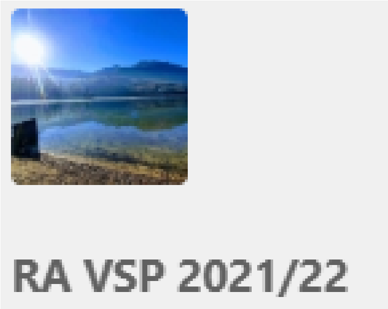
COMPUTER ARCHITECTURE

1 Introduction

- Web classroom: http://ucilnica.fri.uni-lj.si
  https://padlet.com/rawall/RAWall



- MS Teams
  - Team enter code: vf8b4ig

- Office hours: currently on Thursdays from 16:15 to 17:00 in R2.40

  Possible changes will be posted to the web classroom

  Announce: email or https://calendly.com/rrozman/govorilne (experimental)

Team CA

Tutors

Mira Trebar
mira.trebar@fri...

Žiga Pušnik
ziga.pusnik@fri....

Rok Češnovar
Rok.cesnovar@fri...

Miha Krajnc
mk7793@student.uni-lj.si

Anamari Orehar
ao6477@student.uni-lj.si

Robert Rozman
rozman@fri.uni-lj.si

- Literature:
  - Lecture content, lab. exercises and slides (also in English)
    - http://ucilnica.fri.uni-lj.si
  - MS Teams (chat, lecture notes)

RA VSP 2020/21

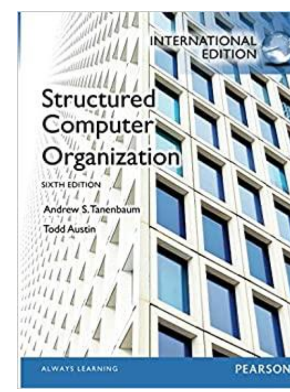
  - Common (shared) notes – Gdocs

------------------------- Skupni zapiski/Shared course notes ---------------------------------

Computer Architecture - Crowd-sourced Shared Notes

Računalniška arhitektura - Deljeni zapiski za skupno dopolnjevanje

  - Basic, (includes wider content than needed):
    - Dušan Kodek: ARHITEKTURA IN ORGANIZACIJA RAČUNALNIŠKIH SISTEMOV, Bi-TIM, 2008



  - Additional (only certain parts):
    - Andrew S. Tanenbaum: STRUCTURED COMPUTER ORGANIZATION, Sixth Edition Pearson Prentice Hall, 2013

Grades STUDIS

Course	**4.65/5 [154/161]**
Lecturer	4.74/5 [154/161]

Important :

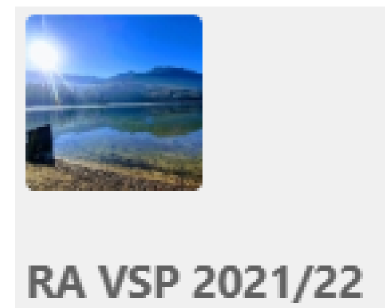
- There are no silly questions,
  - Just those that don‘t ask
- You‘re always welcome
- We all make efforts

Surveys (2018/21) - highlights:

- GOOD:
  - Kahoot
  - Energy of lecturer
  - Practical (every-day life) examples
  - Summarization on table
  - Good learning system for foreign students
- To improve:
  - Too small writting on table
  - Responses to submitted work
  - Topics Lectures <> Lab sessions

What‘s new in 2021:

- Live lectures and lab sessions
  - https://padlet.com/rawall/RAWall
    - Questions, challenges, links, …
- Platforms :
  - e-classrom http://ucilnica.fri.uni-lj.si
    - Computer Architecture - Crowd-sourced Shared Notes
  - MS Teams (board notes, communication)
    - Team entry code : vf8b4ig



- Important :
  - be active
  - cooperate, talk, ask, comment, …
  - all major documents are translated to English
  - testing of realtime Slovene-English translation – project ON

E-classroom : course content RA http://ucilnica.fri.uni-lj.si

Computer Architecture

Dashboard / My courses / ra

Splošne informacije - General info

Izvajalci - Team

- Predavatelj - Lecturer: Robert Rozman (tel. FRI 01 479 8202)
- Govorilne ure - Office hours: Četrtek (Thursday) 16:15 - 17:00 v R2.50, ali po dogovoru (e-mail) tudi izven tega termina (welcome anytime).
- Asistenti - Assistants: Žiga Pušnik, Mira Trebar, Robert Rozman.
- Tutorja: Miha Krajnc, Anamaria Orehar

Vsebina vaj - Lab work

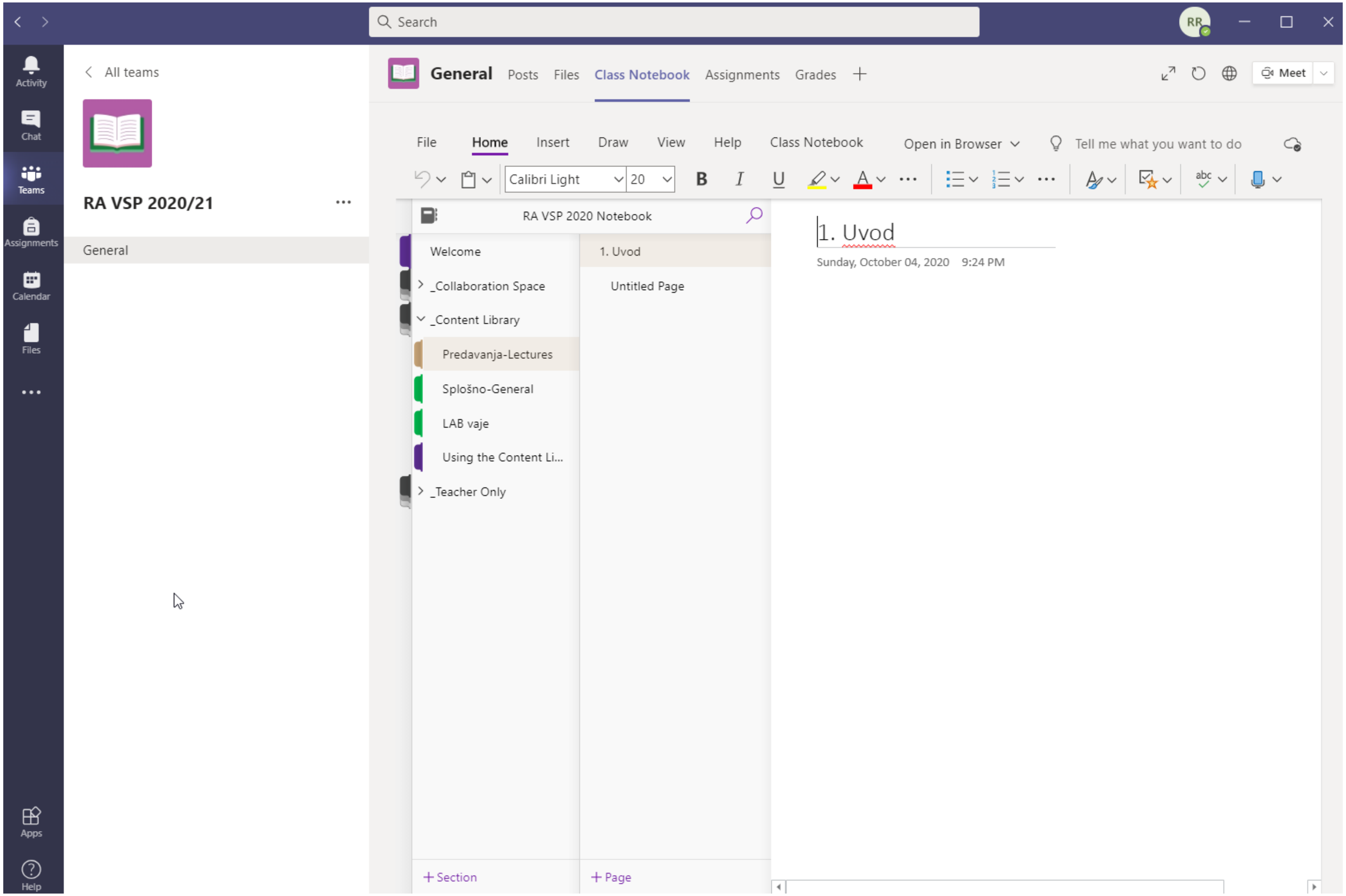
Na laboratorijskih vajah spoznamo zgradbo in programiranje mikrokrmilnika AT91SAM9260 iz družine ARM9 v zbirnem jeziku. Podobni mikrokrmilniki se upor

Zbirni jezik AT91SAM9260 spoznavamo postopno s sprotnimi vajami, ki jih rešujemo na vajah oziroma v okviru domačih nalog. Pri programiranju uporabljamo

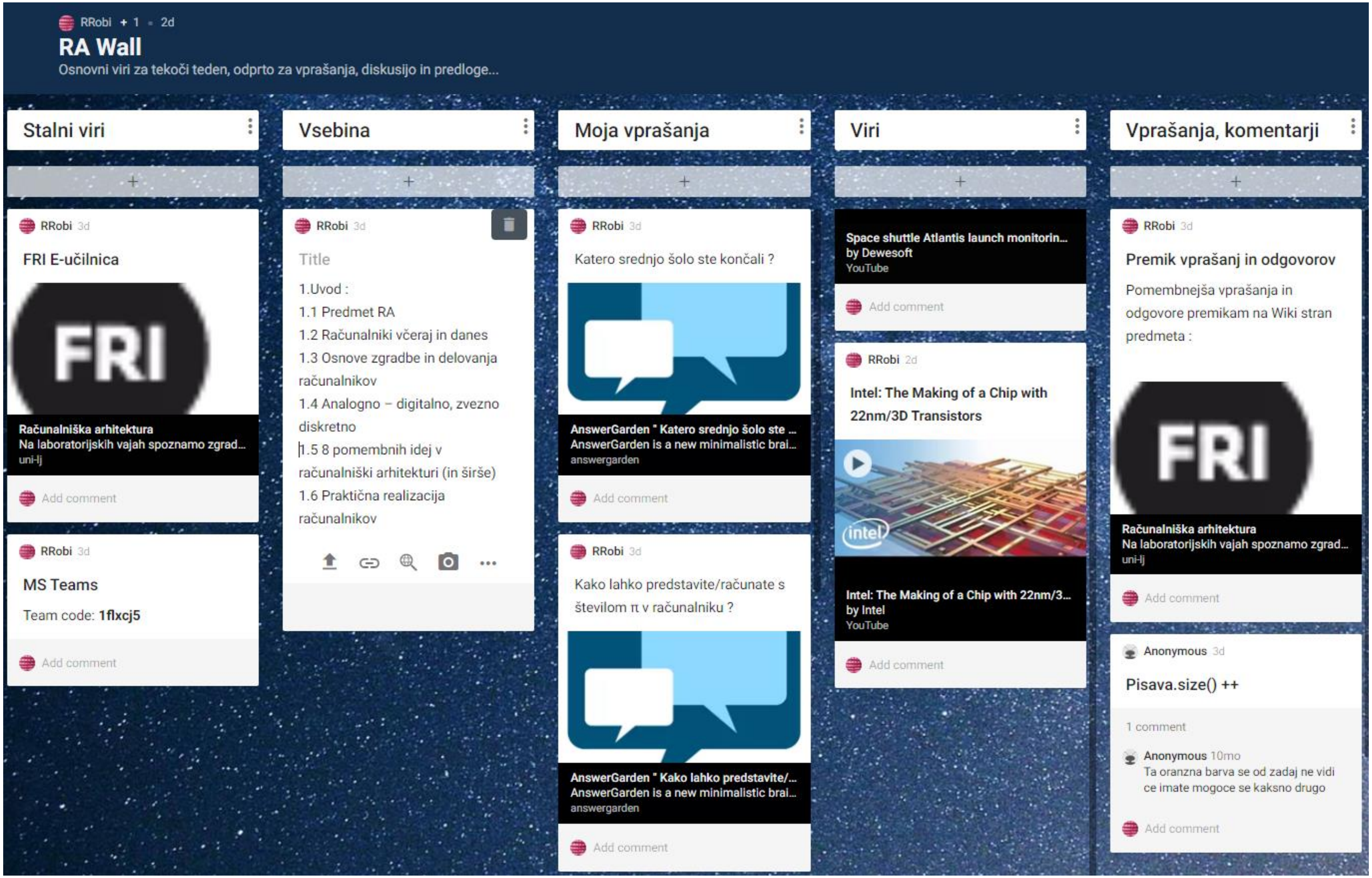
In laboratory exercises, we learn about the structure and programming of the AT91SAM9260 microcontroller from the ARM9 family in the assembly language. mobile phones, and so on.

We get familiar with the AT91SAM9260 assembly language gradually with simple and practical assignments, which we solve at lab sessions or as homework. W installed on home or laptop computers.

MS Teams: chat, OneNote notebook

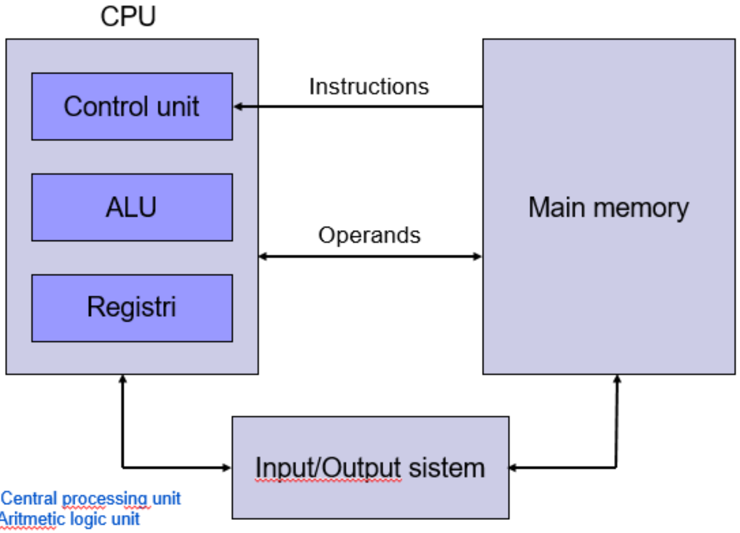


Chapter related content https://padlet.com/rawall/RAWall



What will you learn on the course of Computer Architecture?

