

Computer Architecture Introduction

Dr. Arjan Duresi
Louisiana State University
Baton Rouge, LA 70810
Duresi@Csc.LSU.Edu


These slides are available at:

http://www.csc.lsu.edu/~duresi/CSC7080_05/



Overview

- How
- What
- When
- Why



Overview

- ❑ How am I going to **grade** you?
- ❑ What are **we** going to cover?
- ❑ When are **you** going to do it?
- ❑ Why **you** should take this course?

Grading

- ❑ Learning-centered course:
 - The first priority: Maximize learning
 - Your grade will depend on how much you have learned
- ❑ 3 Quizzes (2 best out of three) 35%
- ❑ Activity in the class (involvement in discussing the papers) (15%)
- ❑ Homework (20%)
- ❑ Project (30%).

Frequently Asked Questions

- Yes, I do use “curve”. Your grade depends upon the performance of the rest of the class.
- All homeworks are due at the beginning of the next class.
- All late submissions must be preapproved.
- Everyone including the graduating seniors are graded the same way.

Homework

- Reading will be assigned for each week.
- Before lecture, every student must submit a one page report of the assigned papers (report should contain a one paragraph summary of the paper, description of three strong points of the paper and three weak points of the paper).
- The reports are due one hour before the class starts (DUE TIME: 12:30 PM), by email to durrezi@byte.csc.lsu.edu, include 7080 - HW# in the subject.
- If more than one paper was assigned, you have to submit a report only on one of the papers. **IMPORTANT:** Submit your homework in PDF format. Include in the title: HW#, your name.

Project

- ❑ Every student must complete a project on one of the topics discussed in the class.
- ❑ Students are required to work in teams of TWO on the project.
- ❑ In addition to the presentation given in the class every team will meet with me to discuss the accomplished results and asses the contribution of each team member.
- ❑ Every project must have a practical component that will require you to do an implementation and demonstration.

Project proposal (2-3 pages), due on February 8

- ❑ Should include:
- ❑ Problem you address.
- ❑ What is your approach.
- ❑ Milestones (main steps and when and how you plan to address them)
- ❑ References: additional reading that you intend to do
- ❑ Tools: if you plan to use tools (software already available), specify if you already have experience with it or you will need first to get to know how to use it.
- ❑ What will be the deliverables: implementation, simulation results, etc,
- ❑ What are the points that if achieved, you will consider that the project was successful.

Project progress (1-2 pages), due March 18

- Should relate to the project proposal:
- What points from the milestones in project proposal were finished.
- What are the main challenges so far.
- Describe if you are stuck in solving a problem (technical or research).
- Sometimes things do not work the way you intended, specify all the modifications from the original proposal, and why were they necessary.

Project final report (10-15 pages), due 1 day before your demonstration of the project

- Should include:
- Problem addressed
- Proposed solution; In case of a system, describe and motivate the chosen architecture, design. If any new algorithm/protocol is designed, include description of the algorithm.
- In case of comparison, simulations, include results.
- What was your personal lessons learnt from the project

Possible Projects

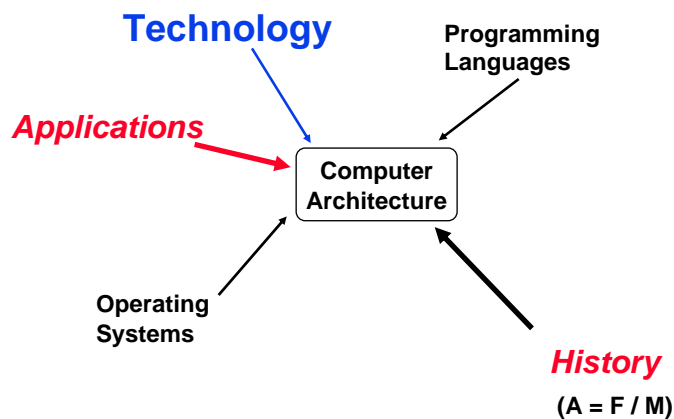
Office Hours

- ❑ Tuesday and Thursday: 3:00 to 4:00 PM
and by appointments
- ❑ Office: 291 Coates Hall
- ❑ Telephone: (225)-578-3902
- ❑ Email: durresi@csc.lsu.edu
- ❑ Course web page:
http://www.csc.lsu.edu/~durresi/CSC7080_05
- ❑ GTA:

