IT Basics
Introductory Lecture

Faculty of Cybernetics, Statistics and Economic Informatics – BUES

Prof. Răzvan ZOTA, Ph.D.

zota@ase.ro

https://zota.ase.ro/itb

Contents

1. Introduction. Short computer history. The basic architecture of a PC.
2. Information theory
3. Numbering basis for a computer
4. Digital logic basics for a computer
5. The microprocessor (+ ASM - lab)
6. Memory and system buses
7. Computer peripherals
8. Storage devices
9. Computer networks, IoT, Cloud Computing, AI – basic notions

Bibliography

1. The Architecture of Computer Hardware, Systems Software, & Networking An Information Technology Approach 4th ed - I. Englander-Wiley-2009

2. Bazele Tehnologiei Informaţiei, Floarea Năstase, Răzvan Zota, Ed. ASE, 2014.

3. https://mihai-gheorghe.gitbook.io/bti-suport-curs/01-informatia

4. https://cs-fundamentals.streamlit.app/

Laboratory contents

Labs 1-7. Applications for:

- Information theory
- Numbering basics and different coding representations
- Digital logic basics
- Intel ASM (Assembler)

Tests after each seminar (exception: last one)

Grading method

The final grade is calculated with the formula:

NF = 0.7xE + 0.3xS if and only if E >= 5. If E<5, NF=[E];

where:

-NF represents the final grade,

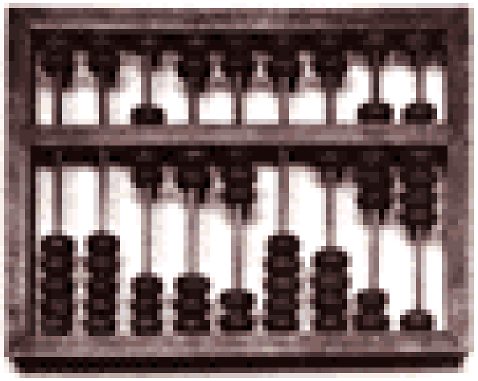
-E represents the grade from the written exam,

-S represents the grade from the seminar.

A minimum of 5 points (before rounding) is required for promotion in the written exam (E).

Short history of the computers
The abacus: the first "Automatic Computer"

- As we know it today, it appears (documented) for the first time around 1200 in China *(suan-pan)* from the need to automate the counting process



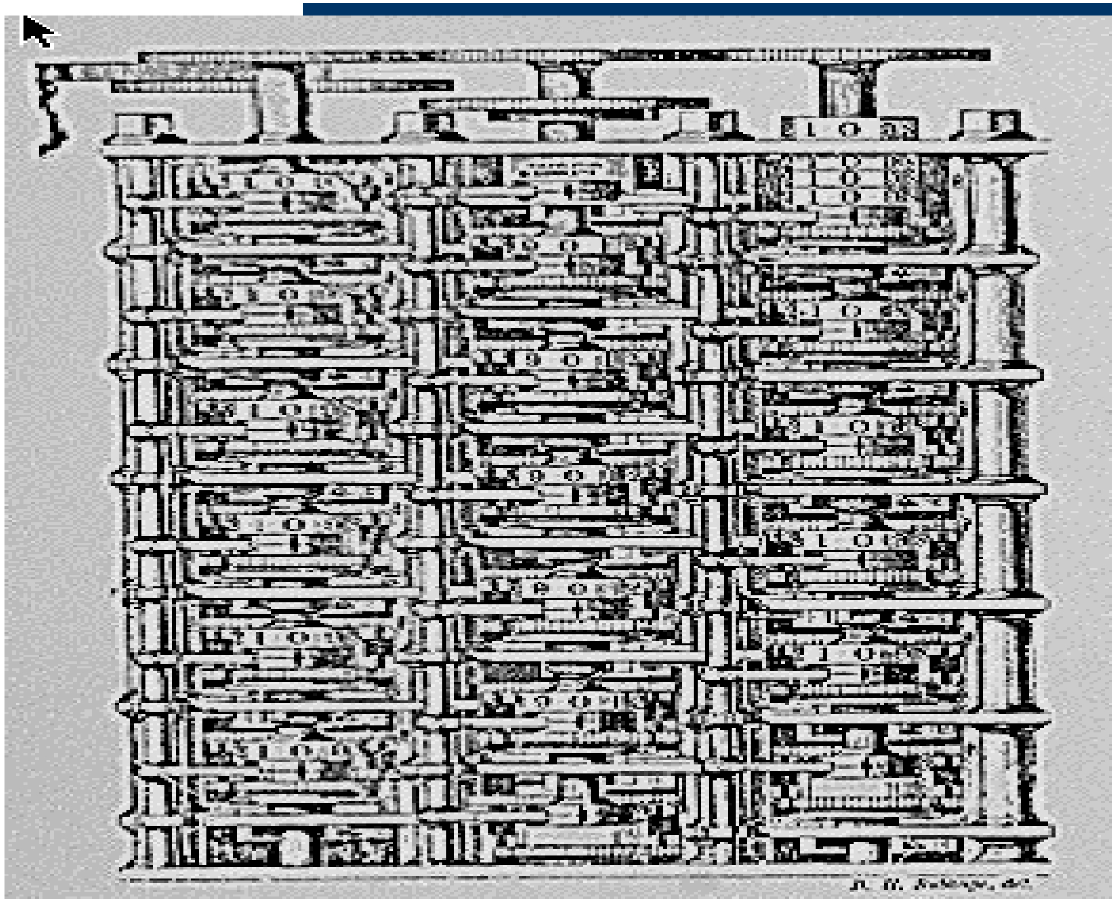
Pioneers - Blaise Pascal (1623-1662)

