university of groningen

Computer Architecture 2023-24 (WBCS010-05)

Lecture 1: Introduction

Reza Hassanpour
r.zare.hassanpour@rug.nl

university of groningen

Topics

- What is the Computer Architecture course about?
  - What does a computer do?
  - How does a computer work?
  - What is a computer system?
  - What are the main components of a computer system?
- What will you learn?
  - The hardware of a computer
  - The relationships with other courses

university of groningen

Topics

- Course details
  - Lectures
  - Tutor sessions
  - Lab sessions
  - Assignments
  - Final exam
  - Evaluation

What does a Computer do?

- A computer is a complex electronic device that processes data according to a set of instructions called a program.

- Computers are versatile devices capable of
  - processing,
  - storing,
  - and communicating
- information.

university of groningen

How does a Computer Work?

- There is **no magic** to computing!
  - Deterministic system – behaves the same way every time
  - Does exactly what we **tell** it to do: no more, no less
  - Complex system made of very simple parts
  - Even recent advances in AI come from our ability to do many (billions!) simple computations very fast!

university of groningen

A Computer Program

- A computer program is a set of instructions that tells a computer how to perform a specific task or series of tasks.

- These instructions are written in a programming language that a computer can understand and execute.

university of groningen

What do Computers Understand?

- Computers "understand" information in a highly structured and binary way.
- At their most fundamental level, computers work with binary code, which consists of sequences of **0**s and **1**s.
- Computers have their own language. The programs should be translated into their language.

How to Execute a Program?

- A program is a sequence of instructions.
- The questions arise are:
  - Where to put the program to execute?
  - Where to execute each instruction?
  - In what order are the instruction executed?
  - How to receive user input?
  - How to communicate the result of execution?

- We need a few components that can interact with each other.