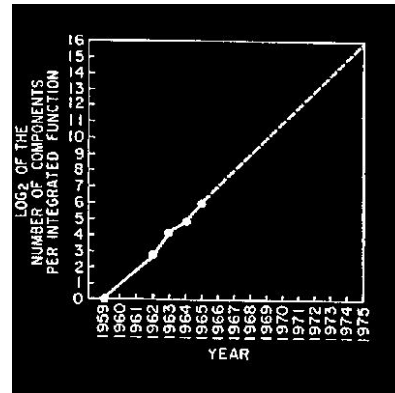
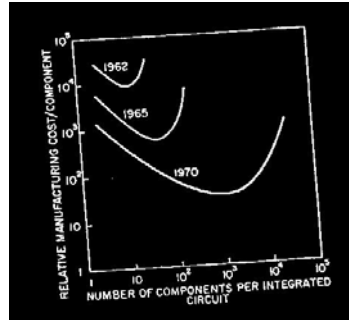
Course Objectives

- ❑ To understand and appreciate the principles and tradeoffs (cost/performance, speed/ flexibility) behind the design of modern computer systems in a qualitative and quantitative fashion;
- ❑ To understand issues in choosing and designing an instruction set architecture;
- ❑ To learn the concepts of basic and advanced pipelining techniques and to understand issues related to exploitations of instruction-level parallelism;
- ❑ To learn issues related to hierarchical memory system design, including cache performance and design issues, as well as main memory design issues;

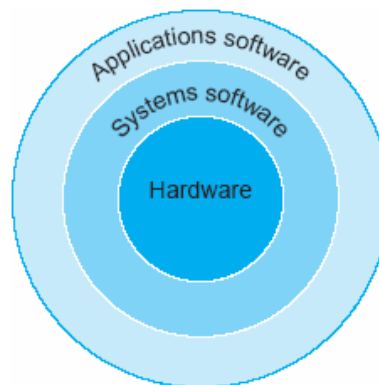
Forces on Computer Architecture



Amazing Underlying Technology Change



Below Your Program



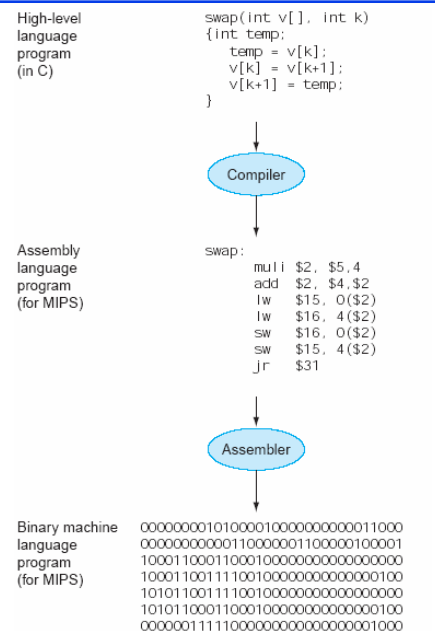
Operating System

- ❑ Handling basic input and output operations
- ❑ Allocating storage and memory
- ❑ Providing for sharing the computer among multiple applications using it simultaneously

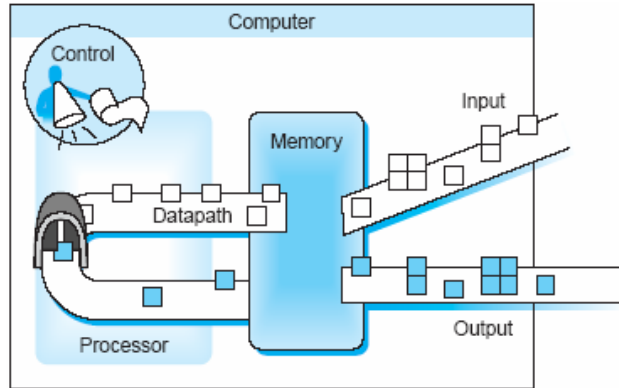
Below Your Program

High level languages:

- 1) Allow to think in a more natural way
- 2) Improve programmer productivity
- 3) Allows programs to be independent of the computer

