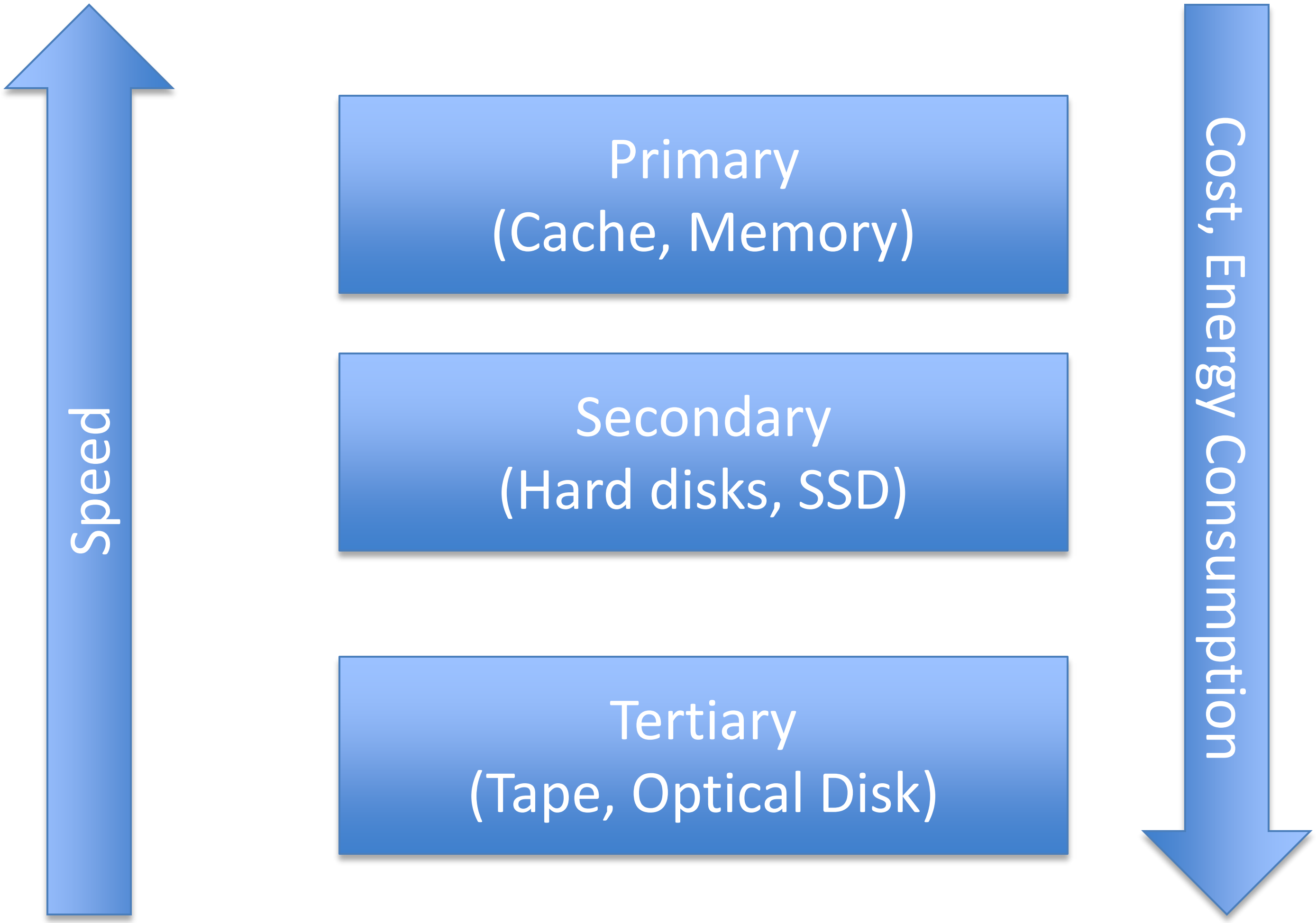
Storage Systems

INF 551

Wensheng Wu

Storage hierarchy



Storage operations: CRUD

- We should think about storage in terms of storage operations
- CRUD:
  - (C)reate
  - (R)Read
  - (U)pdate
  - (D)elete

Storage is not forever

- Transient vs. permanent Errors
- Media failures
  - One part of the storage media stops working
    - Cannot read from/write to physical location in media
- Device failures
  - Failure of entire device, data may not be recoverable
- Detecting failure
  - Periodic scanning of media
  - On-device monitoring
  - Inability to access device

Characterizing a storage device

- Capacity (bytes)
  - How much data it can hold
- Cost ($$$)
  - Price per byte of storage
- Bandwidth (bytes/sec)
  - Number of bytes that can be transferred per second
  - Note that read and write bandwidth may be different
- Latency (secs)
  - Time between initiating a request and an action
  - In the case of storage, action is to deliver 1st Byte

Time to complete an operation

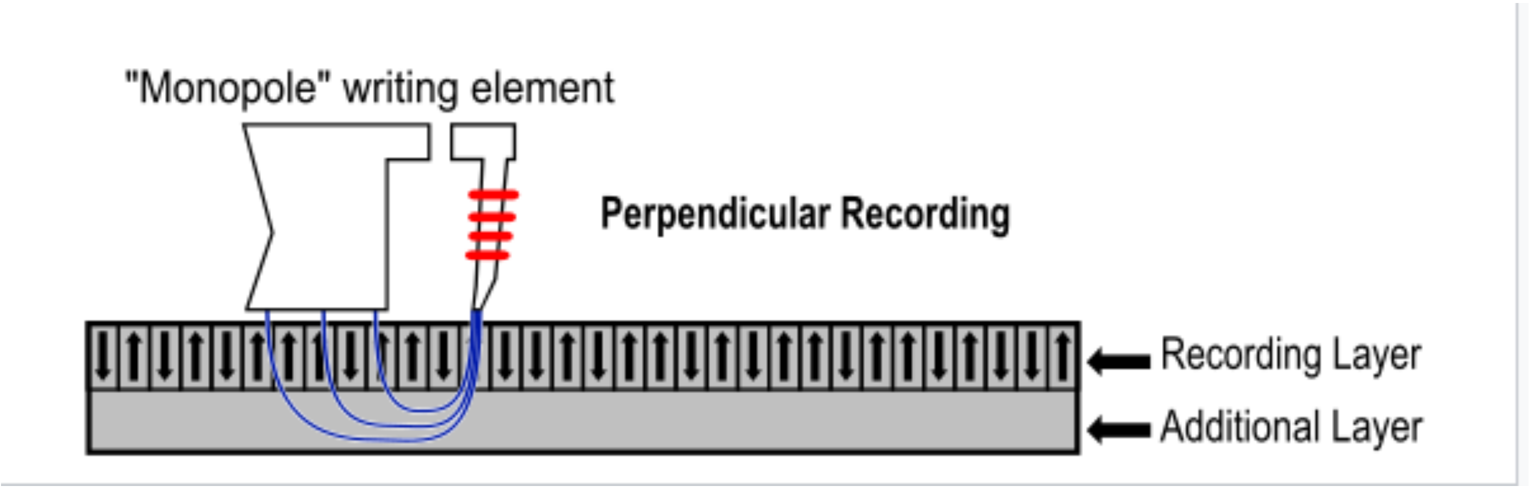
- Time to complete an operation depends on both bandwidth and latency
  - CompletionTime = Latency + Size/Bandwidth
- The time of a work load will depend on
  - Technology, e.g., hard drive/ssd
  - Operation type, e.g., read/write
  - Number of operations in the work load
  - Access pattern (random vs. sequential)

Access pattern

- Sequential
  - Data to be accessed are located next to each other on the device
- Random
  - Access data located randomly on storage device

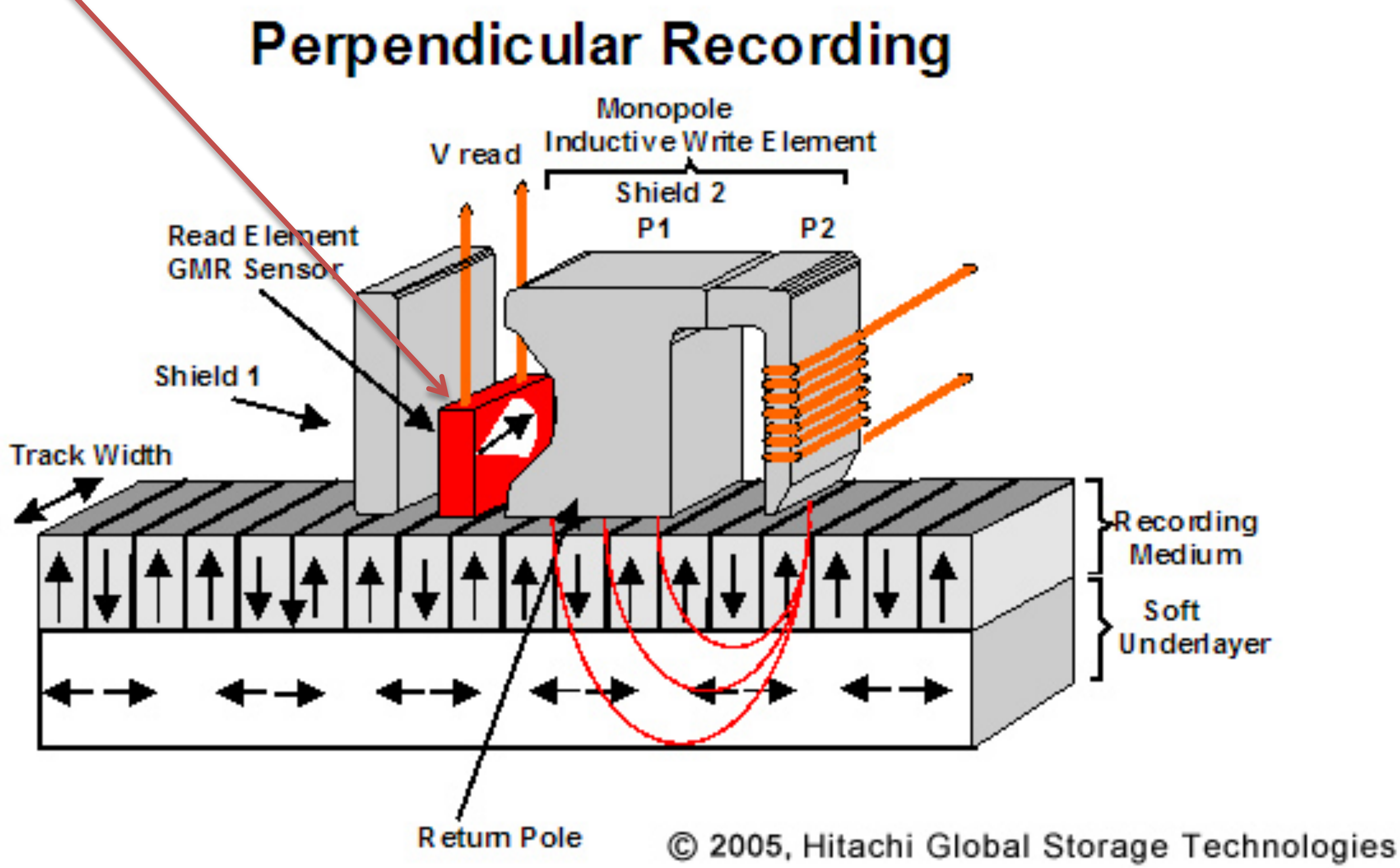
Magnetic recording

- Write head
  - Applies current to write head
  - Changes direction of magnetic field under head