Lectures, Lab sessions



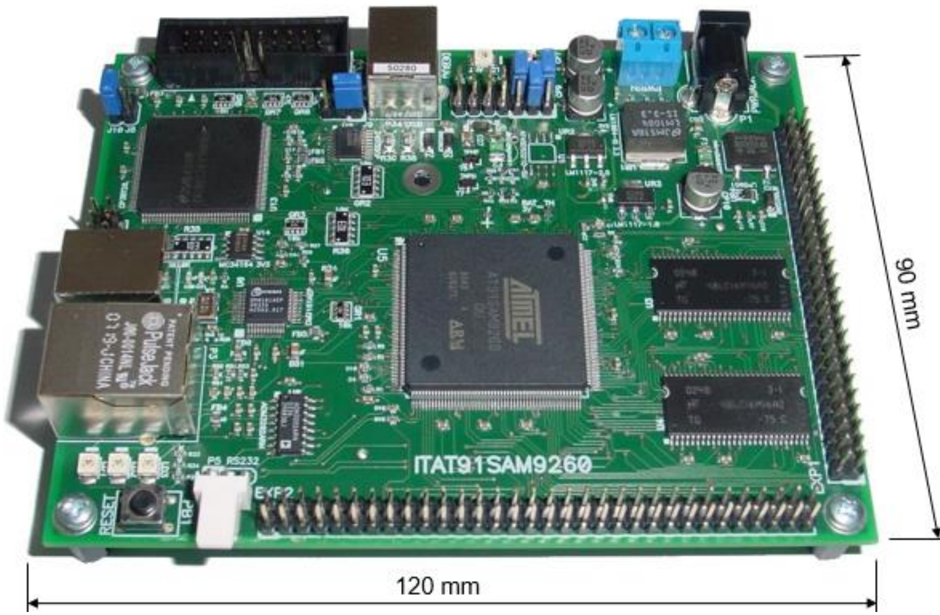
CPU – Central processing unit
ALU – Aritmetic logic unit

Lectures content:

- CA-1 Introduction
- CA-2 Development of computing machines
- CA-3 Basic computing principles
- CA-4 Assembly Instructions
- CA-5 Operands - representation of information
- CA-6 Structure and operation of CPU
- CA-7 CPU performance measurement
- CA-8 Memory
- CA-9 Memory hierarchy
- CA-10 Input and Output Systems

Laboratory work contents:

- FRI-SMS computer (somewhere in-between)
  - Microcontroller AT91SAM9260 of the ARM9 microcontroller family



- Learn the basics of computer architecture from a practical point o view
- Understand the inner workings of computers (ARM) by programing in assembly language
- In-depth view:
  - into computer operation
  - into program execution on computers

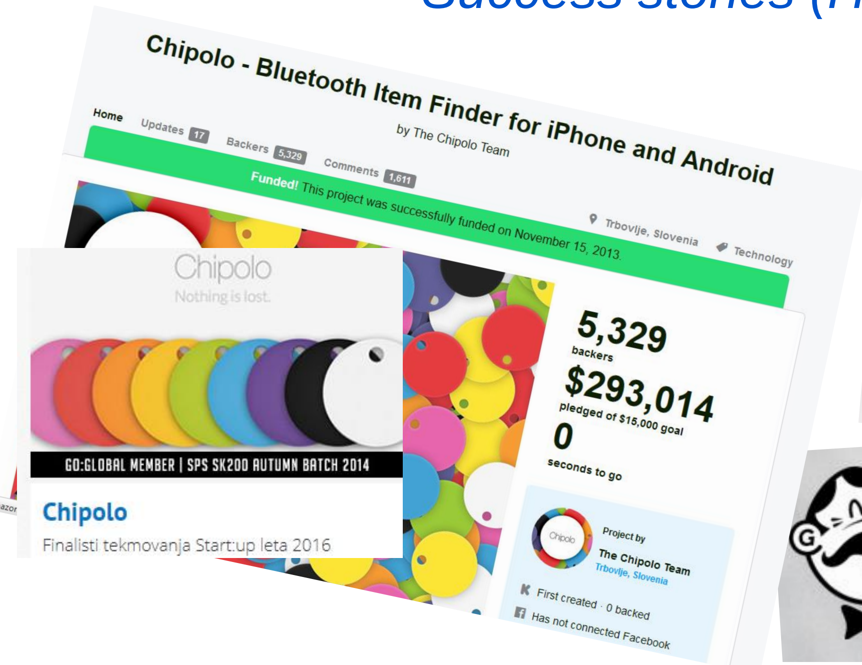
Further knowledge upgrade -> Computer Organization elective course and other related courses

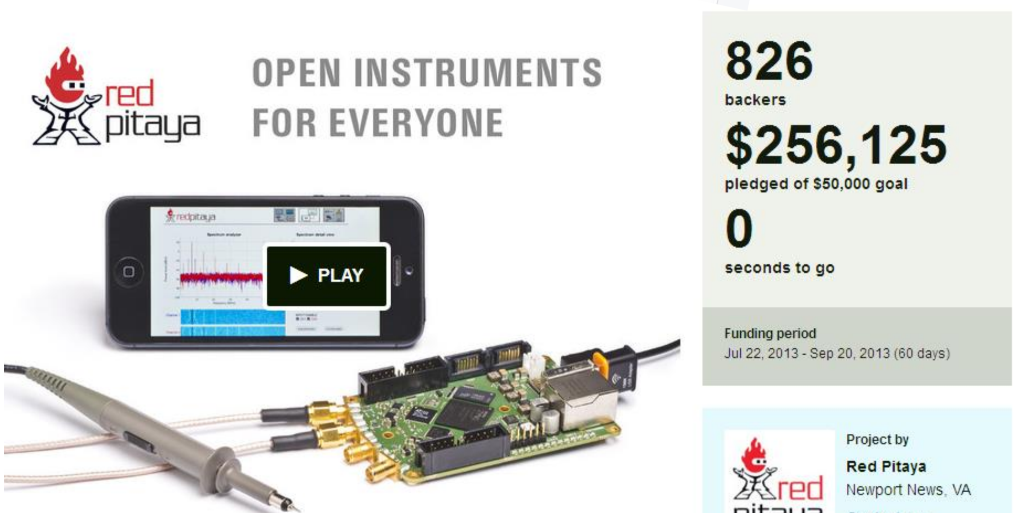
Why Computer Architecture is important ?

- 4 questions and
- 4 answers

1. Why Comp.Arch., HW ?

Success stories (HW+SW)





2. Why Comp.Arch., HW ?

- Because knowledge on computer architecture and operation leads to more efficient programming (programs).
  - Case: program code optimization regarding the operation of caches

```
/* Before */
for (j = 0; j < 100; j = j + 1)
        for (i = 0; i < 5000; i = i + 1)
                x[i][j] = 2 * x[i][j];
/* After */
for (i = 0; i < 5000; i = i + 1)
        for (j = 0; j < 100; j = j + 1)
                x[i][j] = 2 * x[i][j];
```

2. Why Comp.Arch., HW ?

- Because knowledge on computer architecture and operation leads to more efficient programming (programs).
  - Case: program code optimization regarding the parallel execution

us/Iteration	Iterations/sec
2.02500	493827.16
0.53300	1876172.61

```
double results[st];

for(int i = 0; i < st; ++i)
{
    results[i] = a[i] * b[i];
}
```

```
float results[st];

for(int i = 0; i < (st - 8); i += 8)
{
    __m256 i_a = _mm256_load_ps(&a[i]);
    __m256 i_b = _mm256_load_ps(&b[i]);
    __m256 i_c = _mm256_mul_ps(i_a, i_b);
    _mm256_store_ps(&results[i], i_c);
}

for(int i = (st - 8); i < st; ++i)
{
    results[i] = a[i] * b[i];
}
```

Code below is 4-times faster !

Reference: „Pomen poznavanja računalniške arhitekture“, avtor Miha Krajnc.

3. Why still assembly?

„who still knows this language?“

3. Why still assembly? One of the answers

[Dejan Črnila, Dewesoft]:

„because it's „polite" to learn the native language, culture…"

Past Meetup

Code optimization on modern processors [Dejan Črnila, Dewesoft]

„in our company developers „speak in assembly…"

*„by knowing the hardware and assembly we can speedup the code by **64x** !!!…"*

Code optimization is important but often overlooked part of a software project. In this talk we will dive deep and discuss when and why to optimize code, how to approach optimization and how to design data structures and algorithms for scalable performance.

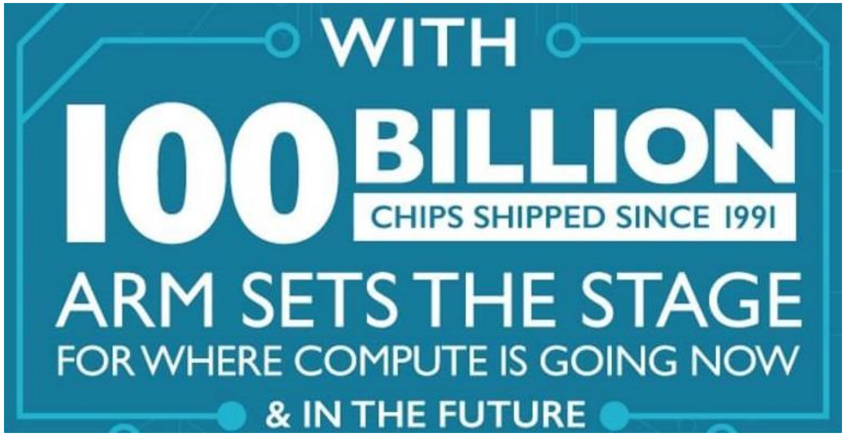
Dejan Črnila Dejan Črnila is lead software engineer at Dewesoft (https://www.dewesoft.com/careers) since 2001. He has designed and implemented core modules of Dewesoft application with particular focus on application performance to keep software in front of competition.

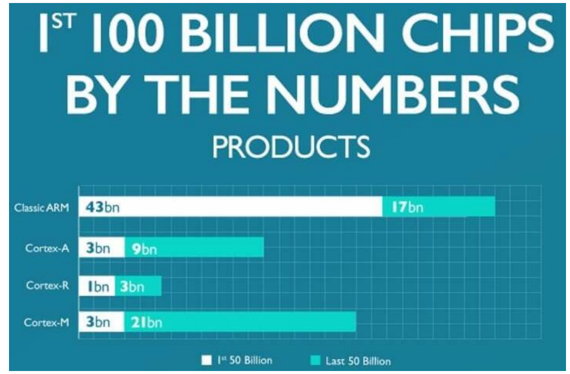
Space shuttle Atlantis launch monitoring with Dewesoft software

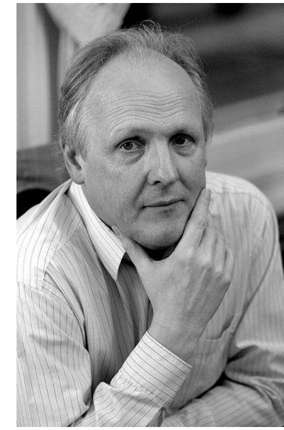
4. Why the ARM architecture?

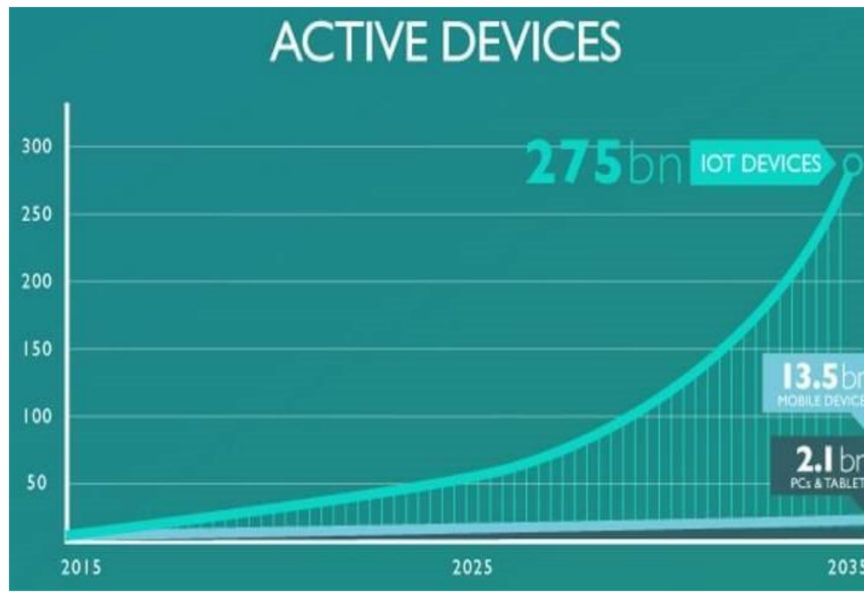
Because ??? …“

„Steve Furber on FRI“









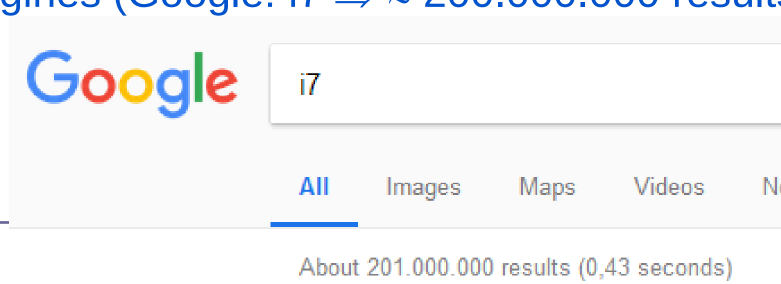
https://community.arm.com/processors/b/blog/posts/inside-the-numbers-100-billion-arm-based-chips-1345571105

Introduction

- Development and use of computers: IT revolution (third revolution in our civilization)

- Extremely rapid evolution over the past 25 years

- Applications that were until recently „impossible“, suddenly became common:
  - Computers in automobiles (autonomous drive)
  - Mobile telephony
  - DNA analysis (The Human Genome Project)
  - The World Wide Web
  - Search engines (Google: i7 $\Rightarrow \approx$ 200.000.000 results in few tenths of a second)



- Huge difference in computer implementation:
  - Supercomputers
  - Simple computers on a chip

Pros ?
Cons ?

- Smaller differences in structure
- With every computer, even the simplest, we can calculate everything that can be calculated (is calculable).

- Currently the 3rd most powerful computer in the world :
  - SunwayTaihuLight National Supercomputing Center in Wuxi, China
  - 10.649.600 processors (cores)
  - 1.310.720 GB main memory
  - Performance 93 014 TFLOPS
  - Power consumption 15 371 kW   (*Hydro PowerPlant Medvode 26 700 kW*)

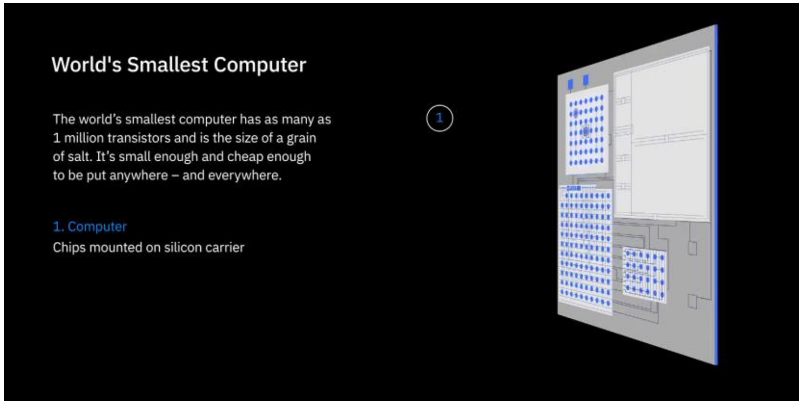
https://www.top500.org/lists/2019/06/

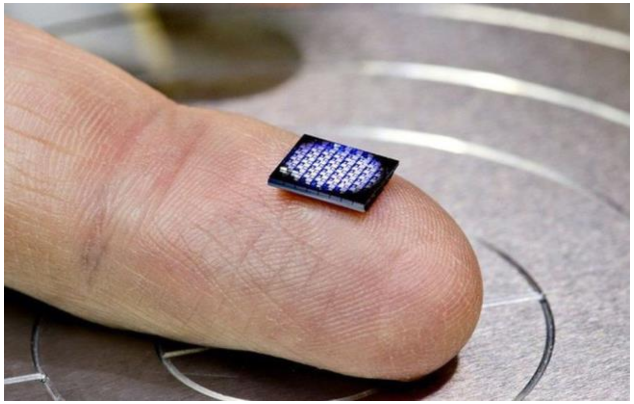
- Currently the most powerful computer in the world :
  - SUPERCOMPUTER FUGAKU in Kobe, Japan
  - 7 630 848 cores
  - Performance 537 212 TFLOPS
  - Power consumption 29 899 kW (*Hydro PowerPlant Medvode 26 700 kW*)

https://www.top500.org/lists/top500/2021/06/

https://www.r-ccs.riken.jp/en/fugaku/3d-models/

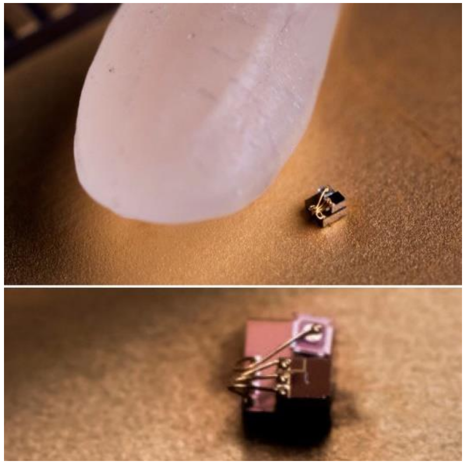
- Currently the smallest computer in the world (year 2018): ?