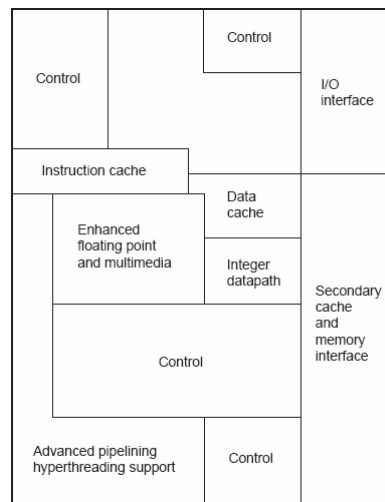
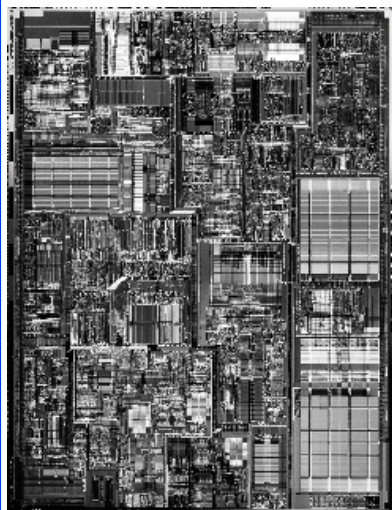


Computer Organization



The five classic components: 1) Input, 2) Output, 3) Memory, 4) Datapath 5) Control

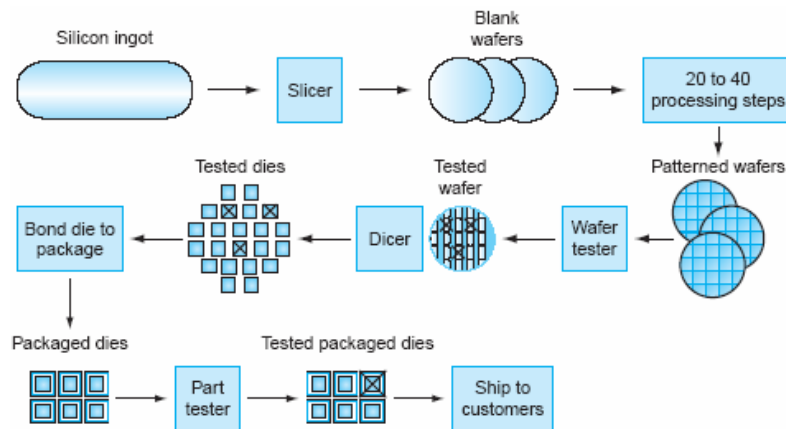
Inside Pentium 4



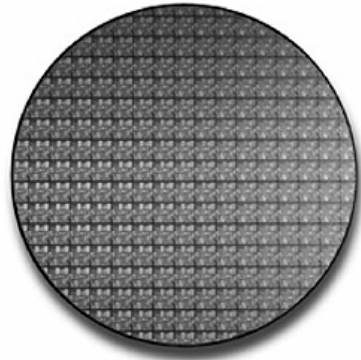
Communicating with other computers

- ❑ Networks are the backbone of the current computing system
- ❑ Information is exchanged among computers at high speed
- ❑ Resource sharing. Especially expensive resources.
- ❑ Nonlocal access. Freedom for users
- ❑ Optical, wireless