Reading

- Read head senses direction of magnetic field

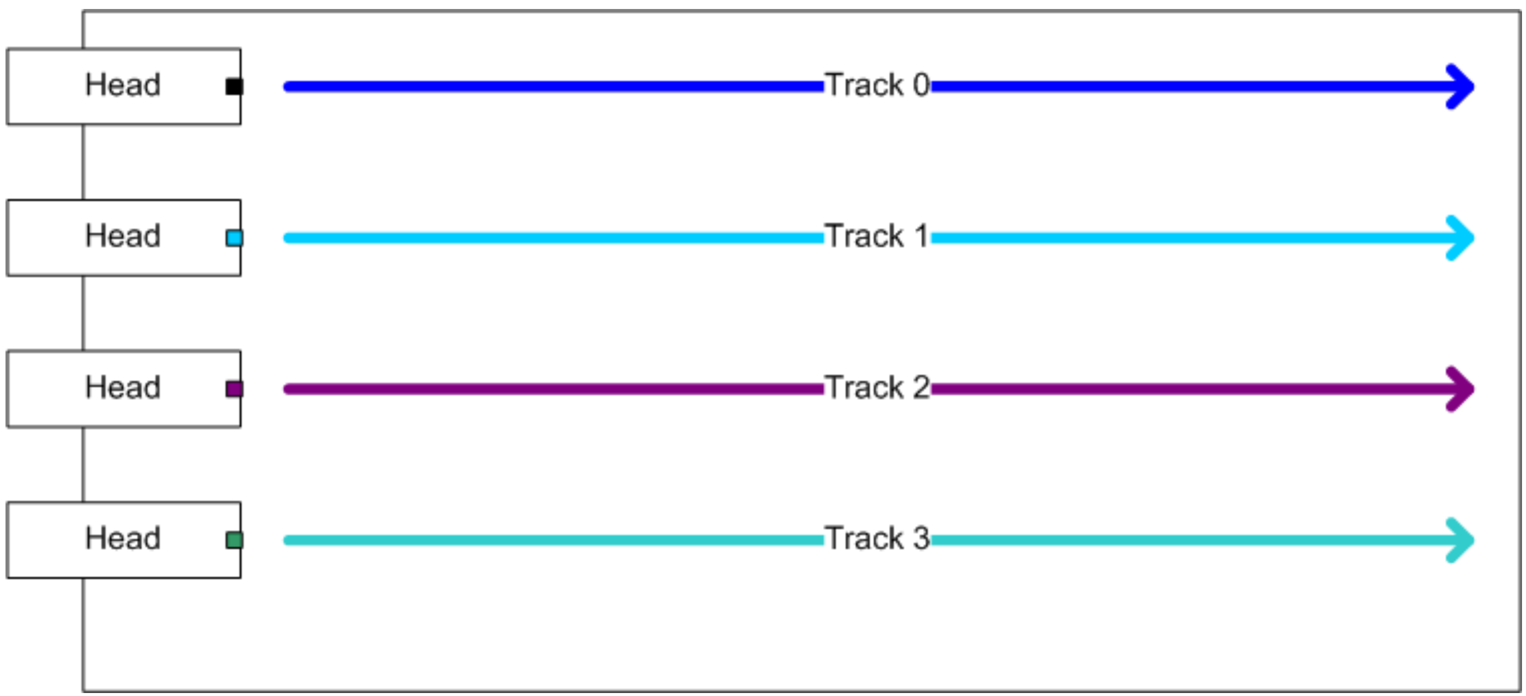


Road map

- Tapes
- Hard disk
- Disk scheduling algorithms
- Solid state drive

Linear tape

- Data recorded on parallel tracks that span the length of the tape



Tapes

- Current technology is LTO
  - Linear Tape-Open (an open standard)
- Characteristics
  - Capacity up to 6.25 TB per tape (LTO-7)
  - Drive cost ~ $2500
  - Tape cost ~ $45 for 2.5TB tape
- Tape access time (~ minute)
  - Time to mount the tape
  - Time to wind the tape to correct position
- Data transmission rates ~ 250MB/sec

$200
300MB/s
LTO-7

Performance characteristics

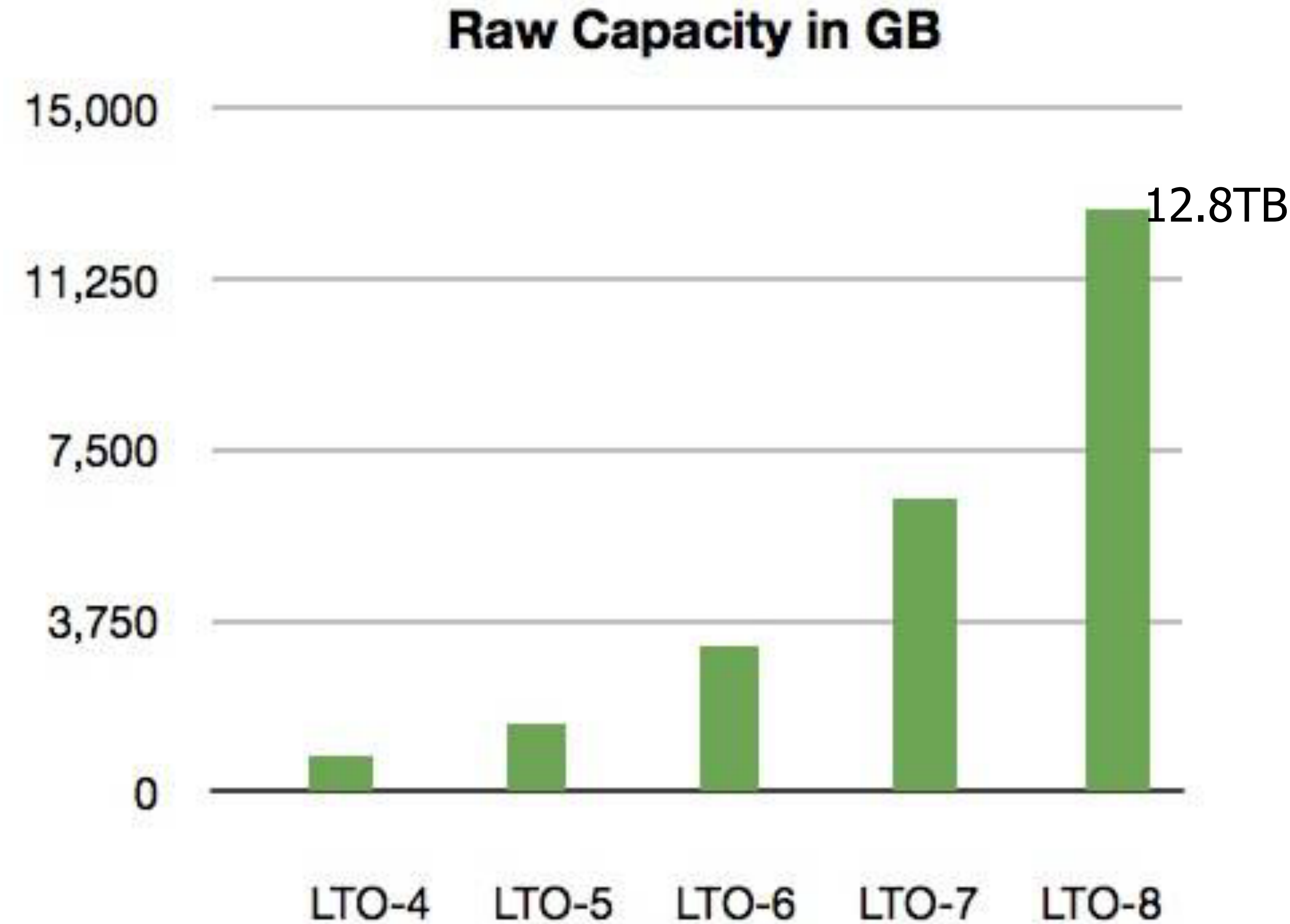
- High latency/low cost makes tape most appropriate for “archival” storage
  - Low frequency of reads
  - Very large data objects
- Random access will be slow due to latency
  - Sequential reads will be fast

Linear tape file system

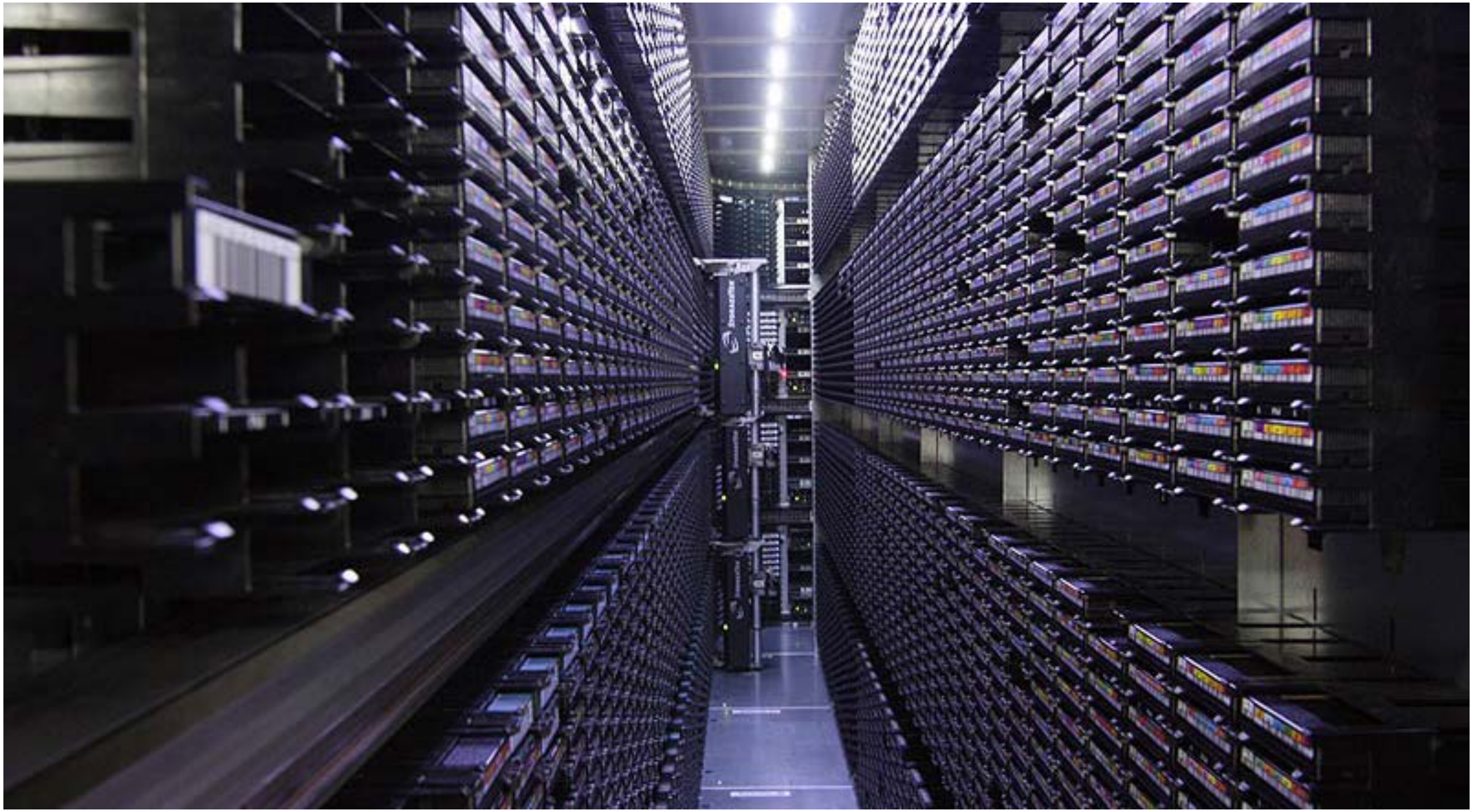
- Two partitions on tape
  - First contains metadata and directories. Tape reader can find and load this very quickly
  - Second contains blocks for data

