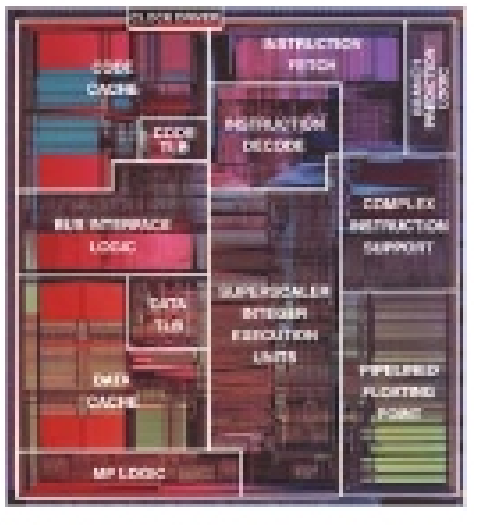
CS333: Computer Architecture
University of Virginia Computer Science
Fall 2008 Michele Co

Class 1: Introduction



Menu

- Course Structure
- Course Goals
- First Assignment
- Course Admin (add class/change section)

Course Structure and Expectations

Staff

- Me: Michele Co
  - 228A Olsson Hall, 982-2688
  - Office Hours: (tentative) posted on course website
- TAs
  - Timothy Hnat (grad)
  - Krasimira Krapitanova (grad)
  - Ugrad TAs TBA
  - Office Hours: TBA

Contacting Us

- Office hours! Finalized times to be posted soon.
- cs333@cs.virginia.edu
  - We will try to respond within 24 hours. Please get in touch with us again if you hear no response within that time frame.
- Anonymous Feedback

Contacting You

- Information (relevant notes, announcements, etc.) will be posted on class website:
  - http://www.cs.virginia.edu/~cs333
- Class Email

Meetings

- Lectures: 3 per week
  - Will include material not in the book
  - Many classes will use slides and notes
  - Some classes will not
- Lab meetings: 1 section per week
  - Won't meet each week. In-lab activities will be announced in advance

Grading Criteria (Subject to Change)

- 2 Midterms (50%)
- Homework and Lab (20%)
- Final Exam (20%)
- Class Participation (10%)

Expectations: Honor

- Everyone will be expected to follow what is on the course pledge
- Exact content will be determined before next week based on your survey responses
  - I hope to place a large burden on the honor system, but won't if students think this puts honest students at a disadvantage

Course Goals

CS333 Roadmap

Levels of Abstraction

Application

High Level Languages

Assembly language

Architecture

Gate-level logic

Electronics

Semiconductor

Real World Physics

Course Goal 1

- Learn to think about the machine from the
  - Architecture level
  - Assembly language level
  - Logic level
- Understand relationships and interactions between levels

Course Goal 2

Be able to ***understand*** key computer organization concepts.

Course Goal 3

Understand the processor design process and tradeoffs made to achieve particular design goals.

General Overview

Terminology

- Machine Language
  - Fundamental instructions expressed as 0s and 1s
- Assembly Language
  - Human-readable equivalent of machine language
- Assembler
  - Converts assembly language program to machine language
- Stored program
  - Program can be stored on computer with its data and be manipulated as if it were data

Machine and Assembly Language

- The assembler converts assembly language to machine language. You must also know how to do this.

Op code Data reg. #5 Data reg. #4

MC68000 Assembly Language	Machine Language
MOVE.W D4, D5	0011 101 000 000 100
ADDI.W #9, D2	00000001 10 111 100 0000 0000 0000 1001

Table 1.2 Two Motorola MC68000 instructions

Assembly Programmer's View

- Instruction set architecture (ISA)
  - Collection of all possible operations on a machine
  - Includes:
    - Instruction set
    - machine memory
    - programmer-accessible registers
- Tools
  - Assembler
  - Linker
  - Debugger