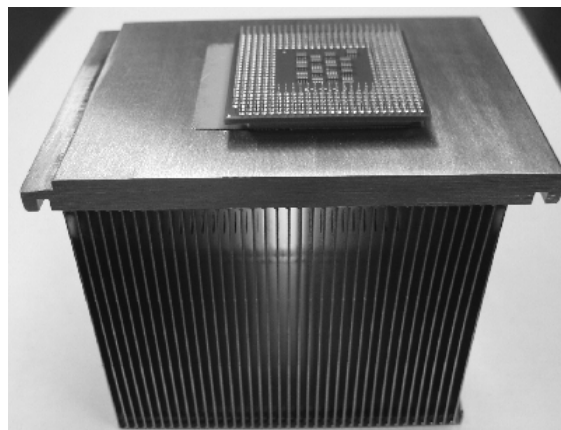
The Chip Manufacturing Process



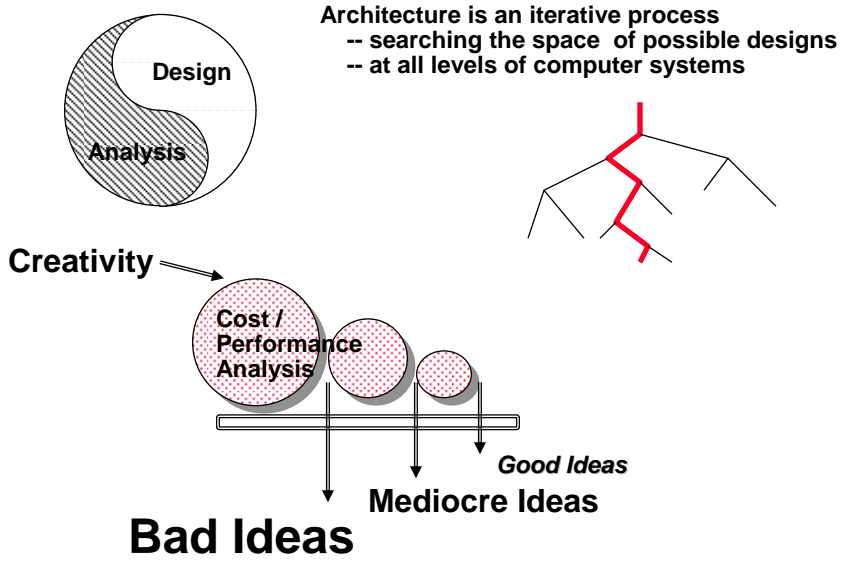
An 8-inch wafer containing Pentium 4 processors



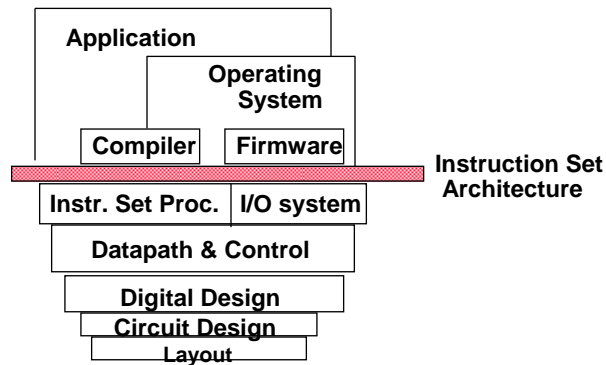
Pentium 4 Processor



Measurement and Evaluation



What is “Computer Architecture”?



- ❑ Coordination of many *levels of abstraction*
- ❑ Under a rapidly *changing set of forces*
- ❑ Design, Measurement, *and* Evaluation

Coping with CSC 7080

- ❑ Students with too varied background?
 - Review Chapters 1 to 8 “Computer Organization and Design (COD)3/e”
 - FAST review of basic concepts

Computer Architecture

- ❑ A modern meaning of the term *computer architecture* covers three aspects of computer design:
 - *instruction set architecture*,
 - *computer organization* and
 - *computer hardware*.
- ❑ *Instruction Set Architecture - ISA* refers to the actual programmer-visible machine interface such as instruction set, registers, memory organization and exception handling. Two main approaches: RISC and CISC architectures.
- ❑ A computer organization and computer hardware are two components of the implementation of a machine.

