Introduction to
Electrical and Computer Engineering

Christopher Batten

Computer Systems Laboratory
School of Electrical and Computer Engineering
Cornell University

ENGRG 1060 Explorations in Engineering Seminar
Summer 2012
ECE is the Study and Application of Electricity, Micro-Electrononics, and Electro-Magnetism
ECE is everywhere!

- Game Consoles
- Electric and Hybrid Vehicles
- Fiber Optic Networks
- Automobiles
- Digital Cameras
- Medical Imaging Portable Medical Devices
- Computing: From Handhelds to Servers
- Internet Routers
- GPS Devices and Satellites
- Humanoid Robots Unmanned Vehicles
- Solar Panels
- Data Centers
What can one do with a background in ECE?

- **ECE Industry**: Intel, AMD, Analog Devices, NVIDIA, HP, Apple
- **General Engineering Industry**: GE, Lockheed Martin, Raytheon
- **Software Industry**: Microsoft, Amazon, Mathworks
- **Join a Startup**: Achronix, Hillcrest Labs
- **Research Lab**: Sandia National Labs, Draper Labs, NASA
- **Consulting**: McKinsey, Accenture, Deloitte, Booz Allen Hamilton
- **Finance**: Deutsche Bank, Capital One, UBS, Bloomberg
- **Graduate School**: Law School, Business School, Med School
- **Found a university!**
Cornell was founded because of ECE!

**Samuel Morse** invented the telegraph (a digital communication device), but needed help building the network.

**Ezra Cornell** built the first telegraph line (the beginning of telecommunications), and invested in the Western Union Telegraph Co.

"What hath God wrought?"

Ezra Cornell’s investments created the fortune that eventually enabled the founding of Cornell University.
“Optional Homework”

- Visit the statue of Ezra Cornell on the Arts Quad
- Does something on the back of the statue relate to ECE?
- Take a picture with your cellphone and send it to your friend!
  - Power systems
  - Computer engineering
  - Electrical circuits
  - Electrical devices
  - Signal processing
  - Telecommunications
## Talk Outline

### ECE Overview

- What is Computer Engineering?
- Trends in Computer Engineering
- Computer Engineering Design
Computer Engineering Artifacts
In its broadest definition, computer system design is the development of the abstraction/implementation layers that allow us to execute information processing applications efficiently using available manufacturing technologies.
The Computer Systems Stack

Sort an array of numbers
2, 6, 3, 8, 4, 5 -> 2, 3, 4, 5, 6, 8

Insertion sort algorithm
1. Find minimum number in input array
2. Move minimum number into output array
3. Repeat steps 1 and 2 until finished

C implementation of insertion sort

```c
void isort( int b[], int a[], int n ) {
    for ( int idx, k = 0; k < n; k++ ) {
        int min = 99
        for ( int i = 0; i < n; i++ ) {
            if ( a[i] < min ) {
                min = a[i];
                idx = i;
            }
        }
        b[k]   = min;
        a[idx] = 99;
    }
}
```
The Computer Systems Stack

- Application
- Algorithm
- Programming Language
- Operating System
- Instruction Set Architecture
- Microarchitecture
- Register-Transfer Level
- Gate Level
- Circuits
- Devices
- Technology

Mac OS X, Windows, Linux
Handles low-level hardware management

MIPS32 Instruction Set
Instructions that machine executes

```
blez  $a2, done
move $a7, $zero
li    $t4, 99
move $a4, $a1
move $v1, $zero
li    $a3, 99
lw    $a5, 0($a4)
addiu $a4, $a4, 4
slt   $a6, $a5, $a3
movn $v0, $v1, $a6
addiu $v1, $v1, 1
movn $a3, $a5, $a6
```
The Computer Systems Stack

- Application
- Algorithm
- Programming Language
- Operating System
- Instruction Set Architecture
- Microarchitecture
- Register-Transfer Level
- Gate Level
- Circuits
- Devices
- Technology

- How data flows through system
- Boolean logic gates and functions
- Combining devices to do useful work
- Transistors and wires
- Silicon process technology

Silicon process technology
Computer Systems: CS vs. EE vs. CE

Computer Engineering is at the interface between hardware and software and considers the entire system.
ENGRG 1060 Computer Systems Labs

Computer Engineering

- Application
- Algorithm
- Programming Language
- Operating System
- Instruction Set Architecture
- Microarchitecture
- Register-Transfer Level
- Gate Level
- Circuits
- Devices
- Technology

Lab 2
Hardware pushing towards software (EE,CE)

Lab 4
Software pushing towards hardware (CS,CE)
In its broadest definition, computer system design is the development of the abstraction/implementation layers that allow us to execute information processing applications efficiently using available manufacturing technologies.
Processors, Memories, and Networks

