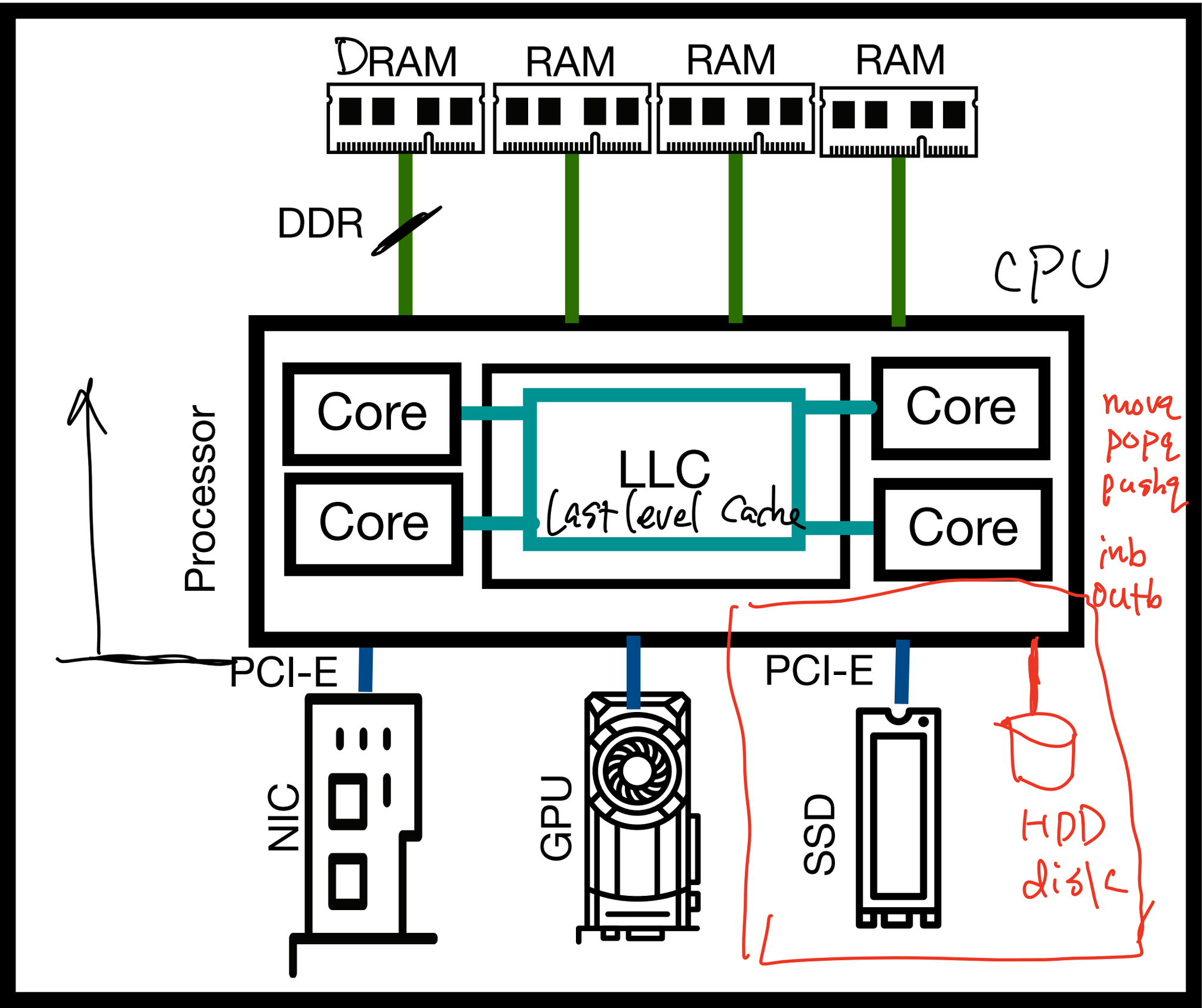
Machine


DRAM
RAM
RAM
RAM
DDR
CPU
Processor
Core
Core
LLC
Last Level Cache
Core
Core
mov q
pop q
push q
inb
outb
PCI-E
PCI-E
NIC
GPU
SSD
HDD
disk

