



SAPIENZA
UNIVERSITÀ DI ROMA

1st Learning unit *processing unit* *architecture*

Alessandra de Vitis



1st Learning unit

1st Lesson:

Computer Architecture



In this lesson we learn :

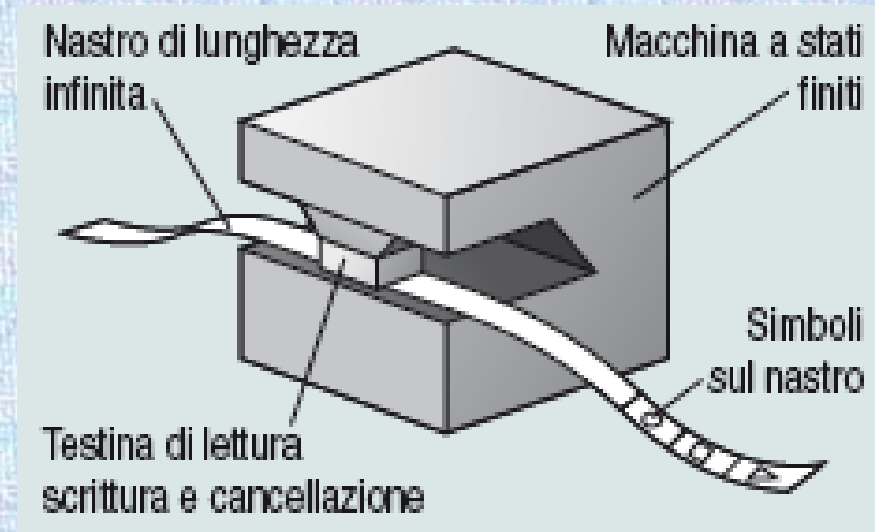
- **To know processing unit architecture**
- **To know Von Neumann and Harvard models**
- **How to recognize the role of processing unit components (CPU, RAM, I/O, BUS)**



Different Kinds of computer

A computer is a physical device that implements the operations of a **Turing machine**.

A **Turing machine** is an imaginary machine that is able to manipulate data written in an endless length ribbon





Turing Machine

**Does always exist, in principle,
a mechanic method (strictly)
by which we can determine if
every mathematical statement
is true or false?**

Turing said NO.



Computer

- **A computer runs (as Turing machine does)**
- **There are general and special purpose computer**
- **PC is a general purpose computer , a microcontroller is a special purpose computer**



Computer: how to access to resources

- A **general purpose computer** can be:
 - **monouser** (monotasking or multitasking)
 - **multiuser** (is multitasking and carries out resources **timesharing**)

