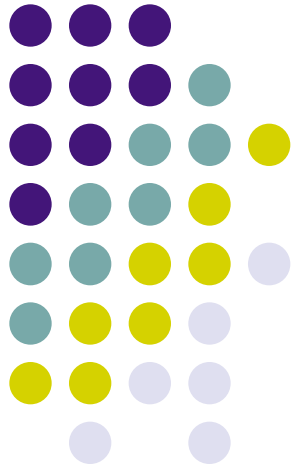
Storage Devices

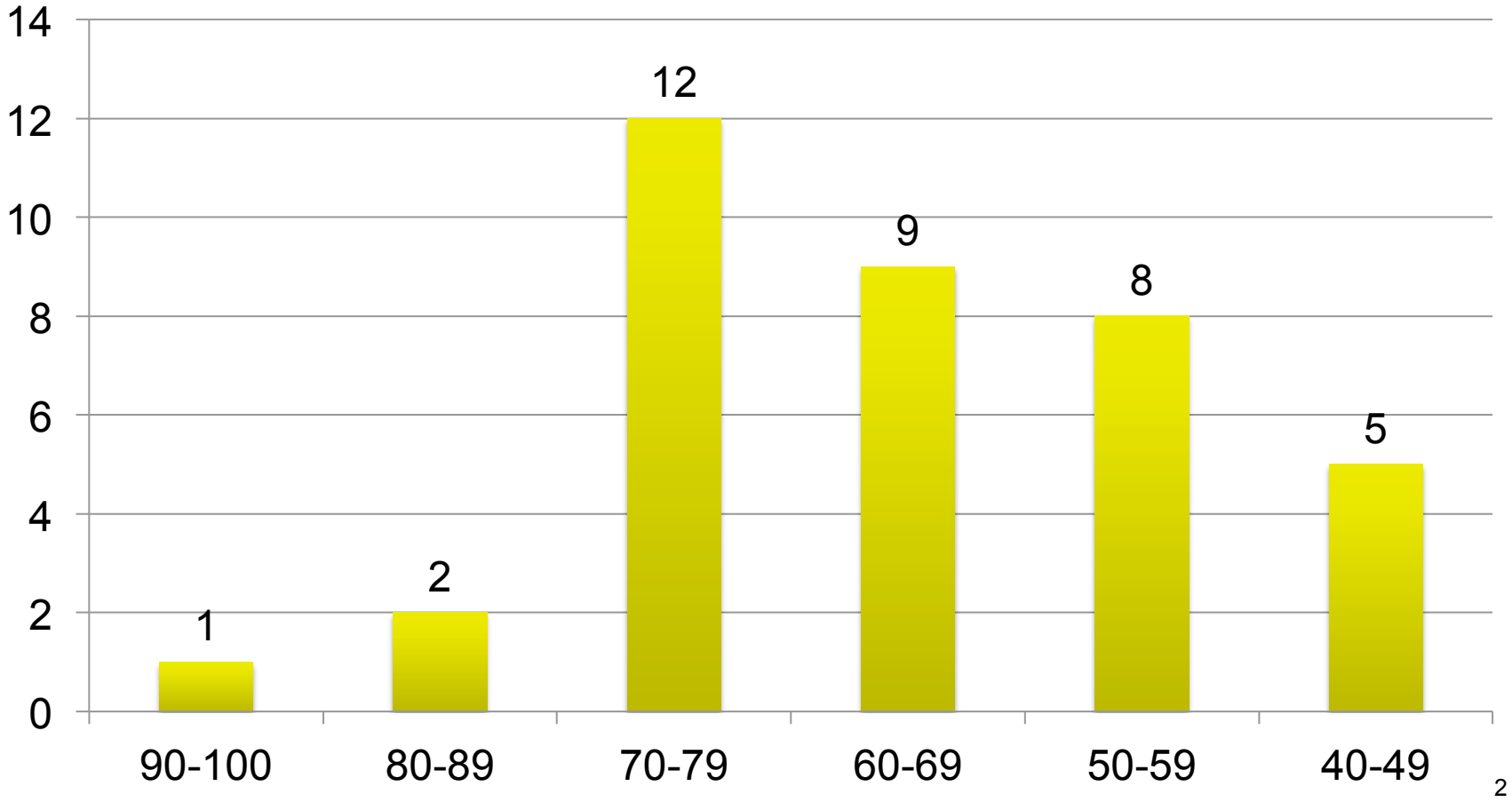
ECE 469, March 28

Yiying Zhang

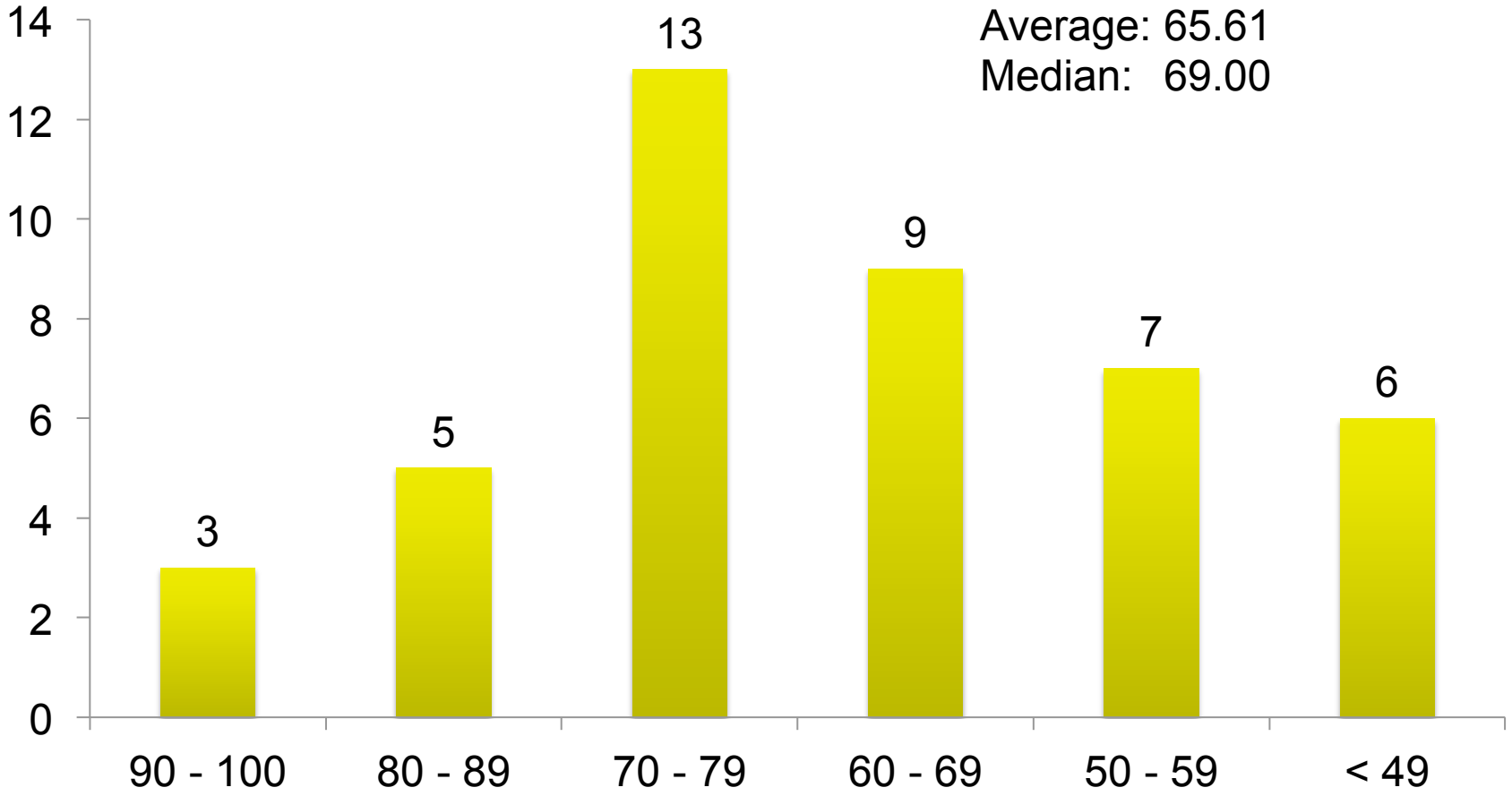


S17 Midterm Stats

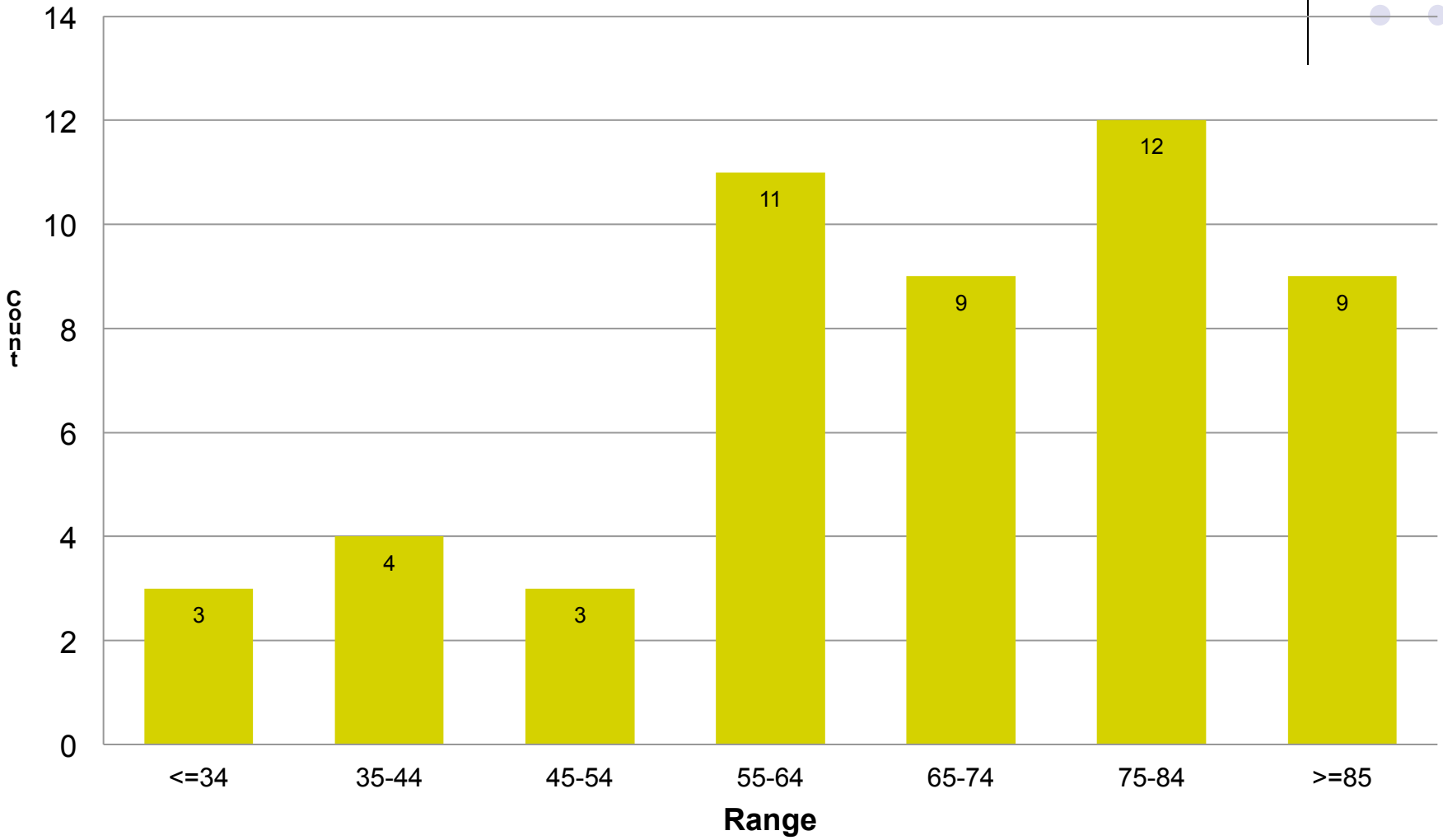
Average: 65.16
Median: 63.00



S16 Midterm Stats

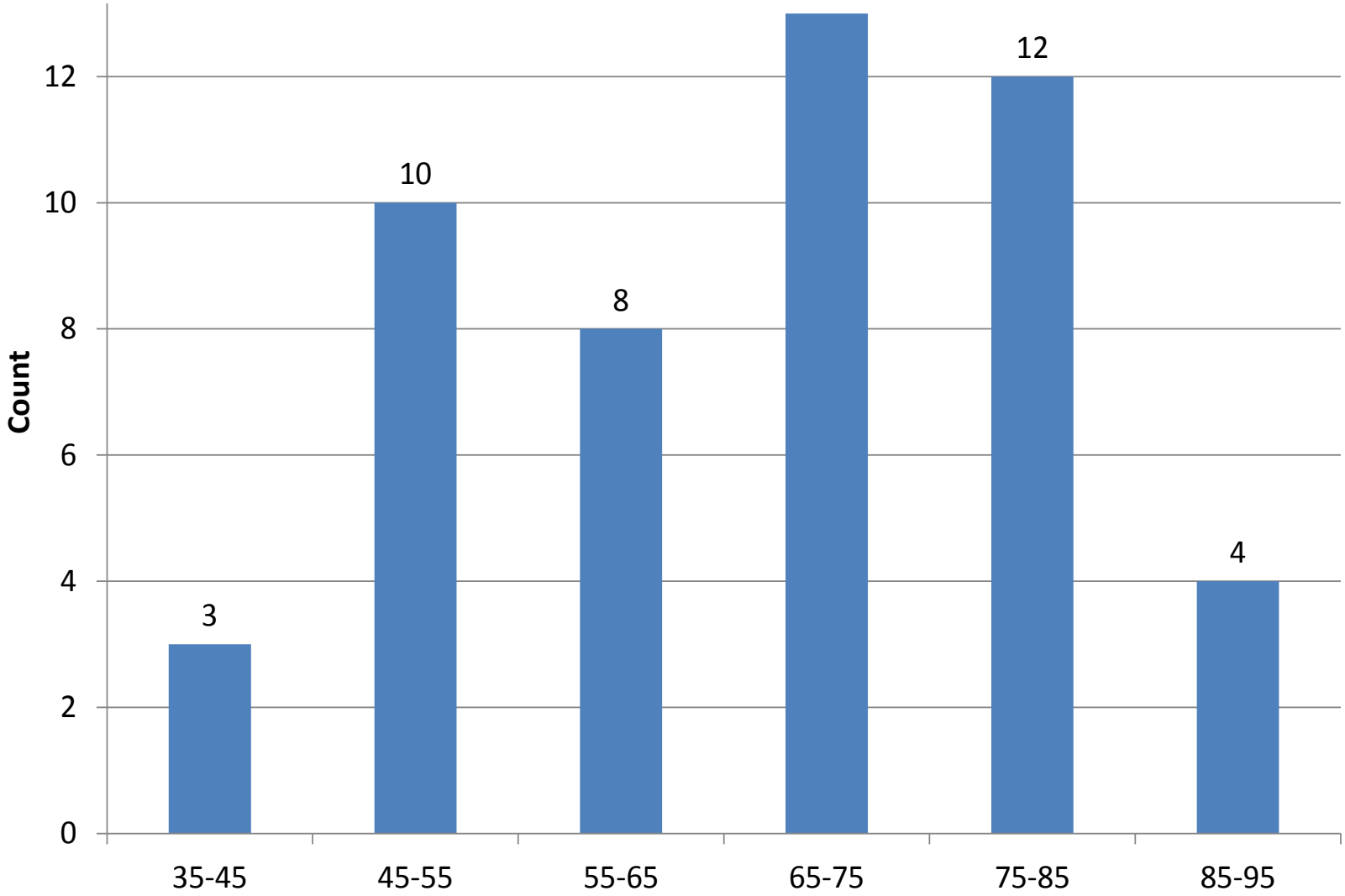


