CS 110
Computer Architecture

I/O: DMA, Disks, Networking

Instructor:
Sören Schwertfeger and Chundong Wang

https://robotics.shanghaitech.edu.cn/courses/ca/22s

School of Information Science and Technology SIST

ShanghaiTech University

Slides based on UC Berkeley's CS61C

Admin

Exam Plan

- Mid-term (20%) and final (33%) exams
  - Online
- Time (90 minutes + 10 minutes)
  - Mid-term
    - 8:15am to 9:45am, 2nd June, Thursday (UTC+8)
    - 9:45am to **9:55:59am**, online submission (UTC+8)
  - Final
    - Not decided yet

Admin

How to take the mid-term exam

- Students partitioned into groups
  - Exactly the same as which lab you are in
- The lab TA is to organize Tencent Meeting Room
  - Except Lab 9 for mid-term, Chen Meng would take the duty
  - Choose the proper Tencent Meeting Room!!!
- Each lab TA would announce the Tencent Meeting information and make a test with you in advance
  - Do attend the test for trial!!!
- Join the Tencent Meeting Room 30 minutes before the exam
  - i.e., 7:45am for mid-term on 2 June, Thursday
- In case there is anything a student wants the TA to clarify on the exam paper, she or he shall initiate a **private** chat with the TA in the Tencent Meeting Room

Admin

When taking the mid-term exam

- Prepare an electronic device with camera, student ID card, pens, blank paper sheets, etc.
  - TA would check your ID during the exam
- Keep your camera and microphone turned on all through exam
- You would receive the exam paper 3 minutes before the exam through email
- Handwriting your answers on the physical blank paper sheets
  - NOT handwriting on iPad, Surface
  - NOT typing in computer
  - Make clear marks for what questions your answers correspond to
- Take pictures of your answers
  - Write your name on every page and organize them in order
  - With your student ID card → do not cover your answers

Admin

How to submit

- Make a PDF file of your answer pictures
  - One file for one student
  - Name it as Lab$x$-StudentID-Name.pdf
    - e.g., Lab1-2020123456-ZhangSan.pdf
- We have two submission channels
  - Gradescope (Preferred)
  - *Backup*: Email to Sören (soerensch) and Chundong (wangchd)
- Do not submit duplicate copies
- Keep the exam answers for future investigations
- Timestamp would be checked against the submission deadline
  - For mid-term, 9:55:59am on 2nd June, Thursday (UTC+8)
  - Submissions past the deadline would not be evaluated and the student would not receive any score for the exam

Admin

Other Info

- If you cannot attend the exams, do let SIST, Sören, and Chundong know as early as possible
- As to the antigen tests, we are communicating with SIST and the college management
- Project 4
  - New deadline: 8th June
  - Please record a demo video and upload it to Gitlab besides the codes and other essential documenets
    - Show your student IDs in the video to identify yourselves

Review: I/O

- "Memory mapped I/O": Device control/data registers mapped to CPU address space
- CPU synchronizes with I/O device:
  - Polling
  - Interrupts
- "Programmed I/O":
  - CPU execs lw/sw instructions for all data movement to/from devices
  - CPU spends time doing 2 things:
    1. Getting data from device to main memory
    2. Using data to compute

Working with real devices

- "Memory mapped I/O": Device control/data registers mapped to CPU address space
- CPU synchronizes with I/O device:
  - Polling
  - Interrupts
- ~~"Programmed I/O":~~ **DMA**
  - ~~CPU execs lw/sw instructions for all data movement to/from devices~~
  - CPU spends time doing ~~2 things~~:
    1. ~~Getting data from device to main memory~~
    2. Using data to compute

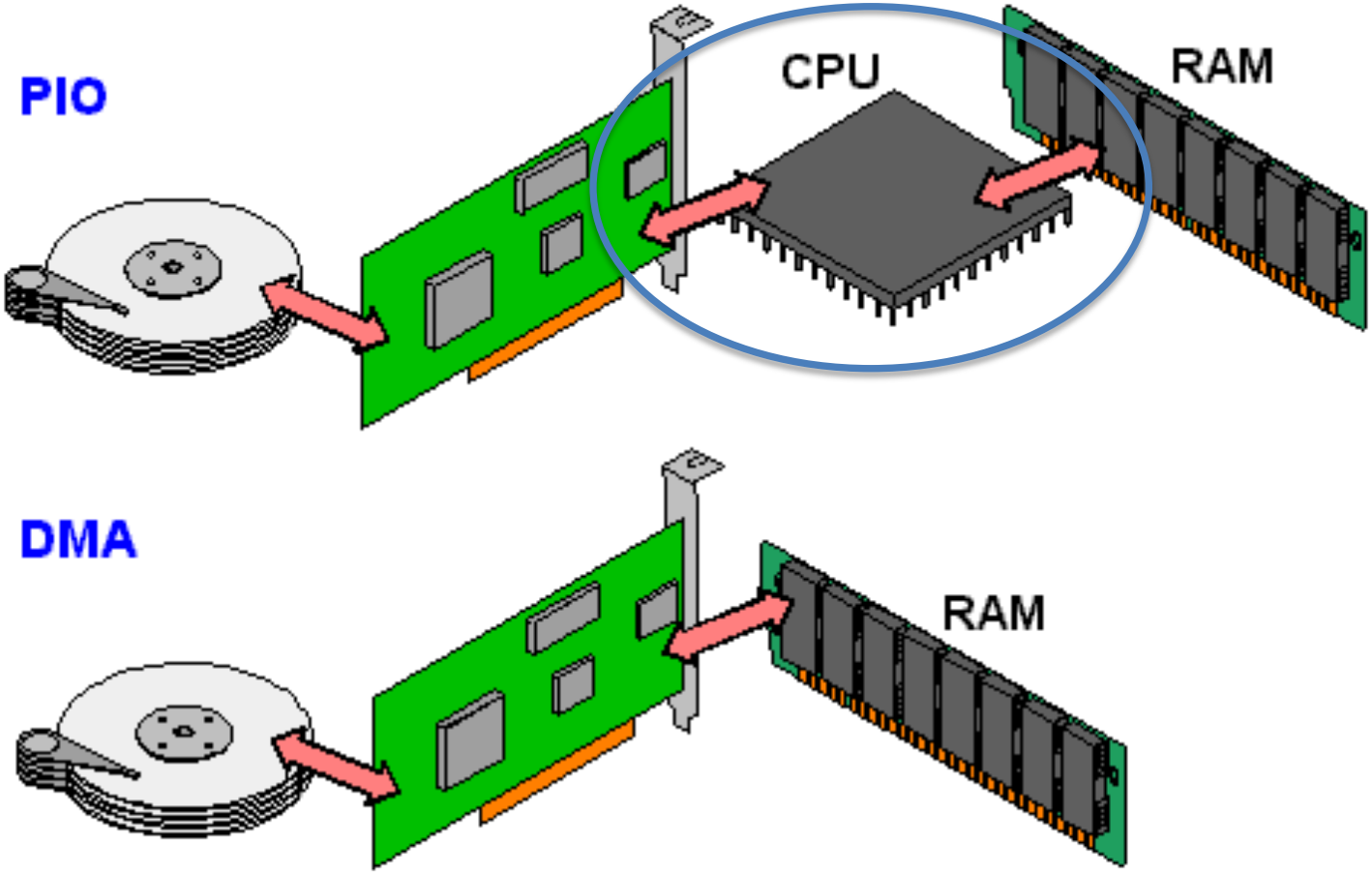
Agenda

- Direct Memory Access (DMA)
- Disks
- Networking

What's wrong with Programmed I/O?

