Introduction

This course is all about how computers work

But what do we mean by a computer?

- Different types: desktop, servers, embedded devices
- Different uses: automobiles, graphics, structural analysis, finance, genomics…
- Different manufacturers: Intel, Apple, IBM, Microsoft, Sun…
- Different underlying technologies and different costs!
**Pedagogy**

Analogy: Consider a course on “automotive vehicles”
- Many similarities from vehicle to vehicle (e.g., wheels)
- Huge differences from vehicle to vehicle (e.g., gas vs. electric)

Best way to learn:
- Focus on a specific instance and learn how it works
- While learning general principles and historical perspectives

**Why learn this stuff?**

- You want to call yourself a “computer scientist”
- You want to build software people use (need performance)
- You need to make a purchasing decision or offer “expert” advice

Debugging skills often benefit from understanding architecture
- better understand system error messages
- better understand translators (compilers and interpreters)

Both hardware and software affect performance:
- Algorithm determines number of source-level statements
- Language/Compiler/Architecture determine machine instructions (Chapters 2 and 3)
- Processor/Memory determine how fast instructions are executed (Chapters 5, 6, and 7)

Assessing and Understanding Performance in Chapter 4
What is a computer?

Components:
- input (mouse, keyboard)
- output (display, printer)
- memory (disk drives, DRAM, SRAM, CD)
- network

Primary Focus

Our primary focus: the processor (datapath and control)
- implemented using millions of transistors
- Impossible to understand by looking at each transistor
- We need...
Abstraction

Delving into the depths reveals more information

An abstraction omits unneeded detail, helps us cope with complexity

What are some of the details that appear in these familiar abstractions?

```
swap(int v[], int k) {
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

High-level language (C)

Compiler

Assembly language (MIPS)

