Lancaster University

SCC131: Digital Systems

Topic 1 - Computer Architecture Introduction

Lecture goals

In this topic, we are going to study:

- What is a Computer?
- A brief Computer history.
- What is Computer architecture?
- Von Neumman architecture vs. Harvard architecture vs. Modified Harvard Architecture.
- Limiting factors on computer speed

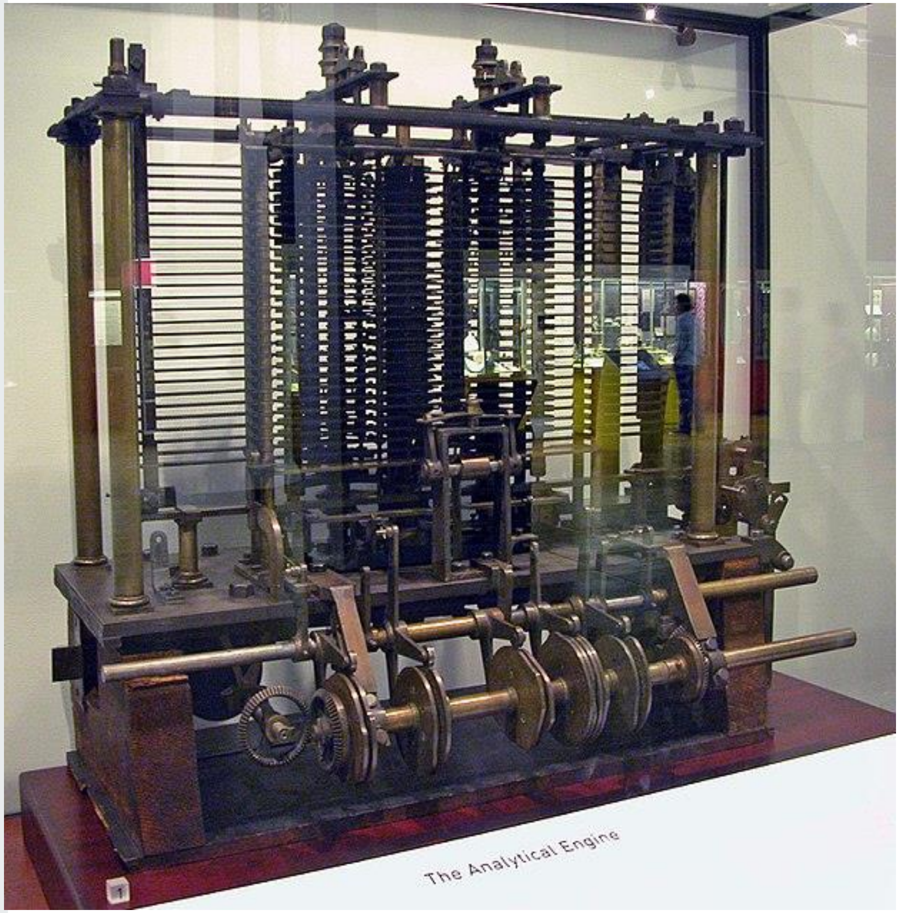
What is a computer?

- Computer is an electronic device for storing and processing data, typically in binary form, according to instructions given to it in a variable program.
  (Source: Lexico.com, Oxford University Press)

Computer history

- Analytical Engine
- ENIAC
- Manchester SSEM
- Cambridge EDSAC
- IBM Personal Computer

The First Computer-Analytical Engine (1)



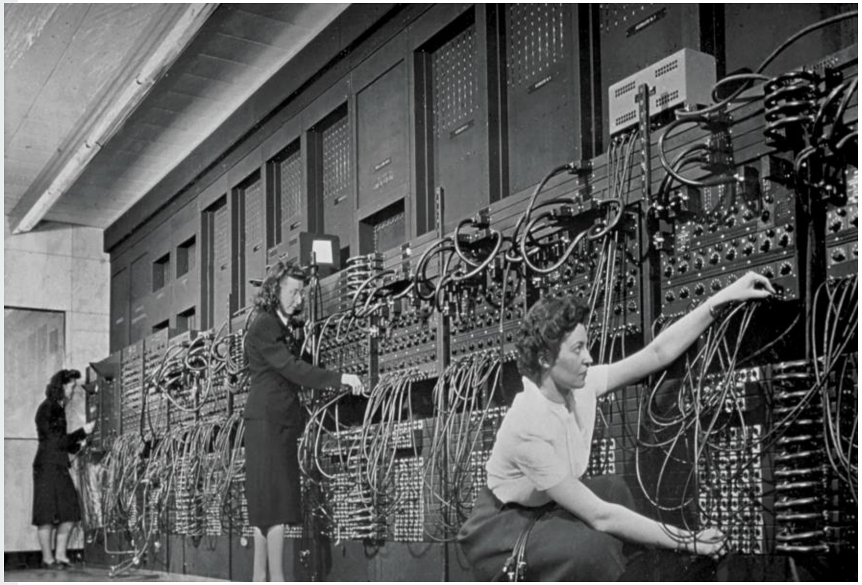
- Analytical Engine, generally considered as the first computer, proposed and partly built by the English inventor Charles Babbage in the 19th century.
- The Analytical Engine was a fully program-controlled mechanical computer.