- Directory structure coded in XML
  - Self describing file format…

Tape Cartridge


Raw Capacity in GB
15,000
11,250
7,500
3,750
0
12.8TB
LTO-4
LTO-5
LTO-6
LTO-7
LTO-8

A tape library



Inside a robotic tape library

- https://www.youtube.com/watch?v=nYfTtvpQ778

Road map

- Tapes
- Hard disk
- Disk scheduling algorithms
- Solid state drive

Hard disk drives

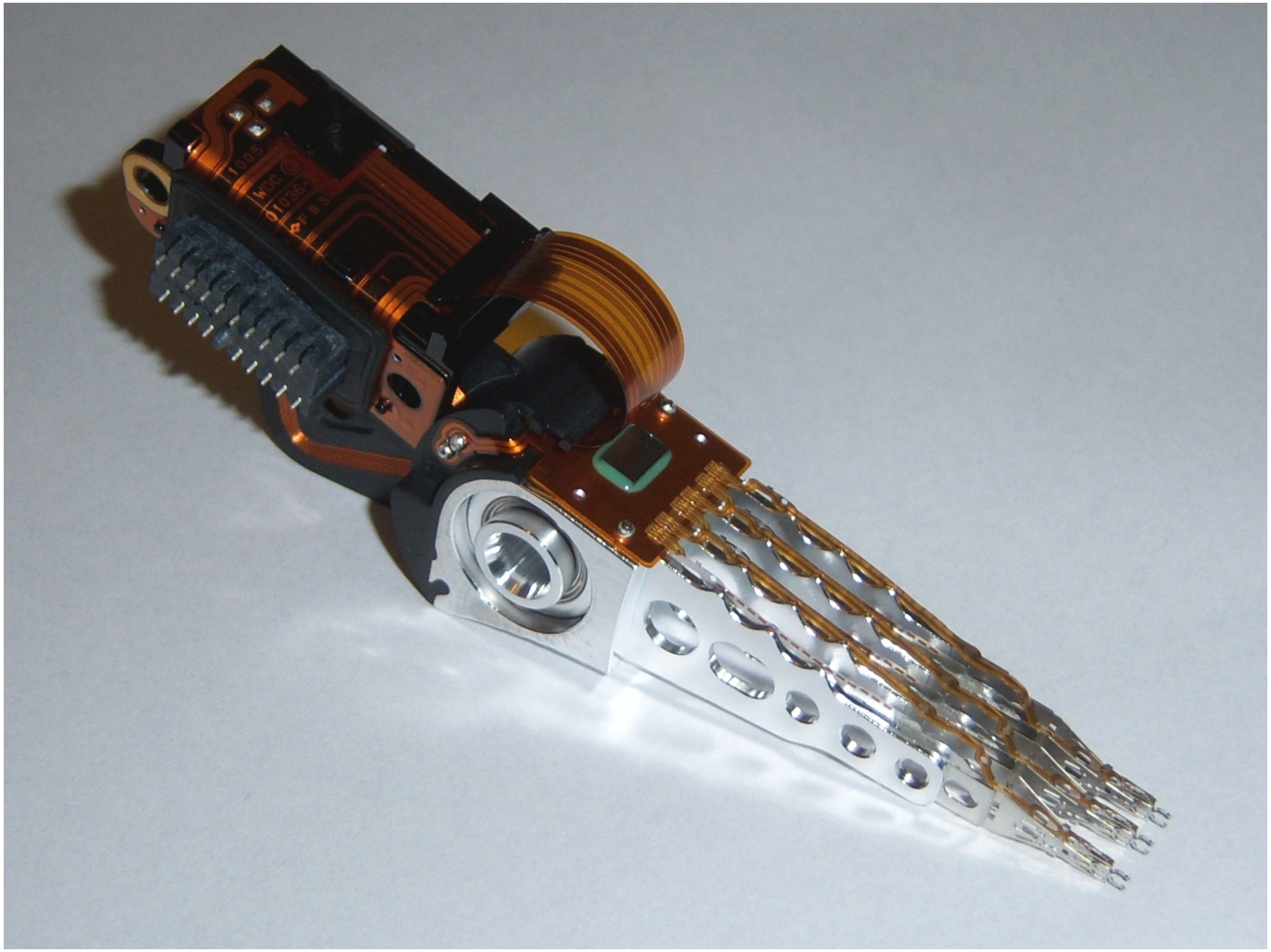
- Perhaps the most pervasive form of storage
- Basic Idea:
  - One or more spinning magnetic platters
    - Typically two surfaces per platter
  - Disk arm positions over the radial position (tracks) where data is stored
    - It swings across tracks (but do not extend/shrink)
  - Data is read/written by a read/write head as platter spins


WD
Western Digital
www.westerndigital.com
3 TB
SATA / 64MB Cache
WD30EZRS
MDL : WD30EZRS
Product warranty will be void if seal, label or cover is removed or damaged.
WWN:
DATE:
DCM :
DCX :
Do not cover any drive holes.
LBA :
5VDC :
12VDC :
R/N :
Product of Malaysia
SERIAL ATA
RoHS
FRAGILE
WD Caviar®Green™

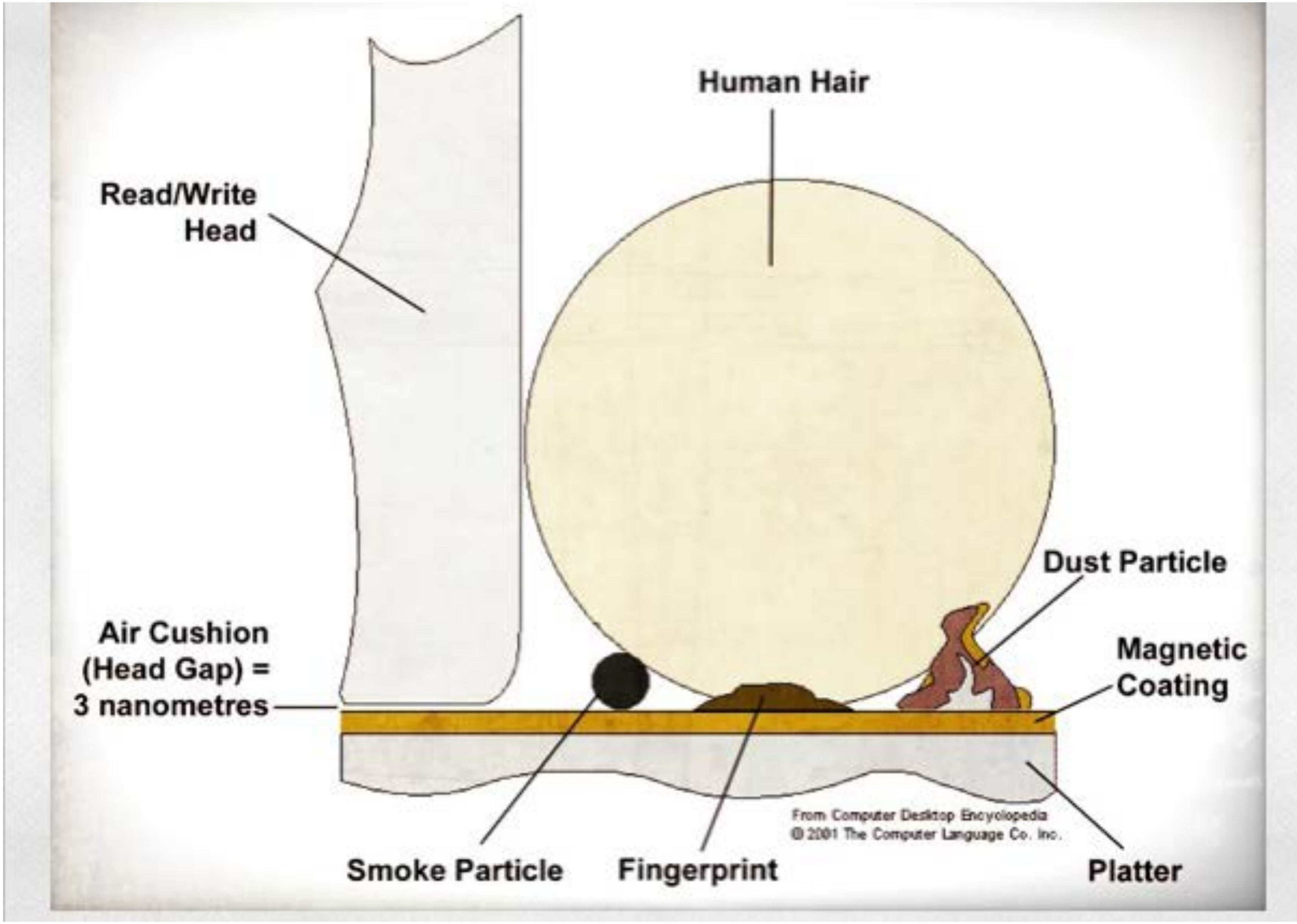
Internal of hard disk

Disk arm and platter

Disk head close-ups



Disk head close-ups


Human Hair
Read/Write
Head
Dust Particle
Air Cushion
(Head Gap) =
3 nanometres
Magnetic
Coating
From Computer Desktop Encyclopedia
© 2001 The Computer Language Co. Inc.
Smoke Particle
Fingerprint
Platter

Disk head movement

- Hard disk head movement while copying files between two folders (e.g., partition c to d)
  - https://www.youtube.com/watch?v=BlB49F6ExkQ

2GB Storage in 1980s ($250,000)

Physical characteristics

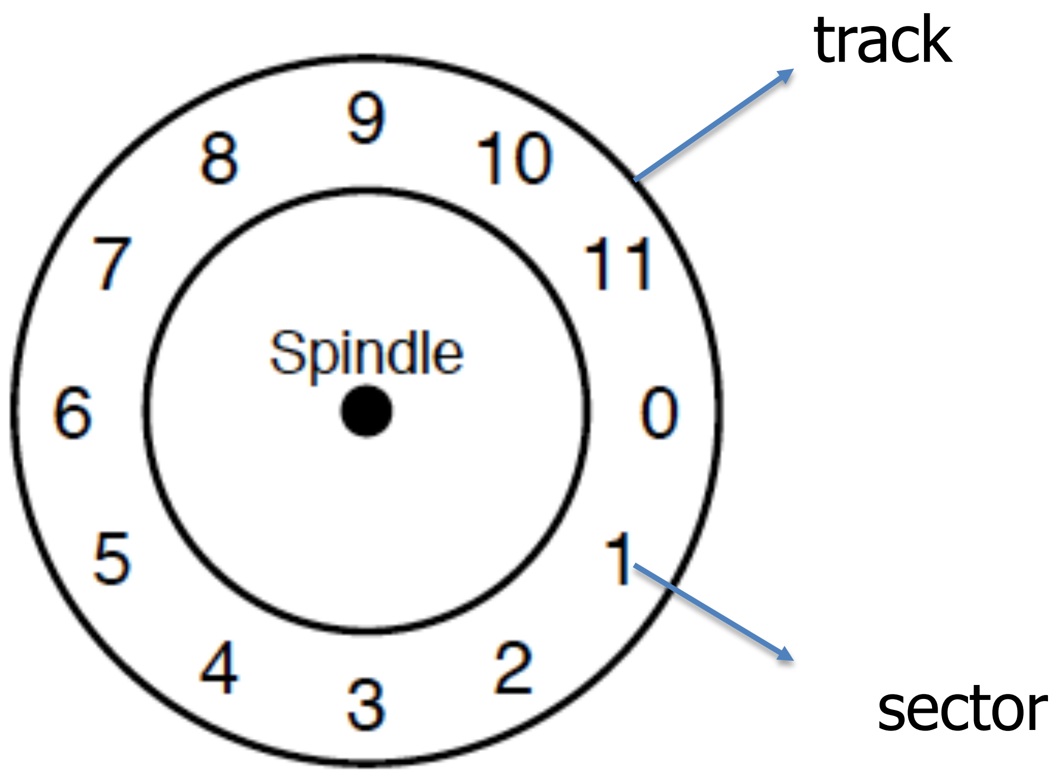
- 3.5" (diameter, common in desktops)
- 2.5" (common in laptops)
- Rotational speed
  - 5,400 RPM
  - 7,200 RPM
  - 4,800 RPM
  - 10,000 RPM (6ms/rotation)
- Between 5-7 platters
- Current capacity up to 10TB (Western Digital)