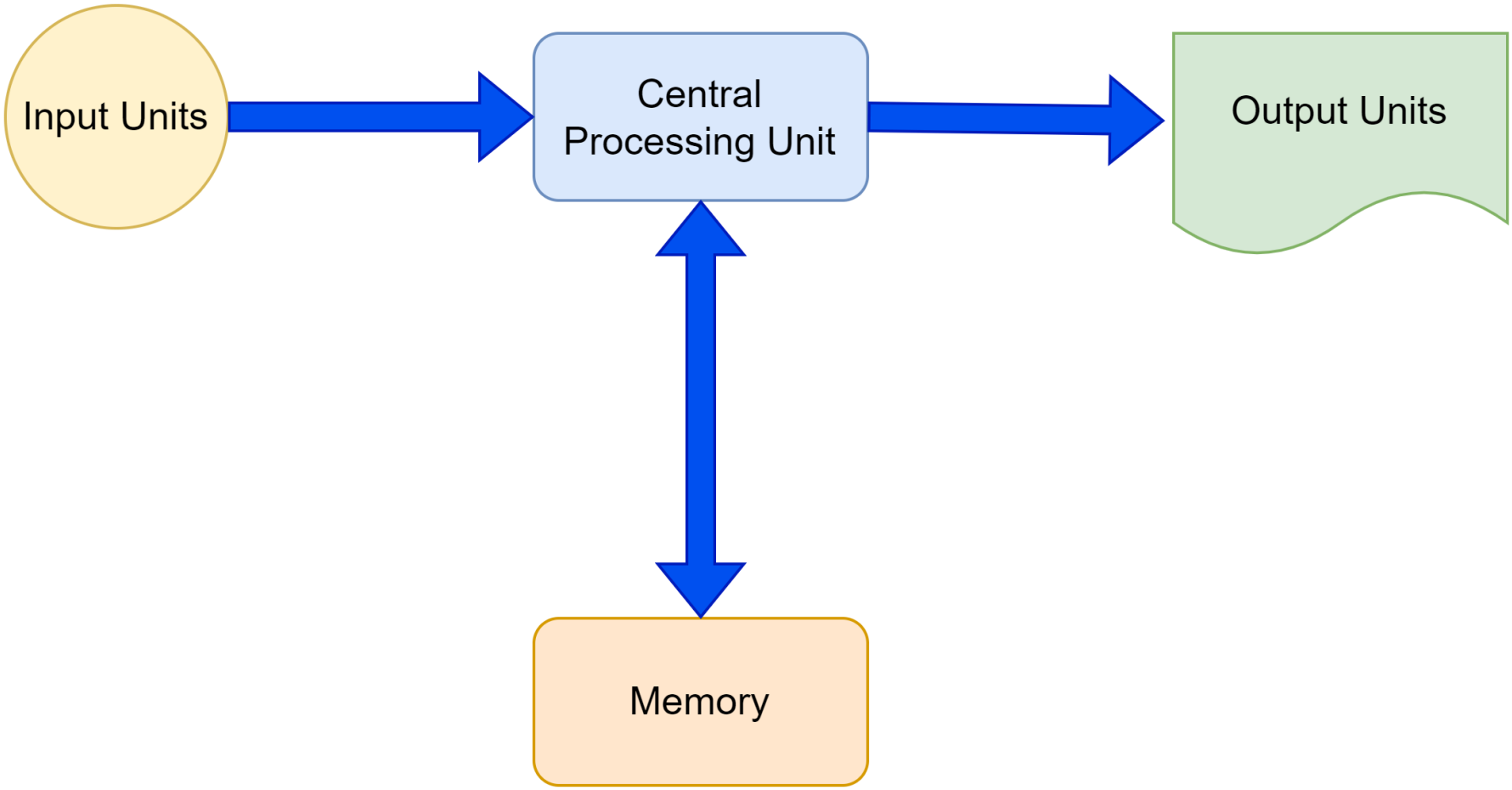
university of groningen

What is a Computer System?

- A computer system is a nominally complete computer that includes
  - the hardware,
  - operating system (main software),
  - and peripheral equipment needed and used for full operation.

university of groningen

Block Diagram of a Computer



Central Processing Unit (CPU)

- In responsible for executing instructions.

- Controls and oversees the execution of instructions, manages data flow, controls input/output operations, and ensuring proper synchronization of the various components.

Memory

- Is used to store programs and data.
- Enables the computer to store and retrieve information efficiently.

Input/Output Units

- Input units allow users to input data and commands
- They facilitate the interaction between users and the computer by converting various forms of input into a format that the computer can understand and process.
- Output units are devices that present information processed by the computer to the user in a human-readable format.

university of groningen

What about Operating Systems?

- The operating system (OS) is a crucial software component that serves as an intermediary between the computer hardware and the user or applications.

- It plays a fundamental role in managing and controlling the computer system.

Brief History of Computers (I)

- 1st Generation:
  - from 1940 to 1955.
  - machine language was developed.
  - vacuum tubes were used for the circuitry.
  - magnetic drums were used as memory.
  - these computers were large, and expensive, using batch operating systems and punch cards.
  - magnetic tape and paper tape were implemented.
  - examples, ENIAC, UNIVAC-1, EDVAC.

university of groningen

Brief History of Computers (II)

- 2nd Generation:
  - from 1957-1963
  - COBOL and FORTRAN were employed as programming languages.
  - hardware advanced from vacuum tubes to transistors.
  - computers became smaller, faster and more energy-efficient.
  - examples, IBM 1620, IBM 7094, CDC 1604, CDC 3600.

university of groningen

Brief History of Computers (III)

- 3rd Generation:
  - from 1964-1971.
  - integrated circuits(IC) were developed.
  - more computing power while lower cost.
  - computers were faster, smaller, and less expensive.
  - high-level programming languages such as PL/1, C, and PASCAL were used.
  - examples, the IBM-360 series, the Honeywell-6000 series, and the IBM-370/168.

Brief History of Computers (IV)

- 4th Generation:
  - 1971-1980
  - invention of the microprocessors.
  - C++ and Java were the programming languages used.
  - examples STAR 1000, PDP 11, CRAY-1, IBM PC, Apple Macintosh.

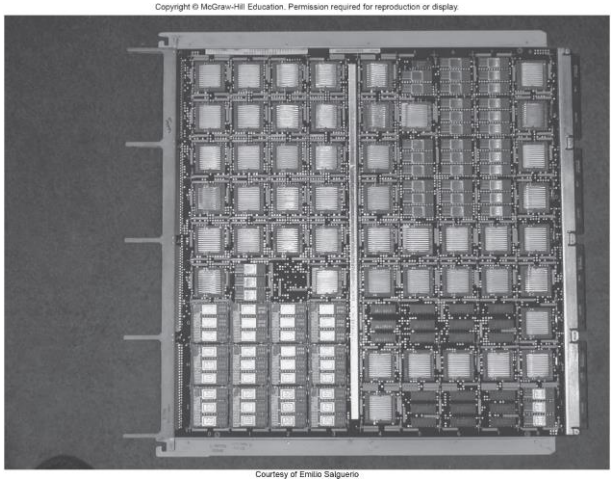
Brief History of Computers (V)

- 5th Generation:
  - since 1980.
  - advances in parallel processing, artificial intelligence, and supercomputing.
  - miniaturization with developments like laptops, tablets, and smartphones.