https://www.research.ibm.com/5-in-5/crypto-anchors-and-blockchain/

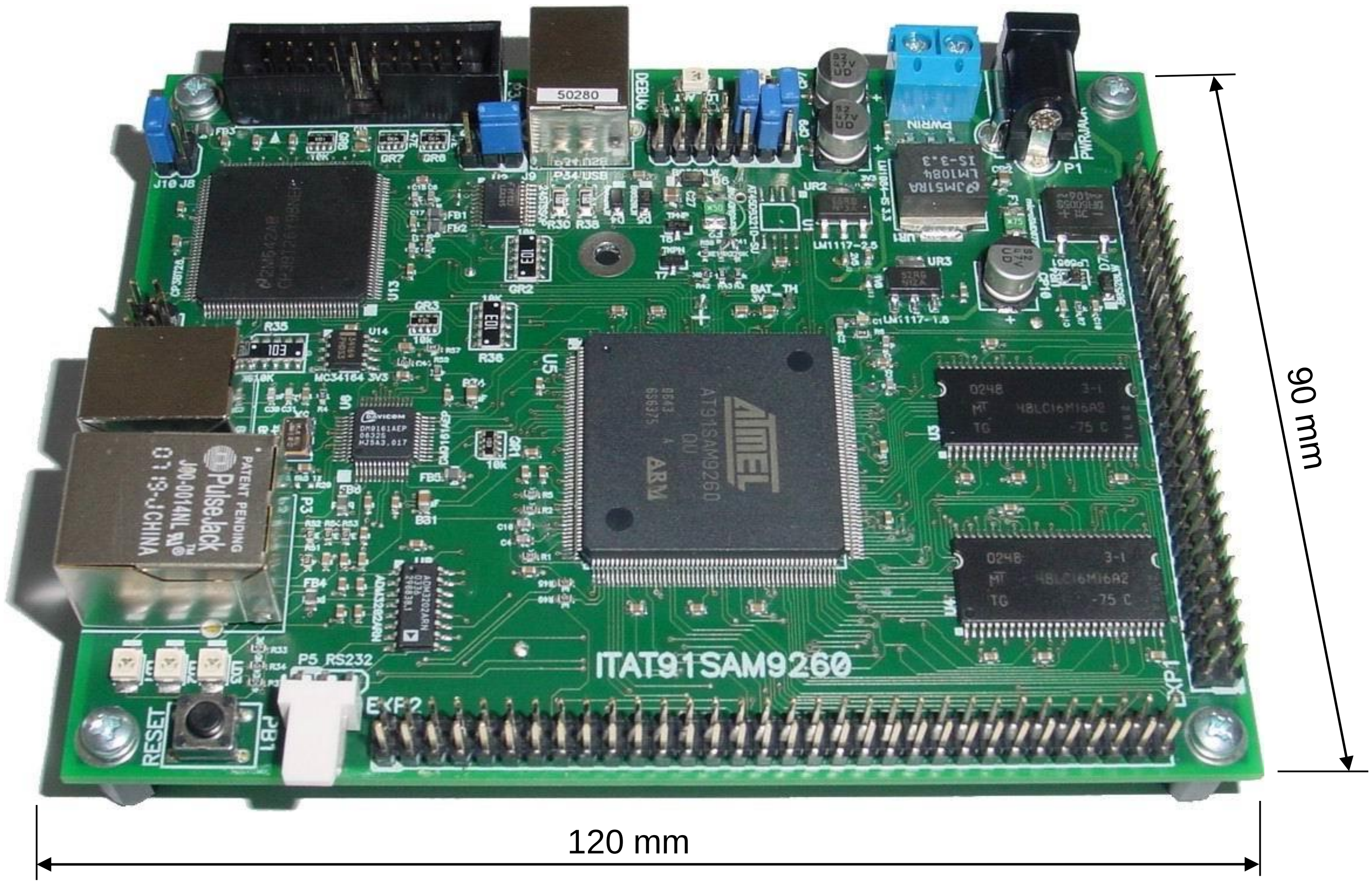
The university on Thursday said its engineers have produced a computer that's 0.3 mm x 0.3 mm -- it would be dwarfed by a grain of rice. While it drew comparisons to IBM's own 1mm x 1mm computer, Michigan's team said the creation is about more than just size.



- Pros: Low power consumption
- Cons: Low performance

https://news.umich.edu/u-m-researchers-create-worlds-smallest-computer/

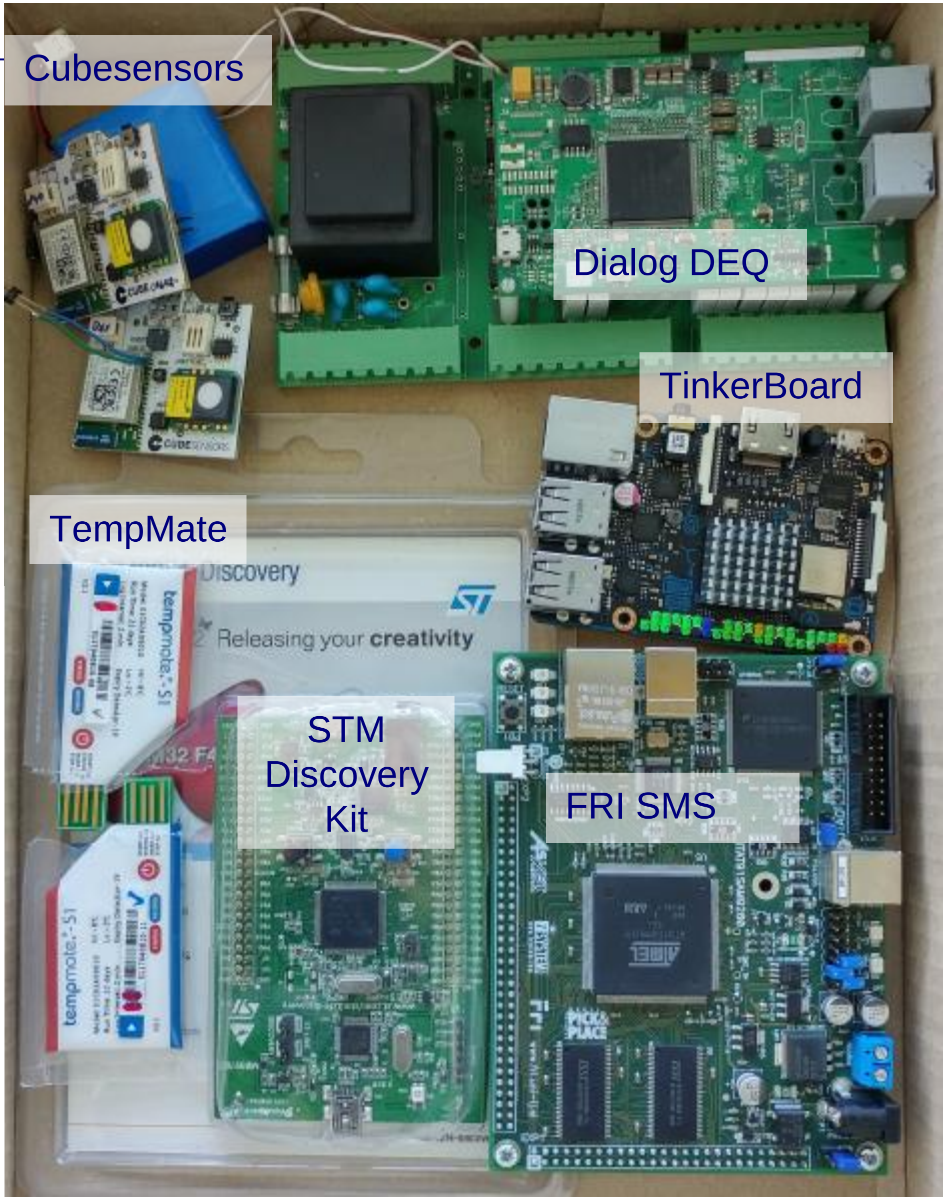
- FRI-SMS computer (somewhere in-between embedded and HPC)
  - Microcontroller AT91SAM9260 of the ARM9 microcontroller family



- Nowadays computers can be attributed into three functional categories:
  - Personal Computers (laptop, tablet, . . .)
  - Servers
    - There are large differences between servers in price and performance
      - A bit more powerful desktop computers on the low-end
      - Supercomputers with terabytes of main memory and petabytes of external storage on the high-end
  - Embedded computers
    - The most numerous group of computers
    - Microprocessors (or microcontrollers) in automobiles, mobile phones, gaming consoles, household appliances, audio and video equipment, …

Embedded computers (practical examples)

All systems (on right picture) are based on the ARM architecture.



Basic definitions

Def: Computer architecture is

- consideration of for the programmer visible computer properties independently of its logical and physical realization [Kodek]
  - *„… what programmers see on the assembly language level …“*

Def: Computer organization (also microarchitecture) :

- explores the logical structure and properties of the computer components and their interconnections [Kodek]
  - *„ … is the architecture of individual components …“*
  - *„ … is closer to the Hardware (HW) level …“*

One architecture can be realized with different types of organization and vice versa.

Operation of (digital) computers

- Computer architecture is also structure of computers as seen by the programmer in assembly code.
- Machine language consists of instructions which can be directly executed by the computer. Those instructions are also called machine code instructions.
- Machine instructions are native instructions build into computers. Computers from different manufacturers can have generally different machine code instructions.

Computer „understands“ own machine instructions only !!!

What is the computer doing ?

(How does it work ?)

Executing instructions !

- A digital computer is a machine for solving problems by executing instructions which were set by people.

- The sequence of instructions which determines how the machine performs a specific task is called a program.

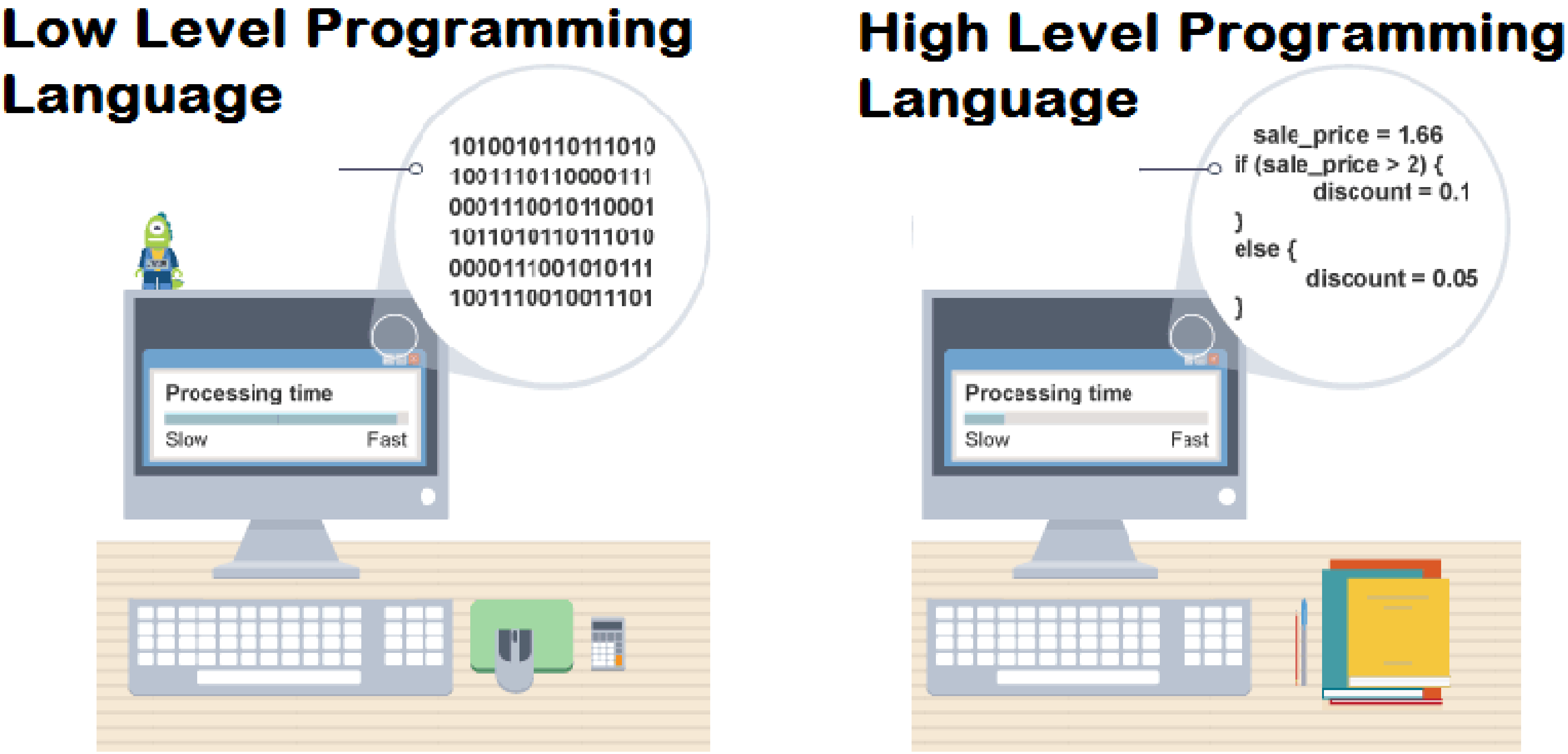
- The electronic circuit in the computer recognizes and directly executes only a limited set of machine code instructions into which every program has to be translated before the execution.

- Different processors can have different machine code instructions.

- Those basic instructions (machine instructions) are very simple, for example:
  - Addition of two numbers
  - Testing if a number is equal to zero
  - Copy data from one part of the memory to another.
- Any program that is written with some other instructions (e.g. instructions from Java, C++, VisualBasic,…) needs to be changed (translated) into those basic machine code instructions.

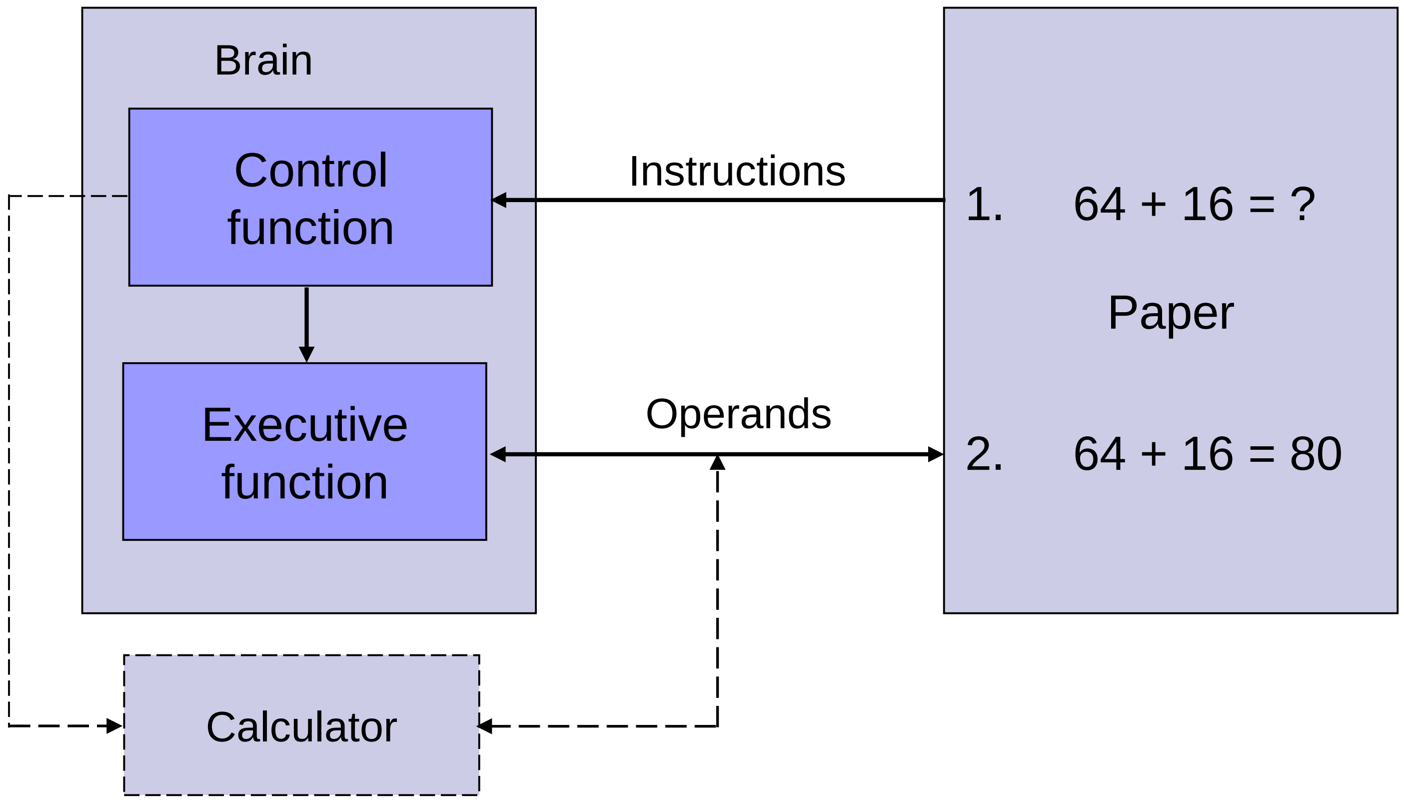
machine langauge <-> high-level languages ?