Hard disk drive organization

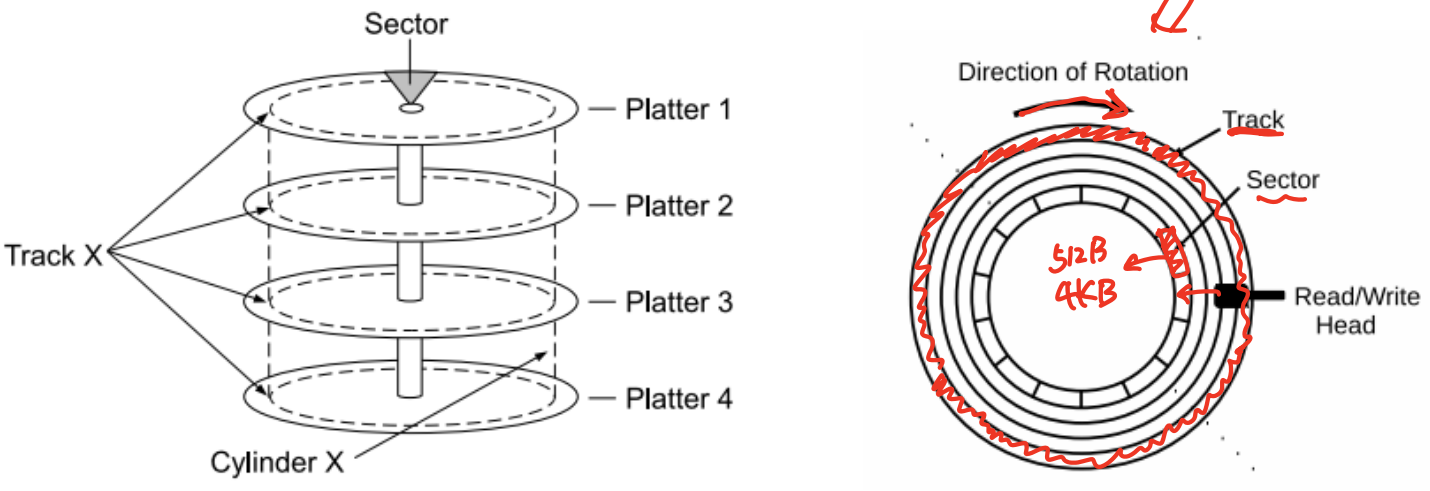
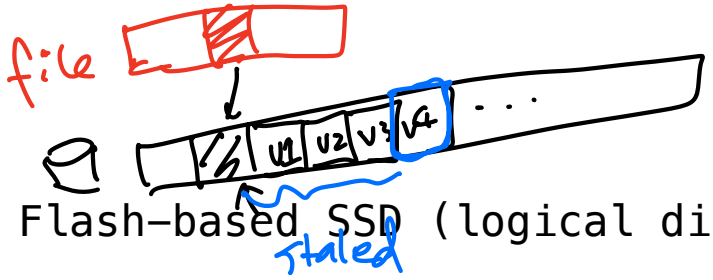


Fig 5.11, 5.12 from
http://www.ccs.neu.edu/~pjd/x600-book-v0903.pdf



A Flash-based SSD (logical diagram)