S15 Midterm Distribution

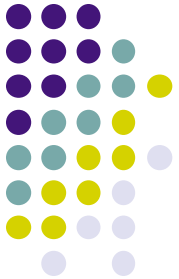


Avg = 67.8

S14 Midterm Distribution

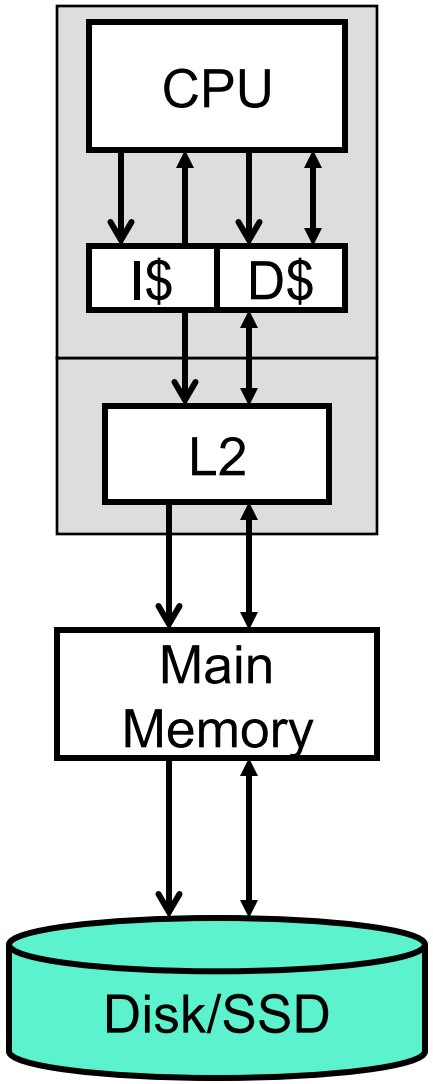


Storage



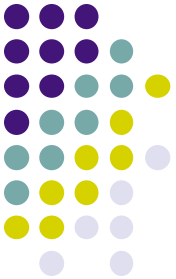
- Reading: Ch 12

Memory Hierarchy

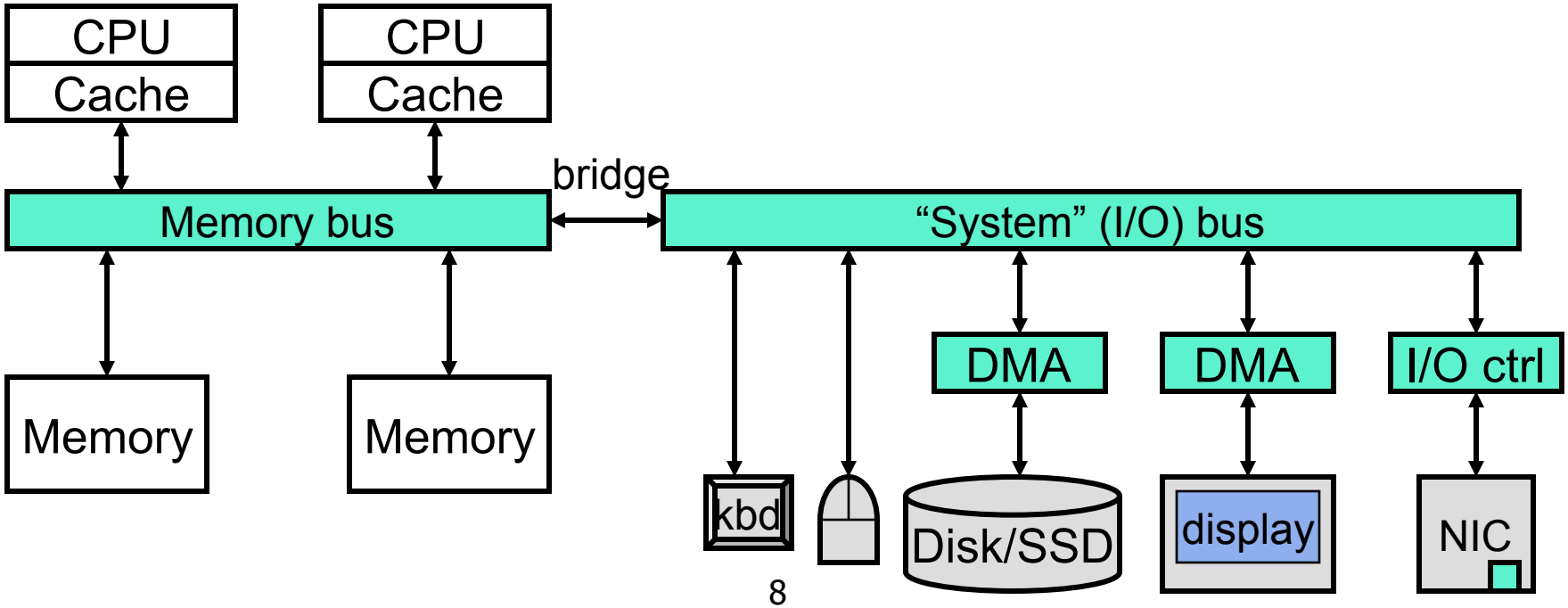


- Storage device (e.g., Disk, SSD)
  - bottom of memory hierarchy

