CS/ECE 5381/7381 Computer Architecture Spring 2023

Dr. Manikas
Computer Science

Lecture 1: Jan. 17, 2023

Instructor Information

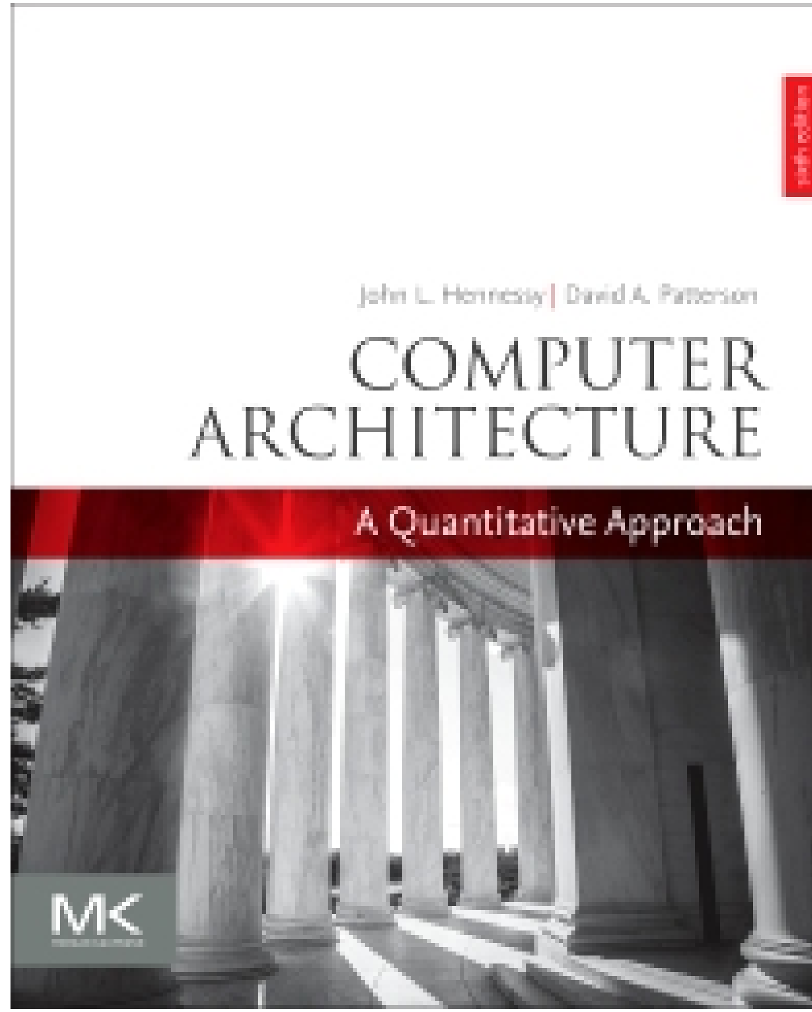
- Instructor: Dr. Theodore Manikas
- E-mail: manikas@smu.edu
- Office: Caruth Hall 477
- Office Hours: Tu, Th 9:30 – 10:30 am, or by appointment
  - The best way to contact me is by e-mail - please include “CS/ECE 5381” or “CS/ECE 7381” in the Subject Line of your e-mail for prompt response.

Prerequisites

- C- or better in CS 4381/ECE 3382 or equivalent: machine organization, instruction set architecture design, memory design, control design, algorithms for computer arithmetic, microprocessors and pipelining.

Textbook

- Text: J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, **6th Edition**, 2017. **ISBN13:** 978-0128119051



Material Covered

- Ch. 1 Fundamentals of Quantitative Design and Analysis
- App. A Instruction Set Principles
- App. C Pipelining
- Ch. 3 Instruction-Level Parallelism and Its Exploitation
- App. B Review of Memory Hierarchy
- Ch. 2 Memory Hierarchy Design
- Ch. 4 Data-Level Parallelism
- Ch. 5 Thread-Level Parallelism
- Ch. 6 Warehouse-Scale Computers
- Ch. 7 Domain-Specific Architectures

