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

Depend on quizzes.

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

  It's a very important design constraint if you work in industry.
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

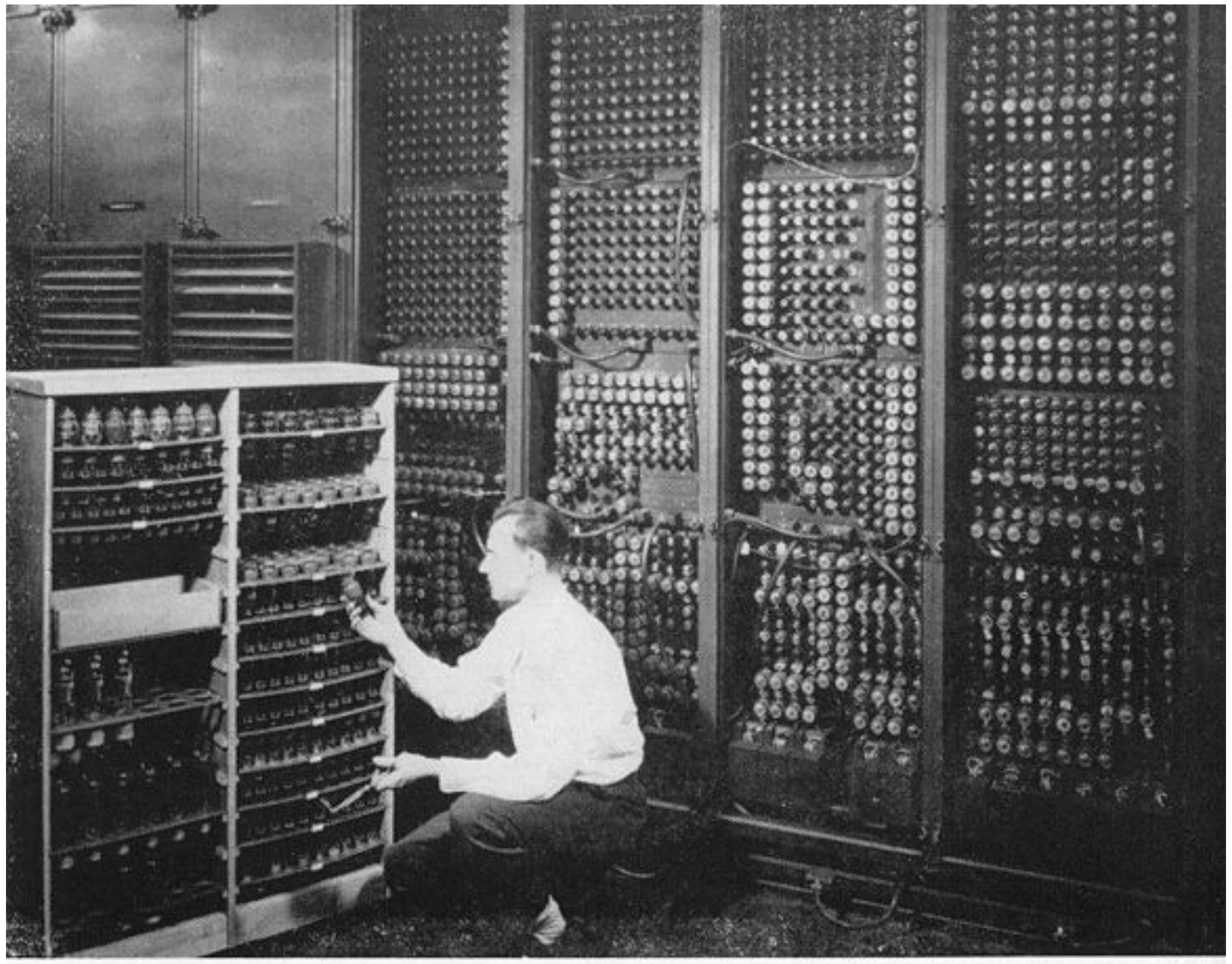
The problem is computer is such an obvious term
that's we sometimes forget why?
Why do computers exist in the first place? And why were they design?

