IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

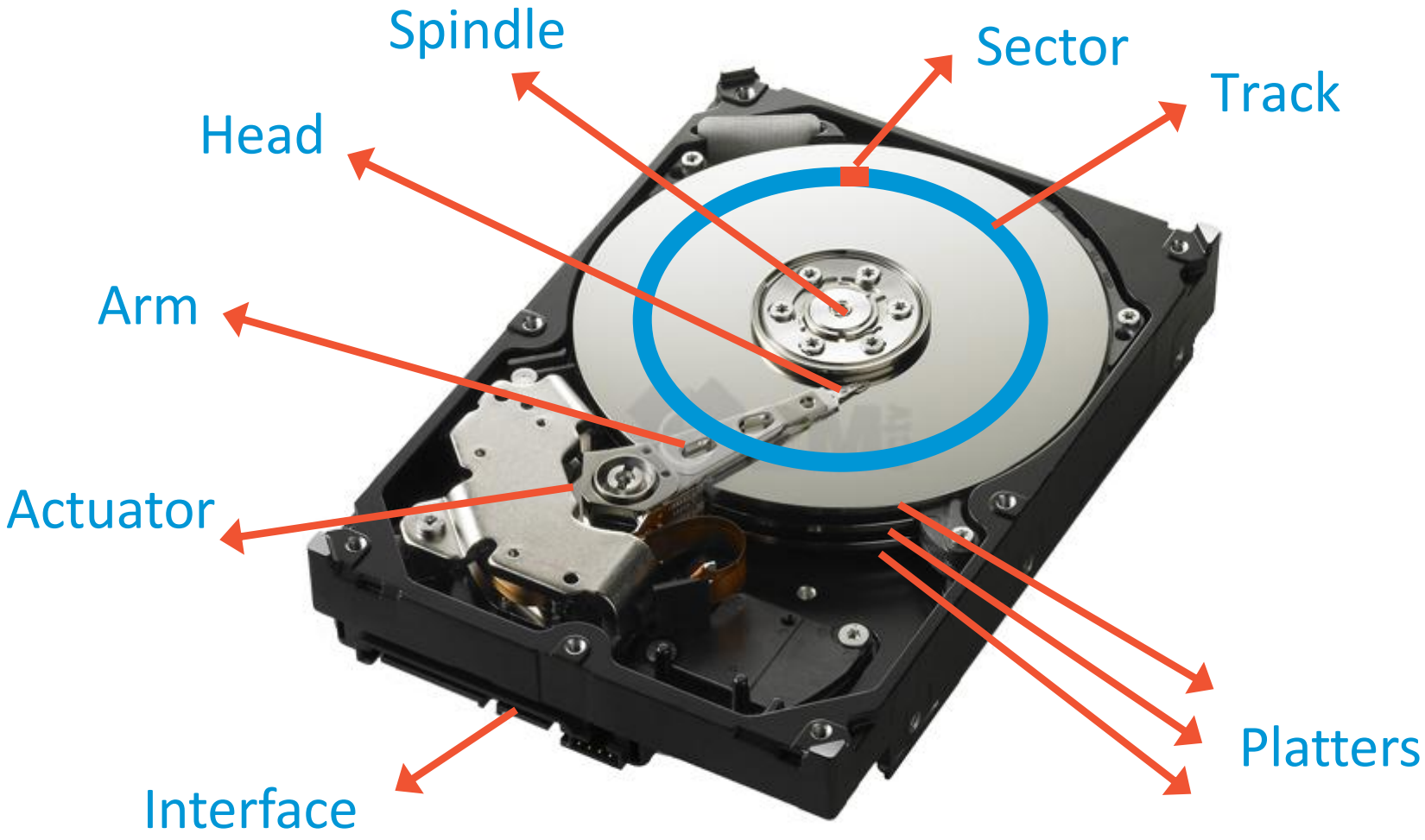
Lecture 28:
Flash-based Solid State Drives

Agenda

- Recap
- Flash-based Solid State Drives (SSDs)
  - Internals
  - Flash Memory
  - Flash Translation Layer (FTL)

Recap

- HDD Internals



Recap

- HDD I/O Time & I/O Rate
  - I/O time ($T_{I/O}$) includes three parts
    - **Seek**
    - Waiting for the **rotational delay**
    - **Transfer**

$$T_{I/O} = T_{seek} + T_{rotation} + T_{transfer}$$

  - I/O rate ($R_{I/O}$):

$$R_{I/O} = \frac{Size_{Transfer}}{T_{I/O}}$$

  - Favor sequential workloads

Recap

- On-drive Cache
  - “Track Buffer”
  - **Writeback** (Immediate reporting)
    - Acknowledge a write has completed when it has **placed the data in on-drive memory**
  - **Write through**
    - Acknowledge a write has completed after the write has **actually been written to surface**.
- I/O Scheduling
  - OS uses heuristics to re-order/merge I/O requests and (hopefully) minimize I/O time