- French matematician who invented the first operational computer machine
- “Arithmetic Machine” – introduced in 1642
  - Addition and substraction
  - Substraction was made by using complementary techniques (similar with those used in the modern computers)
  - Multiplying and dividing were implemented by using multiple additions/substractions

Pioneers - Charles Babbage (1791-1871)

- British matematician who invented the first device which can be considered a computer in the modern sense of the word
- They were computed tables of logarithmic and trigonometric functions; operated by men called "***computers***"
  - "Difference Engine" (1822) – partly built
  - "Analytical Engine" (1830)
- The "Difference Engine" was built after using the original sketches by a team at the Science Museum in London.
  - 8000 parts
  - It weighs 5 tons, aprox. 3 meters in width, 2 ½ in length
  - The equipment had made the first calculus sequence in 1991 with a precision of 31 decimal digits

“Difference Engine”



Claude Shannon

- Around 1850 the British mathematician George Boole have invented “The Boole Algebra”
- The Boole Algebra was relatively unknown and unused till 1938
- The master thesis of C. Shannon from 1938 have demonstrated how the concepts of “TRUE” and “FALSE” may be used to represent the switches functionality of the electronic circuits

Howard Aiken and IBM Harvard Mark I computer

- Harvard Mark 1 *(IBM Automatic Controlled Calculator)* was built between 1939 and 1944
- It was made from multiple computers working for different parts of the same problem under the supervision of a unique control unit
- Built from switches, relays and other mechanical devices
- It used 750.000 components and had 16 m in length, 2 ½ m high and had a weight of 5 tons
- Numbers of 23 digits
  - A multiplication was about 4 seconds long
  - A division took about 10 seconds to complete

William Mauchly, J. Presper Eckert - ENIAC - Electronic Numerical Integrator And Computer

- Was built at the University of Pennsylvania (1943-1946)
- 3 m high, 30 sqm area, 30 tons
- 18.000 vacuum tubes
- 150 kW of power (enough to provide the light for a small village)
- The key problem with this type of computer was the reliability:
  - aprox. 50 tubes were replaced every day
- 1943 - Eckert and Mauchly had initiated the concept of creating a program stored in a computer using an internal memory (storing both instructions and data)

ENIAC - 1946

The next generations

- EDVAC - Electronic Discrete Variable Automatic Computer
  - 4000 tubes
- EDSAC - Electronic Delay Storage Automatic Calculator (1949)
  - 3000 tubes
- UNIVAC I - Universal Automatic Computer (1951)
  - First commercial computer
- ILLIAC I (1949)
  - Built at the Illinois University, the first computer owned by an academic institution

John Von Neumann

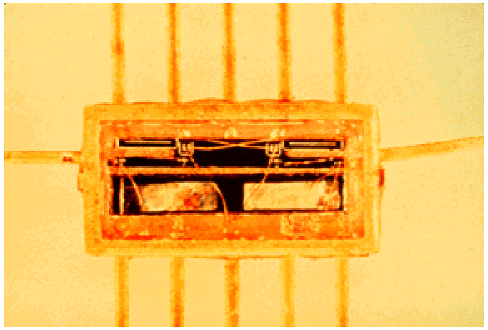
- The great mathematician Von Neumann have worked as a consultant for the ENIAC and EDVAC projects
- *The first version of a report about EDVAC* – 1945
contained the basic elements of a computer stored program:
  - **A memory** for both data and instructions
  - **A computing unit** for doing arithmetic and logical calculus of the data
  - **A control unit** which could translate an instruction from the memory and select different actions using the results of the previous operations

The first transistor

- Bell Laboratories begun the research in the field of semiconductors in 1945
- William Shockley, Walter Brattain and John Bardeen have created the first transistor on December 23rd, 1947
  - They took a break to celebrate the Christmas before publishing the event; this is the reason why some books indicate that the first transistor was created in 1948

The first integrated circuit

- Jack Kilby (Texas Instruments) in 1958 have succeeded to combine more components on a single piece of semiconductor
- In 1961 Fairchild and Texas Instruments have developed the first commercial integrated circuits which contained basic logical functions
  - 2 logical gates (4 bipolar transistors and 4 resistors)
- In 1970 Fairchild had introduced for the first time the memory of 256 bits static RAM