Introduction

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

ENIAC – 1940's

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



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

It was doing is computing ballistics tables

It's like an electric abacus.

IBM System/360 - 1964

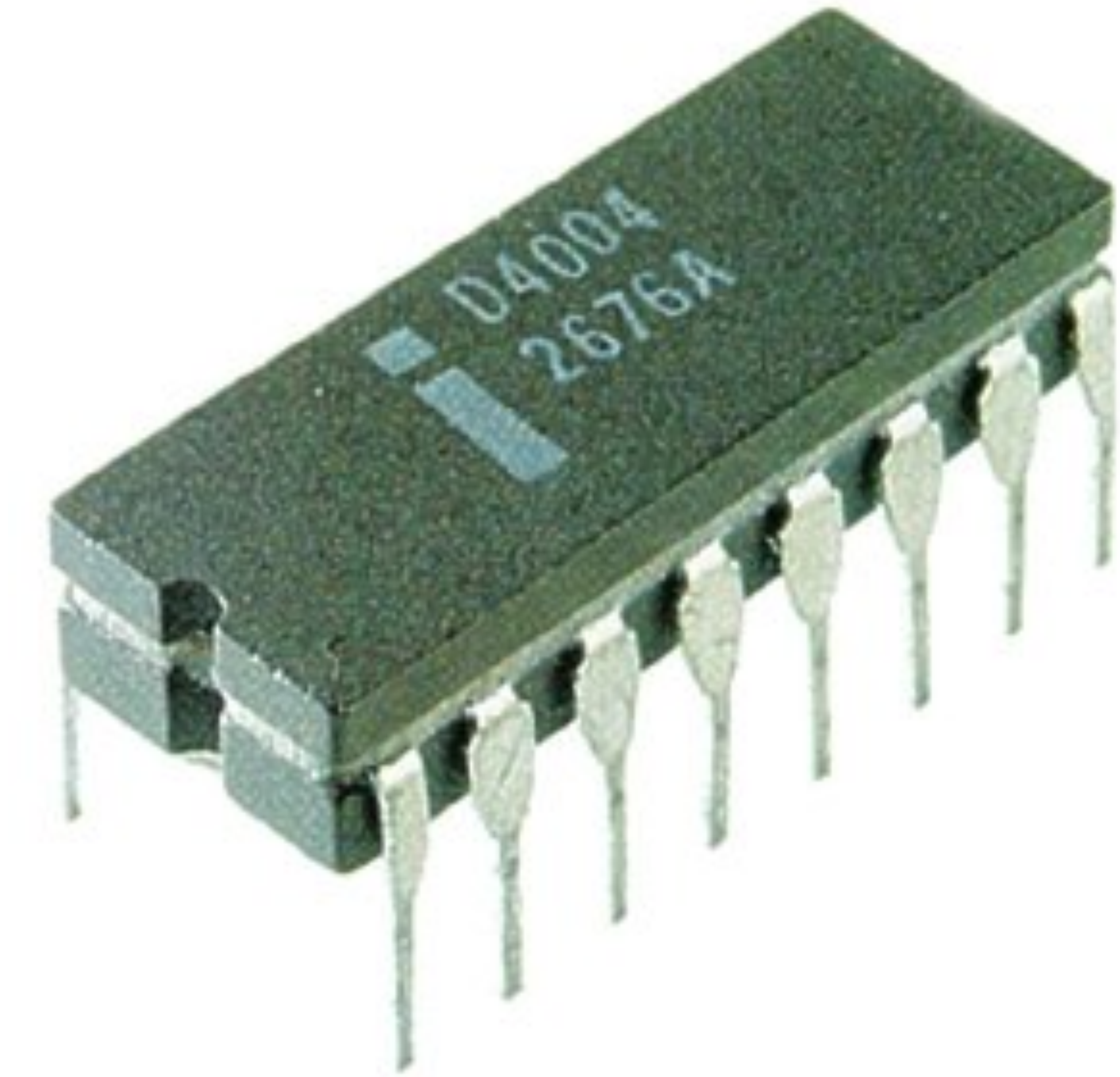
- Business use
- Hardware
  - transistors on PCB's
    晶体管
- Programming
  - FORTRAN

Intel 4004 – 1971

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

it's a 4 bit machine.