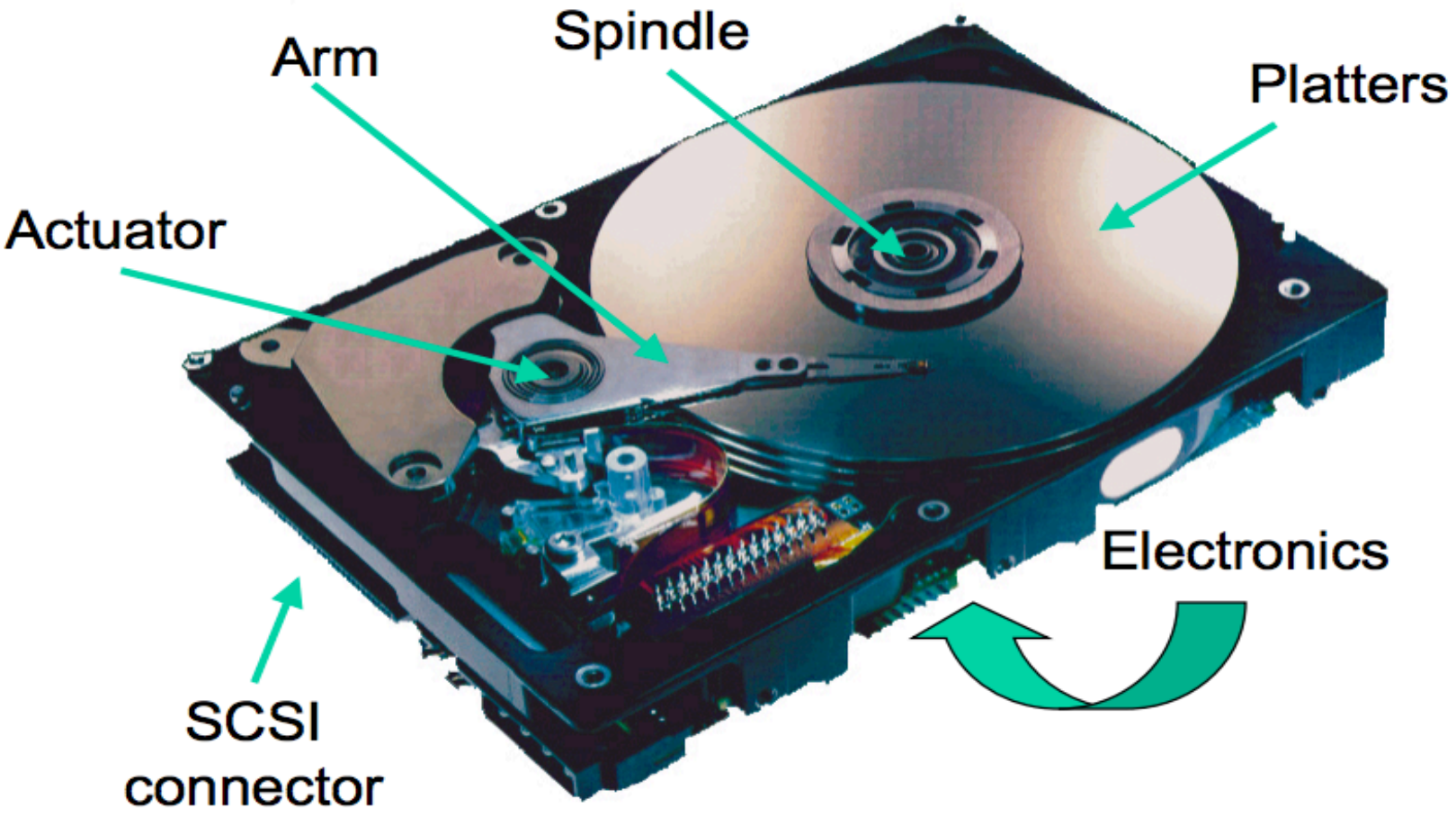
A More General/Realistic I/O System



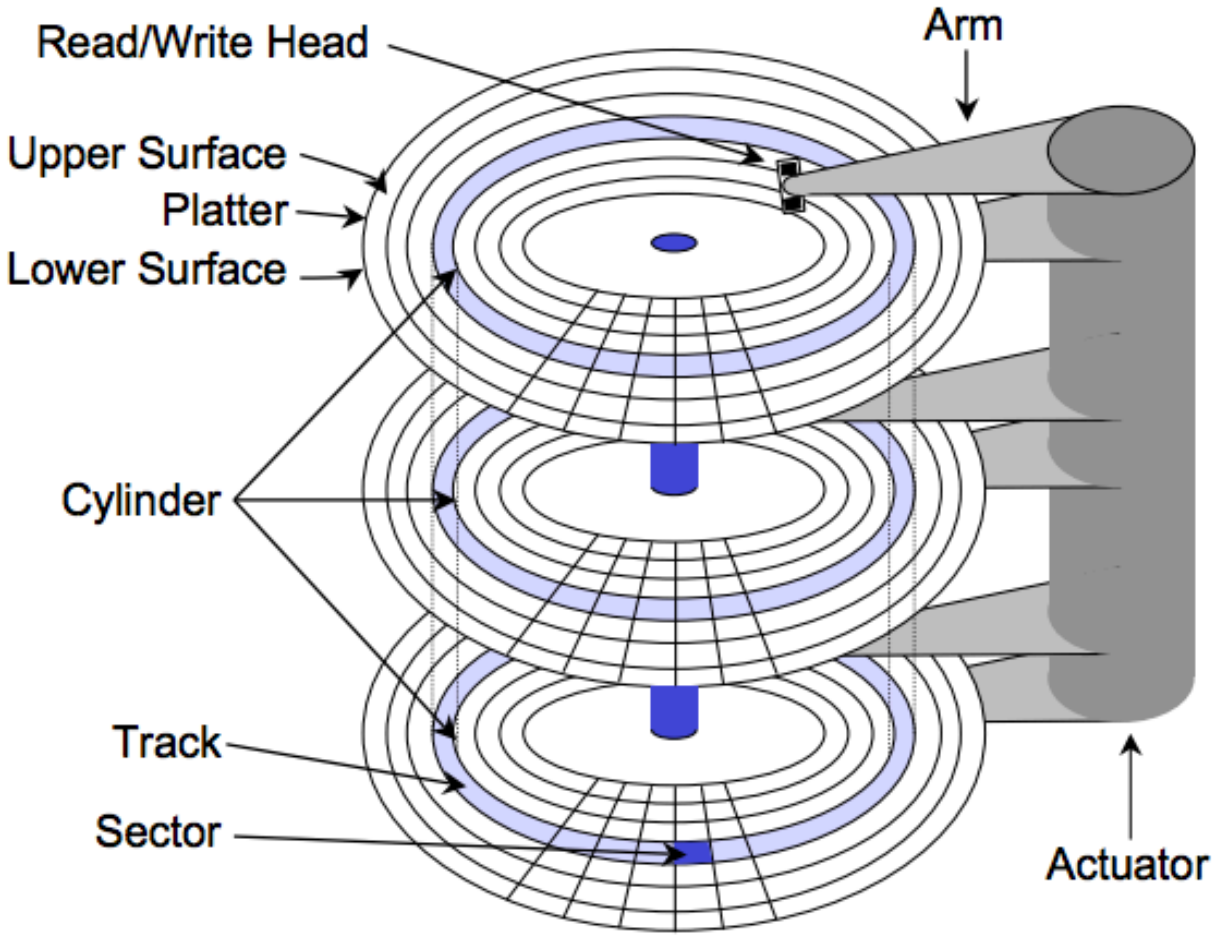
- I/O peripherals: disks, input devices, displays, network cards, ...
  - With built-in or separate I/O (or DMA) controllers
  - All connected by a system bus



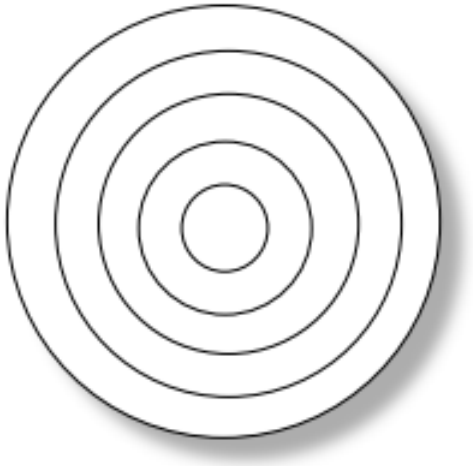