- Portability vs speed



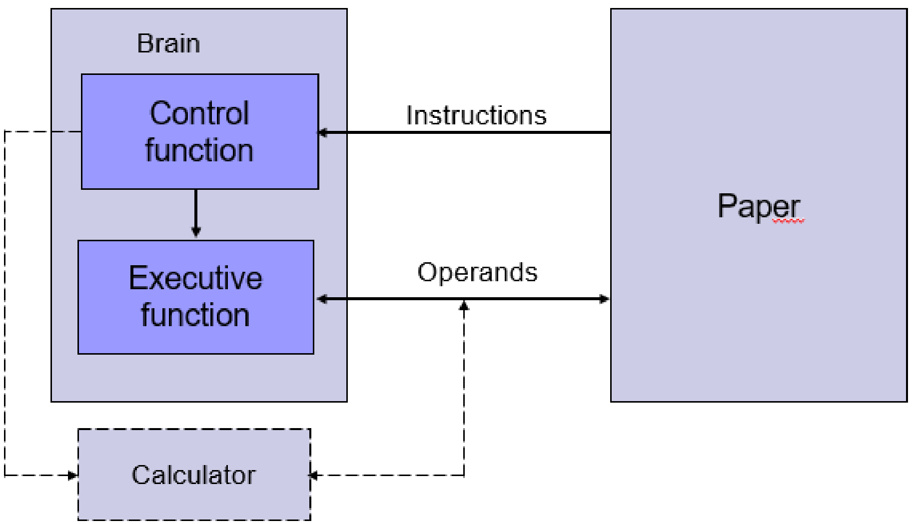
Connection between manual and machine model of computing

Manual computing

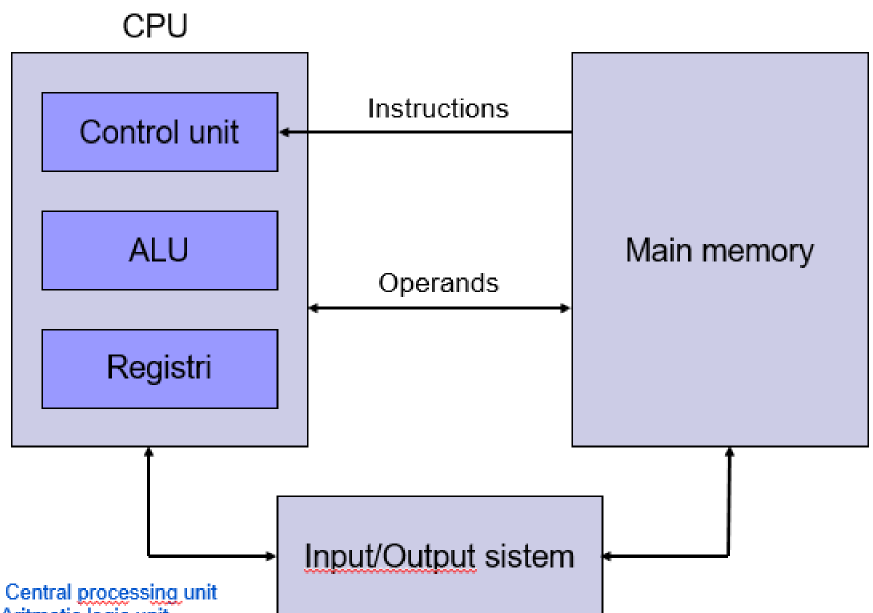


Comparison between models of computing

Manual computing

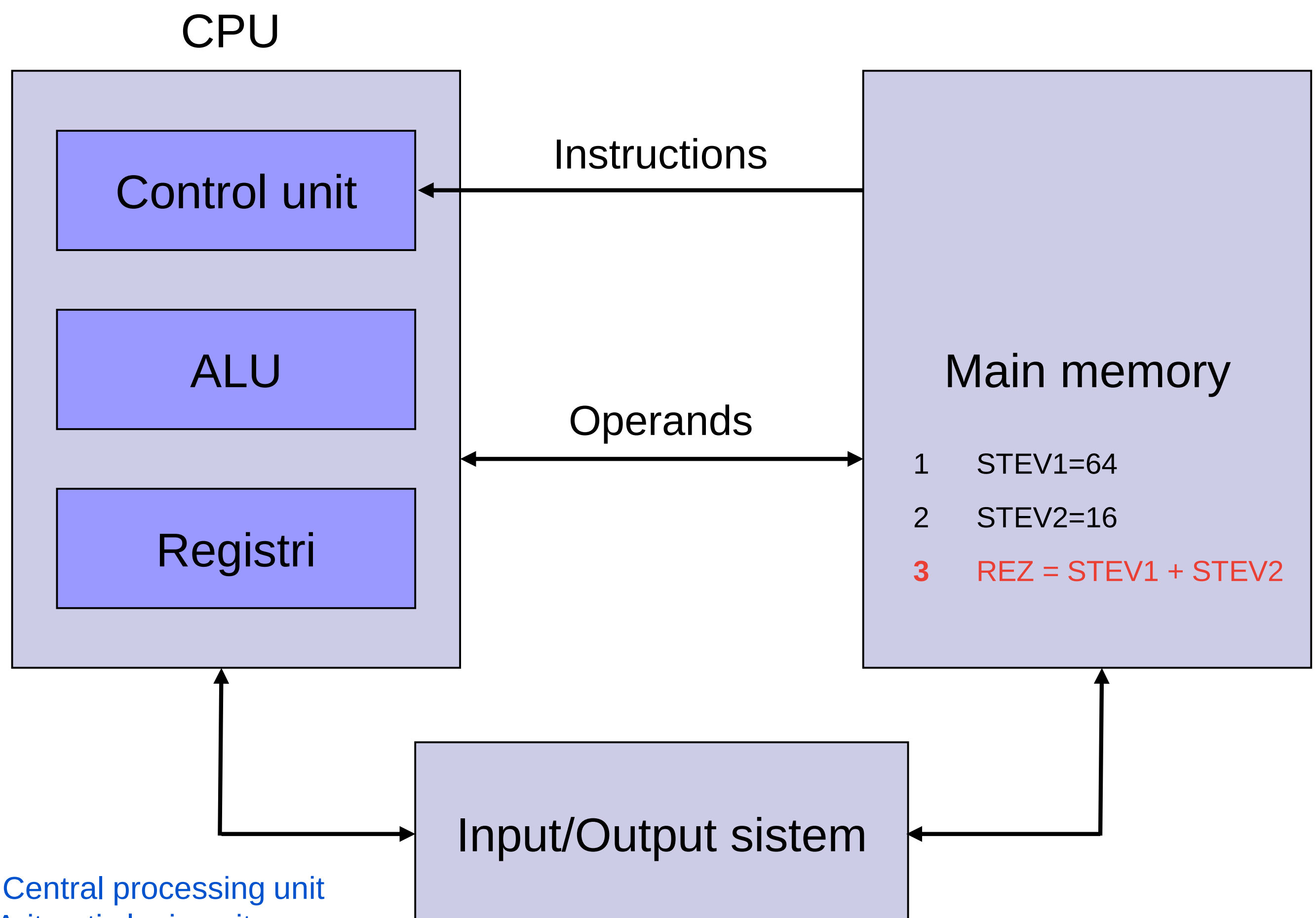


Structure of a typical computer



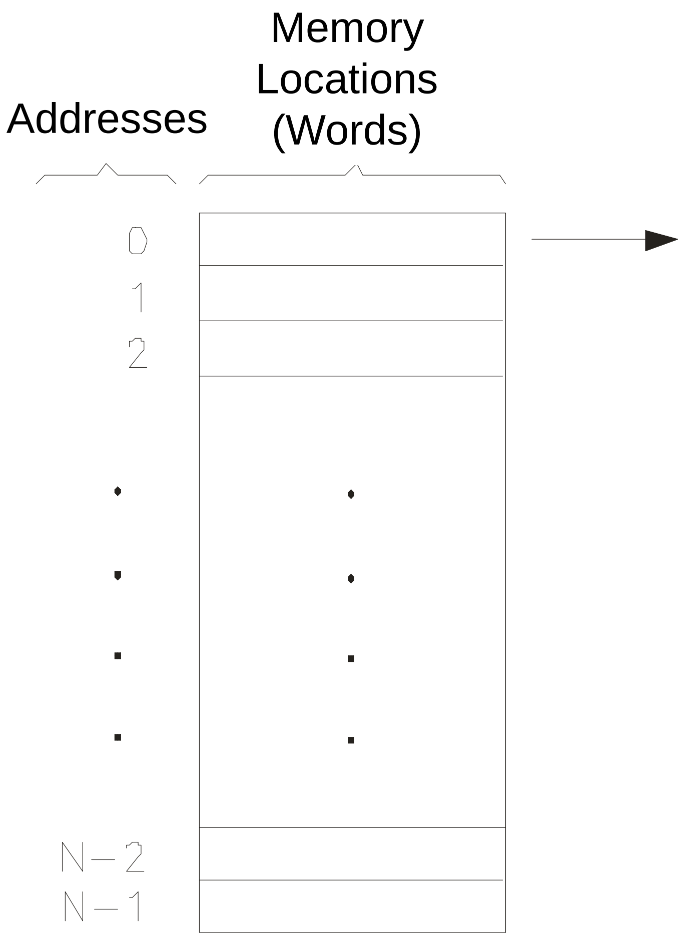
CPU – Central processing unit
ALU – Aritmetic logic unit

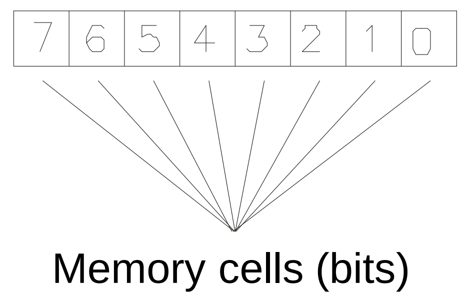
Structure of a typical computer

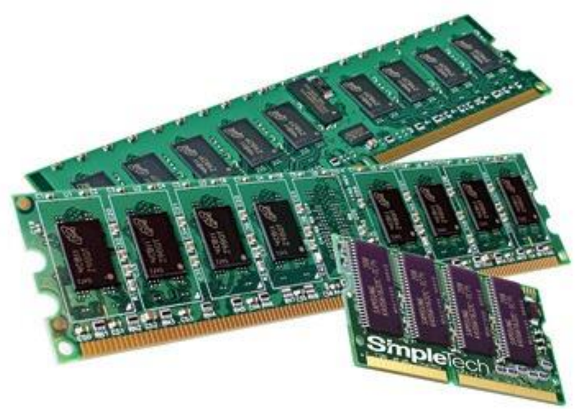


CPU – Central processing unit
ALU – Aritmetic logic unit

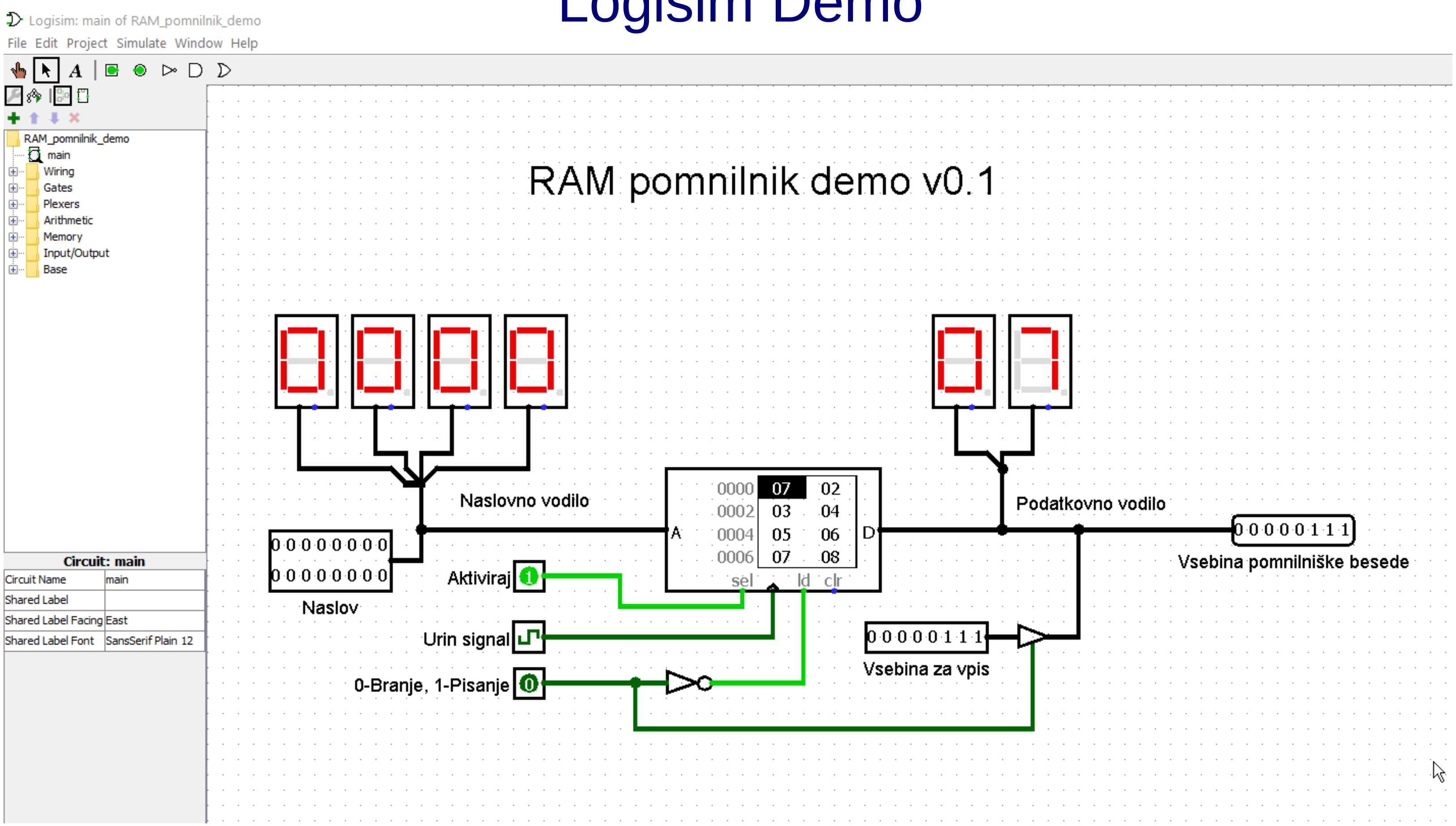
Memory


Addresses
Memory
Locations
(Words)
0
1
2
N−2
N−1


7
6
5
4
3
2
1
0
Memory cells (bits)

