EE 424
DIGITAL SYSTEM ARCHITECTURE

School of Electrical Engineering and Computer Science

WASHINGTON STATE UNIVERSITY

Course Objectives

Students in this course will be able to:

- Understand how modern computer systems work.
- Perform quantitative analysis of computer systems.
- Analyze at system level the impact of changes in the computer systems.
- Estimate the performance of a computer system.
- Design novel schemes that improves the performance of computer systems
- Use tools to design modern systems.
- Recognize the need for further learning in this field (life-long learning).

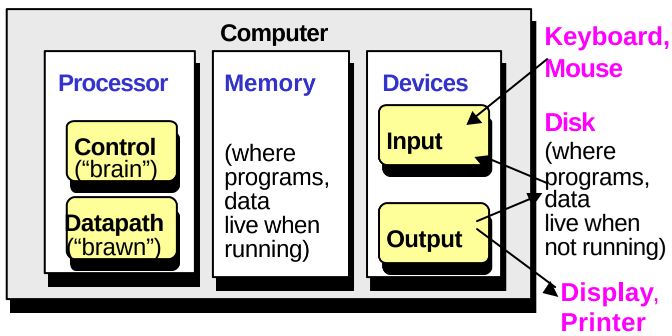
Things you'll be learning

- How computers work, a basic foundation
- Classic/basic components of a computer.
- Stored program concept: instructions and data
- Issues affecting modern processors (caches, pipelines)
- Principles of locality to be exploited by means of memory hierarchy (L1, L2 & L3 cache; main memory; disk,…).
- Greater performance by means of instruction level parallelism
- Principle of abstraction, used to build systems as layers.
- Compilation vs. interpretation thru system layers.
- How to analyze their performance (or how not to!)
- Principles and pitfalls of performance measurement.

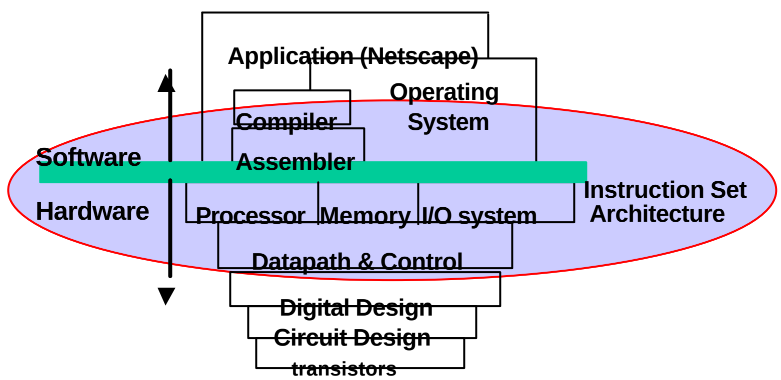
Focus

- Our primary focus: the processor (datapath and control)
  – Implemented using millions of transistors
  – Impossible to understand by looking at each transistor
  – We need to:
    - Have an overall picture of the system.
    - Analyze how the components interact.
    - Consider both Hardware and Software.

Basic components



Computer System

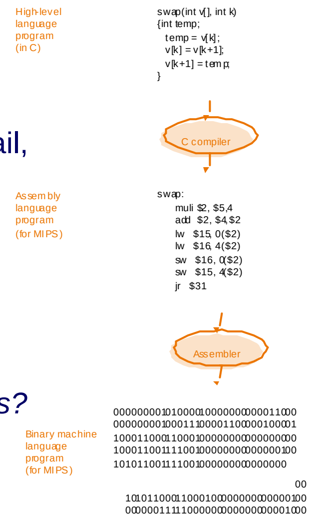


Abstraction

- Delving into the depths reveals more information

- An abstraction omits unneeded detail, helps us cope with complexity

What are some of the details that appear in these familiar abstractions?