What's Inside a Disk Drive?



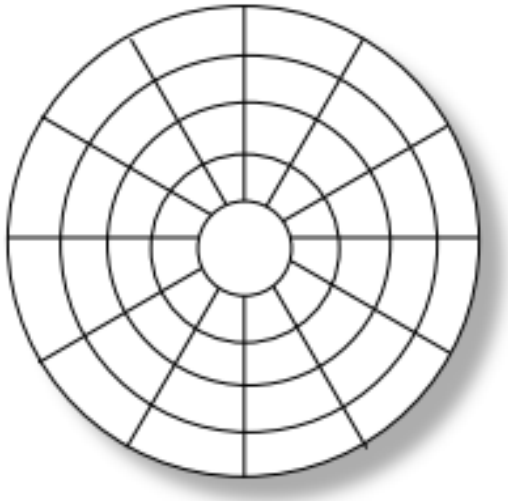
Disk Components



Surface Organized into Tracks

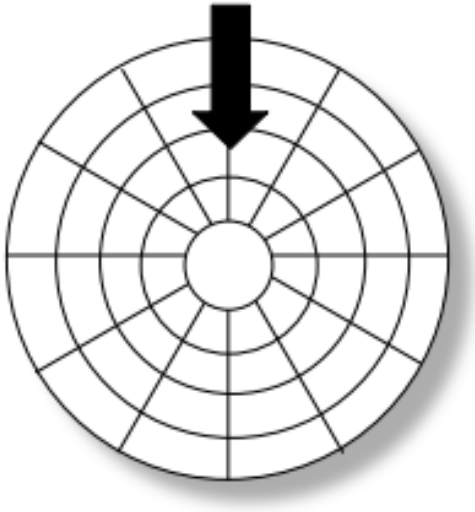


Tracks Broken up into Sectors

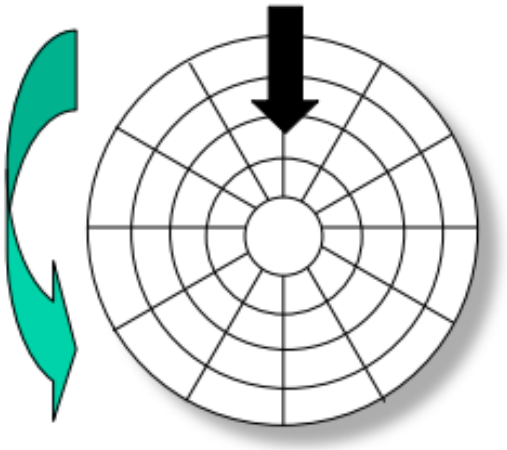


- Disk accesses in the granularity of a sector (usually 512KB)
- This I/O interface is called block I/O interface

