Welcome to Comp 411!

Computer Organization and Design

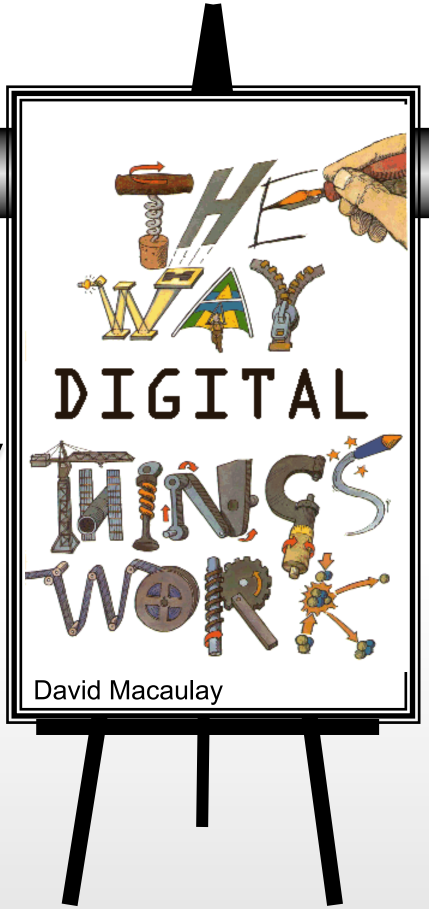
Henry Fuchs

Slides adapted from previous instructors of this class: Don Smith, Montek Singh, Leonard McMillan and from Gary Bishop

Back to McMillan & Chris Terman, MIT 6.004 1999

Thursday, Jan. 8, 2015

Lecture 1



Topics for today

* Course Objectives
* Course Mechanics
* What is Information?
* Computer Abstractions

5pm Lab today postponed to next Tuesday

UNC COMPUTER SCIENCE DISTINGUISHED ALUMNI SPEAKER SERIES

Sponsored by: Jenkins, Wilson, Taylor & Hunt

50 Years of Excellence | 1964 2014

MATT CUTTS

January 8, 2014 – Talk begins at 5:00 p.m., with a reception to follow

Nelson Mandela Auditorium, FedEx Global Education Center Room 1015

“Lessons learned from the early days of Google”

Abstract

Google is a household name now, but it wasn’t always. Matt Cutts joined Google when it was still under 100 people. In his talk, Matt will discuss a few lessons and share some stories from when Google was a scrappy start-up.

Course Objectives

- What You Will Learn:
  - How programs are translated into the machine language
    - And how the hardware executes them
  - The hardware/software interface
  - What determines program performance
    - And how it can be improved
  - How hardware designers improve performance

Meet the Crew...

Instructors:	Henry Fuchs
	Don Smith (for most of January)
Graduate TAs:	Istvan Csapo
(20 hours/week)	Jiangbo Yin
Undergrad TAs:	Dong Yeop Lee
(10 hours/week)	Amanda Lohmann
	Tyler Newsome
	Alex Parker

Required Textbooks

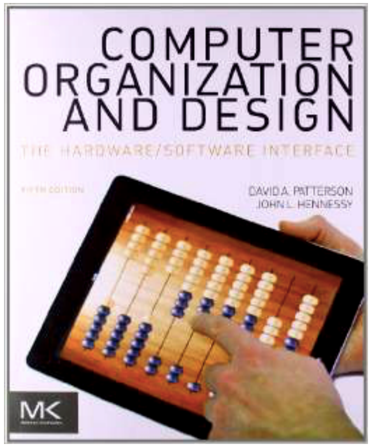
1. *Computer Organization and Design: The Hardware/Software Interface*

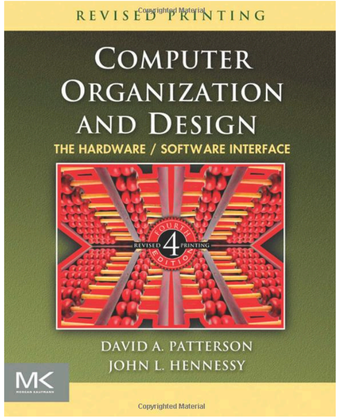
   Patterson and Hennessy

   5th edition, 2014

   ISBN 978-012-407726-3

   Note: the 4th edition, revised printing, can also be used.



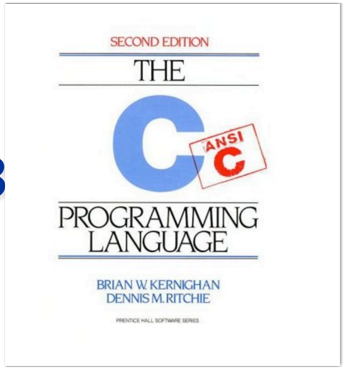


2. *C Programming Language*

   Kernighan and Ritchie

   2nd ed. [paperback], Apr 1988

   ISBN 978-0131103627



Credits

- Most slides in all the lectures adapted from previous instructors of this course: Don Smith, Montek Singh, who adapted them from Leonard McMillan and from Gary Bishop Back to McMillan & Chris Terman, MIT 6.004 1999
- Some slides and other materials are from the book publisher.

Course Mechanics

- Grading:
  - 5-7 Homework Assignments: 10-15%
  - 6-12 Quizzes: 25-30%
  - 2 Exams: 35%
  - 10-12 Lab Assignments: 25%

Course Mechanics

- **Policies:**
  - Problem Sets:
    - Will be distributed on the web. You will typically have 1 week to do them, but sometimes more or less time.
  - Honor Code:
    - The honor code is in effect for all homework, labs, exams etc. Please review the policy on the course website.
  - Lecture Notes:
    - I will attempt to make Lecture Slides, Problem Sets, and other course materials available on the web either before class, or soon after, on the day they are given.

Late Policy

- Late work will be penalized
  - 1 day late (up to 24 hours after due date and time): 10%
  - 2 days late (24-48 hours after the due date/time): 25%
  - 3 days late (48-72 hours after the due date/time): 40%
  - 4+ days late (more than 72 hours after the due date/time):
    - No assurance that assignment will be graded
    - 90% deduction if submitted before the assignment solution is discussed in class
    - no credit if submitted after the assignment solution is discussed in class

- All scores count
  - No dropping of lowest scores

Discussion Board

- Please post questions (even private ones to instructor) on the discussion board on PIAZZA
  - https://piazza.com/unc/spring2015/comp411/home
  - You should receive an enrollment notice soon.
  - The TAs and I will check it regularly.

  - Note: The Honor Code applies to posts on PIAZZA

Prerequisites

- COMP401: Foundations of Programming
  - This is a hard prerequisite
  - You may be able to substitute another programming course, but please first talk to me!
- You need to know at least the following concepts:
  - basic data types: integers, characters, Boolean, etc.
  - basic arithmetic operators and expressions
  - "if-then-else" constructs, and "while" and "for" loops
  - function and procedure calls
  - basic Boolean operators (AND, OR, XOR, etc.)
- If you don't know many of the above concepts, please talk to me!

Detailed Syllabus

- On course website in Sakai.
- Most materials for this course will be on Sakai:

  -Syllabus

  -Lecture slides

  -Problem Sets

  -Labs

  -Schedule for Quizzes, Midterm, Final,..

How NOT to do well in this course

- BIG mistakes
  - Skipping lectures
  - Not reading the book (only reviewing lecture slides)
  - Not spending enough time to do homework
    - Start early. Many problem sets are too hard to attempt the night before.
  - Not asking questions in class
  - Not discuss concepts with other students
    - But all work handed in must be your own (see Honor Code)
  - Looking up solutions from earlier semesters = cheating. Not worth it.

Some Class Mechanics

- You’ll get CS department login so you can access Linux servers (e.g. classroom.cs.unc.edu)
  - You’ll get instructions in email on how to access your login
  - See a TA if you haven’t received the email by Friday
- In general, all (or almost all) homework will be announced, posted and submitted on Sakai.
- WEEKLY QUIZZES – Thursdays 3:30-4pm
  - Covering material covered so far in class
- Weekly schedule
  - Tuesdays 3:30-4:45 lecture
  - Thursday
    3:30-4pm Quiz, closed book
    4:15-5:30 lecture

Comp 411: Course Website

INTRODUCTION	**http://comp411s14.web.unc.edu**
MAIN SITE	**https://sakai.unc.edu/**

Who I am: Henry Fuchs

- U Utah: PhD 1975
  - Computer Science (3D Computer Graphics, virtual reality)
    - Digital logic design, computer architecture
- UNC 1978- now
  - 3D Displays: autostereo, head-mounted displays
  - Real-time graphics image generation engines: Pixel-Planes
  - 3D tracking devices
  - 3D scanning of room-sized environments & people
  - Medical, surgical, training applications
  - Tele-presence
  - Teaching: graphics, intro to robotics, computer organization (years ago)

Goal 1: Demystify Computers

- Strangely, most people (even some computer scientists) are afraid of computers.

