CSSE2010

CSSE2010/CSSE7201
Lecture 22

Memory & Disks ✓

School of Information Technology and Electrical Engineering
The University of Queensland

Outline

- Admin
- Computer Memory
  - Different types of memory
  - Memory Hierarchy
- Disks

Admin

- Quiz 9 from last week: 336 attempts, average 3.9/5 and median 4/5.
- Quiz 10 (the last one) is due this week's Friday 22/10/21 4:00PM AEST
- Lab Assignment 2
  - There have been some corrections posted to the specification. Please take note.
  - More tutoring staff has been allocated for week 12 and 13 to give you extra help.

Reminder

- SECaT Course and Teaching Evaluation now open
  - There are two surveys: one is for the course and one is for teaching.
  - Surveys can be accessed at https://eval.uq.edu.au
  - Please take some time to provide feedback on how it all went for CSSE2010/7201 this semester
  - The surveys close on 11.59PM Friday 5th November (revision week)

From Lecture 9

Computer Memory

- Computers need memory to
  - Store temporary results
  - Store programs
  - Remember settings when power off
- Many different types
  - Random access
  - Static vs. dynamic
  - Volatile vs. non-volatile
  - Read-only
  - Primary vs. secondary

Memory Types

- **Static** memory SRAM , g. reg.
  - Flip-flops or latches used to store bits
- **Dynamic** memory
  - Each cell is a switch (transistor) plus capacitor
- These are both types of RAM = **R**andom **A**ccess **M**emory
  - "RAM" originally meant could access any cell
    - (Compared with sequential access)
  - Now means volatile read/write memory
  - Static memory = SRAM ← AVR data mem 2KB SRAM
  - Dynamic memory = DRAM

Volatility

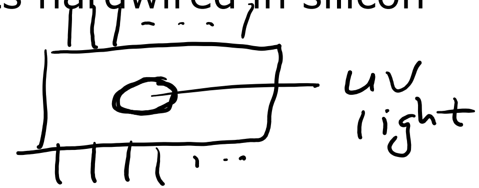
- **Volatile** memory
  - Contents "forgotten" when power off
  - Both SRAM and DRAM are volatile
    (DRAM forgets contents within milliseconds if not refreshed, i.e. contents rewritten)
- **Non-volatile** memory
  - Contents remembered when power off
  - Many types
    - ROM
    - EPROM
    - EEPROM (*)
    - Flash (*)

AVR 32KB flash

EEPROM ← Non ~~vola~~ volatile data mem.