Computer engineering basic building blocks
• **Processors** for computation
• **Memories** for storage
• **Networks** for communication
Activity #1: Sorting with a Sequential Processor

- **Application:** Sort 32 numbers

- **Simulated Sequential Computing System**
  - Processor: You!
  - Memory: Worksheet, read input data, write output data
  - Network: Passing/collecting the worksheets

- **Activity Steps**
  1. Discuss strategy with neighbors
  2. When instructor starts timer, flip over worksheet
  3. Sort 32 numbers as fast as possible
  4. Lookup when completed and write time on worksheet
  5. Raise hand
  6. When everyone is finished, then analyze data
Talk Outline

ECE Overview

What is Computer Engineering?

Trends in Computer Engineering

Computer Engineering Design
Application Requirements vs. Technology Constraints

Traditional Application Requirements
- As much processor compute as possible
- As much memory capacity as possible
- As much network bandwidth as possible

Traditional Technology Constraints
- Exponential scaling of resources
Exponential Scaling for Processor Computation

- Data partially collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond
Exponential Scaling for Memory Capacity

First presentation at ISSCC/Symp. VLSI Circuits

Adapted from [Itoh’07]
Exponential Scaling for Network Bandwidth

Network Peak Bisection Bandwidth (MB/s)

- Bus-Based Interconnection Networks (10's Nodes)
- Multi-Stage Interconnection Networks (100-1000's Nodes)


ENGRG 1060 Intro to ECE – Christopher Batten
Exponential Scaling for Processor Compute/Dollar

Adapted from [Kurzweil'09]
Exponential Scaling of Memory Capacity/Dollar

Adapted from [Kurzweil'09]
Key trends in the application requirements and technology constraints over the past decade have resulted in a radical rethinking of the processors, memories, and networks used in modern computing systems.

**Three Key Trends in Computer Engineering**

1. Growing diversity in application requirements motivate growing diversity in computing systems.

2. Energy and power constraints motivate transition to multiple processors integrated onto a single chip.

3. Technology scaling challenges motivate new emerging processor, memory, and network device technologies.
Trend 1: Bell’s Law

Roughly every decade a new, lower priced computer class forms based on a new programming platform resulting in new usage and industries.
Trend 1: Growing Diversity in Apps & Systems

- Sensor Nets
- Set-top boxes
- Games
- Cameras
- Laptops
- Supercomputers
- Media Players
- Servers
- Robots
- Smart phones
- Automobiles
Trend 2: Energy/Power Constrain All Modern Systems

Power = \frac{Energy}{Second} = \frac{Energy}{Op} \times \frac{Ops}{Second}

- Chip Packaging
- Chip Cooling
- System Noise
- Case Temperature
- Data-Center Air Conditioning

- Battery Life
- Electricity Bill
- Mobile Device Weight
Trend 2: Power Constrains Single-Processor Scaling

Data partially collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond
Trend 2: Transition to Multicore Processors

Intel Pentium 4
Single monolithic processor

Cray XT3 Supercomputer
1024 single-core processors

AMD Quad-Core Opteron
Four cores on the same die

IBM Blue Gene Q Supercomputer
Thousands of 18-core processors
Trend 2: Energy and Performance of Multicores

- Out-of-Order Superscalar Superpipelined
- Superscalar w/ Deeper Pipelines
- Multicore
- General-Purpose Manycores
Trend 2: Multicore Performance Scaling

<table>
<thead>
<tr>
<th>Year</th>
<th>Transistors (Thousands)</th>
<th>Frequency (MHz)</th>
<th>Typical Power (Watts)</th>
<th>MIPS R2K</th>
</tr>
</thead>
<tbody>
<tr>
<td>1975</td>
<td>10^7</td>
<td>10^3</td>
<td>10^0</td>
<td>DEC Alpha 21264</td>
</tr>
<tr>
<td>1980</td>
<td>10^6</td>
<td>10^4</td>
<td>10^1</td>
<td>MIPS R2K</td>
</tr>
<tr>
<td>1985</td>
<td>10^5</td>
<td>10^5</td>
<td>10^2</td>
<td>Intel Pentium 4</td>
</tr>
<tr>
<td>1990</td>
<td>10^4</td>
<td>10^6</td>
<td>10^3</td>
<td>AMD 4-Core Opteron</td>
</tr>
<tr>
<td>1995</td>
<td>10^3</td>
<td>10^7</td>
<td>10^4</td>
<td>Intel 48-Core Prototype</td>
</tr>
<tr>
<td>2000</td>
<td>10^2</td>
<td>10^8</td>
<td>10^5</td>
<td>AMD 4-Core Opteron</td>
</tr>
<tr>
<td>2005</td>
<td>10^1</td>
<td>10^9</td>
<td>10^6</td>
<td>Intel 48-Core Prototype</td>
</tr>
<tr>
<td>2010</td>
<td>10^0</td>
<td>10^10</td>
<td>10^7</td>
<td>AMD 4-Core Opteron</td>
</tr>
<tr>
<td>2015</td>
<td>10^{-1}</td>
<td>10^{11}</td>
<td>10^{8}</td>
<td>Intel 48-Core Prototype</td>
</tr>
</tbody>
</table>