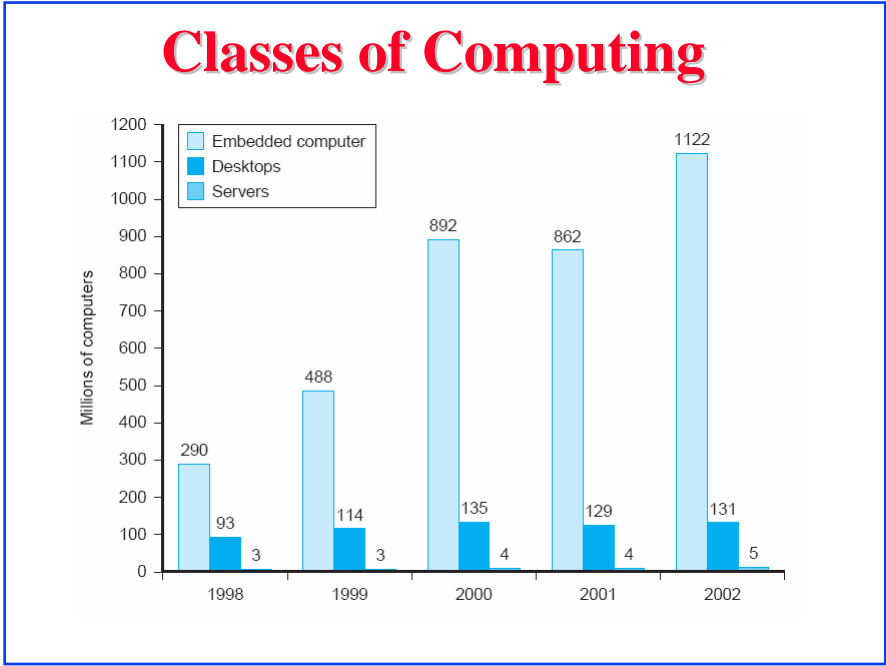
Computer Architecture

- ❑ *Computer organization* includes the high-level aspects of a design, such as the memory system, the bus structure, and the design of the internal CPU (where arithmetic, logic, branching and data transfers are implemented).
- ❑ *Computer hardware* refers to the specifics of a machine, included the detailed logic design and the packaging technology of the machine.
- ❑ For many years the interaction between ISA and implementations was believed to be small, and implementation issues **were** not a major focus in designing instruction set architecture.
- ❑ In the 1980's, it becomes clear that both the difficulty and inefficiency of pipelining could be increased by instruction set architecture complications.

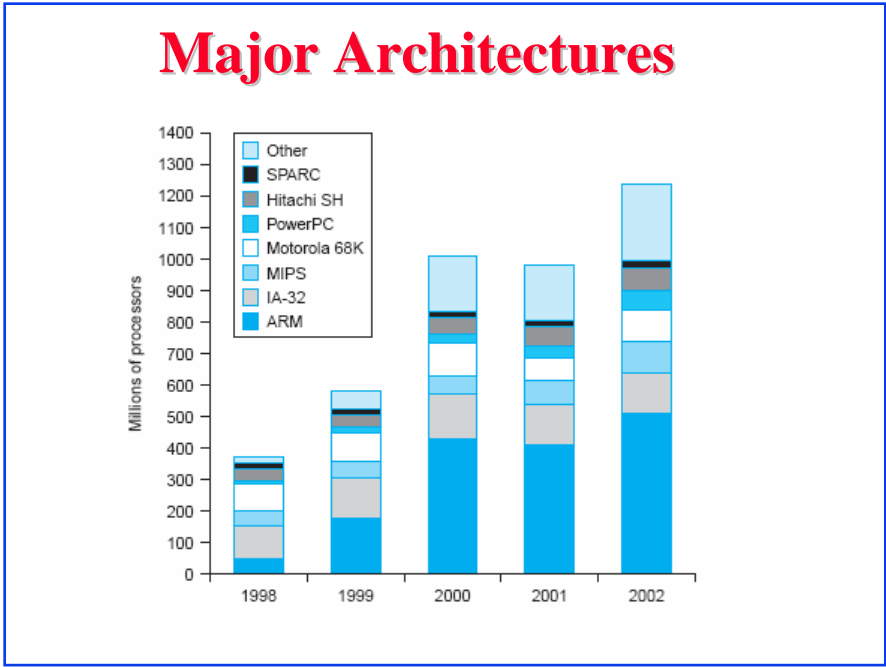
Tasks of Computer Architects

- ❑ Computer architects must design a computer to meet functional requirements as well as price, power, and performance goals. Often, they also have to determine what the functional requirements are, which can be a major task.
- ❑ Once a set of functional requirements has been established, the architect must try to optimize the design. Here are three major application areas and their main requirements:
 - Desktop computers: focus on optimizing cost-performance as measured by a single user, with little regard for program size or power consumption,
 - Server computers – focus on availability, scalability, and throughput cost-performance,
 - Embedded computers – driven by price and often power issues, plus code size is important.

