Basics of Storage

→ Overview

→ Storage Basics / Hierarchy

→ RAM / ROM / HDD, SSD

→ Caching Concepts

- Base
- Questions
- Imp Results
- Extras / doubts

used apps

C++, Java, Go, webd > Application level

C

Application level – HLD – LLD

Compilation, linking, inde ..

Verilog

Assembly ←

↓
os

0/1 language

Database

↑

Hardware

Electronics

Flip Flops

Register

Internal to ↑ com

L1, L2, L3 Cache

Processors

(1000s bits) Registers (temp storage, super fast, expensive)

(10KBs - few MBs) ~~Internal~~ Cache (really fast, higher Storage capacity)

⇓

lower cost

L1, L2, L3 [SRAM]

↑ LOC

, [DRAM]

(8GB - 32GB) RAM (Storage cap ↑, cost/storage ↓, fast)

(Memory), (main memory)

USERS

(512GB → HDD/SSD (Storage) ↑SC, cost ↓
- 1TB)

(Harddisk)

Speed ↓

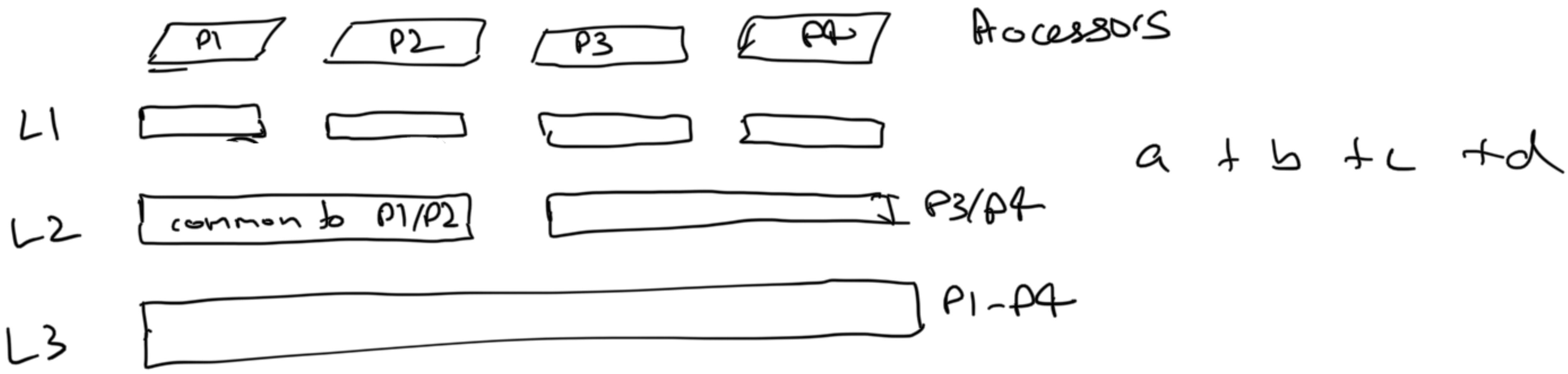
Higher SC
Cheaper/GB
Slower

RAM vs HDD/SSD

