

Basics Of Storage

→ Overview

→ Storage Basics / Hierarchy

→ RAM, ROM, HDD

→ Caching Concepts

Base

Question

Results

✱ Extras/Doubts

High level languages
HLD
LLD
Implementation

← Application Layer

Code

...



Lower level languages

Compilers, Linkers
→ Assembly

OS
mngt

0/1 bytes → Binary Language

Databases

↑ Hardware → Registers
Flip Flops
Electronics
X ————— X

A + B

Cost/Step ↑

Processor

Bits ← Registers

Temp Storage (Very Fast) → inside Proc

kB ~ MB

Internal Cache (Very Fast) Temporary → Feeds Proc ~ match Processing speed
(SRAM)

RAM (Main Memory) 8-32-128 GB (Fast)

(DRAM) | ↗

1
HDD/SSD (Storage) (Slow) (1TB)

↳ External Storage

Storage vs RAM

Volatile

Persistent

RAM → Volatile storage

→ Non Persistent

HDD → non volatile

SSD → Persistent

[Storage]

↓ Slow

Cheaper

Larger Capacity

Database → stored in Storage

→ Redis / Memcache → RAM, generally used for

speed / caching & not

It uses backups

↪ persistence

like snapshot, AOF etc

ROM

Read Only Memory

write once read forever

Master Boot Record

↳ GRUB

↓

ROM

PRoM → Programmable ROM

↳ write once, read forever

EPROM → Erasable PROM

↳ To erase, UV light + loses

✓ EEPROM → Electrically EPROM

↳ EPROM but electrically erasable

RAM

Random Access Memory

0	1	2	3
α	A	F	E
β	B		C
γ	D	1	2
	1	2	3
		3	

0x FFO0 CAB16

↳ Hexadecimal

→ A - F

°

Give me F

↓

Give me F directly in

min time

OS → will know where

F is

↓

OS ° F → 2

E → 3

A → 1

RAM → volatile RAM → static RAM → Flip Flops → no need to constantly recharge

volatile d RAM → dynamic RAM → Capacitors

Low V → 0

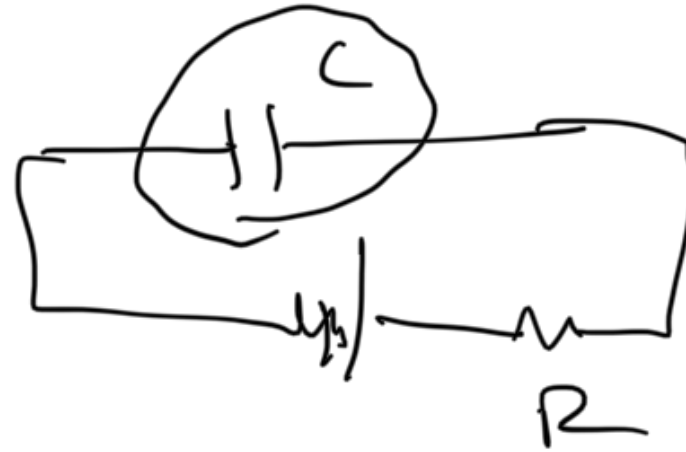
High V → 1

$e^{-\alpha t}$

100s of times/s

→ recharges C

↓
Slow (ev)



