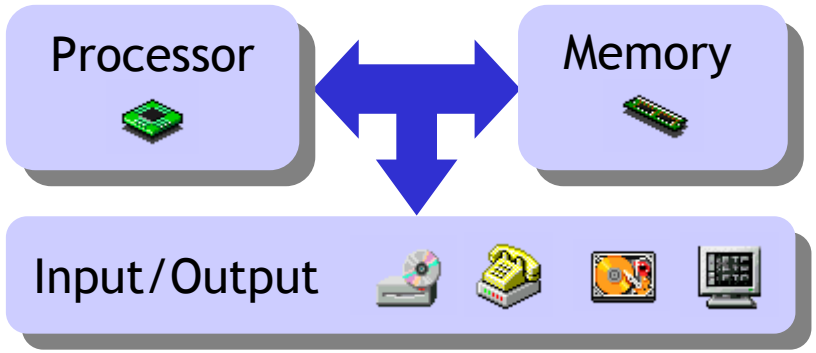
CS232: Computer Architecture II

Spring 2003

What is computer architecture about?

- Computer architecture is the study of building entire computer systems.



- There are numerous factors to consider, many of which are conflicting.
  - *Performance*, *price* and *reliability* are obviously vital concerns.
  - Systems should be *expandable* to accommodate future developments, but must also be *compatible* with existing technology.
  - *Power consumption* is especially important in the growing market of portable devices such as cell phones, PDAs, and MP3 players.

Why should you care?

ALIENWARE

AREA-51 T9™

Dragon Full-Tower Case (400-Watt PS) (Nova Yellow)
Intel® Pentium® 4 Processor 2.53GHz 533MHz FSB w/ 512KB Cache
Hi-Performance Heatsink/CPU Cooling Fan
Intel® D850EMVR Motherboard w/1AGP/5PCI RDRAM
Standard 1.44MB Floppy Drive
512MB RDRAM PC-800
80GB Seagate Barracuda ATA IV 7200RPM 2MB Cache
16/48x DVD-ROM - IDE - Black w/Software MPEG-2 Decoder
PROMO 48x24x48x CD-RW - IDE – Black
Hercules® 3D Prophet 9700 Pro 128MB AGP Dual Monitor
KoolMaxx Video Cooling System (Standard Chrome)
Alien Adrenaline: Video Performance Optimizer
Sound Blaster® Live! 5.1
Microsoft Internet Keyboard (Space Black)
Microsoft IntelliMouse Explorer 3.0 - USB (Standard Color)
Microsoft® Windows® XP Professional
Free Alienware® T-Shirt - Black
Bonus 12-Month Subscription to Computer Games Magazine!
Alien Autopsy: Automated Technical Support Request System
Aliencare Toll-Free 1-Year 24/7 ONSITE Warranty
Personalized Owner's Manual
Optimized & Configured for High-Performance
FREE Custom Alienware® Mouse Pad

Price: $1959.00

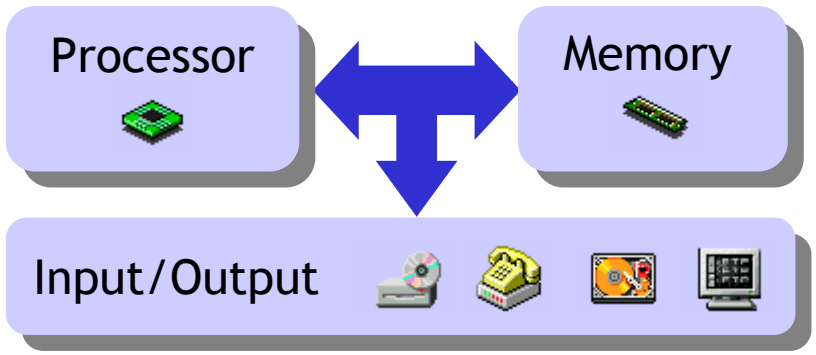
- Computer science majors are often expected to know something about hardware and computer architecture.
  - What are caches, RDRAMs, and AGPs?
  - Is a 2.53GHz processor or a 7200RPM hard disk worth it?

Architecture for programmers

- Knowing about architecture helps to explain why programming languages are designed the way they are.
  - What happens when we compile our source code?
  - Why is computer arithmetic sometimes wrong?
  - What is a bus error or segmentation fault?
- You can also learn how to make your code run faster.
  - Inlined functions are faster than normal function calls.
  - Where and how you store your data makes a big difference.
  - Just rearranging the order of statements can sometimes help!
- A lot of software development requires knowledge of architecture.
  - Compilers generate optimized code for specific processors.
  - Operating systems manage hardware resources for applications.
  - Good I/O systems are important for databases and networking.
  - We'll touch upon concurrent and parallel programming issues like data dependencies and memory consistency.