Lancaster University

Analytical Engine (2)

- The machine was designed to consist of four components: the mill (calculating unit), the store, the reader, and the printer. These components are the essential components of every computer today.
- Data (numbers) were to be entered on punched cards.
- In 1843, a mathematician Ada Lovelace wrote the first algorithm for a computer, based on the Analytical Engine. For this, she has been called the first computer programmer.

ENIAC

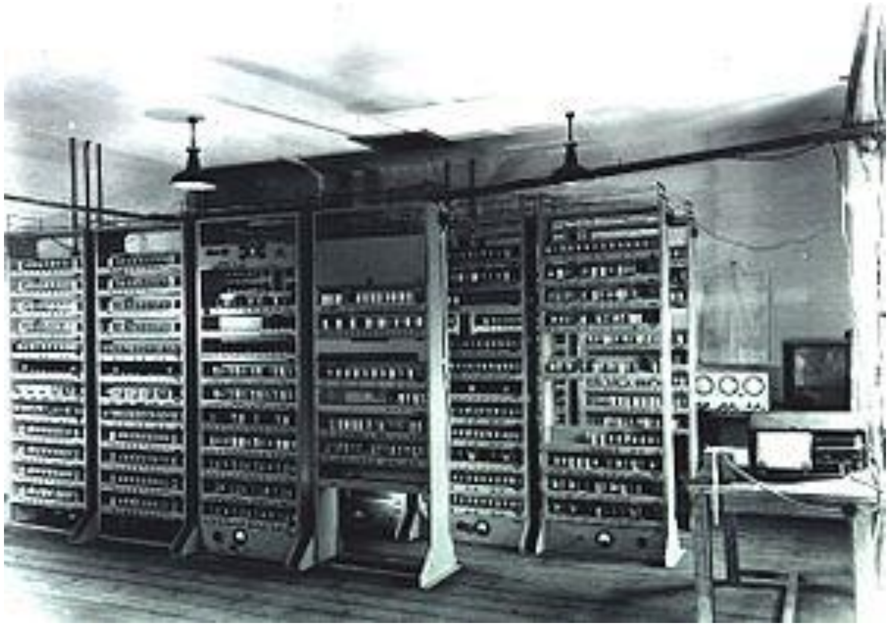


- ENIAC, in full **E**lectronic **N**umerical **I**ntegrator **A**nd **C**omputer, was the first programmable, electronic, general-purpose digital computer. It was built during World War II in the United States.
- It was programmed via switches and cables.
- It was able to solve a large class of numerical problems through reprogramming.

Manchester SSEM – Called “The Manchester Baby"

- Manchester SSEM (Small-Scale Experimental Machine), the first **electronic stored program** computer, built at the U of Manchester.
- It successfully ran its first program on the 21st June 1948.
- Replica available at Manchester Museum

Cambridge EDSAC



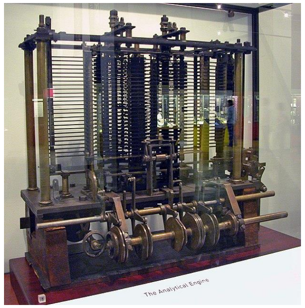
- The **E**lectronic **D**elay **S**torage **A**utomatic **C**alculator (EDSAC), developed at Cambridge University, ran its first programs in 1949.
- It became the first practical stored-program computer in regular use (**from test to tool**).
- Used punched cards to load programs

The IBM Personal Computer

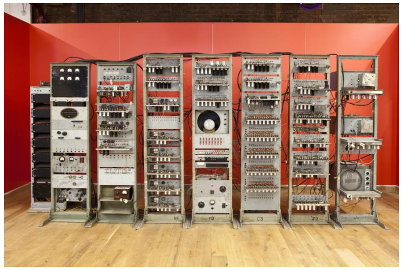
- The IBM Personal Computer (PC) was released in August 12, 1981.
- Led to an era of cheap commodity-based computation
  - ➢Growing adoption of the PC as a basis for Home Computers, servers and super-computers

