What is Computer Architecture?

- **Hardware Designer**
  - thinks about circuits, components, timing, functionality, ease of debugging
  - "construction engineer"

- **Computer Architect**
  - thinks about high-level components, how they fit together, how they work together to deliver performance.
  - "building architect"

Why do I care?

- You may actually do computer architecture someday
- You may actually care about software performance someday
  - The ability of application programs, compilers, operating systems, etc. to deliver performance depends critically on an understanding of the underlying computer organization.
  - That becomes more true every year.
  - Computer architectures become more difficult to understand every year.

Which is faster?

```
for (i=0; i<N; i=i+1)
for (j=0; j<N; j=j+1) {
    r = 0;
    for (k=0; k<N; k=k+1)
        r = r + y[i][k] * z[k][j];
    x[i][j] = r;
}
```

```
for (ij=0; jj<N; jj=jj+B)
for (kk=0; kk<N; kk=kk+B)
for (i=0; i<N; i=i+1) {
    for (jj=jj; jj<min(jj+B-1,N); jj=jj+1)
        r = 0;
        for (kk=kk; kk<min(kk+B-1,N); kk=kk+1)
            r = r + y[i][k] * z[k][j];
        x[i][jj] = x[i][jj] + r;
}
```
Which is faster?

load R1, addr1
store R1, addr2
add R0, R2 -> R3
subtract R4, R3 -> R5
add R0, R6 -> R7
store R7, addr3

load R1, addr1
add R0, R2 -> R3
add R0, R6 -> R7
store R1, addr2
subtract R4, R3 -> R5

Which is faster?

loop1: add ...
load ...
add ...
bne R1, loop1
loop2: add ...
load ...
bne R2, loop2

top:

load ...
add ...
bne R1, loop1

nop

load ...
bne R2, loop2

Administration

• Instructor -- Dr. Dean Tullsen
• TAs:
  – Vineet Kumar
  – Alper Mizrak
• grading
• integrity
• the course workload

What is Computer Architecture?

Computer Architecture =
Machine Organization + Instruction Set Architecture

What the machine looks like

How you talk to the machine
How to Speak Computer

<table>
<thead>
<tr>
<th>High Level Language Program</th>
<th>Assembly Language Program</th>
<th>Machine Language Program</th>
<th>Machine Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compiler</td>
<td>Assembler</td>
<td></td>
<td>Control Signal Spec</td>
</tr>
</tbody>
</table>

```
lw $15, 0($2)
lw $16, 4($2)
sw $16, 0($2)
sw $15, 4($2)
```

temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;

ALUOP[0:3] <= InstReg[9:11] & MASK

The Instruction Set Architecture

- that part of the architecture that is visible to the programmer
  - opcodes (available instructions)
  - number and types of registers
  - instruction formats
  - storage access, addressing modes
  - exceptional conditions

The Instruction Set Architecture

° is the agreed-upon interface between all the software that runs on the machine and the hardware that executes it.

Examples of ISAs

- Alpha AXP
- Intel 80x86/pentium
- VAX
- MIPS
- SPARC
- IBM 360
- Intel IA-64 (Itanium)
- PowerPC
Computer Organization

- Once you have decided on an ISA, you must decide how to design the hardware to execute those programs written in the ISA as fast as possible.
- This must be done every time a new implementation of the architecture is released, with typically very different technological constraints.

The Challenge of Computer Architecture

- The industry changes faster than any other.
- The ground rules change every year.
  - new problems
  - new opportunities
  - different tradeoffs
- It’s all about making programs run faster than the next guy’s machine.

Performance Trends

Technology: Microprocessor Logic Density
The five classic components of computers

- Computer
- Memory
- Datapath
- Control
- Input
- Output

Course Outline

I. Instruction Set Architecture
II. Computer System Performance and Performance Metrics
III. Computer Arithmetic and Number Systems
IV. CPU Architecture
V. Pipelining
VI. Superscalars
VII. The Memory/Cache Hierarchy
VIII. Parallel Machines

What you can expect to get out of this class

- to become conversant with computer architecture terms and concepts.
- to understand fundamental concepts in computer architecture and how they impact computer and application performance.
- to be able to read and evaluate architectural descriptions of even today’s most complex processors.
- to gain experience designing a working CPU completely from scratch.
- to learn experimental techniques used to evaluate advanced architectural ideas.

Key Points

- High-performance software requires a deep understanding of the underlying machine organization.
- The instruction set architecture defines how software is allowed to use the processor. Multiple computers with vastly different organizations and performance can share an ISA.
- Most every component in a computer system falls into one of five categories.