Volatily
Persistence

RAM → stop supplying power
TEMPORARY
→ data wiped
Volatile
non persistent

PERMANENT HDD/SSD → non volitle
or persistent



ROM

Read Only Memory

write only once, & then read forever

BIOS

↳ Basic I/O System

PROM → Programmable ROM

EPROM → Erasable PROM → UV light

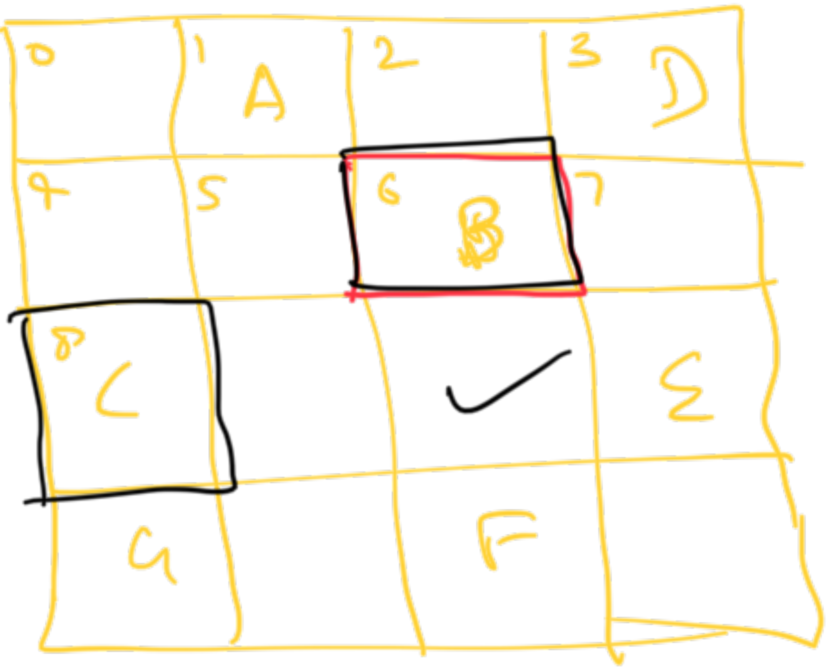
+ losses

EEPROM → Electrically EPROM

↳ erased electrically

RAM

Random Access Memory



A
B
C
D

mem mgmt Unit

MMU

logical – Phy Address

log → 0x0001

↓

0xFFAB105C

→ Give me the data at #6 → B

in O(1)

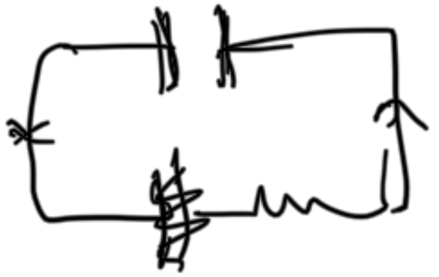
clockcycles

Volatile

RAM — SRAM
static RAM → Flip Flops → x
in recharging

RAM — dRAM
Volatile
dynamic RAM → Capacitors

C → 0V ⇝ 0
↳ 5V ~ 1



if → 0 → 0V
↳ 1 → 5V ⇝ 0V
→ 0

? DRAM slower than SRAM

DRAM cheaper SRAM

DRAM larger SRAM

SSD >> HDD (speed) (cost)