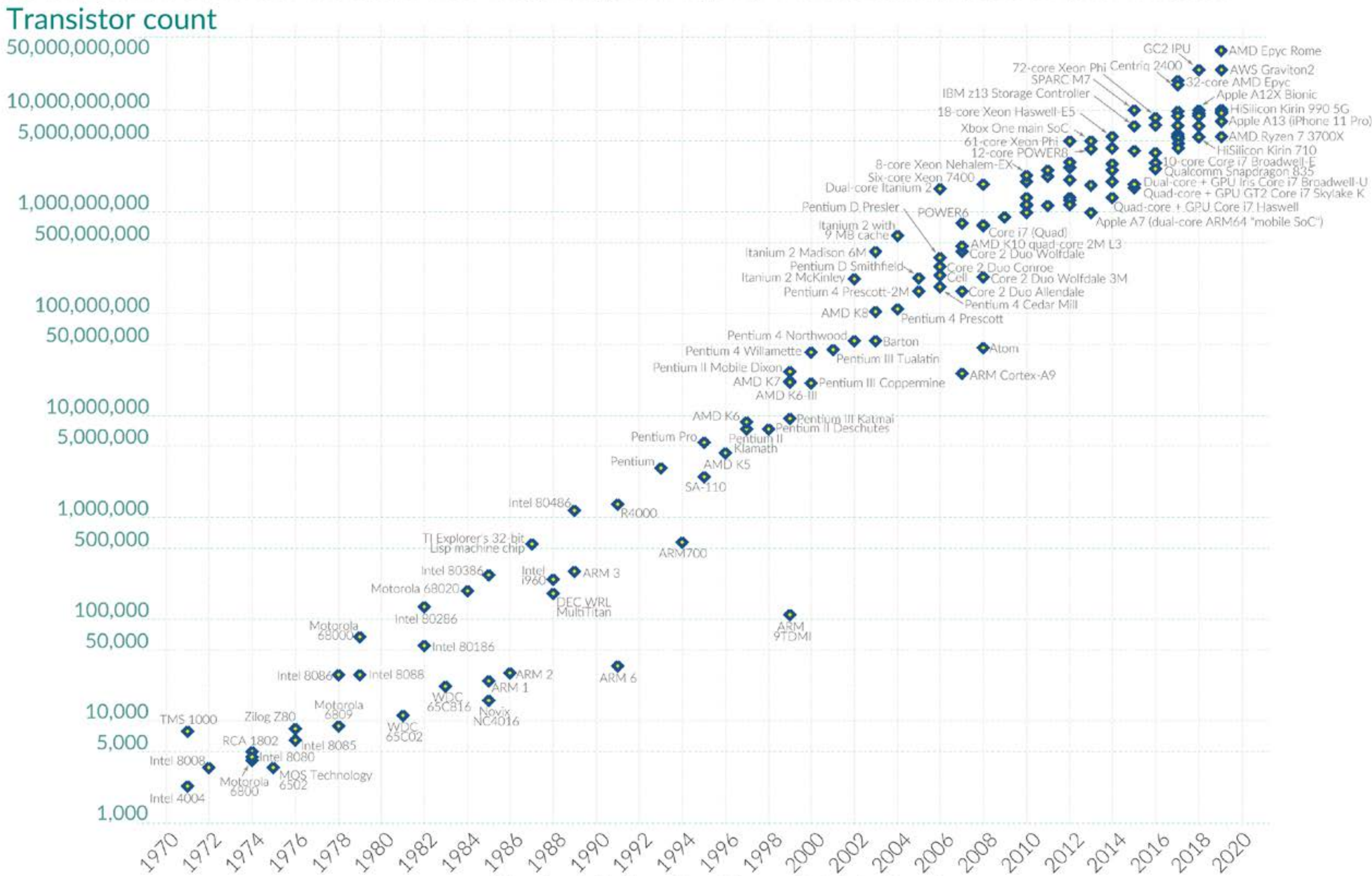
Moore’s law

- In 1964 Gordon Moore foreseen that the number of transistors of a chip will double in a time interval of aprox. 18-24 months (see next slide)

Moore's Law: The number of transistors on microchips doubles every two years

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

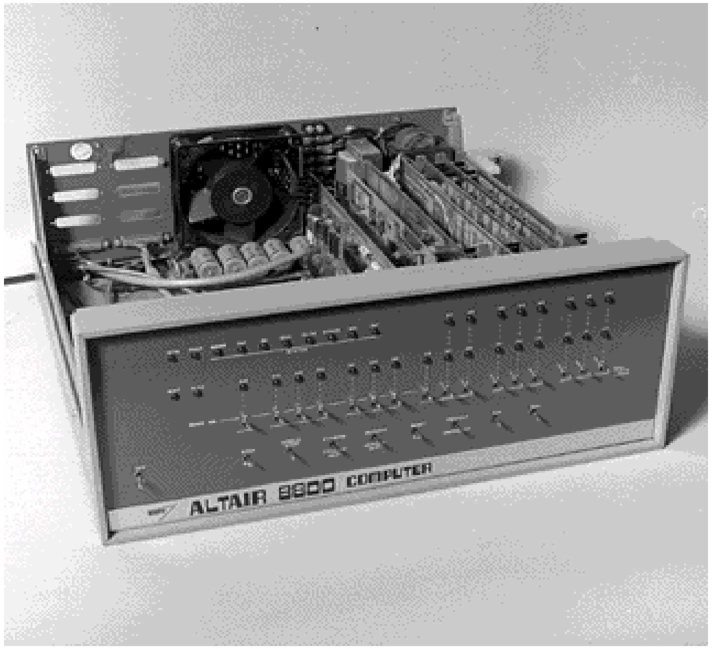
OurWorldinData.org – Research and data to make progress against the world's largest problems.