Agenda

- ~~Recap~~
- Flash-based Solid State Drives (SSDs)
  - Internals
  - Flash Memory
  - Flash Translation Layer (FTL)

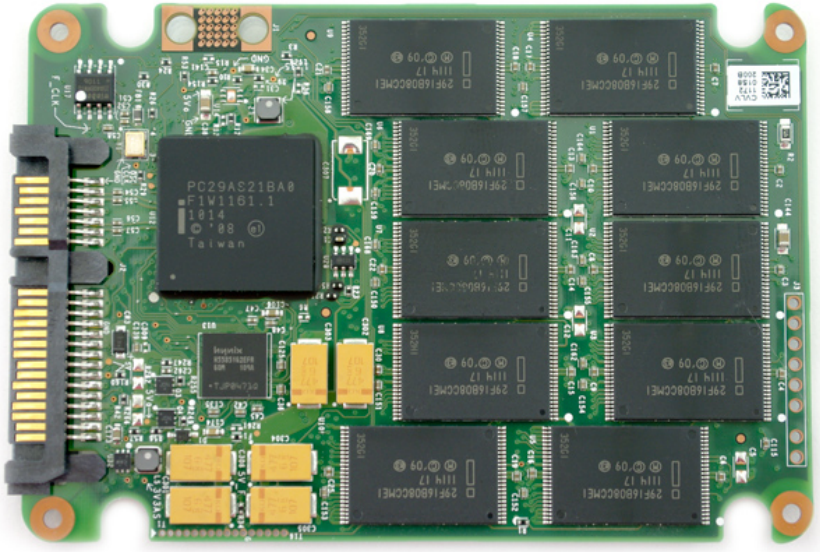
Flash-based SSD

- a “truly revolutionary and disruptive” technology

Western Digital