```
swap:
    mul $2, $5, 4
    add $2, $4, $2
    lw $15, 0($2)
    lw $16, 4($2)
    sw $16, 0($2)
    sw $16, 4($2)
    jr $31
```

Assembler

Binary machine language (for MIPS)

How do computers work?

Need to understand abstractions such as:
- Applications software
- Systems software
- Assembly Language
- Machine Language
- Architectural Issues: i.e., Caches, Virtual Memory, Pipelining
- Sequential logic, finite state machines
- Combinational logic, arithmetic circuits
- Boolean logic, 1s and 0s
- Transistors used to build logic gates (CMOS)
- Semiconductors/Silicon used to build transistors
- Properties of atoms, electrons, and quantum dynamics

So much to learn!
Why learn this stuff?

Assembly language programming
- implementation of low-level system software
- on RISC architectures, good optimizing compilers are usually better than humans
- does that include GCC?

MIPS architecture
- clean, clear example of RISC architectures
- used on some common, contemporary platforms

True or False: Binary compatibility is extraordinarily important?

Modern instruction set architectures:
- IA-32, PowerPC, MIPS, SPARC, ARM, and others

Instruction Set Architecture
History: In the Beginning…

ENIAC: *Electronic Numerical Integrator and Calculator*
- <=1945 (military application)
- 80ft x 8.5 ft x 3ft
- 18,000 vacuum tubes
- 1900 adds/second
- manual programming (really)
- data via punch cards

EDVAC: *Electronic Discrete Variable Automatic Computer*
- Eckert, Mauchly, von Neumann (Penn and the Institute for Advanced Study)
- 1952 (Eckert & Mauchly left in a patent dispute)

History: the Stored Program Computer

EDSAC: *Electronic Delay Storage Automatic Calculator*
- Wilkes (Cambridge)
- first operational, full-scale, stored-program computer (1949)
History: Commercialization

UNIVAC I (1951)
- Eckert & Mauchly
- Remington-Rand
- first commercially successful system
- built 48
- sold at $1,000,000 a pop

System/360 family (1964)
- IBM
- dominated the "big-iron" market
- comprehensive hardware/software solution
- sold at $1,000,000 a pop
- eventually fought US government for years over market practices

History: Mini and Super-size

PDP-8 (1965)
- Digital Equipment Corp
- first successful minicomputer
- sold at under $20,000 a pop
- arguably motivated development of microprocessors

CDC 6600 (1963)
- Seymour Cray
- led to Cray Computers
- Cray-1 shown at right (1976)
- sold at $4,000,000 a pop
History: Populating the Desktop

PDP-8 (1965)
- Xerox
- bit-mapped graphics
- window-based user interface
- mouse
- local-area network connection
- several hundred built
- many donated to universities

50 Years of Progress

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1964</td>
<td>IBM 7040</td>
<td>1.0 x 1.0</td>
<td>250</td>
<td>10,000</td>
<td>84</td>
<td>$1,000,000</td>
<td>1</td>
<td>$1,017,690</td>
<td>1</td>
</tr>
<tr>
<td>1965</td>
<td>IBM 7090</td>
<td>1.0 x 1.0</td>
<td>200</td>
<td>10,000</td>
<td>64</td>
<td>$1,000,000</td>
<td>263</td>
<td>$4,762,330</td>
<td>1</td>
</tr>
<tr>
<td>1970</td>
<td>IBM 360-75</td>
<td>1.0 x 1.0</td>
<td>200</td>
<td>10,000</td>
<td>64</td>
<td>$1,000,000</td>
<td>1,000,000</td>
<td>$12,000,000</td>
<td>1</td>
</tr>
<tr>
<td>1975</td>
<td>IBM 3033</td>
<td>1.0 x 1.0</td>
<td>200</td>
<td>10,000</td>
<td>64</td>
<td>$1,000,000</td>
<td>1,000,000</td>
<td>$12,000,000</td>
<td>1</td>
</tr>
<tr>
<td>1980</td>
<td>IBM 3081</td>
<td>1.0 x 1.0</td>
<td>200</td>
<td>10,000</td>
<td>64</td>
<td>$1,000,000</td>
<td>1,000,000</td>
<td>$12,000,000</td>
<td>1</td>
</tr>
<tr>
<td>1985</td>
<td>IBM 3083</td>
<td>1.0 x 1.0</td>
<td>200</td>
<td>10,000</td>
<td>64</td>
<td>$1,000,000</td>
<td>1,000,000</td>
<td>$12,000,000</td>
<td>1</td>
</tr>
<tr>
<td>1990</td>
<td>IBM 3090</td>
<td>1.0 x 1.0</td>
<td>200</td>
<td>10,000</td>
<td>64</td>
<td>$1,000,000</td>
<td>1,000,000</td>
<td>$12,000,000</td>
<td>1</td>
</tr>
<tr>
<td>1995</td>
<td>IBM 3090</td>
<td>1.0 x 1.0</td>
<td>200</td>
<td>10,000</td>
<td>64</td>
<td>$1,000,000</td>
<td>1,000,000</td>
<td>$12,000,000</td>
<td>1</td>
</tr>
<tr>
<td>2000</td>
<td>IBM 3090</td>
<td>1.0 x 1.0</td>
<td>200</td>
<td>10,000</td>
<td>64</td>
<td>$1,000,000</td>
<td>1,000,000</td>
<td>$12,000,000</td>
<td>1</td>
</tr>
</tbody>
</table>

©William D McQuain, January 2005
## Units of Measurement

### Spatial units:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte (B)</td>
<td>8 bits</td>
</tr>
<tr>
<td>kilobyte (KB)</td>
<td>1024 or $2^{10}$ bytes</td>
</tr>
<tr>
<td>megabyte (MB)</td>
<td>1024 kilobytes or $2^{20}$ bytes</td>
</tr>
<tr>
<td>gigabyte (GB)</td>
<td>1024 megabytes or $2^{30}$ bytes</td>
</tr>
</tbody>
</table>

**Traditional**

<table>
<thead>
<tr>
<th>Unit</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte (B)</td>
<td>8 bits</td>
</tr>
<tr>
<td>kilobyte (KB)</td>
<td>1000 or $10^3$ bytes</td>
</tr>
<tr>
<td>megabyte (MB)</td>
<td>1000 kilobytes or $10^6$ bytes</td>
</tr>
<tr>
<td>gigabyte (GB)</td>
<td>1000 megabytes or $10^9$ bytes</td>
</tr>
</tbody>
</table>

**Secondary storage**

<table>
<thead>
<tr>
<th>Unit</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte (B)</td>
<td>8 bits</td>
</tr>
<tr>
<td>kibibyte* (KiB)</td>
<td>1024 or $2^{10}$ bytes</td>
</tr>
<tr>
<td>mebibyte (MiB)</td>
<td>1024 kibibytes or $2^{20}$ bytes</td>
</tr>
<tr>
<td>gibibyte (GiB)</td>
<td>1024 mebibytes or $2^{30}$ bytes</td>
</tr>
</tbody>
</table>

**IEC standard**

* *Kilo binary byte*