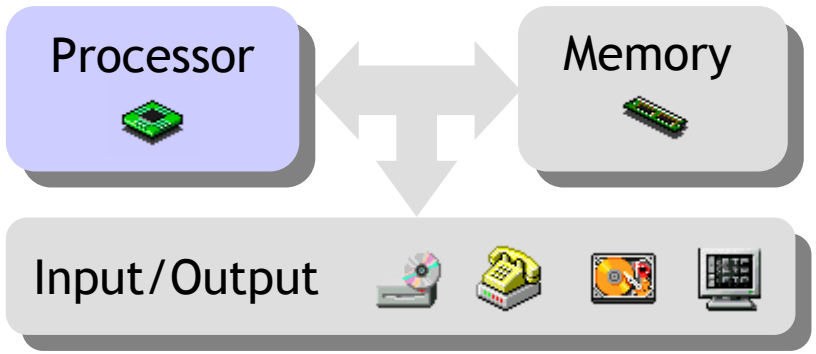
What is CS232 about?

- CS232 is roughly split into three parts.
  - The first third discusses instruction set architectures—the bridge between hardware and software.
  - Next, we introduce more advanced processor implementations. The focus is on pipelining, which is one of the most important ways to improve performance.
  - Finally, we talk about large and fast memory systems, I/O, and how to connect it all together.
- These are basically the same topics from CS231, but in more depth.



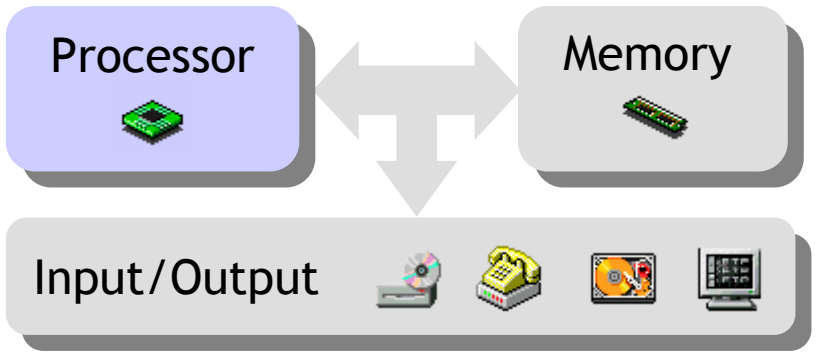
Instruction set architectures

- An instruction set describes the basic functions that a processor can perform. It serves as an interface between hardware and software; programs are sequences of instructions that get executed by hardware.
- We'll talk about several important issues that we didn't see with the simple processor from CS231.
  - The instruction set in CS231 lacked many features, such as support for function calls. We'll work with a larger, more realistic processor.
  - We'll also see more ways in which the instruction set architecture affects the hardware design.
- We (i.e., you) do more assembly-language programming too.



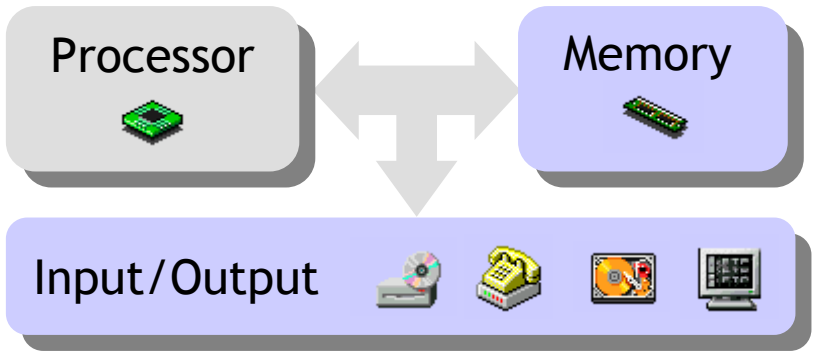
Processor design

- The second part of the semester will address two other limitations of the single-cycle processor from CS231.
  - Supporting more complex instructions would increase the cycle time.
  - The CPU hardware is not fully utilized, so it runs slower than it could.
- We will focus on pipelining, which is one of the most important ways of speeding up processors.
  - The idea behind pipelining is very simple, but there are many details and special cases that must be handled.
  - Every modern processor uses pipelining.



Memory and I/O

- Memory and I/O are often bottlenecks in modern machines.
  - Processor speeds far outpace memory and I/O speeds.
  - A 4GHz processor won't help you browse the web any faster if you're stuck on a 56kbps modem.
- We'll study some of the issues associated with memory and I/O.
  - How caches can dramatically improve the speed of memory accesses.
  - How processors, memory and peripheral devices can be connected, and CPU support for I/O communications.



