Solid-State Drive

Juncheng Yang

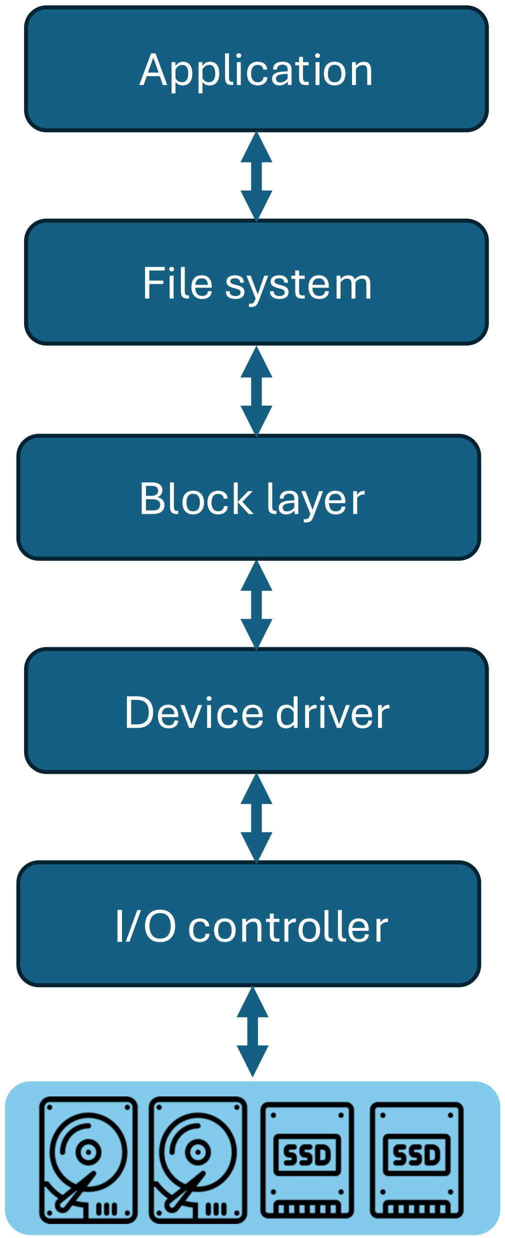
Harvard John A. Paulson School of Engineering and Applied Sciences

HARVARD UNIVERSITY MADSYS
Measurement and Design of Systems Lab

Recap: Hard Disk Drive

Agenda

- SSD internals
- SSD controller
- SSD performance
- HDD reliability
- NVMe interface
- Future trend



Three Key Questions

- What is Flash Translation Layer and what does it do?
- What is SSD's performance characteristics and Why?
- How do we extend SSD lifetime?

* Although this class we will discuss a bit hardware, they are not the focus

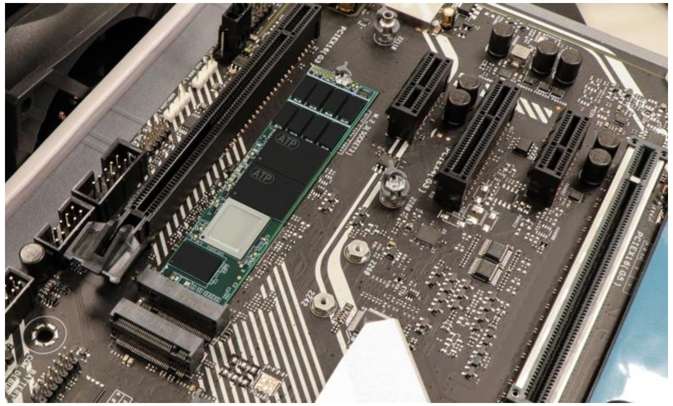
SSD basics

- From HDD to SSD: why?

	SSD	HDD	Difference
cost	~$120/TB	~$20/TB	6x
latency	5-100 µs	10ms	1,000x
IOPS	100k-millions	100-200	10,000x
bandwidth	0.5-14 GB/s	100-200 MB/s	100x
reliability	higher	lower	
density	getting higher	grows very slowly	
physical size	2.5“ or smaller	3.5“ most common	
power	fine-grained control and fast recovery	idle mode higher	

SSD basics

- What do they look like?