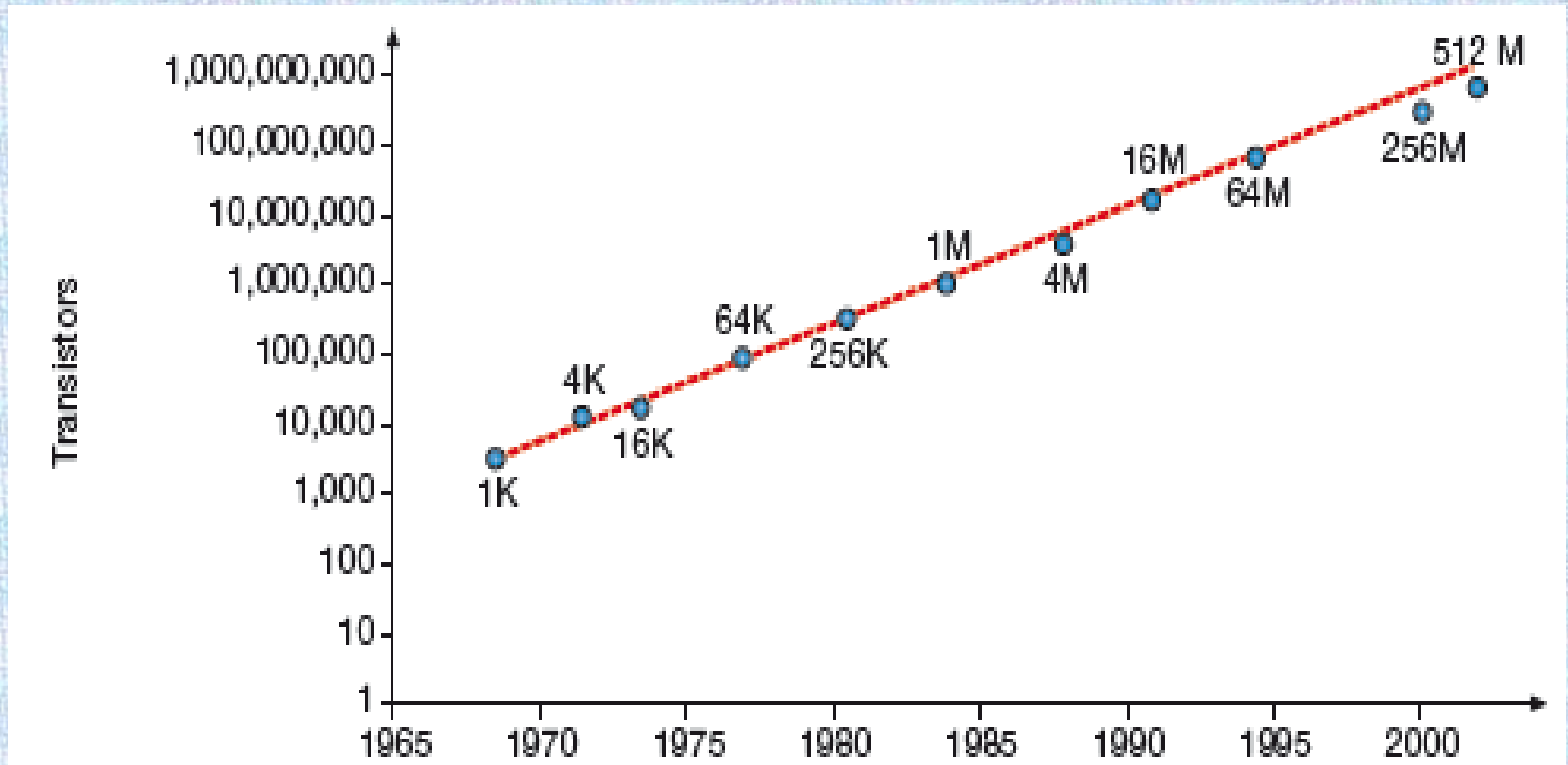


Integration scale

- Integration scale is the measure of how many transistors are built in a single integrated chip
- **SSI** (Small Scale Integration): <100 transistor
- **MSI** (Medium Scale Integration): <1000 transistor
- **LSI** (Large Scale of Integration): <10.000 transistor
- **VLSI** (Very Large Scale Integration): <100.000 transistor
- **ULSI** (Ultra Large Scale Integration): >100.000 transistor



Growth of scale integration





Different types of computers

**From
super computer
to
home computer
tablet smartphone
Videogames console**



Moore Law

- **Gordon Moore (in 1965) said that every year transistors integrable in a single chip would double.**
- **In 1975 Moore had to change his law and said that from that moment the number of integrable transistors in a single chip would double every 2 years.**