Disk organization

- Disk has multiple tracks

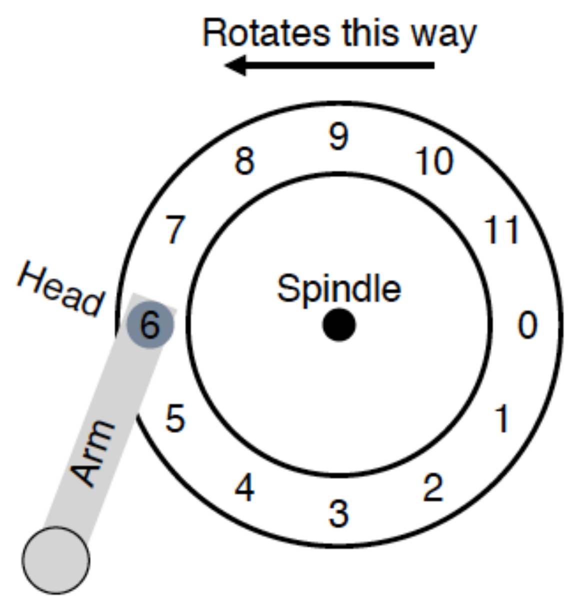
- Each track is divided into N fixed size sectors
  – Typical sector size is 512 bytes (old) or 4KB (new)
  – Sectors can be numbered from 0 to N-1
  – Entire sector is written “atomically”
    - All or nothing

A simple disk drive (one track only)



Rotational latency

- Waiting for the right sector to rotate under the head
  - On average: ½ of time for a full rotation
  - Worst case?
  - Best case?

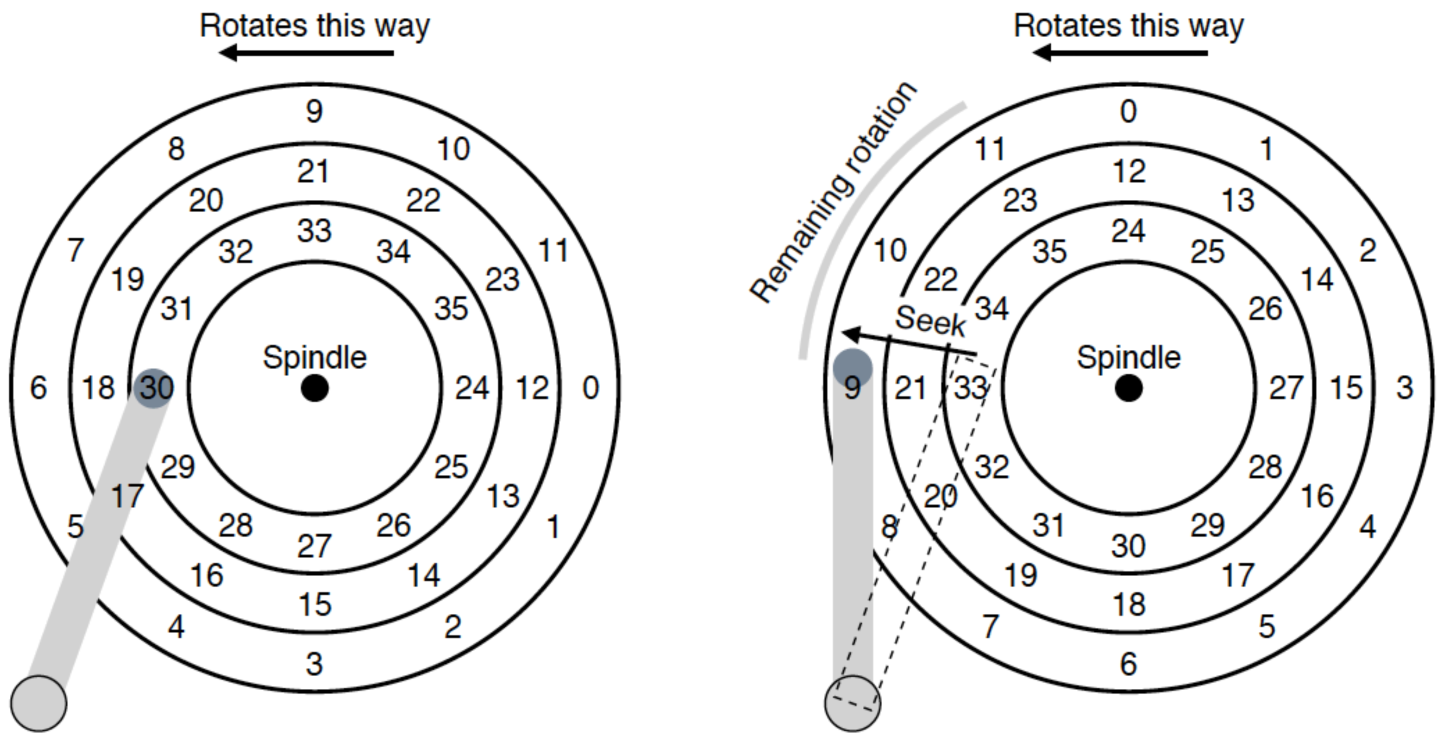


Rotation time

- Assume 10,000 RPM (rotations per minute)

$$\frac{Time\ (ms)}{1\ rotation} = \frac{60000\ ms}{1000\ rotation} = \frac{6\ ms}{rotation}$$

Multiple tracks: add seek times



Average seek time is about 1/3 max seek time

Transmission time

- Assume that we transfer 512KB
- Assume 128 MB/sec transmission bandwidth
  - 512KB/128MB * 1000ms

    = 4ms

Completion time

- $T = T_{seek} + T_{rotation} + T_{transfer}$
  - T_{seek} : Time to get the disk head on right track
  - $T_{rotation}$:Time to wait for the right sector to rotate under the head
  - $T_{transfer}$: Time to actually transfer the data

Example

- Capacity 4TB
- # platters: 4
- # heads: 8
- Bytes per sector: 4096

- Transmission bandwidth: 100MB/sec
- Maximum seek time: 12ms

- RPM: 10,000

Time to transfer a file

- The file occupies 100 sectors (sequentially)
- Avg. seek time =?
- Avg. rotational latency =?
- Transfer time = ?

Sector vs. block

- Block = 1 or more sectors

- Disk typically transfers one block at a time
  - Why?

- We will assume one block = one sector
  - Unless stated otherwise

Sequential operations

- Assume all sectors involved are on same track
  - We may need to seek to the right track
  - And rotate to the first sector
- But no rotation/seeking is needed afterward

Sequential vs. random

- Consider disk with 7ms avg seek, 10,000 RPM platter speed and 50 MB/sec transfer rate, 4KB/block
- Sequential access of 10 MB
  - Completion time = 7 + 3 + 10/50*1000 = 210ms
  - Actual bandwidth = 10MB/210ms = 47.62 MB/s
- Random access of 10 MB (2,500 blocks)
  - Completion time = 2500 * (7 + 3 + 4/50) = 25.2s
  - Actual bandwidth = 10MB / 25.2s = .397 MB/s

Road map

- Tapes
- Hard disk
- Disk scheduling algorithms
- Solid state drive

Scheduling Problem

- Multiple requests in a queue
- Schedule the requests
  - So that they are served as quickly as possible
  - E.g., measured by avg. # of tracks/cylinders travelled per request
- Handled by either OS or disk controller or both

Example

- Queue: 120, 30, 60, 135, 80, 20, 95, 160, 70, 5
- Head on 50

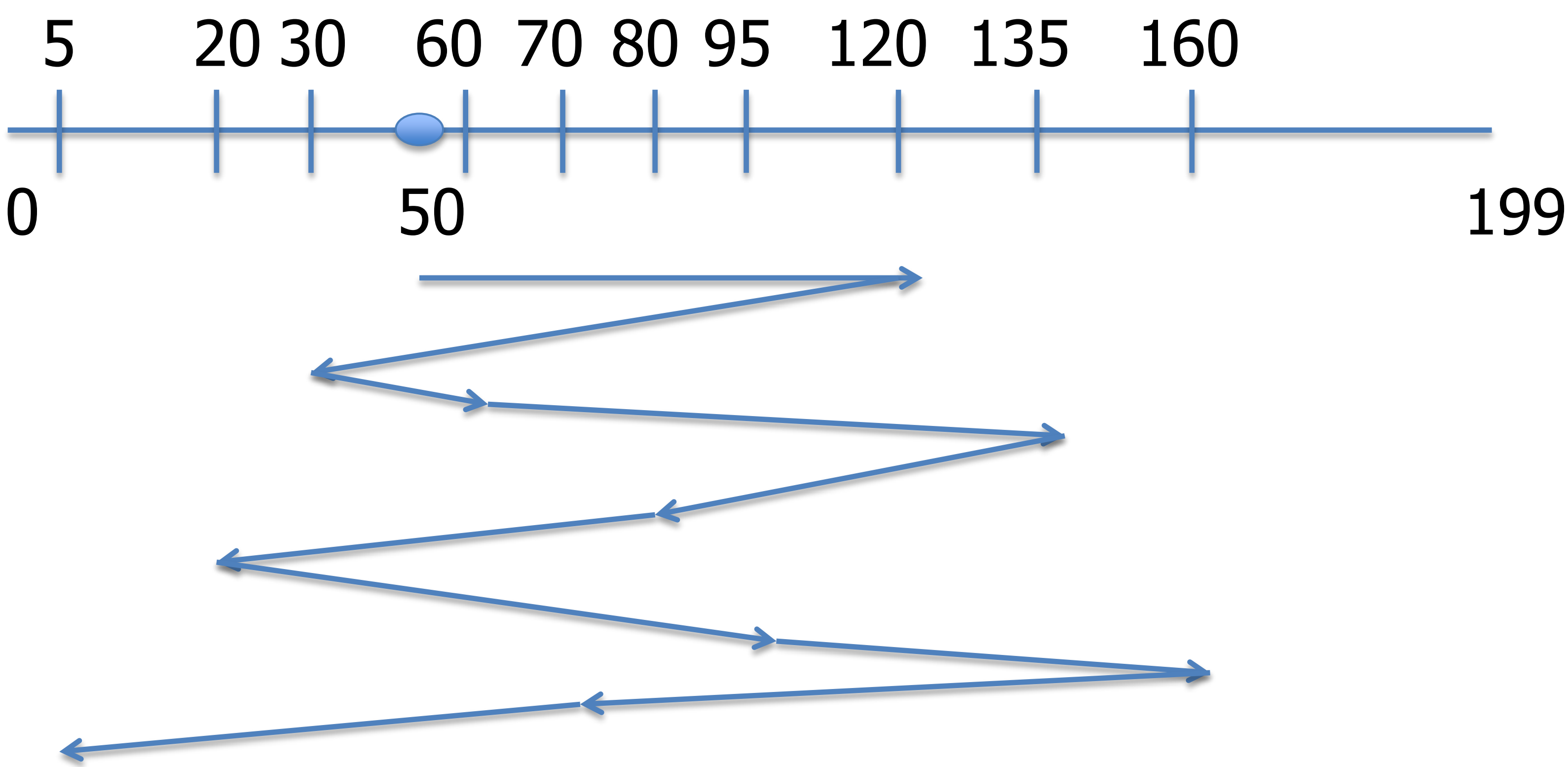
- Innermost track: 0
- Outermost track: 199

