Introduction to Computer Engineering

CS/ECE 252, Fall 2011
Prof. Guri Sohi
Computer Sciences Department
University of Wisconsin – Madison
Place On Desk

- IPod
- Laptop
- Treo
- Etc.

- All Computers
- Software/Hardware separation key
Computers!

• Engineers and scientists of all disciplines rely on computers for many aspects of their work
  – Not just word processing, spreadsheets, CAD, etc.
  – Computational methods, data mining, analysis/synthesis are fundamental to advances in many fields
• Many of the advanced techniques used in today’s microprocessors were invented right here at UW
• Some of the most renowned computer design researchers in the world are on our faculty
• There is a near-100% likelihood that a Wisconsin graduate helped design the computer or processor that you own
Place on Desk

- 7MB Disk Pack
- 6’ Disk
- IPod (80GB)
- (80GB/7MB = 11,000x)

- Computer useful & then 10,000x better!
$16 base; 60% growth

<table>
<thead>
<tr>
<th>Year</th>
<th>Salary</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$16</td>
<td>Base</td>
</tr>
<tr>
<td>3</td>
<td>$64</td>
<td>Still live at home</td>
</tr>
<tr>
<td>15</td>
<td>$16K</td>
<td>Buy car</td>
</tr>
<tr>
<td>24</td>
<td>$100K</td>
<td>Buy house</td>
</tr>
<tr>
<td>36</td>
<td>$300M</td>
<td>Need fundamentally new ways to spend money</td>
</tr>
</tbody>
</table>
Performance Growth

Unmatched by any other industry!
[John Crawford, Intel]

• Doubling every 18 months (1982-1996): 800x
  – Cars travel at 44,000 mph and get 16,000 mpg
  – Air travel: LA to NY in 22 seconds (MACH 800)
  – Wheat yield: 80,000 bushels per acre

• Doubling every 24 months (1971-1996): 9,000x
  – Cars travel at 600,000 mph, get 150,000 mpg
  – Air travel: LA to NY in 2 seconds (MACH 9,000)
  – Wheat yield: 900,000 bushels per acre
This Course

This course will:

• Help you understand the significance and pervasiveness of computers in today’s society and economy
• Teach you how computers really operate and how they are designed
• Introduce you to concepts that students in the Computer Engineering degree program learn in depth over four years
• Prepare and motivate you for study in this degree program
• Will count towards GCR introduction to engineering requirement
Go Over Web Page

http://www.cs.wisc.edu/~sohi/cs252/Fall2011/
Instructor & TAs
Textbook
Lecture Notes
Schedule
Computing and Simulator
Grading
Exams
Homework
Course Outline

• **Prerequisite** – none
• **Major topics in course**
  – Introduction to computers and computing
  – Information representation and manipulation
  – Logic elements and combinational Logic
  – Sequential Logic and Memory
  – Simple computer organization, design and operation
  – Machine language and instruction set architecture
  – Assembly language
  – Programming constructs
Advice

• **Textbook** — read BEFORE corresponding lecture

• **Homework** — completed in *study groups*
  – Will reinforce in-class coverage
  – Will help you prepare for midterm exams

• **Study Groups**
  – Groups of 2, should meet weekly, learn from each other
  – Review material, complete homework assignments
  – Each submitted homework should include consensus-based statement of work
Technology

• Technology advances at astounding rate
  – 19th century: attempts to build mechanical computers
  – Early 20th century: mechanical counting systems (cash registers, etc.)
  – Mid 20th century: vacuum tubes as switches
  – Since: transistors, integrated circuits

• 1965: Moore’s law [Gordon Moore]
  – Predicted doubling of capacity every 18 months
  – Has held and will continue to hold

• Drives functionality, performance, cost
  – Exponential improvement for 40 years
Applications

• Corollary to Moore’s Law: Cost halves every two years
  In a decade you can buy a computer for less than its sales tax today. –Jim Gray

• Computers cost-effective for
  – National security – weapons design
  – Enterprise computing – banking
  – Departmental computing – computer-aided design
  – Personal computer – spreadsheets, email, web
  – Pervasive computing – prescription drug labels

• Countless industries revolutionized
## Some History

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1947</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; transistor</td>
<td>Bell Labs</td>
</tr>
<tr>
<td>1958</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; IC</td>
<td>Jack Kilby (MSEE ’50) @TI Winner of 2000 Nobel prize</td>
</tr>
<tr>
<td>1971</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; microprocessor</td>
<td>Intel (calculator market)</td>
</tr>
<tr>
<td>1974</td>
<td>Intel 4004</td>
<td>2300 transistors</td>
</tr>
<tr>
<td>1978</td>
<td>Intel 8086</td>
<td>29K transistors</td>
</tr>
<tr>
<td>1989</td>
<td>Intel 80486</td>
<td>1M transistors</td>
</tr>
<tr>
<td>1995</td>
<td>Intel Pentium Pro</td>
<td>5.5M transistors</td>
</tr>
<tr>
<td>2006</td>
<td>Intel Montecito</td>
<td>1.7B transistors</td>
</tr>
</tbody>
</table>
Abstraction and Complexity

• Abstraction helps us manage complexity

• Complex interfaces
  – Specify what to do
  – Hide details of how

• Goal: Use abstractions yet still understand details
Computer As a Tool

• Many computers today are embedded
  – Fixed functionality
  – Appliance-like
  – Not really programmable by end user

• Not the focus of this course!
  – Instead, programmable computers
  – Learn to think of computer as a tool

• Program?
  – Algorithm or set of steps that computer follows
  – Human brains wired to work this way