- Not ideal because …
  1. CPU has to execute all transfers, could be doing other work
  2. Device speeds don't align well with CPU speeds
  3. Energy cost of using beefy general-purpose CPU where simpler hardware would suffice
- Until now CPU has sole control of main memory

PIO vs. DMA



Direct Memory Access (DMA)

- Allows I/O devices to directly read/write main memory
- New Hardware: the DMA Engine
- DMA engine contains registers written by CPU:
  - Memory address to place data
  - # of bytes
  - I/O device #, direction of transfer
  - Unit of transfer, amount to transfer per burst

Operation of a DMA Transfer

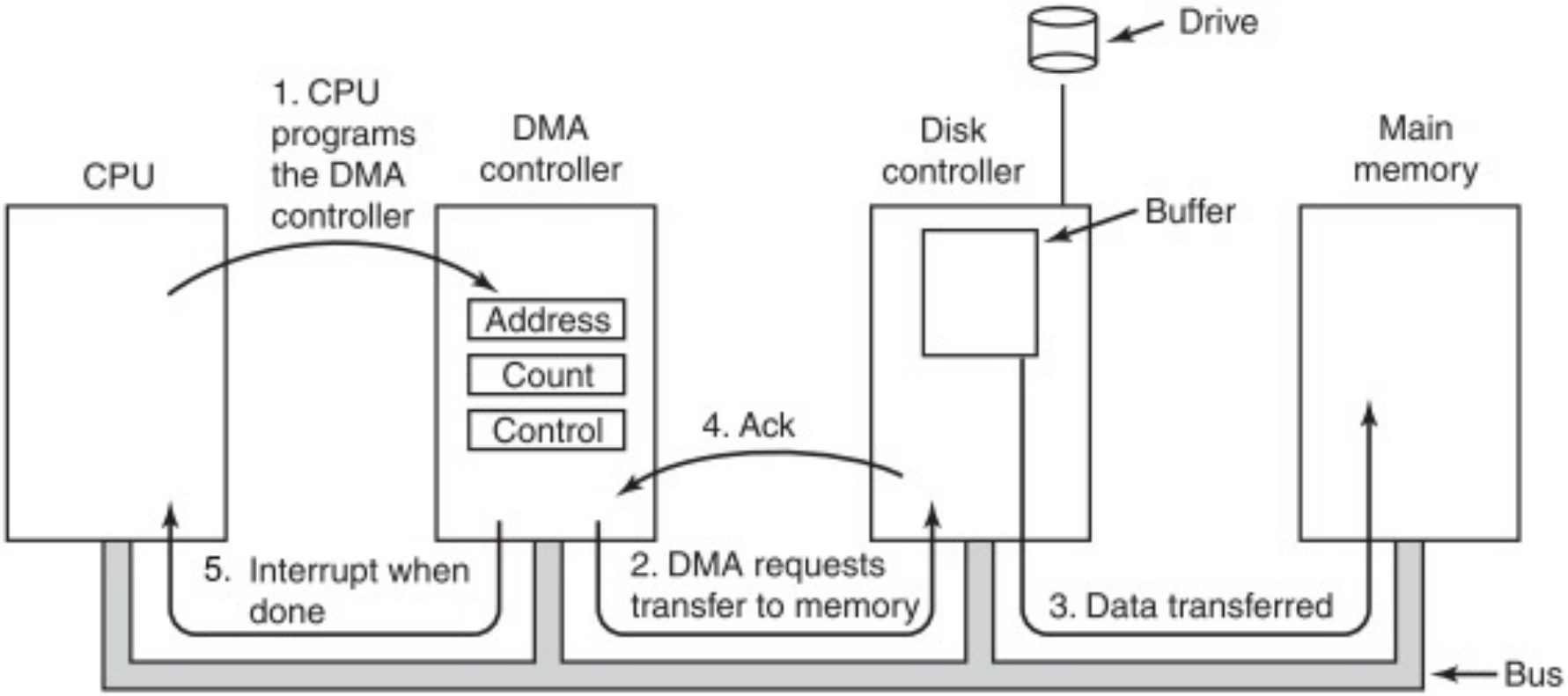


Figure 5-4. Operation of a DMA transfer.

[From Section 5.1.4 Direct Memory Access in *Modern Operating Systems* by Andrew S. Tanenbaum, Herbert Bos, 2014]

DMA: Incoming Data

1. Receive interrupt from device
2. CPU takes interrupt, begins transfer
   - Instructs DMA engine/device to place data @ certain address
3. Device/DMA engine handle the transfer
   - CPU is free to execute other things
4. Upon completion, Device/DMA engine interrupt the CPU again

DMA: Outgoing Data

1. CPU decides to initiate transfer, confirms that external device is ready
2. CPU begins transfer
   - Instructs DMA engine/device that data is available @ certain address
3. Device/DMA engine handle the transfer
   - CPU is free to execute other things
4. Device/DMA engine interrupt the CPU again to signal completion

DMA: Some new problems

- Where in the memory hierarchy do we plug in the DMA engine? Two extremes:
  - Between CPU and L1:
    - Pro: Free coherency
    - Con: Trash the CPU's working set with transferred data
  - Between Last-level cache and main memory:
    - Pro: Don't mess with caches
    - Con: Need to explicitly manage coherency