- We are only afraid of things we do not understand!
  - I do not fear computers. I fear the lack of them.

    - Isaac Asimov (1920 – 1992)

  - Fear is the main source of superstition, and one of the main sources of cruelty. To conquer fear is the beginning of wisdom.

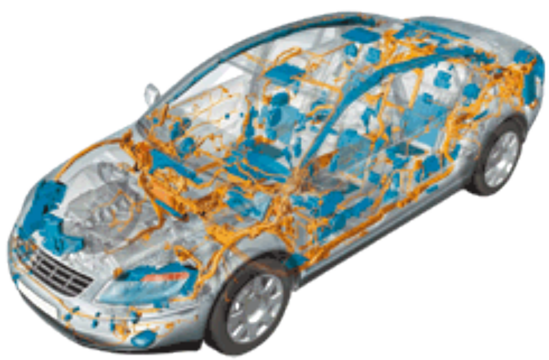
    - Bertrand Russell (1872 – 1970)

Where do you find computers today?

- Besides on your desk, or in your lap, of course!
  - Let's hear from you!
- How have computers revolutionized life in the 21st century?
  - Automobiles
  - Mobile phones, PDAs, games
  - Massive distributed projects: e.g., human genome project
  - World Wide Web
  - Search engines
- What has fueled progress in computer technology?
  - Moore's Law

Computers Everywhere

- The computers we are used to
  - Desktops
  - Laptops
  - Servers
    - Network-based
    - "Cloud computers"
  - Embedded processors
    - Hidden as components inside: cars, phones, toasters, irons, wristwatches, happy-meal toys



Portable gadgets dominate!

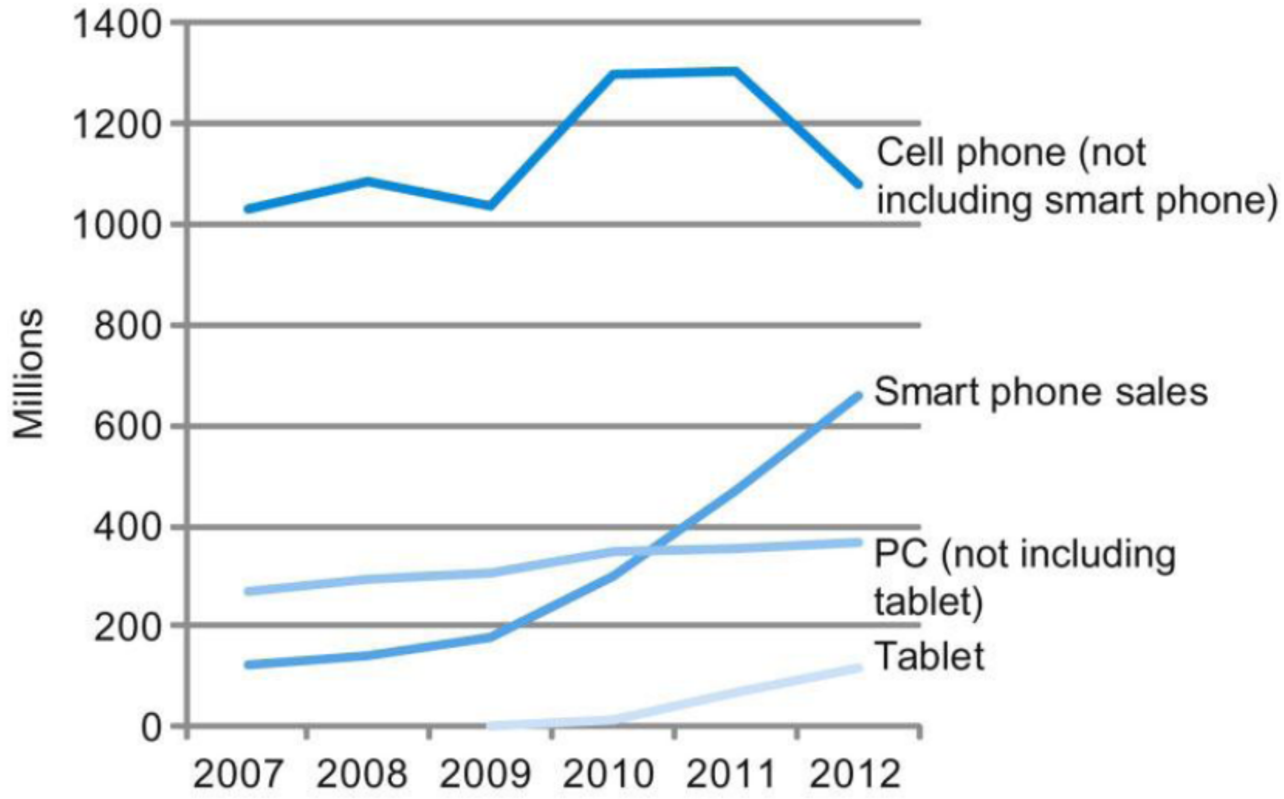
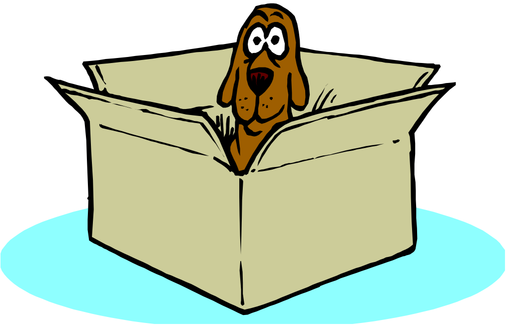


FIGURE 1.2 The number manufactured per year of tablets and smart phones, which reflect the PostPC era, versus personal computers and traditional cell phones. Smart phones represent the recent growth in the cell phone industry, and they passed PCs in 2011. Tablets are the fastest growing category, nearly doubling between 2011 and 2012. Recent PCs and traditional cell phone categories are relatively flat or declining.

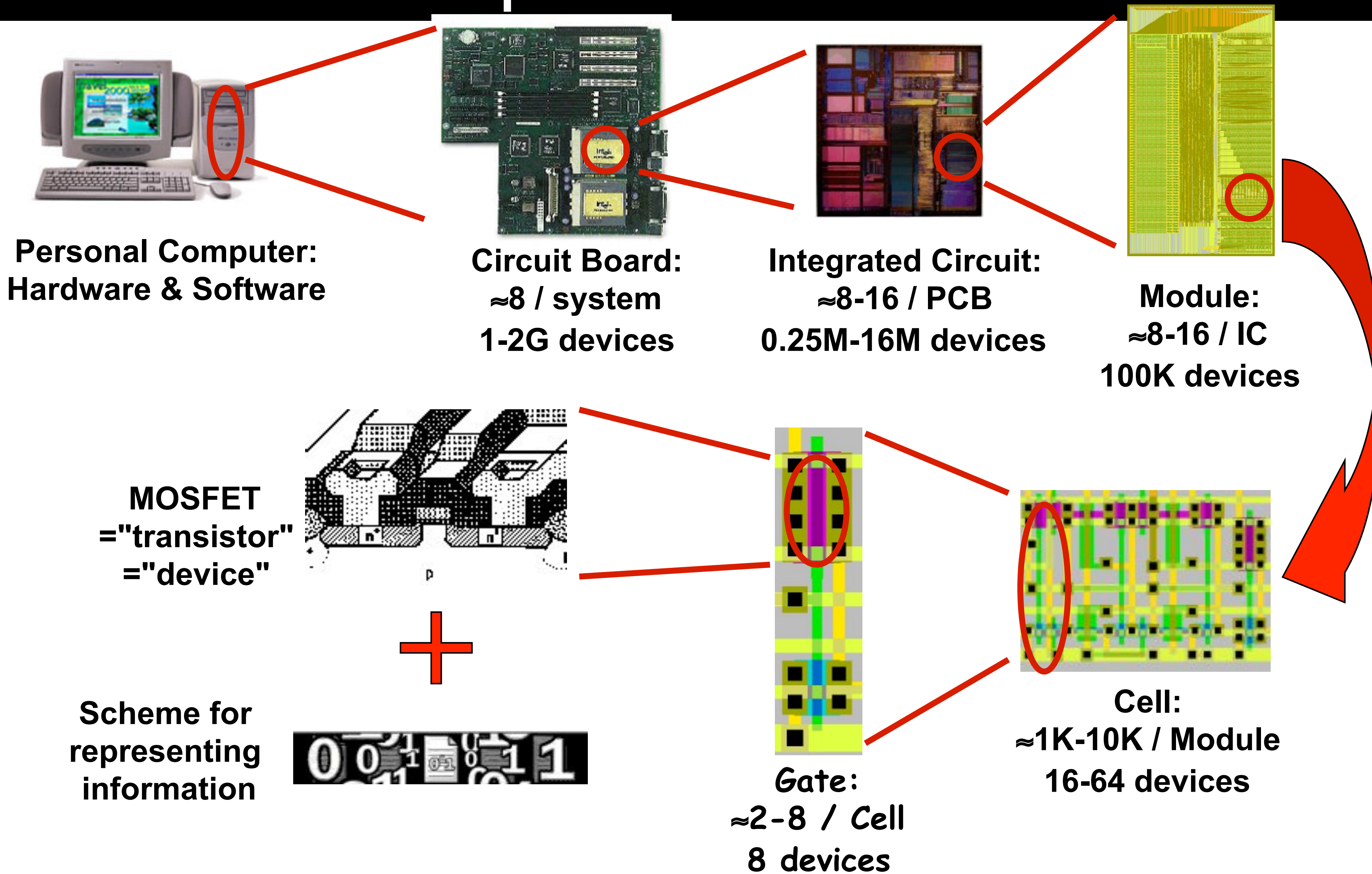
Goal 2: Power of Abstraction

- What is *abstraction?*
  - Define a function, develop a robust implementation, and then put a box around it.