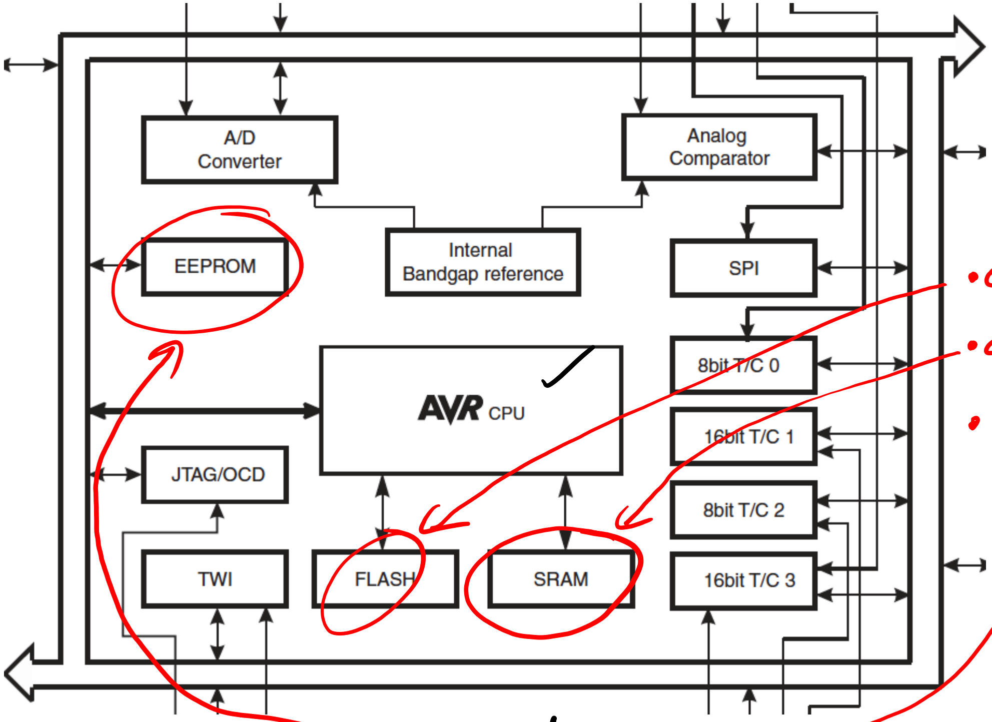
Non-volatile memory chips

- **ROM** = Read Only Memory
  - Programmed by memory manufacturer – contents hardwired in silicon
- **EPROM** = Erasable Programmable ROM
  - Can erase, e.g. with ultraviolet light



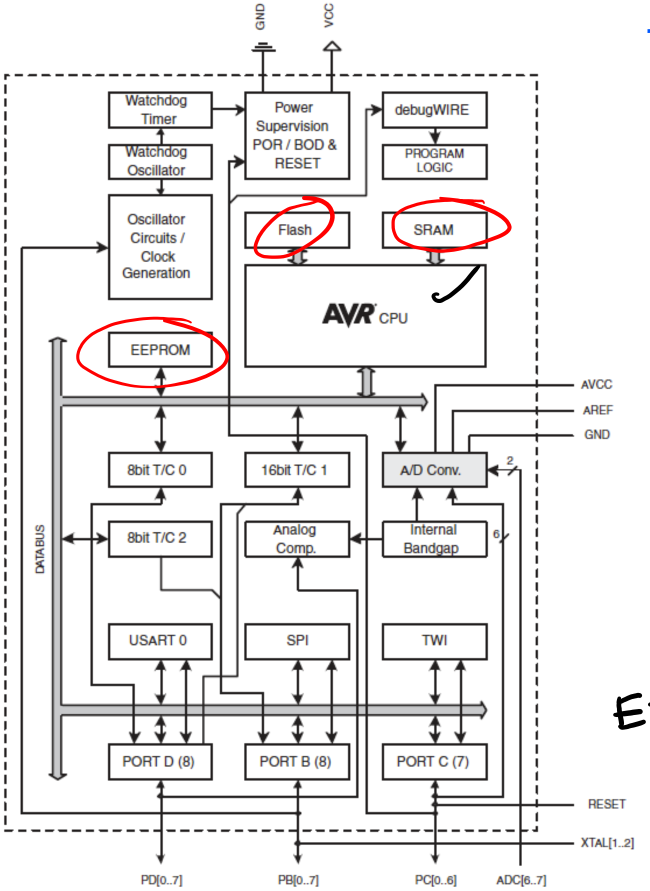
- **EEPROM** = Electrically Erasable PROM
  - Can erase by applying a certain voltage
- **Flash** = New EEPROM
  - (Used to be different technology to EEPROM but converging)
  - Main difference – Flash memory written/erased in blocks, EEPROM written/erased one cell at a time
    - ATMega324A supports two flash blocks – Application section + Boot Loader section
    - Other AVR devices have one, others have more

ATmega324A/328P Memory



•cseg
•dseg
•eseg

IN



Ex

When is a Megabyte not a Megabyte?