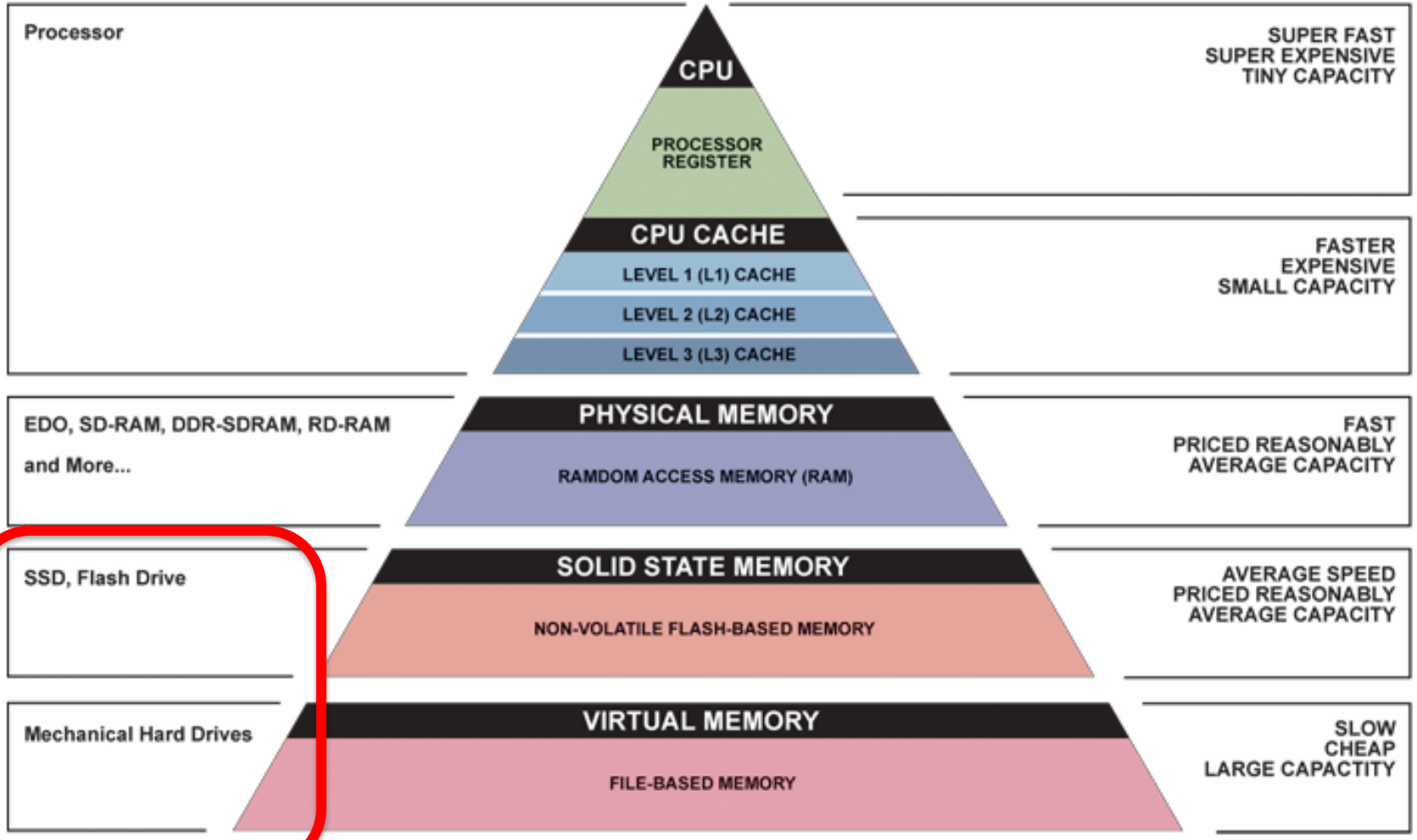
DMA: Some new problems

- How do we arbitrate between CPU and DMA Engine/Device access to memory?
- Three options:
  - Burst Mode
    - Start transfer of data block, CPU cannot access memory in the meantime
  - Cycle Stealing Mode
    - DMA engine transfers a byte, releases control, then repeats - interleaves processor/DMA engine accesses
  - Transparent Mode
    - DMA transfer only occurs when CPU is not using the system bus