Evolution of computer systems

Analytical Engine



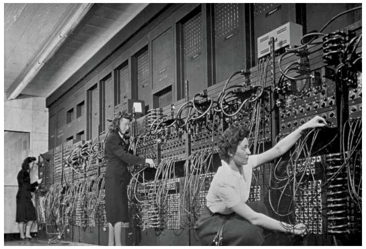
Manchester SSEM



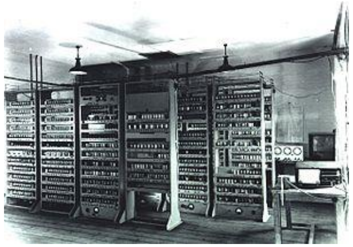
PC

Laptop

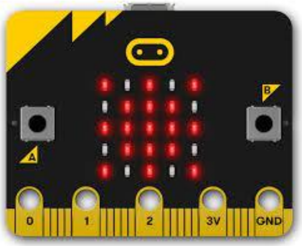
IPhone



ENIAC



Cambridge EDSAC



Microbit

Smart Watch

What is “Computer architecture”?

- In computing, **Computer architecture** is defined as a set of rules and methods that describe the functionality, organization, and implementation of computer systems, which can be applied at many levels/or layers such as:

➢ Processor architecture, memory architecture, instruction set architecture (ISA), etc.

➢ At the System level: how we link processors to devices for input and output, computer networks and other systems (e.g. architecture of the Internet), etc.

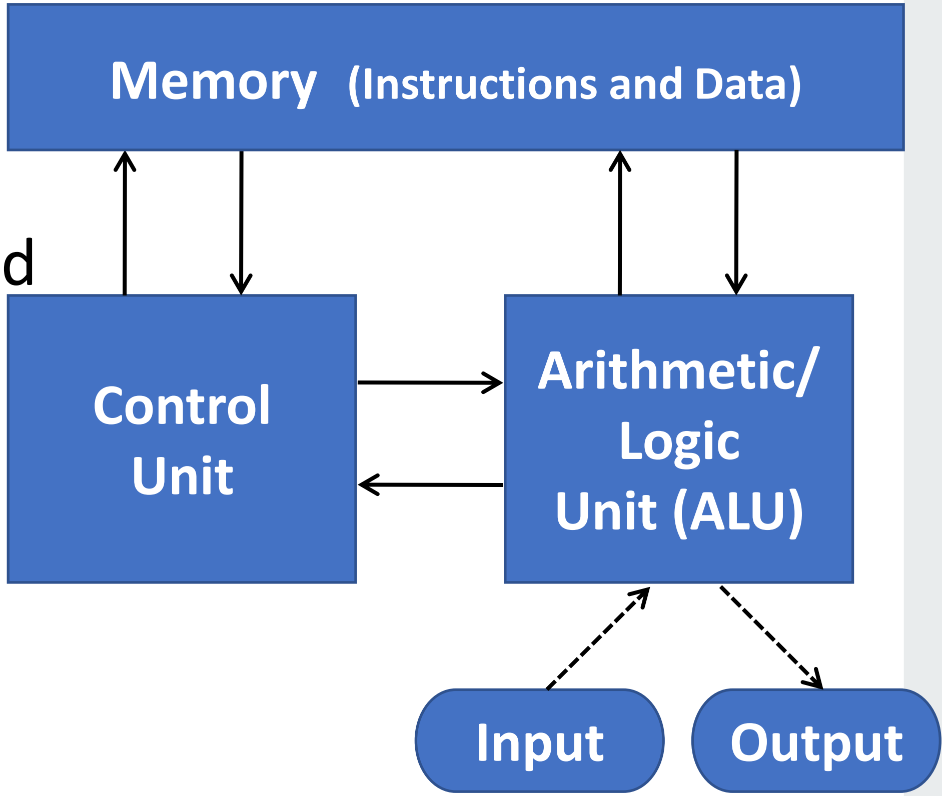
Von Neumman architecture (1)

- The Von Neumann architecture, is a computer architecture proposed by John Von Neumann, a mathematician and computer scientist, in 1945.

- Today, computer architecture is largely standardized, at a high-level of abstraction, on the Von Neumann architecture.

