1 inch to 4 feet.
  - 2–30 surfaces.
  - Rotation speed: 3600–7200 rpm.
- Head per surface.
  - All heads move in and out in unison.



- Surfaces covered with concentric *tracks.*

    - Tracks at a common radius = *cylinder.* Important because all data of a cylinder can be read quickly, without moving the heads.

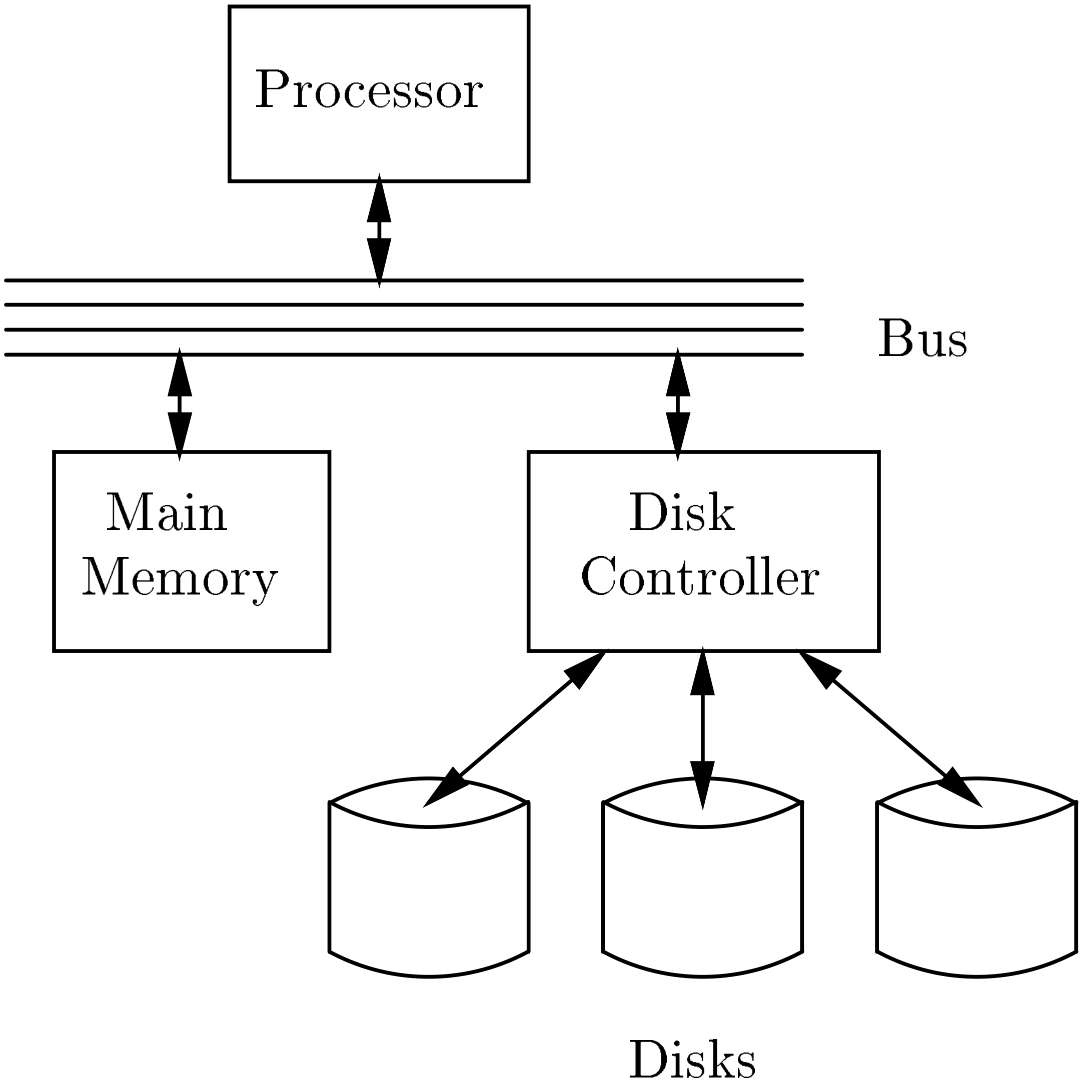
    - Floppy disk: 40 cylinders; typical magnetic disk: 10,000 cylinders (optical disk: several times that).

- Tracks divided into *sectors* by unmagnetized *gaps.*
  - Typical track: 32–512 sectors.
  - Typical sector: 512 bytes.
- Sector is unit of error correction/detection.
  - Parity-check bit chosen so number of 1's is even. Thus, single errors detectable.
  - A bad sector is "cancelled" by the *disk controller*, so it is never used.
- Sectors are grouped into blocks.
  - Typical: one 4K block = 8 512-byte sectors.

MEGATRON 747 Disk

- Four platters, eight surfaces.
- Diameter = 3.5 inches; only outer 1 inch is occupied by tracks.
- $2^{13} = 8192$ cylinders.
- Average 256 sectors/track (fewer near center, more near edge, so average bit density doesn't vary too much).
- 512 bytes/sector.
- Capacity = 8 surfaces $\times$ 8192 tracks/surface $\times$ 256 sectors/track $\times$ 512 bytes/sector = 8Gb.

Disk Controller



1. Buffer data in and out of disk.

2. Schedule the disk heads.

3. Manage the “bad blocks” so they are not used.

Accessing Disk Blocks

Four principal components of access time:

1. *Seek time* = time to move heads to proper cylinder.

2. *Rotational latency* = time for desired block to come under the head.

3. *Transfer time* = time during which block passes under head.

4. *Miscellaneous*, including time for disk controller to process data, contention for controller, bus, etc.

    - Neglect this.

MEGATRON 747 Timing Example

- Megatron 747 rotates at 3840 rpm = 64 times/second.
  - Average rotational latency = 7.8 milliseconds; max = 15.6 ms.
- Seek time = 1 ms + 1 ms per 500 cylinders traveled.
  - 0 seek time if block is on same cylinder. 1.002 ms to move one cylinder, 17.4 ms to move across all cylinders.
  - **Problem:** what is the average seek time?
- Gaps take up 10% of the tracks.
  - Transfer time for a block is thus time for 8 sectors and 7 gaps to pass head.
  - These take an average of 11 degrees (approximately 8/256 of a circle).
  - Transfer time = 0.5 ms.
- Min time to read/write a block = 0.5 ms.
- Max = 0.5 + 15.6 + 17.4 = 33.5 ms.

Writing and Modifying Blocks

- Writing same as reading, unless we verify written blocks.
  - ✦ Takes another rotation of disk.
  - ✦ Verify by observing parity check bits.
  - ✦ Try again if fail (same for reading a block).
- Modifying a block requires:
  1. Read the block into main memory.
  2. Modify the block there.
  3. Write the block back to disk.