01:198:211
Computer Architecture

Instructor: Prof. Santosh Nagarakatte
Rutgers University
santosh.nagarakatte@cs.rutgers.edu

Spring 2017

Instructors

Instructor: Prof. Santosh Nagarakatte

santosh.nagarakatte@cs.rutgers.edu

Office Hour: Friday 3:00-5:00PM (CoRE 328)

TAs: (Sections 5-8)

See Sakai for details.

Office hours/ TA details will be available soon on course website.

Course Web

- Sakai will be primary source of course information:
  - Sakai: https://sakai.rutgers.edu
    - Announcements, course material, assignment hand-in, etc.
    - If you are registered, when you login to Sakai using your netid, you should see a tab “Computer Architecture 05 S17”
      - If not, it may be under the “My Active Sites” tabo

Text Books

- Required:
  - Bryant and O'Hallaron. Computer Systems: A Programmer's Perspective. Prentice Hall.

- Recommended:
  - Patterson and Hennessy. Computer Organization and Design: The Hardware/Software Interface. Morgan Kauffman.
  - Kernighan and Ritchie. The C Programming Language, 2nd Edition, Prentice Hall, 1988.
    - Any good C book should be fine
    - A free online C book: http://publications.gbdirect.co.uk/c_book/
      - Disclaimer: I know nothing about the quality of this book

What You Should Know

- Prerequisite: 198:112 ➔
  - You know some math
  - You know a bit about algorithms and data structures
  - You know at least one programming language (Java)
  - You know something about how to write, run, and test programs

- Goal of this course is to make you learn
  - Requires you to actively participate in class
  - Requires you to read the textbook
  - Requires you to start the programming assignments early

How to write programs to attain good performance on modern architectures?

What You Will Learn …

- How to program in two more programming languages (C and Assembly)
- The major hardware components in computer systems
  - Trends in technology and computer architecture
- How hardware components are built from digital logic
- How programs written in a high-level language (e.g., C) is actually executed by the hardware
- How to understand and improve the performance of programs

Course Expectations

- The fun part
  - 5 programming assignments
  - 3 paper & pencil homework assignments
- The not so fun part
  - 2 midterm and 1 final exams
    - All exams are comprehensive (all materials up to the exam)
- What we expect from you
  - Attend lectures and recitations
    - Read the assigned readings before lecture
    - Read and think about the programming and homework assignments
  - Ask questions
  - Start programming assignments early
  - Don't copy or cheat

No Late Assignments

- We will NOT accept late assignments
- Programming assignments to be handed-in on Sakai
  - Deadline will be enforced by Sakai
    - Assignments will not be accepted after deadline passes
    - Can hand-in assignments multiple times so if you are working at the last minute, hand-in a version early (e.g., ½ hour before deadline)
- Homework assignments to be handed-in in lecture
  - I will stop accepting assignments once I leave the lecture room
- Emails with assignments attached will be discarded

Collaboration vs. Cheating

- Collaboration is encouraged
  - You learn by discussing with others
  - BUT, must not cheat…
- Cheating
  - We hate it, we will not tolerate it, we will look for it
  - If we find cheating, *everyone involved will be punished*
    - Department's academic integrity policy: http://www.cs.rutgers.edu/policies/academicintegrity/
  - If you are having trouble with the course for any reason
    - Come talk to me
      - Please observe my "door policy"
    - I will help you if I can
      - Corollary: once cheating has occurred, there's nothing I can do to help you avoid the consequences

Tentative List of Topics

- Introduction
  - Hardware trends
- C programming
- Information representation
- Assembly (x86) programming
- Digital logic
- Processor architecture
  - Pipelining (?)
- Memory hierarchy
- Compiling & linking (?)
- Measuring & optimizing program performance

Programming Assignments

- 5 programming assignments
- Program in C and/or Assembly
  - Don't wait until the last minute
  - Learn how to use tools
    - Don't program/debug "by accident" or "by blind search"
- Will be done using the Instructional Lab
  - https://www.cs.rutgers.edu/resources/instructional-lab
    - If you don't already have an account, get one asap
      - https://www.cs.rutgers.edu/resources/getting-started-with-technical-resources-at-the-department-of-computer-science
  - You will be programming in a Linux environment

Grading

- Grading:
  - 25%: 2 midterm exams
  - 30%: final exam
  - 5%: class participation
  - 10%: homework assignments
  - 30%: programming assignments
- All exams are cumulative

- No make-up exams except for university sanctioned reasons.
  - Must inform professor before the exam
  - Informing professor ≠ ok to take make-up

Iclicker Question: Class Demography

- Are you a ?
  - **A: Freshman**
  - **B: Sophomore**
  - **C: Junior**
  - **D: Senior**

Iclicker Question: Programming

- Are you a good programmer?
  - **A: Written less than 100 lines of code**
  - **B: Written between 100-300 lines of code**
  - **C: Written more than 1000 lines of code**
  - **D: Written more than 10,000 lines of code**

Iclicker Question: Linux and C

- Do you know Linux and C?
  - **A: I have not heard of it**
  - **B: I have heard about it but I don't know anything**
  - **C: I have written programs in C and used a Linux machine.**
  - **D: I have written a lot of programs in C and used Linux**

Iclicker Question:

- You are given a pointer to the beginning of the linked list written by your friend. The list can have a large number of nodes. Can you write an algorithm to check if the list has a cycle?