Von Neumman architecture (2)

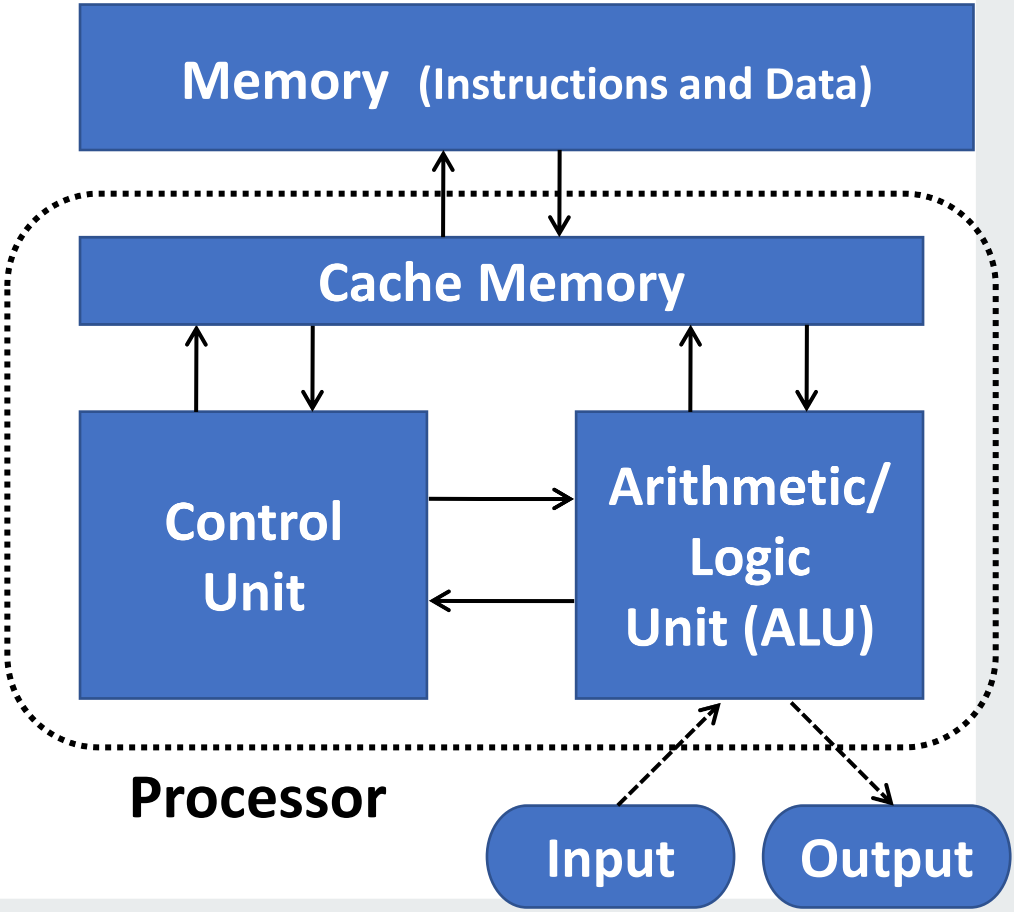
- Von Neumman architecture includes:
  - Memory that stores data and instructions
  - A control unit that contains an instruction register and program counter
  - A processing unit that contains an arithmetic logic unit (ALU) and processor registers
  - Input and output mechanisms



A Von Neumann architecture scheme

Von Neumman architecture – **the processor**

- Often also referred to as the Central Processing Unit (**CPU**), consisting of:
  - ALU + Control Unit; often also contains some internal high-speed *cache memory.*
- Note, this is still logically the same picture as on the previous slide - still a von Neumann architecture.

