EE 4720—Computer Architecture

01-1

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

RSS: http://www.ece.lsu.edu/ee4720/rss_home.xml

Offered by:

David M. Koppelman

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

Office Hours: Monday - Friday, 9:00-10:00.

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.

Optional 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.

Topics cover modern general-purpose microprocessors.

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.

EE 4720 Lecture Transparency. Formatted 10:34, 27 August 2007 from lsli01.

01-4

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.

01-6

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

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

Announcements are on course home page and available as an RSS (Web) Feed.

Instruction Set Architecture v. Implementation

01-7

Quick (and Temporary) Definitions

Instruction Set Architecture (ISA):
The Machine Language
Examples: IA-32, Itanium, PowerPC.

Implementation:
The Chip
Examples: Intel Core, Athlon, G5.

An ISA can have multiple implementations . . .
. . . sometimes developed decades apart.

01-8

ISA Concept History

Early Computer Design

ISA and Implementation not considered totally separate things.

Start with requirements. (*E.g.*, Need division instructions.)

Develop processor hardware.

ISA is documentation of "implementation" (developed hardware).

Problems Faced by Early Computer Manufacturers

Realization that software development expensive & time-consuming.

Would like to sell customers better machine that runs old software.

Engineers say a really better machine would break old software.

Marketing people rather not test customers' loyalty, so don't break sw.

01-9

ISA and IBM System/360

Concept of ISA crystallized by Amdahl, Blaauw, & Brooks.

Engineers designed ISA and implementation as separate entities.

ISA would ease development of new implementations.

System/360 developed in 1964 and its successors are still available.

Some Architectures and their Implementations

01-10

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 under name z/Architecture.

First Implementations: Model 30, Model 75.

01-11

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), Core Duo (2006).

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.

01-12

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.

01-13

Architecture: *Itanium* (née 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.

01-13

Instruction Set Architecture

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 the ISA.

Microarchitecture:
Organization and features used for an implementation.

ISA and implementation descriptions at
`http://www.ece.lsu.edu/ee4720/reference.html`.

What a Typical ISA Defines

Instructions. (Operations, encoding, etc.)

Registers and Memory Organization.

Interrupts, exceptions, and traps.

01-15

ISA Design Goals

Define what program is. (Instructions and their encodings.).

Define what program will do. (Data types, operations.)

Clearly distinguish defined, undefined, and implementation-dependent behavior.

Enable good first implementation.

Enable good future implementations.

01-15

01-16

ISA Undefined Behaviors

Goal: Future ISA extensions, reliable programs.

Behavior for all encodable opcodes.

Example: Error if opcode undefined.

This enables future expansion.

Behavior for unusual register values.

Example: Register value written when operation overflows.

This Course

01-17

This Course

ISA Features and Families

- Reference ISA: MIPS 32
- Families: RISC, CISC, VLIW
- Features: Common to the semi-exotic.
- Specific: SPARC, Itanium, VAX, PowerPC

Microarchitecture (Implementation Techniques)

- Basic Pipelining
- Multiple Issue (Superscalar)
- Branch Prediction
- Dynamic Scheduling (Out-of-order execution.)
- Memory and Caches