- When it's a million bytes
- Conventionally (or at least historically)

  $1\ \text{MB} = 2^{10}\text{kB} = 1{,}024\text{kB} = 2^{20} = 1{,}048{,}576\text{B}$

  - Sometimes, $1\text{MB} = 10^{6}\text{B} = 1{,}000{,}000\text{B}$
  - Or could be in between: $1000\text{kB} = 1{,}024{,}000\text{B}$
- Conventionally

  $1\ \text{GB} = 2^{10}\text{MB} = 2^{20}\text{kB} = 2^{30}\text{B}$
  $= 1{,}073{,}741{,}824\text{B}$

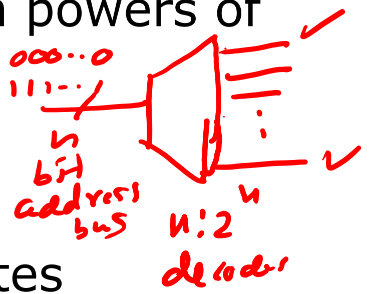
  - Sometimes, $1\text{GB} = 10^{9}\text{B} = 1{,}000{,}000{,}000\text{B}$
  - This is a 7.4% difference
  - Or could be in between: e.g. $1000\text{x}2^{20} = 1{,}048{,}576{,}000$

10^3 2^{10}

TB
GB
10^6 MB 2^{20}
KB 2^x
10^x

When do we use powers of 2? When do we use powers of 10?

- For **main memory** capacity, always based on powers of 2, e.g.
  - 1GB = 1024 x 1024 x 1024 bytes = 2^{30} bytes
- For **hard disks**, manufacturers base sizes on powers of 10, e.g.
  - 1GB = 1000 x 1000 x 1000 bytes = 10^{9} bytes
  - (Makes hard disks seem bigger)
  - Operating Systems (e.g. Windows) base sizes on powers of 2
- For **data speeds (transfer rates)**, usually based on powers of 10, e.g.
  - 1Gbps (gigabit per second) = 10^{9} bits per second
- Occasionally, a mixture
  - 1.44MB floppy disk stores 1.44 x 1000 x 1024 bytes



$\sim 10^3$ $\sim 2^{10}$

Kibibytes, Mebibytes

- Kilo-, mega-, giga- prefixes mean powers of 10 in SI units
- To avoid confusion, standardised terms have been created for binary prefixes, e.g.
  - 1 **kibibyte** = 1KiB = 1024 bytes 2^{10}
  - 1 **mebibyte** = 1MiB = 1024KiB 2^{20}
  - 1 **gibibyte** = 1GiB = 1024MiB 2^{30}
- Unfortunately, very few people use them
- We'll use kilo-, mega-, giga- etc. but meaning could vary depending on context

Types of DRAM

- Bits stored as charge on capacitor, not in flip-flops
- Cells are smaller, so can pack more bits on a chip than for static memory
- Slower than static memory
- Many different types of DRAM over the ages…
  - Fast Page Mode (FPM DRAM)
  - Extended Data Out (EDO DRAM)
  - Synchronous DRAM (SDRAM)
  - Rambus DRAM (RDRAM)
  - Double Data Rate DRAM (DDR DRAM) 1, 2, 3 and 4

What is the approximate cost per <u>Gigabyte</u> of DRAM?

- 0% A. $0.01
- 23% B. $0.10
- 27% C. $1.00
- 41% D. $10.00
- 5% E. $100.00
- 0% F. $1,000.00
- 0% G. $10,000.00
- 5% H. $100,000.00

16 GB $150
RAM

0

What is the approximate cost per Gigabyte of Flash memory?

- 0% A. $0.01
- 42% B. $0.10
- 38% C. $1.00
- 13% D. $10.00
- 4% E. $100.00
- 0% F. $1,000.00
- 0% G. $10,000.00
- 4% H. $100,000.00

512 GB $80
SSD

0

What is the approximate cost per Gigabyte of Hard Disk Storage?

50%	A.	$0.01
41%	B.	$0.10
5%	C.	$1.00
5%	D.	$10.00
0%	E.	$100.00
0%	F.	$1,000.00
0%	G.	$10,000.00
0%	H.	$100,000.00

0

What is the approximate cost per Gigabyte of SRAM?