Data partially collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond
Trend 3: Emerging Device Technologies

- Vertical MOSFETs
- Graphene
- Carbon Nanotubes
- Nanorelays
- Quantum Computing
- Molecular Computing
- Memristers
- Phase-Change Mem
- Spintronics
- 3D Integration
- Nanophotonics

Adapted from [Kurzweil'09]
Key trends in the application requirements and technology constraints over the past decade have resulted in a radical rethinking of the processors, memories, and networks used in modern computing systems.

Three Key Trends in Computer Engineering

1. Growing diversity in application requirements motivate growing diversity in computing systems

2. Energy and power constraints motivate transition to multiple processors integrated onto a single chip

3. Technology scaling challenges motivate new emerging processor, memory, and network device technologies
Activity #2: Sorting with a Parallel Processor

- **Application:** Sort 32 numbers

- **Simulated Parallel Computing System**
  - Processor: Group of 2–8 students
  - Memory: Worksheet, scratch paper
  - Network: Communicating between students

- **Activity Steps**
  1. Discuss strategy with group
  2. When instructor starts timer, master processor flips over worksheet
  3. Sort 32 numbers as fast as possible
  4. Lookup when completed and write time on worksheet
  5. *Master processor only* raises hand
  6. When everyone is finished, then analyze data
Activity #2: Discussion

Distribute
Sort 4 Numbers
Merge Phase 1
> merge 4+4 = 8
Merge Phase 2
> merge 8+8 = 16
Merge Phase 3
> merge 16+16 = 32

Algorithm
Communication
Load Balancing
Fault Tolerance
Dataset Size
Talk Outline

ECE Overview

What is Computer Engineering?

Trends in Computer Engineering

Computer Engineering Design
What do computer engineers actually do?

**General Science**
- Discover truths about nature
- Ask question about nature
  - Construct hypothesis
  - Test with experiment
  - Analyze results and draw conclusions

**Computer Engineering**
- Explore design space for a given system
- Design and build initial system
  - Ask question about system
  - Modify system or build/design alternative
  - Test with experiment to compare alternatives
  - Analyze results and draw conclusions
How do we design something so incredibly complex?

**Computer Engineering**

Explore design space for a given system

- Design and build initial system
- Ask question about system
- Modify system or build/design alternative
- Test with experiment to compare alternatives
- Analyze results and draw conclusions

**Fighter Airplane:** ~100,000 parts

**Intel Sandy Bridge E:** 2.27 Billion transistors
Design Principles

- Abstraction – Hide low-level details to create higher-level models
- Hierarchy – Structurally decompose design; e.g., net → router → queues
- Regularity – Structural and physical regularity; e.g., uniform tiles
- Modularity – Well-defined interfaces; e.g., latency insensitive net interface
- Encapsulation – Hide implementation details; e.g., processor microarch
- Extensibility – Design for future extensions; e.g., new network topo

Design Methodologies

- Incremental design
- Test-driven design
Incremental Design
Test-Driven Design

- **Test Types**
  - Unit tests
  - Directed vs. random tests
  - Whitebox vs. blackbox tests
  - Integration tests

- **Goal**
  - Write tests first then implement design to pass these tests

- **Write tests**
  - for higher level of abstraction, refine implementation until passes tests, add new tests

- **Capture design bugs** with new tests
Single-Core to Multi-Core Design Example

Computer Engineering

Explore design space for a given system

- Design and build initial system
  - Ask question about system
  - Modify system or build/design alternative
  - Test with experiment to compare alternatives
  - Analyze results and draw conclusions

Our "experiments" during the previous activities illustrate the design process
Take-Away Points

- ECE is a broad field focused on the study and application of electricity, microelectronics, and electro-magnetism.

- Computer engineering is the process of designing abstraction and implementation layers to meet application requirements within physical technology constraints.

- We are entering an exciting new era of computer engineering with emerging applications and systems, a remarkable shift towards mainstream parallel processing, and significant technology challenges.