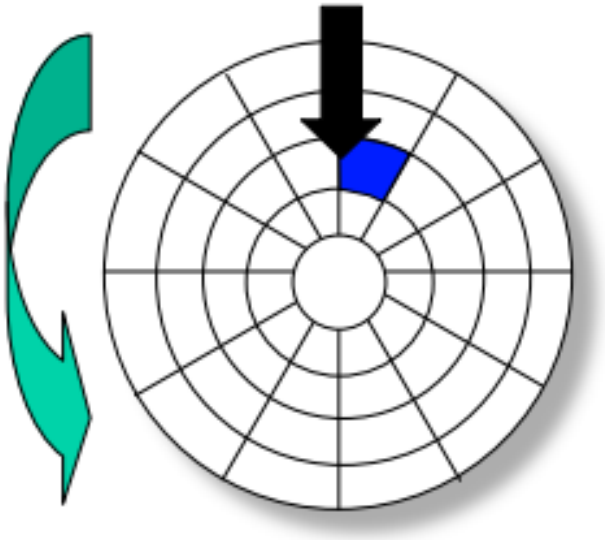
Disk Head Position



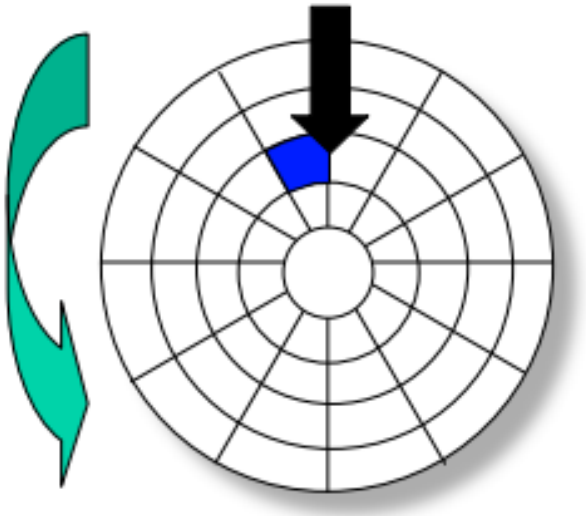
Rotation is Counter-Clockwise



About to Read Blue Sector

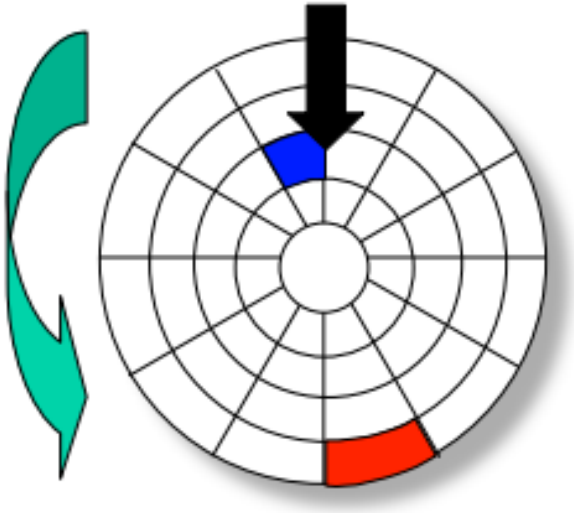


After Reading Blue Sector



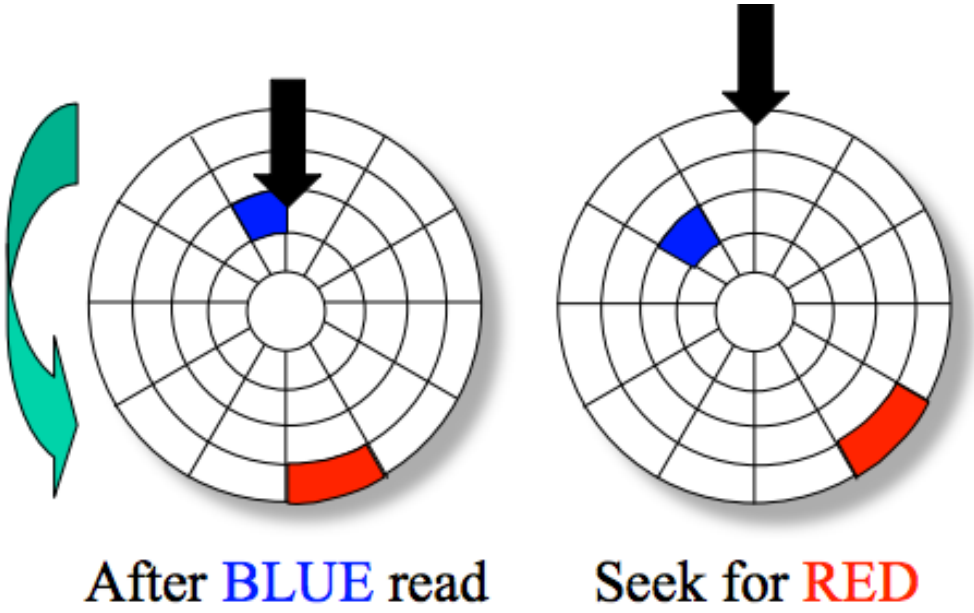
After BLUE read

Red Request Scheduled Next

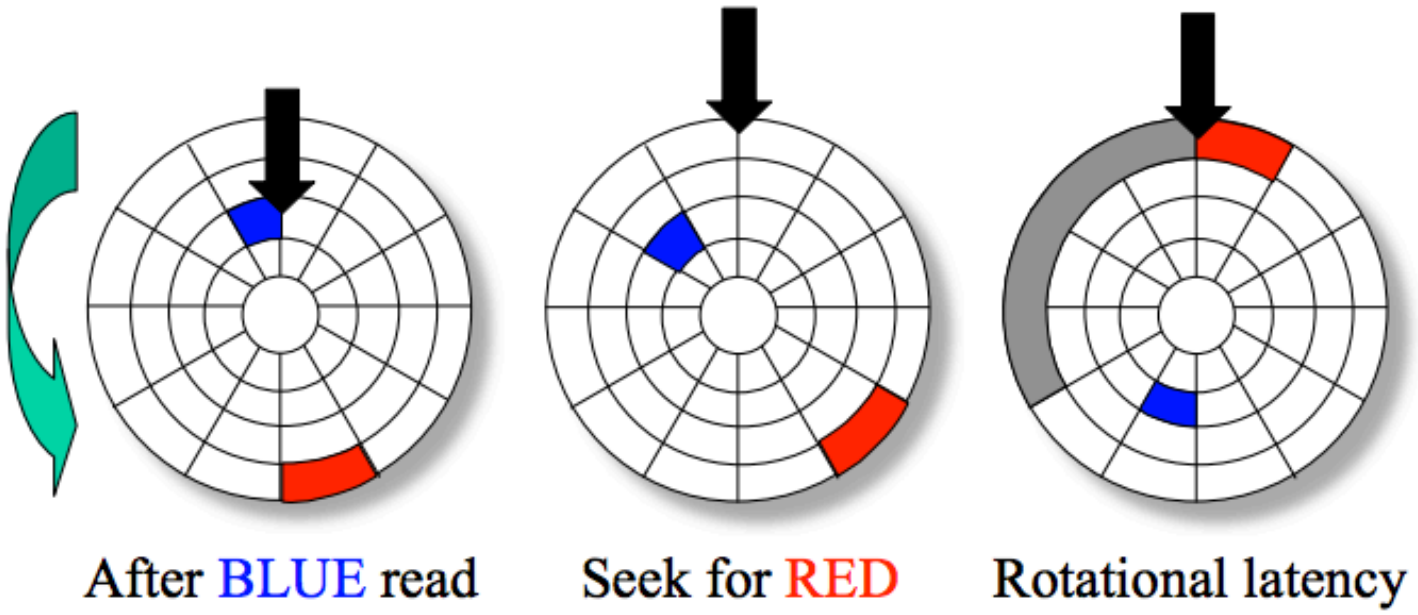


After BLUE read

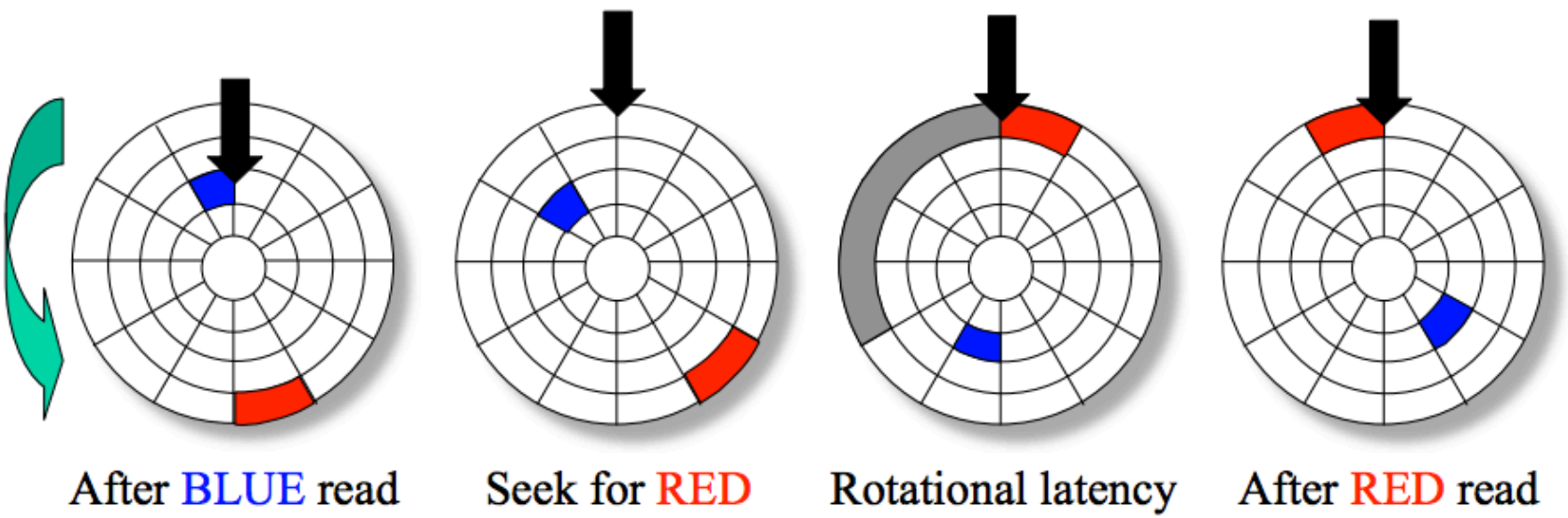
Seek to Red's Track



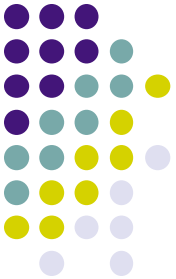
Wait for Red Sector to Reach Head



Read Red Sector



Some real numbers for modern disks



- # of platters: 1-8

– 2-16 surfaces for data

- # of tracks per surface: 10s of 1000s

– same thing as # of cylinders

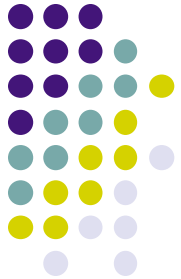
- # sectors per track: 200-1000

– so, 100-500KB

- # of bytes per sector: usually 512

– can be chosen by OS for some disks

Response Time for Disks



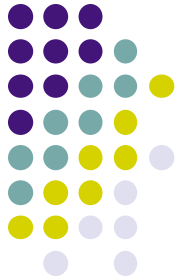
- Access time: (service time for a disk access)

– Command + Seek + Rotation + Transfer

- Response time:

– Queue time + Access time

Seek Time

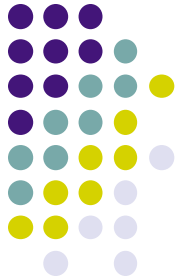


- Time required to move head over desired track
  - Physically moving the head, not electronic => slow!

- A seek has up to four components
  - accelerate
  - coast at max velocity
  - decelerate
  - settle onto correct track

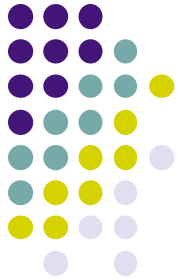
- Seek time depends on workload
  - For random workloads, longer seek time

Rotational Latency



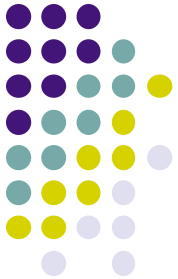
- Time required for the first desired sector to reach head
- Depends on rotation speed
  - measured in Rotations Per Minute (RPMs)
- Computing average rotational latency
  - for almost all workloads, we can safely assume that there is an equal likelihood of landing on any sector of the track
  - this gives equal probability of each rotational latency
    - from 0 sectors to N-1 sectors
  - thus, average rotational latency is time for 1/2 revolution
  - e.g., for 7200 RPM
    - one rotation = 60s / 7200 = 8.33 ms
    - average rotational latency = 4.16 ms

Modern Disk Performance Characteristics



- Seek times: 0.5-15ms, depending on distance
  - average 5-6ms
  - improving at 7-10% per year
- Rotation speeds: 5600-15000 RPMs