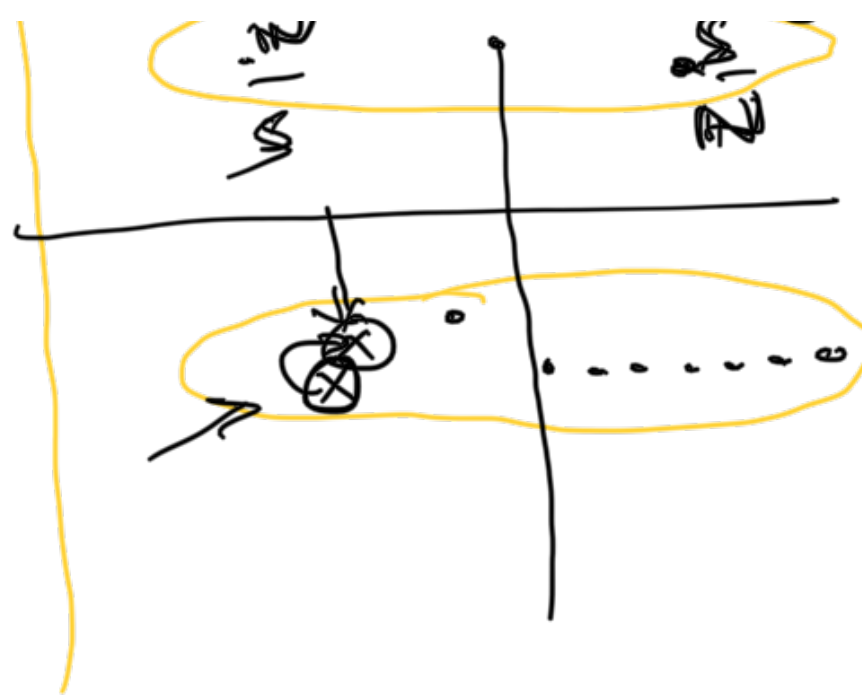
HDD (Hard Disk Drive)

SSD → Solid State Drive

non volatile
mechanical movement
- slow



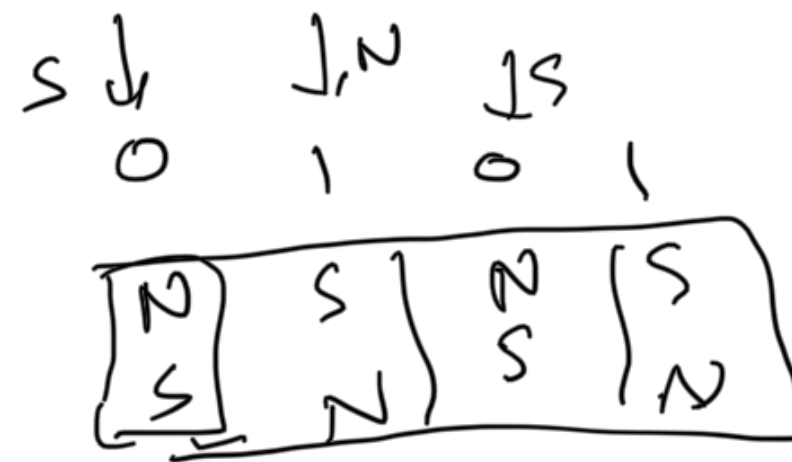
Head → Electro Mag



~~(Cylinders)~~

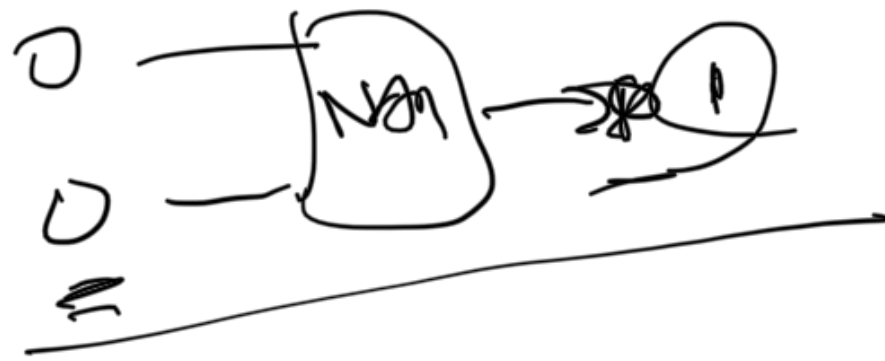


Disc Top View


$$\begin{array}{c} \mathbb{Q} \rightarrow \mathbb{N} \\ \quad \downarrow \\ \quad \mathbb{S} \end{array}$$