- 5% A. $0.01
- 0% B. $0.10
- 32% C. $1.00
- 27% D. $10.00
- 32% E. $100.00
- 5% F. $1,000.00
- 0% G. $10,000.00 *
- 0% H. $100,000.00

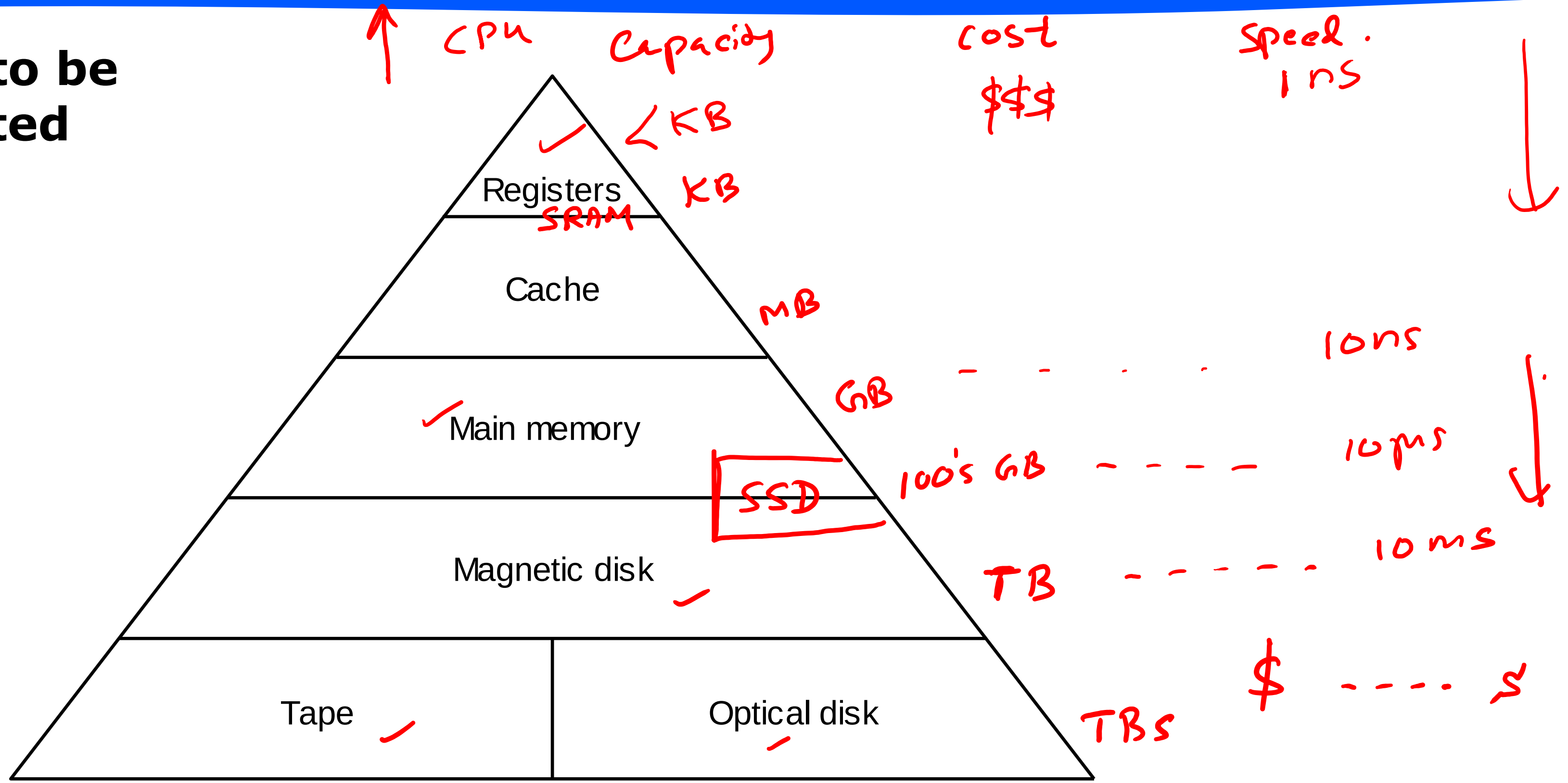
16MB $18

$9000 1GB

0

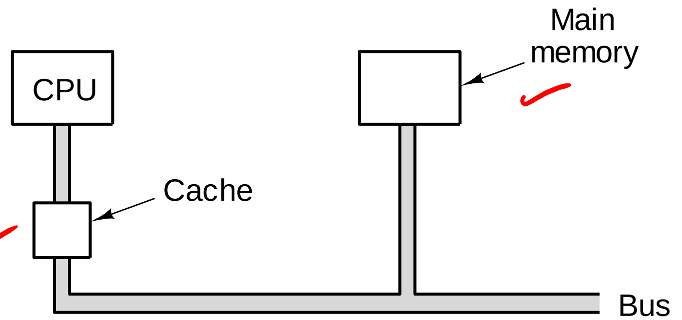
Memory Hierarchy

Figure to be annotated in class



Cache Memory

- CPUs are faster than main memory
  - Very fast memory is expensive
- Performance suffers if CPU always waiting for memory
- **Cache**
  - Small amount of very fast memory combined with larger slow memory
  - Most commonly used memory words kept in cache
- CPU looks in cache before main memory
- Average access time greatly reduced if many words are in the cache



for {
}

Cache Levels

- Often multiple levels of cache
- **Level 1** (L1)
  - Inside CPU itself
  - e.g. 32kB to 128kB
  - Runs at same speed as CPU
- **Level 2** (L2)
  - May be separate chip (possibly inside same module) or on same chip as CPU
  - e.g. 256kB to 16MB
  - May be slightly slower than CPU
- **Level 3** (L3)
  - Some machines have a third level (e.g. 2MB to 256MB)

Note:
B = byte
b = bit

“Cache” for Disks

- Increasingly common
- Small fast solid state drive (Flash memory) caches content from larger slower magnetic disk