- A: Cannot be done

- B: Cannot be done without storing all the nodes

- C: Can be done with constant amount of storage

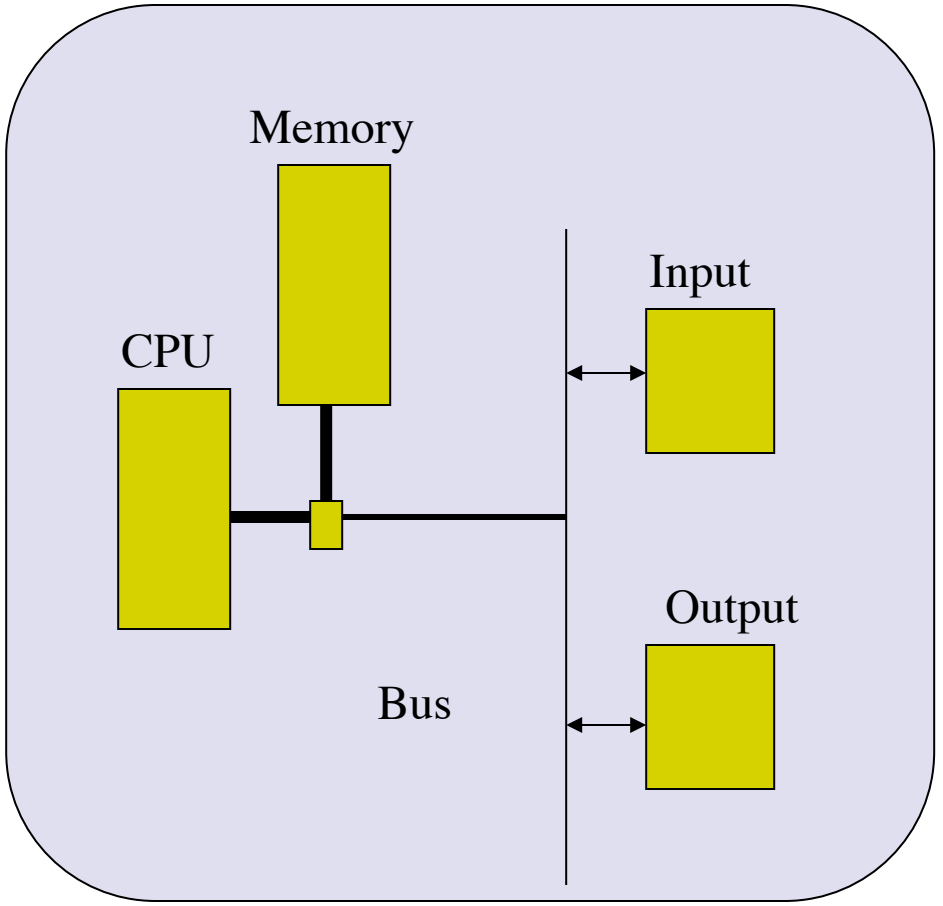
- D: No solution for this problem

Computer Architecture

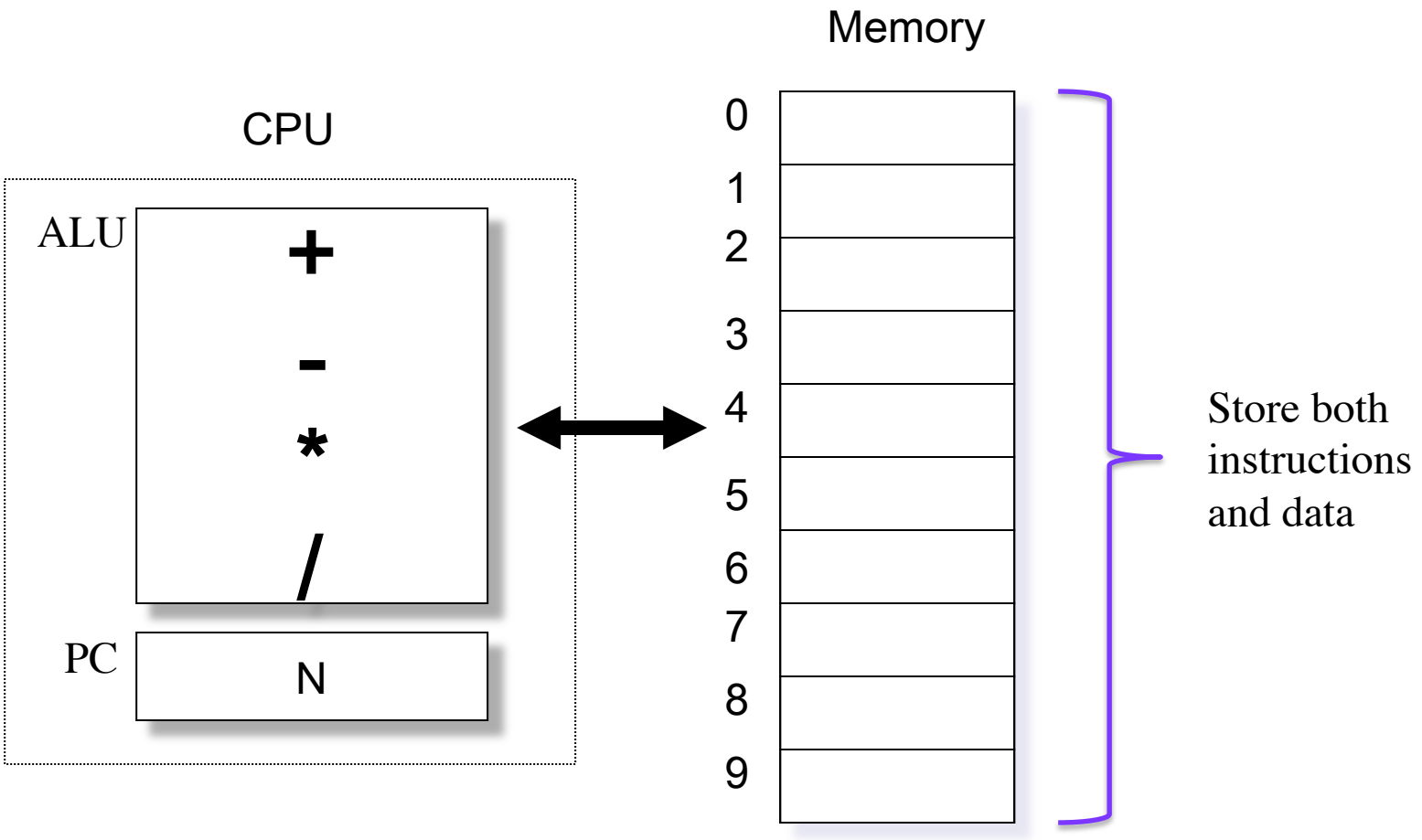
?

Main Components

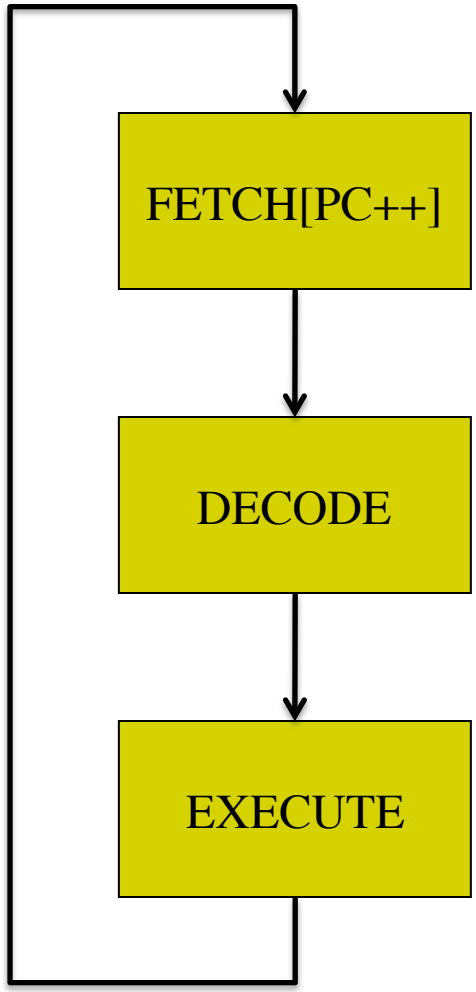
- CPU
- Memory
- Bus
- I/O devices
  - Human Interface
    - Mouse, keyboard, screen etc.
  - Storage
  - Networking
  - Graphics