The path to the first personal computer (PC)

- Computers are beginning to use transistors (the 60ies)
- “*Big iron” years*: IBM mainframes
- In 1970 the Japanese computer company Busicom asked Intel a set of 12 IC to use it in a new computer
  - T. Hoff, project engineer at Intel, inspired by this demand, have designed the first microprocessor, called 4004
  - 2300 transistors; 60.000 operations per sec
- The first general purpose microprocessor - 8080, introduced by Intel in 1974
  - 8-bits, 4500 transistors, 200.000 operations per sec
- Other processors: Motorola 6800, MOS Technology 6502, Zilog Z80

Personal computers

- Ed Roberts made Altair 8800 (1974)
  - Based on 8080
  - Price of 375$
  - No keyboard, no screen, no storage capacity
  - 4k memory, programmable by a central panel with switches
- Bill Gates and Paul Allen are founding Microsoft (1975)
  - BASIC 2.0 with Altair 8800
  - The first high level programming language available on a PC



Personal computers (cont.)

- S. Wozniak and S. Jobs:
  - Apple 1 - 1976
  - Apple II - 1977
  - 16k ROM, 4k of RAM, keyboard and colour display
  - Price of 1300$, in 1977 business of 700.000 $ and in 1978 7 mil.
- TRS-80 (based on Z80) from Radio Shack - 1977
  - 4k ROM, 4k RAM, keyboard and tape
  - price 600$
- The first PC (Personal Computer) from IBM - 1981
  - Microprocessor on 16-bits 8088, ROM BASIC, floppy-disc 360K, DOS 1.0
  - price1365 $

Personal computers (cont.)

- 1983 IBM XT has hard-disk (10Mb costs 3000$)
- 1985 Intel is introducing 80386
  - The first member on 32-bits in the 80x86 family
- 1986 Compaq is introducing the first system based on 80386
- 1989 Intel is introducing 80486 (with math coprocessor)
- 1992 Intel Pentium (64-bits) memory bus
  - AMD, Cyrix 486 compatible processors
- 1996 Intel Pentium Pro
- 1998 Intel Pentium II
- 2000 Intel Pentium IV la 1.5 GHz

Personal computers (cont.)

- Pentium M, Celeron M – 2003
- Intel Core (65nm) Duo/Solo – Jan. 2006
- Dual Core Xeon – 2006
- Intel Core 2 (65nm) Duo – July 2006
- **Intel Core i3, i5, i7** (45nm) – 2009 (***Nehalem***) with variants of 2,4,6,8,10,12 *core (731 million transistors for quad-core)*
- *Intel Core i3, i5, i7 – 4th generation Intel Core Processors – 2013 (Haswell arch.)*

Personal computers (cont.)

- Intel's 8th generation: Core i9

Intel® Core™ i9-8950HK Processor

- 12 MB SmartCache Cache
- 6 Cores
- 12 Threads
- 4.80 GHz Max Turbo Frequency
- HK - High performance graphics, unlocked
- 8th Generation

- 9th Generation from Intel: Core i9 – 8 cores, 16 MB Cache, 5 GHz Max Turbo Frequency
- 10th Gen Intel: Core i9 – 10 cores, 20 MB Cache
- 11th Generation Intel: Core i9 – 8 cores, 16 MB Cache, 5.20 GHz Max Turbo Frequency
- 12th and 13th Gen Intel® Core™ - up to 24 cores

Portable computers (laptop) and PDA

iPhone (and not only)!

John von Neumann legacy

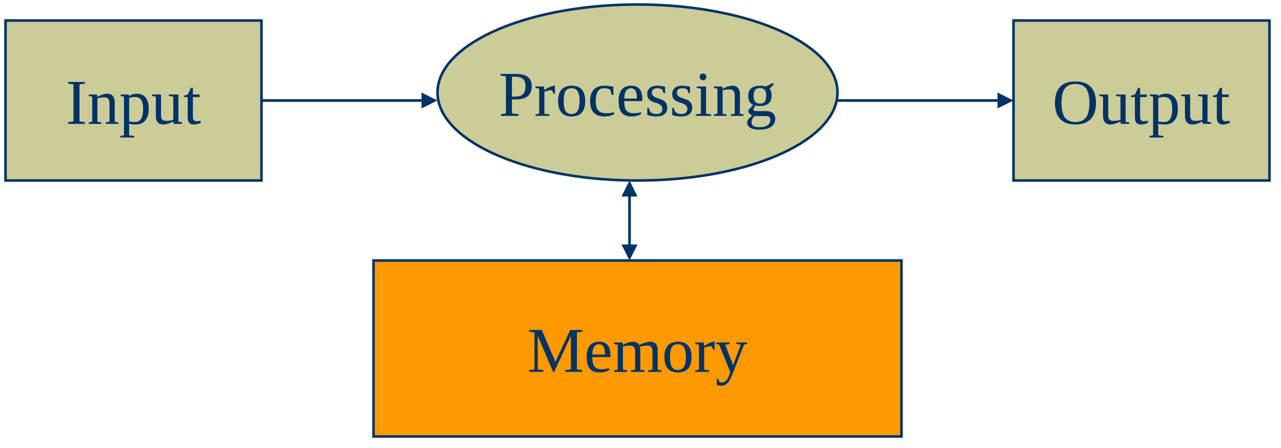
Hardware

Software

- Operating systems (are offering the operating medium for applications – used by these to access the computer’s resources)
    - Are specific to the platform designed for
- Applications (text processors, database programs, web browsers, etc.)