Scheduling Algorithms

- FCFS: first-come first-served
- SSTF: shortest seek time first
- SCAN (Elevator), C-SCAN (circular SCAN)
- LOOK, C-LOOK (circular LOOK)

FCFS (first-come first-served)

- Queue: 120, 30, 60, 135, 80, 20, 95, 160, 70, 5



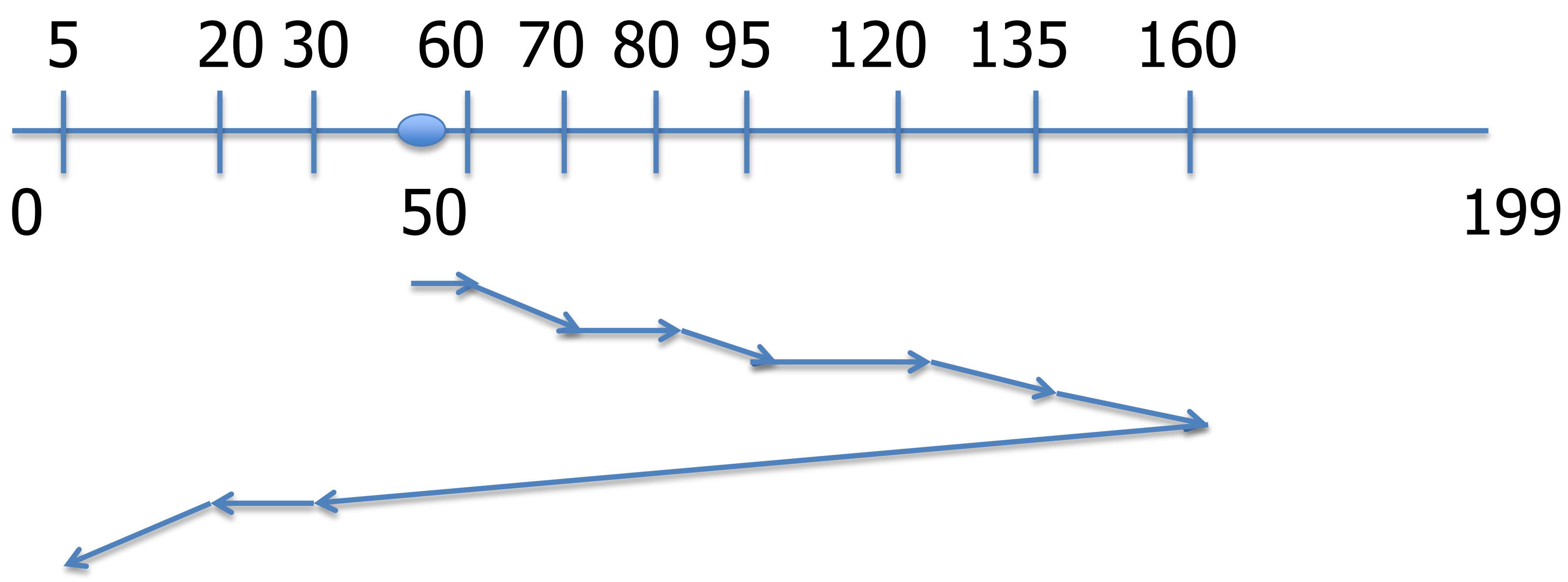
FCFS

- # of tracks travelled: 675 (avg 67.5/request)
  - 50 -> 120 = 70
  - 120 -> 30 = 90
  - …

- Problems:
  - A lot of head movements
  - Going back and forth

SSTF (shortest seek time first)

- # of tracks travelled: 160-50 + 160-5 = 265
- Is this solution optimal?



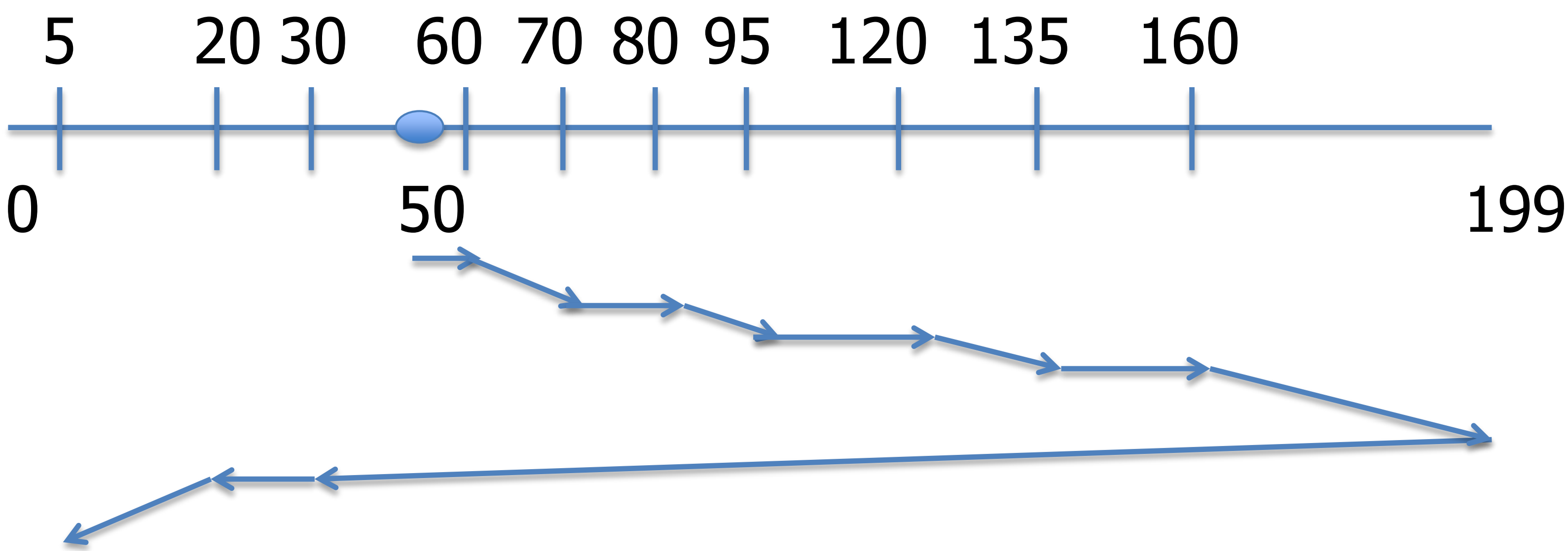
SSTF problems

- Starvation possible unless queue is frozen

- E.g., 160 was just served, but more requests near 160, say 150 or 170, arrive
  - Requests 5, 20, 30 may need to wait for long time

SCAN (Elevator)

- Seek to closest track (assume idle now)
  - continue to the end
  - then reverse the direction
- # of tracks travelled: 199-50 + 199-5 = 343



SCAN problems

- Travel right back to the area just served
- Middle area has more chances to be served
  - Worst case for request on edge: ~ 400 tracks
  - Worst case for middle area: ~200 tracks

SCAN problems

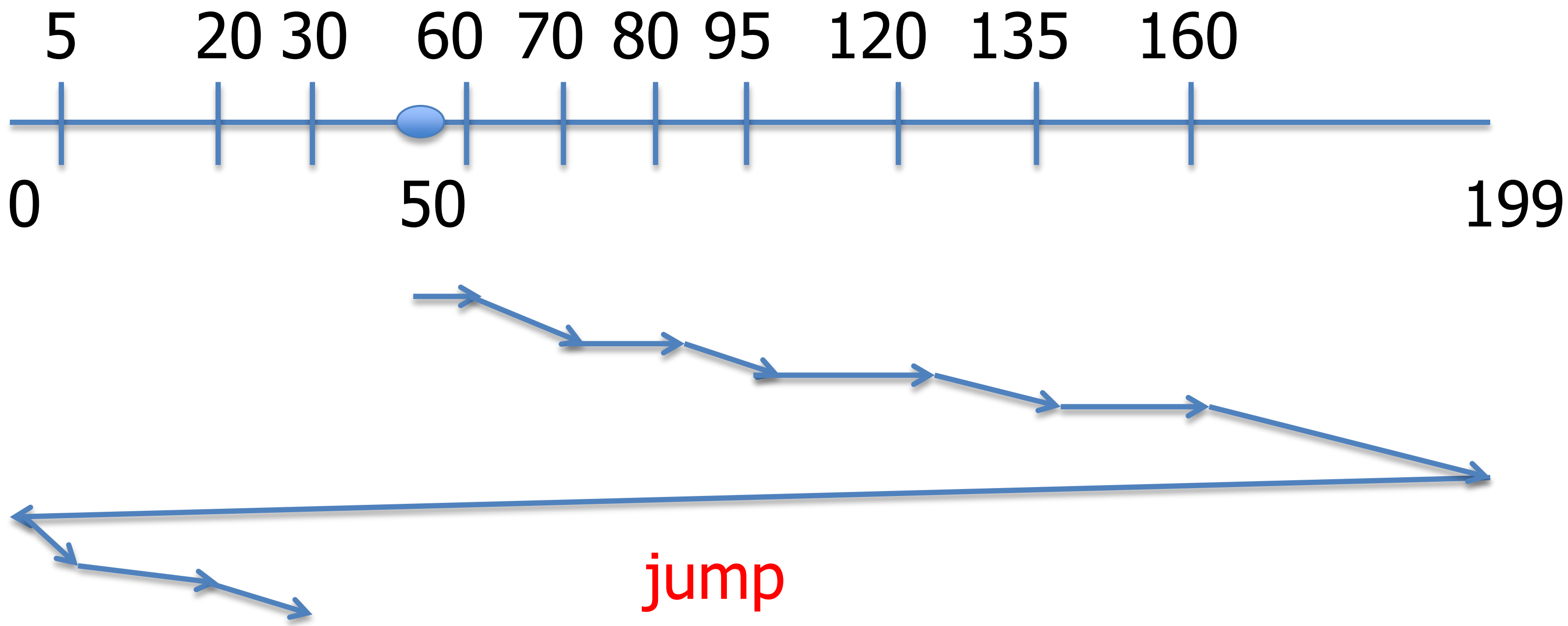
- Starvation still possible if queue is not frozen
  - E.g., right after serving 120, more requests on the same track come

C-SCAN

- Scan in only one direction:
  - from innermost (track 0) to outermost track (199)
- Jump back to innermost when reaching end
  - 0 → 199
  - Jump to 0
  - 0 → 199
  - ….