university of groningen

Computers from 1940s to 2010s

1940s
ENIAC
17,000 vacuum tubes 300 sq feet (8 ft tall) 30 tons 140 Kilowatts
1980s
Burroughs A Series
Dozens of IC boards 60 sq feet 1 ton 25 Kilowatts
2020s
Smartphone
1 board – system-on-chip 20 sq inches 6-8 ounces Battery-powered



› Smartphone has 4,000,000x more computing power than ENIAC

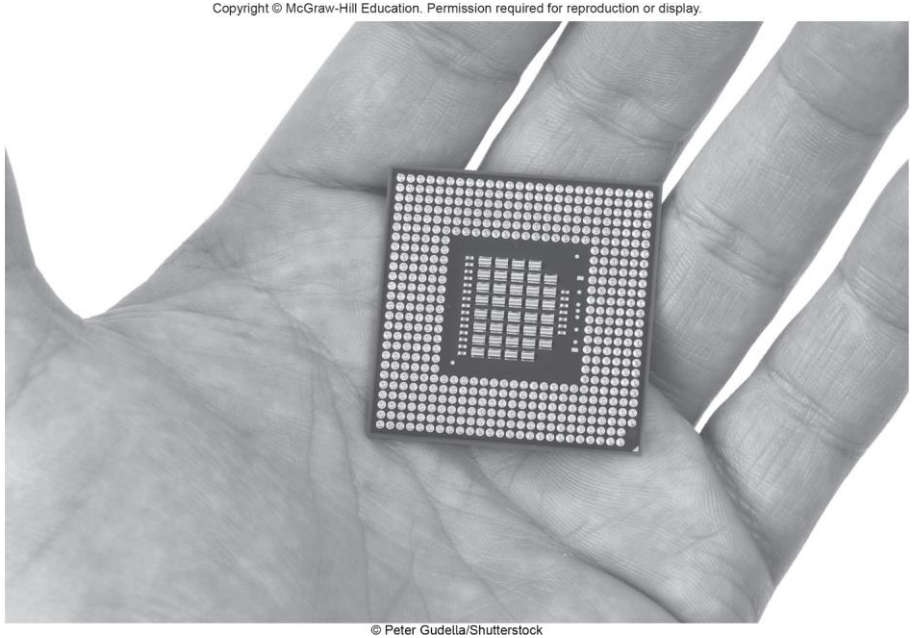
university of groningen

Microprocessors

1971
Intel 4004
2,300 transistors 106 kHz

1992
Intel Pentium
3.1 million transistors 66 MHz

2018
Intel Core i9
7 billion transistors 5 GHz



© Peter Gudella/Shutterstock

- This is the processor or
- CPU (central processing unit)
- of the computer system –
  where most computation takes place

university of groningen

Using Computer Systems

university of groningen

What Can Computers Do?

- **All computers**, given enough time and memory, are capable of computing exactly the same things
  - Smartphone, laptop, supercomputer… limited only by time and memory
- If you can describe something in terms of computation, it can be done by a computer… again, given enough time and memory
- Important to understand how each component works, but thinking at higher levels is more efficient.

Problem Solving using Computers

- Describe the requirements of the problem
  - We want to use sensors to detect objects and guide a robot towards them ..
- Design a solution
  - Choose the algorithms, data structures, etc.
- Write the program in high level language
- Translate the program into machine language
- Load the instructions into the memory
- Execute each instruction by bringing them into the central processing unit (CPU) one-by-one
- Use input/output devices to communicate information

Abstraction

- Abstraction: Productivity Enhancer
- You don't need to worry about the details…
  - You can drive a car without knowing how an internal combustion engine works.
- Important to understand the components and how they work together.
  But thinking at higher levels of abstraction is more efficient.
- Hardware **and** Software
  - It's not either/or – both are essential components of a computer system.
  - Even if you specialize in one, you must understand the capabilities and limitations of the other.

How do We Represent Data in a Computer?