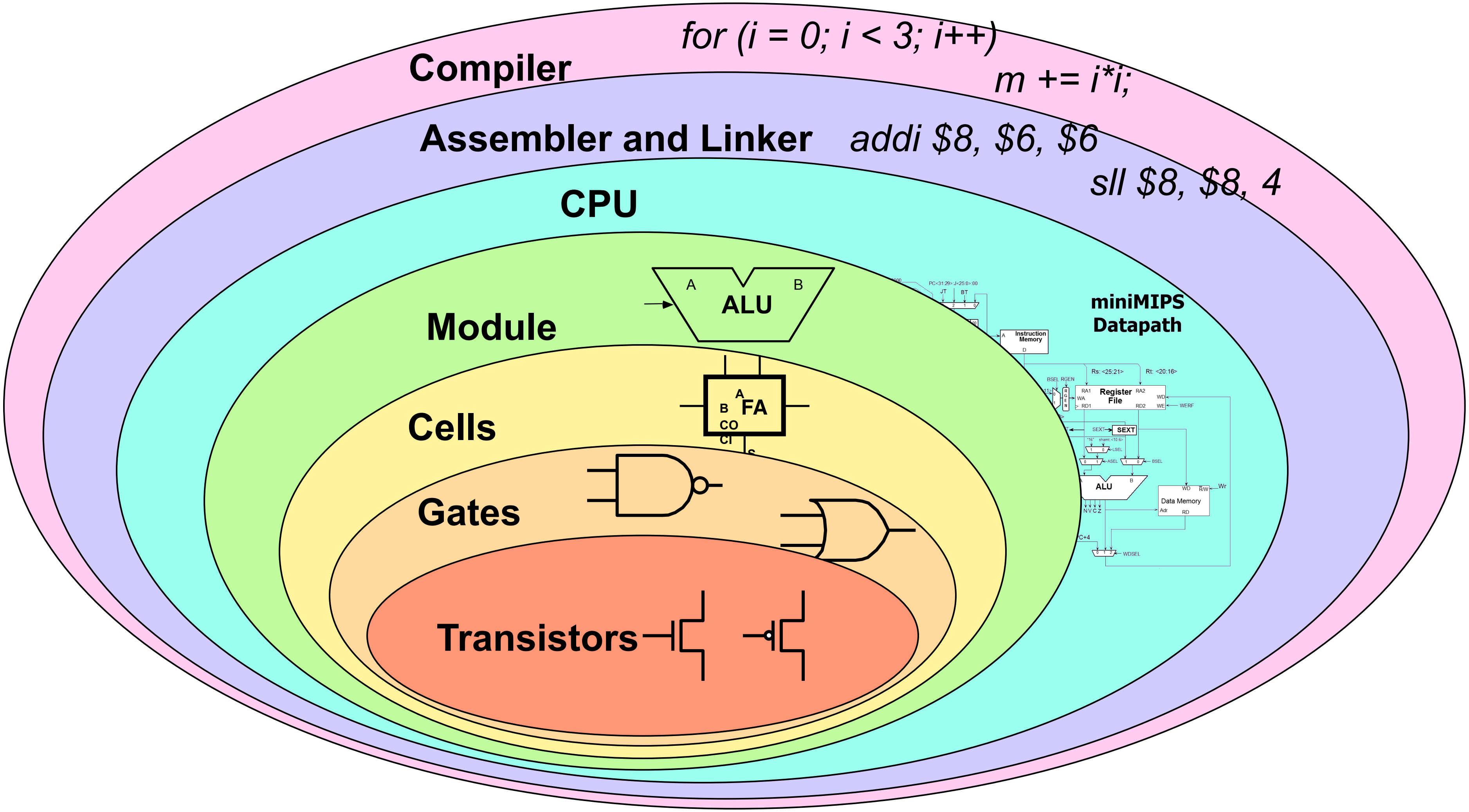
- Why is abstraction useful?
  - enables us to create unfathomable machines called computers
  - imagine a billion --- 1,000,000,000 (#of transistors in a CPU)

Abstraction is key to building systems with >1G components



A Computer System

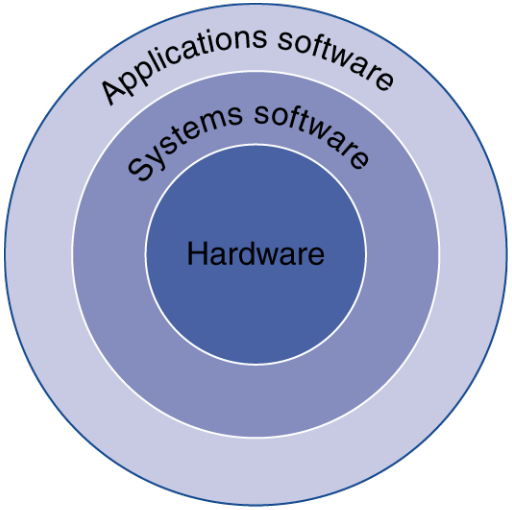
- What is a computer system?
- Where does it start?
- Where does it end?



Understanding Performance

- Algorithm
  - Determines number of operations executed
- Programming language, compiler, architecture
  - Determine number of machine instructions executed per operation
- Processor and memory system
  - Determine how fast instructions are executed
- I/O system (including OS)
  - Determines how fast I/O operations are executed