Seagate

TOSHIBA



intel

SAMSUNG

FUSION-iO

Micron

crucial

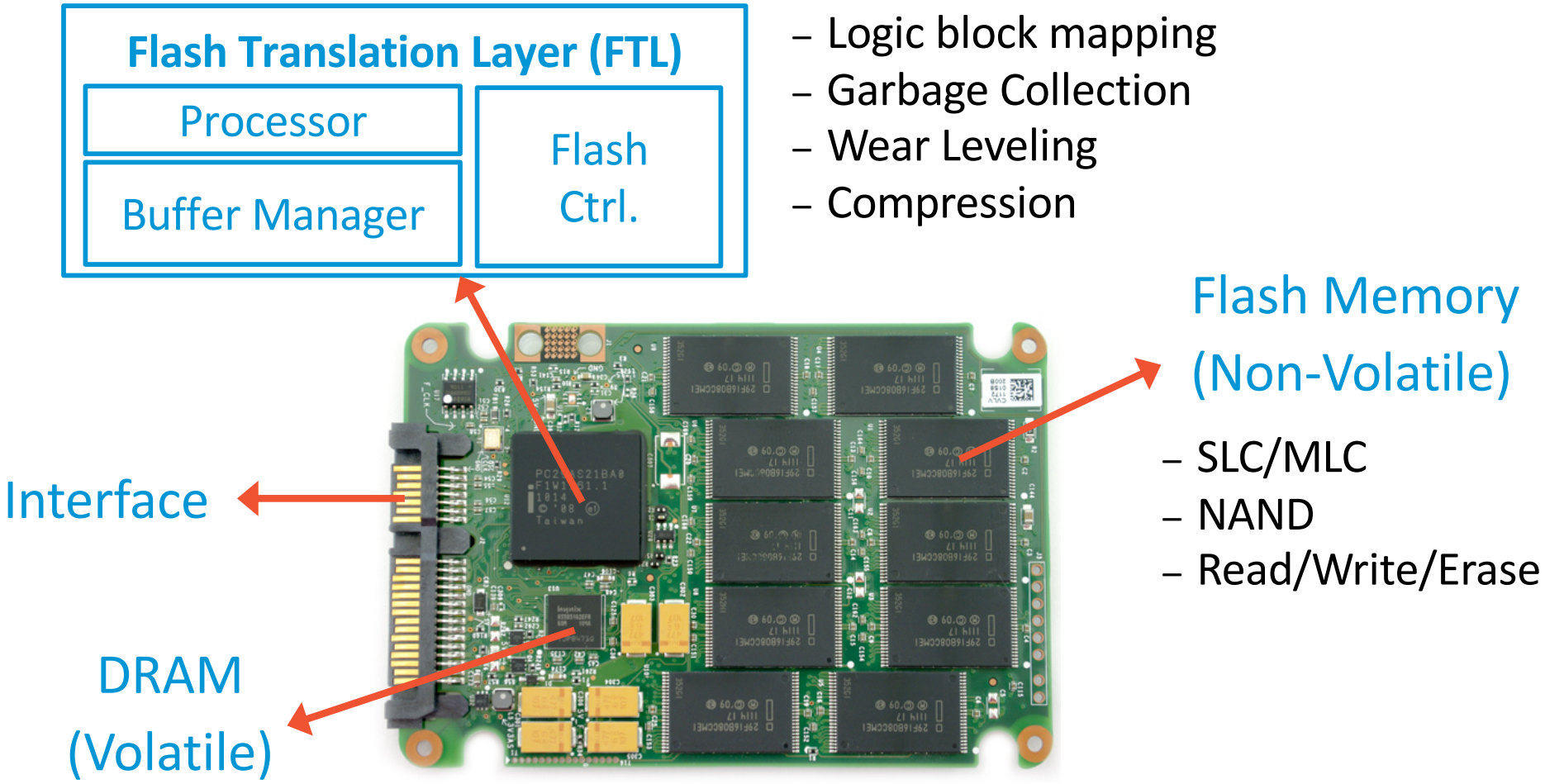
SanDisk

Verbatim

OCZ Technology

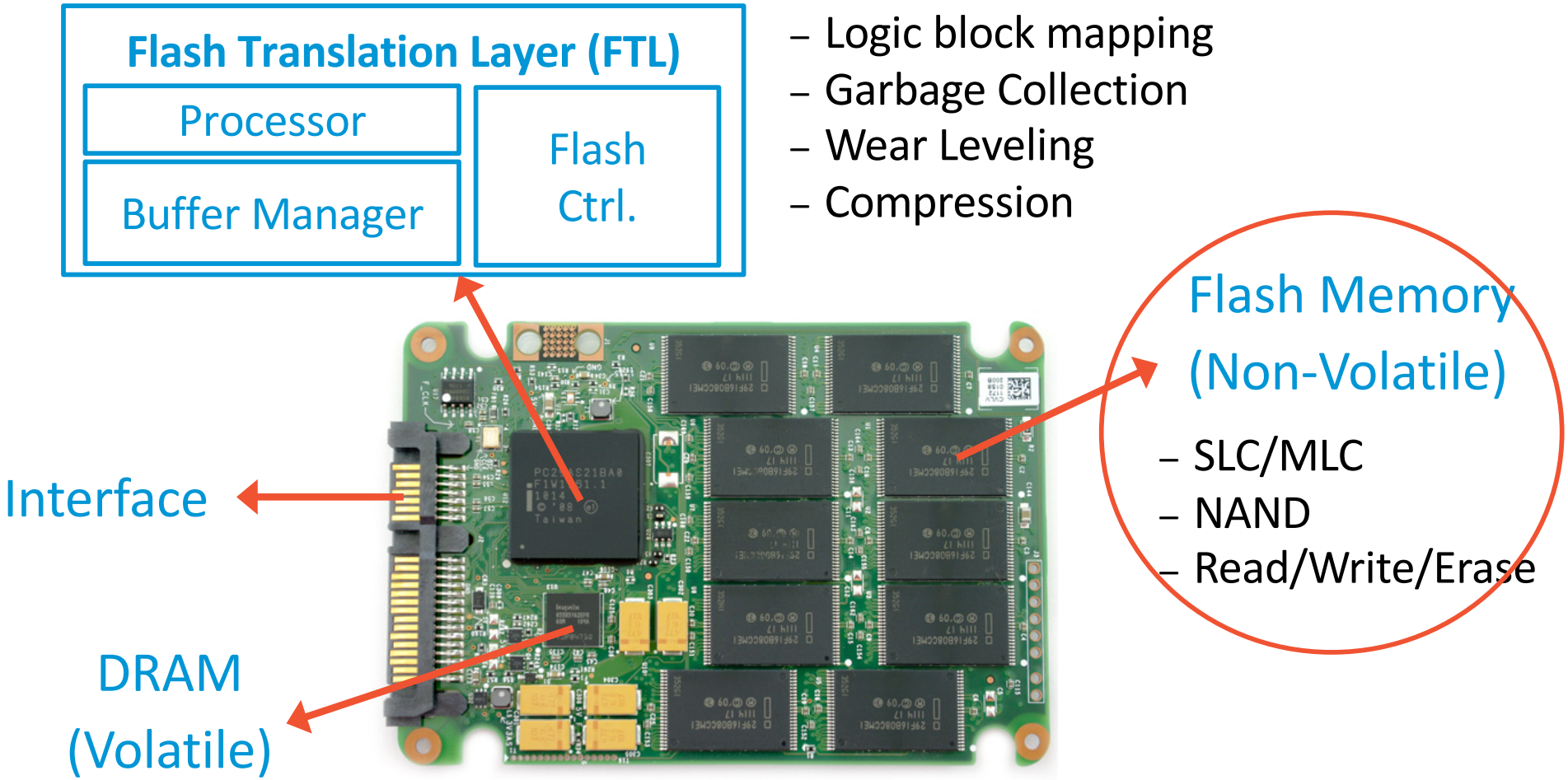
Flash-based SSD

- Internals



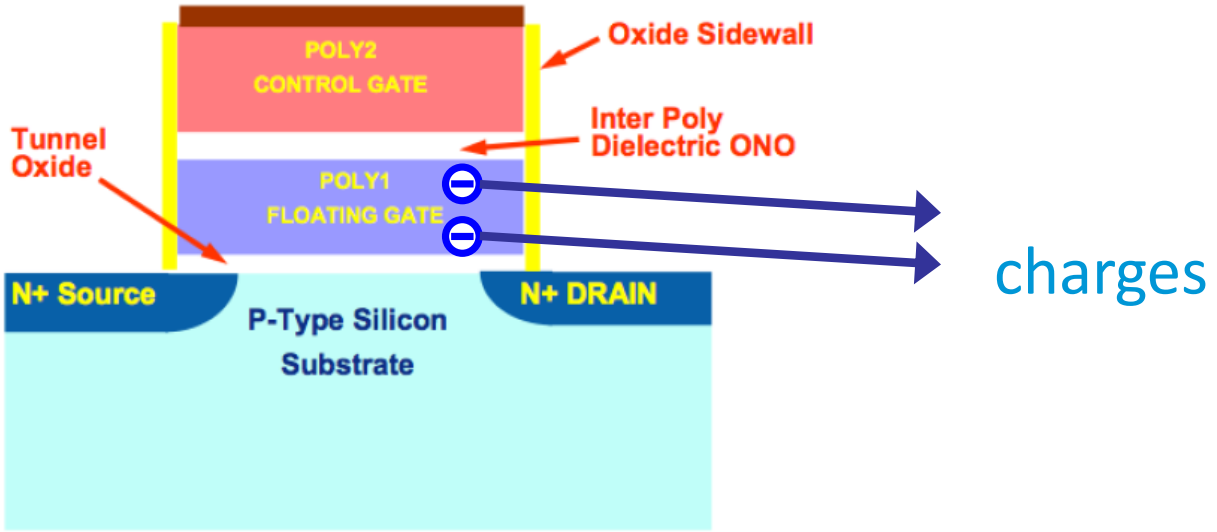
