Computer Organization and Design

Dr. Arjan Durresi
Louisiana State University
Baton Rouge, LA 70803
durresi@csc.lsu.edu

These slides are available at:
http://www.csc.lsu.edu/~durresi/CSC3501_07/

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) 55%
- Activity in the class (15%)
  - Questions and discussions in class give you points and improve the quality of teaching
- Homework (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.
- All quizzes are open-book and extremely time limited.
- Quizzes consist of numerical as well as multiple-choice (true-false) questions.
- There is negative grading on incorrect multiple-choice questions.

What Is This Course About?

- Understanding of basic issues, concepts, principles of Computer Architecture.
- Organizational paradigms that determine: Capabilities, Performance and the Success
- Relationship between hardware and software
- Focus on the concepts that are the basis of current computers.

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?

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/CSC3501_07
- GTA:
  - Email:

Why learn this stuff?
- You want to call yourself a "computer scientist"
- You want to build software people use (need performance)
- You need to make a purchasing decision or offer "expert" advice
- Both Hardware and Software affect performance:
  - Algorithm determines number of source-level statements
  - Language/Compiler/Architecture determine machine instructions (Chapter 2 and 3)
  - Processor/Memory determine how fast instructions are executed (Chapter 5, 6, and 7)
- Assessing and Understanding Performance in Chapter 4

Introduction
- This course is all about how computers work
- But what do we mean by a computer?
  - Different types: desktop, servers, embedded devices
  - Different uses: automobiles, graphics, finance, genomics...
  - Different manufacturers: Intel, Apple, IBM, Microsoft, Sun...
  - Different underlying technologies and different costs!
- Analogy: Consider a course on "automotive vehicles"
  - Many similarities from vehicle to vehicle (e.g., wheels)
  - Huge differences from vehicle to vehicle (e.g., gas vs. electric)
- Best way to learn:
  - Focus on a specific instance and learn how it works
  - While learning general principles and historical perspectives

What is a computer?
- Components:
  - input (mouse, keyboard)
  - output (display, printer)
  - memory (disk drives, DRAM, SRAM, CD)
  - network
- Our primary focus: the processor (datapath and control)
  - implemented using millions of transistors
  - Impossible to understand by looking at each transistor
  - We need...
**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?**

---

**How do computers work?**

- Need to understand abstractions such as:
  - Applications software
  - Systems software
  - Assembly Language
  - Machine Language
  - Architectural Issues: i.e., Caches, Virtual Memory, Pipelining
  - Sequential logic, finite state machines
  - Combinational logic, arithmetic circuits
  - Boolean logic, Is and Os
  - Transistors used to build logic gates (CMOS)
  - Semiconductors/Silicon used to build transistors
  - Properties of atoms, electrons, and quantum dynamics
- So much to learn!

---

**Historical Perspective**

- ENIAC built in World War II was the first general purpose computer
  - Used for computing artillery firing tables
  - 80 feet long by 8.5 feet high and several feet wide
  - Each of the twenty 10 digit registers was 2 feet long
  - Used 18,000 vacuum tubes
  - Performed 1900 additions per second

- Since then: Moore’s Law: transistor capacity doubles every 18-24 months

---

**Forces on Computer Architecture**

- Technology
- Programming Languages
- Applications
- Operating Systems

**History**

\[ A = F / M \]

---

**Amazing Underlying Technology Change**

**Below Your Program**

- Applications software
- Systems software
- Hardware
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

Measurement and Evaluation
Architecture is an iterative process

- searching the space of possible designs
- at all levels of computer systems

Creativity

Good Ideas
Bad Ideas

Cost / Performance
Analysis

Mediocre Ideas

What is “Computer Architecture”?
- Coordination of many levels of abstraction
- Under a rapidly changing set of forces
- Design, Measurement, and Evaluation

Pentium 4 Processor

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.

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.
A take on Moore's Law

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.

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

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.

Performance Trends

Growth in Microprocessor Performance

[Graphs and diagrams illustrating the trends described in the text]
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 - Reduced Instruction Set Computer
- CISC - Complex (and Powerful) Instruction Set Computer
- What does MIPS stand for?
- Answer: Microprocessor without Interlocked Pipeline Stages. 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 sourred, 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.
- CPU time = Instruction_count * CPI / 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
  CPI = CPU_clock_cycles_for_a_program / Instruction_count
- CPU time - a time to execute a given program
  CPU time = Instruction_count * CPI / Clock_rate
- Since clock rates were identical, MIPS had almost three times better performance (measured by CPU time) than VAX.
Summary

- There will be a lot of self-reading
- Get ready to work hard

Thank You!