2.5” SATA

M.2 SATA

M.2 NVMe

consumer

U.2

E3.S

enterprise

SSD internal

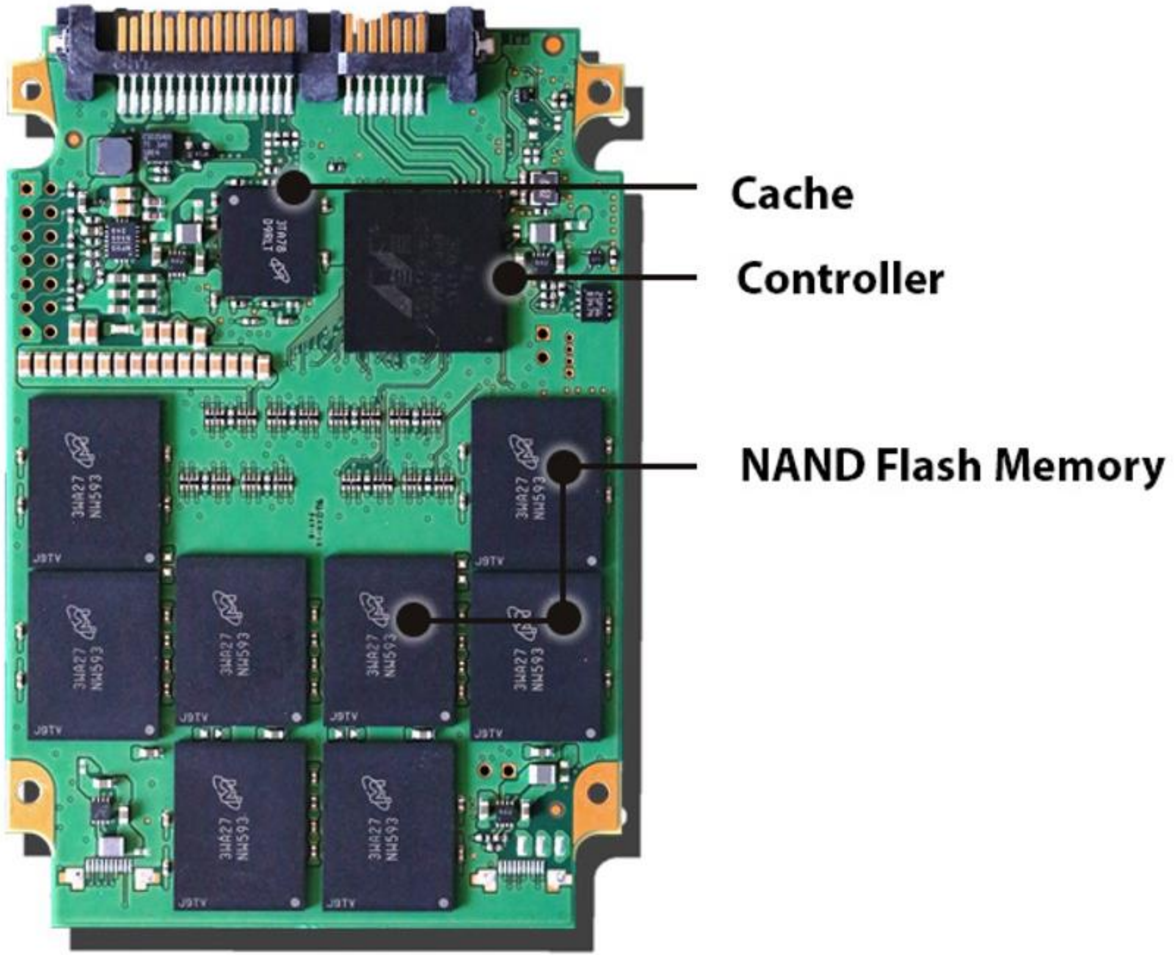
SSD technologies

- In 1970s and 1980s, SSD means “a drive with no moving parts”
  - RAMdisk: lots of DRAM with a battery
    - very fast, but very expensive and limited capacity
    - limited data retention, more realistic when coupled with a disk
- Modern SSD, ***almost*** exclusively imply NAND Flash
  - we use SSD to refer to NAND flash in the lecture
  - NOR: low density, slow write, word-accessible, low latency, small erasure unit (4-64KB), highly reliable
  - NAND: high density, write in *pages* (4KB), erase in *blocks* (4-16MB)

Confusing terms

SSD components

- NAND flash
  - storage medium
- Controller
  - FTL, command processing…
- DRAM cache
  - FTL mapping table
  - write buffer and read cache
  - ~1GB / TB capacity
  - optional nowadays
- Other
  - temperature sensors, PCB

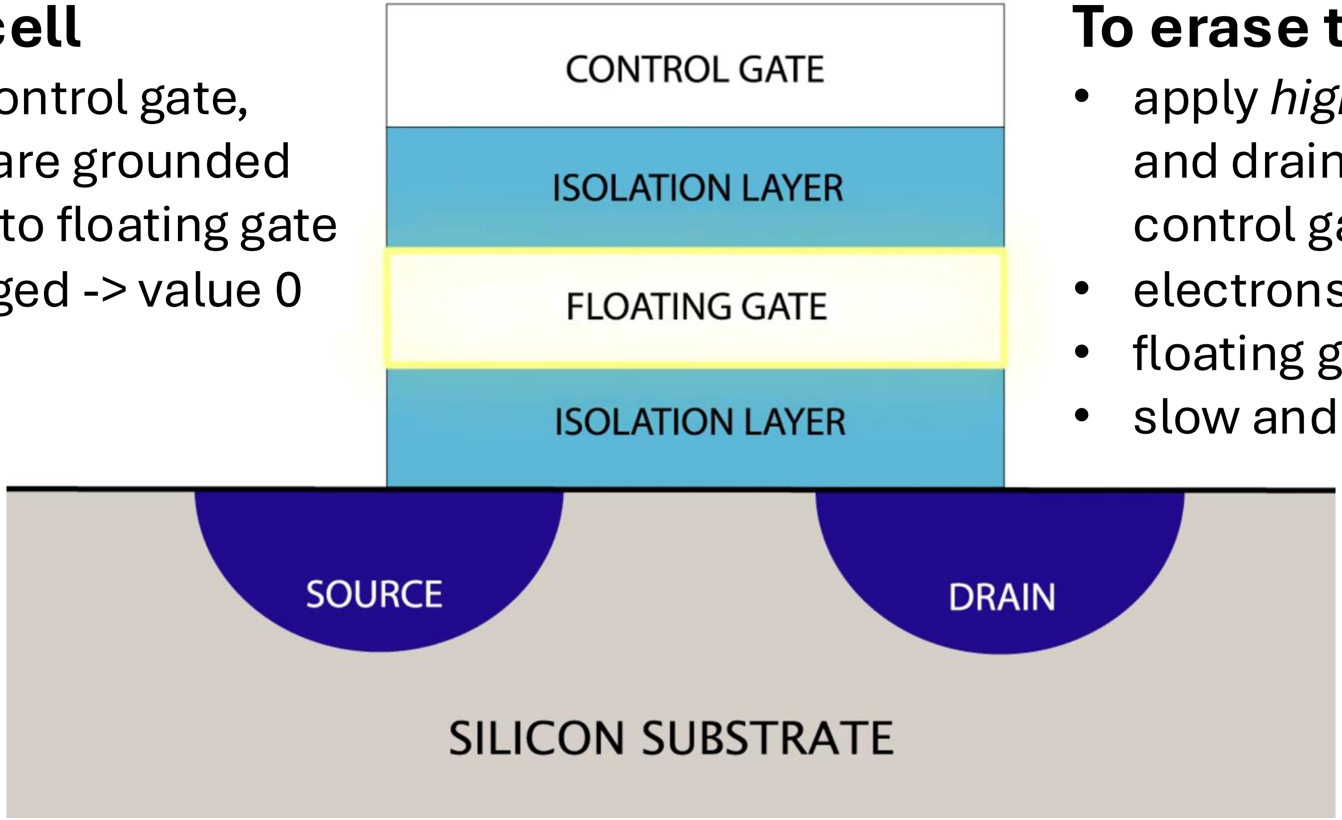


NAND flash cell

Based on MOSFET (Metal-Oxide-Semiconductor-Field-Effect) transistors + floating gate