Flash-based SSD

- Internals



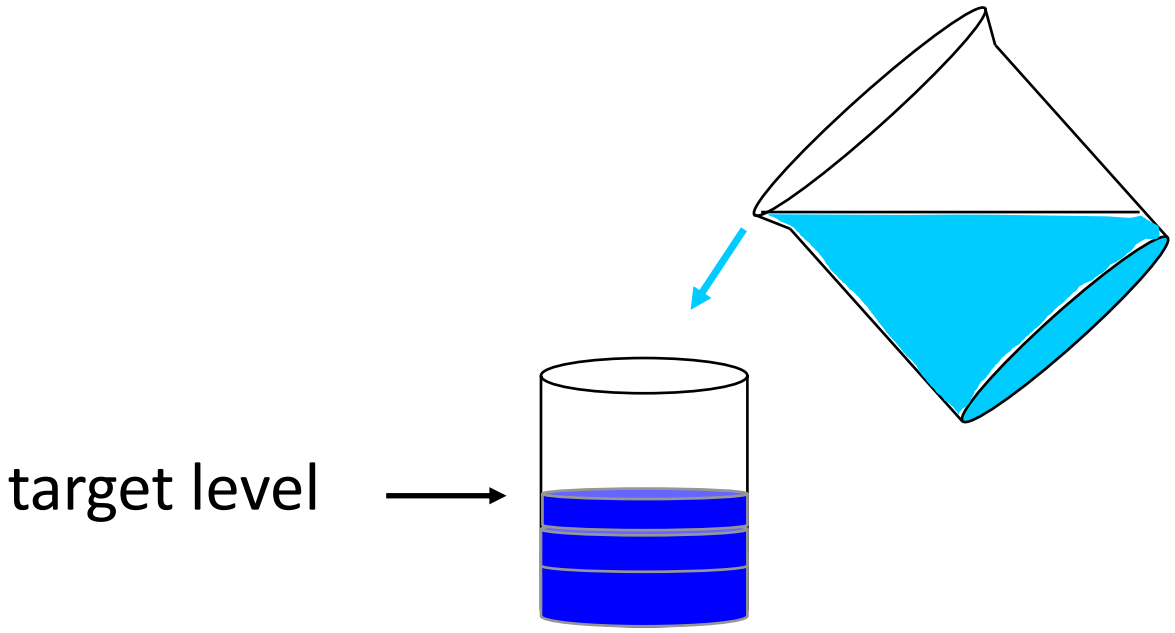
Flash Memory

- Flash Memory Cell
  - floating-gate transistor
  - electrical charges can be injected (ejected) into (out of) the floating gate
  - charge level represents data (0/1)