Example

- # of tracks travelled : 199-50 + 199 + 30 = 378

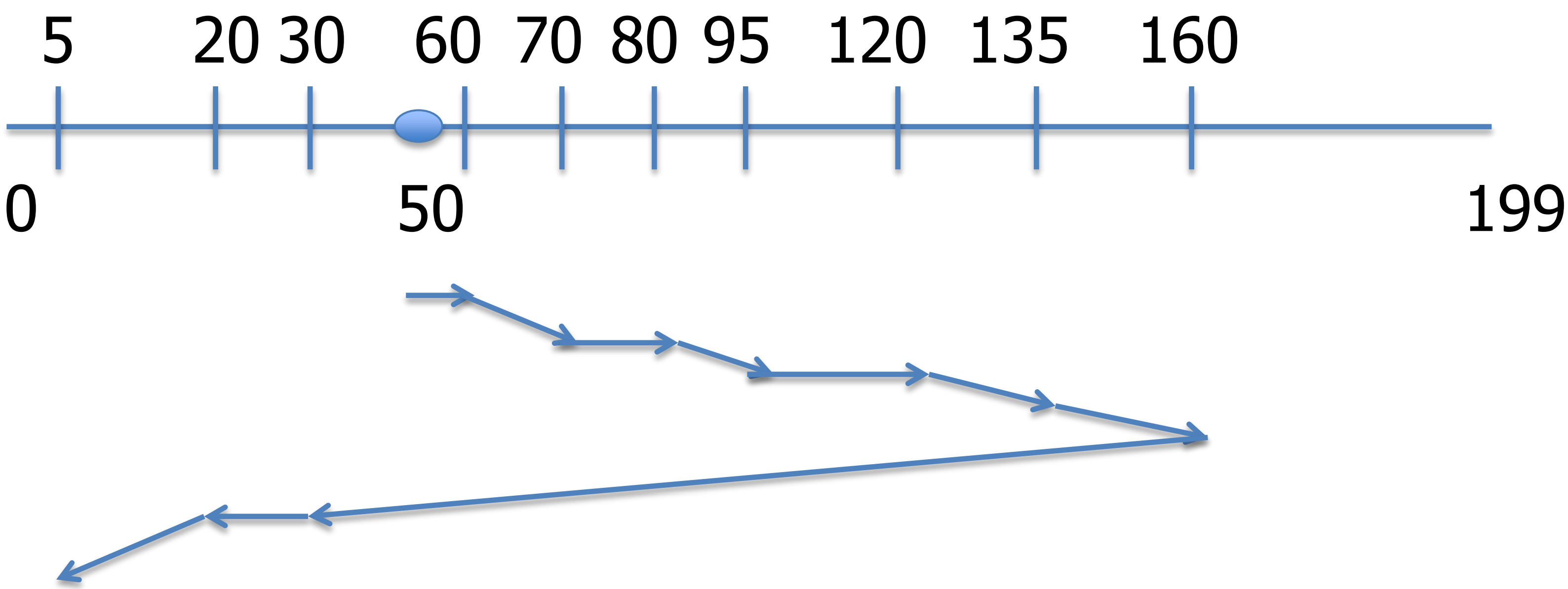


Observations

- More uniform treatments on all tracks
  – Worst case for request on edge: 200 tracks + jump time
- But time is wasted on jump
  – although jumping is faster than moving one track at a time
  – Only one acceleration and one deceleration

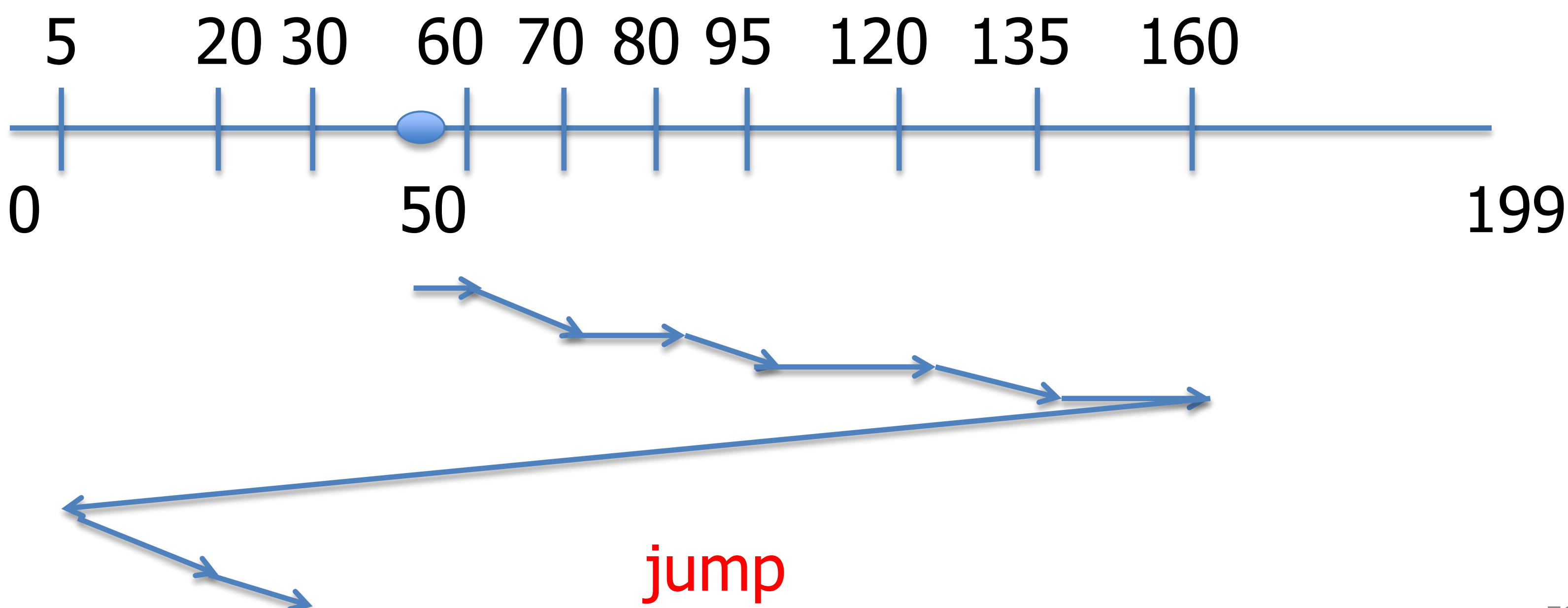
LOOK

- Similar to SCAN, but scan to last request only
- # of tracks travelled : 265
  - Behave the same as SSTF in this example



C-LOOK

- Similar to C-SCAN, but scan to last request
  - Jump to the last request (the opposite side) too
- # tracks travelled: 160-50 + 160-5 + 30-5 = 290



Comparisons

- FCFS: 675
- SSTF: 265
- SCAN: 343
- C-SCAN: 378
- LOOK: 265
- C-LOOK: 290

Comparisons

- Lightly-loaded system
  – SSTF and LOOK are good choices

- Heavily-loaded system
  – SCAN and C-SCAN are better
  – SSTF more likely now to suffer from starvation
  – Saving from LOOK likely less significant now

Road map

- Tapes
- Hard disk
- Disk scheduling algorithms
- Solid state drive

Solid State Drive

Chips

Solid State Drives

- All electronic, made from flash memory
- Lower energy consumption than hard drive
- Significantly more expensive, less capacity
  – About a factor of 10 more expensive
- Limited lifetime, can only write a limited number of times.
  – E.g., 100, 000 write cycles for SLC (single-level cell) memory

Solid State Drives

- Same form-factor and control interface as magnetic disks
- Significantly better latency
  - No seek or rotational delay
- Consistent bandwidth for sequential & random:
  - Benefits from improved latency
  - However, writes take significantly longer then reads

Writing to SSD is complicated

- Can not overwrite a page
  - Need to erase its block (at certain point) instead

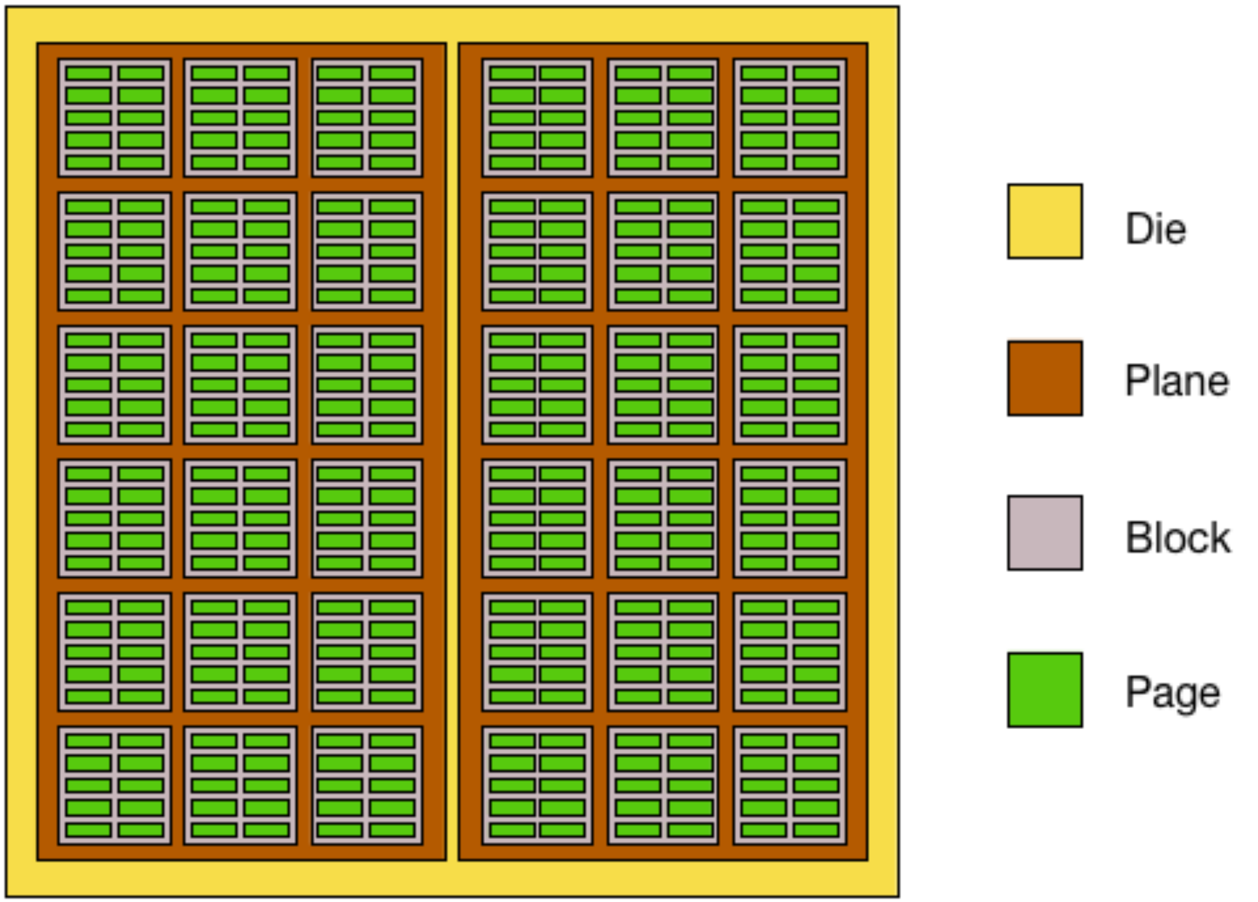
- SSD controllers take care of all these details

SSD

- Contains a number of flash memory chips
  - Chip -> dies -> planes -> blocks -> pages (rows) -> cells
  - Cells are made of floating-gate transistors

- Page is the smallest unit of data transfer between SSD and main memory
  - Much like a block in hard disk

Die Layout



[Understanding Flash: Blocks, Pages and Program / Erases]