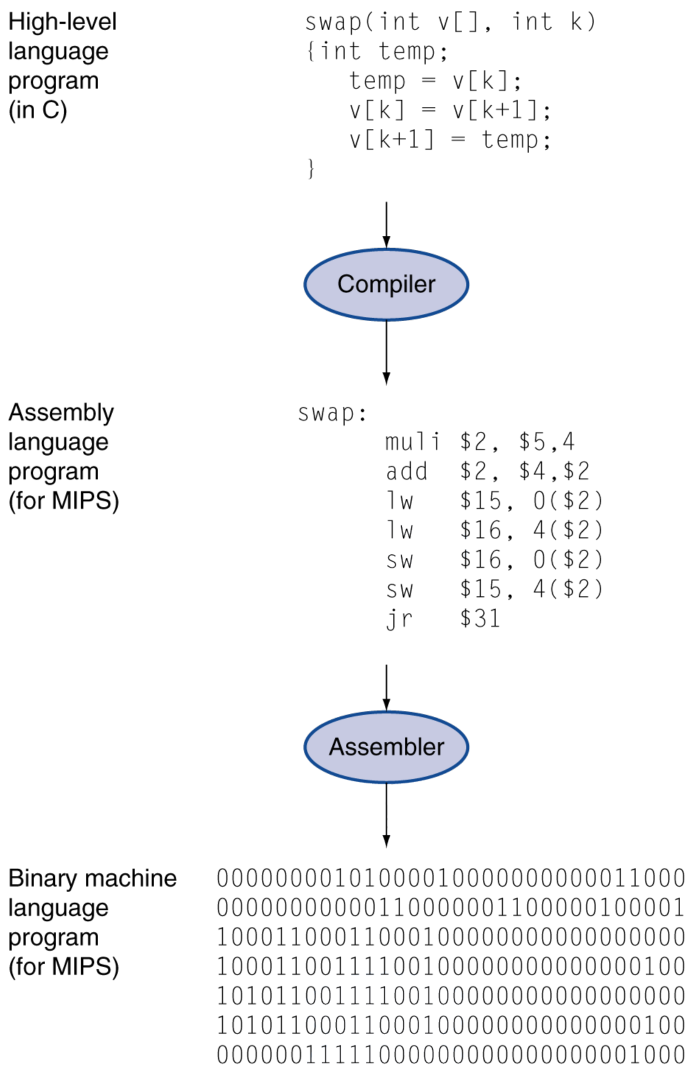
Below Your Program



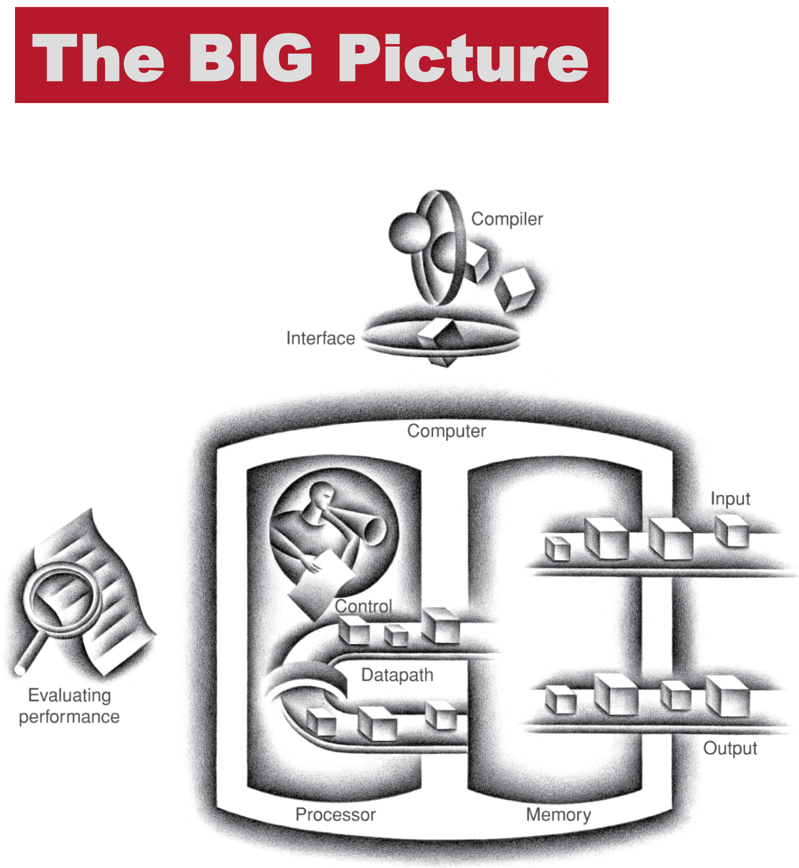
- Application software
  - Written in high-level language
- System software
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- Hardware
  - Processor, memory, I/O controllers

Levels of Program Code

- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data

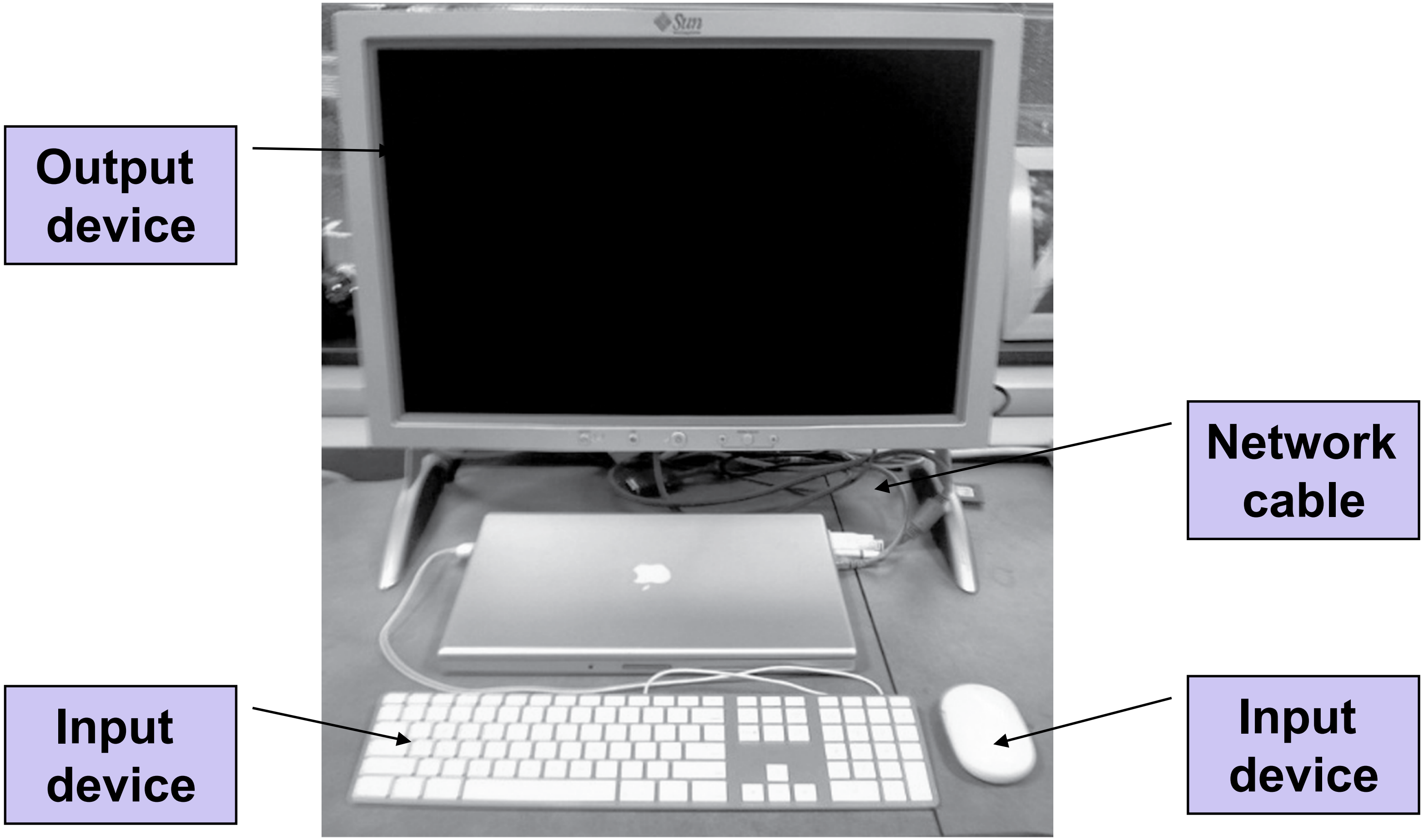


Components of a Computer



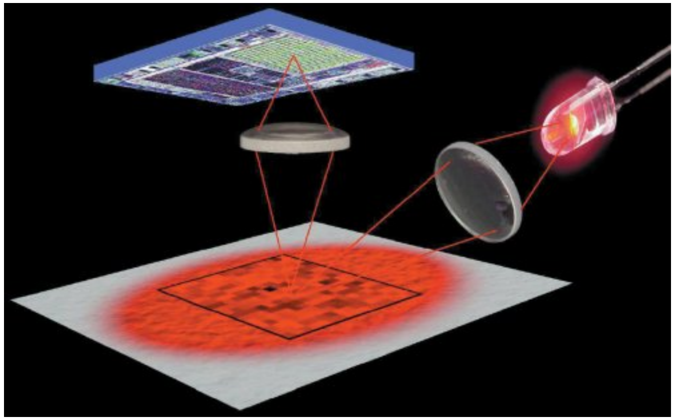
- Same components for all kinds of computer
  - Desktop, server, embedded
- Input/output includes
  - User-interface devices
    - Display, keyboard, mouse
  - Storage devices
    - Hard disk, CD/DVD, flash
  - Network adapters
    - For communicating with other computers