Flash Memory

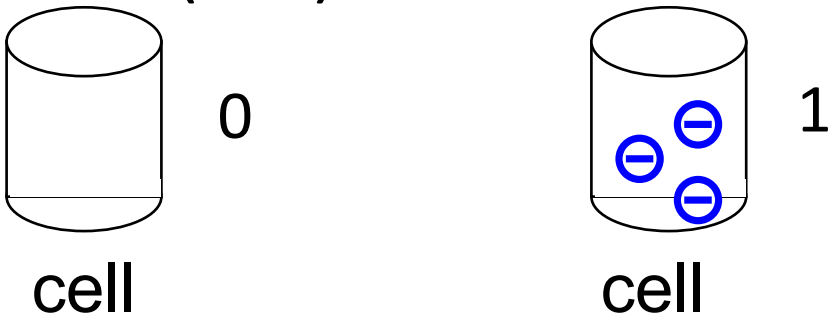
- How is a cell programmed?
  - through multiple rounds of charge injection



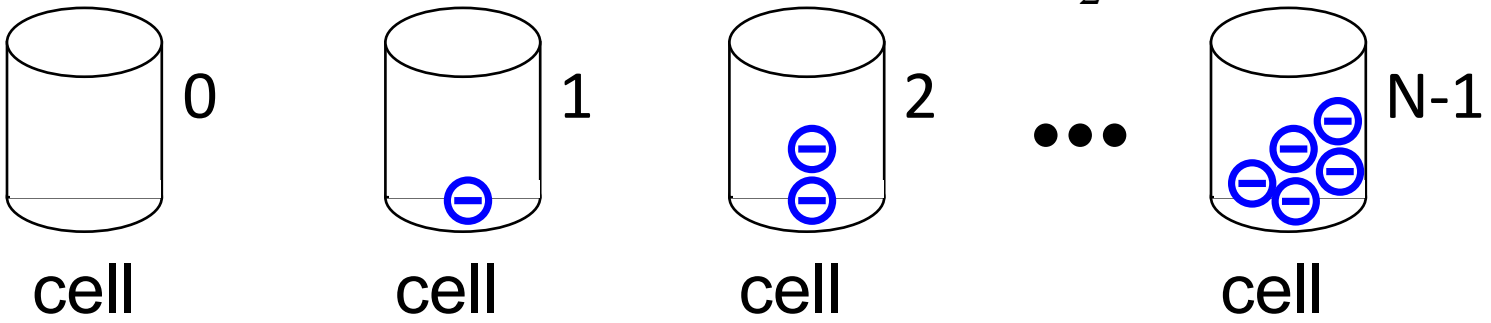
A flash cell

Flash Memory

- Single-level cell (SLC): two levels -> one bit



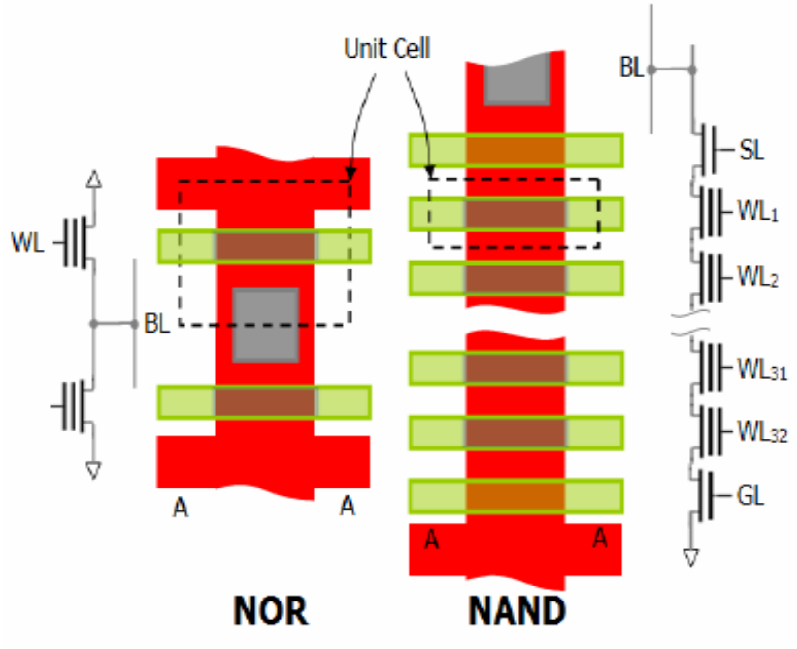
- Multi-level cell (MLC): N levels -> $\log_2 N$ bits