HDD / SSD

→ Electromagnet

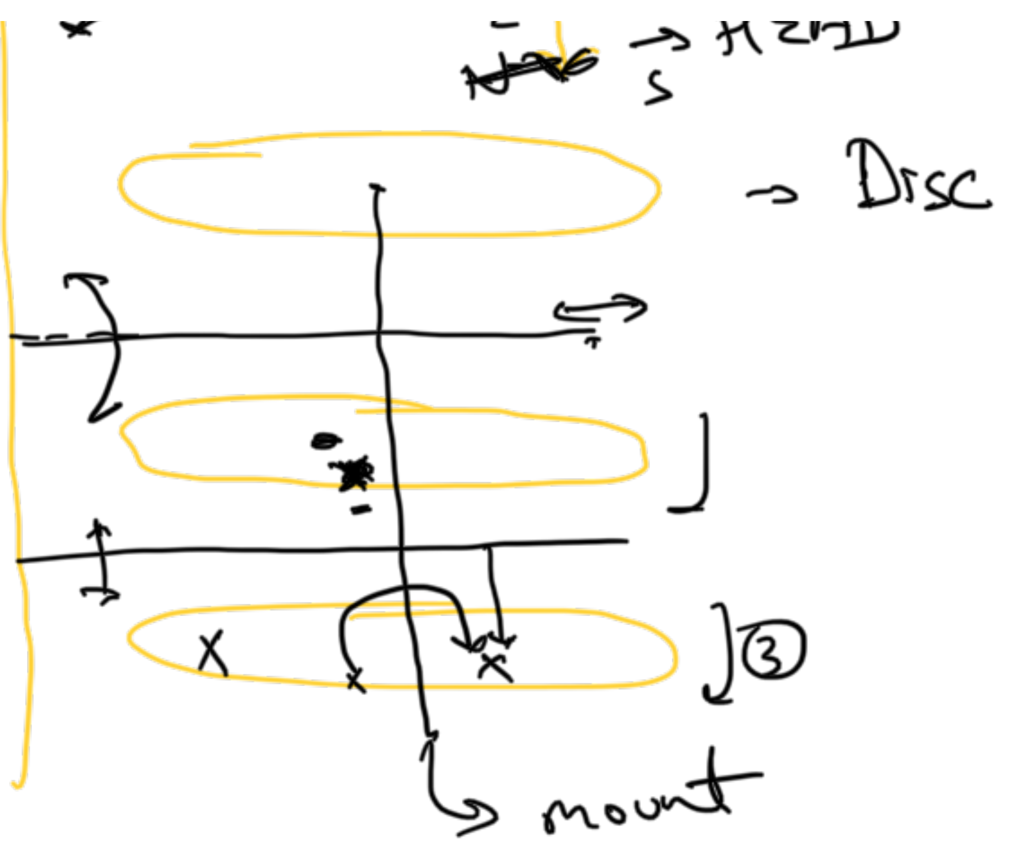
SSD

→ persistent ✓

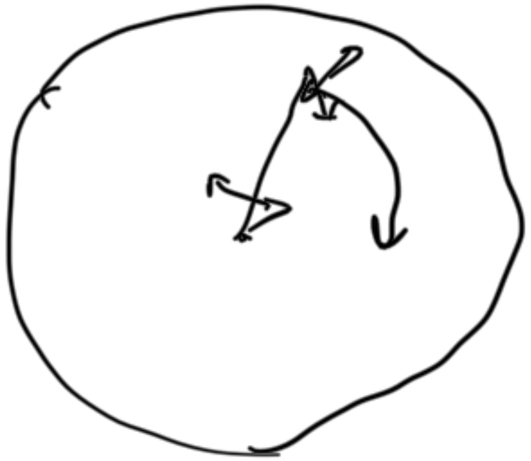
→ slow ✓

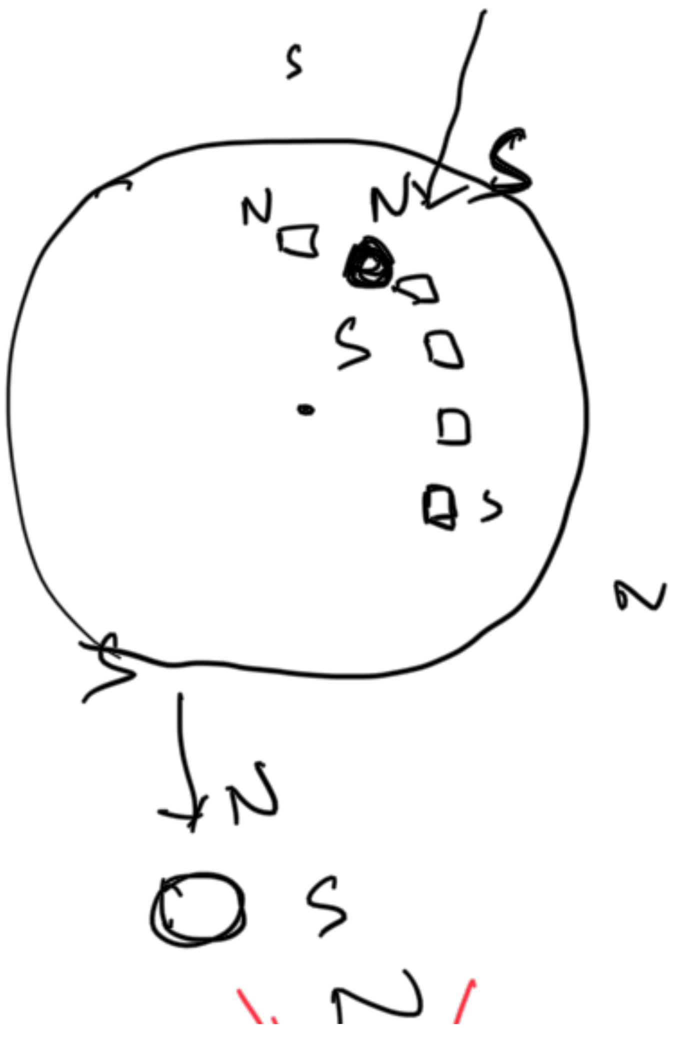
→ cheap ✓

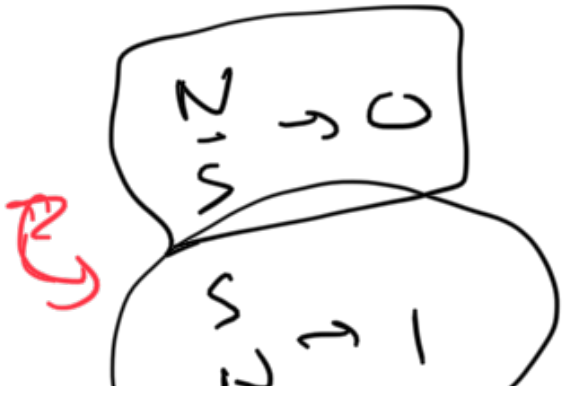
→ large SC ✓

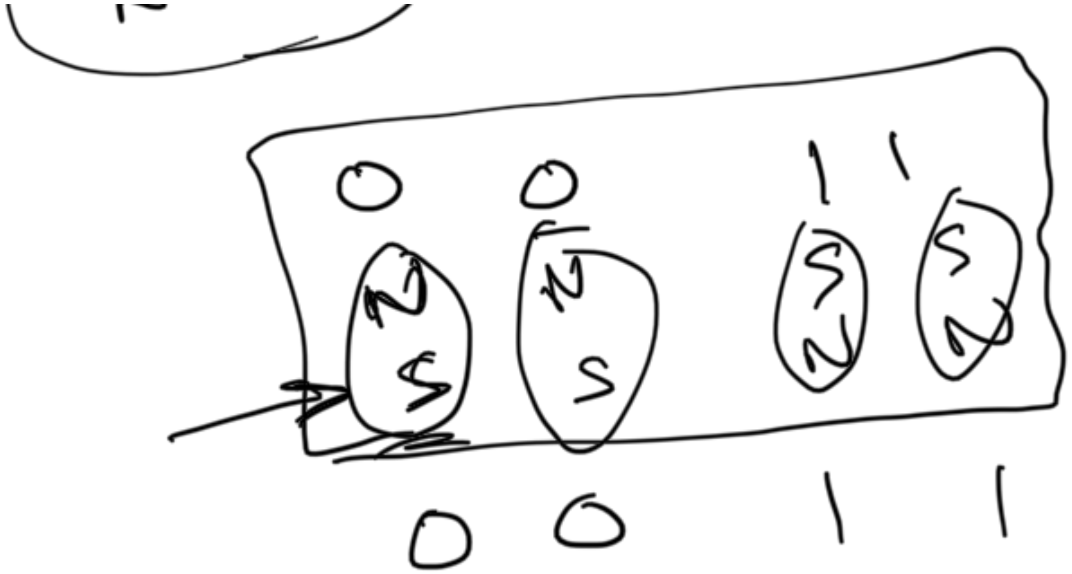