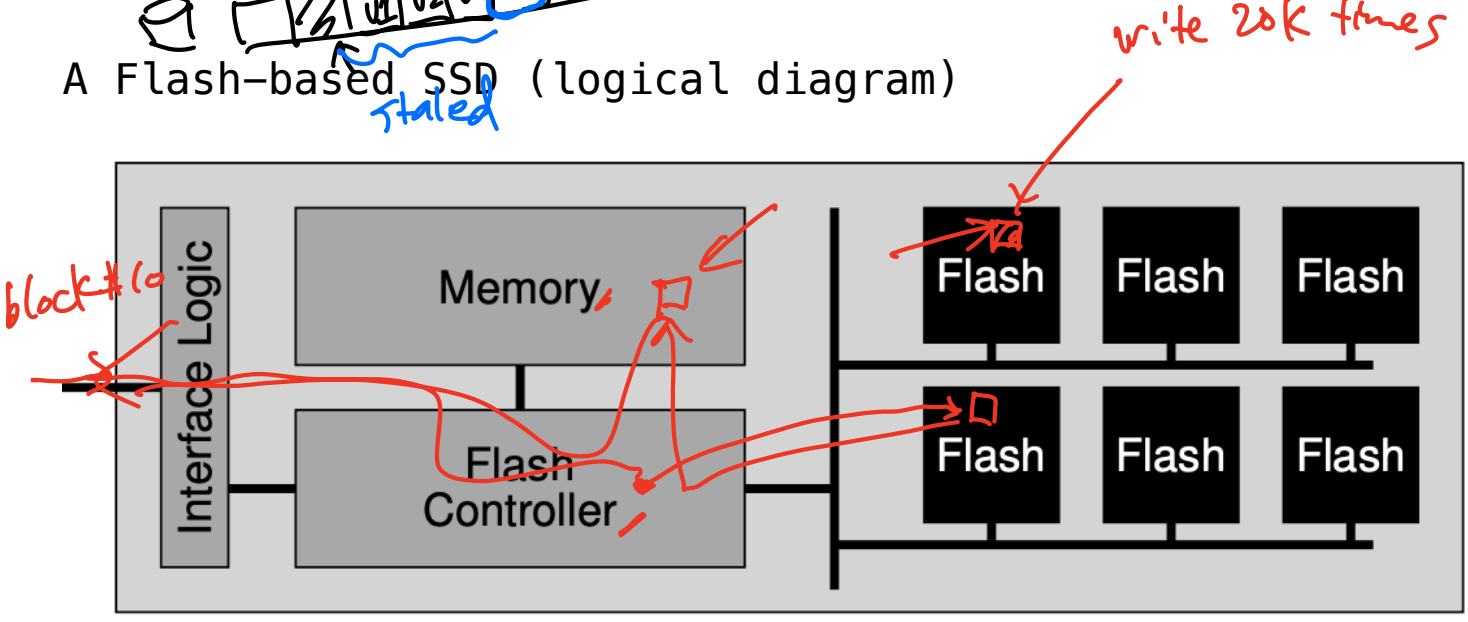


Fig 44.3 from
https://pages.cs.wisc.edu/~remzi/OSTEP/file-ssd.pdf

wear-out

Q: What to do to prevent/mitigate the wear-out?

Week 10.a
CS3650
03/11 2024
https://naizhengtan.github.io/24spring/

1. I/O architecture and storage devices
2. Intro to file systems
3. Files

- Fancy storage devices
  - Low latency? persistent memory
  - long-term storage? project silica

- Intro to FS

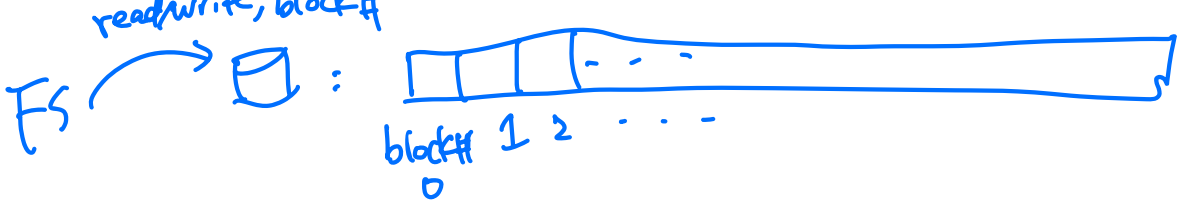
  Q: What does FS do?
  - read pdf from fs
  - deleting files
  - Creating files

  ① provide persistency

  ② name a seq of bytes (files)

  ③ give a way mapping from user-friendly names to named bytes (dirs. fs namespase)

- Q: Where are FSes impelemented?

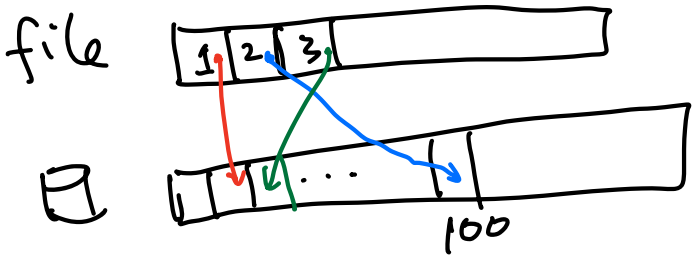


- Files Q: what is a file?
  - user: a seq of bytes.
  - FS: a collection of blocks.

