Introduction

CSE 410, Spring 2005
Computer Systems

http://www.cs.washington.edu/education/courses/410/05sp/

Reading and References

• Reading
  » Chapter 1, Computer Organization and Design, Patterson and Hennessy
  » Chapter 3, read 3.1 through 3.4

Administrative

• Instructor:
  » Hank Levy
  » levy@cs.washington.edu
• TAs:
  » Scott Schremmer (scotths@cs.washington.edu)
  » Charles Giefer (cgiefer@cs.washington.edu)
• All class info is on the web site
  » http://www.cs.washington.edu/410/CurrentQtr

Class Overview

• Provide an introduction to the inner workings of computer systems
• Levels of abstraction
  » bits, bytes, assembly language
  » operating system concepts
  » higher level languages - C, C++, Java, …
  » application programs
Goal

• You will understand
  » what is actually happening when a computer system is running application programs
• So that you will be able to
  » make good design choices as a developer, project manager, or system customer
  » calibrate your hype-o-meter with facts

The structure of this class

• The hardware / software interface
  » the elements of a computer system
  » what parts are visible to the software
  » instruction set architecture (ISA)
  » what happens inside the CPU
• Operating systems
  » services an OS performs for an application
  » design of various OS components
  » OS mechanisms and policies
  » why my OS crashes

Computers

• Computers impact our lives in a huge number of ways:
  » Computer-controlled brakes in your car
  » You look up everything with Google
  » You take a picture of a bad cut with your cell phone and email it to your doctor
  » You download music for your MP3 player
• All this has been enabled by an incredible advance in microprocessor technology
Illustration of Moore’s Law

A modern CPU

- Latest Intel P4
  » 3.6 gigahertz
  » 2 MB L2 cache
  » 20-stage pipeline
  » out-of-order instruction execution
  » branch prediction
  » 100s of instructions executing at once
  » “hyper-threading” technology
  » …….

What’s next

- We’re in trouble
  » hard to go much faster with uniprocessors
  » chips have gotten so big, it’s a long way from one side to the other (in cycles)
  » as chips get bigger, chance of errors in the chip goes up
  » we need new ways to build faster computers
  » these new ways usually involve adding more parallelism
- In a few years, every chip will have multiple CPUs on it (maybe 4 to 16) [called “multi-core”]

Layers of abstraction

- Abstraction
  » defines a layer in terms of functions / interfaces
  » isolates a layer from changes in the layer below
  » improves developer productivity by reducing detail needed to accomplish a task
  » helps define a single architecture that can be implemented with more than one organization
Architecture and Organization

• Architecture
  » defines elements and interfaces between layers
  » ISA: instructions, registers, addressing
• Organization
  » components and connections
  » how instructions are implemented in hardware
  » many different organizations can implement a single architecture

Computer Architecture

• Specification of how to program a specific computer family
  » what instructions are available?
  » how are the instructions formatted into bits?
  » how many registers and what is their function?
  » how is memory addressed?
  » how does I/O work?
• The MIPS 1 architecture is the basis for the first half of this course

Architecture Families

• IBM 360, 370, … (the first computer family)
• PowerPC 601, 603, …
• DEC VAX, PDP-11
• Intel x86: 286, 386, 486, Pentium, P4,…
• Intel IA64 Itanium
• MIPS R2000, R3000, R4000, R5000, …
• SUN Sparc

Computer Organization

• Processor
  » datapath (functional units) manipulate the bits
  » control hardware manages the manipulation
• Memory
  » Registers – 100s of bytes, very fast, on the CPU
  » cache memory – 1000s of bytes, fast, on the CPU
  » main memory – millions of bytes, slower, off the CPU
• Input / Output
  » interface to the rest of the world
Architecture and Organization

- Architecture is a layer of abstraction
- One architecture can be implemented with many organizations
- One organization can support multiple architectures
- Different manufacturing technologies

Many possible implementations

A typical organization

Change Organization or Architecture?

- Theory
  » Organization changes provide incremental changes in speed and cost for same software
  » Architecture changes enable breakthrough changes in speed and cost for new software
- Real life
  » incremental changes are very rapid (once a year)
  » breakthrough changes are very costly (once a decade)