Agenda

- Direct Memory Access (DMA)
- Disks
- Networking

Computer Memory Hierarchy



▲ Simplified Computer Memory Hierarchy
Illustration: Ryan J. Leng

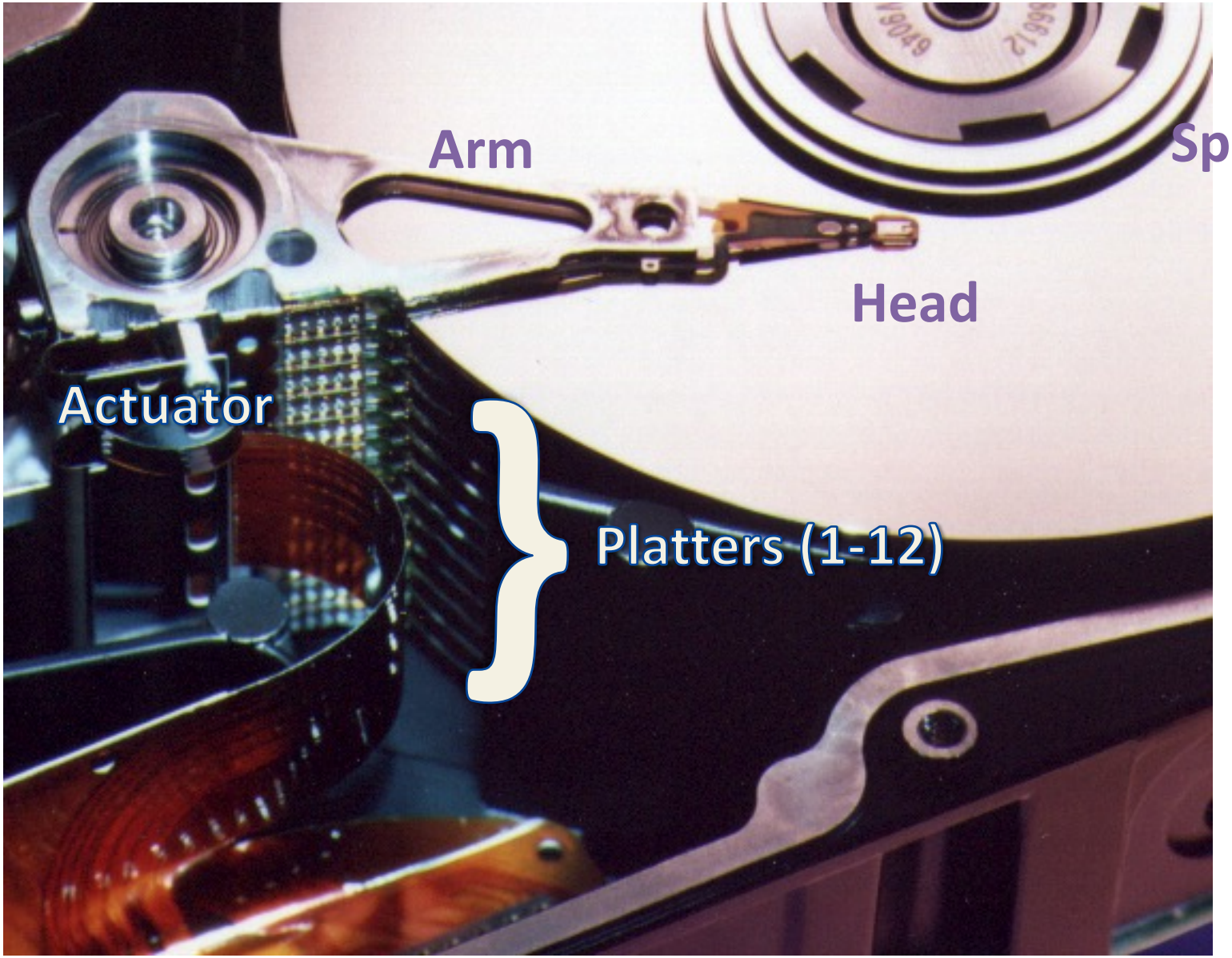
Today

Magnetic Disk – common I/O device

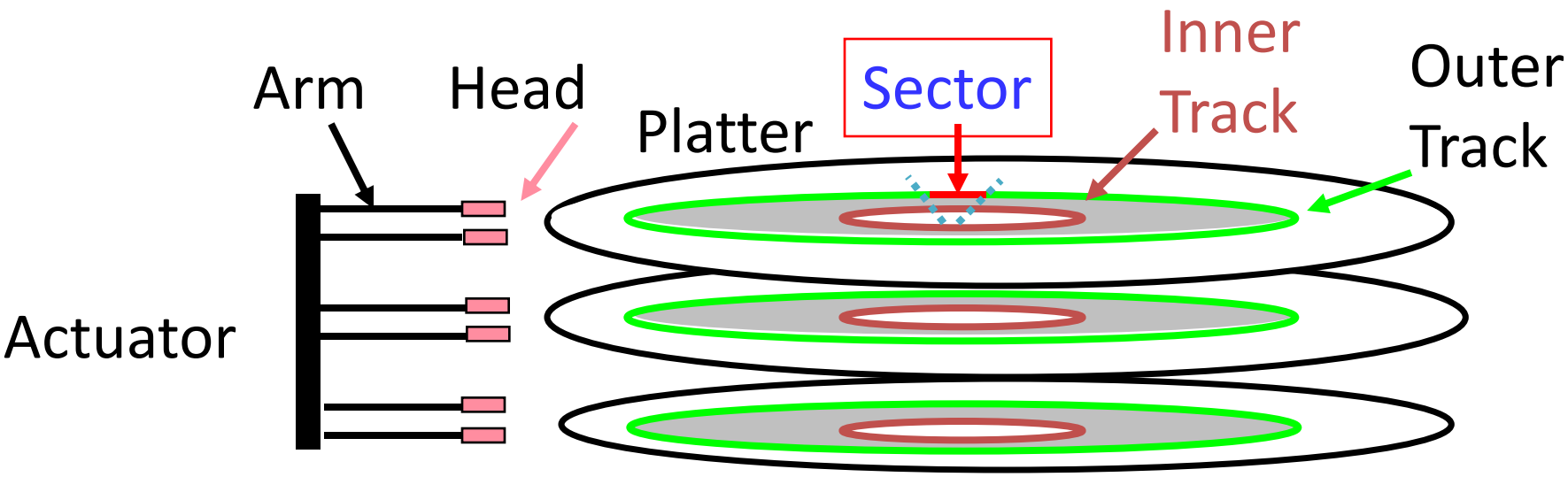
- A kind of computer memory
  - Information stored by magnetizing ferrite material on surface of rotating disk
    - similar to tape recorder except digital rather than analog data
- A type of non-volatile storage
  - Retains its value without applying power to disk.
- Magnetic Disk
  1. Hard Disk Drives (HDD) – faster, more dense, non-removable.

- Purpose in computer systems (Hard Drive):