Von Neumann Model

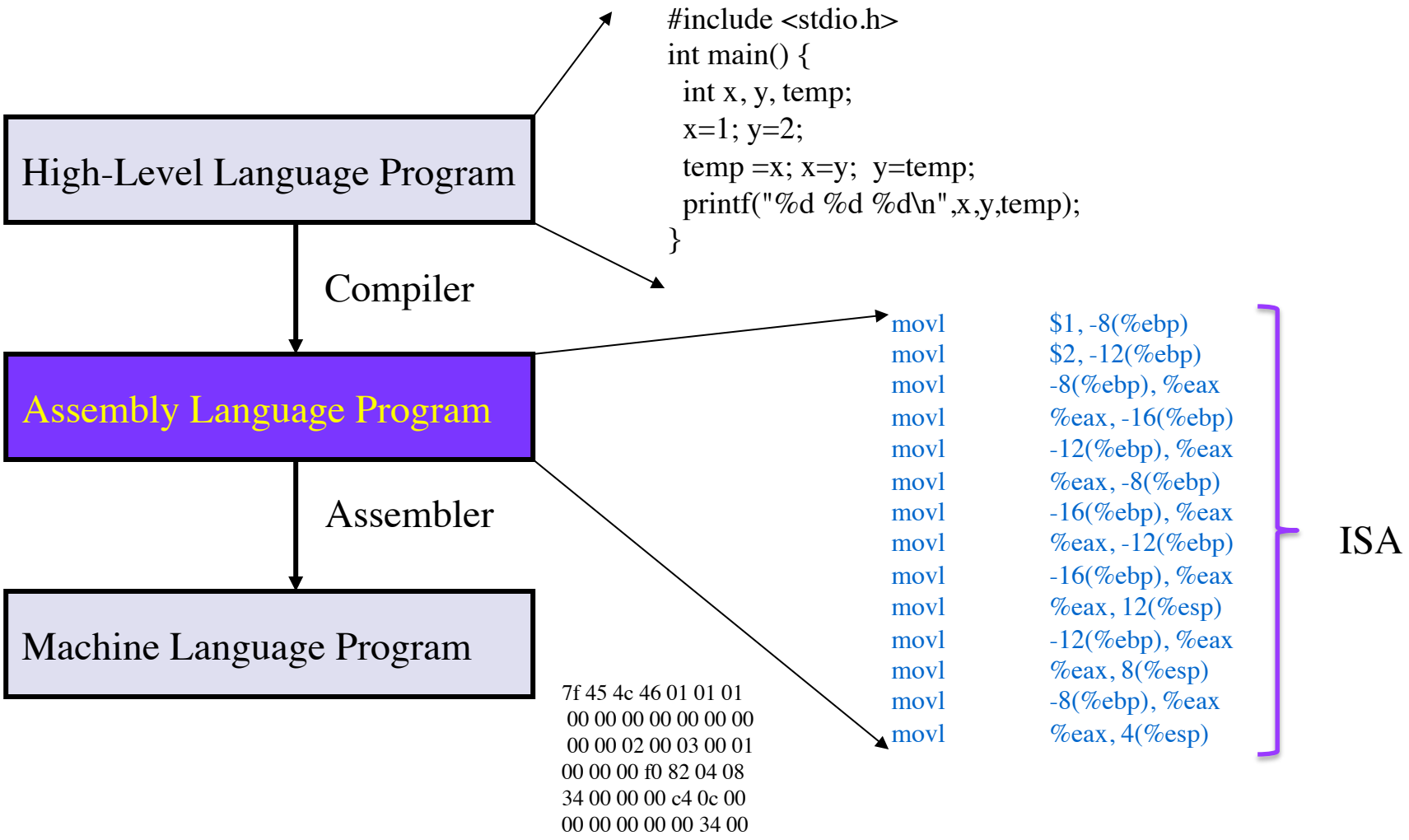


Basic CPU Function

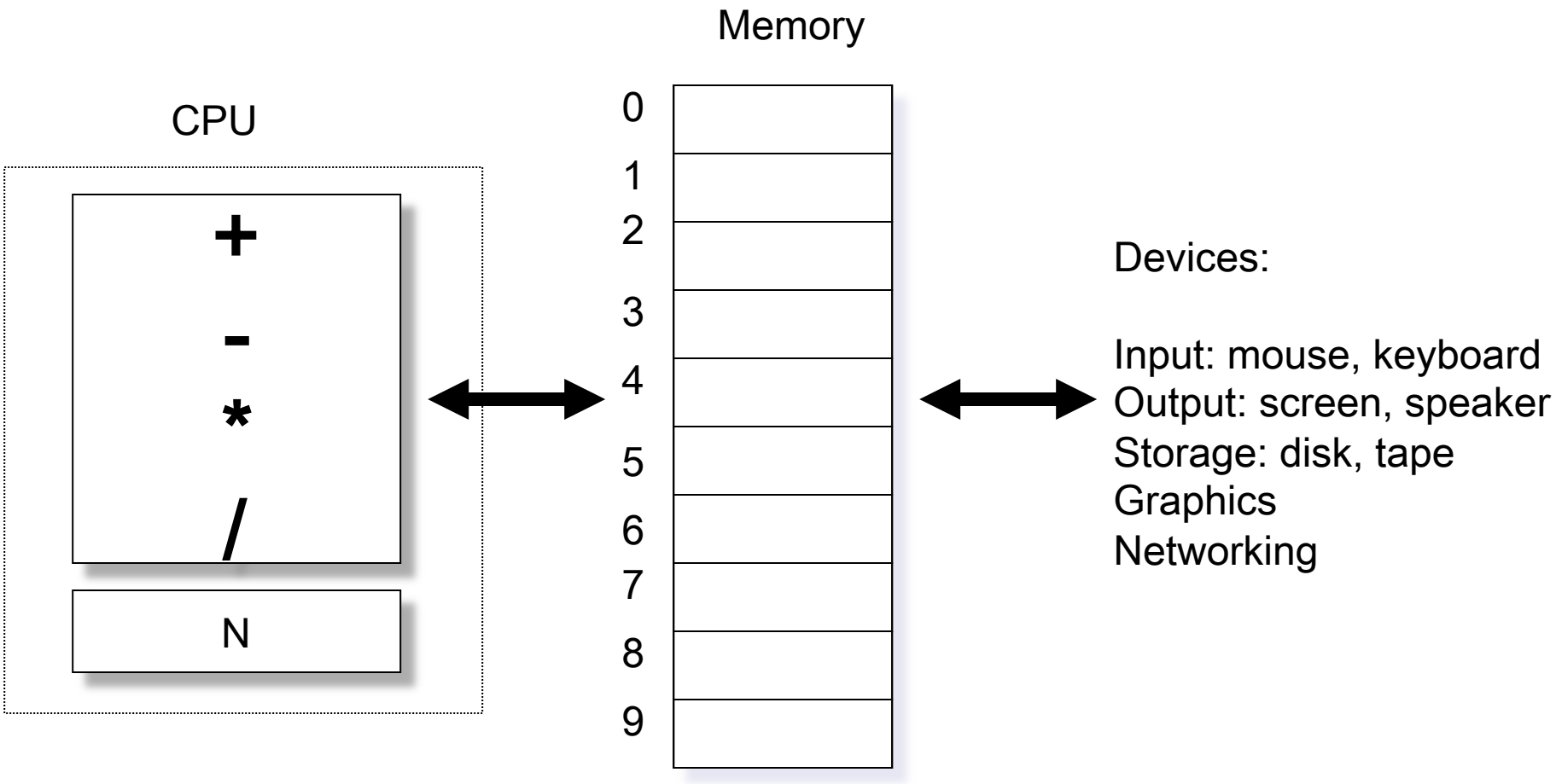


Arithmetic: +, -, *, /
Logic: bre, jmp

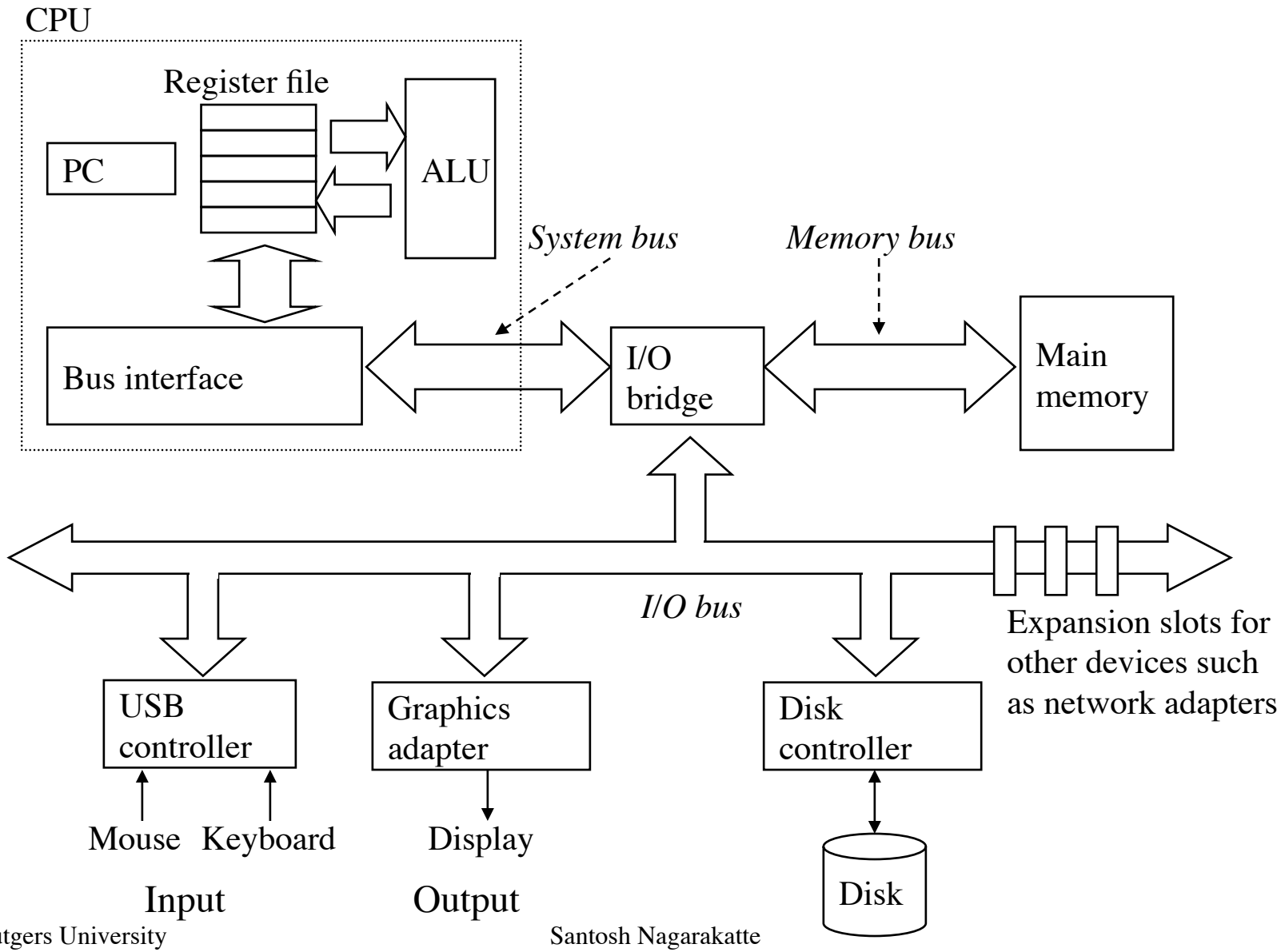
Programming Meets Hardware



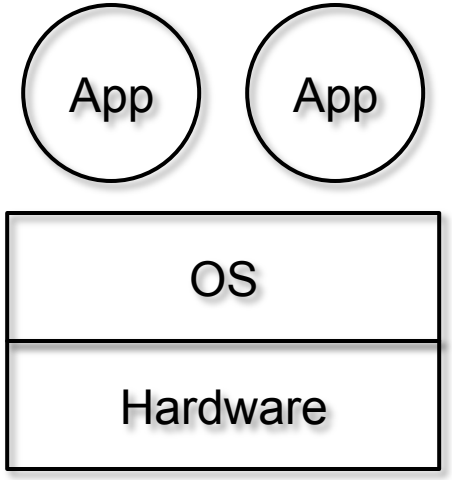
Input/Output

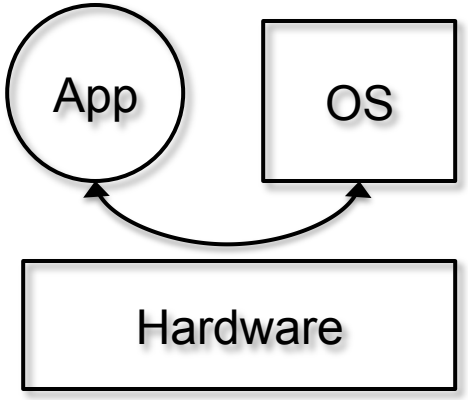


Von Neumann in Practice



Where Is the OS?





Does this mean apps run on OS rather than the hardware?

Apps typically runs directly on hardware. Will switch over to OS when need help.

Actually, 1st picture is used frequently because OS does extend the hardware, making the hardware appear more "powerful" through abstractions such as processes, threads, virtual memory, files, etc.

