EE 4720—Computer Architecture

URL: `https://www.ece.lsu.edu/ee4720/`

RSS: `https://www.ece.lsu.edu/ee4720/rss_home.xml`

Offered by:

David M. Koppelman

3316R P. F. Taylor Hall, 578-5482, `koppel@ece.lsu.edu`, `https://www.ece.lsu.edu/koppel`

Office Hours: Monday - Friday, 14:00-15:00.

This Course, in a Nutshell:

Should already know:

How to design a computer.

Will learn:

How to design a *good* computer.

Prerequisites By Course:

- EE 3752, Microprocessor Systems.
- 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.

Course Content

Topics that cover modern general-purpose microprocessors:

- CPU pipelined implementations.
- Instruction set families: RISC ($\longleftarrow$ normal), CISC ($\longleftarrow$ legacy), VLIW ($\longleftarrow$ still niche).
- Code optimization and performance.
- Benchmarks, and when to believe them.
- Implementing interrupts and floating point operations.
- Single-core parallelism: pipelining, superscalar, vector instructions.
- Basic parallel computer organizations: multi-core, cluster.
- Branch prediction and related techniques.
- Dynamic scheduling.
- Caches and memory.

Graded Material

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

Material in Course Needed For:

- General-purpose processor designers and testers.
- Special-purpose processor designers and testers.
- Compiler writers.
- Programmers of high-performance libraries, including those for games and machine learning.
- Developing exploits for low-level vulnerabilities.
- Answering job interview questions.

Course Resources

- Slides and other material via `https://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 a Web (RSS) Feed.

Instruction Set Architecture (ISA) Concept History

Key Concept: Split computer design into *Architecture Design* and *Implementation*.

Quick (and Temporary) Definitions

Instruction Set Architecture (ISA):

The Machine Language

Examples: Intel 64, ARM A64, RISC-V, Power, VAX.

Implementation:

The Chip

Examples: Xeon 5418 Sapphire Rapids, Ryzen Threadripper 5965WX, Cortex-X1 (Exynos 2100), Apple M2.

An ISA can have multiple implementations . . .

. . . sometimes developed decades apart.

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).

Issues Discovered by Early Computer Manufacturers

Realization that software development expensive & time-consuming.

Would like to offer past customers better machine that runs old software.

Engineers naïvely argue that to improve hardware need to break old software.

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

Note: IBM is run by salesman.

Modern ISA-then-Implementation Approach

Concept of ISA as distinct from implementation . . .
. . . crystallized by Amdahl, Blaauw, & Brooks [1] . . .
. . . when developing the IBM System/360

IBM wanted to develop implementations for different market segments . . .
. . . such as entry-level, mid-range, etc.

They wanted a customer who bought and wrote software for an entry-level machine . . .
. . . to be able to upgrade to a mid-range machine . . .
. . . and run their software unmodified.

They also wanted that customer's software to run unmodified . . .
. . . on systems purchased years later.

To achieve this they carefully defined the System/360 ISA . . .
. . . and made sure that all systems implemented this ISA.

System/360 developed in 1964 and its successors are still available . . .
. . . even in the year 2021.

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

It's all that programmer needs to program machine.

It's all that 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
`https://www.ece.lsu.edu/ee4720/reference.html`.

What a Typical ISA Defines

Instructions. (Operations, encoding, etc.)

Data Formats (Integer, Floating Point, Vector/Packed)

Registers and Memory Organization.

Interrupts, exceptions, and traps.

Implementation-Dependent Features. (Memory control, custom features.)

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.

  Easy.

- Enable good future implementations.

  Requires foresight: See future needs and technology (*e.g.*, gates).

  Requires discipline: Sacrifice first-implementation performance.

  And maybe foolishness: Insufficient first-implementation sales.

Examples of Bad (now) ISA Features

- Undefined FP rounding behavior. System/360
- Base + 16-Bit Offset Addressing. (IA-32)
- Not enough registers. (IA-32)
- Overly complex instructions. (VAX)

Bad features can be overcome by implementation. . .
. . . especially if engineering budget is very large.

Some Architectures and their Implementations

Currently In Use and Going Strong

- 1978- (IA-32), Intel 64 (x86) — Very successful, by accident.
- 1985- (ARM A32) ARM A64 — Slow but steady rise, now very successful
- 1990- Power — Used in scientific computing.
- 2011- RISC-V — For research and teaching, increasingly used for embedded applications.

Historically Important

- 1964- IBM System/360 — Very successful *mainframe computers*.
- 1978- DEC VAX — Very successful *minicomputers*.
- 1992- DEC Alpha — Great product, business flop.
- 2000- Intel Itanium — Maybe too bloated, business flop.

Architecture: IBM *System/360[1]*

Developed in 1964 for large business computers.

Designers appreciated and popularized the difference between ISA and implementation.

First planned family of computers.

Very successful, successor machines still in use under name z/Architecture.

First Implementations: Model 30, Model 75.

Recent Implementation: z15.

Architecture: DEC *VAX*

Developed by the Digital Equipment Corporation.

First implementation: VAX 11/780, introduced in 1978.

Came to dominate *minicomputer* market in 1980s.

Single-chip implementations were developed but not successful.

Later DEC developed the Alpha ISA, more suitable to 1990s technology.

Architecture: Intel *IA-32* and *Intel 64*

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), Core i7 6gen (2015), Core i9-9980XE (2018), Xeon W-3345 (Ice Lake) (2021) . . .

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.

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.

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.

Did not succeed.

Architectures: *ARMv7* and substantially different *ARMv8*

Designed for personal computers by Acorn Computers in mid 1980s.

Later became popular for embedded systems, and after that general use.

Two major variants: A32, A64.

A32 has many quirky features such as making PC an integer register.

A64 is more conventional and is being adopted in mass-market devices.

Used in many *System-on-Chip (SoC)* designs, including those by Apple.

Becoming dominant.

Architecture(s): *RISC-V[2]*

Developed in 2011 for teaching and research, but complete enough for commercial use.

Used in Hennessy/Patterson computer organization texts starting with the x'th edition.

Relatively small number of instructions . . .
. . . especially compared to ISAs that have gone through decades of development.

Defined in terms of base architectures and extensions.

A base architecture is very simple, for example lacking multiplication and any floating-point instructions.

Extensions add particular capabilities, including multiplication, floating point, and vector instructions.

This Course

ISA Features and Families

- Reference ISA: MIPS 32
- Families: RISC, CISC, VLIW
- Features: Common to the semi-exotic.
- Specific: SPARC, RISC-V, IA-32/Intel 64, Itanium, VAX, Power, ARM A64, etc.

Microarchitecture (Implementation Techniques)

- Basic Pipelining
- Deep Pipelining, Superscalar, VLIW, Multicore.
- Branch Prediction
- Dynamic Scheduling (Out-of-order execution.)
- Memory and Caches

References:

[1] Amdahl, G. M., Blaauw, G. A., and Brooks, F. P. Architecture of the IBM System/360. *IBM Journal of Research and Development 8*, 2 (Apr. 1964), 87101. `https://doi.org/10.1147/rd.82.0087`.

[2] Waterman, A., Lee, Y., Patterson, D. A., and Asanović, K. The RISC-V instruction set manual, Volume I: Base user-level ISA. Tech. Rep. UCB/EECS-2011-62, EECS Department, University of California, Berkeley, May 2011.