  - improving at 7-10% per year

Disk Bandwidth: Sequential vs Random

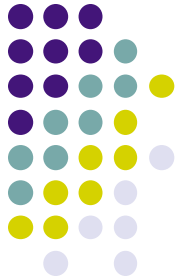


- Disk is bandwidth-inefficient for page-sized transfers
  - Sequential vs random accesses

- **Random accesses:**
  - Need seeks, slow (one random disk access latency ~10ms)
  - Randomly reading 4KB pages: ~400KB/sec
- **Sequential accesses:**
  - Stream data from disk (no seeks)
  - 128 sectors/track, 512 B/sector, 6000 RPM
    - 64KB per rotation, 100 rotation/per sec
    - 6400KB/sec ➔ 6.4MB/sec

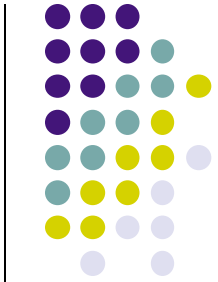
- Sequential access is ~10x or more bandwidth than random
  - Still no where near the 10sGB/sec of memory

Disk Reliability



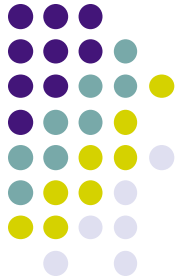
- Disks fail more often....
  - When continuously powered-on
  - With heavy workloads
  - Under high temperatures
- How do disks fail?
  - Whole disk can stop working (e.g., motor dies, firmware errors)
  - Transient problem (cable disconnected, firmware errors)
  - Individual sectors can fail (e.g., head crash or scratch)
    - Data can be corrupted or block not readable/writable
- Disks can internally fix some sector problems
  - ECC (error correction code): Detect/correct bit flips
  - Retry sector reads and writes: Try 20-30 different offset and timing combinations for heads
  - Remap sectors: Do not use bad sectors in future
    - How does this impact performance contract??

Disk Buffering



- Disks contain internal memory (2MB-16MB) used as cache
- Read-ahead: “Track buffer”
  - Read contents of entire track into memory during rotational delay
  - What does this resemble? – memory prefetching
- Write caching with volatile memory
  - Immediate reporting: Claim written to disk when not
  - Data could be lost on power failure
- Command queuing
  - Have multiple outstanding requests to the disk
  - Disk can reorder (schedule) requests for better performance

Disk Scheduling



- Goal: Minimize positioning time

- First come, first serve (FCFS): requests are served in the order of arrival
  + Fair among requesters
  - Poor for accesses to random disk blocks

- Shortest seek time first (SSTF): picks the request that is closest to the current disk arm position
  + Good at reducing seeks
  - May result in starvation