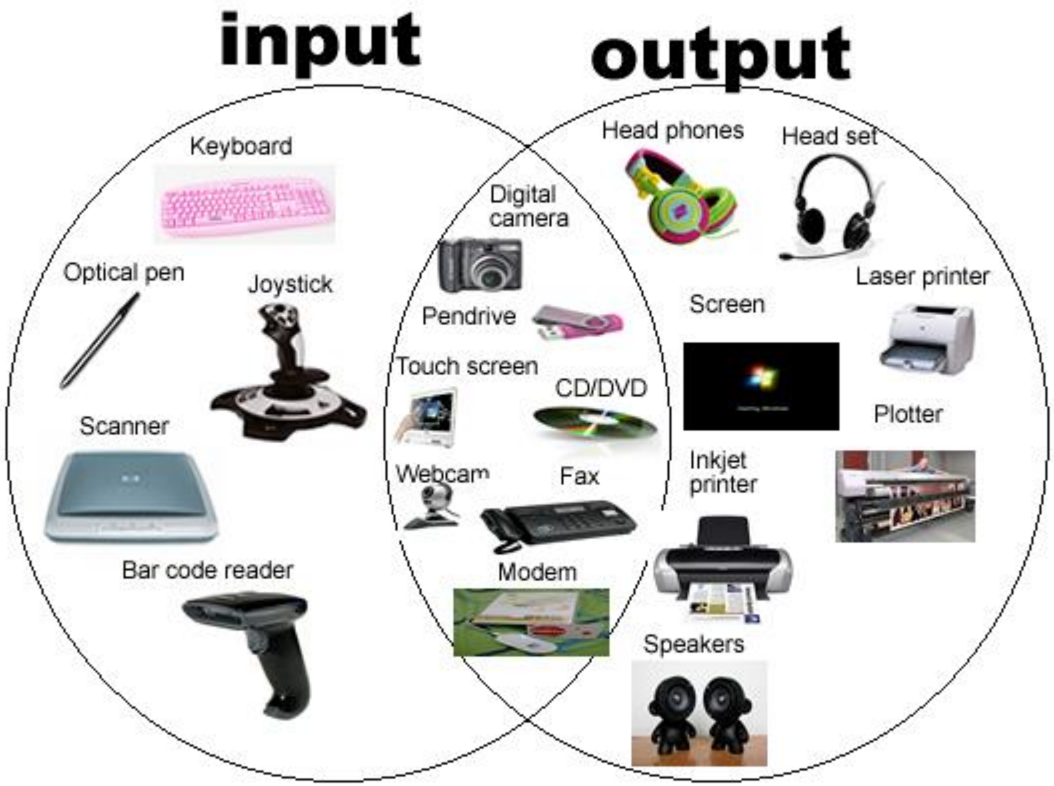

SimpleTech

Memory
Logisim Demo

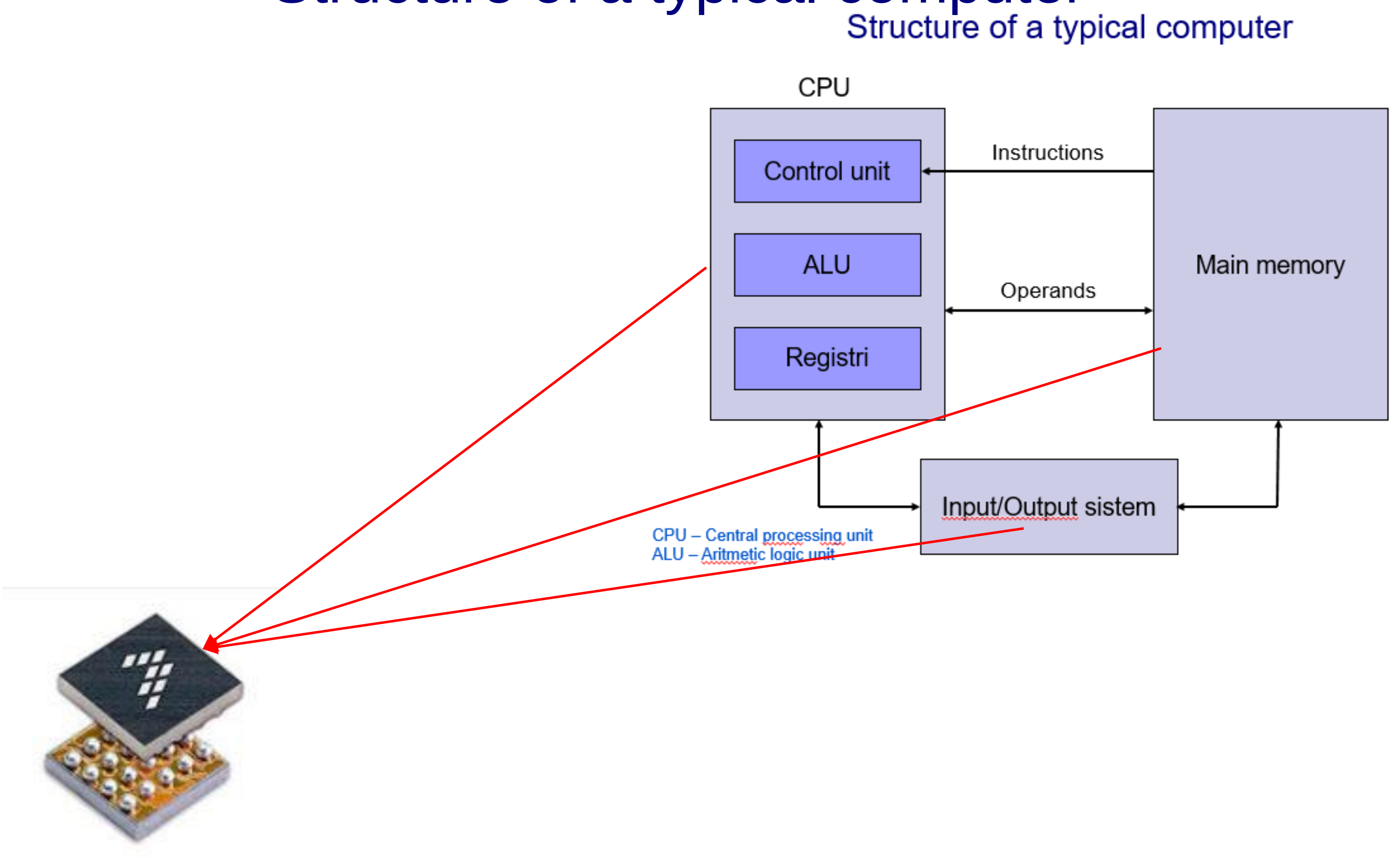


RAM_pomnilnik_demo.circ

Input-Output system (devices)

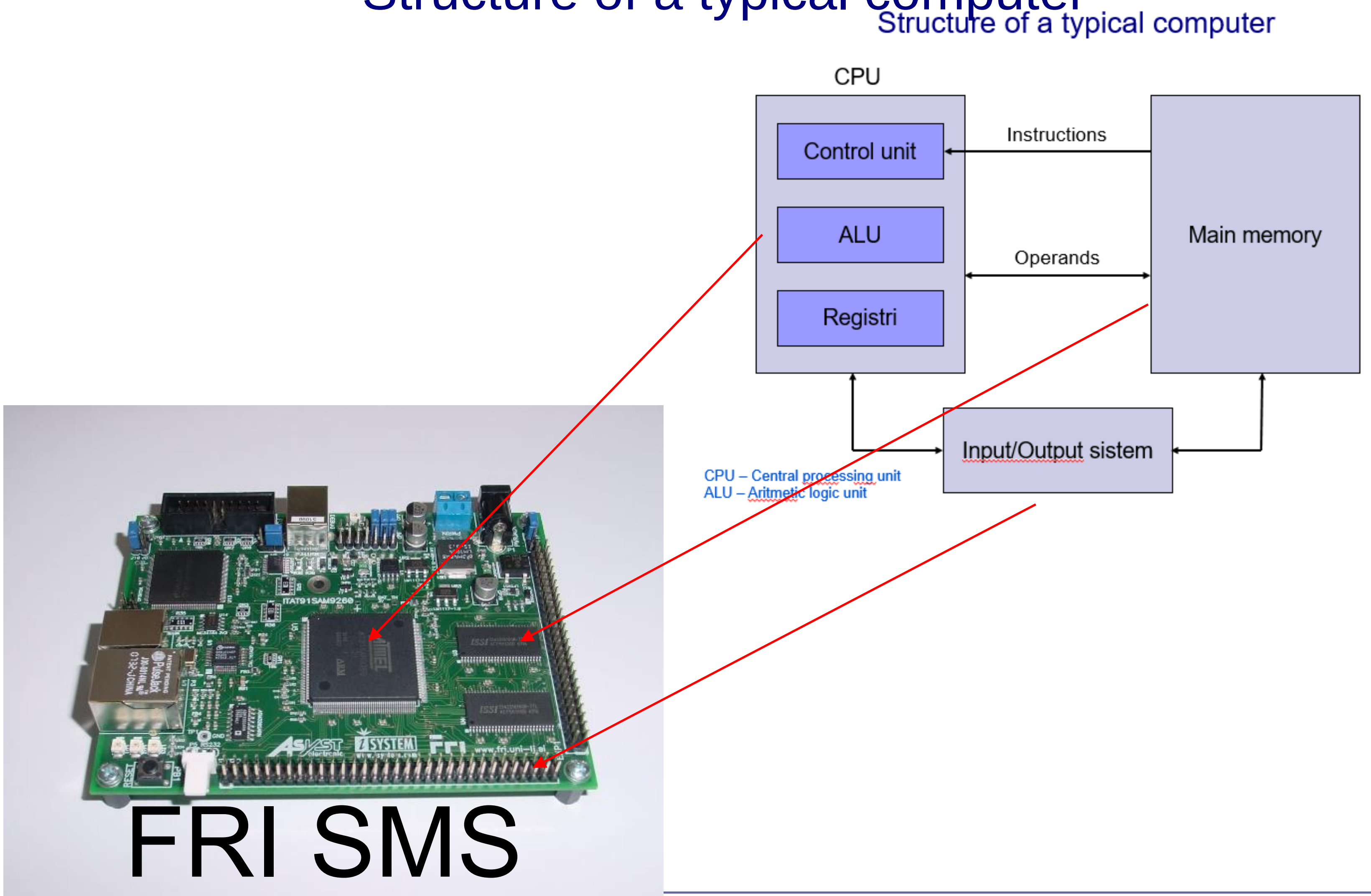


Structure of a typical computer

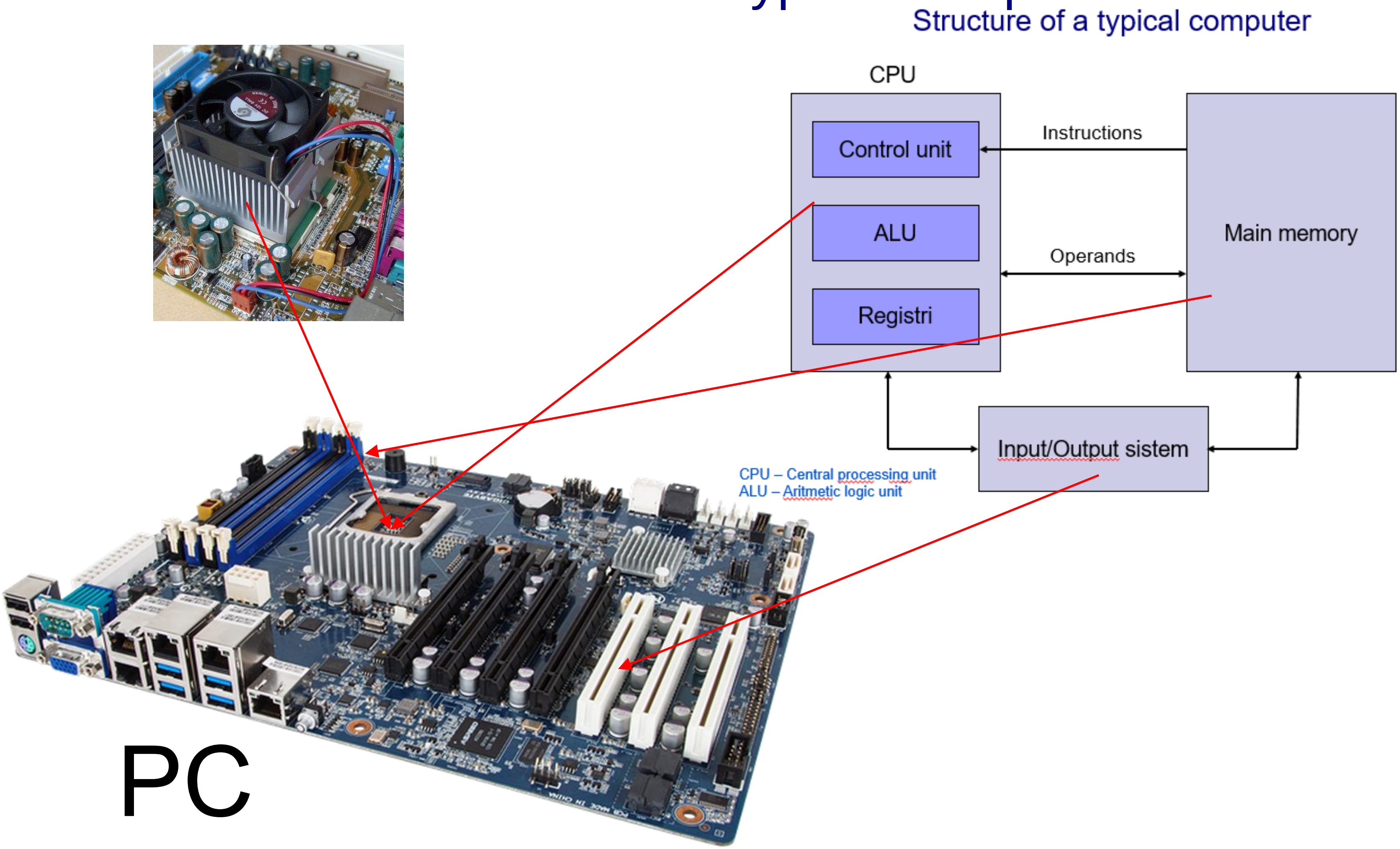


Microcontroller

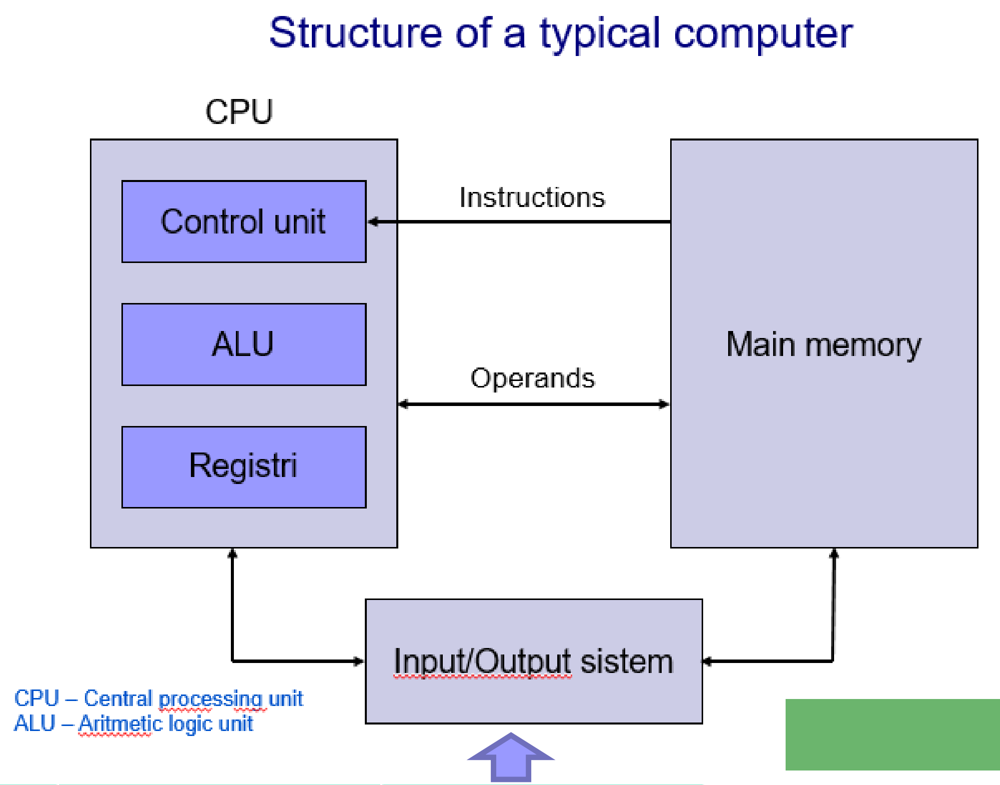
Structure of a typical computer



Structure of a typical computer



Structure of a typical computer and addition of two numbers



Zbirni jezik	Opis ukaza	Strojni jezik
ldr r1, stev1	R1 ← M[0x20]	0xE51F1014
ldr r2, stev2	R2 ← M[0x24]	0xE51F2014
add r3, r2, r1	R3 ← R1 + R2	0xE0823001
str r3, rez	M[0x28] ← R3	0xE50F3018