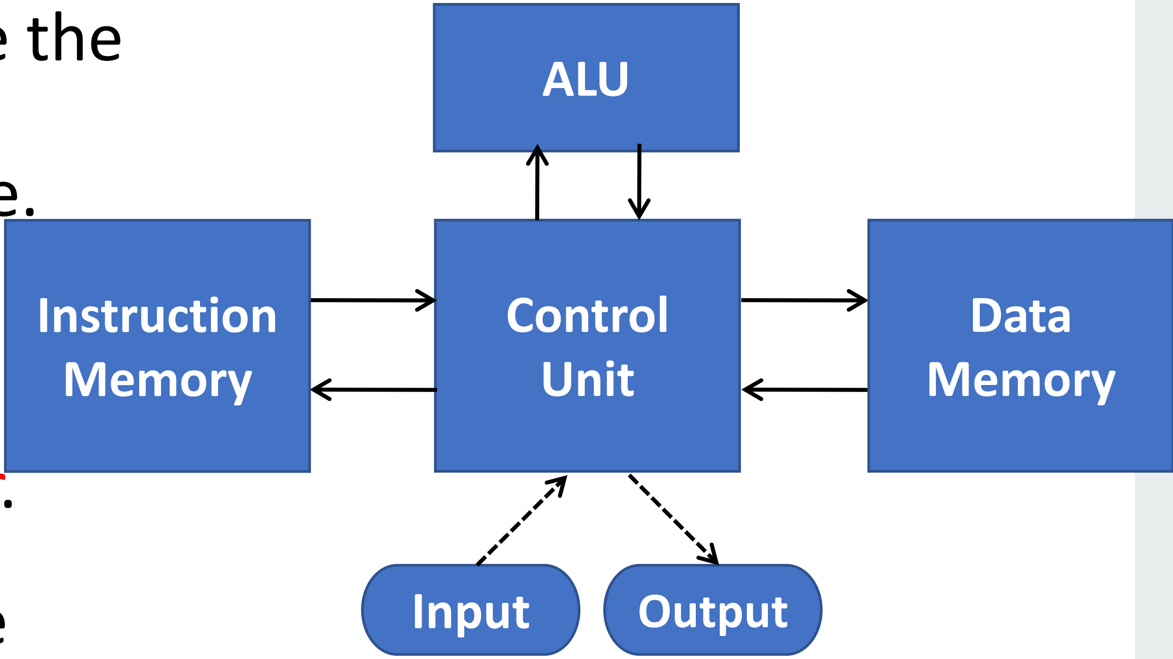


Harvard Architecture

- Instructions memories and data memories are separate, to overcome the bottleneck of Von Neumann Architecture.

- Parallel access to instruction and data memory, can be faster.

- Better cyber resilience against potential cyber attacks.

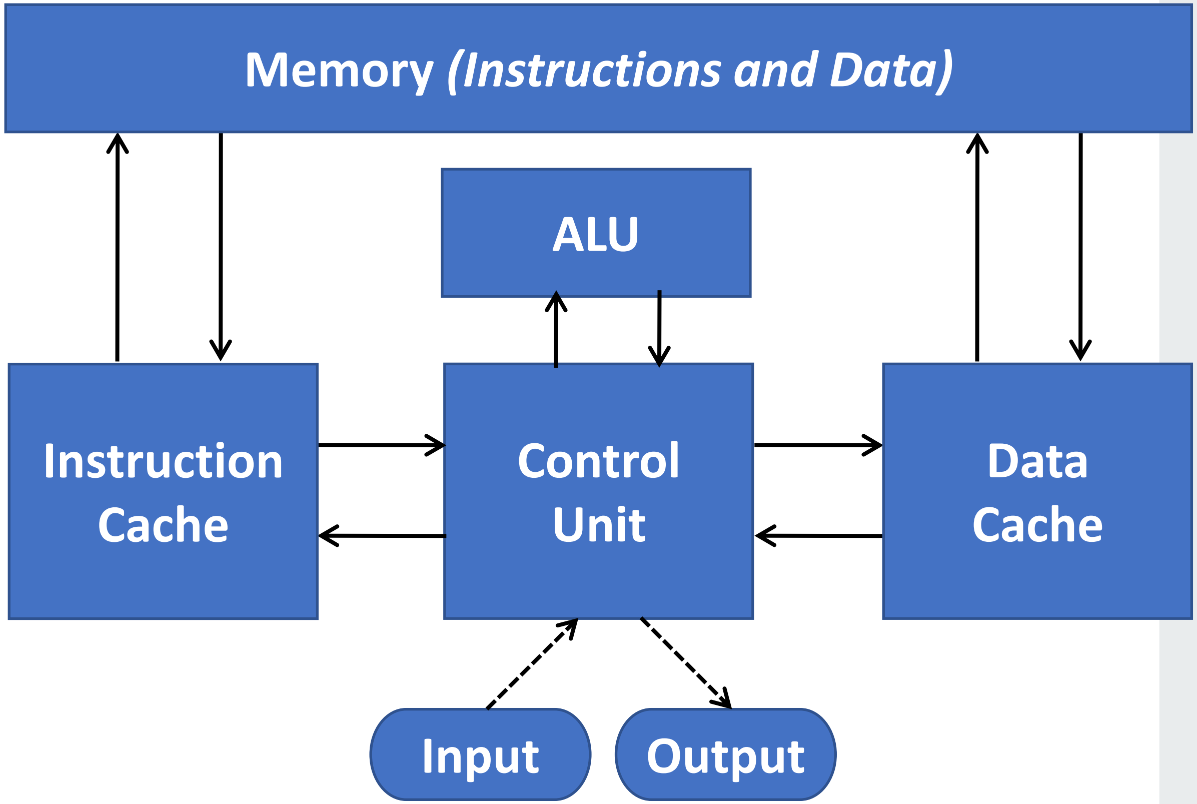


A Harvard architecture scheme

Question: **Which one is more expensive**: Von Neumman vs. Harvard structure?