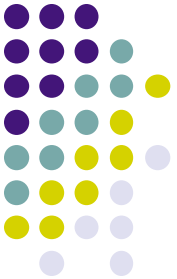
Jim Gray

- Database/Systems
- Database Locking
- Replication

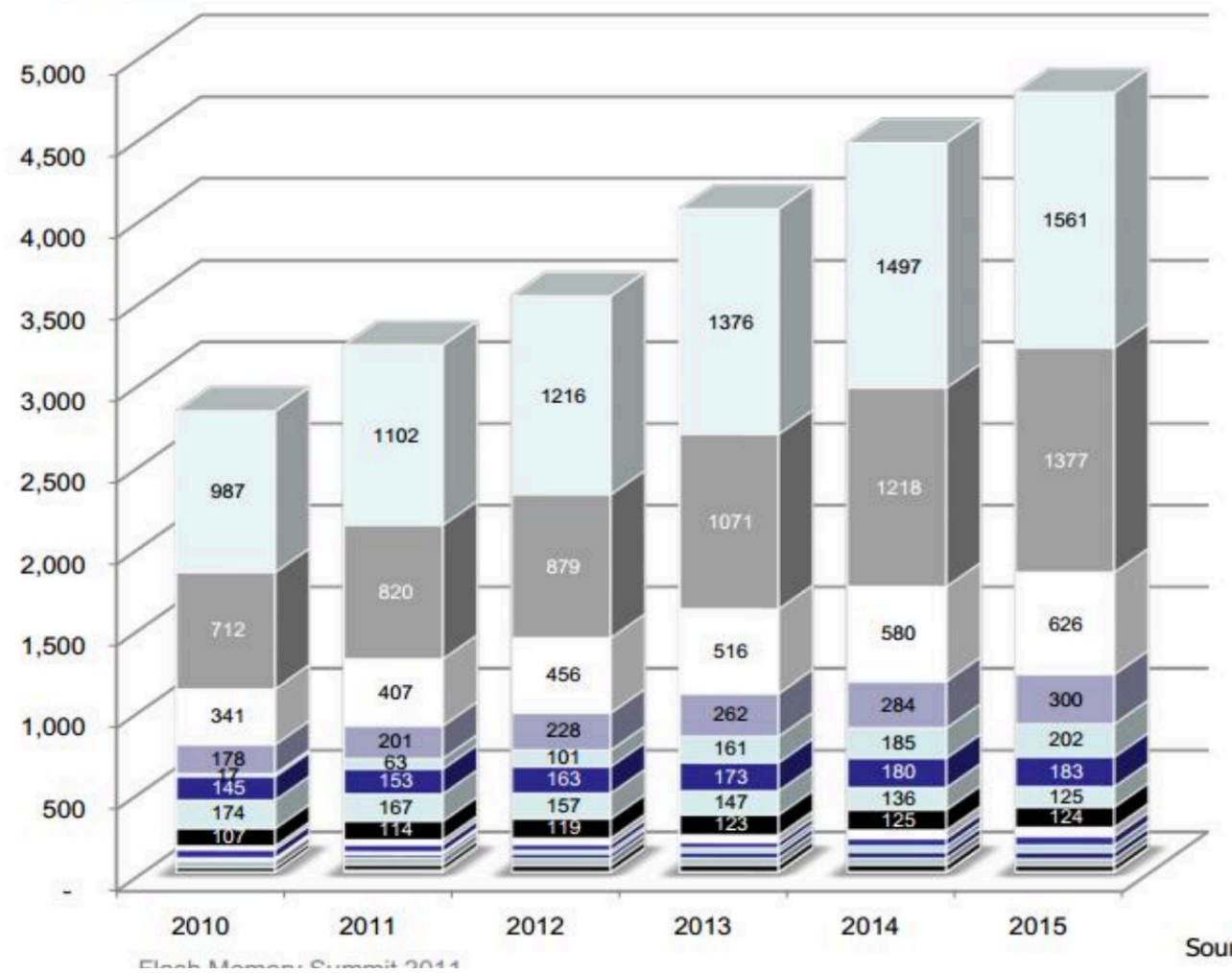
Edgar Codd

- Databases
- Relational Database

Flash Memory and Flash-Based SSDs

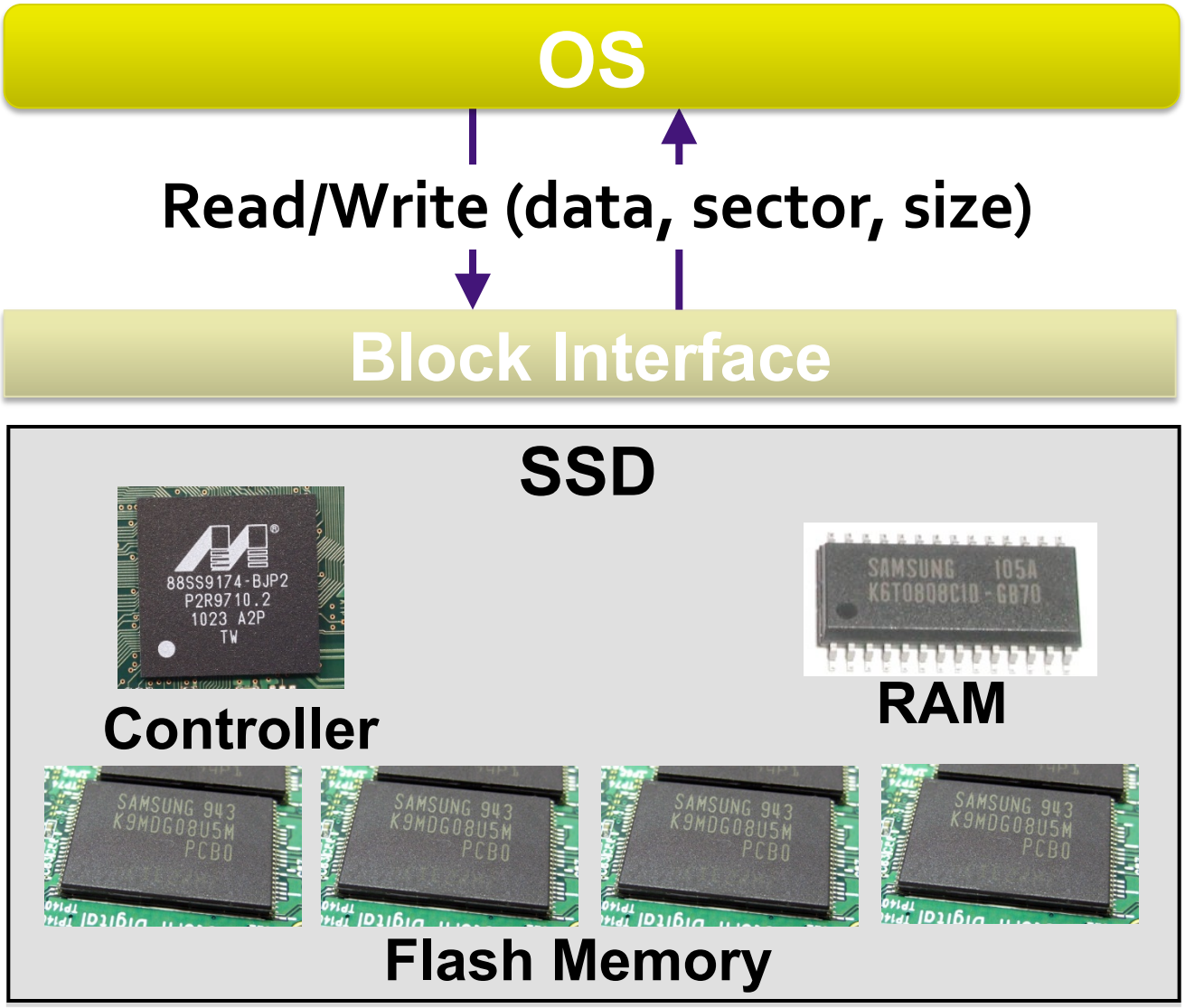


Number of NAND flash units (millions)