To program the cell

- apply voltage to control gate, source and drain are grounded
- electrons move into floating gate
- floating gate charged -> value 0

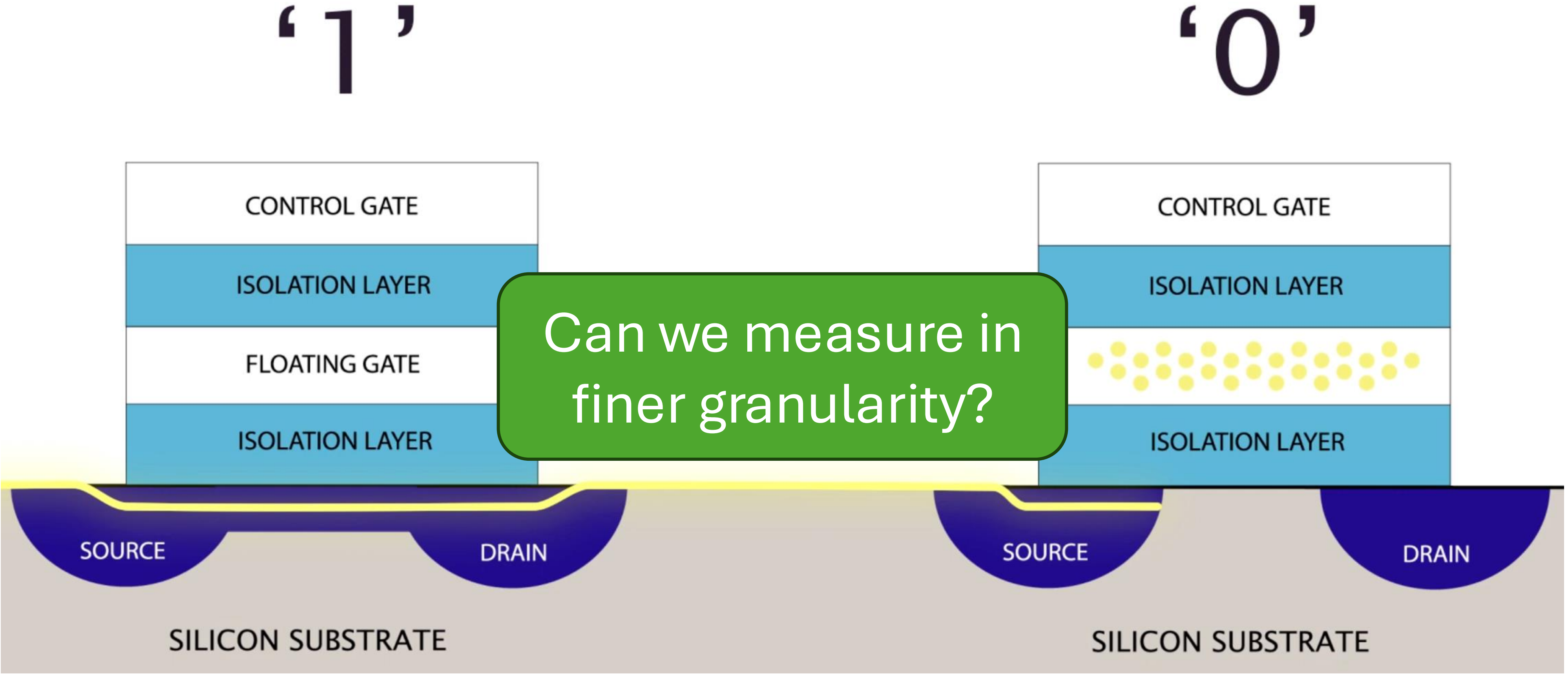


To erase the cell

- apply *high* voltage across source and drain, and negative voltage in control gate (flush)
- electrons leave floating gate
- floating gate NOT charged -> value 1
- slow and apply to all cells