SSD/Flash → they don't have moving parts
↳ Floating Gate Transistor

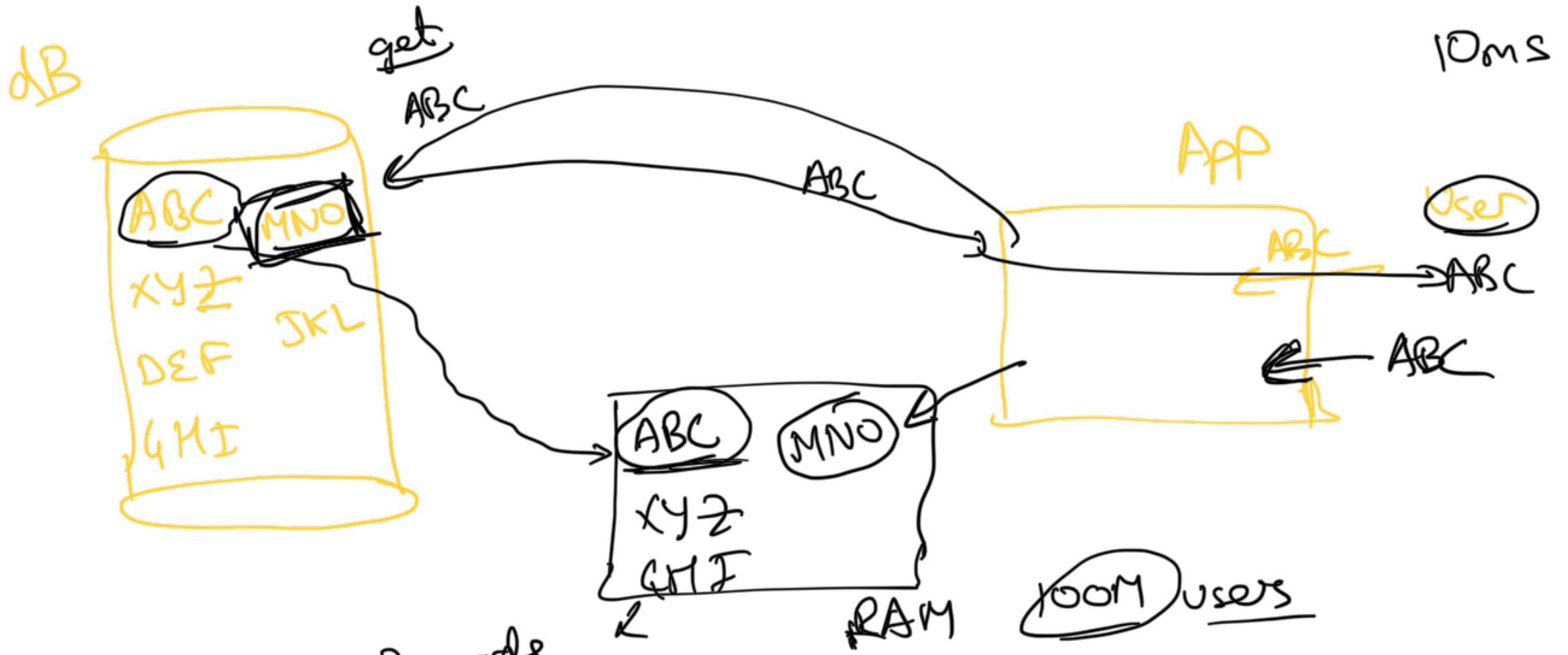
→ NAND/NOR Codes



Transistors

Caching Concepts

Redis / Memcached → dB in RAM



α ' Hey ABC is a cool word &

80M JB Fans

xy \Rightarrow

60M

10ms \rightarrow 180 μ s

Keep important data

closer to user \rightarrow Caching

① How to choose what to cache?