  1. Working file system + long-term backup for files
  2. Secondary "backing store" for main-memory. Large, inexpensive, slow level in the memory hierarchy (virtual memory)

Photo of Disk Head, Arm, Actuator



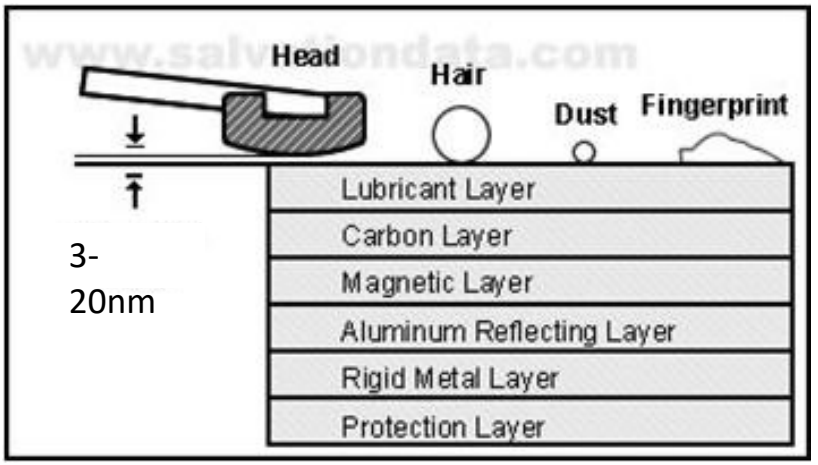
Disk Device Terminology

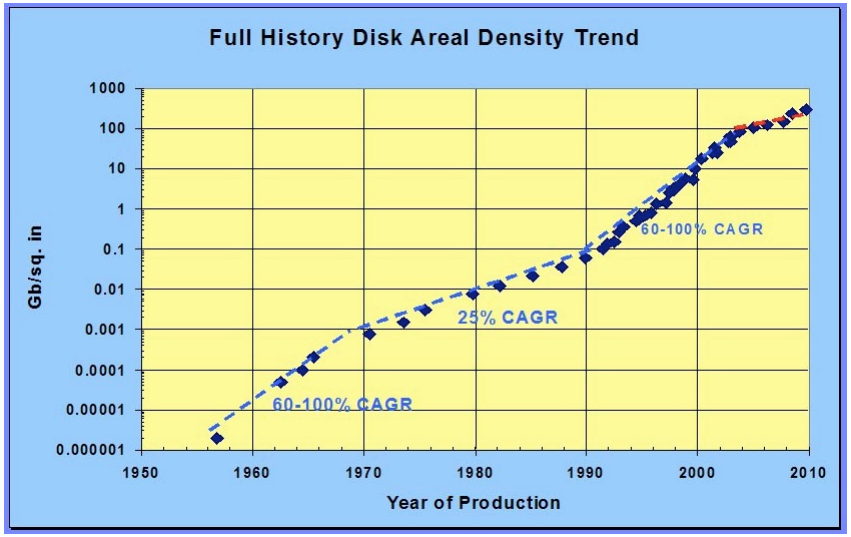


- Several platters, with information recorded magnetically on both surfaces (usually)
- **Bits recorded in tracks, which in turn divided into sectors (e.g., 512 Bytes)**
- **Actuator moves head (end of arm) over track ("seek"), wait for sector rotate under head, then read or write**

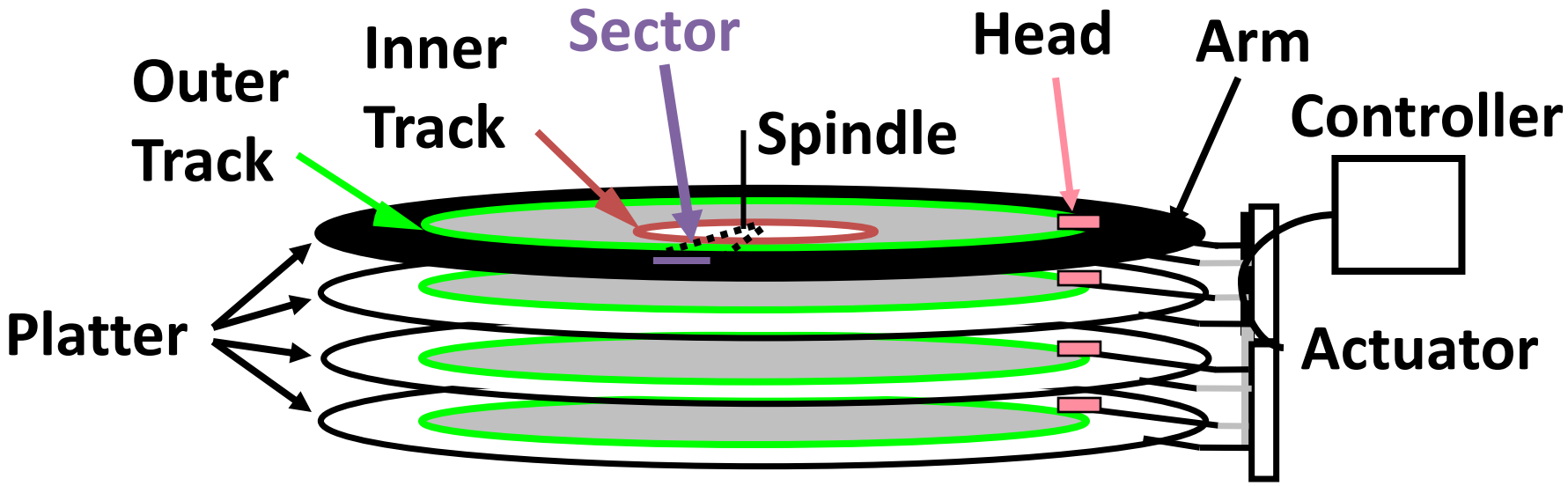
Hard Drives are Sealed. Why?