- MLC is used in most consumer-grade SSDs

Flash Memory

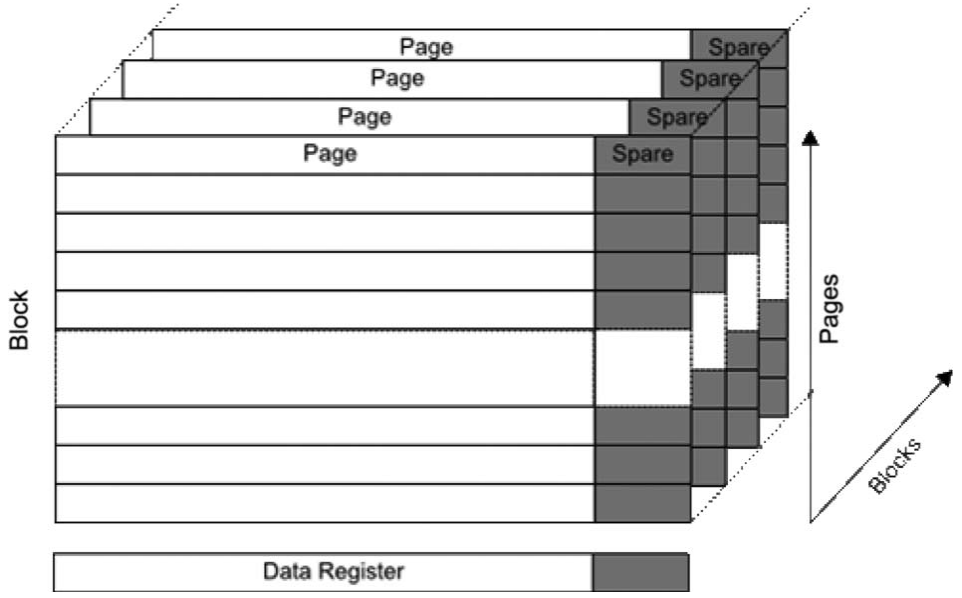
- **NOR** flash vs **NAND** flash
  - different ways to connect cells



- **NAND is used in most SSDs**

Flash Memory

- **Blocks** and **Pages** of NAND Flash
  - flash cell array is arranged as independent blocks
  - each block consist of a set of pages
    - e.g., a block contains 128 pages, each page is 4KB