I/O, processing, storage

- Input
- Processing
- Output
- Information storage



Flynn's taxonomy – parallel calculus

- ***SISD -Single Instruction (Stream), Single Data (Stream)***

All von Neumann computers.

Ex. IBM 370, DEC VAX, SUN, IBM PC, MacIntosh.

- ***SIMD - Single Instruction (Stream), Multiple Data (Stream)***

One instruction flow applied on several data sets in parallel.

Ex. CRAY-1, DAP CM-1, WARP, CM-2, ILLIAC IV.

- ***MISD - Multiple Instruction (Stream), Single Data (Stream)***

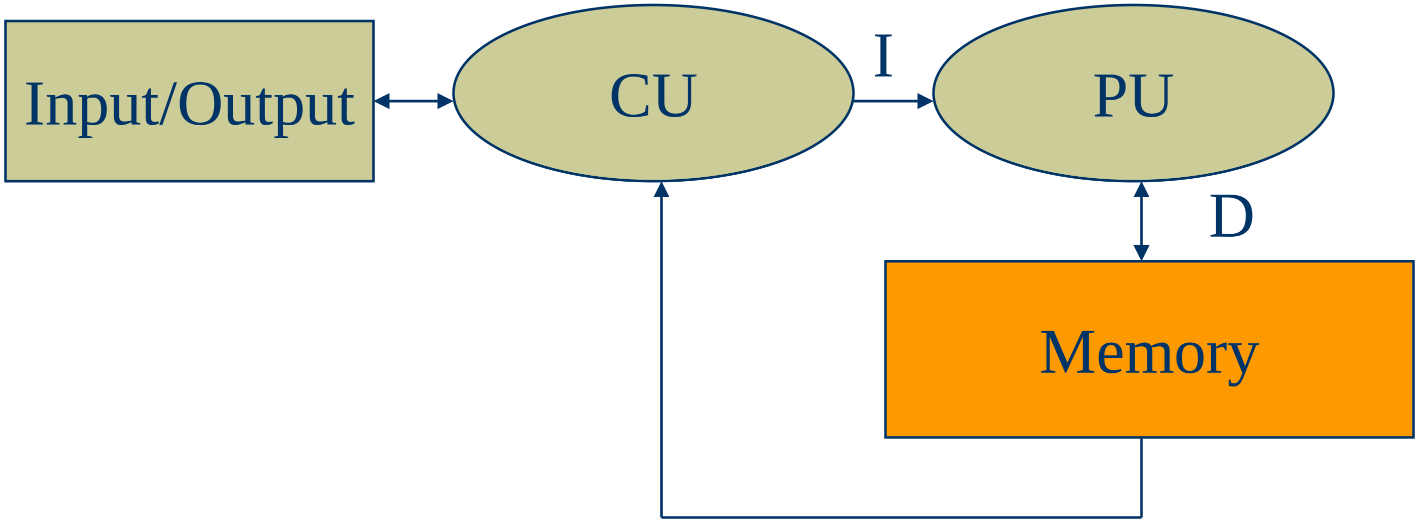
More instructions operating to the same set of data

- ***MIMD - Multiple Instruction (Stream), Multiple Data (Stream)***

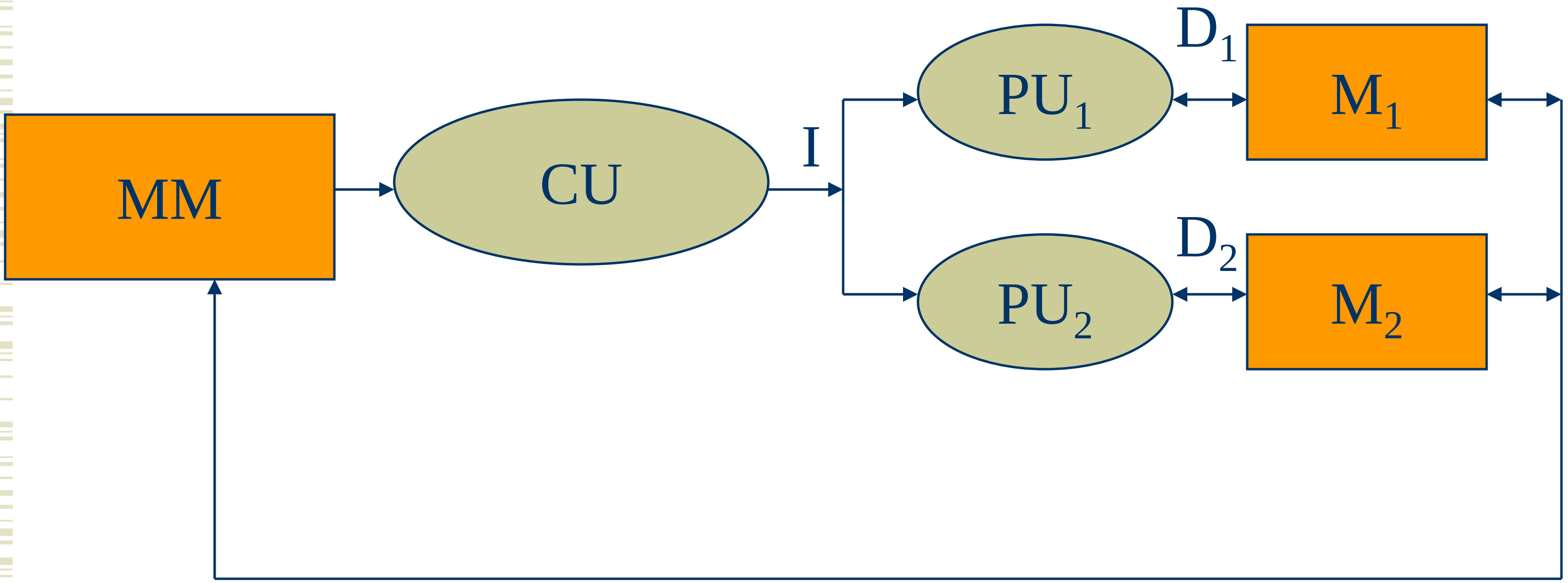
More independent CPU operating as parts of a bigger system. The majority of parallel processors are in this category.