Anatomy of a Computer



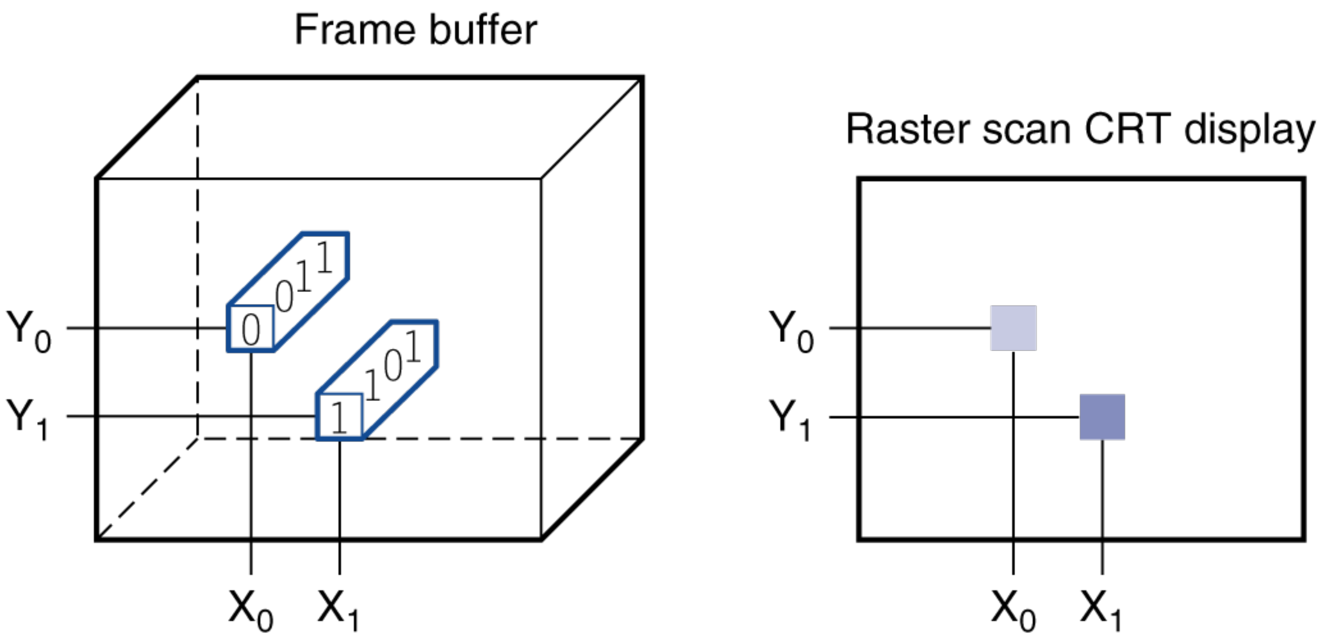
Anatomy of a Mouse

- Optical mouse
  - LED illuminates desktop
  - Small low-res camera
  - Basic image processor
    - Looks for x, y movement
  - Buttons & wheel
- Supersedes roller-ball mechanical mouse



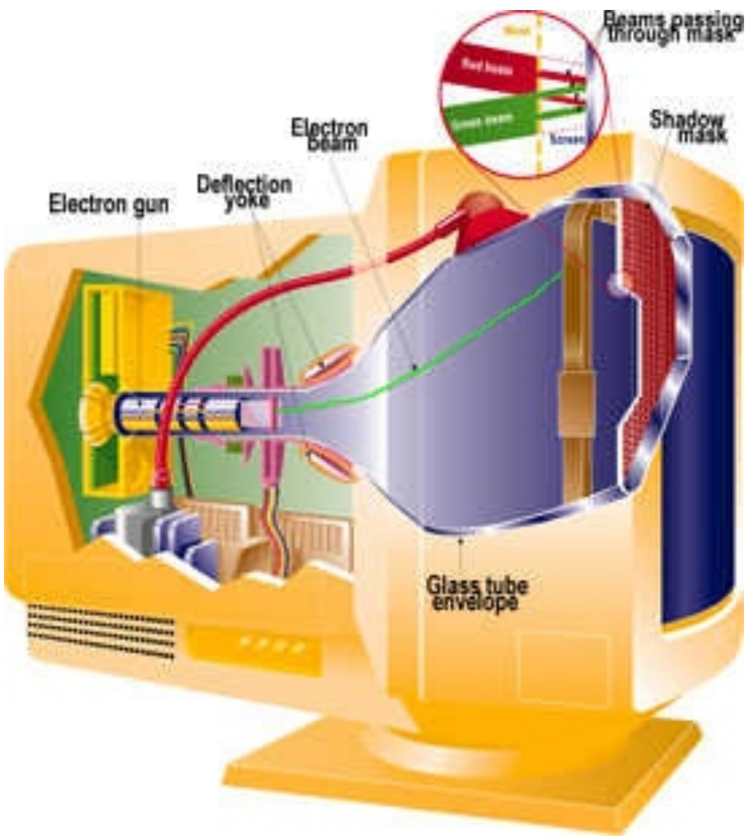
Through the Looking Glass

- LCD screen: picture elements (pixels)
  - Mirrors content of frame buffer memory