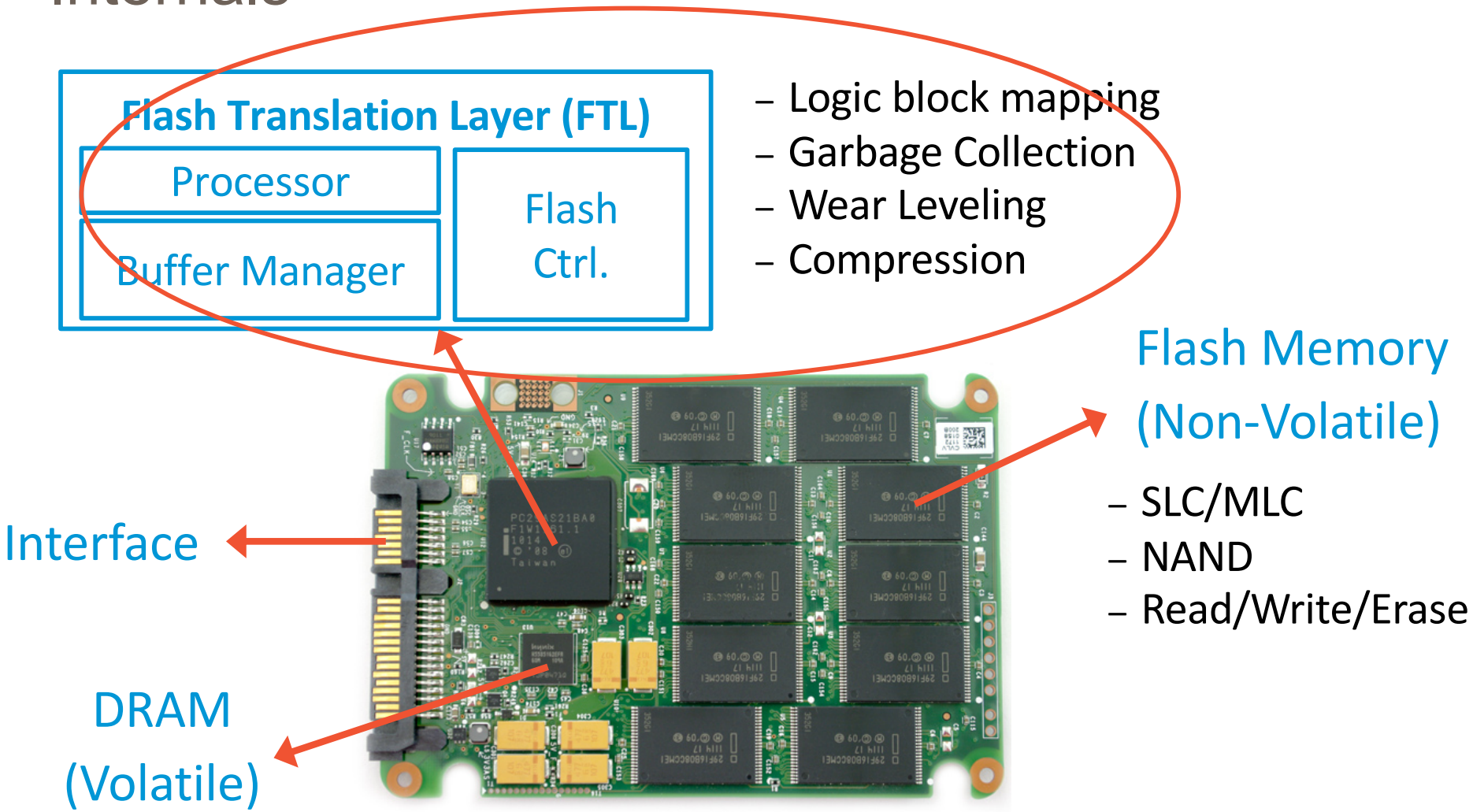


Flash Memory

- Basic Operations of NAND
  - **Erase**: set all bits of a block to ‘1’
    - a block is the smallest erasable unit
  - **Program** (Write): clear some bit(s) of a page to ‘0’
    - a page is the smallest programmable unit;
    - must erase a block before updating any pages inside
  - **Read**: get bit(s) out of a page
- each cell can only stand a limited number of program/erasure cycles (**P/E cycles**)
  - need to even out the usage of cells

Flash-based SSD

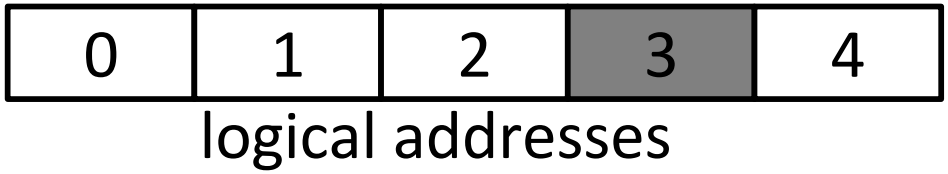
- Internals



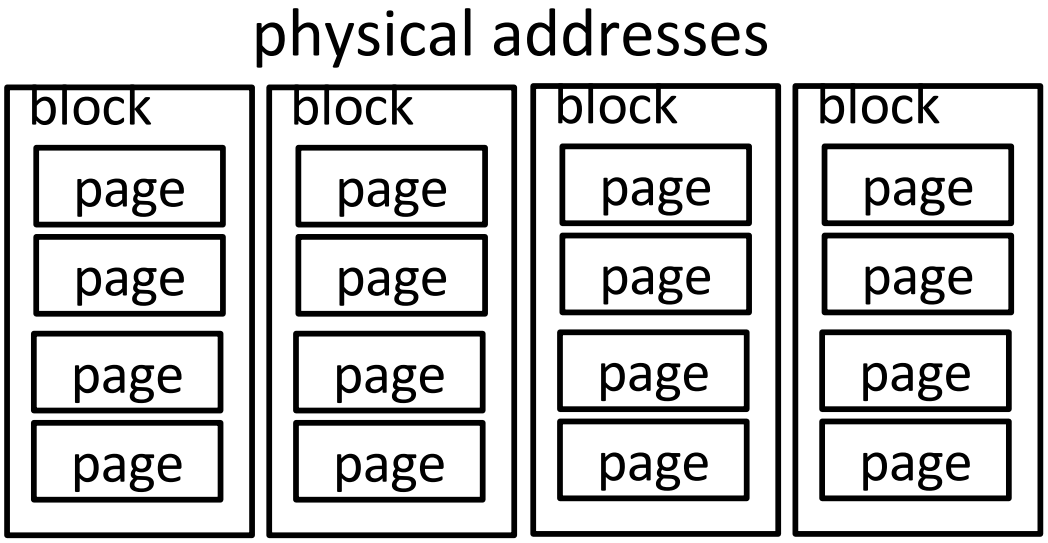
Flash Translation Layer (FTL)

- Logical Block Mapping
  - maps logical addresses to physical addresses
  - maintains a mapping table