Ex. Transputere, Supernode, DADO, N-cube, Ultracomputer, Butterfly, Alliant, Sequent Balance, CRAY X-MP.

SISD (Single Instruction, Single Data)



SIMD (Single Instruction, Multiple Data)



Obs. Like SPMD

SIMD Examples

SIMD (Single Instruction, Multiple Data) is a type of parallel processing architecture that allows a single instruction to perform the same operation on multiple data elements simultaneously. SIMD is commonly used in various computer systems, especially in multimedia and scientific applications where processing large amounts of data in parallel is beneficial. Here are some examples of computers and processors that use SIMD technology:

1. Intel SSE (Streaming SIMD Extensions):

- Intel introduced SSE as an extension to their x86 architecture. SSE provides a set of SIMD instructions for multimedia and data processing tasks.
- Example CPUs: Intel Pentium 4, Intel Core series (e.g., Core i7), and later Intel processors.

2. AMD SSE (Streaming SIMD Extensions):

- AMD processors also support SSE instructions, offering similar SIMD capabilities as Intel processors.
- Example CPUs: AMD Athlon 64, AMD Ryzen series.

SIMD Examples

3. ARM NEON:

- ARM's NEON technology is a SIMD architecture extension for ARM processors. It is often used in mobile devices, digital signal processing, and multimedia applications.
- Example CPUs: Various ARM-based processors used in smartphones and tablets.

4. NVIDIA GPUs (Graphics Processing Units):

- Graphics cards from NVIDIA, such as those in the GeForce and Quadro series, use SIMD processing to accelerate graphics rendering and scientific computing.
- Example GPUs: NVIDIA GeForce RTX 30 series, NVIDIA Quadro P series.

5. AMD Radeon GPUs:

- AMD Radeon graphics cards also employ SIMD architecture for parallel processing, benefiting gaming and compute workloads.
- Example GPUs: AMD Radeon RX 6000 series.

SIMD Examples

6. PlayStation and Xbox Consoles:

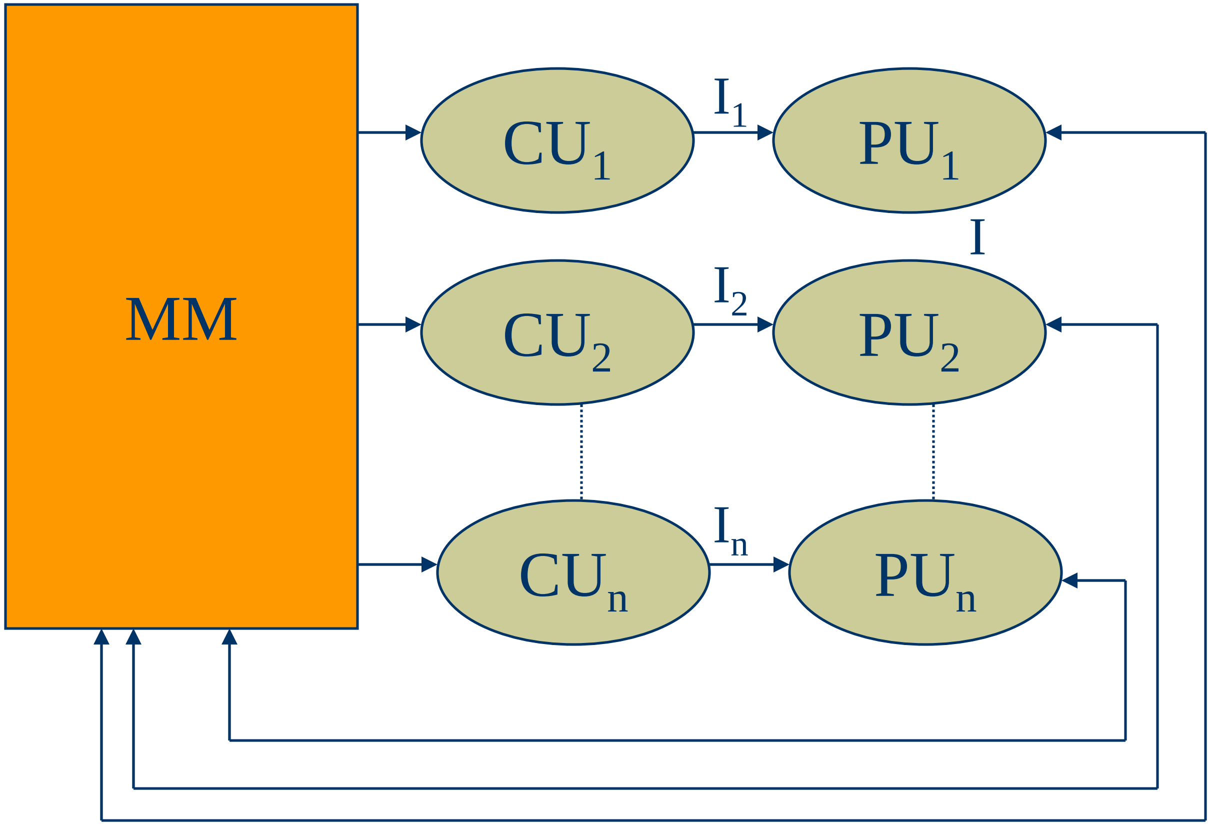
- Gaming consoles like the PlayStation and Xbox series utilize SIMD processors to enhance gaming graphics and performance.
- Example consoles: PlayStation 5 (uses AMD RDNA 2 architecture), Xbox Series X (uses AMD RDNA 2 architecture).

