CSE 240 -- Principles of Computer Architecture

Fall Quarter, 1999
Dean Tullsen

UCSD

Computer Architecture is... ???

- Computer Architect (building architect)
  - high-level design
  - organization
  - functionality
  - performance

- Hardware Designer (builder, construction engineer)
  - materials
  - implementation details

the first computer science discipline
Eckert and Mauchly, the first computer scientists, were computer architects, as was John von Neumann and Maurice Wilkes
Computer Architecture is... ???

- Instruction Set Design
- Computer Organization

That part of the machine that is visible to the user (programmer/compiler-writer/OS writer/user)
  - the software interface (Instruction Set Design)
  - performance (Computer Organization)

PERFORMANCE!!!!!!

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?

```c
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;
    }
```

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

Which is faster?

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

The Forces on Computer Architecture

- Technology
- Parallelism
- Programming Languages
- Applications
- Operating Systems
- History
- Computer Architecture:
  - Instruction Set Design
  - Hardware Organization
What is Computer Architecture?

Computer Architecture = Machine Organization + Instruction Set Architecture

What the machine looks like

How you talk to the machine

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

Examples of ISAs

- Alpha AXP
- 80x86/pentium
- VAX
- MIPS
- SPARC
- IBM 360
- ...

Who are you?
- Who am I? Instructor -- Dr. Dean Tullsen

syllabus, other details
The Instruction Set Architecture

- ISA provides a level of abstraction for both the hardware and the software

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

- This 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
### Course Outline

I. Introduction  
II. Computer System Performance  
III. Instruction Set Architecture  
IV. Pipelining  
V. Instruction-Level Parallelism  
VI. The Memory/Cache Hierarchy  
VII. 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 learn experimental techniques used to evaluate advanced architectural ideas. In other words, to do architecture research!

### Key Points

- Computer Architecture defines the software-visible machine description (ISA) and the overall machine organization.  
- 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.  
- We can’t improve performance unless we can quantify it, model it, and argue about it.