Host's View



Flash Controller's View

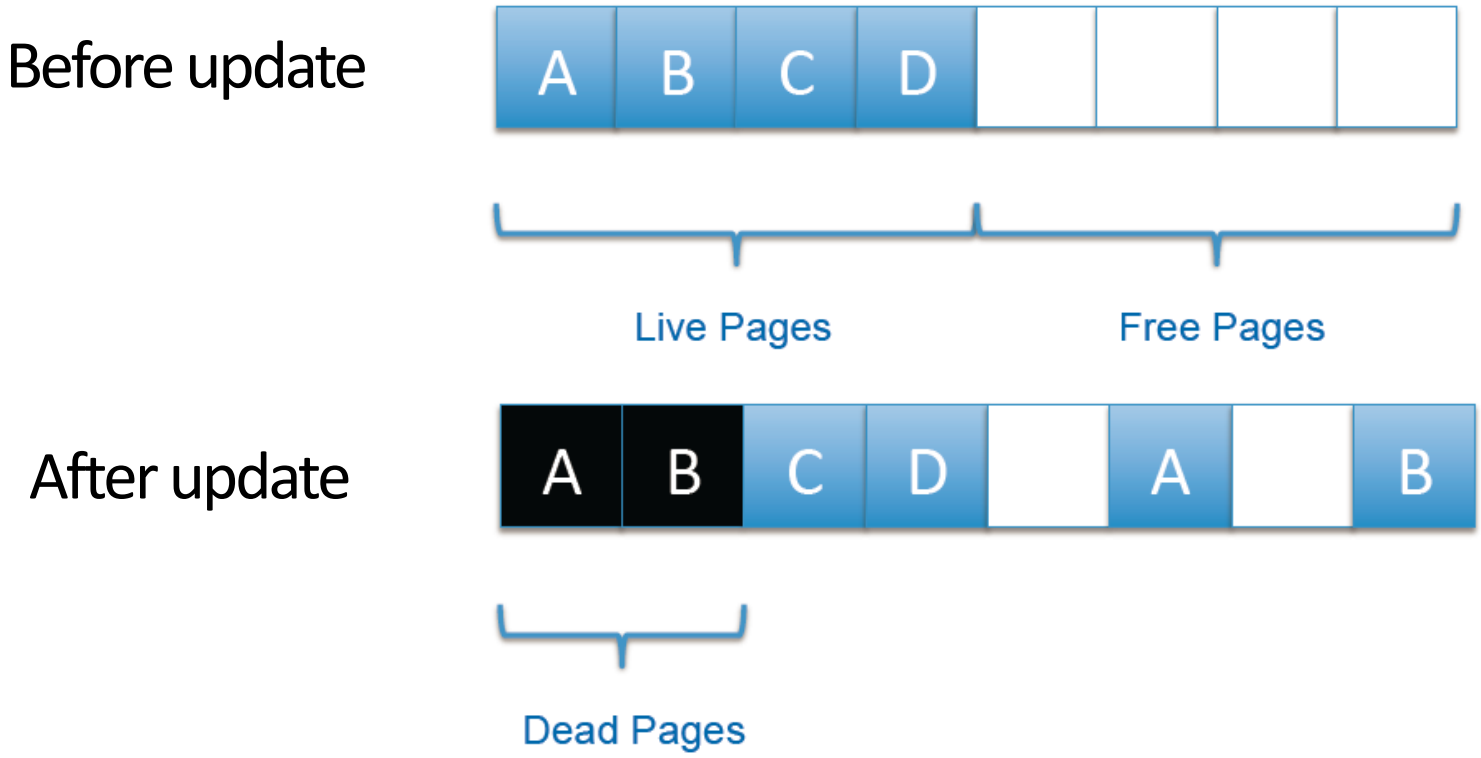


Flash Translation Layer (FTL)

- How to map?
  - borrow idea from a classic paper (LFS)
    - “The Design and Implementation of a Log-Structured File System”, SOSP’92
  - alleviate the “erase-before-write” constraint
  - out-of-space update:
    - maintain free (erased) pages, always append updates, and invalidate old copies
    - the same logical address is re-mapped to a different physical address

Flash Translation Layer (FTL)

- Example: update pages A and B



Flash Translation Layer (FTL)

- **Garbage Collection**
  - re-use pages that contains invalid (obsolete) data
  - happen as background work or on demand
  - source of unpredictable latency in I/O operations

- **Wear leveling**
  - each cell can only stand a limited number of program/erasure cycles
  - let the pages be erased/programmed about the same number of times

Agenda

- ~~Recap~~
- ~~Flash-based Solid State Drives (SSDs)~~
  - ~~Internals~~
  - ~~Flash Memory~~
  - ~~Flash Translation Layer (FTL)~~

Questions?

*acknowledgement: slides include content from “Modern Operating Systems” by A. Tanenbaum, “Operating Systems Concepts” by A. Silberschatz etc., “Operating Systems: Three Easy Pieces” by R. Arpaci-Dusseau etc., and anonymous pictures from internet.