* ② When cache is full, then what?

Prior knowledge

① • Configs

13 million/min

, 250h (s)

\rightarrow • Whenever something is used, put it in cache

(2)

ABC

MNO

XYZ

GHI

LRU

least recently used

I Store MNO
⇒ remove something in cache

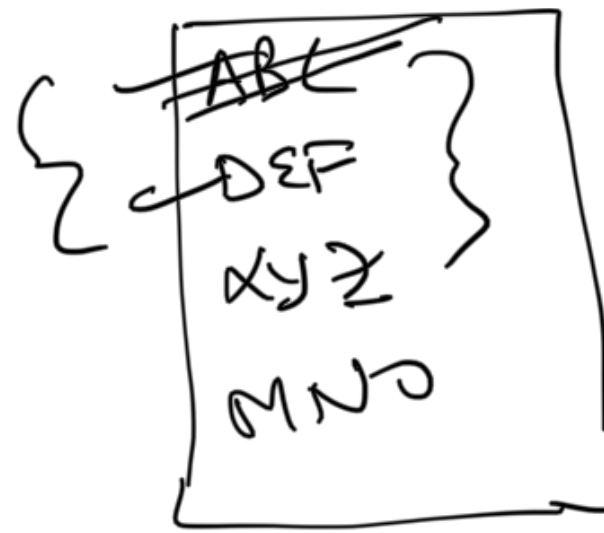
II don't store MNO

Request

LRU

LFU

→ ABC ✓
 → DEF ✓
 → ABC ✓
 → XYZ ✓
 → DEF ✓
 → MNO ✓



RAM



MNO



MNO

LFU

Least Frequently Used

Every time, DEF,

cat = cat + 1 ✓

active old

$$cat = \frac{(0.2 \times cat)}{2} + \underline{0.8(1)}$$

date

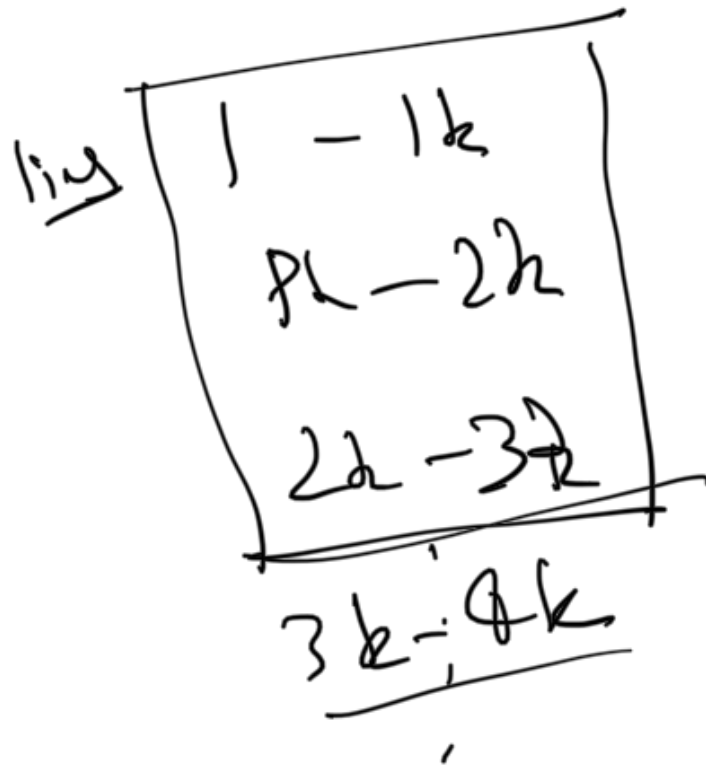
RAM

↳ Paging / Swapping

10 GB

8 GB RAM

→



10k

3k lines
limit

RAM



7k-10k

32 rows at
a time

2-5k

~~2k-3k~~

7k-8k

8-9k

9-10k