NAND flash cell

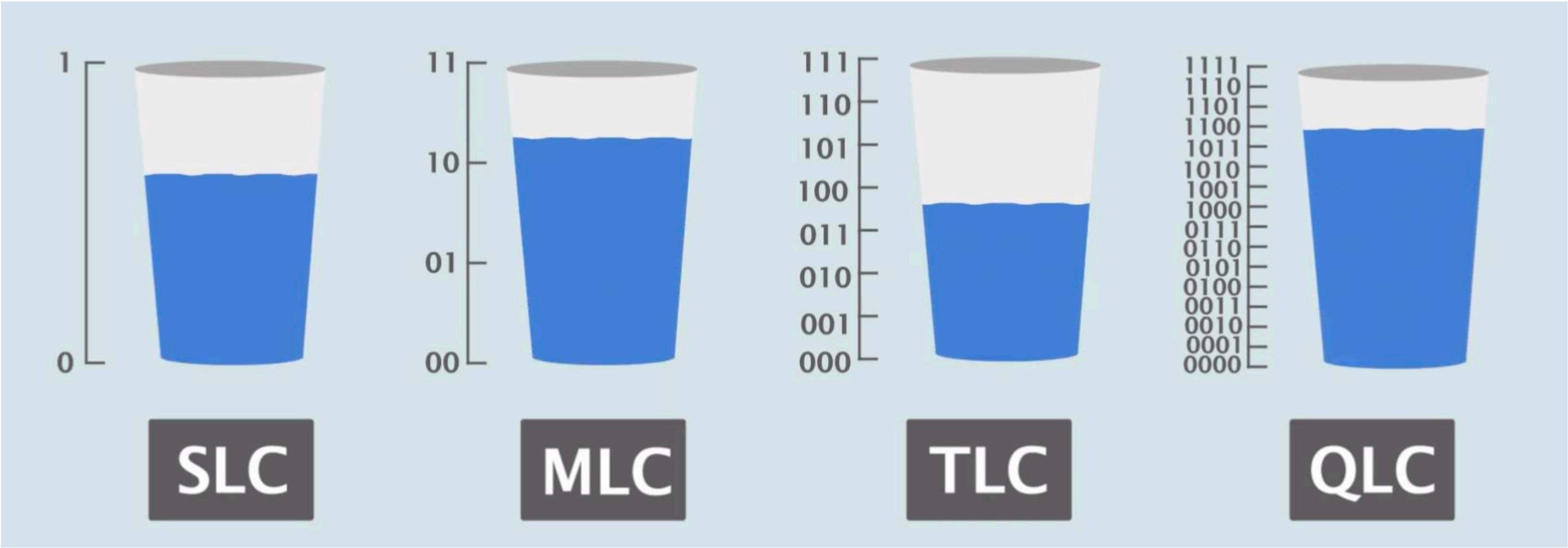
To read the value: apply a reference voltage across source and drain and measure the current



Can we measure in finer granularity?

NAND flash cell density

- Different amounts of binary information can be stored in each cell depending on the granularity of voltage thresholds
- SLC (1 bit per cell), MLC (2 bits per cell), TLC, QLC, and PLC
- voltage levels: 2^n

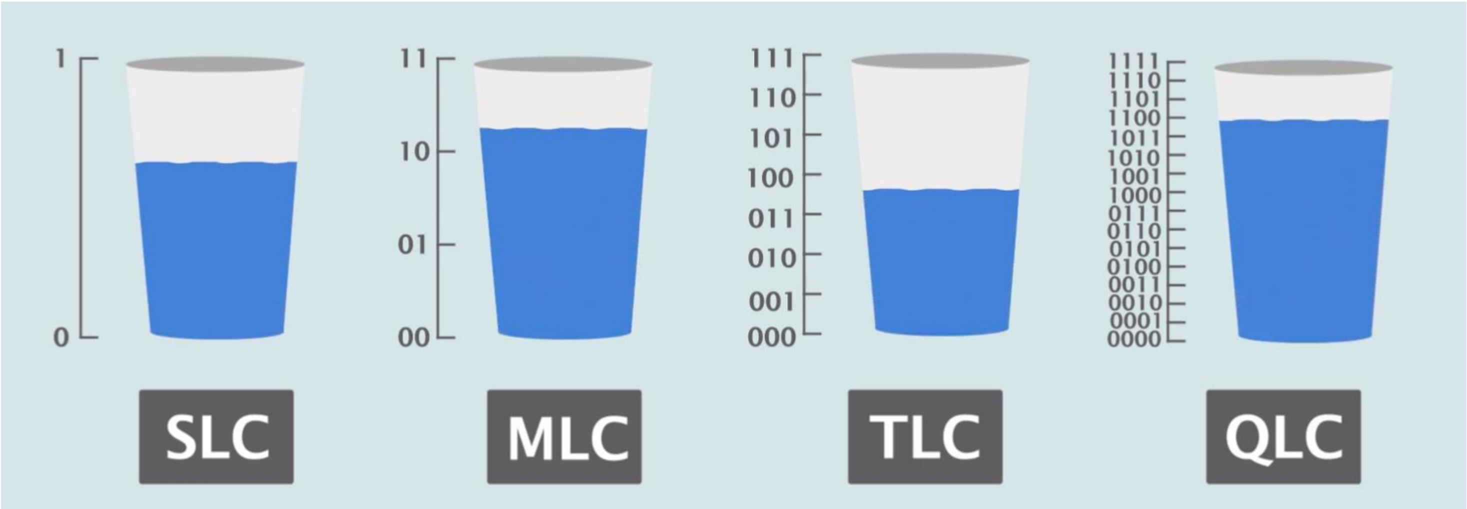


Store more data, but at what cost?

NAND flash cell density

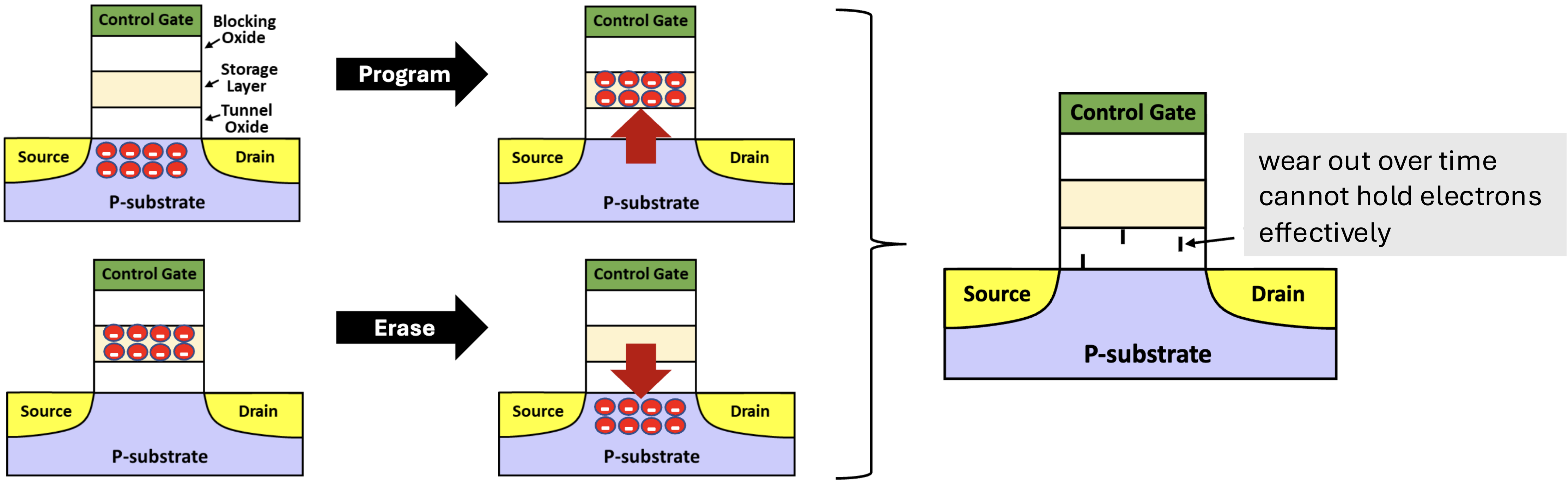
- Lower bit per cell, e.g., SLC
  - Fewer voltage levels -> simpler sensing
  - Lower error rates
  - Higher write speed , higher endurance
  - Higher cost per GB