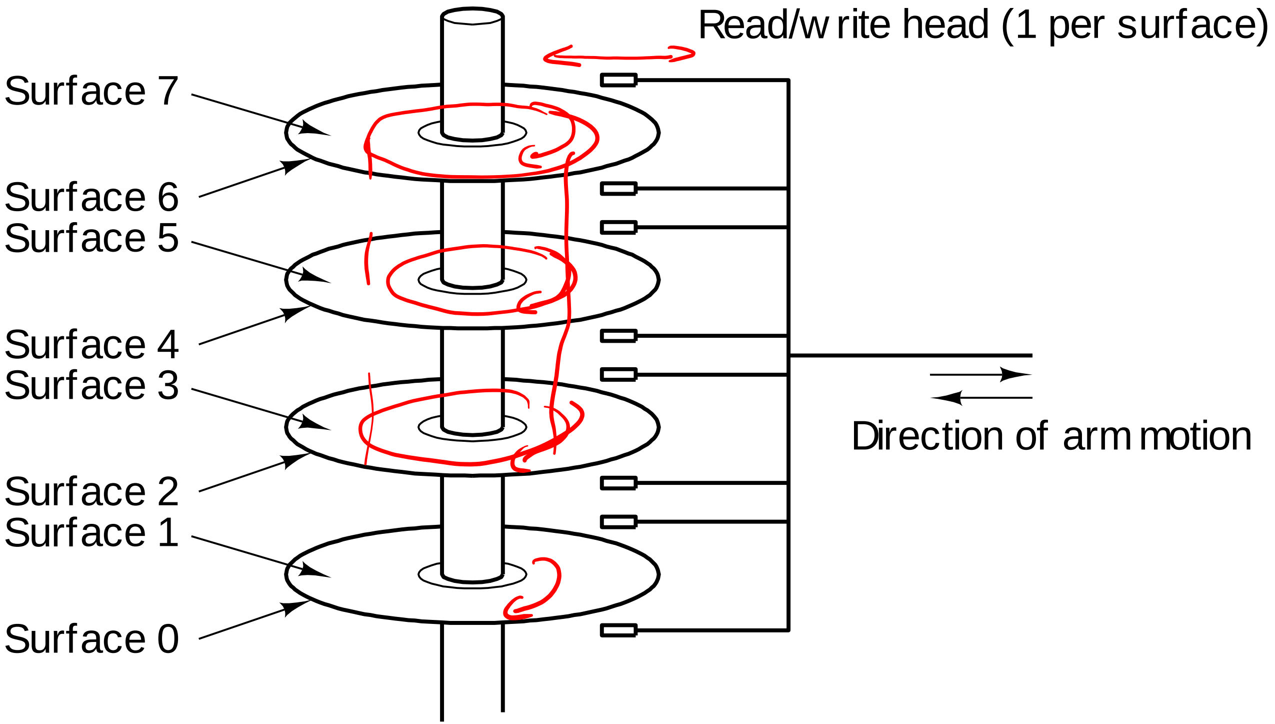
Magnetic Disks

- Rotating platters with magnetised coating
  - Stack of platters
- Data stored magnetically in circular tracks
- Read/write heads float above platter surfaces
- Usually use both sides of platters