Dies, planes, block, and pages

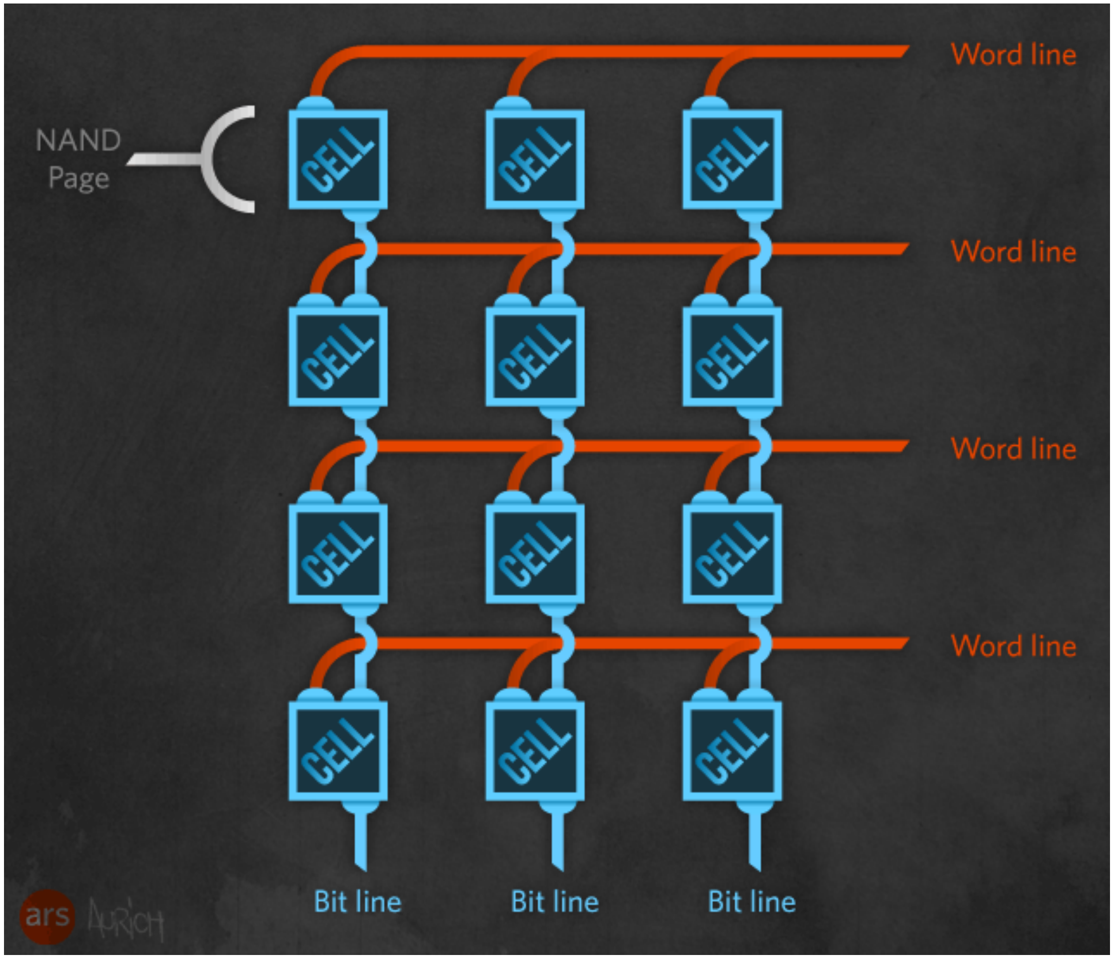
- Typically, a chip may have 1, 2, or 4 dies
- A die may have 1 or 2 planes
- A plane has a number of blocks
  - Block is the smallest unit that can be erased
- A block has a number of pages
  - Page is the smallest unit that can be programmed/written to

Typical page and block sizes

- Common page sizes: 2K, 4K, 8K, and 16K
- A block typically has 128 to 256 pages

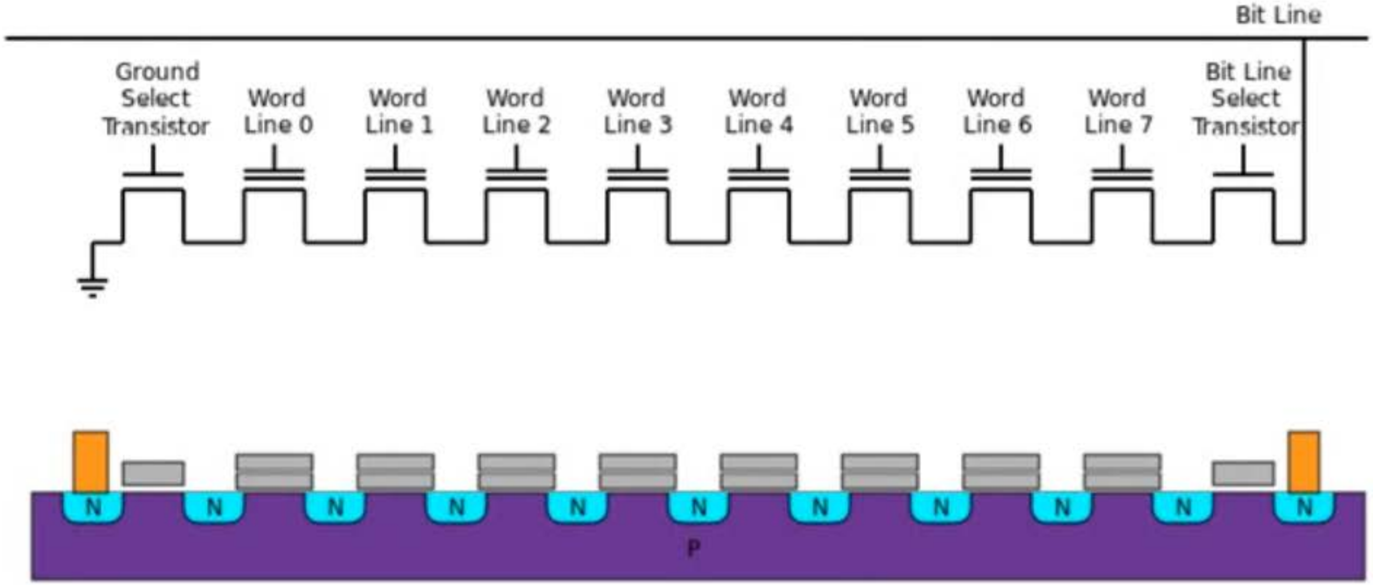
=> Block size: 256KB to 4MB

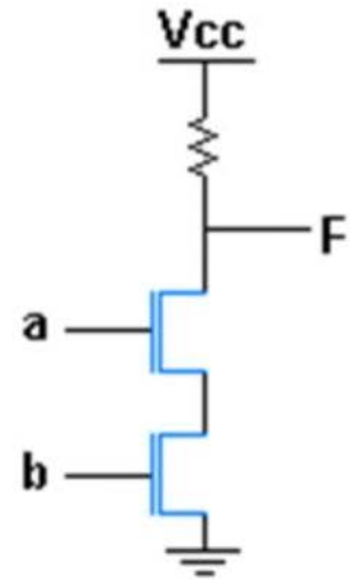
NAND flash


Word line
NAND
Page
CELL
Word line
Word line
Word line
Bit line
Bit line
Bit line
ars

NAND flash layout

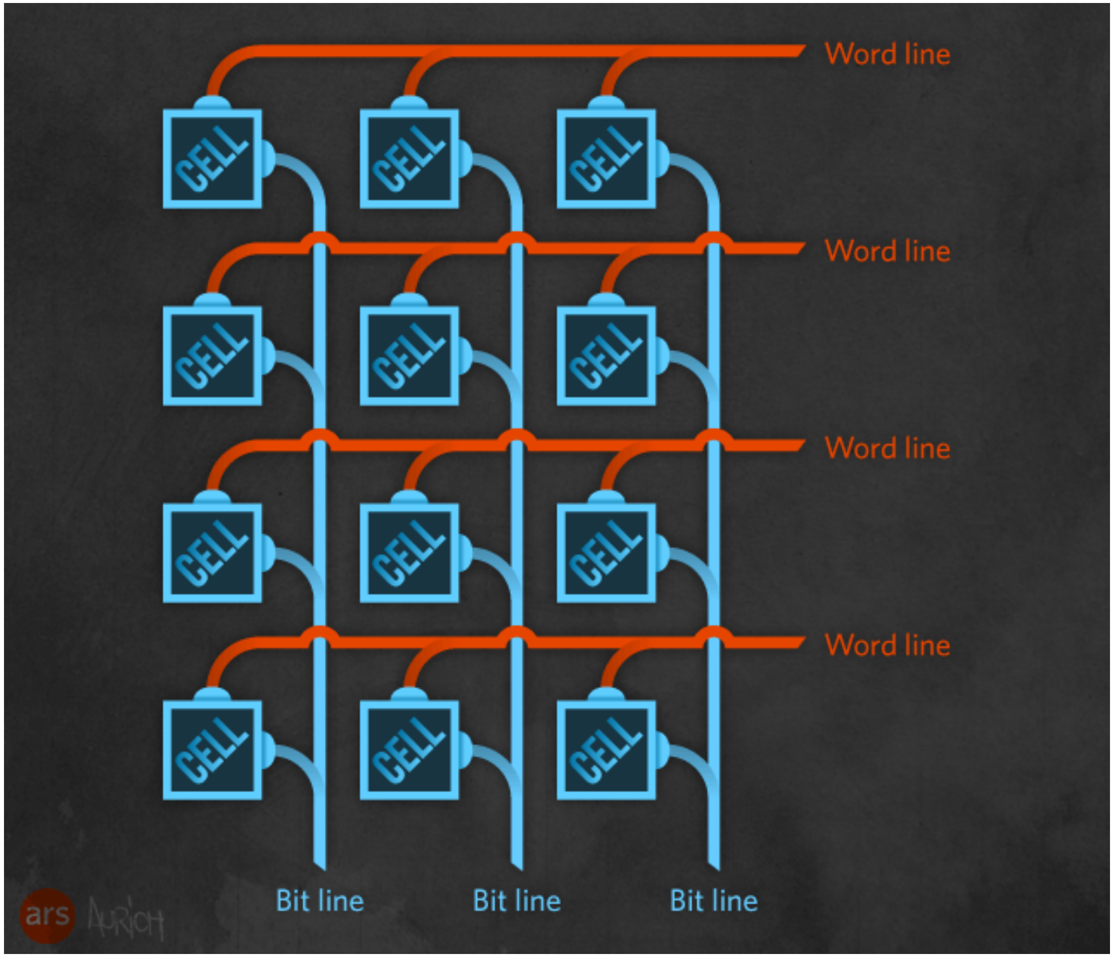
- Transistors are strung together in a series
  - Similar to the transistors in an NAND gate





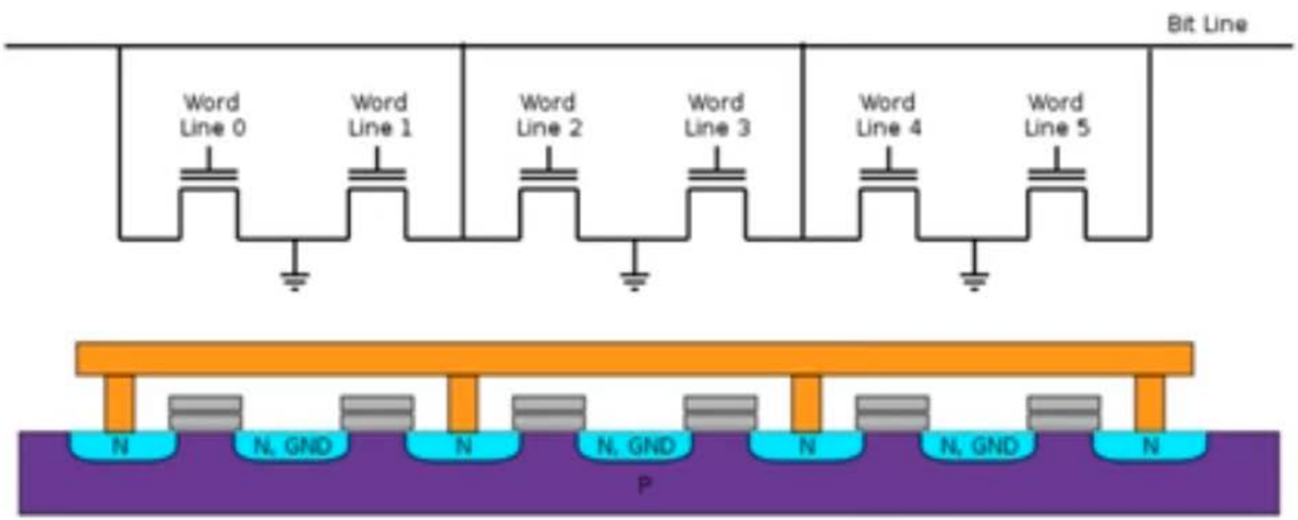
NMOS NAND gate

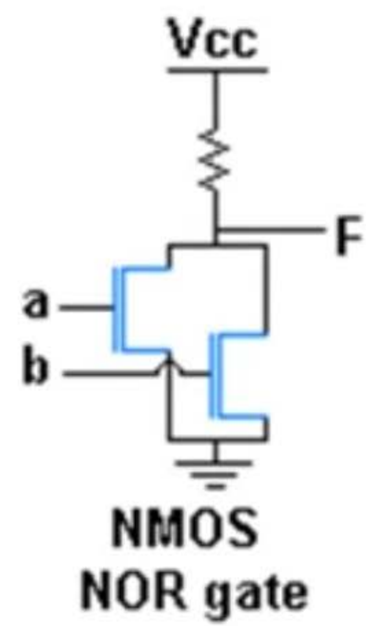
NOR flash layout



NOR flash layout

- Floating gate transistors are wired in parallel
  - Each is directly connected to bit line (also ground)
  - Similar to transistors in a NOR gate (to output F)