So this was allowed us to lead to the lap-top designs.



IBM PC – 1981

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

It's a command line similar to Linux.

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 much more portable, tactile.
    - 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)

The reason we've going to talk about these is because this is what modern computers do. How do we make things faster and faster? How do we make program run more efficiently?

- These require explicit restructuring of the application

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

Classes of Computers

- Personal Mobile Device (PMD)
  - e.g. ~~start~~ smart phones, tablet computers
  - 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

Go with Amazon, would like to get as many orders as they can throughput once again.

How many orders can I get in a given time window?

Classes of Computers

sometimes called

- 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

good for if you're doing machine learning.

- Embedded Computers
  - Emphasis: price

嵌入式计算机

Where do we see these smart appliances?

Smart thermostat.

Modern cars in the past ten years any look there brake system that's a computer. Very dedicated computer with limited. You can't surf the web on that thing, but it has a very dedicated job.

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) Will talk about how to do that.
- 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
- 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

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

So you have these little magic boxes that do things for us. If we just understand what goes on inside the box, that's enough.

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

外部设备

- Arithmetic/logic
  - Arithmetic - add, multiply
  - Logic - and, or, not
- Control flow - branching

分支

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

}
```

BRANCHING HIGH-LEVEL LANGUAGE

```
for (i=0; i<N; i++)
    A[i] = 0;
```

BRANCH.

We've got a branch back there.