- The closer the head to the disk, the smaller the "spot size" and thus the denser the recording.
  - Measured in Gbit/in^2
  - ~900 Gbit/in^2 is state of the art
  - Started out at 2 Kbit/in^2
  - ~450,000,000x improvement in ~60 years
- Disks are sealed to keep the dust out.
  - Heads are designed to "fly" at around 3-20nm above the surface of the disk.
  - 99.999% of the head/arm weight is supported by the air bearing force (air cushion) developed between the disk and the head.





Disk Device Performance (1/2)



- **Disk Access Time = Seek Time + Rotation Time + Transfer Time + Controller Overhead**
  - Seek Time = time to position the head assembly at the proper track
  - Rotation Time = time for the disk to rotate to the point where the first sectors of the block to access reach the head
  - Transfer Time = time taken by the sectors of the block and any gaps between them to rotate past the head

Disk Device Performance (2/2)

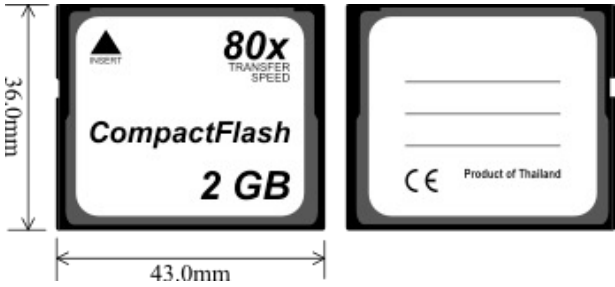
- Average values to plug into the formula:
- Rotation Time: Average distance of sector from head?
  - 1/2 time of a rotation
    - 7200 Revolutions Per Minute ➔120 Rev/sec
    - 1 revolution = 1/120 sec ➔ 8.33 milliseconds
    - 1/2 rotation (revolution) ➔ 4.17 ms
- Seek time: Average no. tracks to move arm?
  - Number of tracks / 3
    - Check Page 9 at http://pages.cs.wisc.edu/~remzi/OSFEP/file-disks.pdf
  - Then, seek time = number of tracks moved × time to move across one track

But wait!

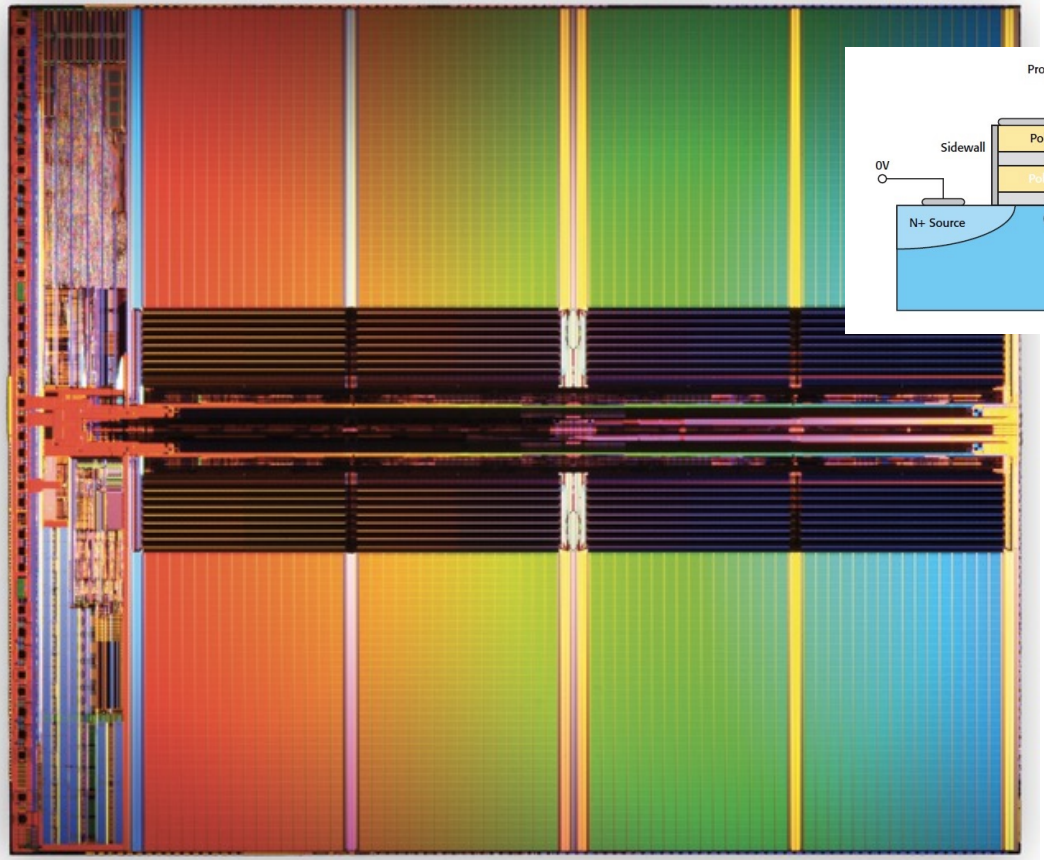
- Performance estimates are different in practice:
- Many disks have on-disk caches, which are completely hidden from the outside world
  - Previous formula completely replaced with on-disk cache access time

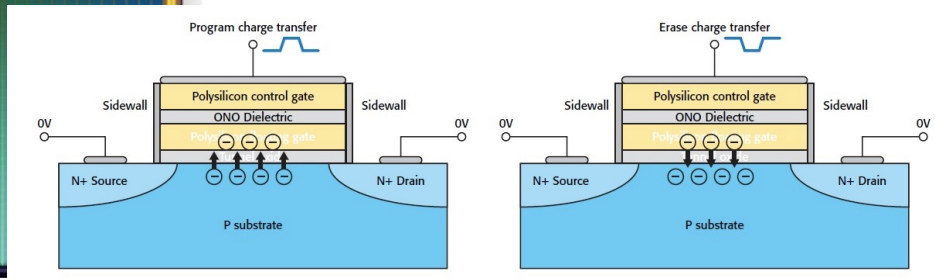
Where does Flash memory come in?

- >15 years ago: Microdrives and Flash memory (e.g., CompactFlash) went head-to-head
  - Both non-volatile (retains contents without power supply)
  - Flash benefits: lower power, seldom crashes (no moving parts, need to spin µdrives up/down)
  - Disk cost = fixed cost of motor + arm mechanics, but actual magnetic media cost very low
  - Flash cost = most cost/bit of flash chips
  - Over time, cost/bit of flash came down, became cost competitive

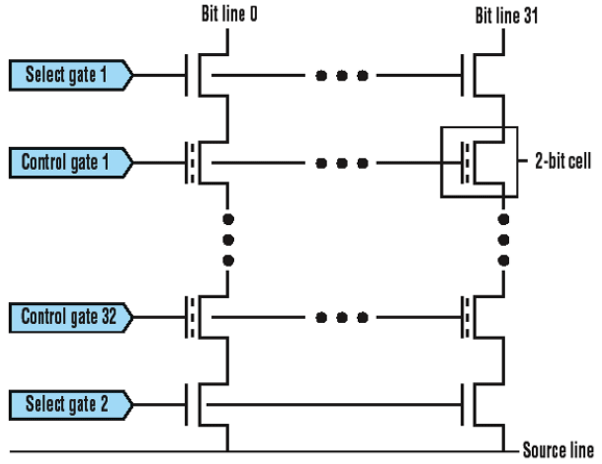


Flash Memory / SSD Technology





2. Micron's triple-level cell (TLC) flash memory stores 3 bits of data in each transistor.



In the basic functional block used in multilevel NAND flash memories, 32 rows of bit lines and 32 control-gate lines form a building block that's repeated many times to form the memory array. The select gate lines are used with the control gate lines to control access to the array.