- Higher bit per cell, e.g., QLC
  - More voltage levels -> tighter margins
  - Higher raw error rate, needs stronger ECC
  - Lower write speed, lower endurance
  - Higher density, lower cost per GB



NAND cell wear out

- Each program/erase (P/E) cycle degrades the cell's oxide over time



NAND cell wear out

- Each program/erase (P/E) cycle degrades the cell's oxide over time
- Wear manifests as higher raw bit error rate and lower retention
- **Endurance**: the number of program erase cycles a flash cell can undergo before data can no longer be retained effectively
  - also measured in TBW (TB written) and DWPD (disk write per day)
- Typical P/E cycles
  - higher bits/cell, e.g., QLC, typically tolerate fewer P/E cycles
  - SLC: 50,000–100,000, MLC: 3,000–10,000, TLC: 1,000–3,000, QLC: 100–1,000

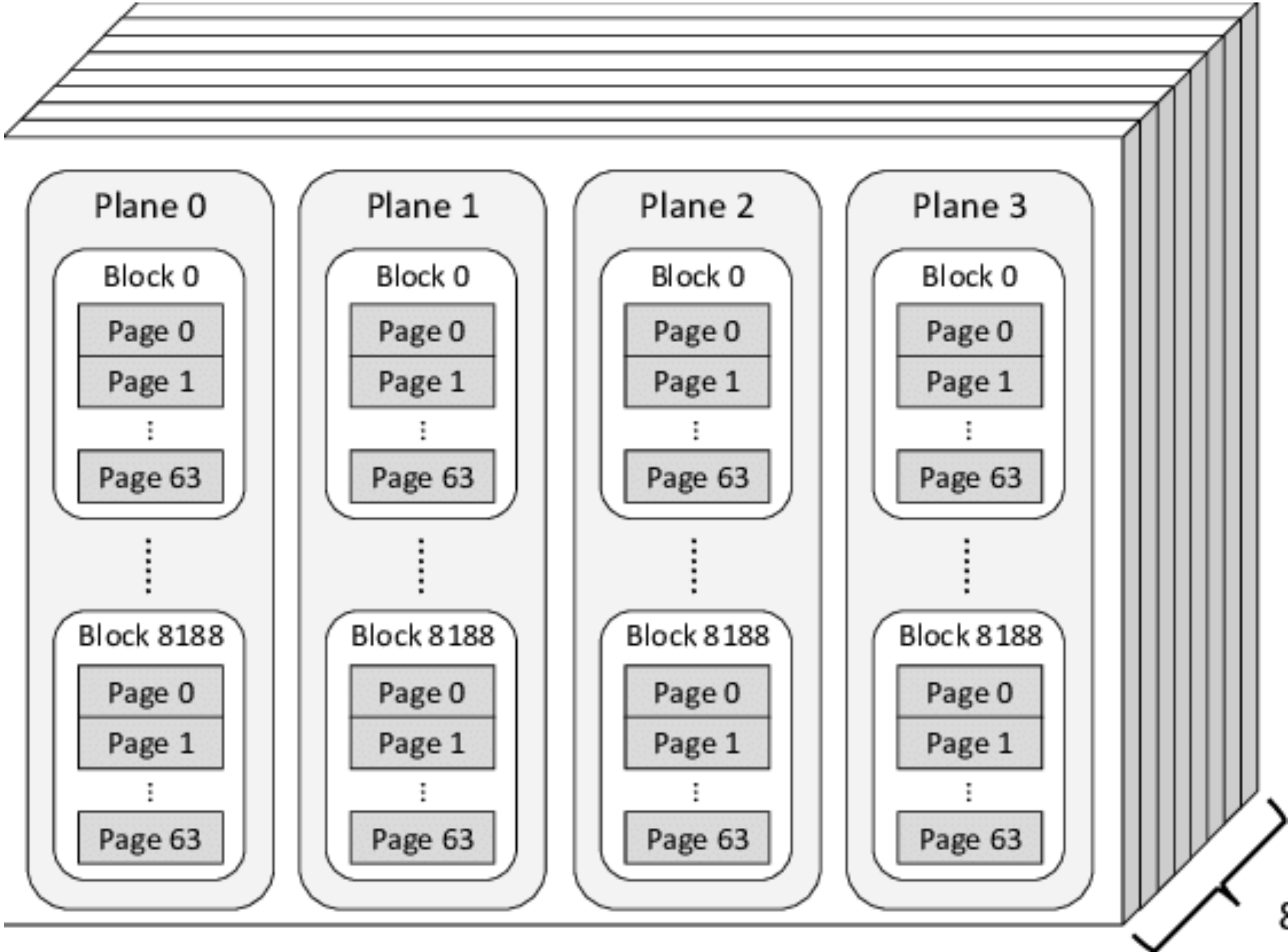
From Cell to SSD Architecture

SSD internal organization

Hierarchical structure

- Cell
- Page
- Block
- Plane
- Die
  - also referred to as LUN (logical unit number) in many contexts
- Package