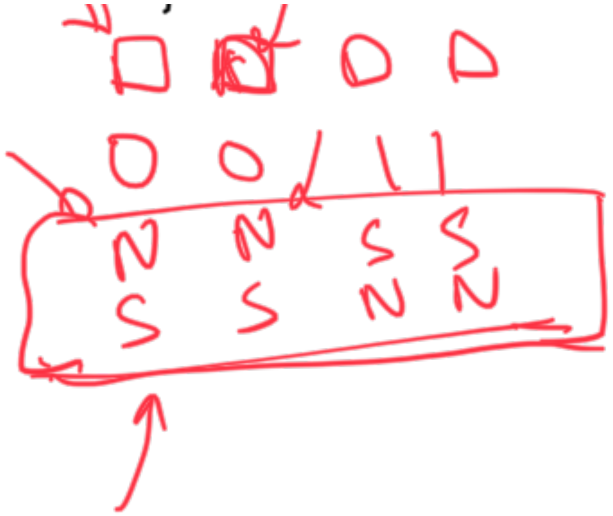
(Yellow) →











Flash

↑

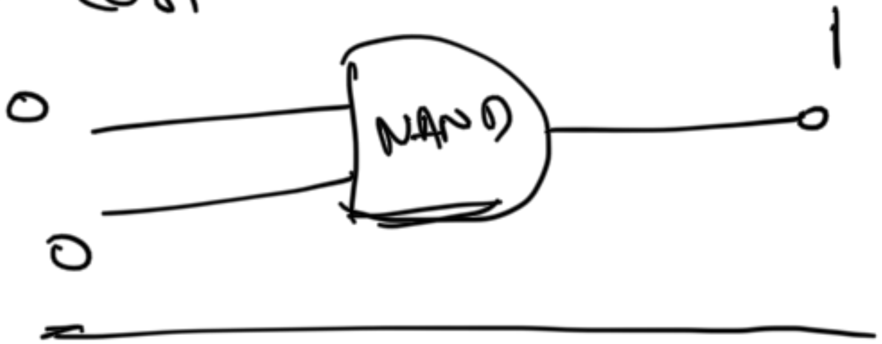
SSD faster access, but higher cost

(white) Solid State Drive

↳ NAND gates

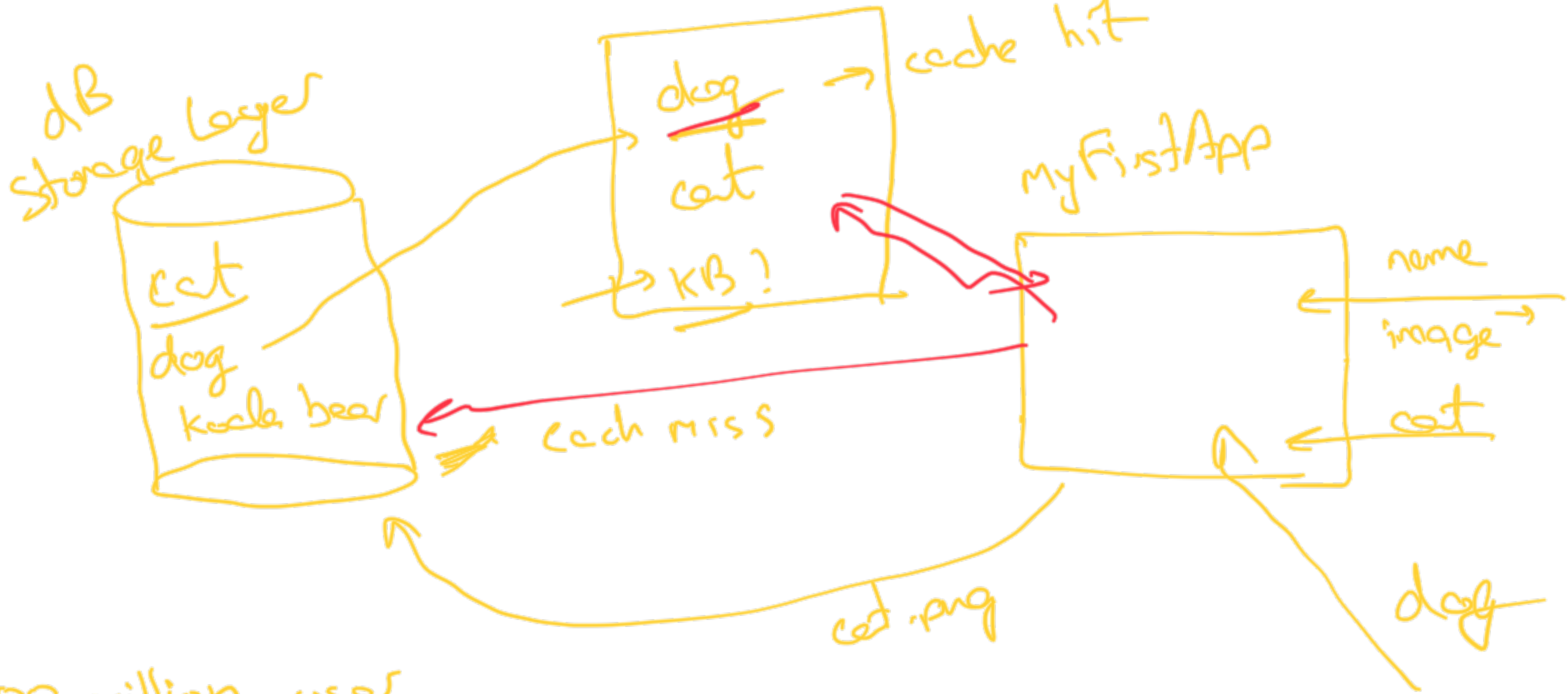
↳ Not And

↳ Floating gate transistor



Caching Concepts

RAM → 2 images



-> 100 million user

latency = 10ms   ws -> 'cat are cute'   ws -> 'KBs are best'

JB -> " Hey dogs are awesome "   60 mill -> 10ms

60ks

-> 80 million

↳ 60 mill

60 mill -> 80 ns

Caching

Keep more frequently reqd data in an easier to access place

✓ ① Choosing what to store in my RAM / Cache ✓ Everything

② When cache is full, how do we add new stuff?

16GB, 2GB is used

Ans 1

↳ Cache not full → ↳ dog ↓

3GB → store

2GB → no store

~~↳ Cache is full~~

Ans 2

Dog	Cat	KB

new → Dolphin

↳ which 3 of total 4 should I keep

↳ which 1 should I not keep ?

?