- NMOS transistor with an additional conductor between gate and source/drain which "traps" electrons. The presence/absence is a 1 or 0
- Memory cells can withstand a limited number of program-erase cycles. Controllers use a technique called *wear leveling* to distribute writes as evenly as possible across all the flash blocks in the SSD.

What did Apple put in its iPods?

Toshiba flash
2 GB

Samsung flash
16 GB

Toshiba 1.8-inch HDD
80, 120, 160 GB

Toshiba flash
32, 64, 128 GB

shuffle

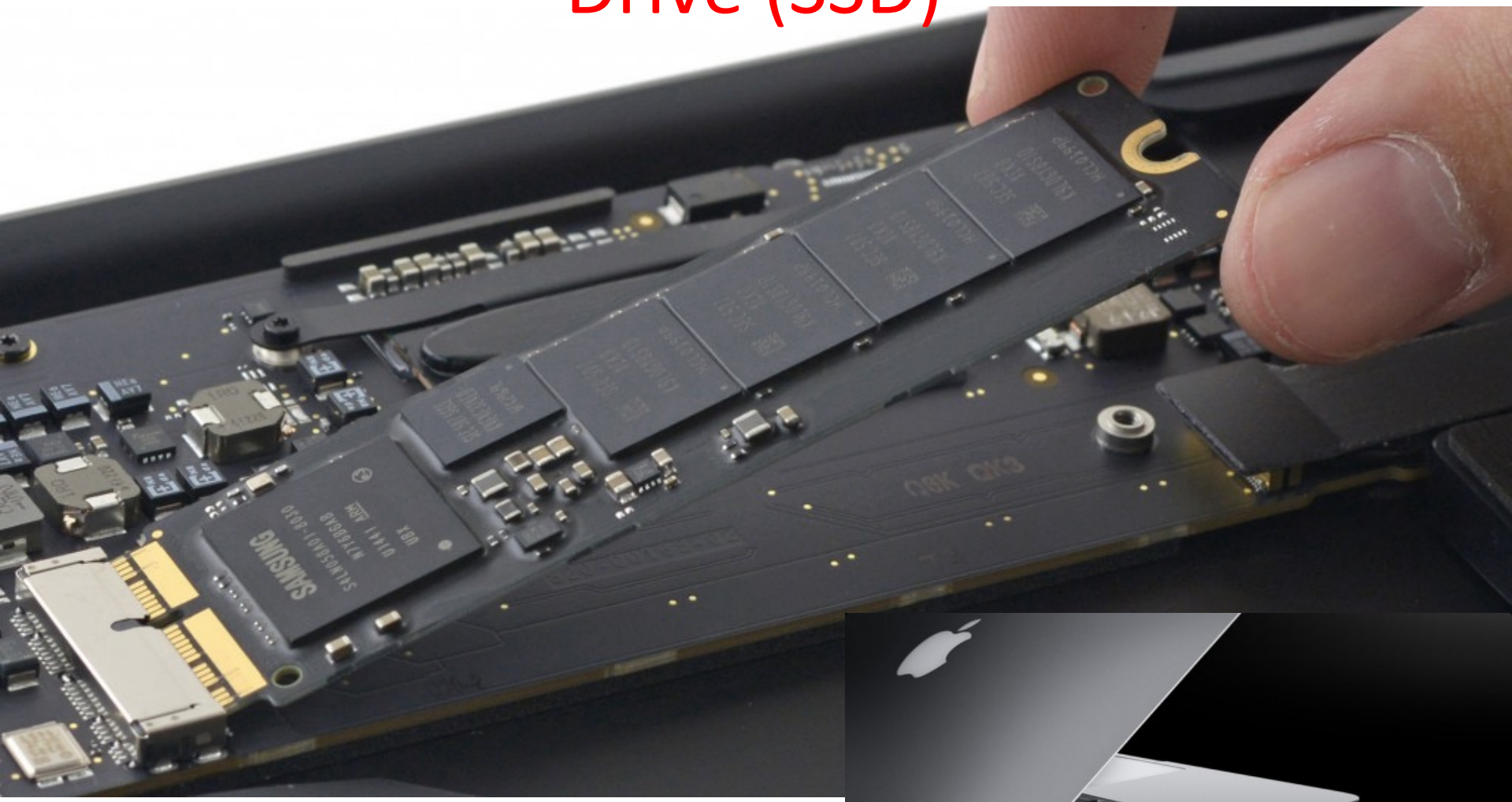
nano

classic

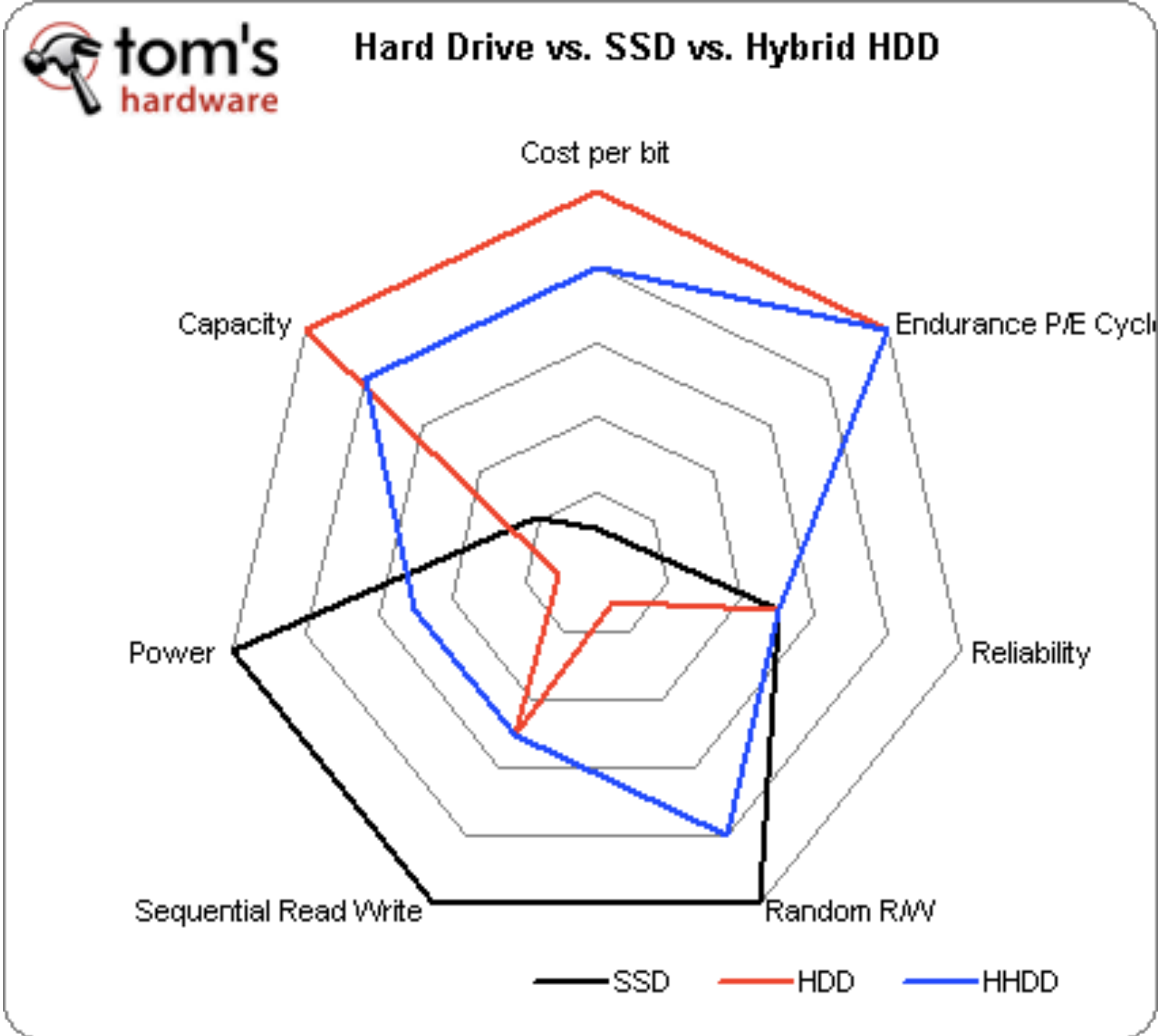
touch

Flash Memory in Smart Phones

Flash Memory in Laptops – Solid State Drive (SSD)



capacities up to 2TB


tom's hardware
Hard Drive vs. SSD vs. Hybrid HDD
Cost per bit
Endurance P/E Cycl
Reliability
Random R/W
Sequential Read Write
Power
Capacity
SSD
HDD
HHDD

Agenda

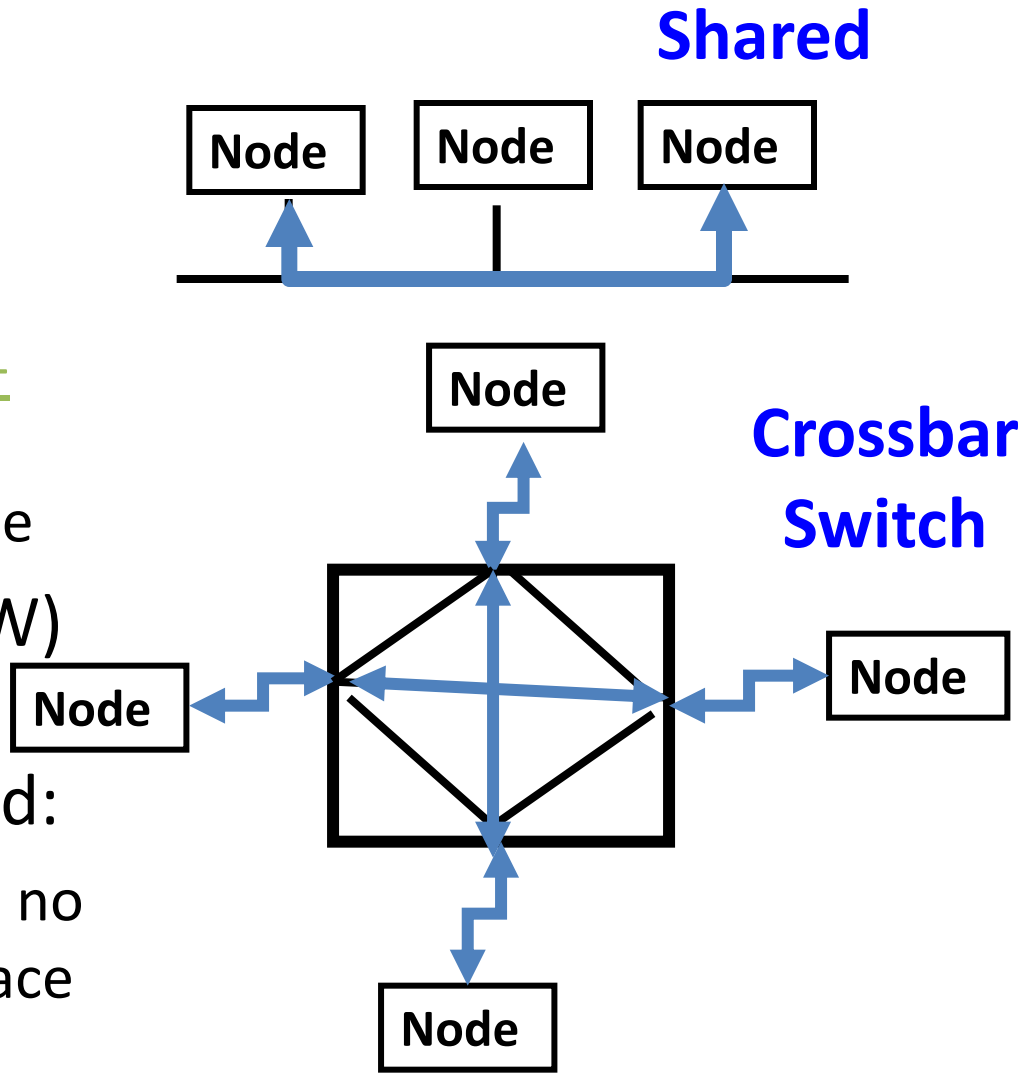
- Direct Memory Access (DMA)
- Disks
- Networking

Networks: Talking to the Outside World

- Originally sharing I/O devices between computers
  - E.g., printers
- Then communicating between computers
  - E.g., file transfer protocol (FTP)
- Then communicating between people
  - E.g., e-mail
- Then communicating between networks of computers
  - E.g., file sharing, www, …