Grading

Quizzes	15%
Exams	75%
Projects	10%

Quizzes

- Will be assigned weekly
- Open book, open notes
- Lowest quiz score will be dropped

Exams

- 3 exams
- Open book, open notes
- Will require Lockdown Browser

Projects

- Programming projects
- Will be assigned as we cover key concepts in course
- There will be extra projects for CS/ECE 7381 students (graduate sections)

Course Syllabus

- The course syllabus is on Canvas
- Contains more details on course schedule, grading, policies, etc.
- Students should become familiar with all course and university policies as they will be followed during the semester
- Contact Dr. Manikas for questions

QUESTIONS?

- Any questions on general course schedule and assignments?

Fundamentals of Quantitative Design and Analysis

(Chapter 1, Hennessy and Patterson)

Note: some course slides adopted from publisher-provided material

Outline

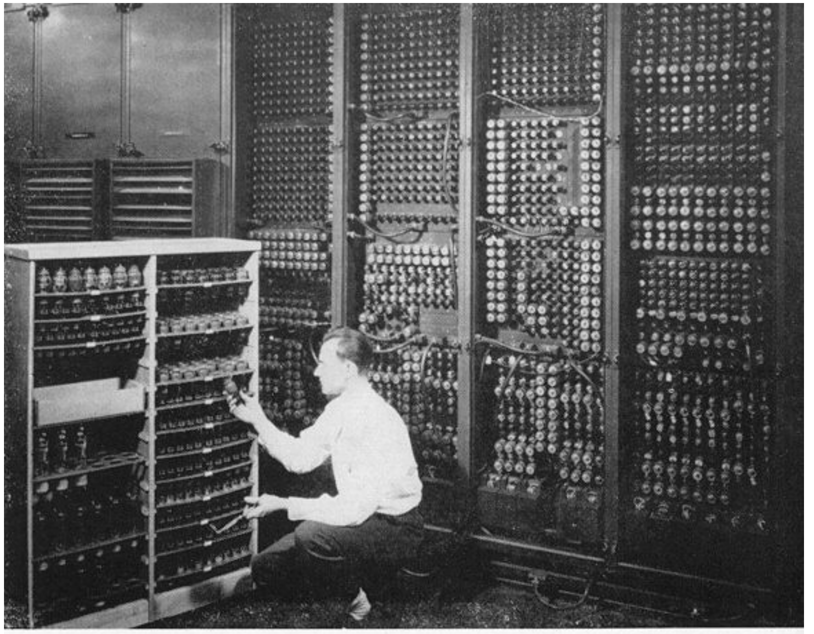
- 1.1 Introduction
- 1.2 Classes of Computers
- 1.3 Defining Computer Architecture
- 1.4 Trends in Technology
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

Introduction

- Original purpose – arithmetic operations
  - "Computer" – to compute numbers
- Digital computer – 1940's

ENIAC – 1940's

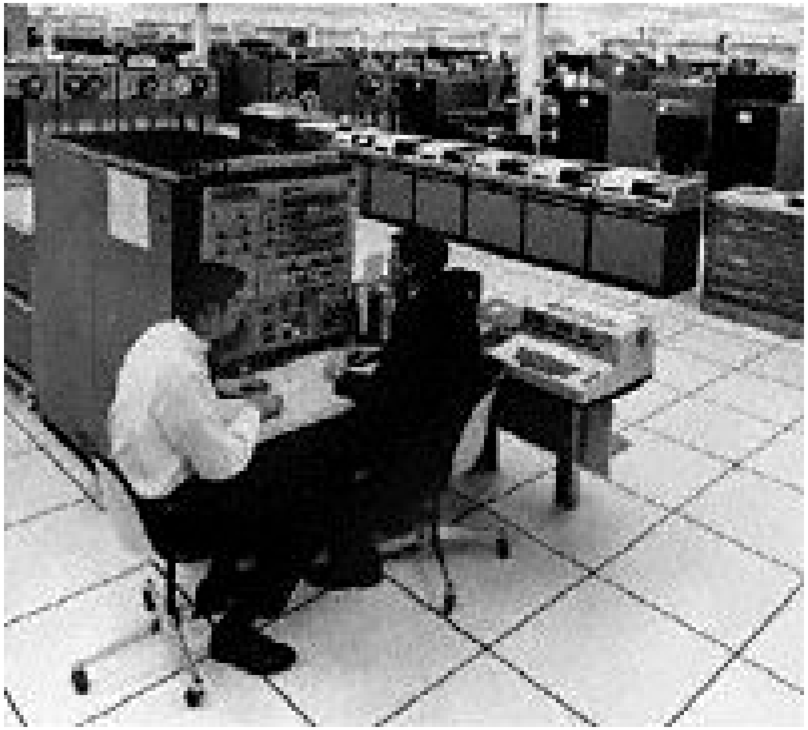
- University of Pennsylvania
- WWII – computing ballistic tables
- Hardware
  - 18,000 vacuum tubes
  - 1,500 electromagnetic relays
- Programming
  - hardwired – no software!