Classes of Computing



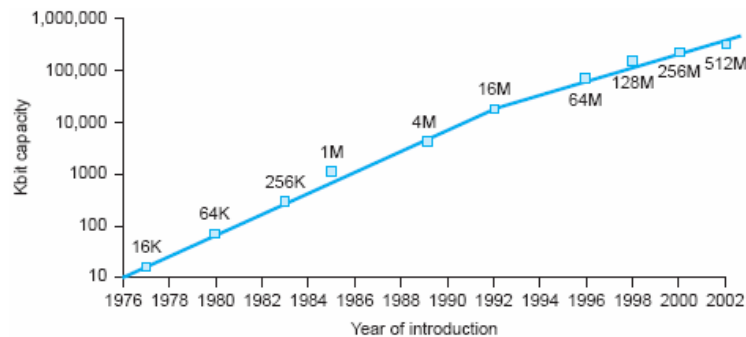
Major Architectures



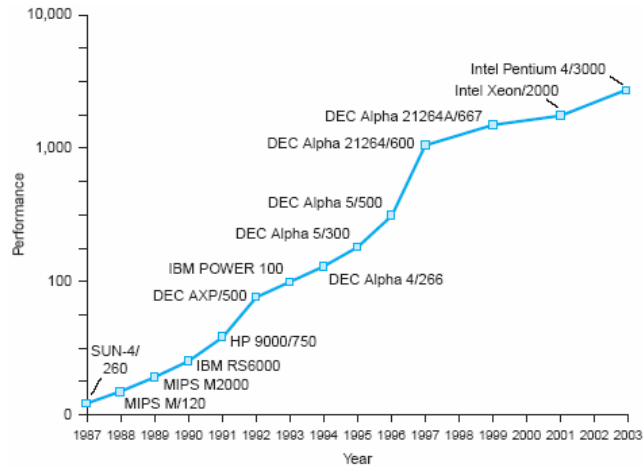
Rapid Rate of Improvements

- ❑ Now, less than one thousand dollars purchases a personal computer that has more performance, more main memory, and more disk storage than a computer bought in 1980 for one million dollars.
- ❑ For many applications, the highest-performance microcomputers of today outperform the supercomputers of less than 10 years ago.
- ❑ This rapid rate of improvement has come from two forces:
 - technology used to build computers and
 - innovations in computer design.

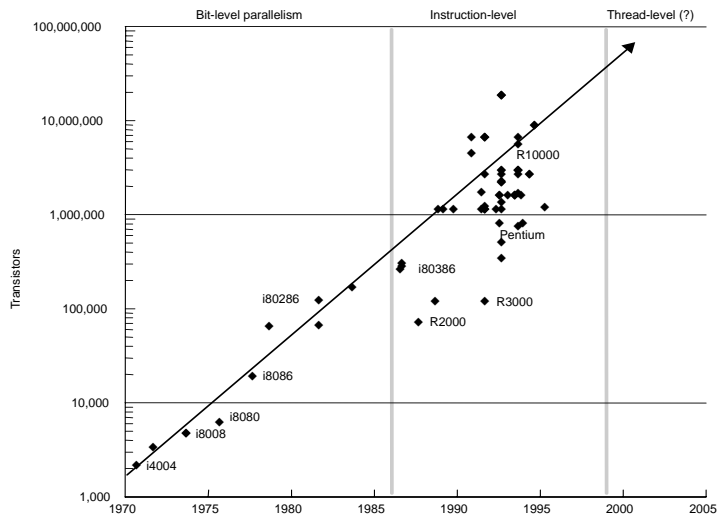
Growth of capacity per DRAM chip over time