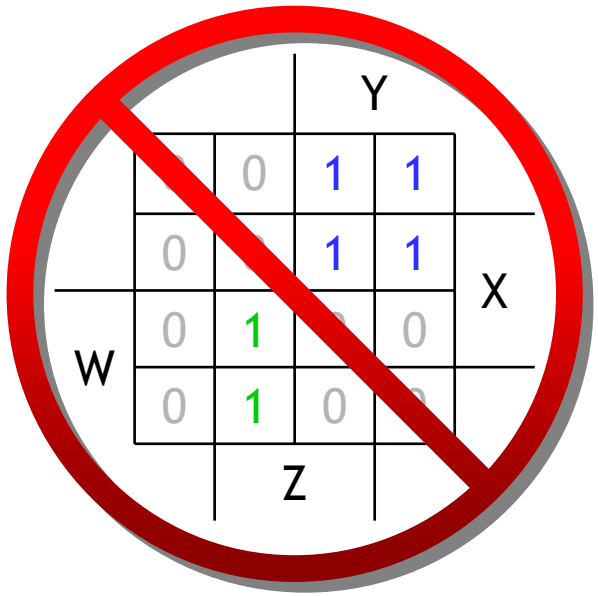
Performance

- So in a sense, CS231 shows how to make processors, while CS232 shows how to make processors *fast*.
- We will talk about how to measure performance accurately.
  - Quantifying performance improvements is critical in evaluating the costs and benefits of different system designs.
  - Unfortunately, many companies provide misleading or incomplete information about the performance of their products.
- Our knowledge of performance will help us figure out just how good pipelining and caching can be.

The Earth Simulator, a really fast computer.

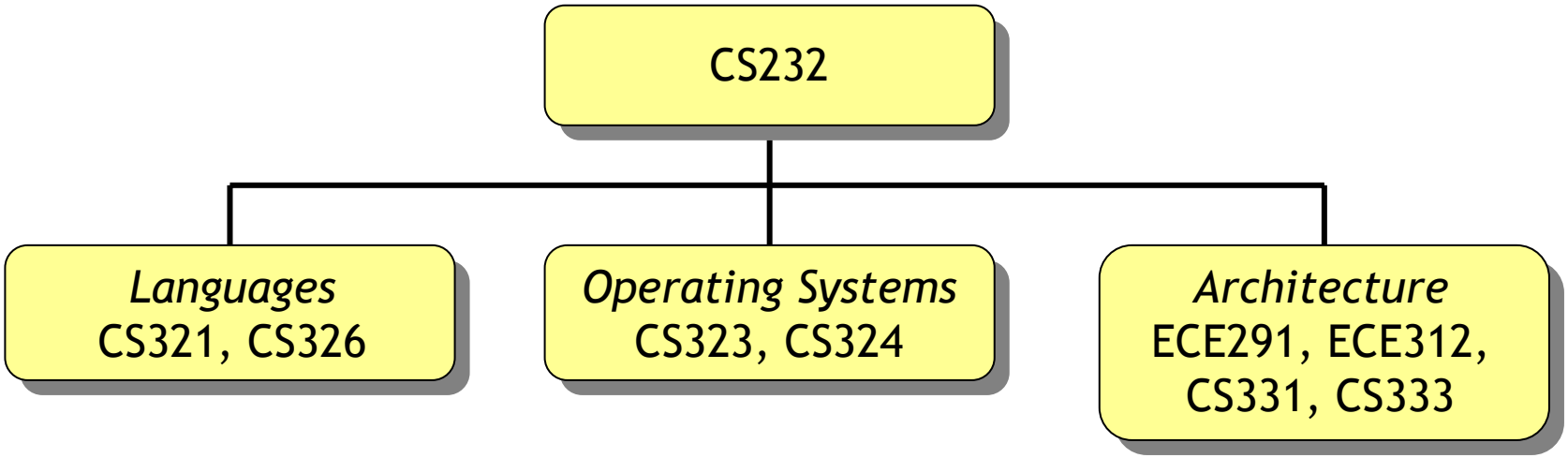
CS231 vs. CS232

- This class expands upon the computer architecture material from the last few weeks of CS231, and we rely on many other ideas from CS231.
  - Understanding binary, hexadecimal and two's-complement numbers is still important.
  - Devices like multiplexers, registers and ALUs appear frequently. You should know what they do, but not necessarily how they work.
  - Finite state machines and sequential circuits will appear again.
- We do *not* spend much time with logic design topics like Karnaugh maps, Boolean algebra, latches and flip-flops.



What to do after CS232

- CS232 is a prerequisite for several language and systems classes.
  - CS321 and CS326 cover programming languages and compilers.
  - CS323 and CS324 are about operating systems and real-time systems.
- You can keep going with computer architecture as well.
  - ECE291 focuses on Intel x86 assembly language and architecture.
  - ECE312 is similar to CS232, but includes processor simulations and a little more information about virtual memory and operating systems.
  - CS331 discusses embedded systems like cell phones and car parts.
  - CS333 goes into even more depth than CS232.



General hints to reach CS232 nirvana

- **Remember the big picture.**
  What are we trying to accomplish, and why?
- **Read the textbook.**
  It's clear, well-organized, and well-written. The diagrams can be complex, but are worth studying. Work through the examples and try some exercises on your own. Read the "Real Stuff" and "Historical Perspective" sections.
- **Talk to each other.**
  You can learn a lot from other CS232 students, both by asking and answering questions. Find some good partners for the homeworks (but make sure you all understand what's going on).
- **Help us help you.**
  Come to lectures, sections and office hours. Send email or post on the newsgroup. Ask lots of questions! Check out the web page:

http://www-courses.cs.uiuc.edu/~cs232