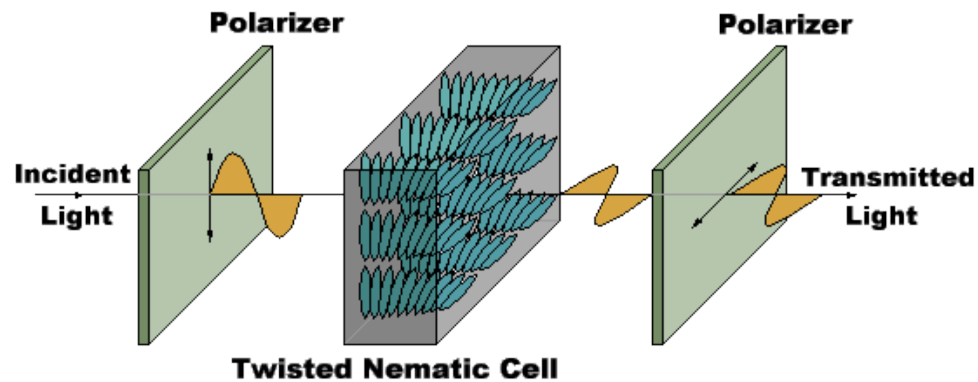


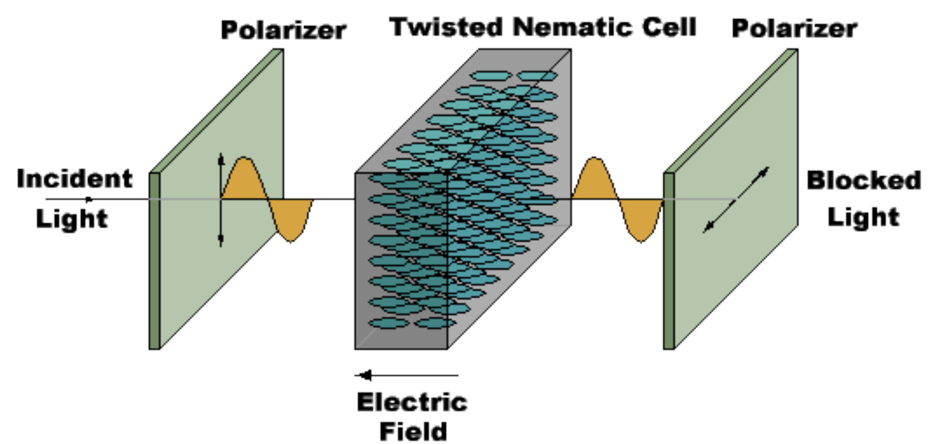
Displays

- Cathode Ray Tube (CRT)
  - The "last vacuum tube"
  - Now nearing extinction

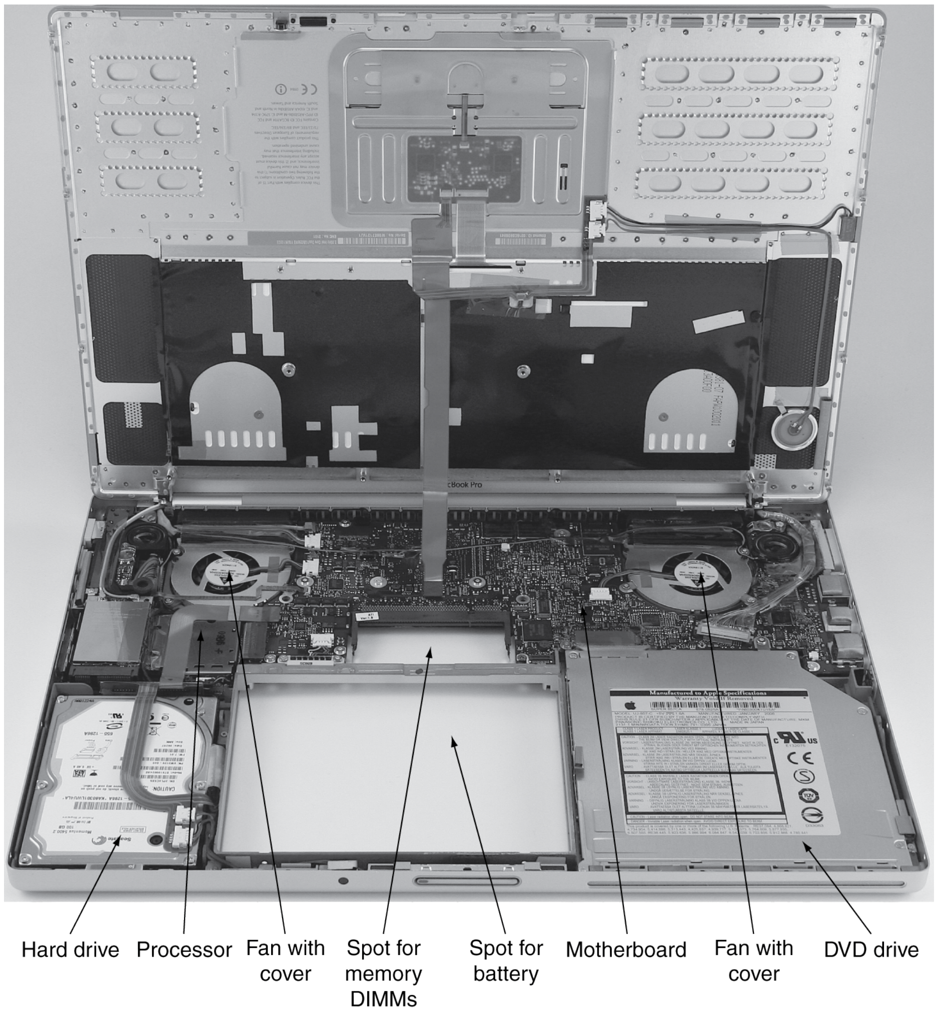


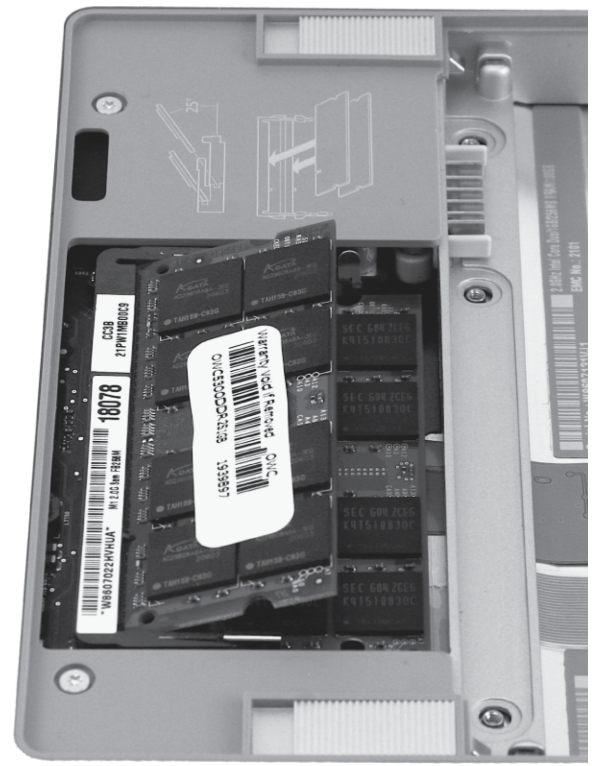
- Liquid Crystal Displays (LCDs)





Opening the Box



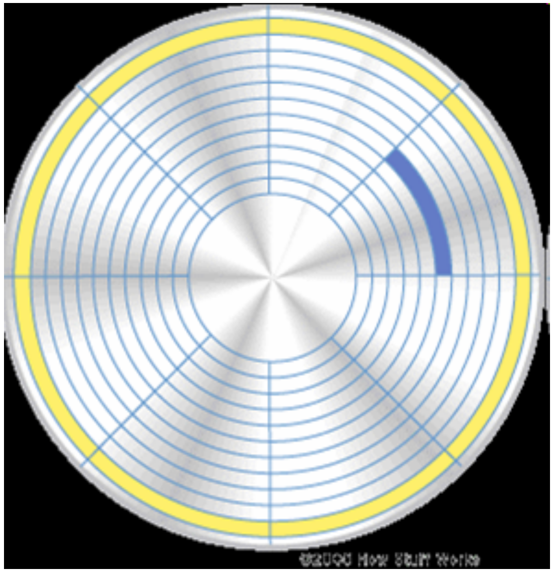


Example: Laptop as today's desktop

FIGURE 1.5 A desktop computer. The liquid crystal display (LCD) screen is the primary output device, and the keyboard and mouse are the primary input devices. On the right side is an Ethernet cable that connected the laptop to the network and the Web. The lap top contains the processor, memory, and additional I/O devices. This system is a Macbook Pro 15" laptop connected to an external display.

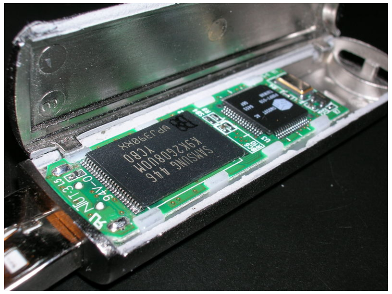
Under the Covers

- Input
- Output
- Memory
- Processing
  - Datapath
  - Control



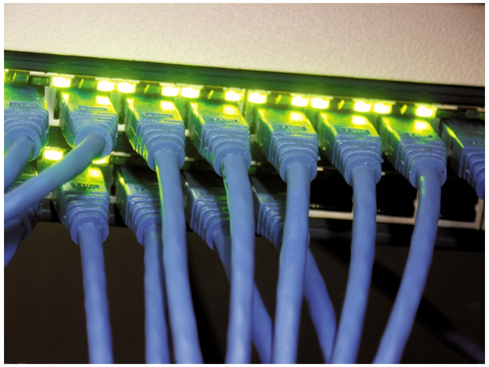
A Safe Place for Data

- Volatile main memory
  - Loses instructions and data when power off
- Non-volatile secondary memory
  - Magnetic disk
  - Flash memory
  - Optical disk (CDROM, DVD)



Networks

- Communication and resource sharing
- Local area network (LAN): Ethernet
  - Within a building
- Wide area network (WAN): the Internet
- Wireless network: WiFi, Bluetooth



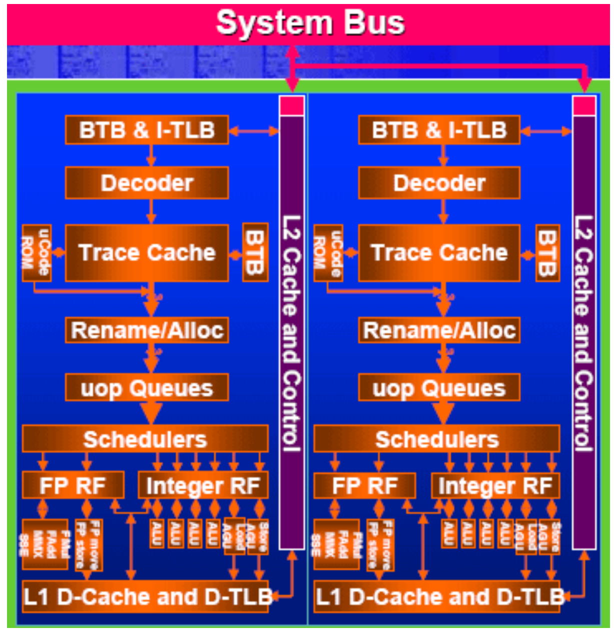
Memory (=RAM)

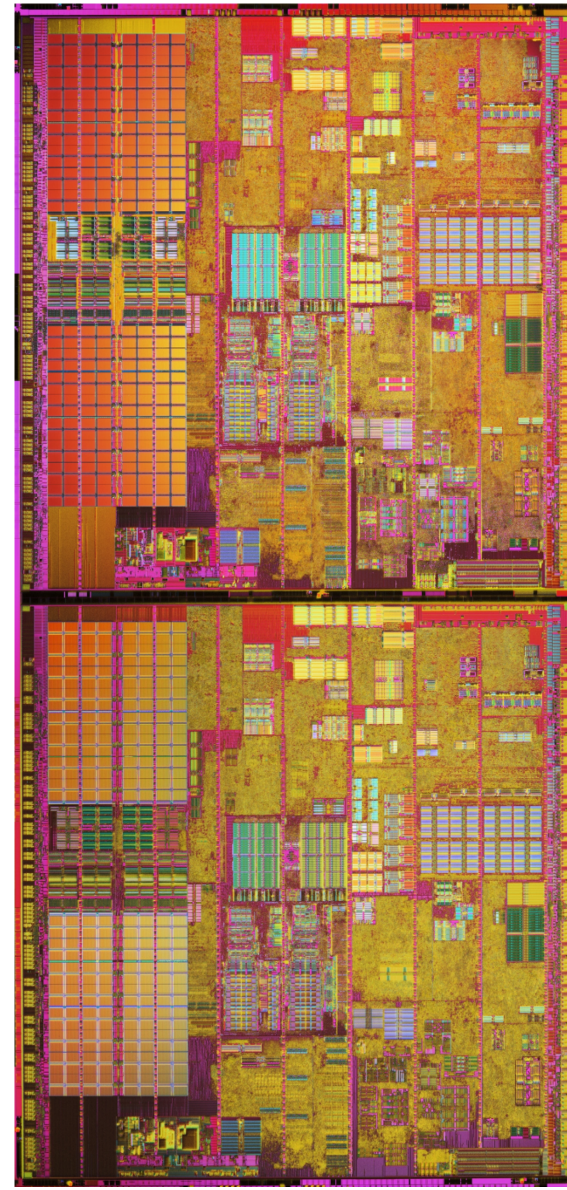
- RAM = volatile memory

DRAM chip capacity

DRAM	
Year	Size
1980	64 Kb
1983	256 Kb
1986	1 Mb
1989	4 Mb
1992	16 Mb
1996	64 Mb
1999	256 Mb
2002	1 Gb
2014	4-8Gb