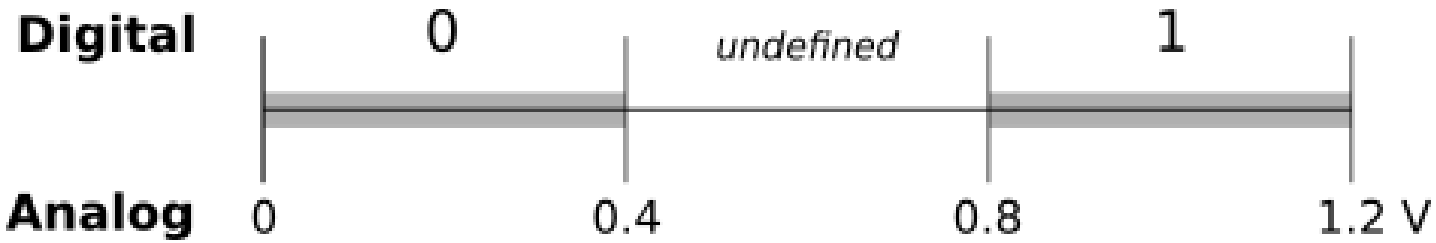
- At the lowest level, a computer is an electronic machine
  - Works by controlling the flow of electrons
- Easy to recognize two conditions:

1. Presence of a voltage – we'll call this state "1"
2. Absence of a voltage – we'll call this state "0"

- We could base the state on the value of the voltage, but control and detection circuits are more complex

Computer is a Binary Digital system

- Digital = finite number of values (compared to "analog" = infinite values)
  Binary = only two values: 0 and 1
- Unit of information = binary digit, or "bit"



- Circuits (see Chapter 3) will pull voltage down towards zero, or will pull up towards the highest voltage
- Grey areas represent noise margin -- allowable deviation due to resistance, capacitance, interference from other circuits, temperature, etc. More reliable than analog.

More Than Two Values…

- Each "wire" in a logic circuit represents **one bit** = 0 or 1
- Values with more than two states require multiple wires (bits)

- With two bits, can represent four different values:
- 00, 01, 10, 11
- With three bits, can represent eight different values:
- 000, 001, 010, 011, 100, 101, 110, 111

- With **n** bits, can represent $\mathbf{2^n}$ different values

university of groningen

What Kinds of Values do We Want to Represent?

- **Numbers** - integer, fraction, real, complex, …
- **Text** - characters, strings…
- **Images** - pixels, colors, shapes…
- **Sound**
- **Video**
- **Logical** - true, false
- **Instructions**

- All data is represented as a collection of bits
- We encode a value by assigning a bit pattern to represent that value
- We perform operations (transformations) on bits, and we interpret the results according to how the data is encoded

Questions

- How are programs and data stored in memory?
- How are the instructions defined?
- How does the computer interpret instruction?
- How are data and programs displaced in a computer system?
- How is the user input received, and how are the outputs delivered?
- How do different components of a computer system work in harmony?

university of groningen

Learning Outcomes

- 1) Describe the main concepts, design techniques, issues, and solutions across the abstraction layers of modern computer architectures;
- 2) Apply the acquired knowledge to programming with assembly language, including register operations, control structures, bitwise operations, and subprograms;
- 3) Describe the basic concepts of hierarchical data storage and the techniques to store and retrieve data.

Relationships with Other Courses

- As a practical matter: knowledge of computer architecture is needed for later courses, such as systems programming, compilers, operating systems, and embedded systems

- The most successful software engineers understand the underlying hardware

university of groningen

Questions?