7. Intel Xeon Phi (formerly Knights Landing):

- Intel Xeon Phi is a family of high-performance processors designed for parallel computing and scientific simulations, making extensive use of SIMD technology.

SIMD technology plays a crucial role in optimizing performance for tasks that require repetitive operations on large datasets, such as multimedia processing, 3D graphics rendering, and scientific simulations.

MIMD (Multiple Instruction, Multiple Data)



MIMD Examples

MIMD (Multiple Instruction, Multiple Data) is a parallel processing architecture that allows multiple processors or cores to execute different instructions on different sets of data independently. This type of architecture is commonly used in multi-core processors and distributed computing systems. Here are some examples of computers and systems that use MIMD architecture:

1. Intel Multi-Core Processors:

- Many Intel processors, such as those in the Intel Core i series and Intel Xeon series, employ MIMD architecture with multiple cores.
- Example CPUs: Intel Core i7, Intel Xeon E5, E7 series.

2. AMD Multi-Core Processors:

- AMD processors, including Ryzen and EPYC series, utilize MIMD architecture with multiple cores.
- Example CPUs: AMD Ryzen 7, AMD EPYC 7000 series.

3. NVIDIA CUDA:

- NVIDIA's CUDA architecture allows for MIMD parallel processing on their GPUs (Graphics Processing Units). This is widely used for scientific simulations and GPU

MIMD Examples

4. Heterogeneous Computing Systems:

- Systems that combine CPUs and GPUs, such as NVIDIA's CUDA-enabled systems or AMD's heterogeneous systems, effectively use MIMD architecture for parallel processing tasks.

5. Distributed Computing Clusters:

- High-performance computing clusters often consist of multiple computers connected together to solve complex problems using MIMD architecture.
- Example systems: Beowulf clusters, Hadoop clusters.

6. Supercomputers:

- Supercomputers like IBM's Blue Gene series and Cray supercomputers use MIMD architecture to tackle large-scale scientific and engineering simulations.

7. Cloud Computing Instances:

- Many cloud computing providers offer virtual machines (VMs) and instances with multiple CPU cores, allowing users to harness MIMD parallelism for various workloads.