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

IBM System/360 - 1964

- Business use
- Hardware
  - transistors on PCB's
    晶体管
- Programming
  - FORTRAN



Intel 4004 – 1971

- First commercial microprocessor
  - “computer on a chip”
  - 2300 transistors
    晶体管
  - 4 bits

IBM PC – 1981

- First "commercial" personal computer
- Hardware
  - 4.77 MHz Intel 8088
  - 16 KB RAM
  - 160 KB floppy drive
- Operating system
  - MS-DOS

Apple Macintosh - 1984

- Jan. 1984 – Apple Macintosh released
  - List price: $2495

- Processor: Motorola 68000 (8 MHz)
- 3.5” floppy drive (400 KB)
- 128 KB RAM
- Black and white 9” CRT screen
- First PC with GUI and mouse (early “Windows”)

Dell Laptop - 2013

- Hardware
  - 2.7 GHz Intel Core i5
  - 6 GB RAM
  - 500 GB hard drive
- Operating system
  - Microsoft Windows 8

HP Pavilion Laptop 15 - 2022

- Hardware
  - 4.7 GHz Intel Core i7
  - 16 GB RAM
  - 512 GB flash storage
- Operating system
  - Windows 11

Computer Technology

- Performance improvements:
  - Improvements in semiconductor technology
    半导体
    - Feature size, clock speed
  - Improvements in computer architectures
    - Enabled by HLL compilers, UNIX
    - Lead to RISC architectures
  - Together have enabled:
    - Lightweight computers
    - Productivity-based managed/interpreted programming languages

Current Trends in Architecture

- Cannot continue to leverage Instruction-Level parallelism (ILP)
  - Single processor performance improvement ended in 2003

- New models for performance:
  - Data-level parallelism (DLP)
  - Thread-level parallelism (TLP)
  - Request-level parallelism (RLP)

- These require explicit restructuring of the application

Outline

- 1.1 Introduction
- <u>1.2 Classes of Computers</u>
- 1.3 Defining Computer Architecture
- 1.4 Trends in Technology
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

Classes of Computers

- Personal Mobile Device (PMD)
  - e.g. ~~start~~ phones, tablet computers
    smart
  - Emphasis on energy efficiency and real-time
- Desktop Computing
  - Emphasis on price-performance
- Servers
  - Emphasis on availability, scalability, throughput

n. 可扩展性；可伸缩性；可量测性

Classes of Computers

- Clusters / Warehouse Scale Computers
  - Used for "Software as a Service (SaaS)"
    - (cloud computing)
  - Emphasis on availability and price-performance
  - Sub-class: Supercomputers, emphasis: floating-point performance and fast internal networks
- Embedded Computers
  - Emphasis: price

Comparison of Classes

Feature	Desktop/Laptop	Server	Embedded
Price range	$300 - $3K	$5K - $10M	$10 - $100K
Price/MPU	$50 - $500	$200 - $2K	$0.01 - $100
Critical System Design Issues	Price/performance, power consumption (laptop)	Throughput, availability, scalability, energy	Price, power consumption, application-specific performance