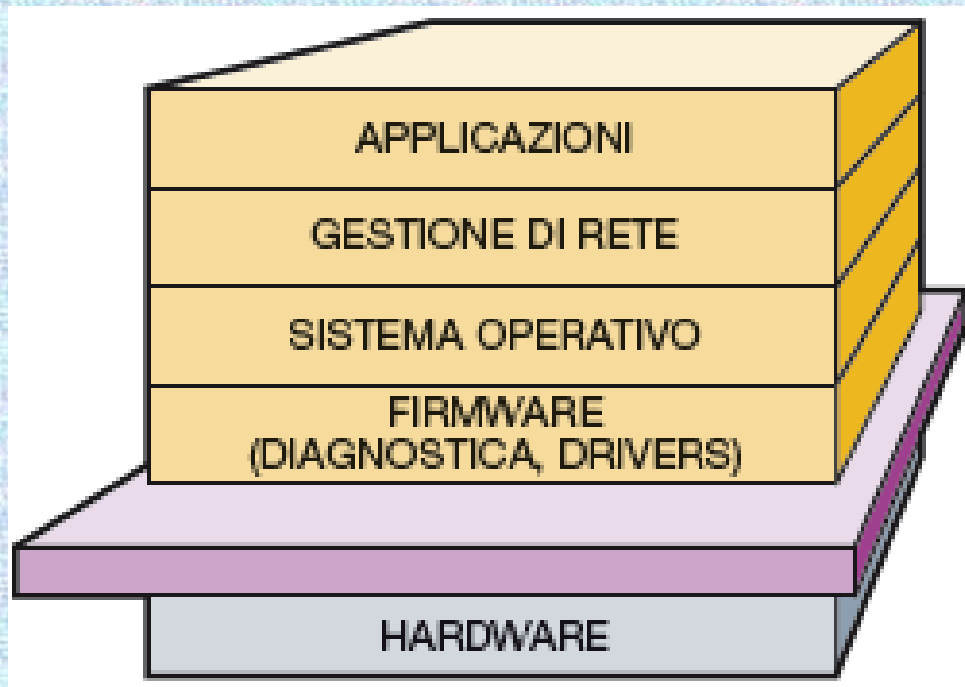
Virtual machines

- A virtual machine or **VM** make it possible to reproduce the functions of other operation systems, smartphones, or other kind of computer directly on a PC or other device with **EMULATION** process.
- VM are used to simulate or test a software without physically installing it.



Computer architecture

**Computer = Hardware +
software**





Computer architecture

- **Computer architecture is the set of concepts, and technics to define, project, evaluate a processing system.**
- **The goal is to obtain the best performance from electronic components. Electronic goal is to product more and more fast and efficient circuitery.**



Electronic components in a computer

Logical gates

Signals generators



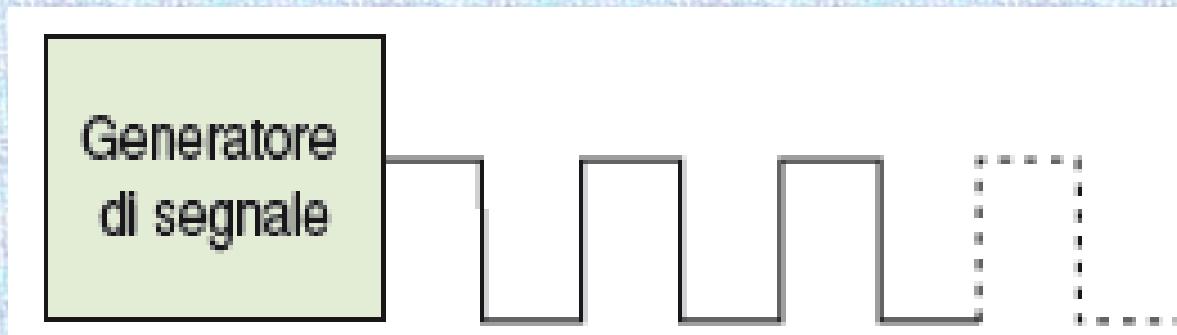
Logical gates

Name	NOT	AND	NAND	OR	NOR	XOR																																																																																	
Alg. Expr.	\bar{A}	AB	\overline{AB}	$A+B$	$\overline{A+B}$	$A \oplus B$																																																																																	
Symbol																																																																																							
Truth Table	<table border="1"> <thead> <tr> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	X	0	1	1	0	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	B	A	X	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	B	A	X	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	0
A	X																																																																																						
0	1																																																																																						
1	0																																																																																						
B	A	X																																																																																					
0	0	0																																																																																					
0	1	0																																																																																					
1	0	0																																																																																					
1	1	1																																																																																					
B	A	X																																																																																					
0	0	1																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	0																																																																																					
B	A	X																																																																																					
0	0	0																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	1																																																																																					
B	A	X																																																																																					
0	0	1																																																																																					
0	1	0																																																																																					
1	0	0																																																																																					
1	1	0																																																																																					
B	A	X																																																																																					
0	0	0																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	0																																																																																					



