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.

Administration

- Instructor -- Dr. Dean Tullsen
- Who are you?
- TAs:
  - Leo Porter
  - ??
- 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

High Level Language Program

Assembly Language Program

Machine Language Program

Control Signal Spec

Compiler

Assembler

Machine Interpretation

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
1000110011110010000000000001001010110011110010000000000000000
101011000110001000000000000100

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

The Instruction Set Architecture

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

Applications

Compiler

Operating System

Instr. Set Proc.

I/O system

Instruction Set Architecture

Digital Design

Circuit Design
The Instruction Execution Cycle

- **Fetch**
  - Obtain instruction from program storage

- **Decode**
  - Determine required actions and instruction size

- **Operand Fetch**
  - Locate and obtain operand data

- **Execute**
  - Compute result value or status

- **Result Store**
  - Deposit results in storage for later use

- **Next Instruction**
  - Determine successor instruction

**Key ISA decisions**

- operations
  - how many?
  - which ones

- operands
  - how many?
  - location
  - types
  - how to specify?

- instruction format
  - size
  - how many formats?

**Examples of ISAs**

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

**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 (or as cheaply as possible, or using as little power 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. Or more efficiently.
The five classic components of computers

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.
• The era of single-threaded computing is over. We now enter the world of multi-core architectures. But core design hasn’t changed that much.
• Most every component in a computer system falls into one of five categories.