Python	
1	STEV1=64
2	STEV2=16
3	REZ = STEV1 + STEV2

Comparisons:

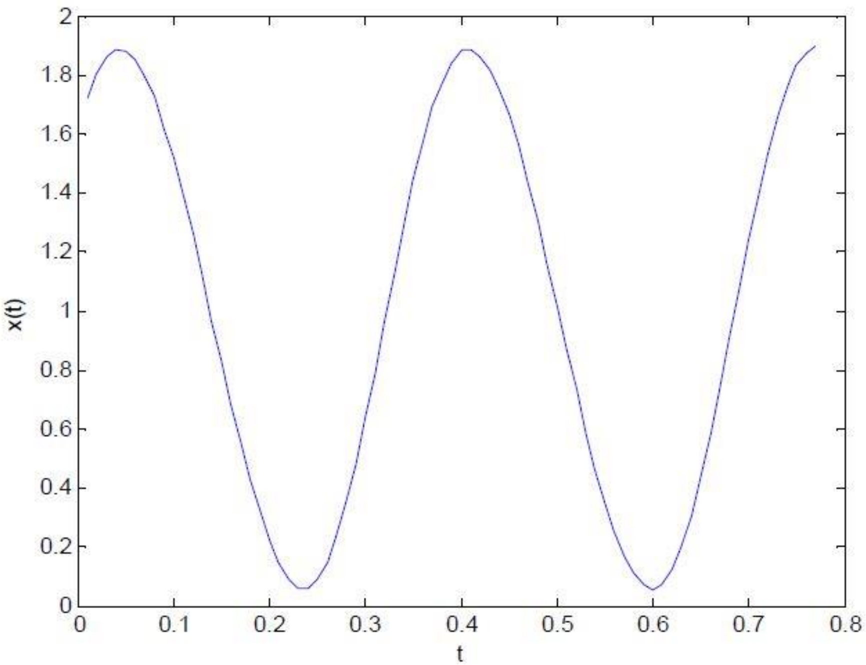
Analog – digital

Continuous – discrete

Analog –
- continuous representation

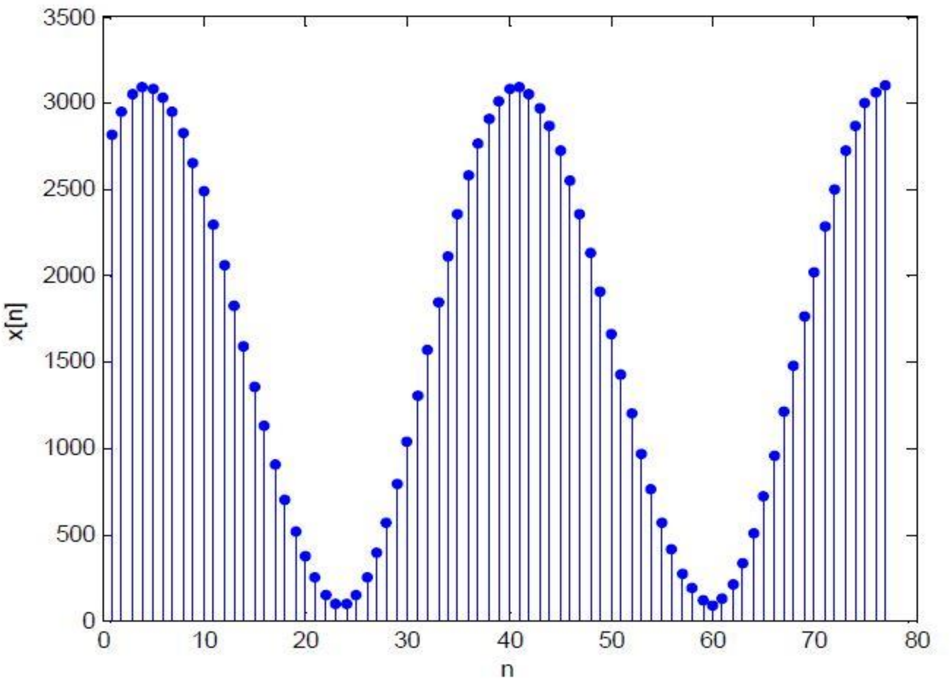
Digital –
- discrete representation

Analog –
- continuous representation

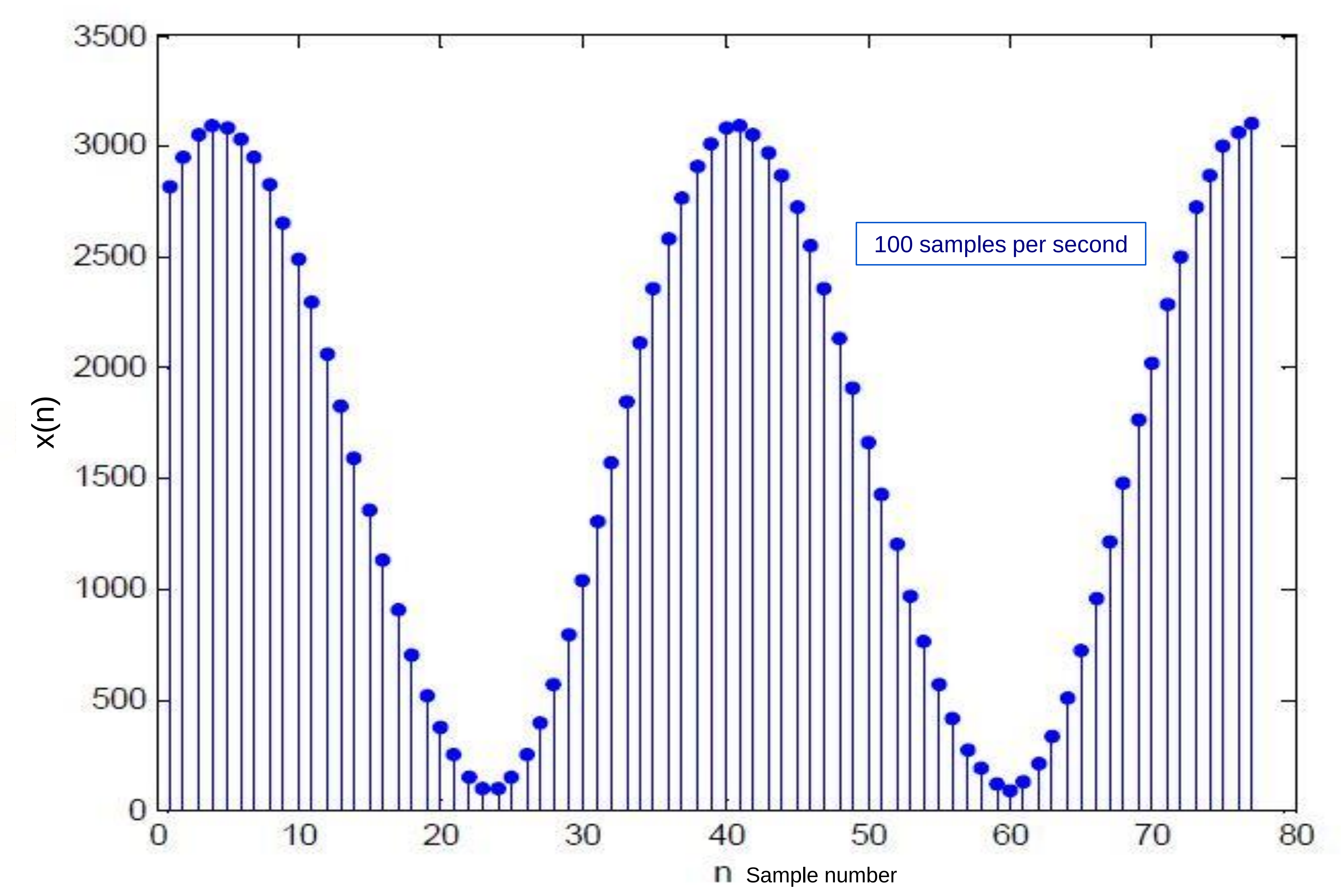


Time in seconds

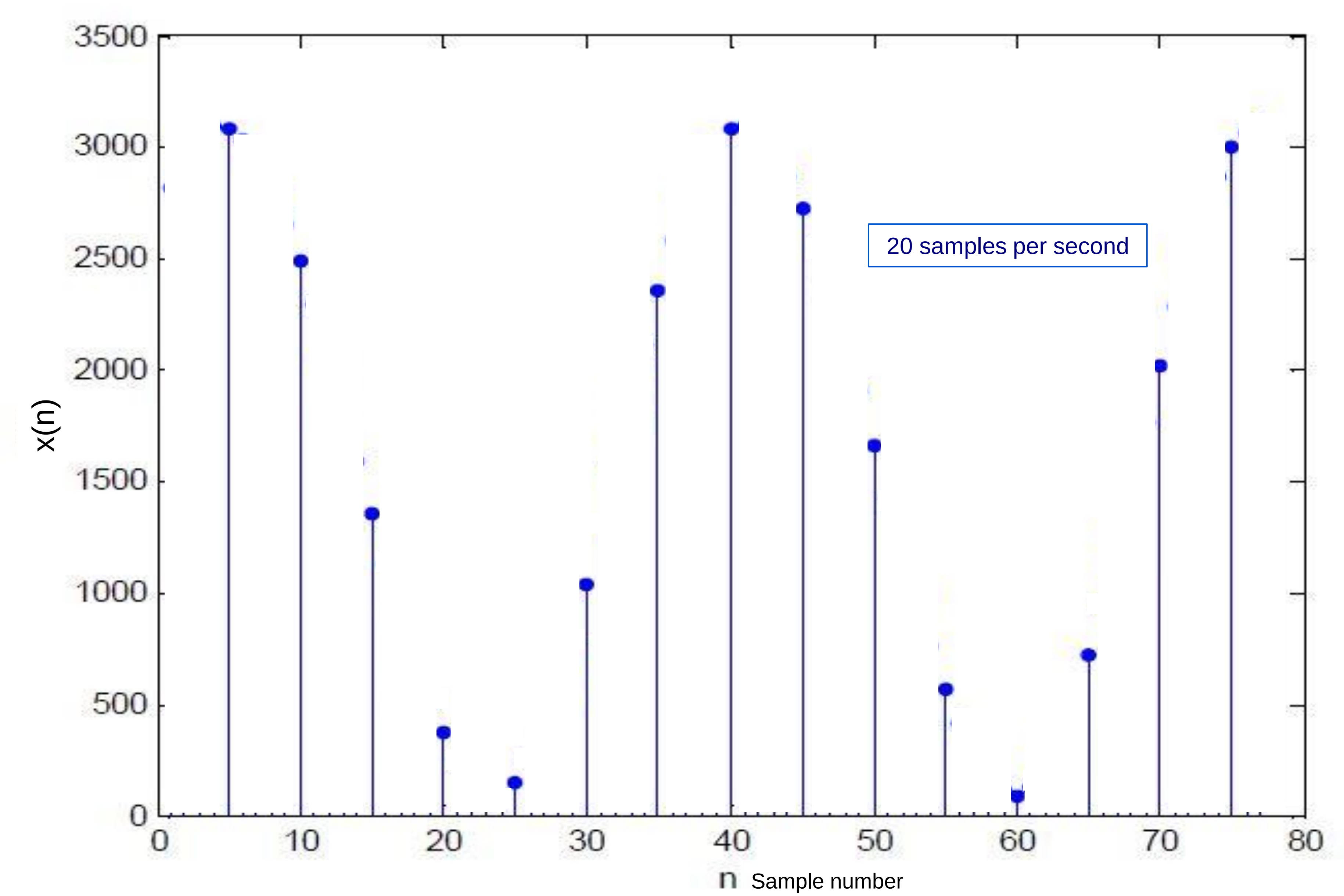
Digital –
- discrete representation



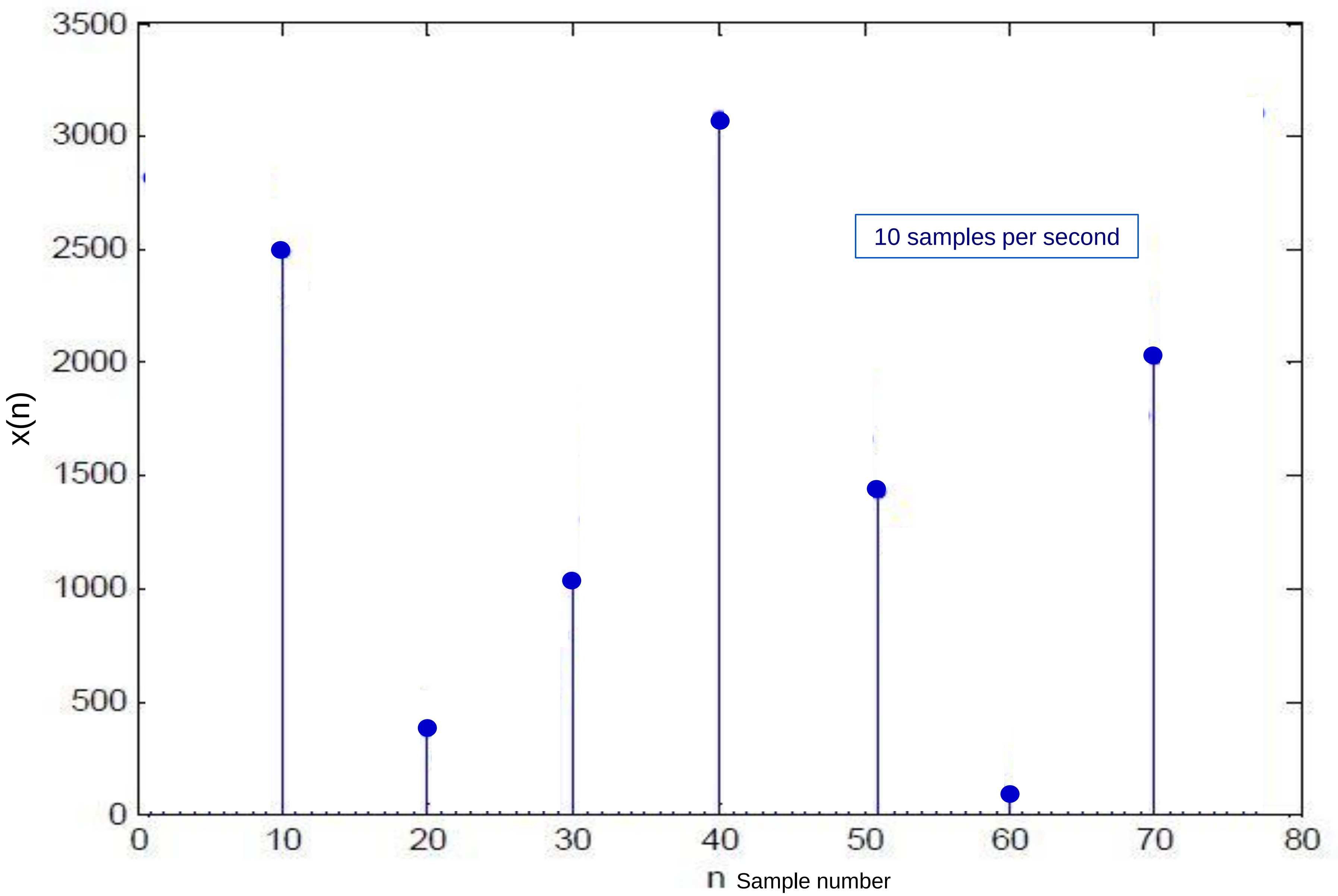
Sample number
100 samples per second



100 samples per second
x(n)
n Sample number
0
500
1000
1500
2000
2500
3000
3500
0
10
20
30
40
50
60
70
80