Modified Harvard Architecture

- It separate instruction and data caches **internally**
- But a single unified main memory is still visible to users/ programs
- Used in chips such as ARM9, MIPS, PowerPC, x86



Note: From users/programs view, the Modified Harvard Architecture looks as a Von Neumann Architecture

Evolution of computer

- 1950s/60s/70s there are many incompatible design:
  - Supercomputers, Mainframes, etc.
  - Diverse hardware
  - Different system software - **applications need to be re-written** for each!
- 80s: Micro-electronics lead the reduced costs
  - Appeared more companies …
  - and soon, kit computers and commodity software appeared
- Late 80s-90s: Wave of commodity microcomputers (PCs)
- 2000s/10s: “Computers everywhere” (phones, watches, etc.)

All are von-Neumann architectures

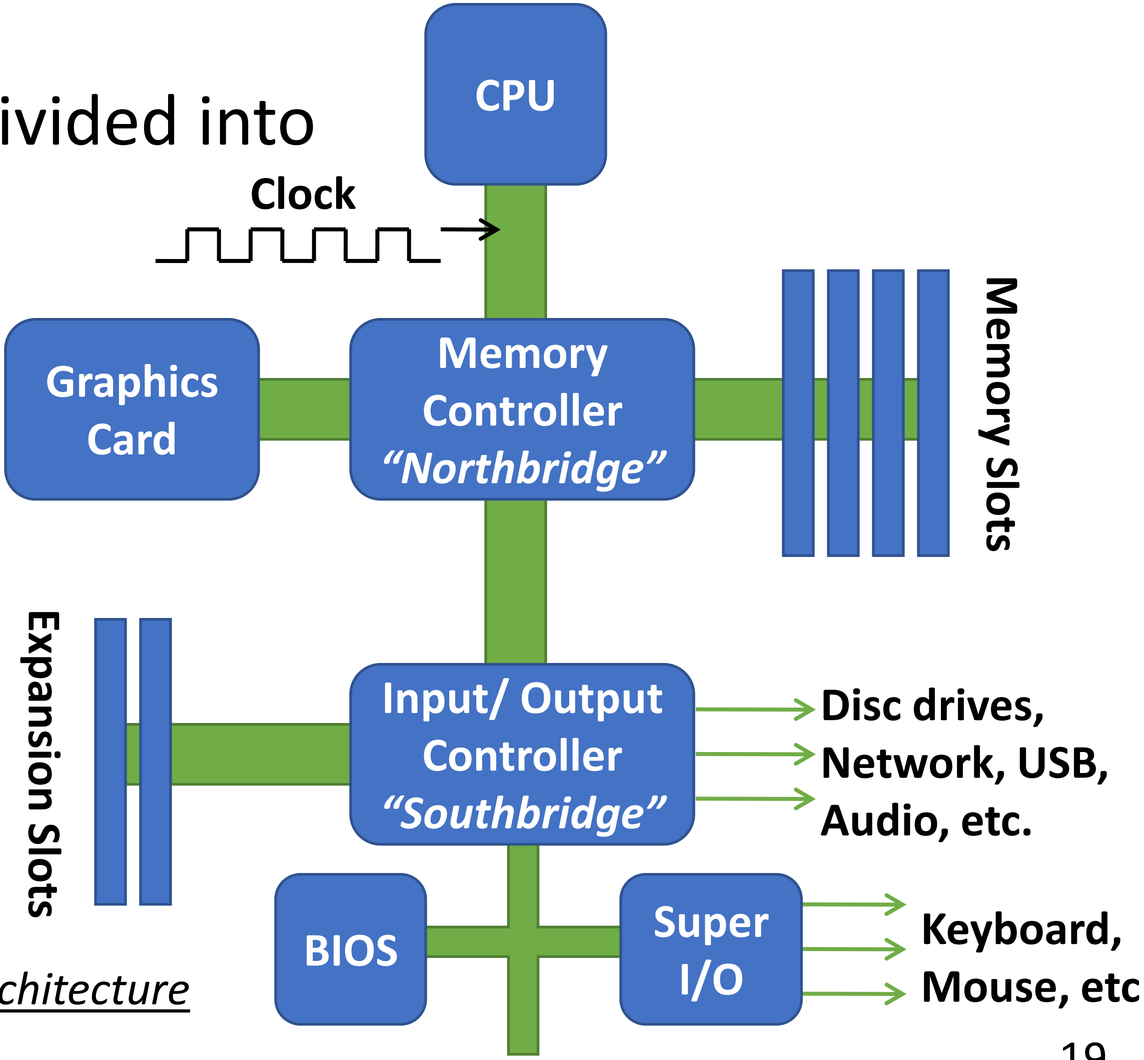
A (fairly) modern PC architecture

- The architecture is divided into *regions* according to speed of operation

Higher speed

↕

Lower speed


CPU
Clock
Graphics Card
Memory Controller "Northbridge"
Memory Slots
Expansion Slots
Input/ Output Controller "Southbridge"
Disc drives, Network, USB, Audio, etc.
BIOS
Super I/O
Keyboard, Mouse, etc