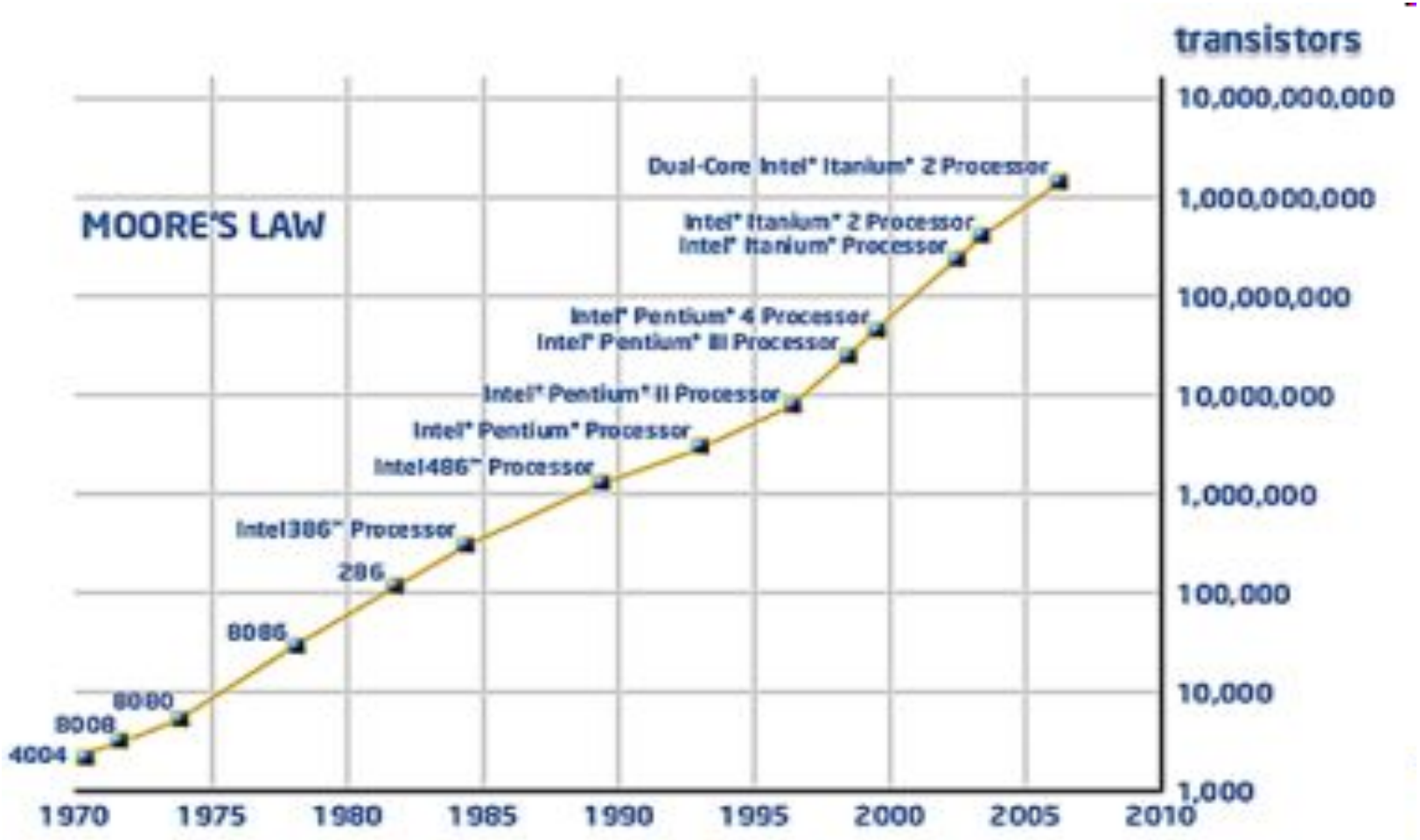
Computer Architecture

- How to design and build the components
- How to design and build systems from the components
- In this class:
  - Understand basics of current components and systems
  - Understand how programs run on current systems
  - Understand how current architecture affect my high-level language programs
    - How can I make my program run faster?
    - How can I make my program more power efficient?
      - Run longer when on battery

Architecture Trends: Moore's law

- Gordon Moore was an Intel Engineer
- An observation about improvements in hardware
- No of transistors on a chip double every 18 months
- Exponential growth seen in other hardware dimensions
  - Processor speed: 2x every 18 months
  - Memory capacity: 2x every 2 years
  - Disk capacity: 2x every year