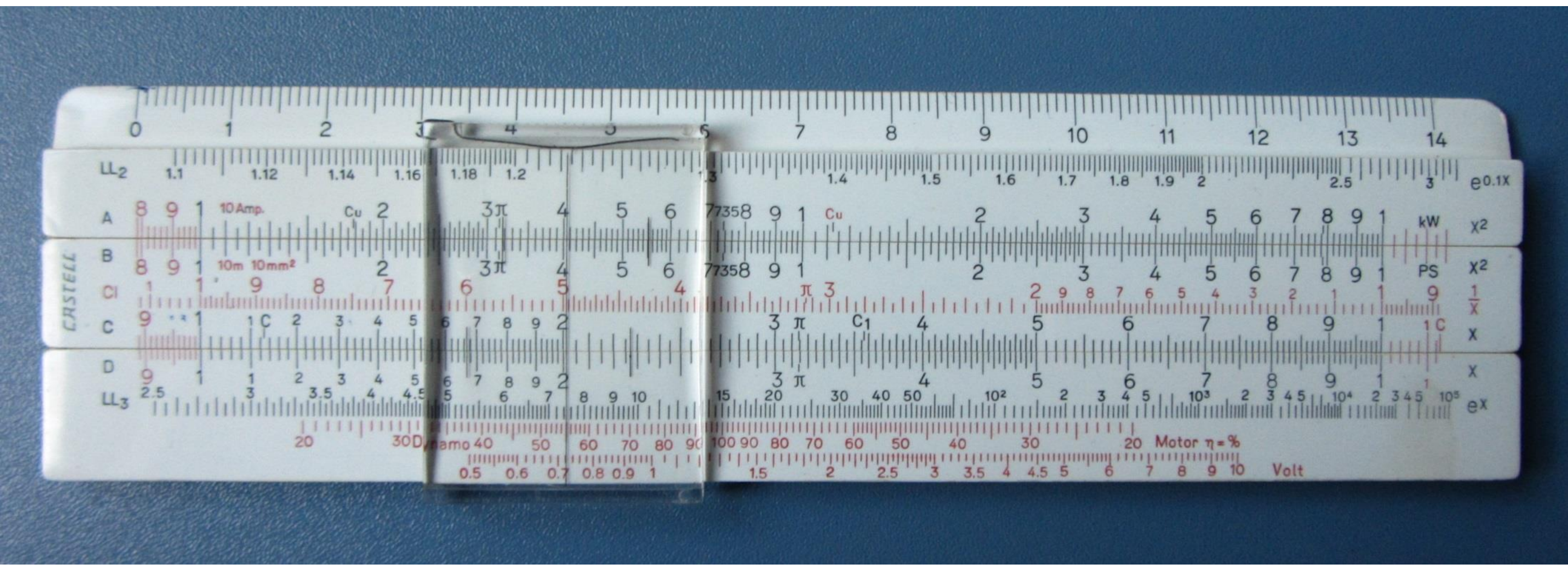
3500
3000
2500
2000
1500
1000
500
0
x(n)
20 samples per second
0
10
20
30
40
50
60
70
80
n Sample number


3500
3000
2500
2000
1500
1000
500
0
x(n)
0
10
20
30
40
50
60
70
80
n Sample number
10 samples per second

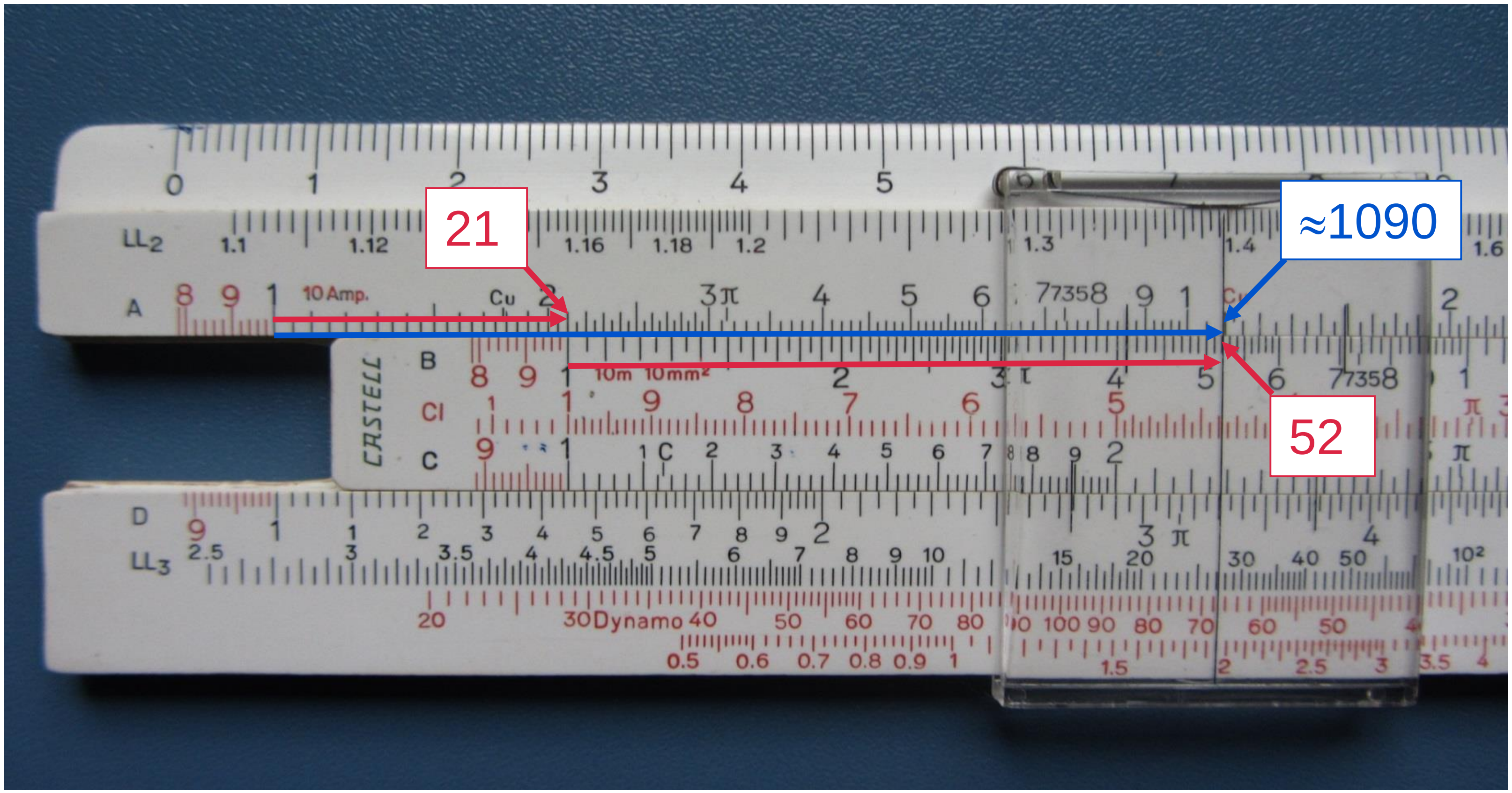
Analog computing – continuous presentation of numbers

Digital computing – discrete presentation of numbers

- Analog computing is carried out by representing numbers with some other physical quantity:
  - With distance ⇒ Logarithmic calculator
  - Idea: $\log_{10}(a \cdot b) = \log_{10} a + \log_{10} b$



- Example of multiplication of 21 x 52 with the logarithmic calculator:



21 x 52 ≈ 1090
Measured result

21 x 52 = 1092
Exact result

- using continous Voltage ⇒ Analog computer

- Discrete computing with beads
- With digits from 0 to 9

Digital computing

- With digits 0 and 1

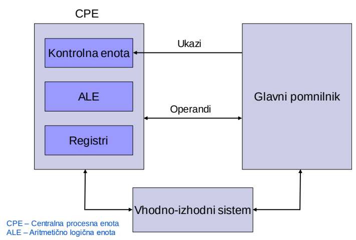
⇩

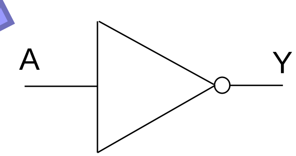
- Binary numeral system:
  - base number is 2
  - digits 0 and 1
- binary digit = bit
- Bit = one of the two digits (0 or 1) of the binary numeral system
- Digital computer is built on top of binary numeral system

8 important ideas in computer architecture (and broader) [Patt]

- Moore's law
  - Number of transistors in integrated circuits double every 18-24 months
- Abstraction as simplification
  - Design of hardware and software, programing languages, subprograms, …
- Speed up common procedures
  - It's most profitable to speed up the most common used procedures
- More performance with parallelism
  - Considering the current technology evolution: it's the only way
- Performance with pipelines
  - Effective, transparent way to speed up the CPU
- Performance with speculations
  - „Better work according to some speculation then just do nothing - wait"
- Memory hierarchy
  - Compromise between memory speed and cost
- Reliability with redundancy
  - Cost of the backup system may be lower then the cost of failure

Theoretical model <-> Practical realization





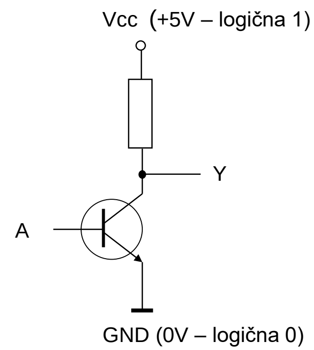
Simbol

mathematical (logical) view: logic gate

logical level 0,1

Mathematical ideal

electrical realization **:** electrical circuit



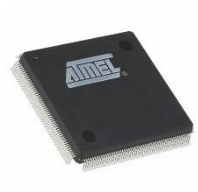
voltage levels ≈0V, ≈3.3 (5) V

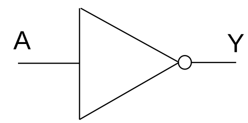
Electrical realization

Cons :

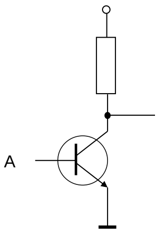
- *analogue voltagesi*
- *time-delays*
- *noises*

Physical structure of the computer





Simbol



Information (instructions and operands) are in computers represented in binary system, with the help of electrical signals

- Two states (symbols) 0 and 1 are represented with two voltage levels.
  - **State 0** can be represented with low voltage (around 0V)
  - **State 1** can be represented with high voltage (upto +5V)
- Simple realization with a switch - example:
  - **State 0** – switch open – low voltage
  - **State 1** – switch closed – high voltage

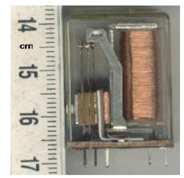
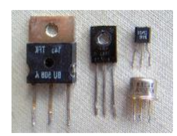
- One switch can be in two states, state 0 or 1.
- Such switch can memorize 1 bit of information.
- Basic memory cell can be imagined as such a switch. It shows its state and we can store 1 bit (0 or 1) of information into it.
- If we want to store more then only 1 bit of information, we need more cells.

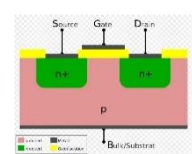
Realization of switches in the development of digital computers – technology evolution

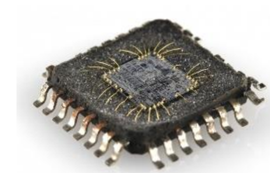
Prefixes for units of measurement

Abbreviation	Name	Value	Exponent (scientific notation)
p	pico	0,000 000 000 001	10^{-12}
n	nano	0,000 000 001	10^{-9}
μ	micro	0,000 001	10^{-6}
m	milli	0,001	10^{-3}
K	kilo	1 000	10^{3}
M	mega	1 000 000	10^{6}
G	giga	1 000 000 000	10^{9}
T	tera	1 000 000 000 000	10^{12}

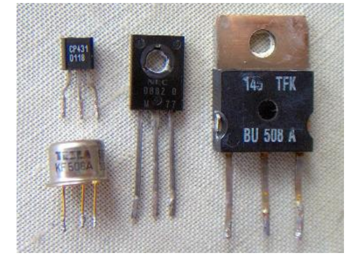
Realization of switches as the basic building block - summary:

- Electormechanical switch
  - 1939: Relay,  switching time 1-10ms
- Electrical switch
  - 1945-1955: Vacuum tube,  switching time ~ 5μs
  - 1955: Transistors → ,  switching time ~10ns
    - 1958: Integrated circuit - chip, switching time 2-10ns
    - 1980: VLSI integrated circuit switching time < 0.1ns
      - Very Large Scale Integration

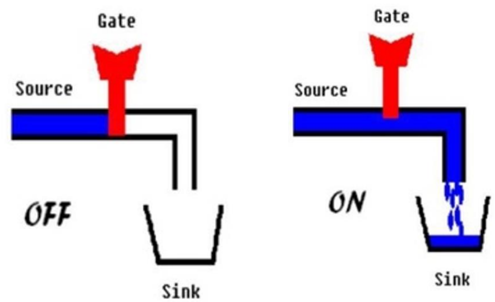


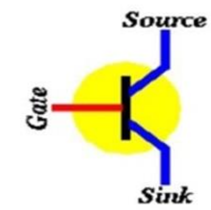


Transistor can be used as :

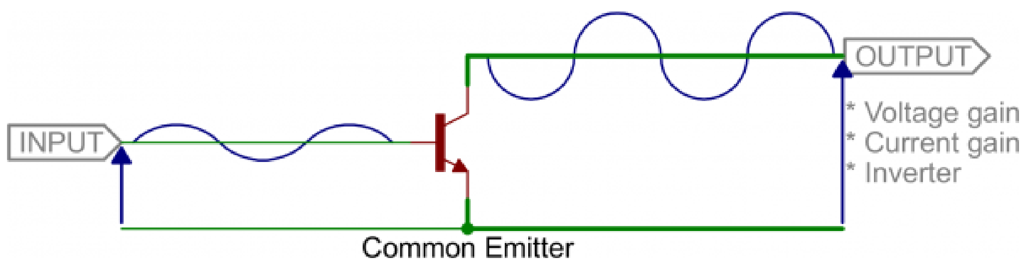


- switch
  - In digital circuits

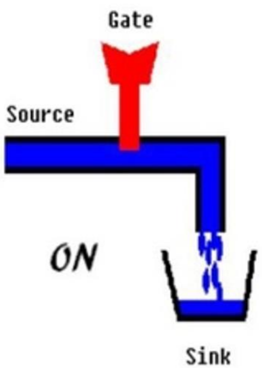


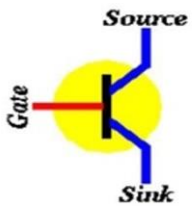


- Signal amplifier
  - Electronic circuits (amplifiers)