Still a von Neumann architecture

The faster, the better…

- Below are several popular metrics of computer "speed":
  - Clock rate
    - E.g. a 1.87GHz processor makes 1.87 billion ticks per second
    - But, different instructions may take different numbers of ticks – sometimes unfair as a comparing metric
  - Millions of instructions per second (MIPS)
    - a better indication of speed, but it depends on which instructions are counted (number of instructions)
    - Different results for different programs – again could be unfair
  - Floating point operations per second (FLOPS)
    - Arguably a better indication of speed "where it counts"–again maybe unfair

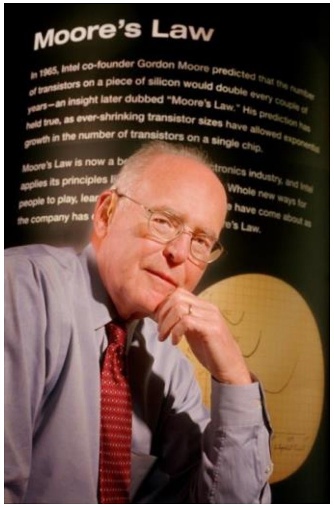
None of these metrics are ideal! Also, none of these measures take into account input/output speed.

Limiting factors on speed

- Density limitations
  - Number of transistors per square inch
  - “Moore’s Law” (1965, updated 1975):