Performance of Workstations



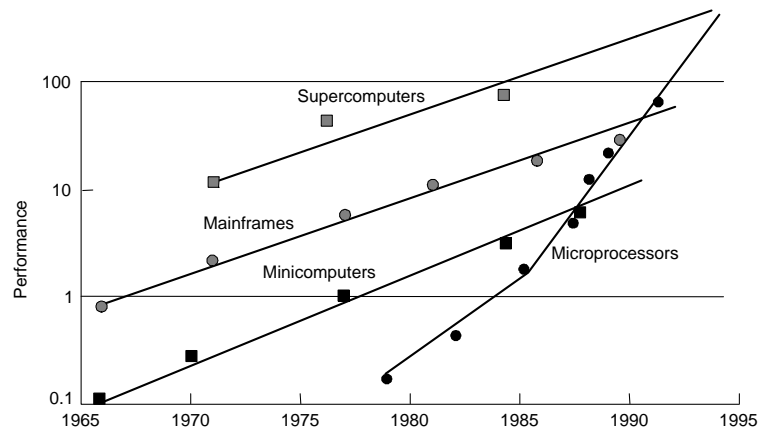
A take on Moore's Law



Technology Trends

- ❑ Clock Rate: ~30% per year
- ❑ Transistor Density: ~35%
- ❑ Chip Area: ~15%
- ❑ Transistors per chip: ~55%
- ❑ Total Performance Capability: ~100%
- ❑ by the time you graduate...
 - 3x clock rate (4-6 GHz)
 - 10x transistor count (1 Billion transistors)
 - 30x raw capability
- ❑ plus 16x dram density, 32x disk density

Performance Trends



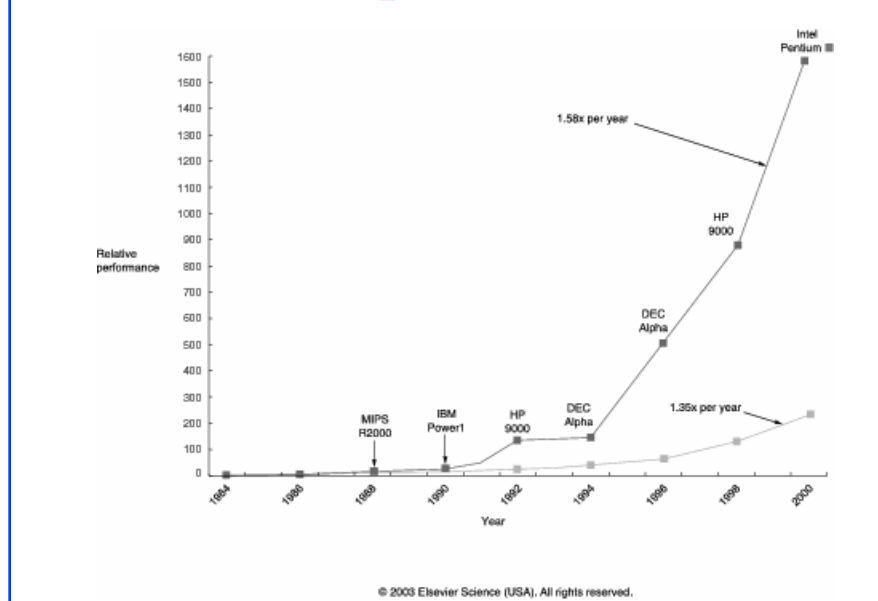
Technology Trends

- ❑ *Integrated circuit logic technology* – a growth in transistor count on chip of about 55% per year.
- ❑ *Semiconductor RAM* – density increases by 40% to 60% per year, while cycle time has improved very slowly, decreasing by about one-third in 10 years. Cost has decreased at rate about the rate at which density increases.
- ❑ *Magnetic disc technology* – disk density has been recently improving more than 100% per year, while prior to 1990 about 30% per year.
- ❑ *Network technology* – Latency and bandwidth are important, though recently bandwidth has been primary focus. Internet infrastructure in the U.S. has been doubling in bandwidth every year.

Developments in Computer Design

- ❑ During the first 25 years of electronic computers both forces, technology and innovations in computer design made major contributions.
- ❑ Then, during the 1970's, computer designers were largely dependent upon integrated circuit technology, with roughly 35% growth per year in processor performance.
- ❑ In the last 20 year, the combination of innovations in computer design and improvements in technology has led sustained growth in performance at an annual rate of over 55%. In this period, the main source of innovations in computer design has come from RISC-style pipelined processors.

Growth in Microprocessor Performance



RISC Architecture

- After 1985, any computer announced has been of RISC architecture. RISC designers focused on two critical performance techniques in computer design:
 - the exploitation of instruction-level parallelism, first through pipelining and later through multiple instruction issue,
 - the use of cache, first in simple forms and later using sophisticated organizations and optimizations.