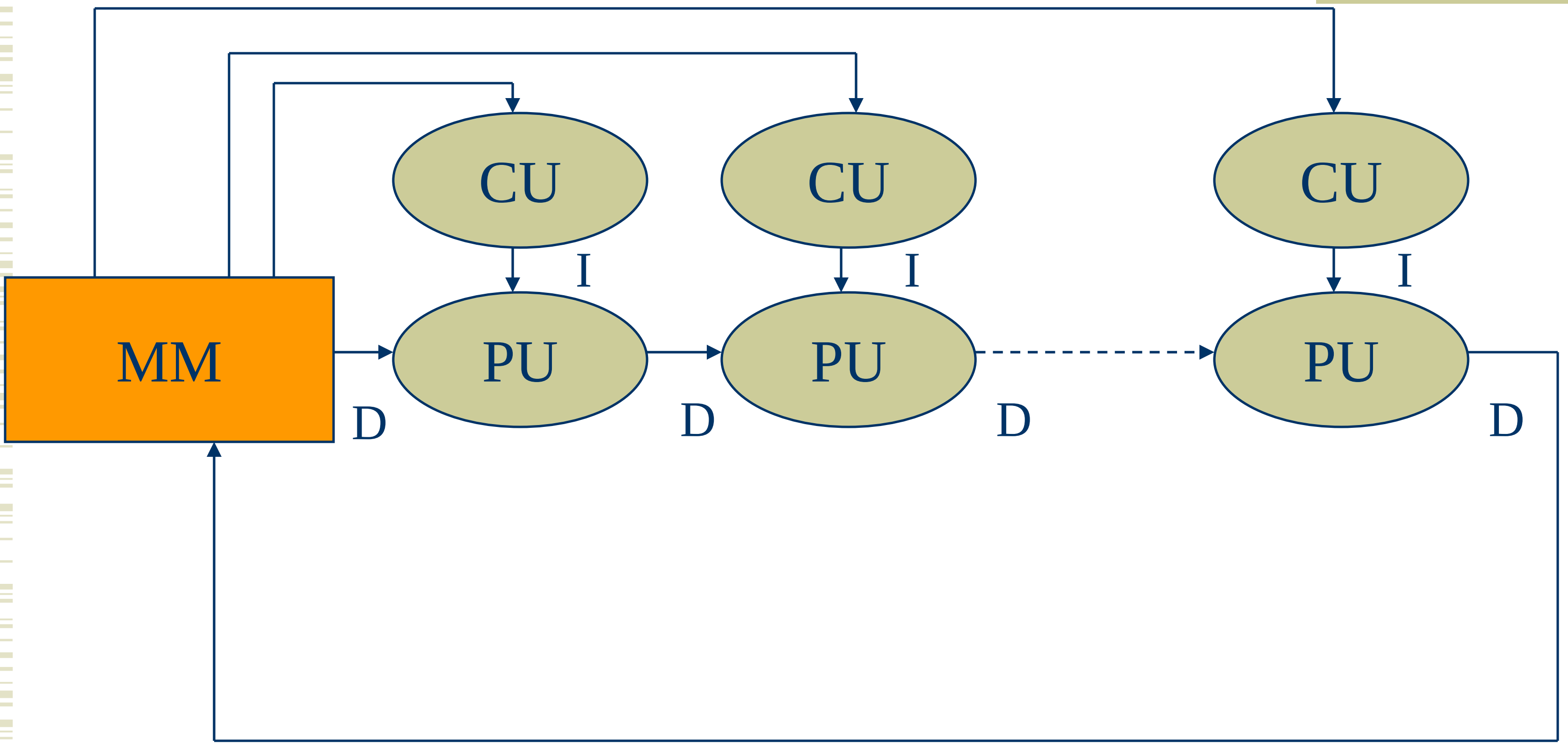
MIMD Examples

8. High-End Workstations:
- High-end workstations for tasks like 3D rendering, video editing, and scientific computing often feature multi-core processors, leveraging MIMD architecture.

9. Clusters for Scientific Research:
- Scientific research often requires MIMD processing for tasks like molecular modeling, weather simulations, and genome sequencing, using clusters of computers.

MIMD architecture is widely used in modern computing systems to improve performance and efficiency by dividing tasks among multiple processors or cores. Each processor or core can execute its own set of instructions on separate data, making it suitable for tasks that require a high degree of parallelism and concurrent processing.

MISD (Multiple Instruction, Single Data)



MISD (Multiple Instruction, Single Data)

MISD (Multiple Instruction, Single Data) is a less common parallel processing architecture compared to SIMD and MIMD. In MISD, multiple processing units or cores execute different instructions on the same set of data. MISD architecture is rarely used in practical computing applications, and it's often considered more of a theoretical concept. As a result, there are limited real-world examples of MISD systems.

A hypothetical example to illustrate the concept may be one used for *Fault-Tolerant Systems.*

In some fault-tolerant computing systems, multiple redundant processing units may operate in a MISD fashion to enhance reliability. These systems aim to detect and correct errors that may occur during data processing. Here's how it might work:

Suppose you have a critical aerospace control system responsible for ensuring the safety of a spacecraft during a mission. This system is designed to process sensor data and make decisions based on that data.

MISD (Multiple Instruction, Single Data)

1. Triple Modular Redundancy (TMR): This is a common fault-tolerant technique. Three identical processing units (A, B, and C) are used. Each unit runs the same set of instructions, but they do so independently.
2. Single Data Source: All three processing units receive the same data from the sensors, representing a single set of data.
3. Comparison and Voting: After processing the data independently, the results from each processing unit are compared. If any unit produces a different result from the other two, a voting mechanism is used to determine the correct output. For example, if units A and B produce the same output, but unit C differs, the system will consider the result from A and B as correct.
4. Error Detection and Correction: If an error is detected in one of the processing units (e.g., unit C produces an incorrect result), the system can take corrective action. It may disable the faulty unit (C) and continue to operate using the results from the two correct units (A and B).

MISD (Multiple Instruction, Single Data)

In this hypothetical scenario, the MISD architecture is used to enhance fault tolerance. While it's not a conventional use of MISD, it demonstrates the concept of multiple instructions (processing units running independent code) acting on a single set of data (sensor inputs) to achieve reliability and fault tolerance.

It's important to note that MISD is not commonly found in typical computing applications. Most real-world parallel processing systems use SIMD (Single Instruction, Multiple Data) or MIMD (Multiple Instruction, Multiple Data) architectures because they are more practical and versatile for a wide range of tasks.