NAND vs NOR

- Bit line
  - NOR has individual bit line (for cell), so circuit more complex
  - NAND ties up all bit lines, save space, to allow larger capacity

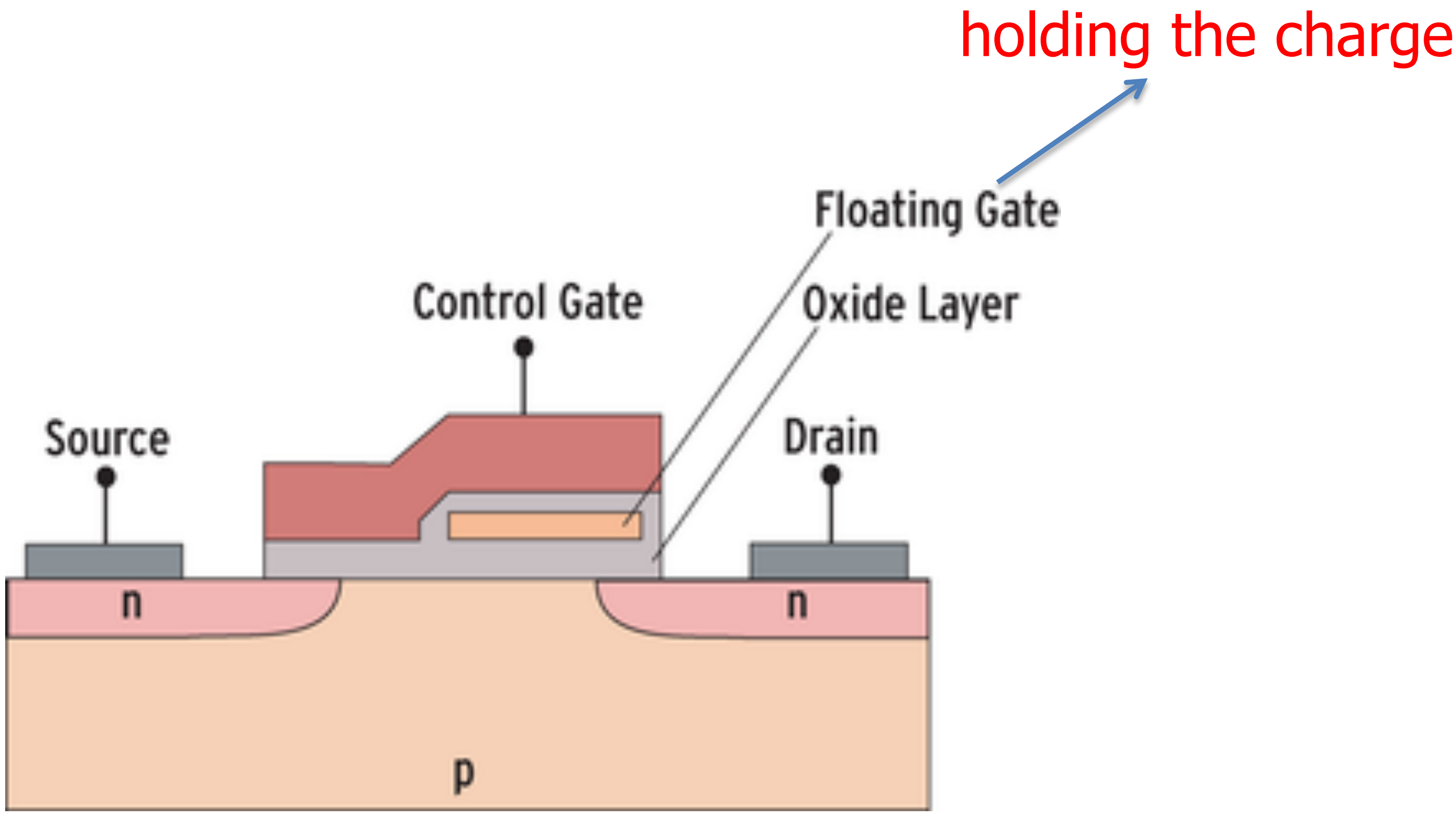
- Default value of cell
  - NOR: 0
  - NAND: 1

Write vs. erase

- Page is the smallest unit that can be read or written (also called programmed)
- Block is the smallest unit that can be erased
  - i.e., make cells "empty" (storing default values)

Operation	Area
Read	Page
Program (Write)	Page
Erase	Block

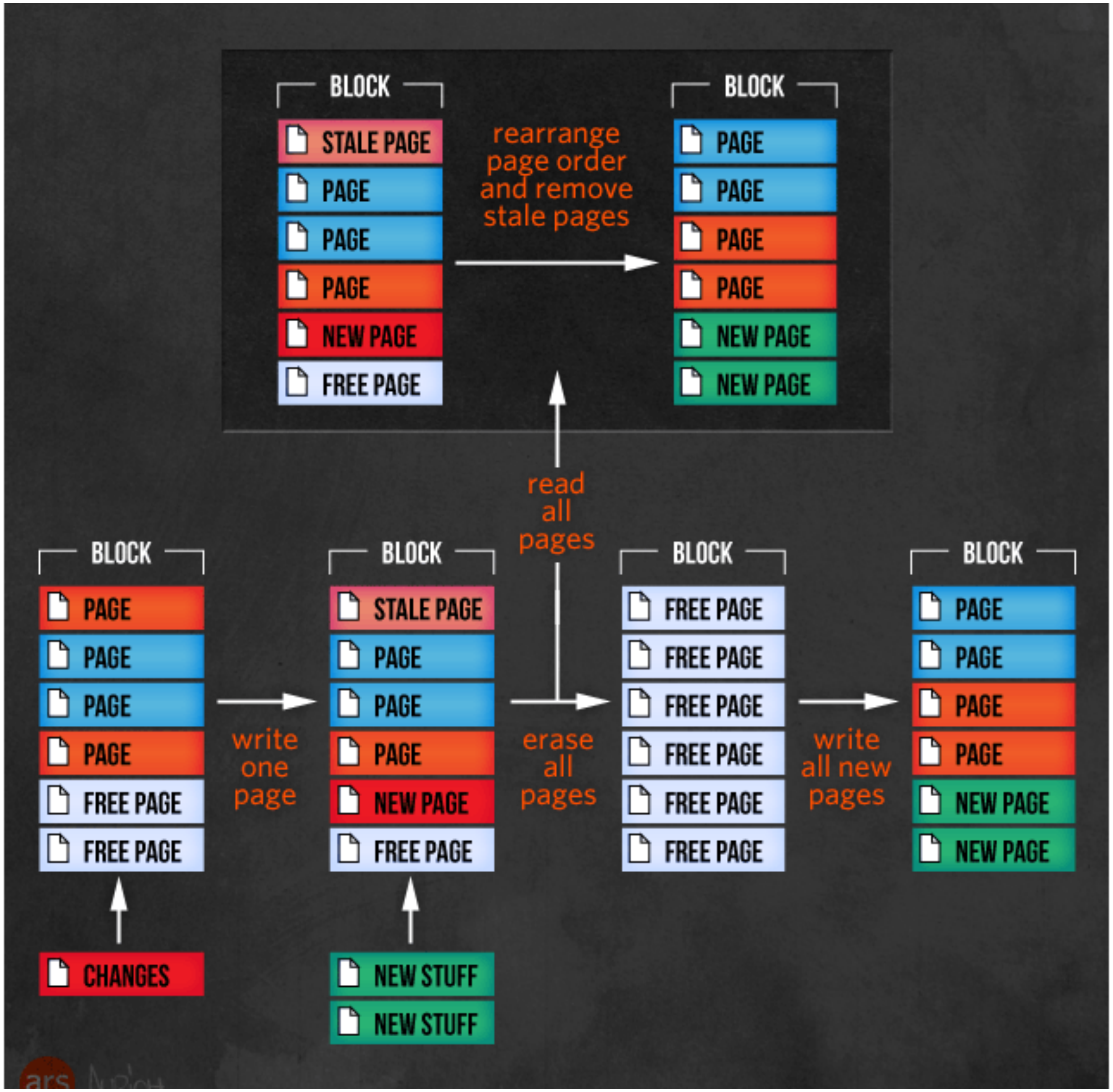
Floating gate transistor



Write vs. overwrite (NAND flash)

- Write: 1 => 0
  - Need to apply high voltage to the gate
- Overwrite: 0 => 1
  - Need to apply much higher voltage than write
  - May stress surrounding cells
  - So dangerous to do on individual pages

Example



BLOCK
STALE PAGE
PAGE
PAGE
PAGE
NEW PAGE
FREE PAGE
rearrange page order and remove stale pages
BLOCK
PAGE
PAGE
PAGE
PAGE
NEW PAGE
NEW PAGE
read all pages
BLOCK
PAGE
PAGE
PAGE
PAGE
FREE PAGE
FREE PAGE
CHANGES
write one page
BLOCK
STALE PAGE
PAGE
PAGE
PAGE
NEW PAGE
FREE PAGE
NEW STUFF
NEW STUFF
erase all pages
BLOCK
FREE PAGE
FREE PAGE
FREE PAGE
FREE PAGE
FREE PAGE
FREE PAGE
write all new pages
BLOCK
PAGE
PAGE
PAGE
PAGE
NEW PAGE
NEW PAGE

Latencies: read, write, and erase

	SLC	MLC	TLC	HDD	RAM
P/E cycles	100k	10k	5k	*	*
Bits per cell	1	2	3	*	*
Seek latency (μs)	*	*	*	9000	*
Read latency (μs)	25	50	100	2000-7000	0.04-0.1
Write latency (μs)	250	900	1500	2000-7000	0.04-0.1
Erase latency (μs)	1500	3000	5000	*	*
Notes	* metric is not applicable for that type of memory				
Sources	P/E cycles [20] SLC/MLC latencies [1] TLC latencies [23] Hard disk drive latencies [18, 19, 25] RAM latencies [30, 52] L1 and L2 cache latencies [52]				

P/E cycle

- P: program/write
- E: erase
  - Every erase damages oxide layer surrounding the floating-gate to some extent

- P/E cycle:
  - Data are written to cells (P): cell value from 1 -> 0
  - Then erased (E): 0 -> 1

Read more

- Solid-state revolution: in-depth on how SSDs really work
- How do SSDs work?
  - http://www.extremetech.com/extreme/210492-extremetech-explains-how-do-ssds-work

References

- How Flash Memory Works
  - https://www.youtube.com/watch?v=msi5GDz9JIw
- Floating Gate Basics
  - http://www.cse.scu.edu/~tschwarz/coen180/LN/flash.html
- Friend of Flash
  - http://www.nnc3.com/mags/LM10/Magazine/Archive/2008/86/040-041_logfs/article.html