Inside the Processor



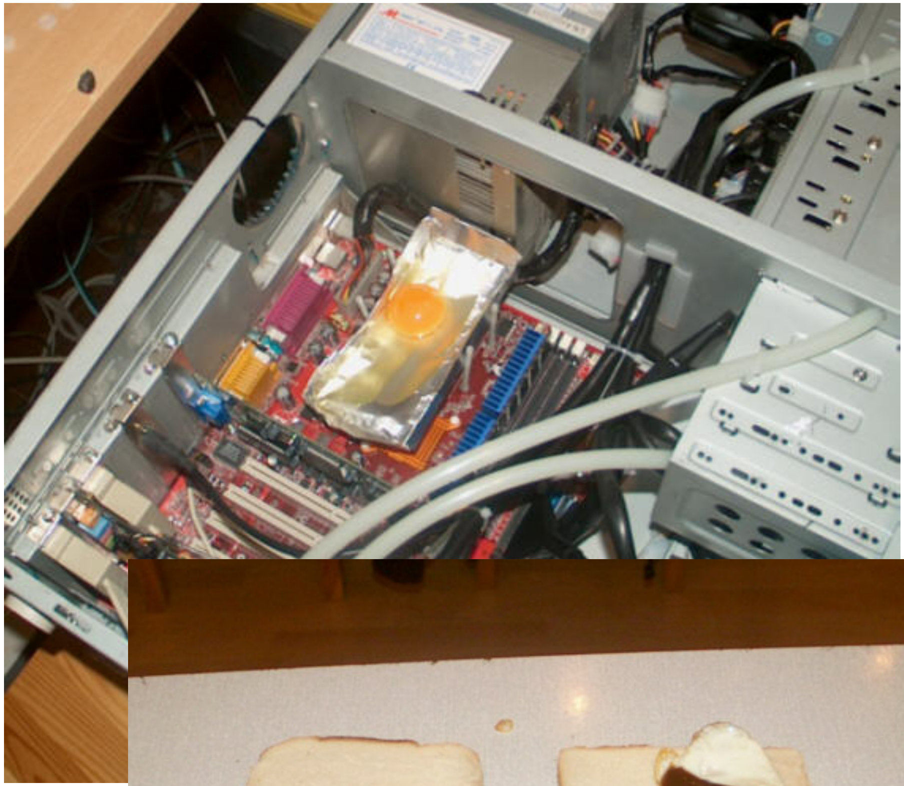


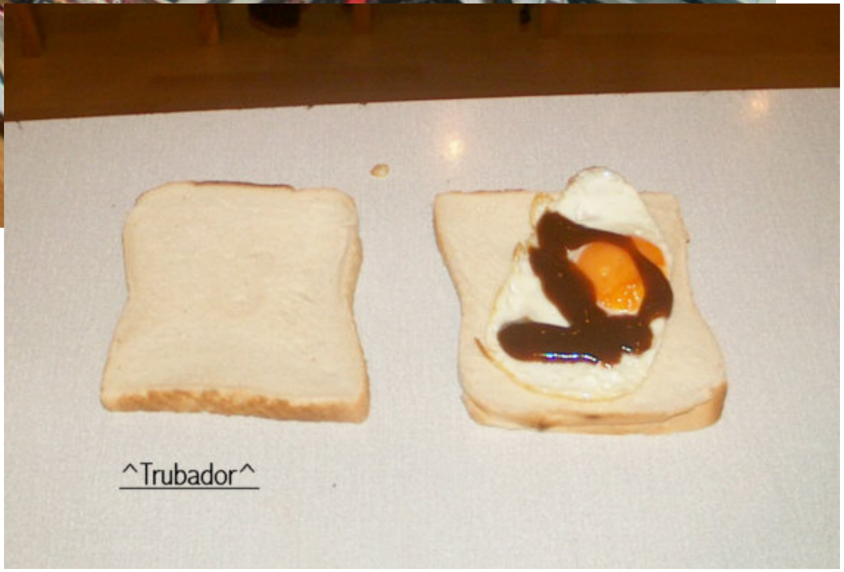
Intel® Pentium® Core 2 Duo
Extreme processor

The hottest chip you can get???

Issues for Modern Computers

- Energy/Power consumption has become a major challenge
  - It is now perhaps the limiting factor in most processors



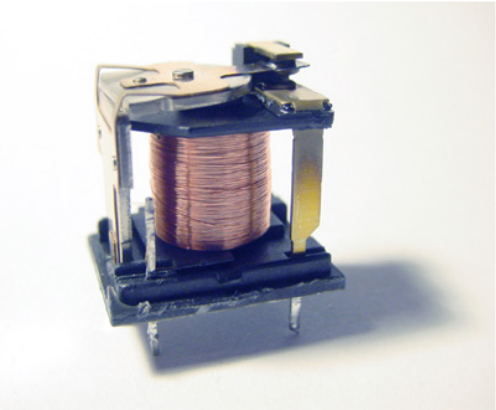


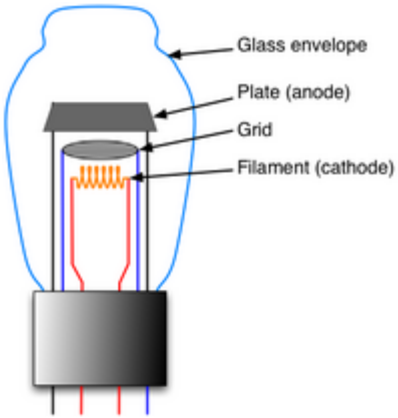
Courtesy Troubador

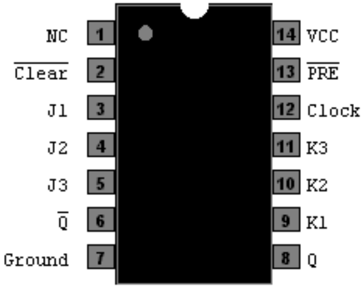
http://www.hotchips.org/

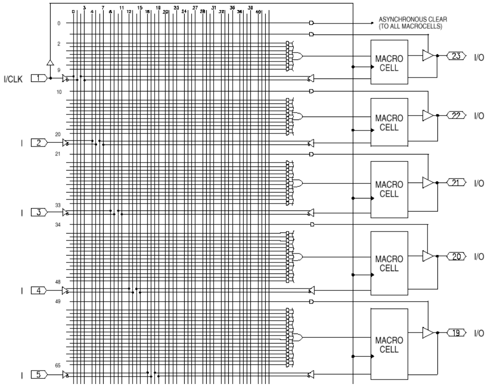
Implementation Technology

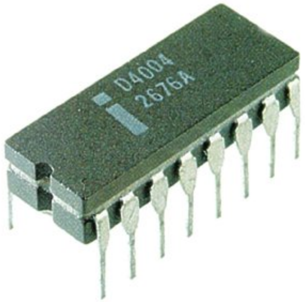
- Relays
- Vacuum Tubes
- Transistors
- Integrated Circuits
  - Gate-level integration
  - Medium Scale Integration (PALs)
  - Large Scale Integration (Processing unit on a
  - Today (Multiple CPUs on a chip)
- Nanotubes??
- Quantum-Effect Devices??