Number of Transistors on a Chip

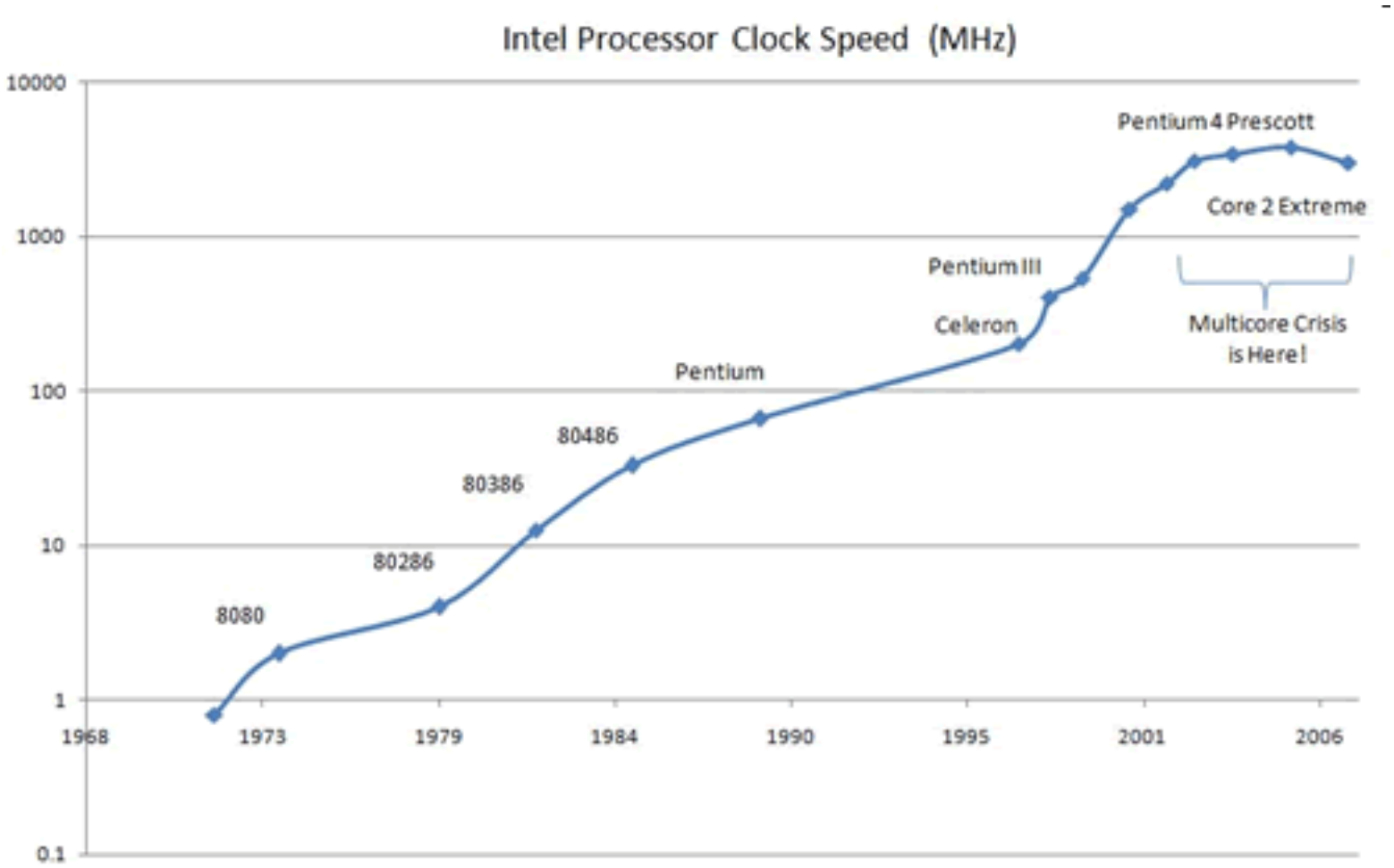


Any guess on the number of transistors in the latest Intel chip in your laptops?

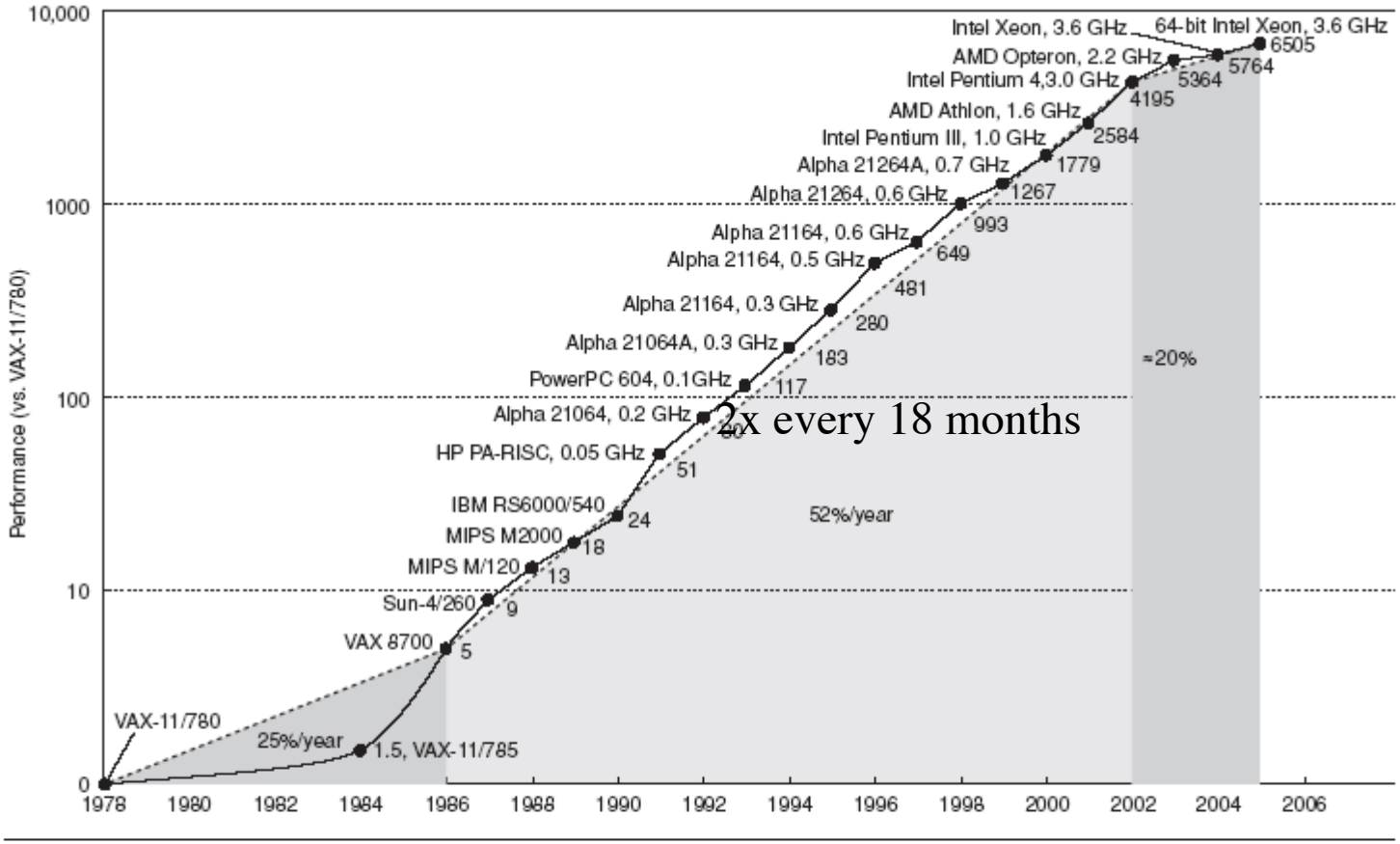
Any guess on the number of transistors in the latest Intel chip in your laptops?

Intel Broadwell Xeon chip has 7.2billion transistors!

