CMSC 611
Spring 2000

Prof. Ethan L. Miller (elm@csee.umbc.edu)

Welcome to CMSC 611!

- Instructor: Professor Ethan Miller (`elm@csee.umbc.edu`)
- TA: Joy Su (`xsu1@umbc.edu`)
- Text: *Computer Architecture: A Quantitative Approach* (Hennessy and Patterson)
- Supplementary reading material: available online
- Course web pages
  - `http://www.csee.umbc.edu/courses/graduate/611/Spring00/`
  - Students are required to read Web pages regularly
  - Assignments & announcements will be on the Web
  - Class slides will be on the Web (available only within the `umbc.edu` domain)

Why take this course?

- Introduction to many of the methods used by designers to implement
  - General purpose processors
  - Memory systems
  - I/O systems
  - Multiprocessors
- Introduction to methods of analyzing CPU performance

More reasons to take this course

- Exposure to many characteristics of today's architectures
  - CISC: Pentium/Merced, 680x0
  - RISC: PowerPC, DEC Alpha, Sun SPARC, HP PA-RISC, MIPS, …
- Learn how to do graduate research in computer systems
- It's required….

Course mechanics

- Semester long project (40% of grade)
  - Work in groups of 3-4 students
  - Grade based on technical content & presentation
- Homework every 1-2 weeks (total 20% of grade)
- Exams
  - Midterm (15% of final grade)
  - Final exam (20% of final grade)
- Class participation: 5% of grade
- Project & both exams ***required*** to pass the class

Projects

- Done in groups of 3-4 people
- Many topics available
  - Pick one that the whole group is interested in
    - Work-related?
    - Issue of current relevance
  - Do something different from what everyone else is doing
- Meet project milestones throughout the semester
- Make a presentation on the project to the class
- Hand in a paper on the project

Course summary

- Measuring performance & cost
- Instruction sets
- Improving CPU performance
  - Dynamic instruction scheduling & instruction-level ||ism
  - Branch prediction
- Vector processors
- Memory hierarchies
- Storage systems
- Multiprocessors

What is computer architecture?

- Answer: the design of computer systems
- What must a computer system provide?
  - Application support
  - Software compatibility
  - OS requirements
  - Standards
- Computer system designer must take into account
  - Cost & performance
  - Design complexity
  - reliability and fault tolerance

Trends in hardware

- IC technology
  - Transistor density improves about 50% per year
  - 60% - 80% increase per year of transistors per chip
- Memory (RAM)
  - Density increases at around 60%/year
  - Cycle time decreases around 30% in ten years.
  - Bandwidth proportional to cycle time and data width

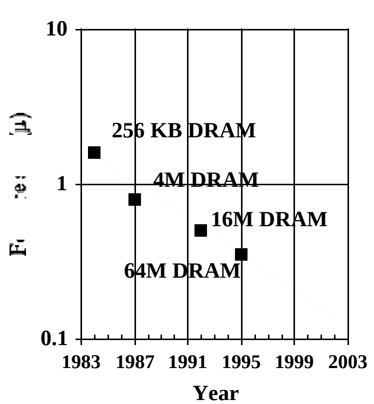
Moore's Law

- Gordon Moore (Intel Chairman, 1965) said "# of transistors in an IC will double every 18 months"
- Accurate for last 20 years; will it hold for the next 15 years?
  - Physics : lithographic limits
  - Economics: Design and test complexity, fabrication costs

	1995 (350 nm)	1998 (250 nm)	2001 (180 nm)	2004 (130 nm)	2007 (100 nm)	2010 (80 nm)
DRAM (bits)	64M	256M	1G	4G	16G	64G
MPU transistors / cm^2	4M	7M	13M	25M	50M	90M
DRAM chip size (mm^2)	190	280	420	640	960	1400
MPU chip size (mm^2)	250	300	360	430	520	620

Moore's Law (2)

- Minimum transistor feature size must decrease by a factor of 0.7 every three years
- Where is the lower limit for feature size?



Moore's Second Law

- Moore's Second Law:
  The cost of building a semiconductor fab is doubling every three to four years.
- In 1995, approximately 50 fabs in operation worldwide
  - Another 50 in some state of completion
  - Current cost > $1 billion
- Physical Limits
  - "In 2010, we will run into the physical limitation of having a fraction of an electron show up at a gate to switch the state of the transistor" (Joel Birnbaum, HP senior president of R&D)

More trends in hardware

- Disk technology
  - Storage density increases around 50% per year
  - Rotation speed increases slowly
  - Bandwidth proportional to square root of density; increases around 25% per year
  - Access time (seek) drops around 30% in ten years
- Systems must take trends into account
  - Moving targets!
  - Optimal system today may not be optimal tomorrow

Trends in cost

- Learning curve : products drop in cost over time
  - Minor improvements in design & manufacturing
  - Amortization of startup costs (R&D, etc.)
- Volume decreases per-unit cost
  - Fixed costs amortized over more units
  - May have more competitors to help keep prices down
  - Commodities: standardized components available from many vendors
    - Little or no profit margin for commodities
    - Often *much* less expensive than low-volume components
- Relative prices of components can change...

Where does the money go?

- Component costs: raw material costs.
- Direct costs: costs incurred to actually make a single item (usually 20% to 40% of component costs)
- Gross margin: overhead not associated with a single item — R&D, plant equipment, profit, taxes, etc. (typically 20% to 55% of average selling price)
- Average selling price: direct cost + gross margin.
- List price: ASP + reseller's margin

Design tradeoff examples

- Designing for low cost
  - Make choices that minimize cost no matter what
  - Design for ease of manufacture as well as low component cost
- Designing for high performance
  - Spend more money on R&D
  - Worry less about how much it costs to manufacture
- Designing for good cost/performance
  - Use more expensive parts if they have "bang for the buck"
  - Control costs, but not if they lower performance
- Cray vs. desktop workstation vs. Palm III