```
if (x != y) {

} else {

}
```

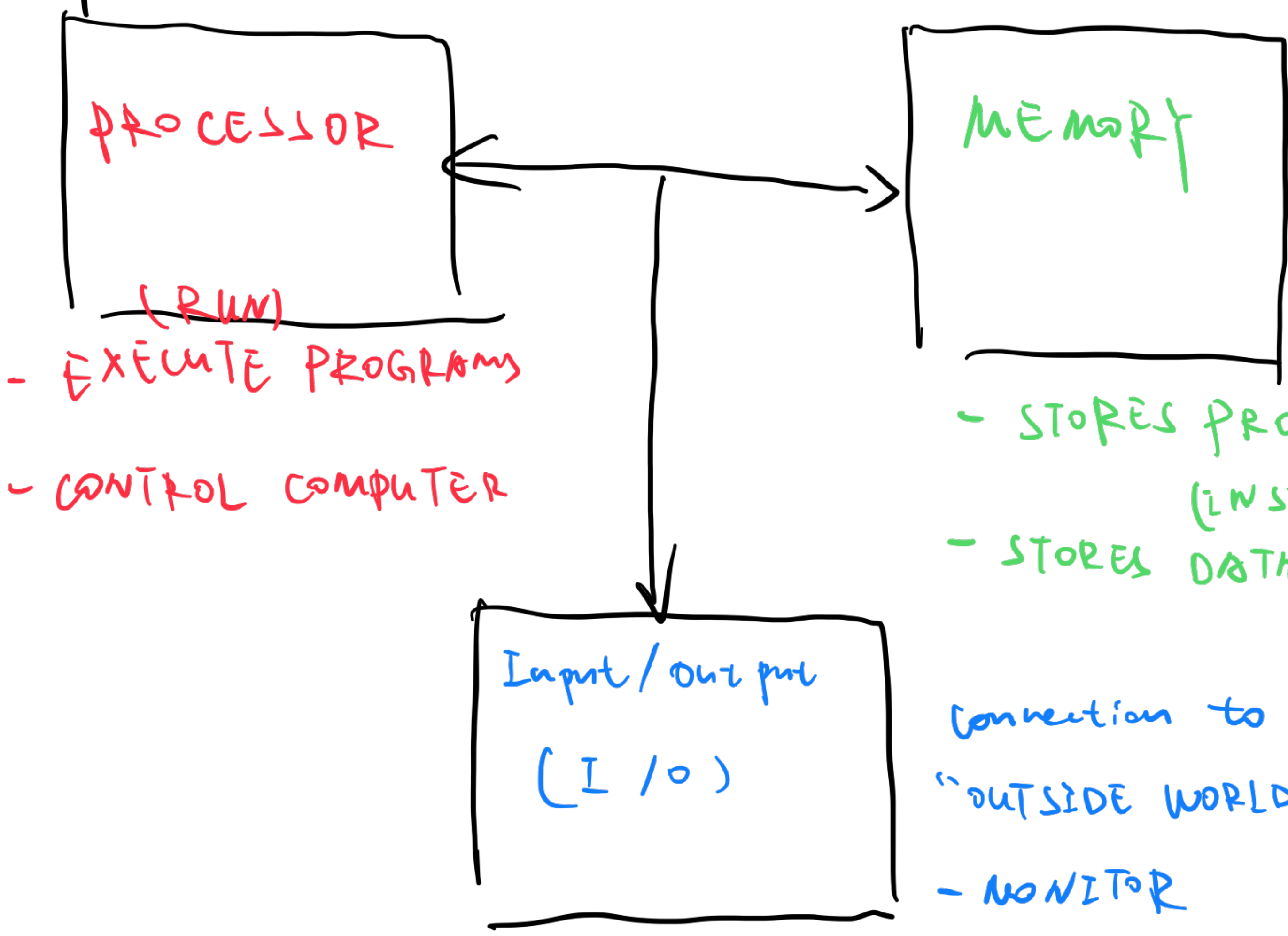
Branching

Your instruction set architecture to do that means your computer has to be able to do that.

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

GENERAL COMPUTER SYSTEM



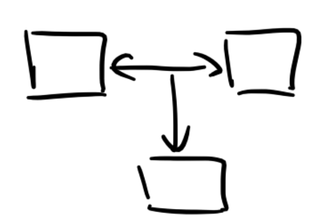
(RUN)
- EXECUTE PROGRAMS
- CONTROL COMPUTER

- STORES PROGRAM (instructions)
- STORES DATA

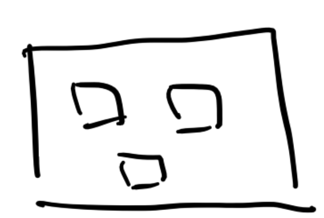
connection to "OUTSIDE WORLD"
- MONITOR
- KEYBOARD
- NETWORK
⋮ etc.

LEVELS of ABSTRACTION

1. BLOCK DIAGRAM

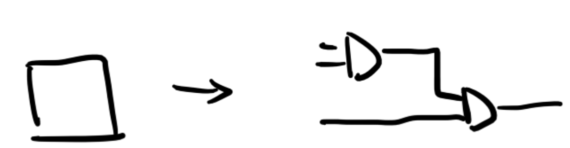


2. INSIDE THE BLOCKS: "SUB-BLOCKS"



more detailed.

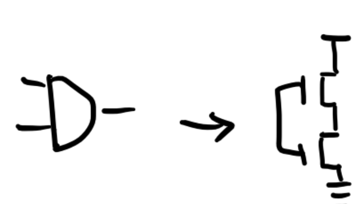
3. INSIDE SUB-BLOCK
   - DIGITAL LOGIC



This course

This is as far as we're going to go.

4. INSIDE LOGIC GATES
   - TRANSISTORS



5. INTEGRATED CIRCUIT

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
- <u>1.4 Trends in Technology</u>
- 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

/ˌɪntɪɡreɪtɪd/ /ˈsɜːkɪt/ 集成电路

- Integrated circuit technology (chip)
  - Transistor density: 35%/year

    晶体管
  - 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)

Year	Processor	# TRANSISTIONS
1971	INTEL 4004	2,300
1980	MOTOROLA 68000	68,000
1989	INTEL 80486	~ 1 MILLION
RECENT	INTEL CORE SERIES	~ 1 Billion.

MORE TRANSISTORS $\Rightarrow$ MORE DIGITAL LOGIC

$\Rightarrow$ MORE FUNCTIONS ON PROCESSORS.

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