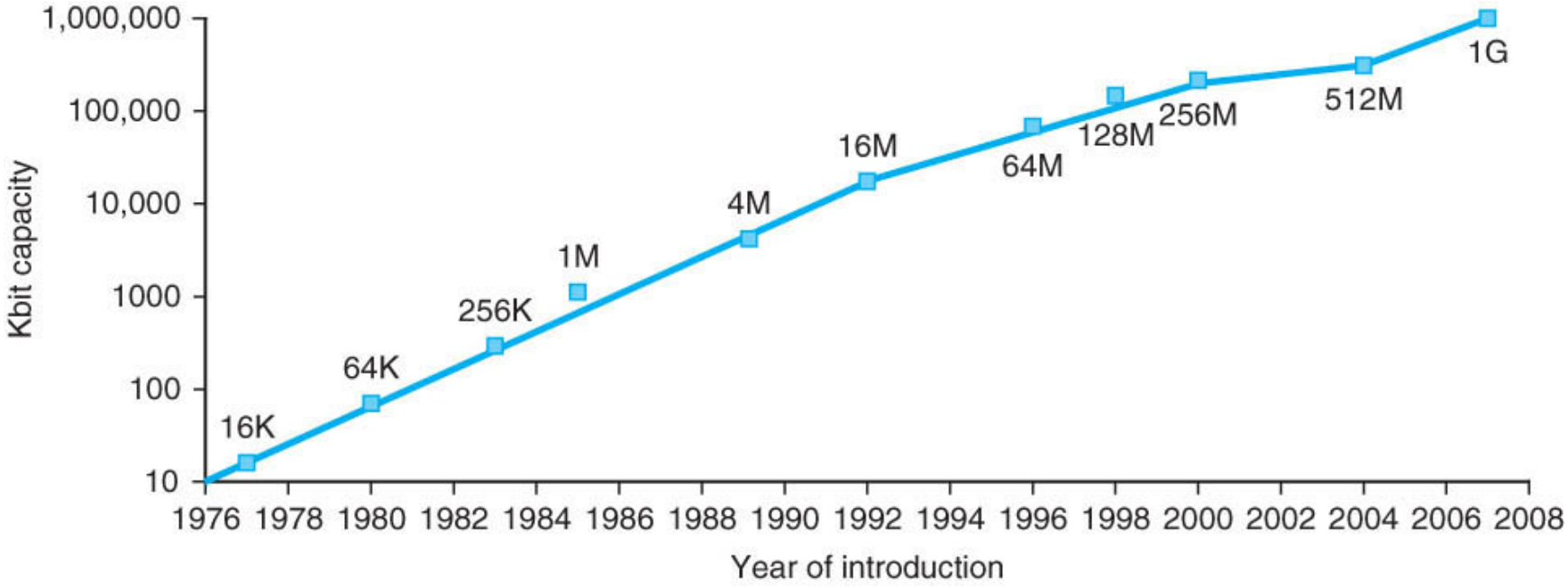
Clock speed


Intel Processor Clock Speed (MHz)
10000
1000
100
10
1
0.1
1968
1973
1979
1984
1990
1995
2001
2006
8080
80286
80386
80486
Pentium
Celeron
Pentium III
Pentium 4 Prescott
Core 2 Extreme
Multicore Crisis is Here!