Organizational Issues

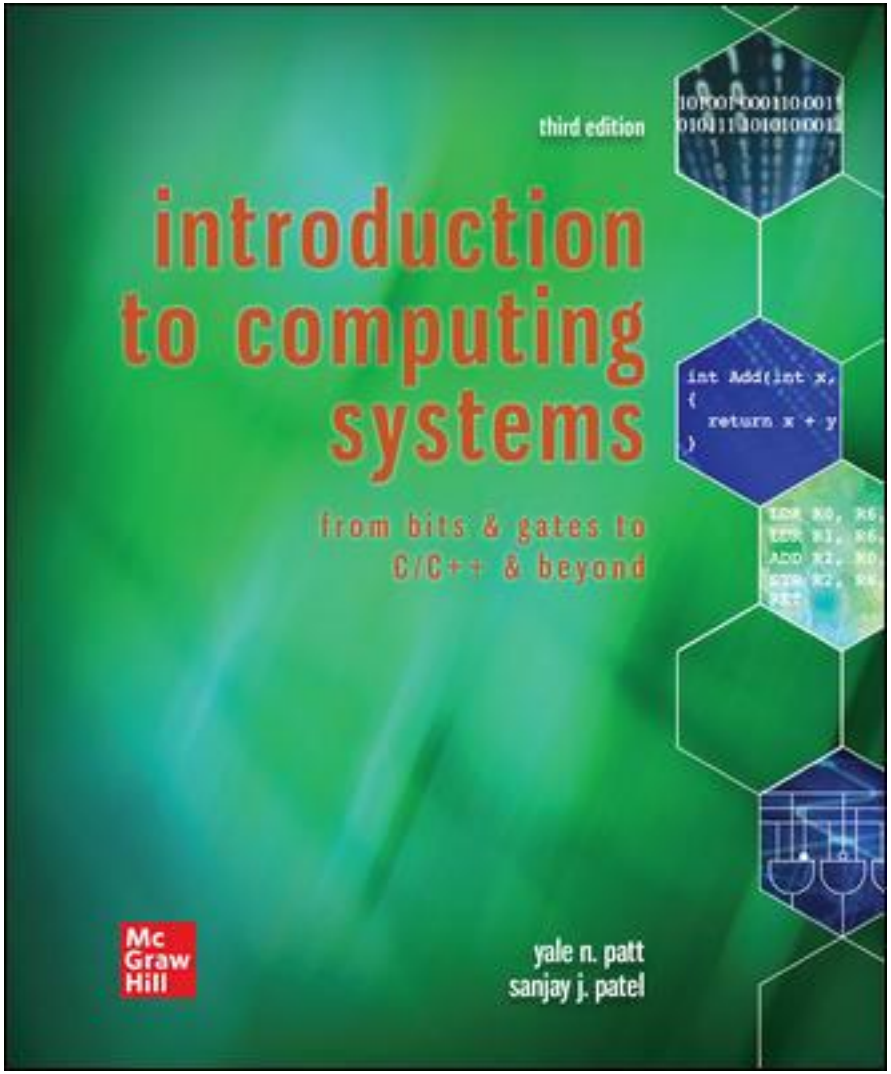
university of groningen

Course structure (2023-2024)

- **Sixteen** (16) **lectures**
  - Roughly following Chapters 1 to 10 in the book,
  - Some additional information not covered in the book
  - Final review and Q&A lecture on Wednesday, January 17, 2024
- **Seven** (7) **tutorials**
  - Every Wednesday 15:00-17:00
  - For receiving guidance by the TAs
- **Seven** (7) **lab sessions** (Thursday)
  - For receiving feedback from the TAs
- **Seven** (7) **graded homework assignments**
- Slides on Brightspace
  - New relevant material will be added when necessary

university of groningen

The Book (the 3rd edition)



https://www.mheducation.com/highered/product/introduction-computing-systems-bits-gates-c-c-beyond-patt-patel/M9781260150537.html

university of groningen

Grading

- Final exam and/or resit, and graded practical solutions for seven (7) homework assignments

- Final grade **F=(0.4 H+0.6 E)** with
  - **H** = grade Homework **(0.125(H1..H6)+0.25H7)**
  - **E** = grade Exam/resit
  - **iff H and E >= 5.0**
- Otherwise if (**H<5 or E<5**) **F=min(H, E)**
  - 6.0 still the minimum for passing grade
  - Rounding **F** to the .5 up except for **[5.5, 6) ➔ 6**

Assignments

- Goal: to gain understanding of the computing technology internal organization, operation and the main mechanisms
- Seven (7) practical assignments
  - Content, information and deadlines **on Brightspace**
- You will work **individually**
- **Support:**
  - join the lab sessions and ask questions; or
  - send mail to **your TA**

Assignments schedule

- Seven (7) practical assignments
  - Content, information and deadlines **on Brightspace**
- Made available on
  - **Monday**, after the lecture
  - starting in the second week on Mon, Nov 20
- Delivery deadline:
  - **Monday**, one week later at 23:59 CET

university of groningen

Assignment delivery

- Submit your non-programming assignments in PDF format
- Solutions are to be uploaded in time
  - before 23:59CET on Tuesday one week after announcement
- Grading results will be on Brightspace
  - feedback is also provided on **Brightspace**
- Grading penalties apply
  - No delivery: **100%**
  - Unacceptable delivery: **100%**
  - Lecturers and TAs decide on irregular cases

university of groningen

Assignment delivery (continued)

- What constitutes *unacceptable* delivery?
  - Delivery **after** the deadline
  - **Plagiarism** of another student answers (penalty applies to **all students involved**, disputes to be resolved by BoE)
  - The submission is **unreadable** (e.g., as a failing PDF)
  - The submission documents contain any items that are **scanned from notes or photographed**, e.g., no handwritten solutions will be accepted (use Word/LaTeX!)

university of groningen

Corner cases

- Exam/lab grades from the last year **can't be** used
  - **Repeating students** should perform all homework as everybody else
  - Please contact me if you have questions about this
- Assignment grades are valid for the same year's resit

Questions about the organization?