  transistor number on a silicon doubles

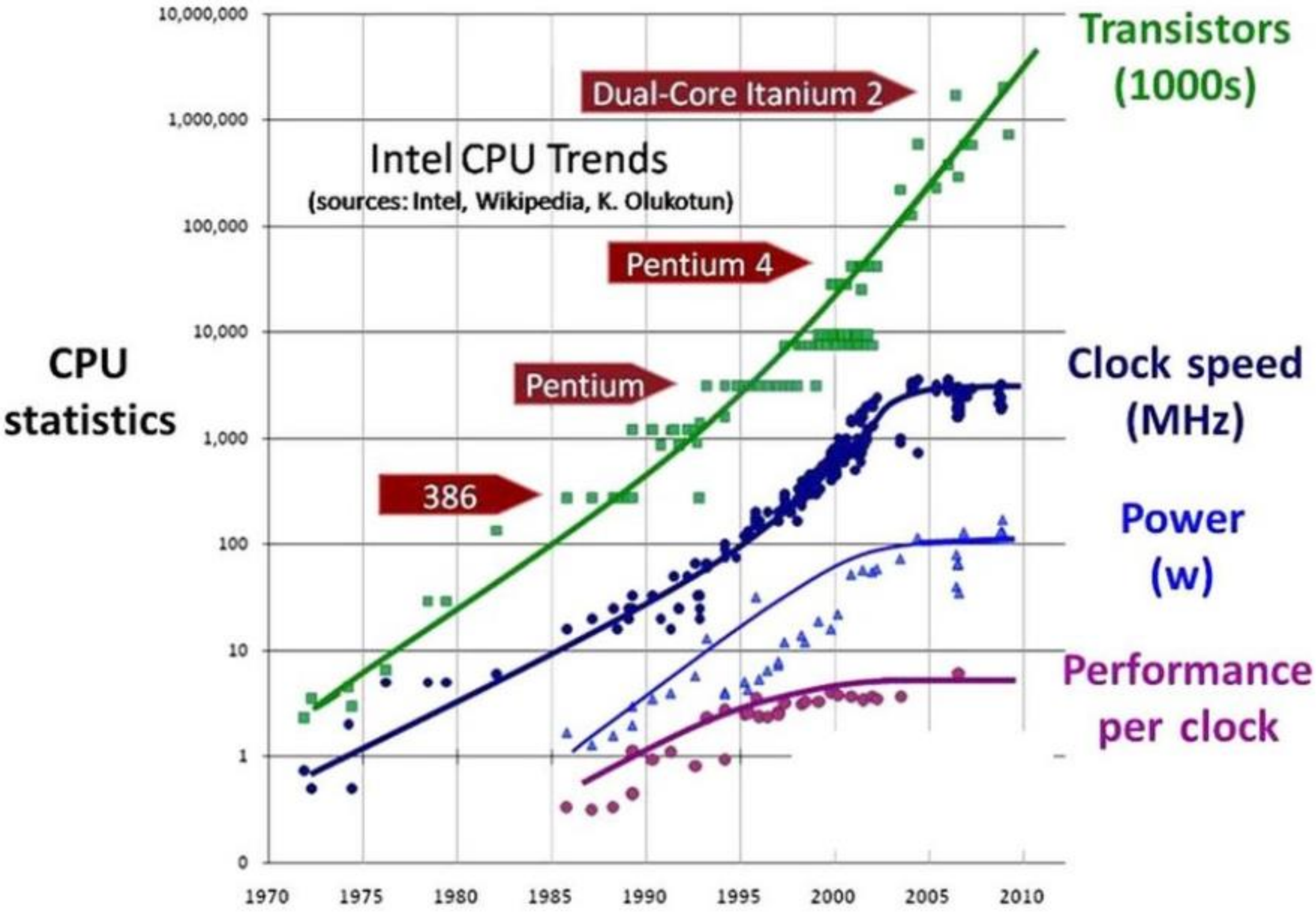
  every 2 years



- **Power limitations (critical challenge)**
  - Around 1/3 of the power used to propagate the clock signal around the processor,
  - So, power and heat problems **increase** as clock rate **increases** – Cooling becomes very challenging to accommodate this problem

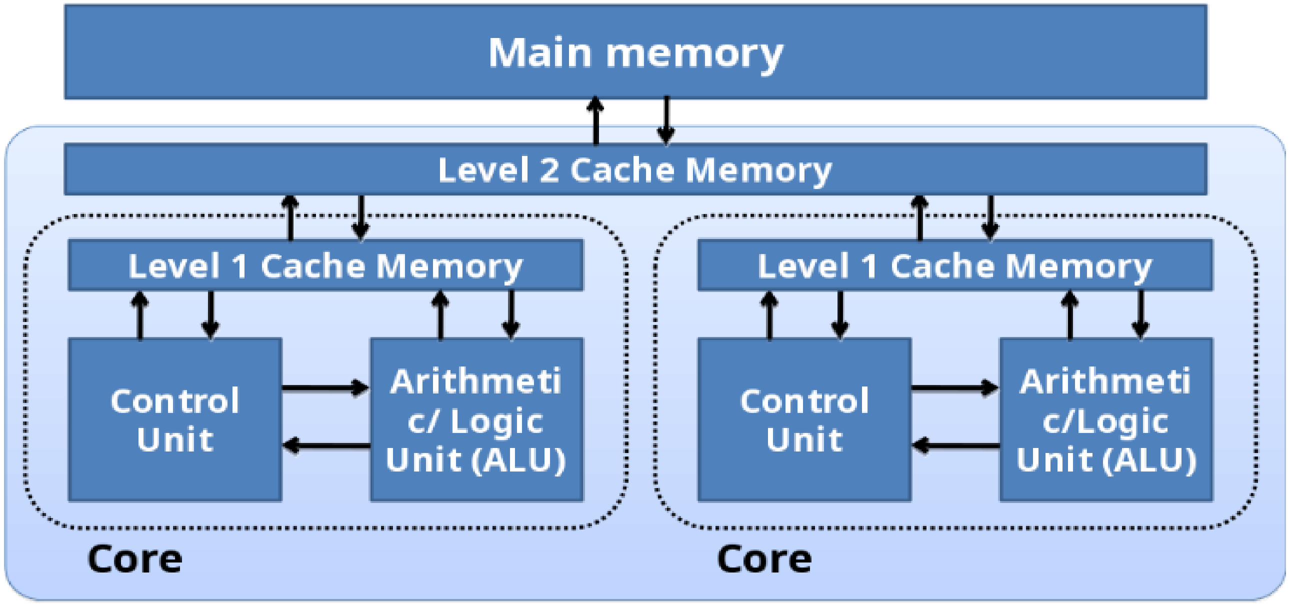
Background material on Moodle: Gordon Moore’s 1965 paper; Gordon Moore’s 1975 speech

CPU design trends



As performance demands increase…

- The inability to significantly increase the clock speed has led most CPU manufacturers to focus on multi-core processors as an alternative method to improve performance.



A dual-core processor

Coarser-grained **parallelism: Clustering**

- We can increase performance by linking computers using high-speed networks:

- Leads to idea of “blade servers”
  - Obviously they don’t all need screens, etc.
- Applications run across the cluster (ideally)
  - Although, some applications can’t easily be decomposed in this way

Summary

- We understood “what is a computer” .
- Learnt a brief Computer history.
- We then learnt computer architecture.
- Learnt about Von Neumman vs Harvard architecture vs. Modified Harvard Architecture.
- Moore's Law and Limiting factors on computer speed.

- Next: How to represent numbers in a computer.