Processor Performance

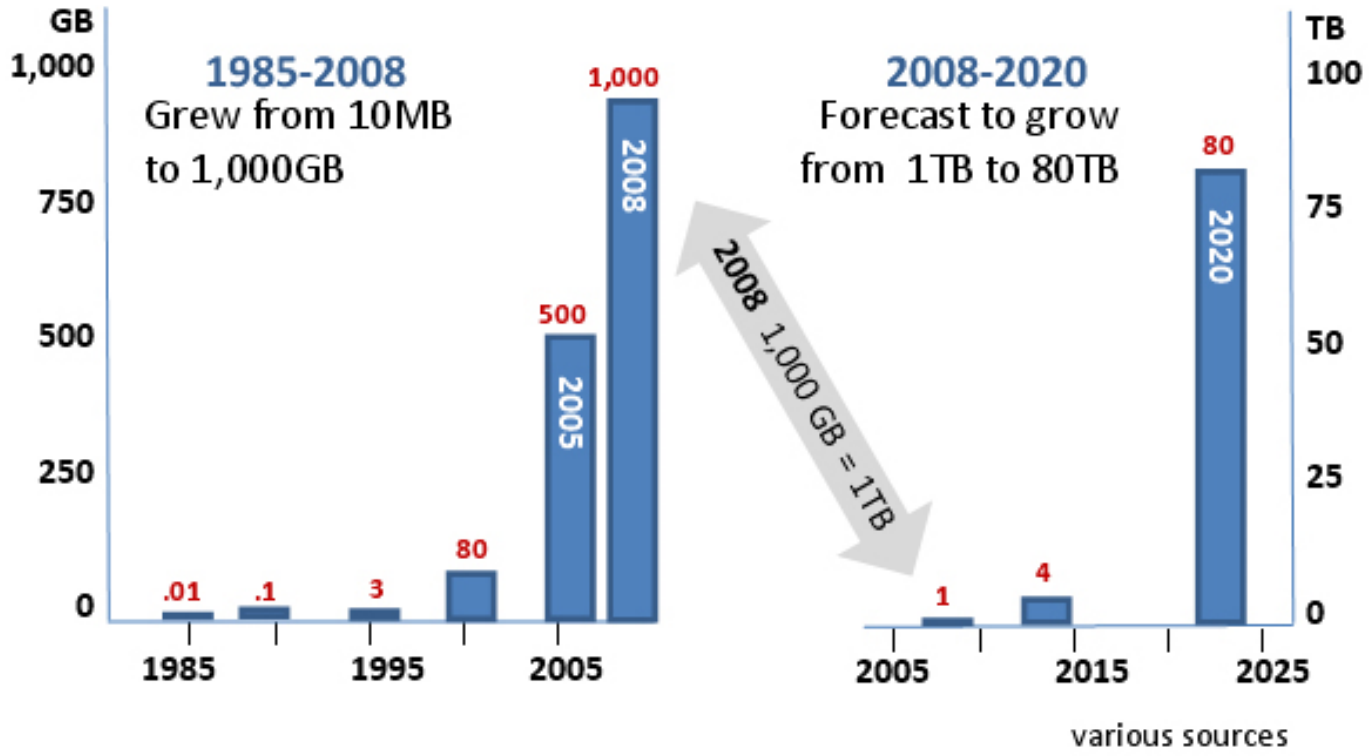


Memory Capacity

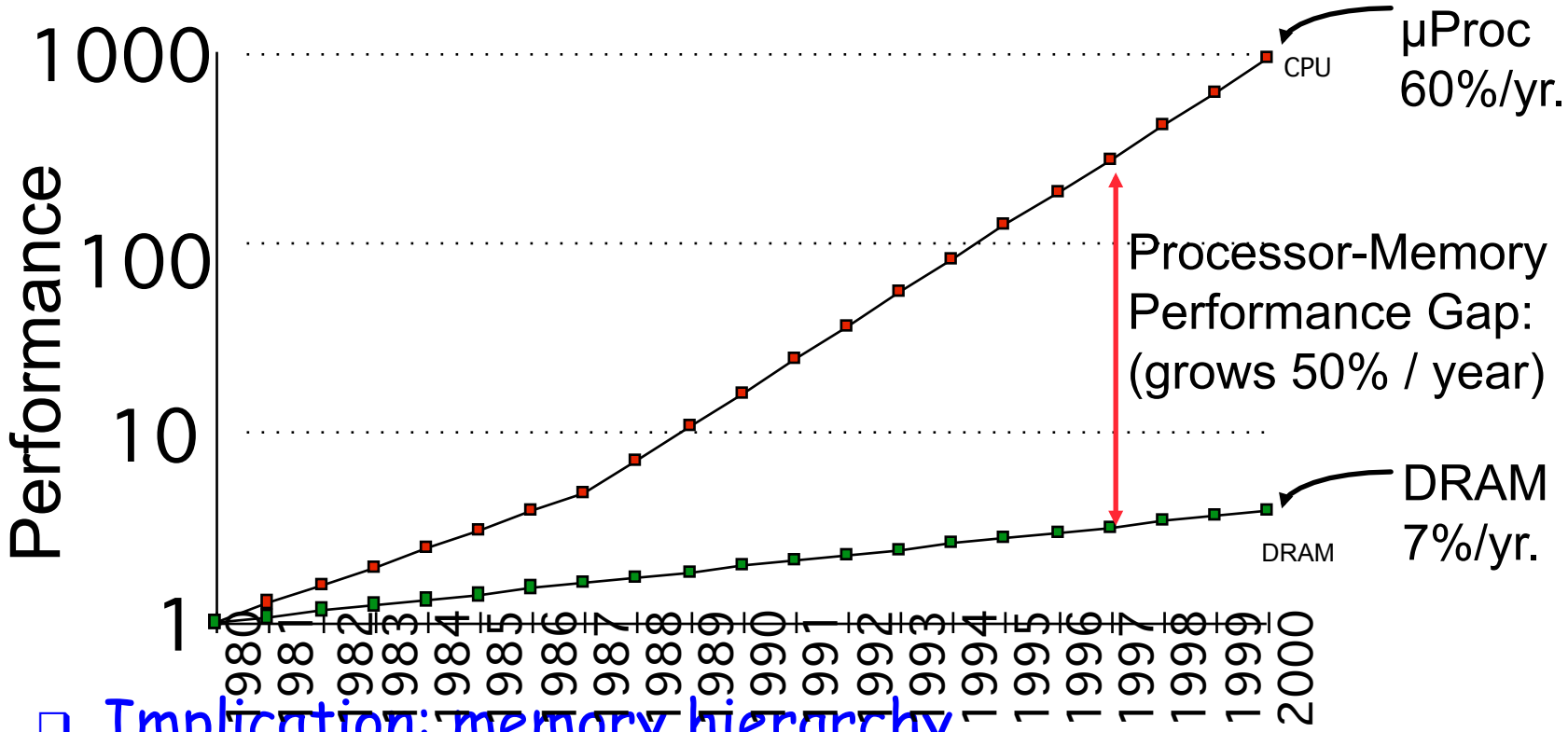


Now laptops have 16GB RAM

Disk Capacity

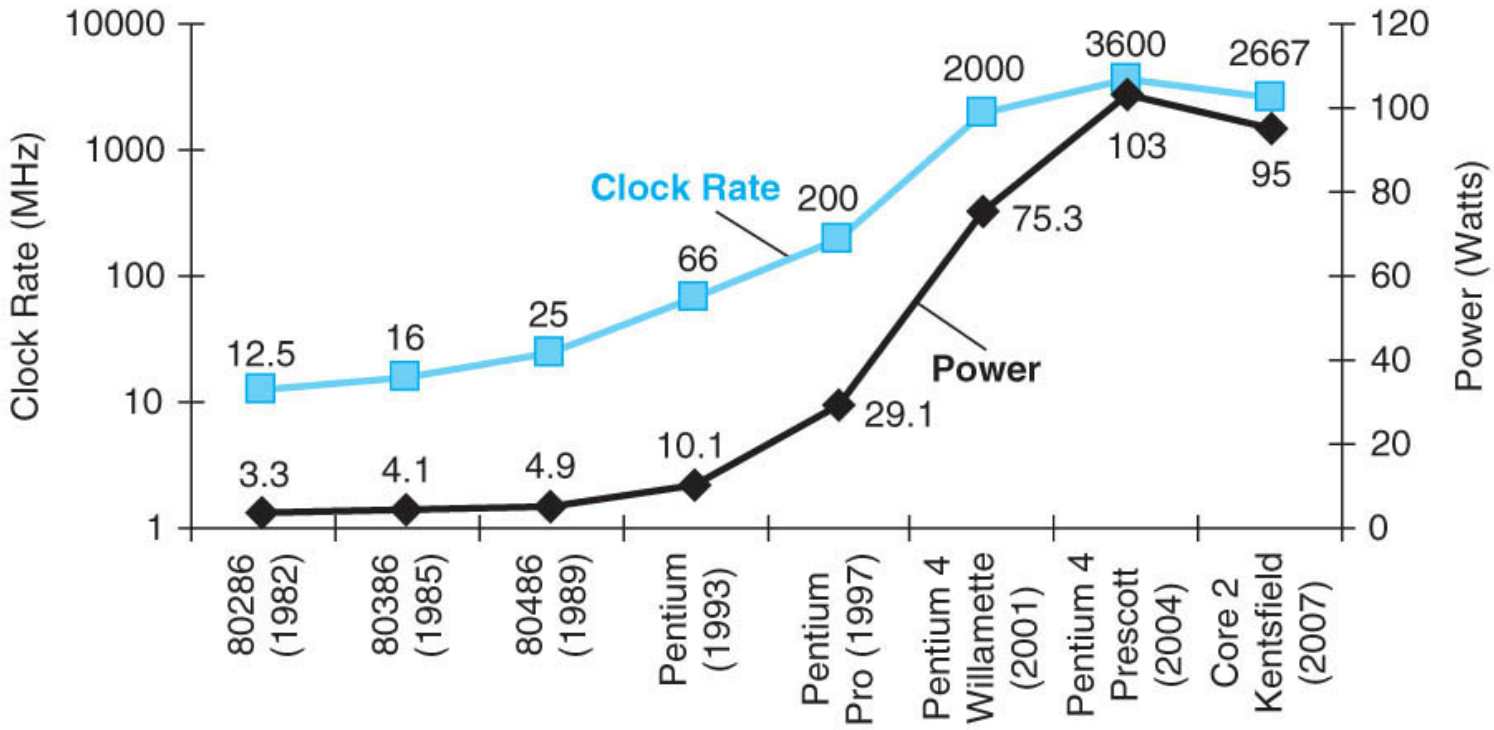


CPU/Memory Performance Gap



- Implication: memory hierarchy
  - registers, caches

"Power Wall"


Clock Rate (MHz)
10000
1000
100
10
1
Power (Watts)
120
100
80
60
40
20
0
Clock Rate
Power
12.5
16
25
66
200
2000
3600
2667
3.3
4.1
4.9
10.1
29.1
75.3
103
95
80286 (1982)
80386 (1985)
80486 (1989)
Pentium (1993)
Pentium Pro (1997)
Pentium 4 Willamette (2001)
Pentium 4 Prescott (2004)
Core 2 Kentsfield (2007)

Summary

Today's systems are complex with various components

Faster processors and systems

- Enables new domains of applications
- Imagine the running your favorite game on a Pentium machine.

Understanding the systems crucial

- To program these systems
- To make pragmatic performance/power/energy tradeoffs
- Address various walls --- memory wall, power wall, etc