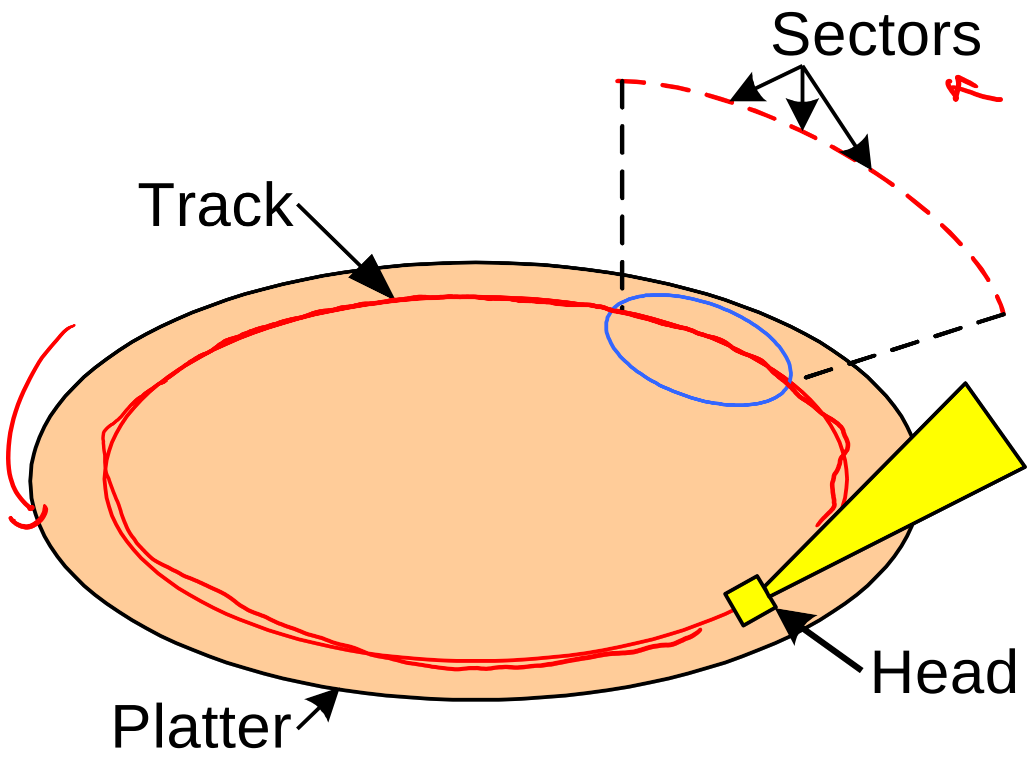
1

0

Magnetic Disks – Simplified view



Sectors and Tracks

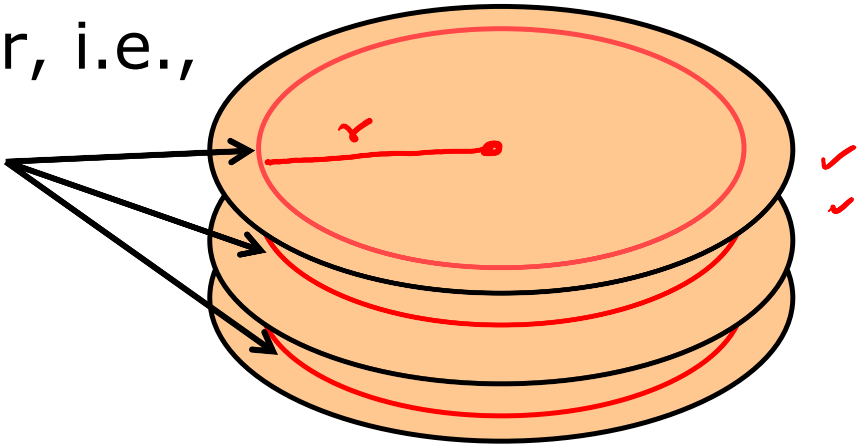


← smallest block read/written.

4096 bytes.

Cylinders

- Set of tracks at a given radial position
- Number of tracks in cylinder =
  2 x number of platters
  (assuming both sides of all platters used)
- Heads move together, i.e., to a certain cylinder

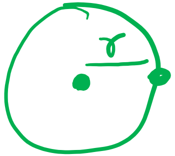


CSSE2010

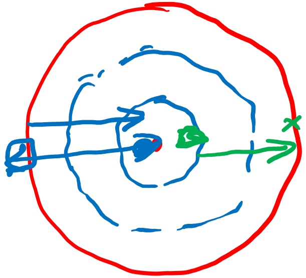
Sectors

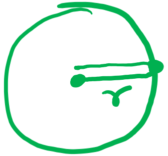
- Tracks divided into fixed-length sectors — 4096 bytes.
  - Smallest data unit, i.e. must read/write a whole sector at a time
- Sector consists of:
  - Preamble 1010101010 - - -.
    - Allows head to synchronise to data
  - Data ← 4096 bytes. ← 85% of the quoted capacity
  - Error correction codes (ECCs)
  - Inter-sector gap
- About 85% of disk capacity usable by operating system
  - About 95% of this usable for user data
  - Remaining ~5% file system overhead

Access Time