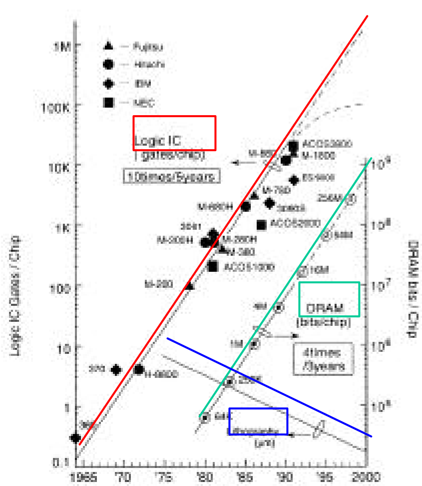
Computer history

Generation -1:	The early days	????-1642
Generation 0:	Mechanical	1642-1935
Generation 1:	Electromechanical	1935-1945
Generation 2:	Vacuum tubes	1945-1955
Generation 3:	Discrete transistors	1955-1965
Generation 4:	Integrated circuits	1965-1980
Generation 5:	VLSI	1980-????

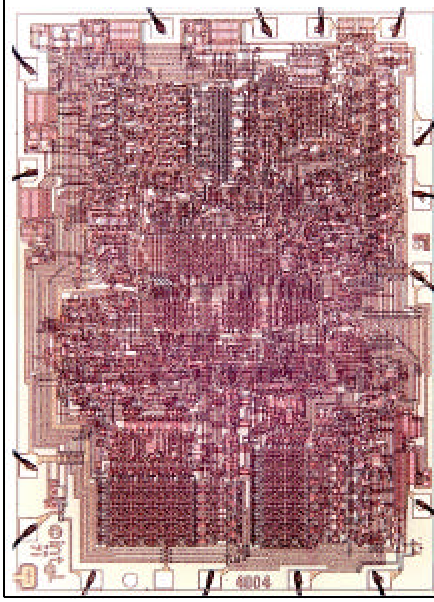
Integrated Circuits

- Rapidly changing field:
  - – vacuum tube → transistor → IC → VLSI (see section 1.4)
  - – doubling every 1.5 years:
    memory capacity
    processor speed
    - *(Due to advances in technology <u>and</u> organization)*

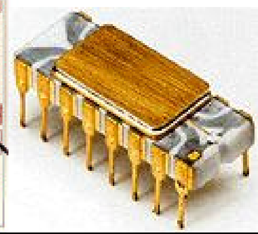
Complexity



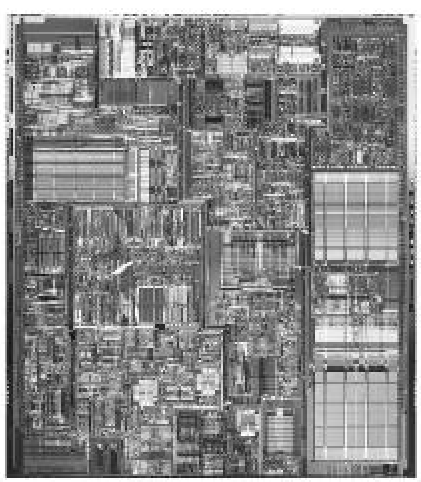
Intel 4004



- In 1971, Ted Hoff produced the Intel 4004 in response to the request from a Japanese company (Busicom) to create a chip for a calculator
- It is the first microprocessor, i.e. the first processor-on-a-chip



Intel Pentium 4



Fall 2002

- 2.80 GHz
- 0.13-micron technology
- 478-pin package
- 512 KB L2 Cache
- 50 Amps
- Vcc = 1.5V

Sony Playstation 2000

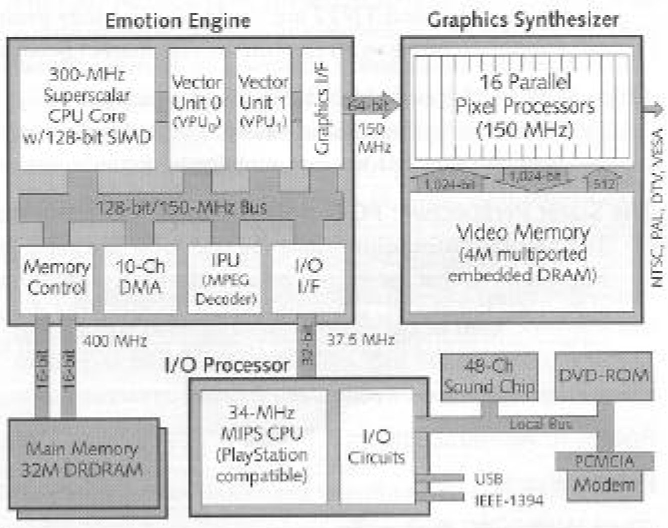


Figure 1. PlayStation 2000 employs an unprecedented level of parallelism to achieve workstation-class 3D performance.