Enable parallelism and high performance



SSD internal organization

- **Cell**: smallest unit, stores 1-5 bits
- **Page**: read/write unit*, 4KB, 8KB, or 16KB (common in QLC)
- **Block**: erase unit*, typically 2–4 MB, up to 16 MB (QLC)
- **Plane**: a group of blocks, 2–8k, up to 16k
- **Die**: 2-8 planes, mostly common 4
- **Package**: physical NAND chip
  - consumer SSDs: 2–8 NAND chips
  - enterprise SSDs: 16–32+ chips for parallelism
  - more chips = more parallel channels = higher performance

*You cannot overwrite a page—must erase entire block first

SSD components: others

- Controller: next slide
- DRAM:
  - FTL mapping tables
  - buffers writes and supports coalescing/merging
  - DRAM size decides mapping table size and flash reads
  - NVMe Host Memory Buffer (HMB) can expose host RAM as a cache
- Power Loss Protection (PLP)
  - capacitors and flush logic
- Thermal sensors:
  - thermal throttling to reduce error and wear out

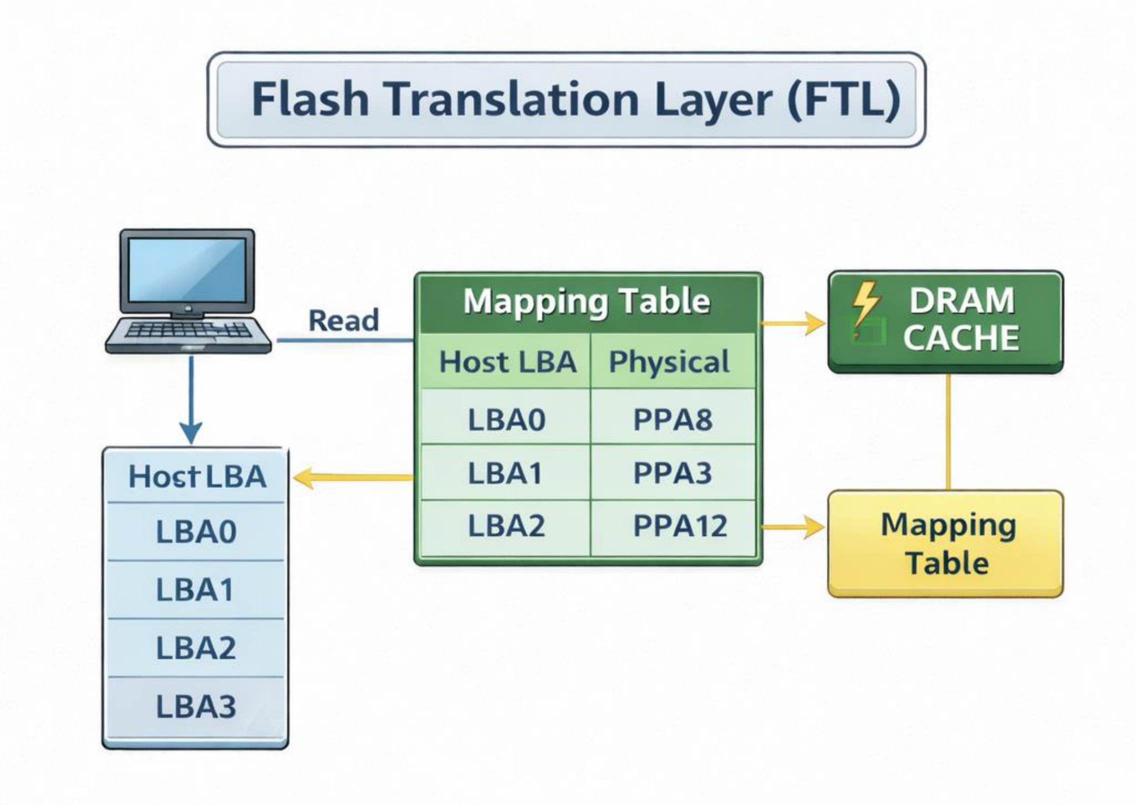
SSD controller and firmware

SSD controller

- Controller's role
  - flash translation layer (FTL): mapping, wear-leveling, garbage collection
  - command processing and scheduling: maximize performance
  - reliability related functions: ECC and power loss protection
  - others: encryption, metadata operation, DMA engine
- Controller is often the performance bottleneck, not NAND speed
  - more cores and channels lead to better performance
  - high-end drives often achieve close to line rate

Flash translation layer: why do we need FTL?

- Logical-to-physical (L2P) abstraction
- Write / erase asymmetry
  - read/write unit: page
  - erase unit: block
  - no in-place overwrite
  - update written to a new block
  - how should I find it later?
  - what to do with invalidated page?