file:

<file, offset> → block address

per-file: [SOMETHING (inode)] → file mapping



block size: 4KB
file size: 12KB

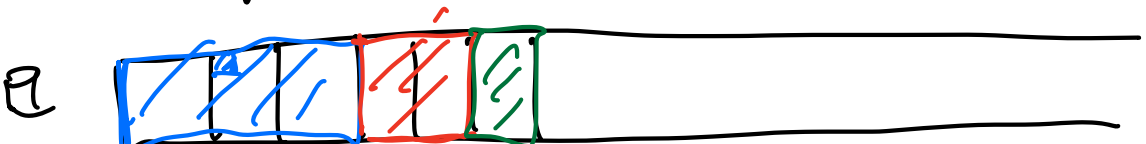
offset: 4096+1 ⇒ block # 100

- FS parameters
  - small files vs. large files
  - disk utilization (metadata, fragmentation)
  - access patterns
    - sequential access vs. random access

Q: How will you design this SOMETHING?

Candidates:

- continuous allocation (extent-based)

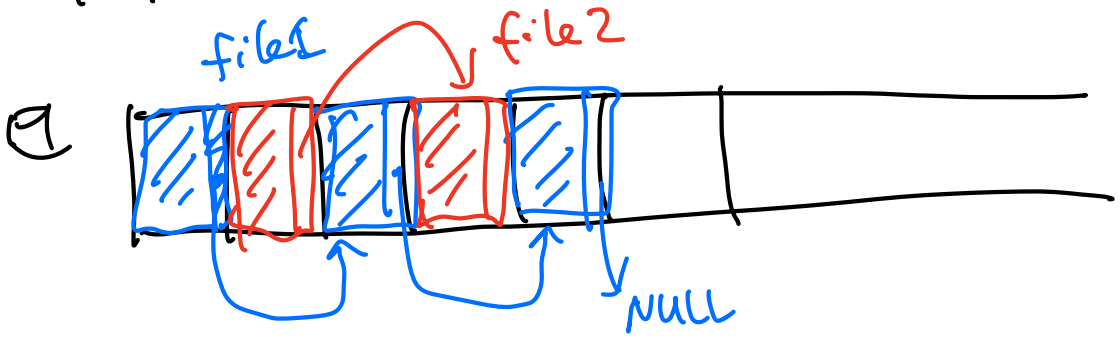


file1 file2 file3

(base, size)

$\left\lfloor \frac{\text{offset}}{\text{block size}} \right\rfloor + \text{base}$

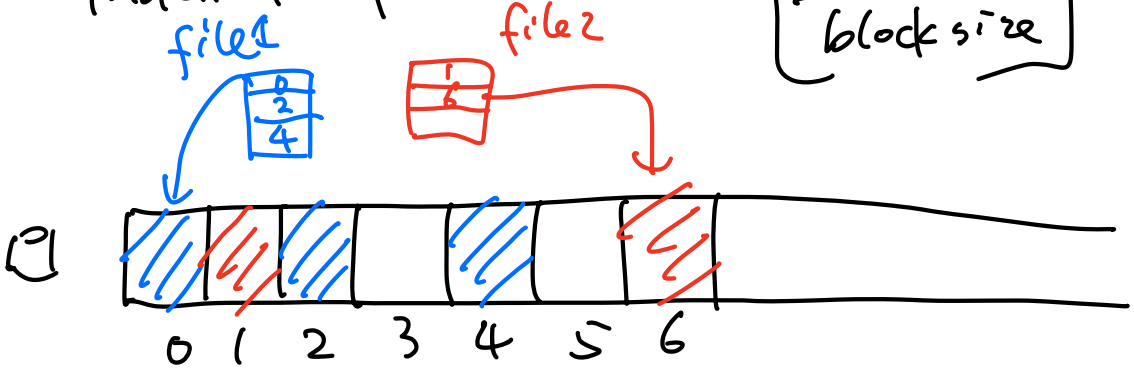
- linked-base files.



- indexed files

- $\left\lfloor \frac{\text{offset}}{\text{block size}} \right\rfloor$

$\frac{4GB}{4KB} = 2^{20}$



- proposal

file