CS 230 – Introduction to Computers and Computer Systems

Module 0 – Introduction

Murray Dunne
mdunne@uwaterloo.ca

(Slides based on materials prepared by Sandy Graham and Martin Karsten)
Goals and Overview
Course Goals

- Overview of computer systems
  - from bottom to top
- Understand basic challenges & techniques
- Understand performance implications
Course Information

- Course materials and syllabus found on the course website
  - https://www.student.cs.uwaterloo.ca/~cs230/
- Communication
  - Piazza – Invite or Code
  - Email
  - Office Hours
Expected Background

- One of CS116, CS 136, CS 138, CS 146
  - basic understanding of an imperative programming language
  - Python/C/Java/JavaScript/C#/C++/etc.
- Basic arithmetic
  - exponentiation and logarithms
Course/Assignment Tools

- UWaterloo student.cs environment
- UNIX tools
- MIPS assembler
- MIPS emulator

- Attend the tutorials!
About the Slides

- Some material and figures taken from the optional textbook and accompanying slides:
  David Patterson and John Hennessy. Computer Organization and Design – The Hardware/Software Interface

- Figures taken from other sources are shown with reference

- Other material newly developed for this course
Motivation and Background
Models of Computation

- Functional: e.g., Scheme or Racket
  - mathematical, function/expression evaluation
  - ties in with formal systems
- Imperative: e.g., Python or C
  - value oriented
  - load and store architecture
  - closer to reality
Below Your Program

- Application software
  - high-level language
- System software
  - operating system: service code
  - tools and libraries
- Hardware
  - processor, memory, I/O
System Layers

Transistors and Electrical Properties

Logic Gates

Binary Signals & Number Representation

CPU Instructions and Pipelining

Memory and Caching

Build and Runtime Environments

Multiprocessing and Operating Systems

Python/C/Racket Code

CS 230 - Spring 2019
Course Topics / Modules

- arithmetic, hardware, data
- assembly language
- system internals
- build and runtime
- multiprocessing
- operating systems (if time allows)
High Level Concepts
Model of a Computer

- von Neumann model
- CPU
  - control & data path
- I/O
  - user, storage, network
- Memory
  - program and data stored in same memory
Performance: Latency vs. Throughput

- Tim Hortons
  - time to coffee vs. customers/hour
  - low latency => high throughput
    - but not vice versa
  - faster coffee makers vs. more (and more space)

- Latency (response time)
  - completion time of specific task

- Throughput
  - total work done over time period
Technology Trends

- Electronics technology continues to evolve
  - increased capacity and performance
  - reduced cost

<table>
<thead>
<tr>
<th>Year</th>
<th>Technology</th>
<th>Relative performance/cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1951</td>
<td>Vacuum tube</td>
<td>1</td>
</tr>
<tr>
<td>1965</td>
<td>Transistor</td>
<td>35</td>
</tr>
<tr>
<td>1975</td>
<td>Integrated Circuit</td>
<td>900</td>
</tr>
<tr>
<td>1995</td>
<td>Very large scale IC (VLSI)</td>
<td>2,400,000</td>
</tr>
<tr>
<td>2005</td>
<td>Ultra large scale IC</td>
<td>6,200,000,000</td>
</tr>
</tbody>
</table>

DRAM capacity
Moore's Law

- Transistor density doubles every two years
  - every year 1959-1975
- In the past
  - transistor density translated into processing power
  - almost double speed every 2 years...
  - reduce latency, increase throughput
- Recently
  - Memory and power consumption wall
Uniprocessor Performance

Constrained by power, instruction-level parallelism, memory latency
Trade-Offs

- almost everything in CS is a trade-off
  - very few absolute truths
- “fast, good, or cheap – pick two”