RISC ISA Characteristics

- ❑ All operations on data apply to data in registers and typically change the entire register;
- ❑ The only operations that affect memory are load and store operations that move data from memory to a register or to memory from a register, respectively;
- ❑ A small number of memory addressing modes;
- ❑ The instruction formats are few in number with all instructions typically being one size;
- ❑ Large number of registers;
- ❑ These simple properties lead to dramatic simplifications in the implementation of advanced pipelining techniques, which is why RISC architecture instruction sets were designed this way.

RISC and CISC Architecture

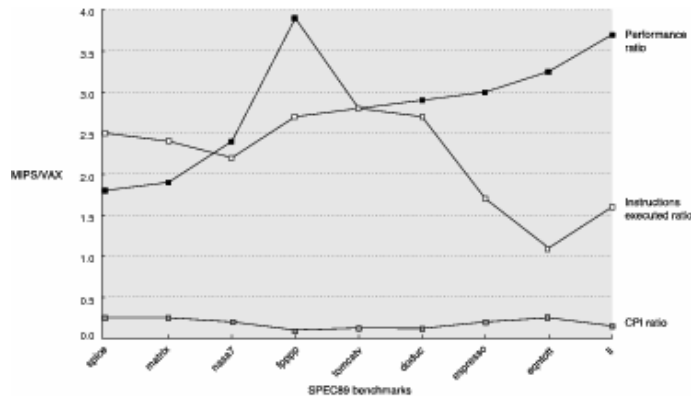
- ❑ **RISC** – **R**educed **I**nstruction **S**et **C**omputer
- ❑ **CISC** – **C**omplex (and **P**owerful) **I**nstruction **S**et **C**omputer
- ❑ What does **MIPS** stand for?
- ❑ Answer: **M**icroprocessor without **I**nterlocked **P**ipeline **S**tages. MIPS processor is one of the first RISC processors. Again, all processors announced after 1985 have been of RISC architecture.
- ❑ What is the main example of CISC architecture processor?
- ❑ Answer: Intel IA-32 processors (in over 90% computers).
- ❑ Intel IA-32 processors, from 80386 processor in early 80's to Pentium IV today, and the next one to be introduced this or next year, are of CISC architecture. All Intel IA-32 processors are having as a base the Identical instruction set architecture designed in early 1980's.

Intel IA-32 Processors

- ❑ The improvements in technology have allowed the latest Intel IA-32 processors (of CISC architecture) to adopt many innovations first pioneered in the RISC design.
- ❑ Since 1995, Pentium processors consist of a front end processor and a RISC-style processor.
- ❑ The front end processor fetches and decodes Intel IA-32 complex instructions and maps them into microinstructions.
- ❑ A microinstruction is a simple instruction used in sequence to implement a more complex instruction. Microinstructions look very much as RISC instructions.
- ❑ Then, the RISC-style processor executes microinstructions.
- ❑ As a transistor counts soared, the overhead (in transistors) of interpreting the more complex IA-32 architecture becomes negligible.

VAX vs. MIPS processors

- ❑ This graph compares results of VAX 8700 (CISC) and MIPS M2000 (RISC) processors that execute nine programs from SPEC89 benchmarks.
- ❑ $\text{CPU time} = \text{Instruction_count} * \text{CPI} / \text{Clock_rate}$



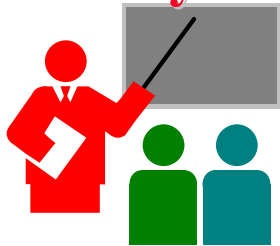
Comparison of VAX and MIPS processors

- ❑ The graph indicates that, on average, MIPS executed about twice as many instructions as VAX. (Bad for MIPS)
- ❑ The graph indicates that, on average, the CPI for the VAX was about six times larger than that for MIPS. (Bad for VAX)
- ❑ CPI – the average number of clock cycles per instruction
$$\text{CPI} = \text{CPU_clock_cycles_for_a_program} / \text{Instruction_count}$$
- ❑ CPU time – a time to execute a given program
$$\text{CPU time} = \text{Instruction_count} * \text{CPI} / \text{Clock_rate}$$
- ❑ Since clock rates were identical, MIPS had almost three times better performance (measured by CPU time) than VAX.

Reading Assignments

- ❑ See Class Web Page

Summary



- There will be a lot of self-reading
- Get ready to work hard
- Next Week papers are online

Thank You!