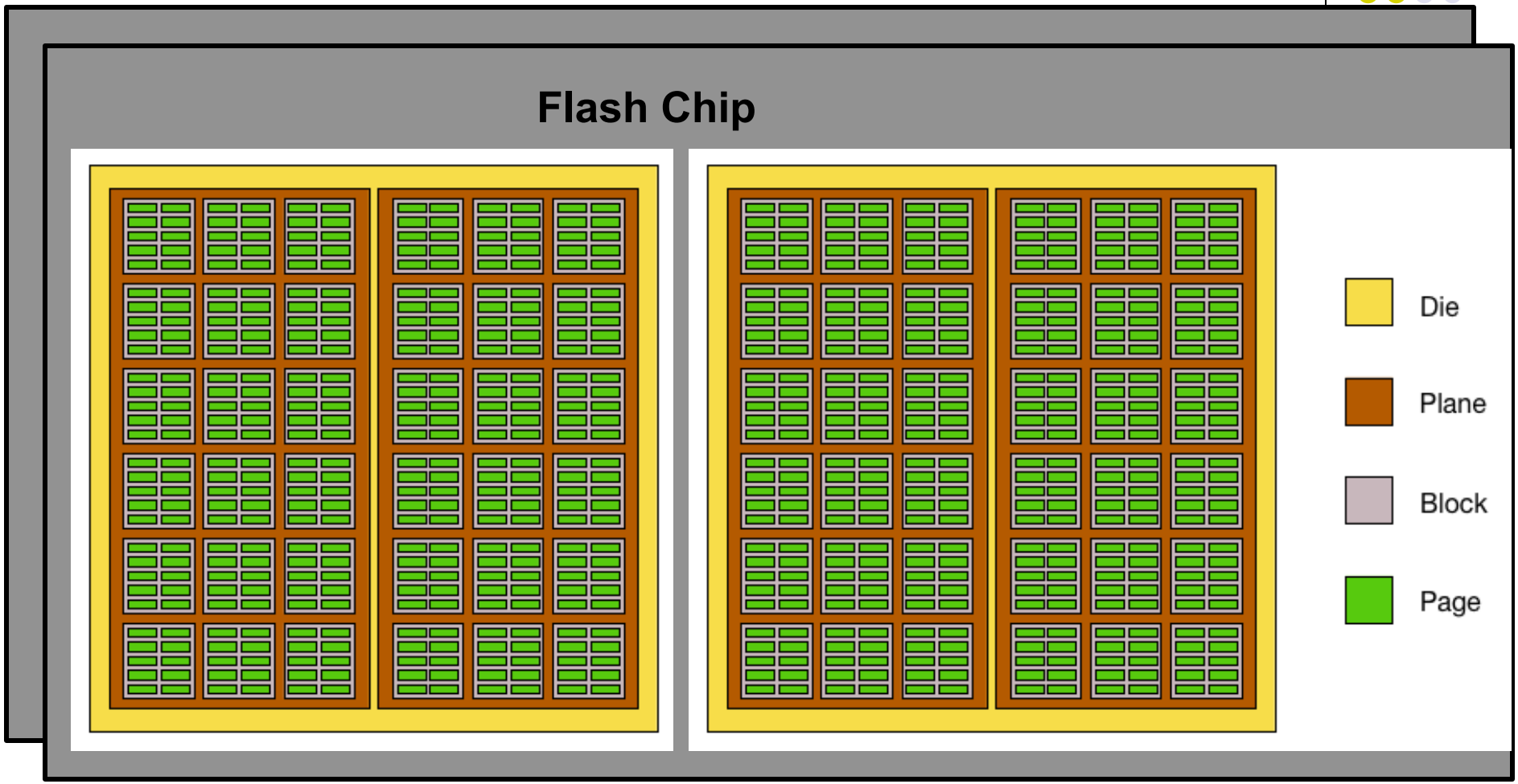


Source: iSuppli Q1 2011

Flash-based SSD

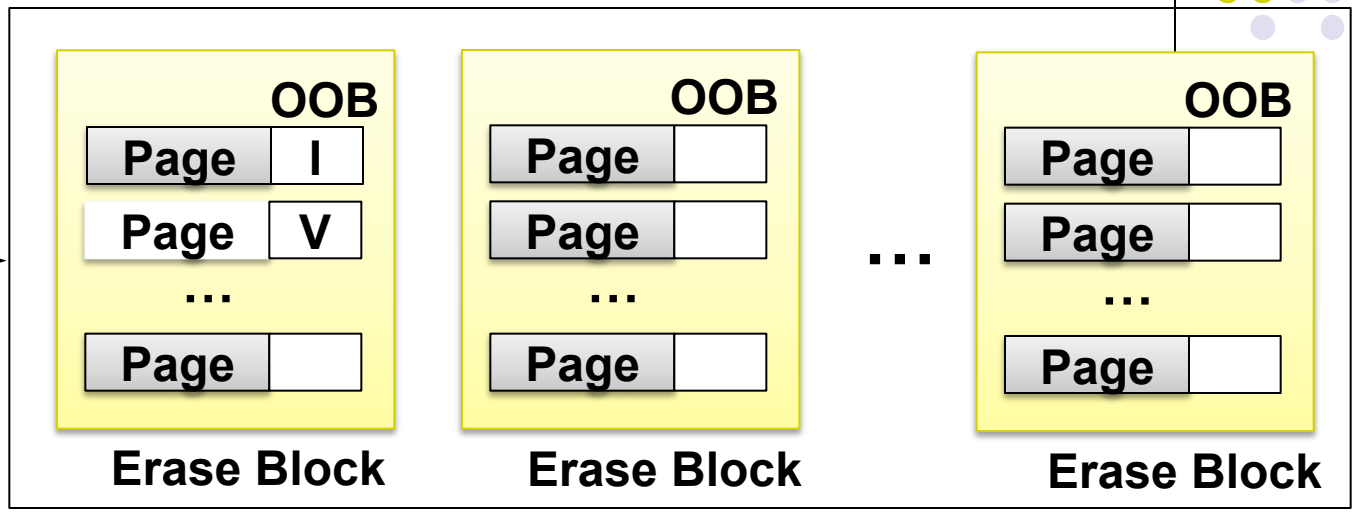


SSD Internal Organization



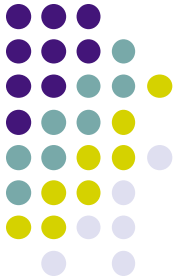
NAND Flash Memory

NAND Flash

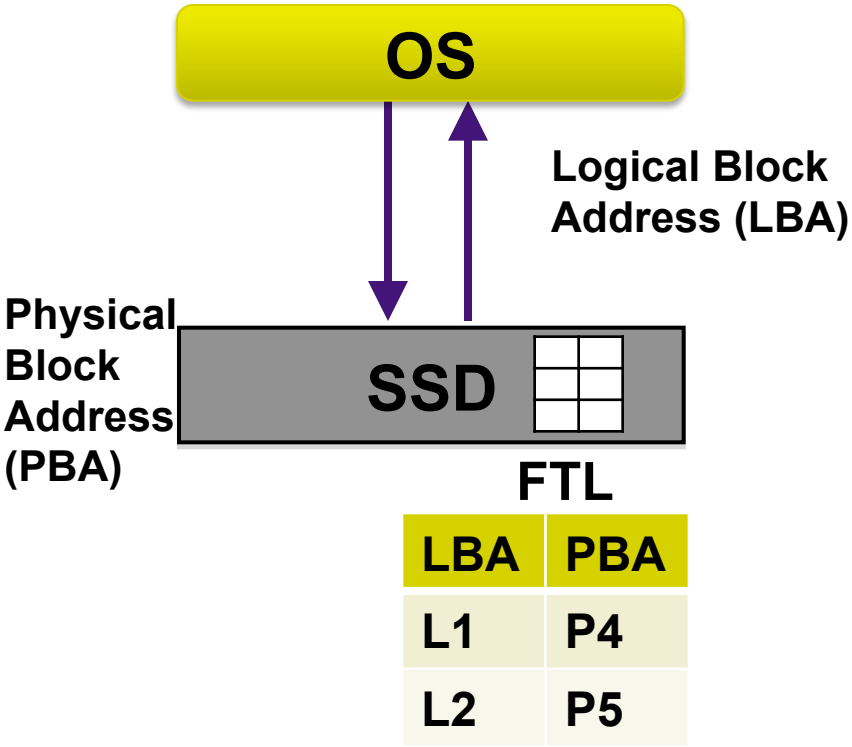


- Three operations: read, write, erase
  - Reads and writes at the granularity of page (e.g., 4KB)
  - Pages needs to be erased before writing (usually through garbage collection process)
  - Erases at the granularity of erase block (e.g., 256KB)
  - Flash wears out, wear leveling helps

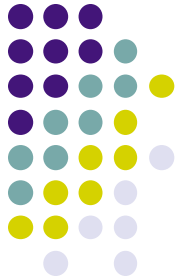
Flash Translation Layer (FTL)



- Map logical address to physical address to simulate traditional disk
  - Provides block I/O interface (like disks)
  - Hides erase-before-write
  - Hides GC and wear-leveling
- Mapping table in RAM in SSD
- FTL granularity
  - How many LBAs associated with an entry of FTL
- FTL granularity ↑
  - Smaller FTL table size
  - Larger update overhead

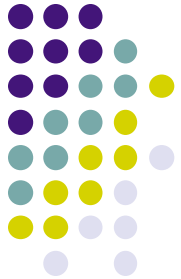


Garbage Collection



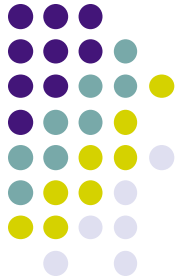
- Reclaims invalid pages
- Typically, called when free space falls below a threshold
- Victim block selection
  - Small # valid pages (reduce copying overhead)
  - Small # overall erases (wear level)

Wear Leveling



- Objective: keep all the flash memory space usable as long as possible
- Based on the number of erasures or writes performed on a block
  - < mean value : cold block
  - > mean value: hot block
  - Maintain the gap between hot and cold block as small as possible
  - Periodically swap data from hot blocks to cold blocks (costly)

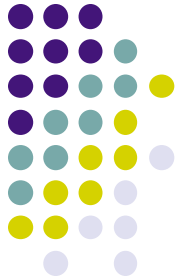
Design Issues of Flash-Based SSD



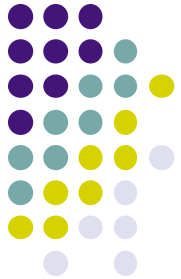
- FTL design
  - Reduce FTL's memory requirement
- Efficient garbage collection
  - Reduce garbage collection overhead
- Wear-leveling techniques
  - Prevent blocks from being unevenly worn so as to lengthen the overall lifespan

- Question: Are flash-based SSDs fundamentally different from hard disks?

Backup Slides



Block-Level FTL

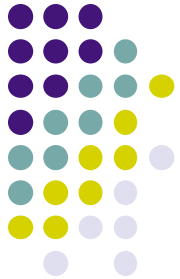


- Only blocks numbers in the mapping table
- Page offsets remain unchanged

- Pro: Small mapping table (memory footprint)
- 32GB flash memory, 2KB per page and 8 bytes/table entry, 64 pages/block:
  - 2MB page table

- Con: Very bad performance for write updates

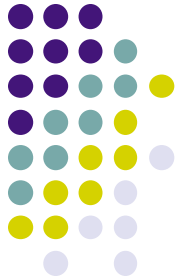
Page-Level FTL



- Each page mapped independently
- High flexibility

- Pro: Ideal performance
- Con: High RAM usage, infeasible
  - 32GB flash memory, 2KB per page and 8 bytes/table entry => 128MB table !!!
  - Worse when flash capacity increases or when restricted memory size/energy (e.g., mobile environment)

Hybrid FTL



- Page Mapping + Block Mapping
  - Most blocks are block mapped (BM)
  - Some blocks are page mapped (PM)

- Performance of PM with a memory footprint approaching BM