CSE 240 -- Principles of Computer Architecture

Spring, 1997
Dean Tullsen

Computer Architecture is... ???

- Computer Architect (building architect)
  - high-level design
  - organization
  - functionality
  - performance
- Hardware Designer (builder, construction engineer)
  - materials
  - implementation details

Computer Architecture is... ???

- the first computer science discipline
- Eckert and Mauchly, the first computer scientists, were computer architects, as was John von Neumann

---

Computer Architecture is... ???

- Instruction Set Design
- Computer Organization

Computer Architecture is... ???

- 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)

Computer Architecture is... ???

PERFORMANCE!!!!!

Why do I care?

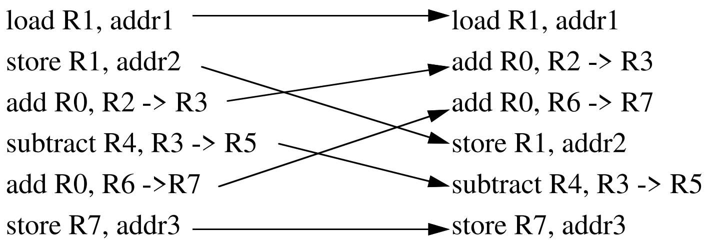
- **You may actually do computer architecture someday**
- **You may actually care about software performance someday**
  - The ability of application programs, compilers, operating systems, etc. to deliver performance depends critically on an understanding of the underlying computer organization.
  -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

Which is faster?

```
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 (jj=0; jj<N; jj=jj+B)
for (kk=0; kk<N; kk=kk+B)
for (i=0; i<N; i=i+1) {
   for (j=jj; j<min(jj+B-1,N); j=j+1)
     r = 0;
      for (k=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?



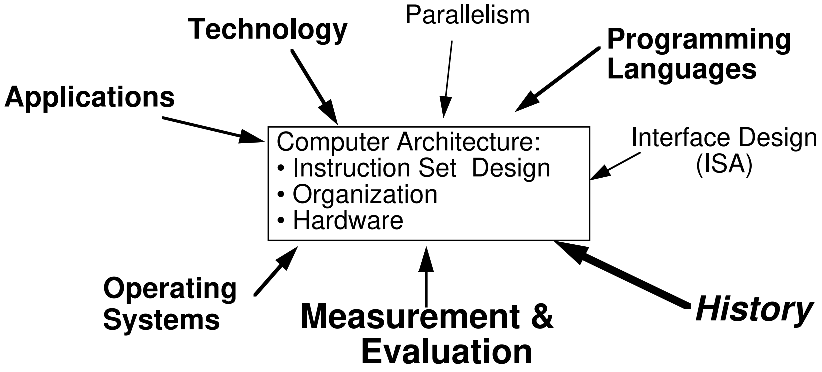
load R1, addr1
store R1, addr2
add R0, R2 -> R3
subtract R4, R3 -> R5
add R0, R6 ->R7
store R7, addr3

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

Which is faster?

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

The Forces on Computer Architecture



Computer Architecture:
- Instruction Set Design
- Organization
- Hardware

Administration

- Who am I? Instructor -- Dr. Dean Tullsen
- Who are you?
- grading comments
- workload comments
- my expectations

What is Computer Architecture?

Computer Architecture =
Machine Organization +
Instruction Set Architecture

What the machine looks like

How you talk to the machine

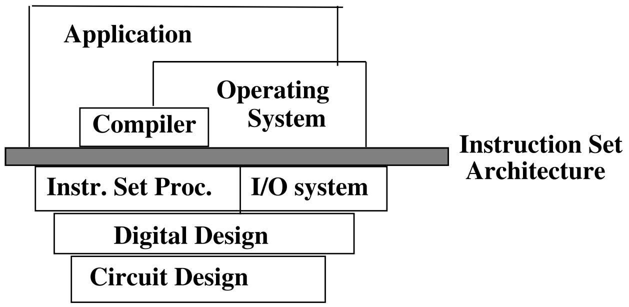
The Instruction Set Architecture

- 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

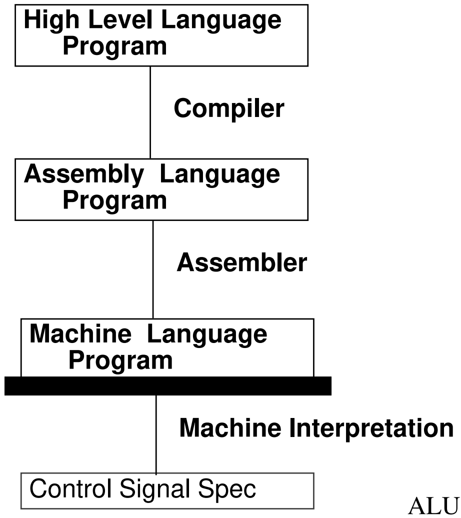
- Alpha AXP
- 80x86/pentium
- VAX
- MIPS
- SPARC
- IBM 360
- ...

The Instruction Set Architecture



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

How to Speak Computer



High Level Language Program

Compiler

Assembly Language Program

Assembler

Machine Language Program

Machine Interpretation

Control Signal Spec

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

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

```
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
```

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

Machine Organization

- So, if the ISA is an abstraction that hides the details of the machine from the programmer, compiler-writer, system programmer, etc., do we really care what the underlying machine organization is?

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.

Processor Performance

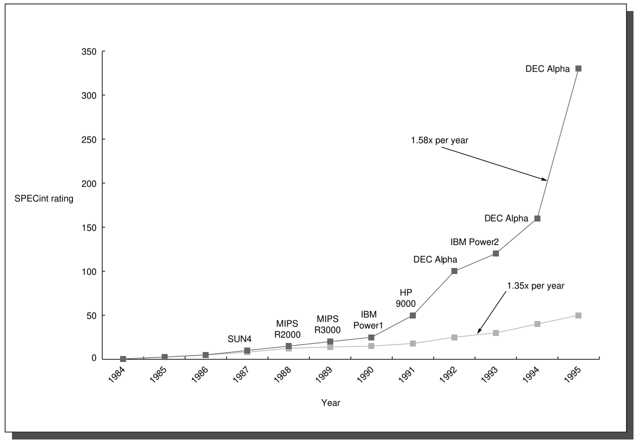


FIGURE 1.1 Growth in microprocessor performance since the mid 1980s has been substantially higher than in earlier years.

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.