Signals generators

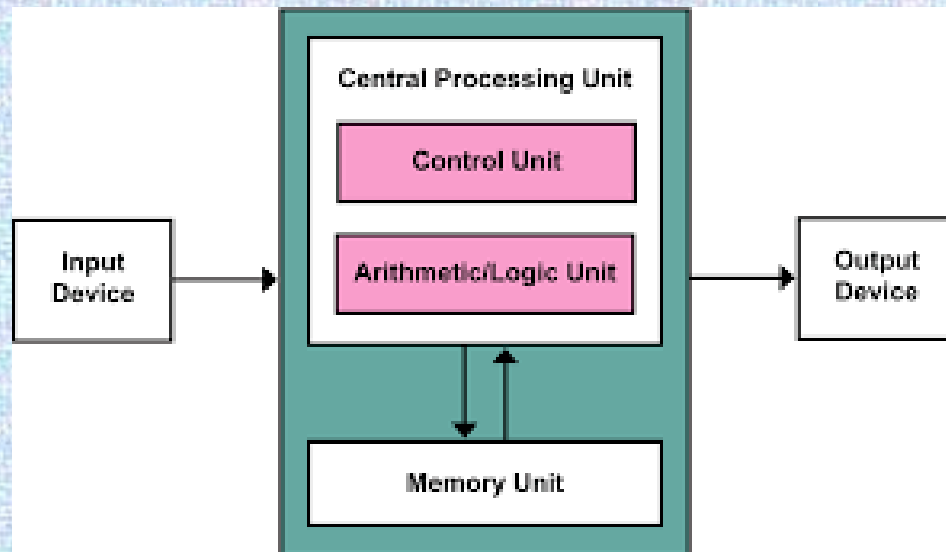
They are able to product a periodic signal useful to sincronize all hardware components in a computer.





Von Neumann Model

Von Neumann model describes the behavior of a machine stored-program computer





Von Neumann Model

Is the CPU that takes actions as:

- **Pick up or modify memory content**
- **Pick up or modify informations from input/output devices giving output informations or reading informations in input**

CPU takes actions lin a sequential way one by one very



Von Neumann Model: stored program

- **stored-program is the instructions set executed by CPU stored in computer memory.**
- **Instructions set is the program to be executed.**
- **In the memory there are instruction (in assembly) and data used in execution programs**



Harvard Model

- **Harvard model has 2 different memories : 1 for data e 1 for instructions.**
- **It is used in specialized processors as DSP or PIC.**



Memory

- **RAM (Random Access Memory) oppure ROM (Read Only Memory)**
- **RAM is random access**
- **ROM contains BIOS**
- **It is organized in locations or cells (1 byte long) with its own address (memory address)**
- **Access time: nanoseconds**



I/O Input Output

- **Input** devices are able to acquire signals from outside
- **Output** devices are able to send signals to outside.
- **Signals** are represented by bits sent and received from those devices



CPU & I/O

- CPU manages communication between I/O in **asynchronous** way with a signal called interrupt (**IRQ**)
- I/O devices are often just **controller** interfaces with the real device.



Controllers

Controllers are devices near the real device and manage communication between the device and the **BUS connected through a **communication protocol** represented by the communications rules set between **CPU** and the device**



BUS

- **Data Bus** enables data transmission from **CPU** to other elements and viceversa (bidirectional)
- **Address Bus**) contains location or I/O device address on or from **CPU** has to work (monodirectional)
- **Control Bus**) transports orders from **CPU** and returns **condizion/state** signals from devices



CPU

- **Executes data and it is represented at physical level by the microprocessor**
- **Executes program instructions, written in high level language and the translated in machine language**



Something about how CPU works

- CPU **extracts** instructions from memory, **le codes** and **executes** them
- Data transfer between variuos components (as for memory and I/O) occurs with system Buses.
- Every process are synchronized with a system
- During every time lapse CPU establish what operation run.