LRU

Least Recently Used

RAM → 3 images

T0 ✓ Dog

T1 → Cat

T2 → Dog

T3 → KB

T4 → KB

T5 → Cat

→ Dolphin

~~T0~~ T2 — ~~Dog~~ Dolphin

T5 — ✓ Cat

T4 ~~T3~~ — ✓ KB

Dog

LFU

Least Frequently Used

w1 D: 1

D → 2

D ✓

K ✓

w5 K ✓

C ✓

w10 Dolphin ✓

1 → 2

↓

0.5 +1 → 1.5

↓

0.75 +1

→ 1.75

0.5 RAM

3	D
2	K
1	C
	Dolphin

LFU DA

$$\text{newVar} = \alpha(\text{old Var}) + (1-\alpha)\,\text{new Inp}$$

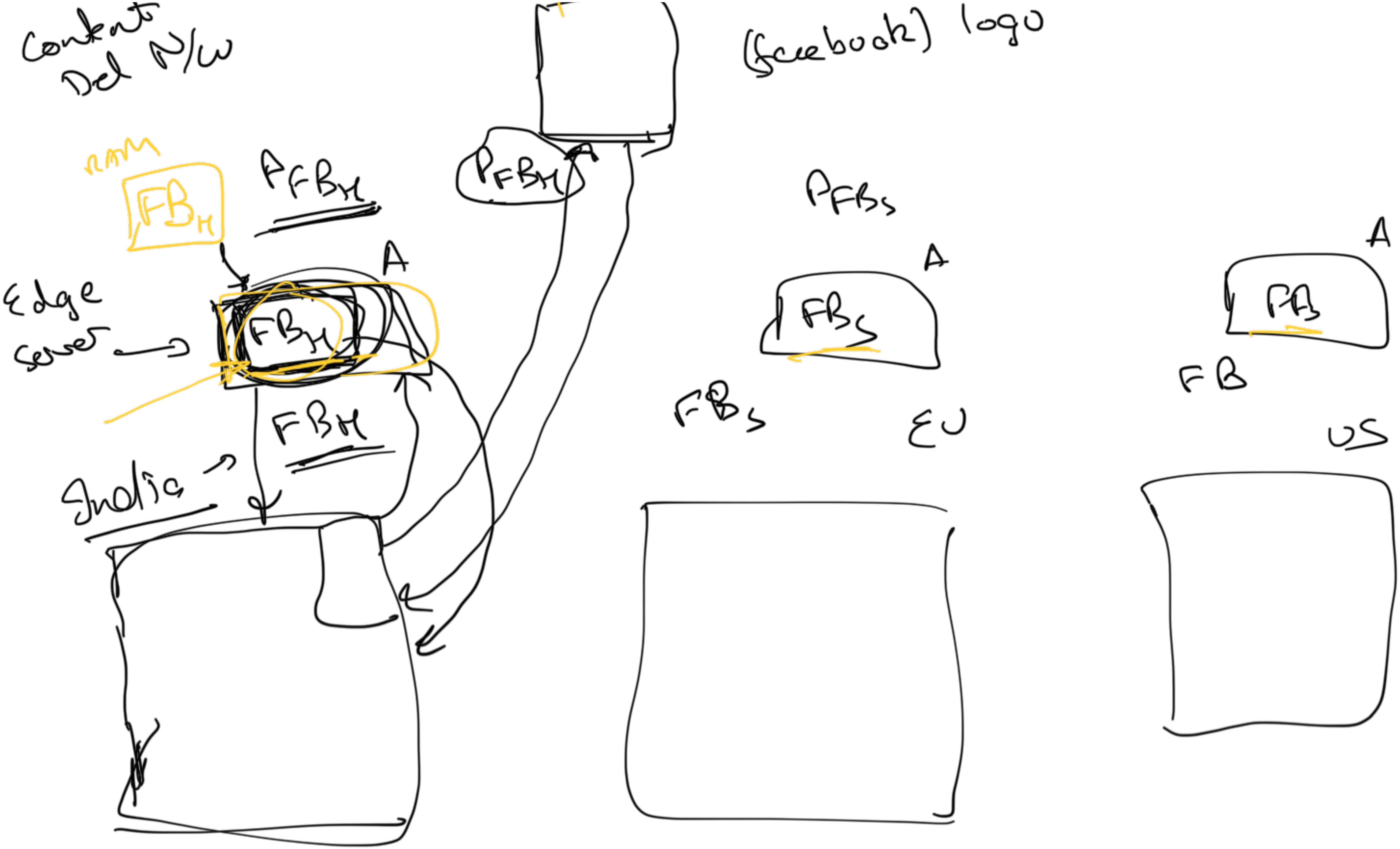
? LFU with Dynamic Adaption?

CDN Akamai

Server in CA

Content Del N/W

(facebook) logo


RAM
FBM
PFBM
A
Edge server
FBM
India
PFBM
PFBs
A
FBs
FBs
EU
A
FB
FB
US