Where we are heading

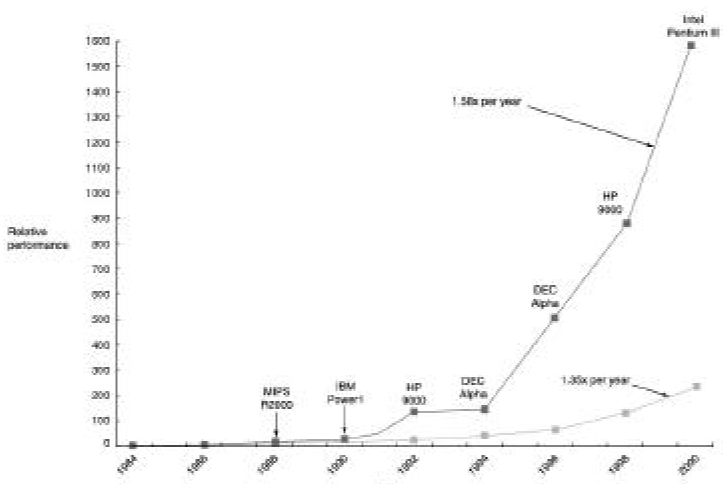
- Performance issues (Chapter 2)
  - *Terminology and motivation*
- A specific instruction set architecture (Chapter 3)
- Arithmetic and how to build an ALU (Chapter 4)
- Constructing a processor to execute our instructions (Chapter 5)
- Pipelining to improve performance (Chapter 6)
- Memory: caches and virtual memory (Chapter 7)

Key to a good grade: read your textbook
- Attend the lectures (and pay attention!)
- Work on all assignments.

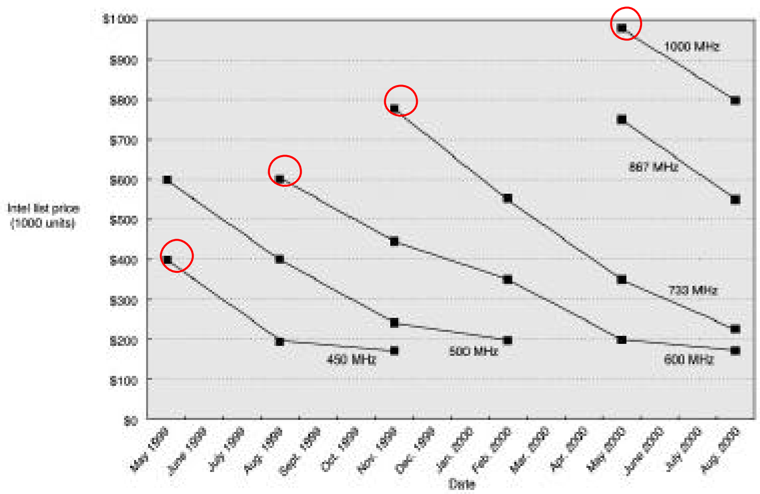
Why Such Change in 12 years?

- Performance
  - Technology Advances
    - CMOS VLSI dominates older technologies (TTL, ECL) in cost *AND* performance
  - Computer architecture advances improves low-end
    - RISC, superscalar, RAID, …
- Price: Lower costs due to …
  - Simpler development
    - CMOS VLSI: smaller systems, fewer components
  - Higher volumes
    - CMOS VLSI: same dev. cost 10,000 vs. 10,000,000 units
  - Lower margins by class of computer, due to fewer services
- Function
  - Rise of networking/local interconnection technology