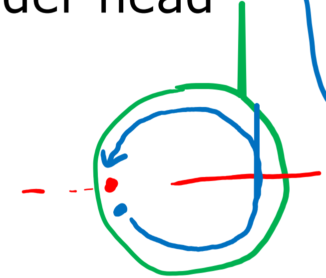
- **Access time** =
  - *seek time* +
  - *rotational latency* +
  - ~~*transfer time*~~ ← neglected.





→ average seek time = $\frac{1}{3}$ [edge to edge seektime]

- 1 **Seek time**
  - Time to move heads to right cylinder
- 2 **Rotational latency**
  - Time to wait for sector to arrive under head
- 3 **Transfer time**
  - Time for sector to pass under head
  - Negligible compared with above



avg rotation latency = $\frac{1}{2}$ · rotation time

Calculating Access Time

- Example:
  What's the *average* access time for a hard disk which rotates at 7200rpm and has an edge-to-edge seek time of 10ms? (Assume that seek time is proportional to the number of tracks to seek.)

120 rps

Avg access time = Avg seek time + Avg rotational latency

$$= \frac{1}{3}\left[\text{edge to edge seek time}\right] + \frac{1}{2}\left[\text{rotation time}\right]$$

$$= \frac{1}{3}\left[10\text{ ms}\right] + \frac{1/120}{2}\text{ s} \Rightarrow \underline{\underline{7.5\text{ ms}}}$$

✓