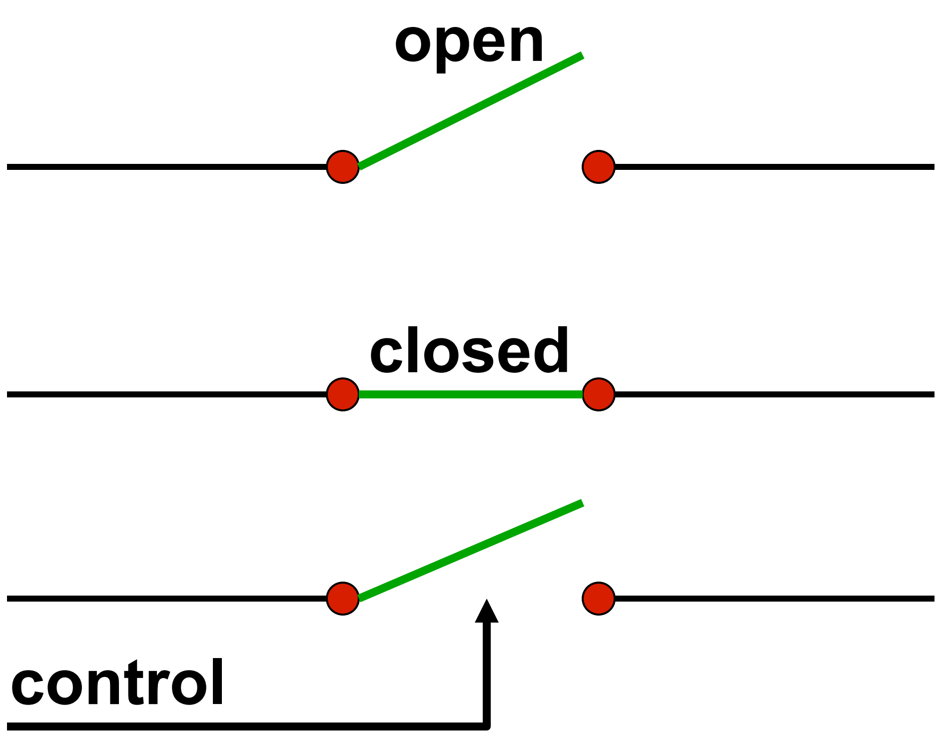






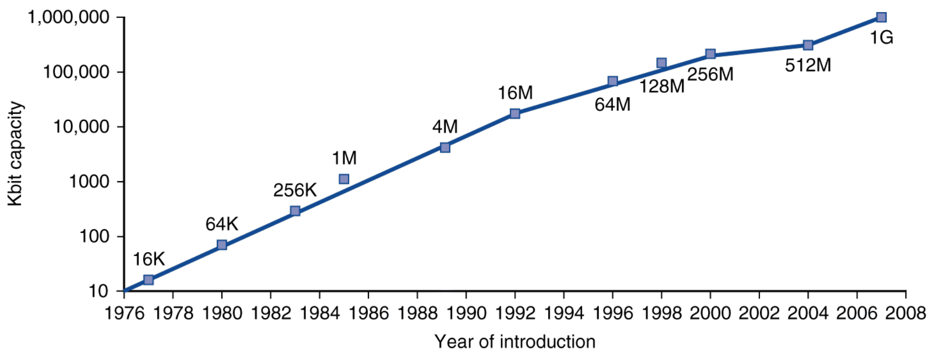
Implementation Technology

- Common Links?
  - A controllable switch
- Computers are wires and switches



Technology Trends

- Electronics technology continues to evolve
  - Increased capacity and performance
  - Reduced cost



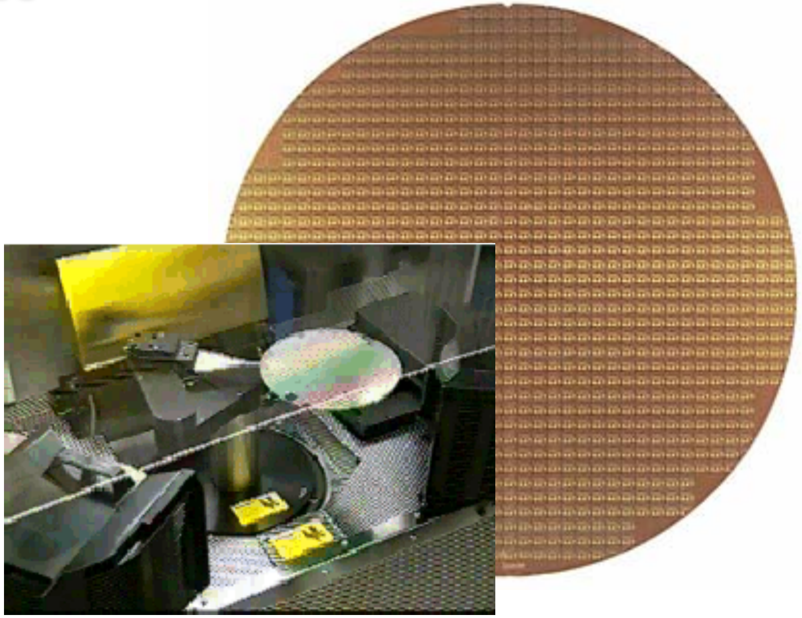
DRAM capacity

Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2005	Ultra large scale IC	6,200,000,000

Chips

✱ Silicon Wafers

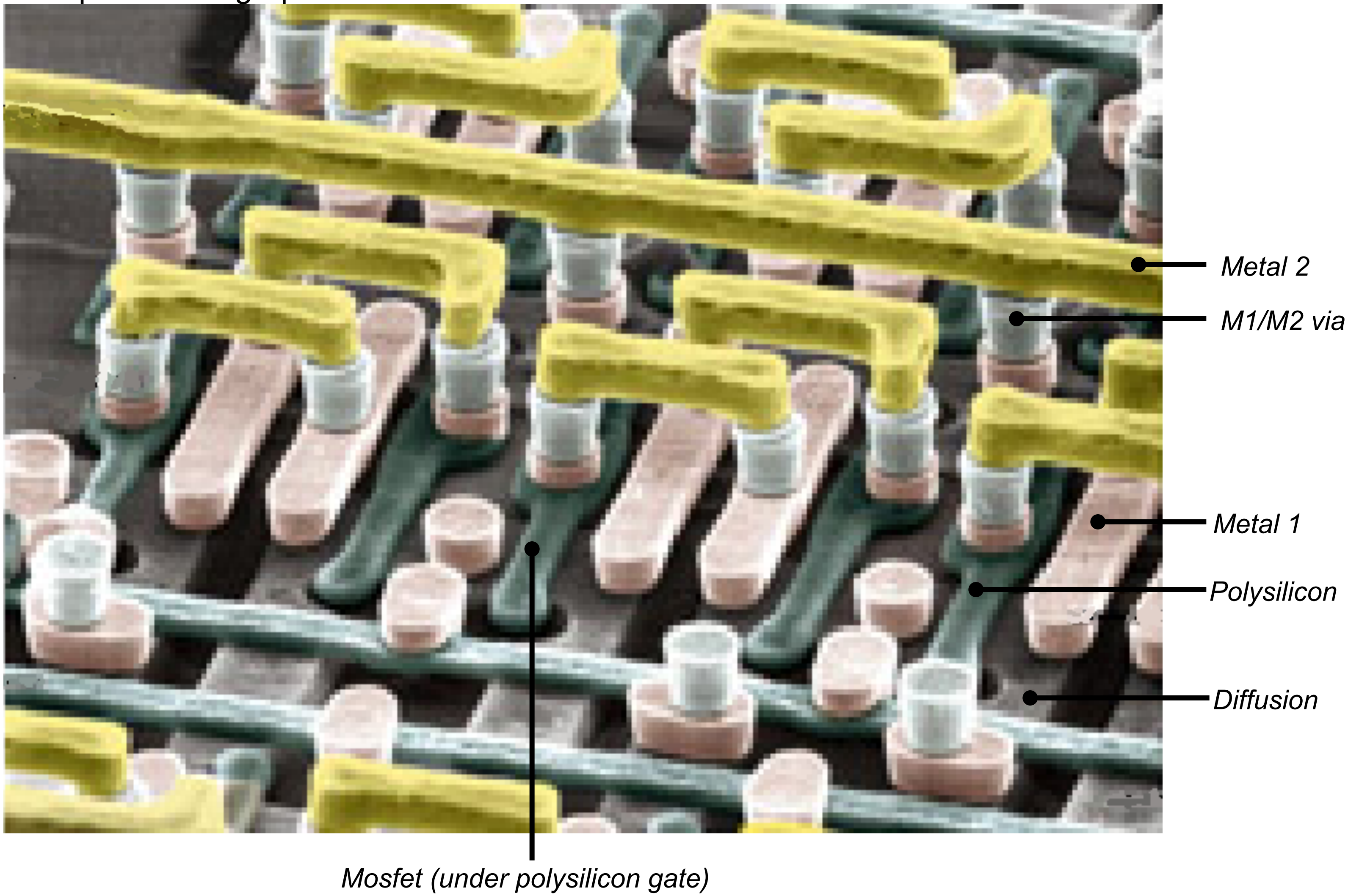
- Chip manufactures build many copies of the same circuit onto a single wafer.
  - ➢ Only a certain percentage of the chips will work; those that work will run at different speeds. The yield decreases as the size of the chips increases and the feature size decreases.
- Wafers are processed by automated fabrication lines.
  - ➢ To minimize the chance of contaminants ruining a process step, great care is taken to maintain a meticulously clean environment.



Chips

⁕ Silicon Wafers

IBM photomicrograph



Next Lecture

- Computer Performance
  - How to measure performance
  - How to compare performance