Parallelism

- Classes of parallelism in applications:
  - Data-Level Parallelism (DLP)
  - Task-Level Parallelism (TLP)

- Classes of architectural parallelism:
  - Instruction-Level Parallelism (ILP)
  - Vector architectures/Graphic Processor Units (GPUs)
  - Thread-Level Parallelism
  - Request-Level Parallelism

Flynn’s Taxonomy

/tækˈsɑːnəmi/分类学

- Single instruction stream, single data stream (SISD)
  - Use ILP for parallel processing (Ch. 3)
- Single instruction stream, multiple data streams (SIMD) (Ch. 4)
  - Vector architectures
  - Multimedia extensions
  - Graphics processor units

Flynn's Taxonomy

- Multiple instruction streams, multiple data streams (MIMD)
  - Tightly-coupled MIMD
    - Uses Thread-Level Parallelism (Ch. 5)
  - Loosely-coupled MIMD
    - Uses Request-Level Parallelism (Ch. 6)

Outline

- 1.1 Introduction
- 1.2 Classes of Computers
- <u>1.3 Defining Computer Architecture</u>
- 1.4 Trends in Technology
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

WHAT IS COMPUTER ARCHITECTURE?

- OLD DEFINITION: INSTRUCTION SET ARCHITECTURE (ISA)
  - Instruction set - set of all possible operations in a machine's language
  - Machine's memory
  - Programmer accessible registers
  - Boundary between software and hardware
- TODAY'S DEFINITION IS MUCH BROADER: HARDWARE ORGANIZATION OF COMPUTERS (HOW TO BUILD COMPUTER)--INCLUDES ISA

Typical Modern ISA's

- *General-purpose register* architecture
  - Operands are either
    - Registers, or
    - Memory locations
- *Load-store* classification
  - Can only access memory with *load* or *store* instructions
- Byte addressing often used to access memory operands

Instruction Classes

- Data movement - move data between memory and/or registers, peripherals
- Arithmetic/logic
  - Arithmetic - add, multiply
  - Logic - and, or, not
- Control flow - branching

分支

```
for (i=0; i < N; i++){

}
```

Computer Architecture

- Components
  - ISA
  - Organization
  - Hardware
- Computer Architect Concerns
  - Design and performance of **entire** computer system
  - Optimize with respect to cost, size, time to market

Modern Computer Architecture

- Specific requirements of the target machine
- Design to maximize performance within constraints: cost, power, and availability
- Includes ISA, microarchitecture, hardware

Outline

- 1.1 Introduction
- 1.2 Classes of Computers
- 1.3 Defining Computer Architecture
- 1.4 Trends in Technology
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

Trends in Technology

- Integrated circuit technology (chip)
  - Transistor density: 35%/year
  - Die size: 10-20%/year
  - Integration overall: 40-55%/year
- DRAM capacity: 25-40%/year (slowing)
  - This is the RAM in your computer
  - 8 GB (2014), 16 GB (2019)

Trends in Technology

- Flash capacity: 50-60%/year
  - 8-10X cheaper/bit than DRAM

- Magnetic disk capacity: recently slowed to 5%/year
  - 8-10X cheaper/bit than Flash
    - Eventually to be replaced by Flash (SSD) as Flash costs decrease
  - 200-300X cheaper/bit than DRAM

Bandwidth and Latency

- Bandwidth or throughput
  - Total work done in a given time
  - 32,000-40,000X improvement for processors
  - 300-1200X improvement for memory and disks

- Latency or response time
  - Time between start and completion of an event
  - 50-90X improvement for processors
  - 6-8X improvement for memory and disks

Transistors and Wires

- Feature size
  - Minimum size of transistor or wire in x or y dimension
  - 10 microns in 1971 to .011 microns in 2017
    - 1 micron = 1 micrometer = 10^{-6} meters
  - Transistor performance scales linearly
    - Wire delay does not improve with feature size!
  - Integration density scales quadratically

Outline

- 1.1 Introduction
- 1.2 Classes of Computers
- 1.3 Defining Computer Architecture
- 1.4 Trends in Technology
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design