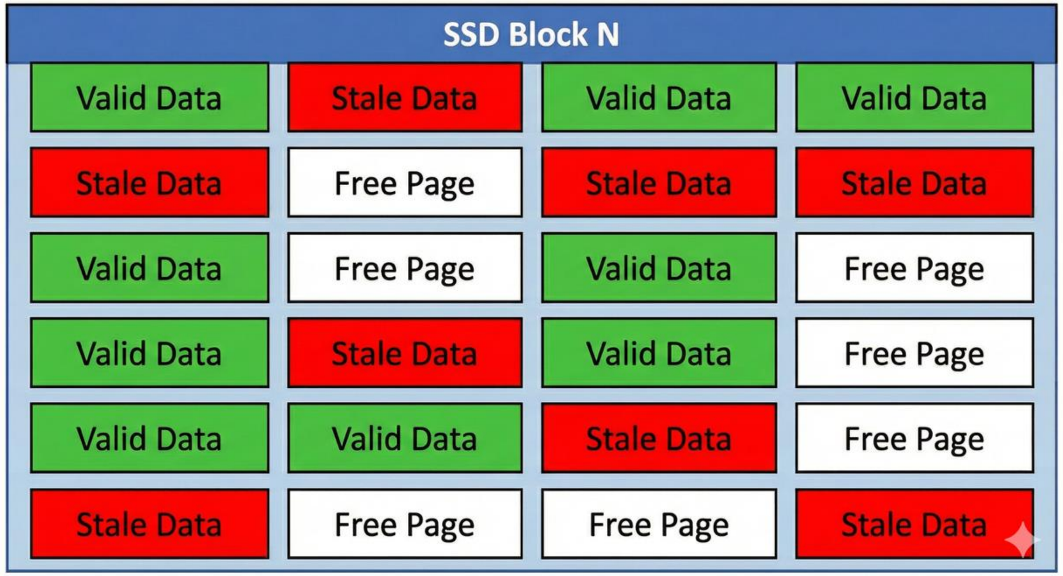
Flash translation layer: mapping granularity

- Page Level Mapping
  - standard, high RAM usage
  - table size: (Capacity / 4KB) x 4 bytes, 1 GB DRAM per TB NAND
- Block Level Mapping
  - historical, slow random write, huge write amplification
- Hybrid Mapping
  - data blocks: block level for most of the data
  - log blocks: page level for recent page updates
  - read first check log blocks then data blocks

Flash translation layer: garbage collection

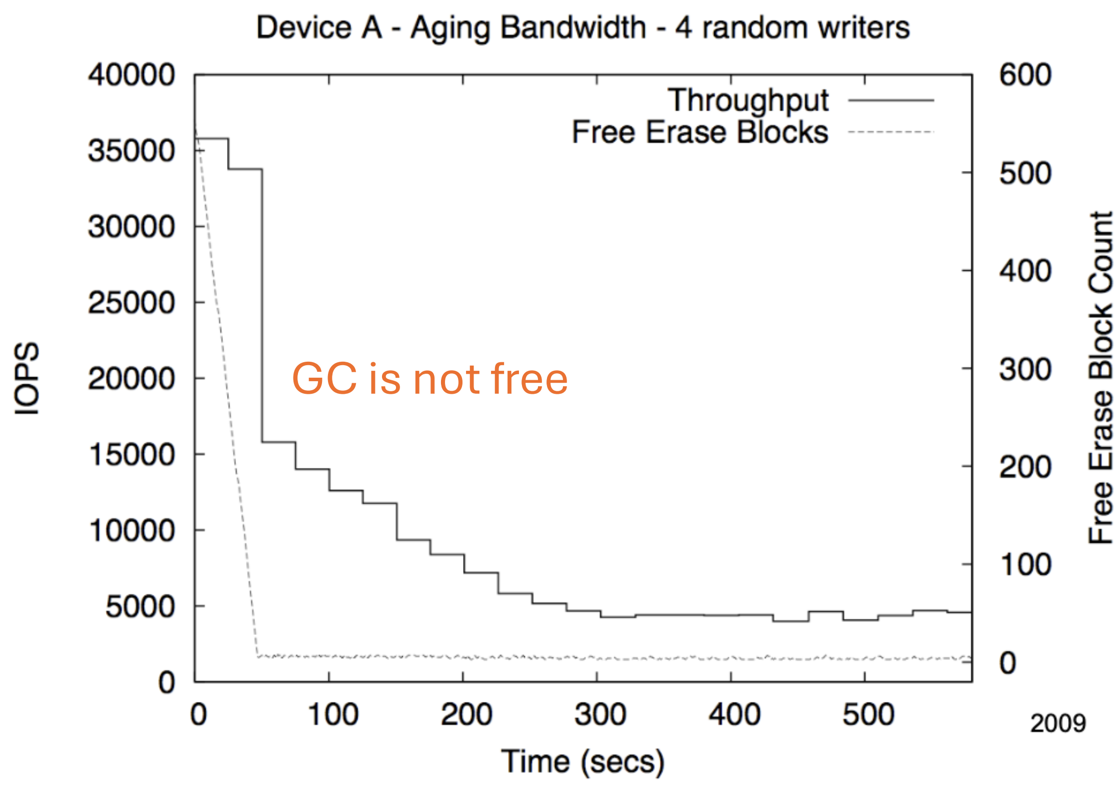
Select victim block → **Copy valid pages** → **Update mapping** → **Erase block** → **Return free space**

- Problem
  - some pages in a block are invalidated
- Solution
  - merge multiple partially-valid blocks
- Garbage collection is hard
  - may block other operations and increase latency
  - cause write amplification
- Block selection and scheduling are major firmware differentiator



Flash translation layer: garbage collection

- **When** to GC?
  - during background low I/O time
  - when free blocks < threshold
  - late: may block writes
  - early: some copied page may be invalidated soon after gc
- **Which** block to clean?
  - greedy: block with fewest valid pages
  - cost-benefit: considers age or wear
    - prefer *older* block with *fewer* valid pages
  - hot/cold separation: reduce mixing to lower copy cost



Flash translation layer: garbage collection

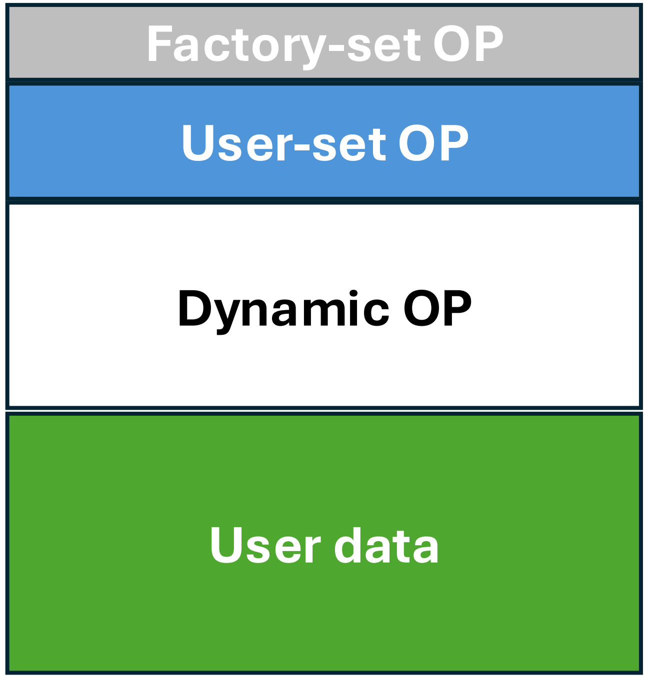
- Measure GC cost: write amplification factor (WAF):
  - definition:

