EE 4720—Computer Architecture

01-1

Call Number 1726 (Spring 2003)

URL: http://www.ece.lsu.edu/ee4720

Offered by:

David M. Koppelman

349 EE Building

578-5482, koppel@ece.lsu.edu, http://www.ece.lsu.edu/koppel

Tentative office hours: Mon 15:00-16:00; Wed 9:30-10:30; Tue & Thr 14:00-15:30.

Should already know:

How to design a computer.

Will learn:

How to design a *good* computer.

01-2

Prerequisites By Course:

EE 3755, Computer Organization.

Prerequisites By Topic:

- Logic design.
- Computer organization.
- Assembly-language programming.

Text

"Computer architecture, a quantitative approach," John L. Hennessy & David A. Patterson, or "Computer organization & design," David A. Patterson & John L. Hennessy.

01-3

Course Content

- Instruction set design.
- Pipelined processor design.
- Multiple-issue processor design.
- Caches and memory.

Course content will follow text.

Graded Material

01-4

Midterm Exam, 40%

Fifty minutes, closed book.

Final Exam, 40%

Two hours, closed book.

Yes, it's cumulative.

Homework, 20%

Written and computer assignments.

Lowest grade or unsubmitted assignment dropped.

Course Usefulness

01-5

Material in Course Needed For:

- General-purpose processor designers and testers.
- Special-purpose processor designers and testers.
- Compiler writers.
- Programmers of high-performance systems.
- Answering job interview questions.

Material Covered in This Set

01-6

Coverage: Sections 1.2 and 1.3.

Slides and other material via http://www.ece.lsu.edu/ee4720

Web site also has homework assignments, exams, grades, and other material.

ISA and Implementation Distinction

01-7

What is a computer?

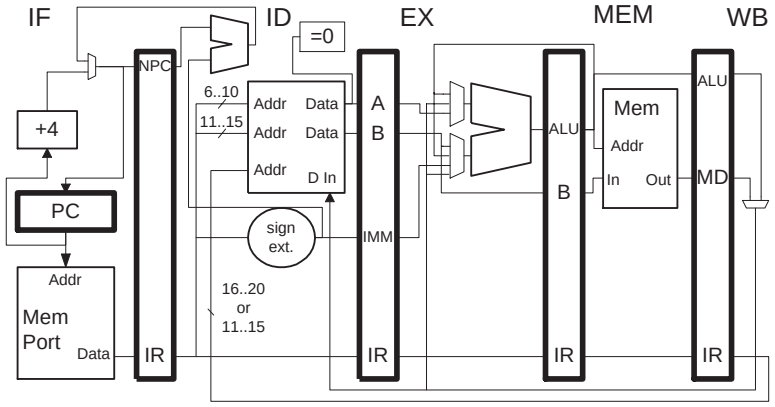
A machine that executes instructions which read and write memory.

What a computer engineer does:

- Develops an *instruction set architecture* (ISA):

```
ld  r1, 0(r2)  ! Load register r1 with contents of mem at r2.
add r3, r1, r4
sw  0(r2), r3
```

- Designs hardware to execute, *implement*, the instruction set:



01-7

01-8

Definitions

Instruction Set Architecture (ISA):
Precise definition of computer's instructions and their effects.

- It's all programmer needs to program machine.
- It's all hardware designer needs to design machine.

Implementation: [of an ISA] (noun)
Hardware that executes instructions defined by ISA.

Some Architectures and their Implementations

01-9

Architecture: IBM *System/360*

Developed in 1964 for large business computers.

Designers appreciated and popularized the difference between architecture and implementation.

First planned family of computers.

Very successful, successor machines still in use.

First Implementations: Model 30, Model 75.

01-10

Architecture: Intel *IA-32*

Initially developed in 1978 for small systems.

First processor: 8086, implements small part of IA-32.

Major improvements in amount of memory addressable by subsequent chips, 80186, 80286 (1982).

The 80386 (1985) could host a modern 32-bit operating system.

Later chips implemented ISA extensions for multimedia and data movement ...
... and continued to incorporate microarchitectural innovations.

80486 (1989), Pentium (1992), Pentium Pro (1995), Pentium II (1997), Pentium III (1999) Pentium 4 (2000).

Unlike System/360, the way it would be used was not forseen.

Includes unpopular features, such as small memory segments.

Nevertheless, implementations have competed well with modern ISAs.

EE 4720 Lecture Transparency. Formatted 11:48, 24 January 2003 from lsli01.
01-10

01-11

Architecture: DEC (then Compaq, now HP) *Alpha*

An example of a *RISC* processor.

Designed for easy programming.

Designed for easy implementation.

RISC programs are larger than others, but run faster.

Developed for a 25-year lifetime.

First implementation: DECchip 21064 (1992).

Later implementations: 21264, 21364.

Implementations are usually the fastest processors.

Alas, Compaq plans to discontinue it.

01-11

01-12

Architecture: *Itanium* (IA-64)

First general purpose *VLIW* ISA.

ISA helps processor overcome problems in turn-of-the-century processors.

First implementation: Itanium (same name as architecture) (2000).

Radically different from other processors.

So far unproven.

01-12

01-13

Who ISA Developed For

- Compiler writers.
- Compute-intensive library writers.
  E.g., graphics and scientific libraries.

Instruction set requirements don't change very much over time.

Scope of ISA Specification

Describes instruction codings, and what they should do.

Should specify action of all codings, used or not ...

Implementation

01-14

Two aspects of implementation: *organization* and *hardware*.

Organization:
Details of functional units, data paths, control, etc.

Also called *microarchitecture*.

This includes memory system, bus, and CPU.

Technology (Hardware):
Logic design and packaging.

Course focus: ISA and organization, not hardware.

01-14

Principles of Computer Design

01-15

Principles computer designers apply widely.

- Make the common case fast.
  Obviously.

  (Not covered.)

- *Amdahl's Law:* Don't make common case too fast.
  As speed of one part increases...
  ...impact on total performance drops.

  (Not covered.)

- *Locality of Reference.*
  Temporal: It might happen again soon.
  Spatial: It might happen to your neighbors soon too.

Locality

01-16

The first two principles are "common sense".

However, locality is a characteristic of executing programs ...

... which has held and is expected to continue to hold.

Because many designs work only when locality is present ...

... if it all of a sudden programs did not exhibit locality ...

... computers would run them many, many times slower!

Locality usually applied to memory addresses issued by processor.

Temporal: there's a good chance that an address used will be used again soon.

Spatial: once an address is used there's a good chance a nearby address will be used.

For examples, analyze execution of almost any program.

Summary

01-17

What a computer engineer does:

- Develops an *instruction set* (ISA).
- Designs hardware to execute instruction set.