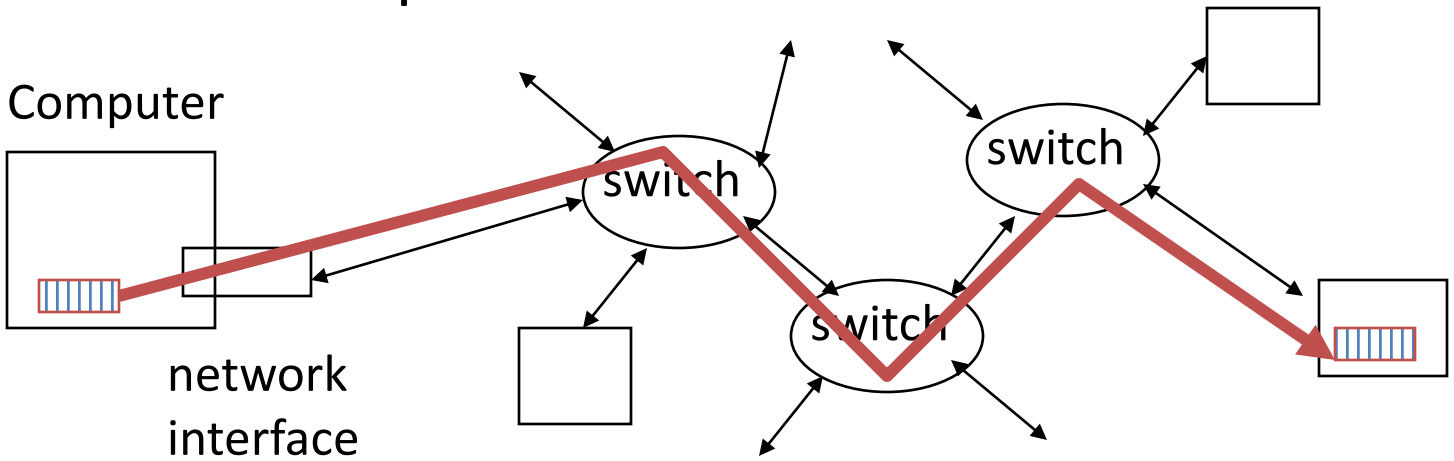
Shared vs. Switch-Based Networks

- Shared vs. Switched:
  - Shared: 1 at a time (CSMA/CD)
  - Switched: pairs ("point-to-point" connections) communicate at same time
- Aggregate bandwidth (BW) in switched network is many times that of shared:
  - point-to-point faster since no arbitration, simpler interface



What makes networks work?

- Links connecting switches and/or routers to each other and to computers or devices

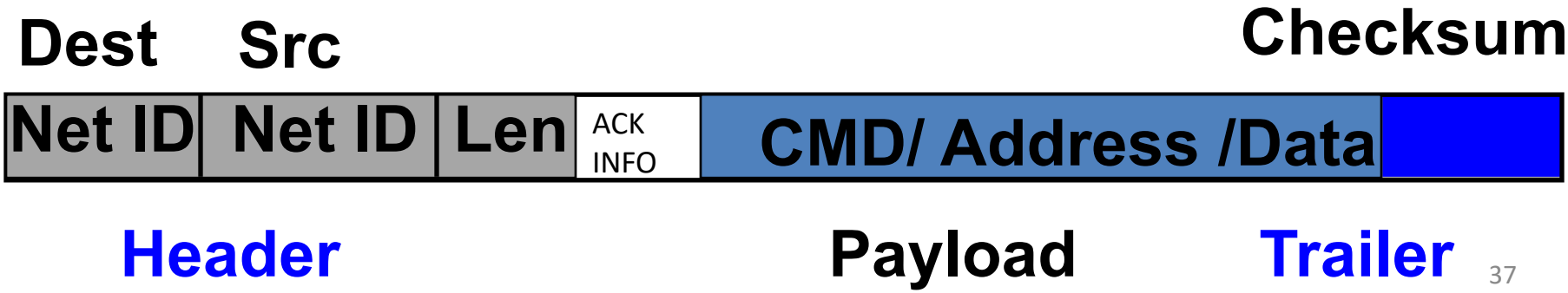


- Ability to name the components and to route packets of information - messages - from a source to a destination

- Layering, redundancy, protocols, and encapsulation as means of abstraction (big idea in Computer Architecture)

Software Protocol to Send and Receive

- SW Send steps
  - 1: Application copies data to OS buffer
  - 2: OS calculates checksum, starts timer
  - 3: OS sends data to network interface HW and says start
- SW Receive steps
  - 3: OS copies data from network interface HW to OS buffer
  - 2: OS calculates checksum, if OK, send ACK; if not, delete message (sender resends when timer expires)
  - 1: If OK, OS copies data to user address space, & signals application to continue



Protocols for Networks of Networks?

What does it take to send packets across the globe?

- Bits on wire or air
- Packets on wire or air
- Delivery packets within a single physical network
- Deliver packets across multiple networks
- Ensure the destination received the data
- Create data at the sender and make use of the data at the receiver

Protocol for Networks of Networks?

Lots to do and at multiple levels!

Use abstraction to cope with complexity of communication

- Hierarchy of layers:
  - Application (chat client, game, etc.)
  - Transport (TCP, UDP)
  - Network (IP)
  - Data Link Layer (Ethernet)
  - Physical Link (copper, wireless, etc.)

Protocol Family Concept

- *Protocol*: packet structure and control commands to manage communication
- *Protocol families (suites)*: a set of cooperating protocols that implement the network stack

- Key to protocol families is that communication occurs logically at the same level of the protocol, called peer-to-peer…

  …but is implemented via services at the next lower level

- Encapsulation: carry higher level information within lower level "envelope"

Inspiration…

- CEO A writes letter to CEO B
  - Folds letter and hands it to assistant
- Assistant
  - Puts letter in envelope with CEO B's full name
  - Takes to China Post
- China Post Office
  - Puts letter in larger envelope
  - Puts name and street address on China Post envelope
  - Puts package on China Post delivery truck
- China Post delivers to other company

Dear Jack,

Hello World

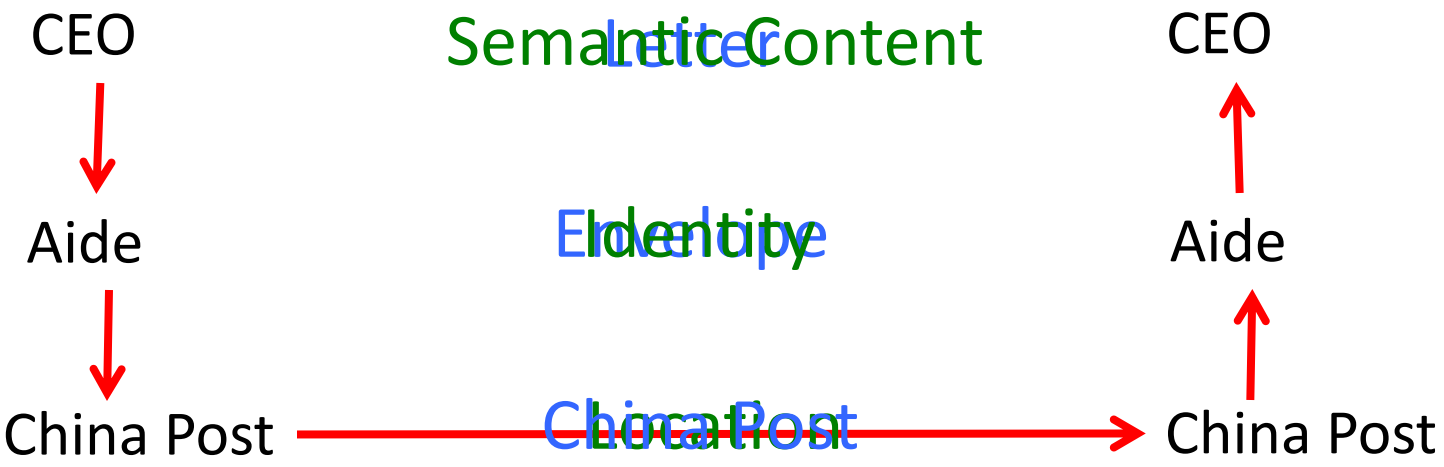
-Pony

The Path of the Letter

"Peers" on each side understand the same things

No one else needs to

Lowest level has most packaging



“And in conclusion…”

- I/O gives computers their 5 senses
- I/O speed range is 100-million to one
- Polling vs. Interrupts
- DMA to avoid wasting CPU time on data transfers
- Disks for persistent storage, replaced by flash
- Networks: computer-to-computer I/O
  - Protocol suites allow networking of heterogeneous components. Abstraction!!!