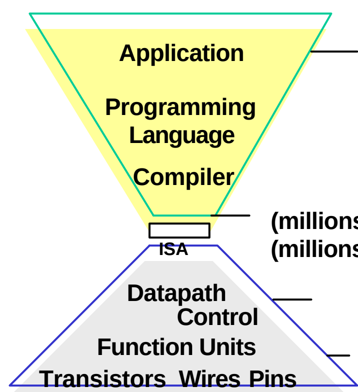
Performance



Cost



Metrics of Performance



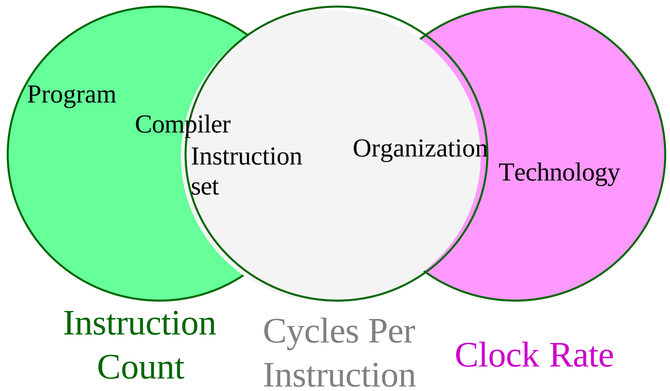
Answers per month
Operations per second

(millions) of Instructions per second: MIPS
(millions) of (FP) operations per second: MFLOP/s

Megabytes per second

Cycles per second (clock rate)

CPU time



Computer Architecture

Computer Architecture =

Instruction Set Architecture +

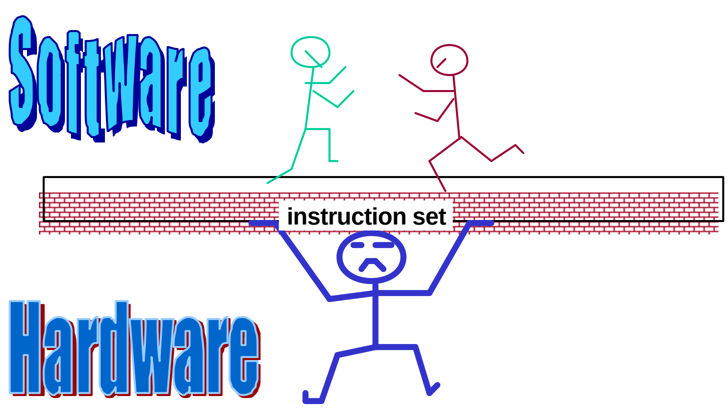
Machine Organization + …

Computer Architecture's Changing Definition

- 1950s to 1960s: Computer Architecture Course Computer Arithmetic
- 1970s to mid 1980s: Computer Architecture Course Instruction Set Design, especially ISA appropriate for compilers
- 1990s: Computer Architecture Course Design of CPU, memory system, I/O system, Multiprocessors
- 2000s: Computer Architecture Course

  Design of CPU (Performance & Power), Memory system, I/O, embedded systems, wireless.

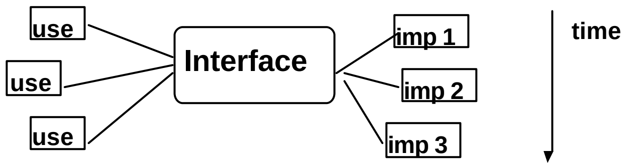
Instruction Set Architecture (ISA)



Interface Design (ISA)

A good interface:

- Lasts through many implementations (portability, compatibility)
- Is used in many different ways (generality)
- Provides convenient functionality to higher levels
- Permits an efficient implementation at lower levels



Evolution of Instruction Sets

Single Accumulator (EDSAC 1950)

Accumulator + Index Registers
(Manchester Mark I, IBM 700 series 1953)

Separation of Programming Model
from Implementation

High-level Language Based
(B5000 1963)

Concept of a Family
(IBM 360 1964)

General Purpose Register Machines

Complex Instruction Sets
(Vax, Intel 432 1977-80)

Load/Store Architecture
(CDC 6600, Cray 1 1963-76)

RISC
(Mips,Sparc,HP-PA,IBM RS6000, . . .1987)