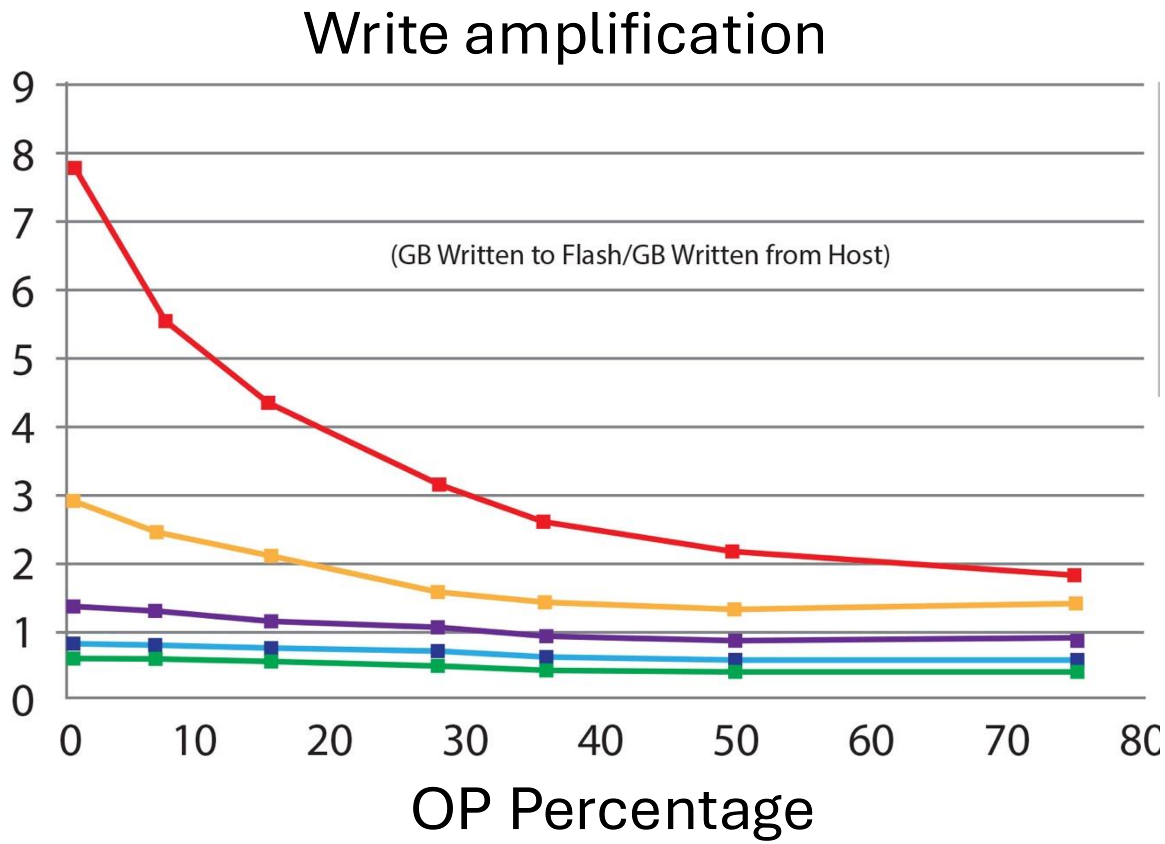
$$WAF = \frac{\text{Flash write bytes}}{\text{Host write bytes}}$$

  - impact on SSD lifespan and throughput
  - application WA vs. device WA
- Source of write amp
  - GC copy, small update, wear leveling move, metadata update

Flash translation layer: garbage collection

- Reducing GC and write amplification
  - over-provisioning
  - Consumer: 7-12%, Enterprise: 20-28%





Flash translation layer: garbage collection

- Reducing GC and write amplification
  - Trim/discard
    - deletes at filesystem level don’t automatically tell the SSD
    - TRIM/discard marks LBAs as unused so GC can skip copying them

Flash translation layer: other functions

- Wear Leveling
  - distribute writes evenly across all blocks to prevent some blocks from wearing out faster than others
  - static: move cold data
  - dynamic: spread new writes across low-wear blocks
  - trade-off: static is more effective but can increase WA
- Bad block management
  - reserved space (overprovisioning) to replace bad block
  - FTL maintains block state and remaps unusable blocks

Request flow

Read flow

- **Host** sends read command via NVMe interface
- **Controller** checks DRAM cache, return on cache hit
- **Cache miss:** looks up FTL mapping table (in DRAM)
- **Flash interface** sends command to appropriate NAND chip/channel
- **NAND** reads page
- **ECC engine** checks and corrects bit errors
- **Data** transferred through controller to host interface

Write flow

- **Host** sends write command via NVMe
- **Controller** receives data, stores in DRAM write buffer
- **Acknowledge to host** (write complete from host perspective)
- **Controller** decides where to physically write:
  - Checks FTL for free pages
  - Applies wear leveling algorithm
  - ECC encode and per-frame CRC calculation
- **Write to NAND:**
  - If TLC/QLC: First write to SLC cache
  - Later: Migrate to TLC/QLC during idle (folding)
- **Update FTL mapping table** in DRAM and periodically flush to NAND