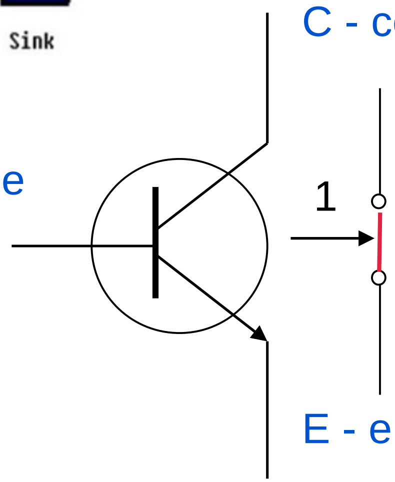
- Transistor as switch - ON



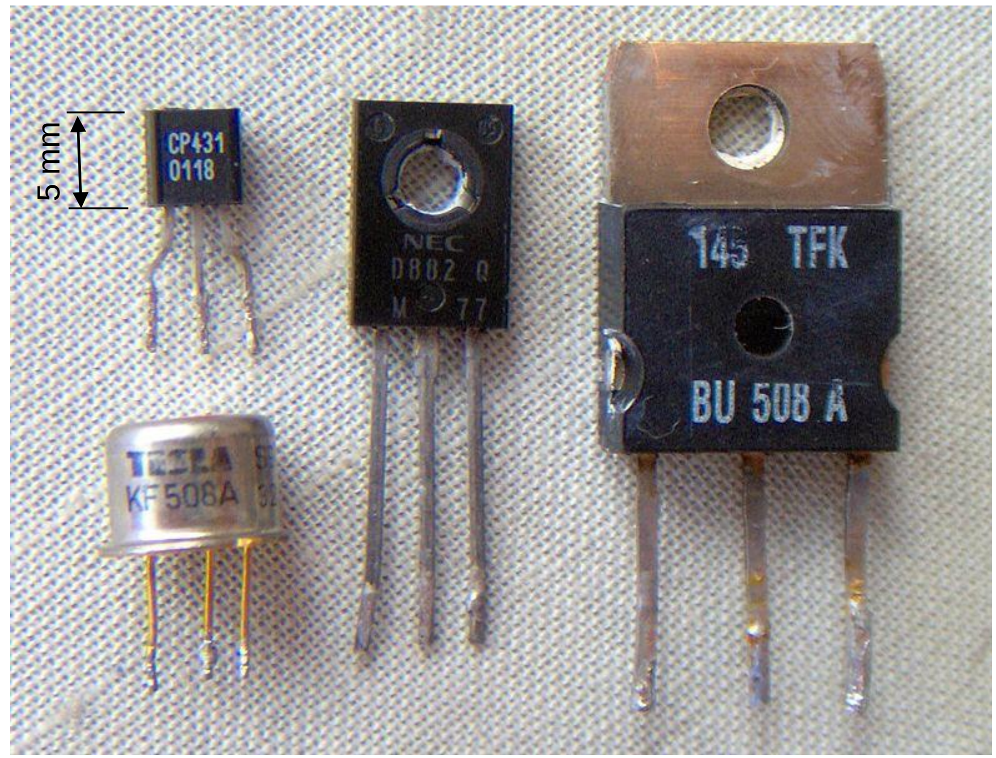


C - collector

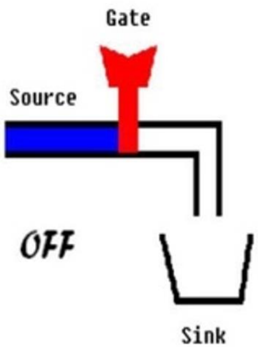
B - base

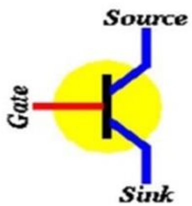


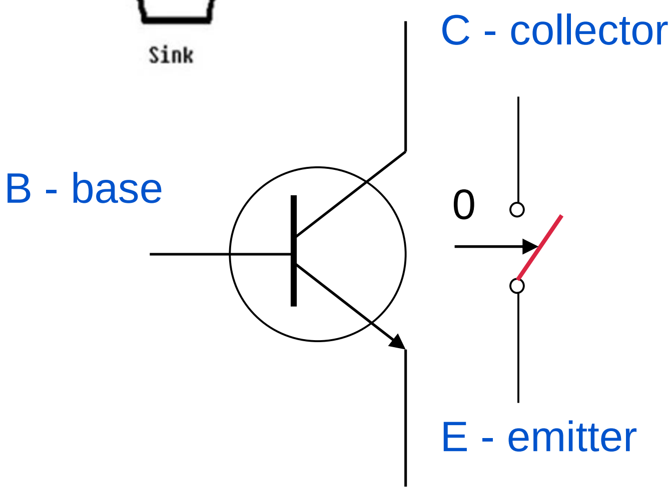
E - emitter

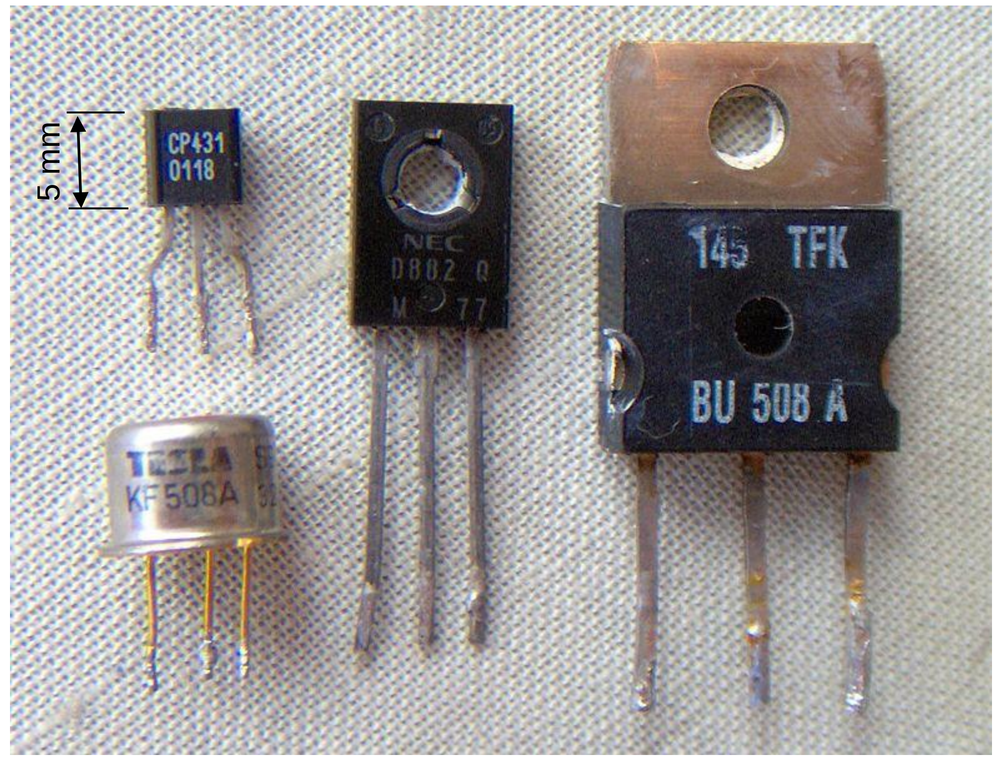


- Transistor as switch - OFF

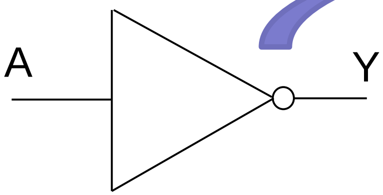








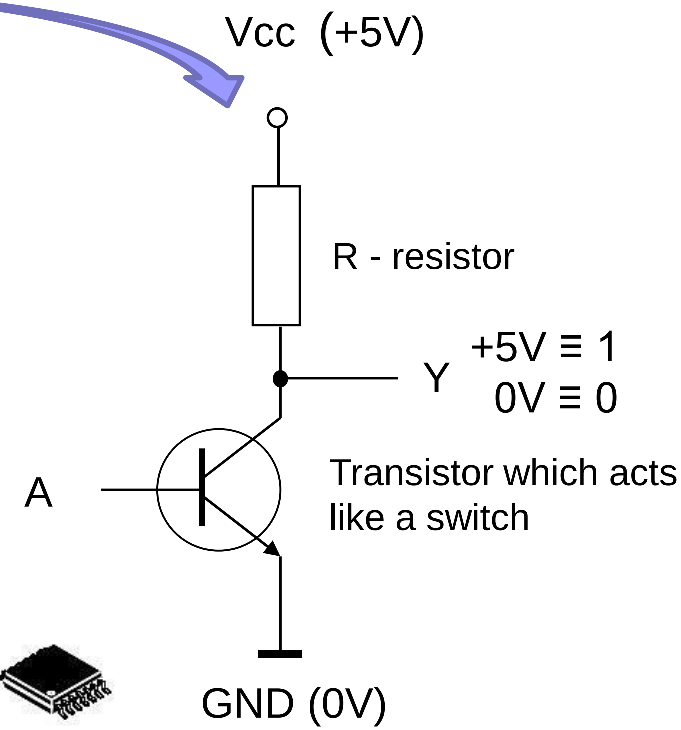
Realization of the logical function NEGATION (NOT)



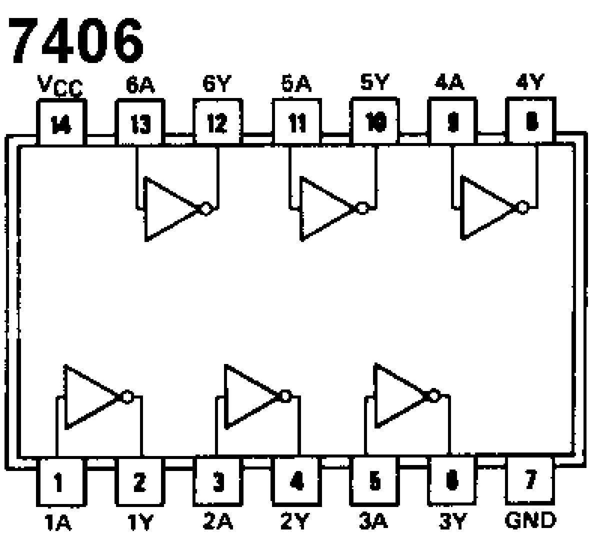
Symbol

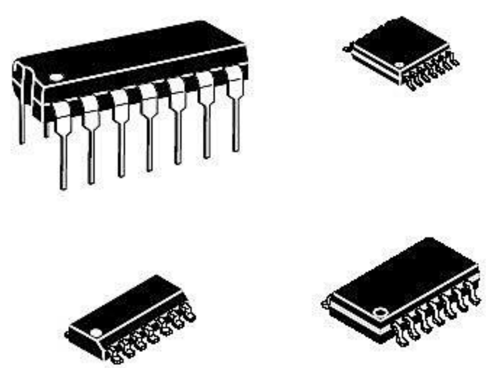
A	Y
0	1
1	0

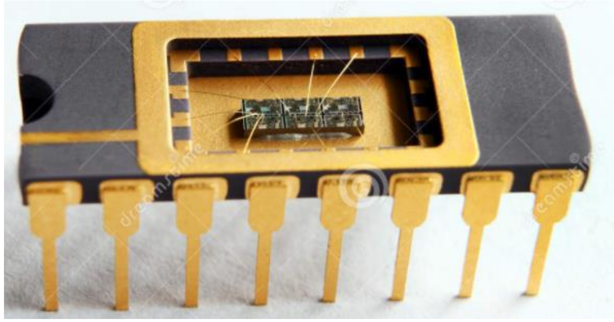
Truth table



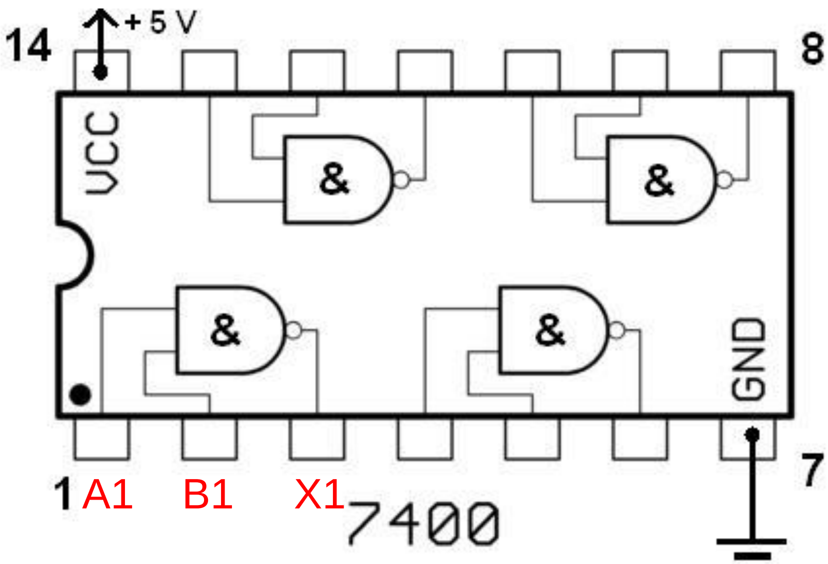
IC (Integrated Circuit) with 6 negators

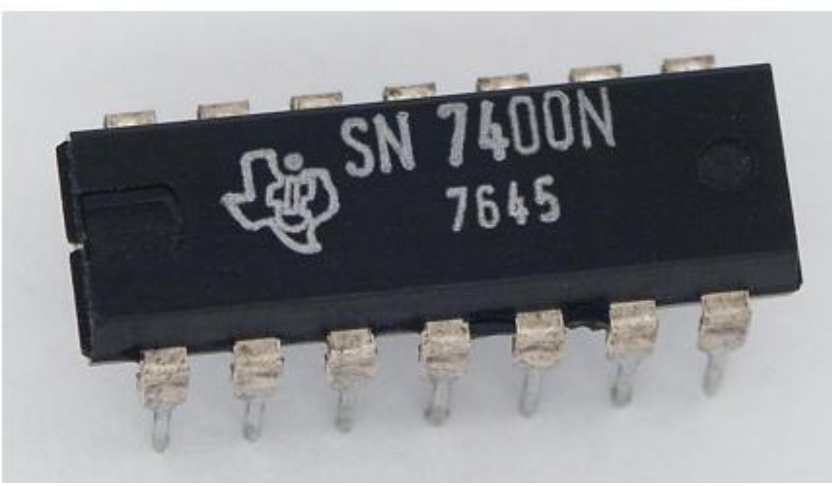


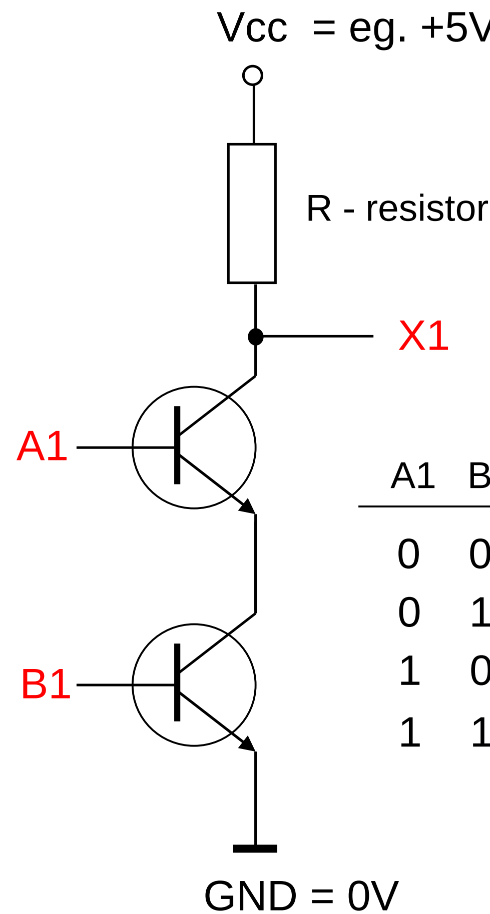




Realization of the logical function NAND (Negated conjunction)

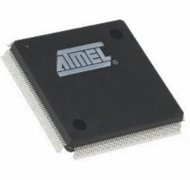


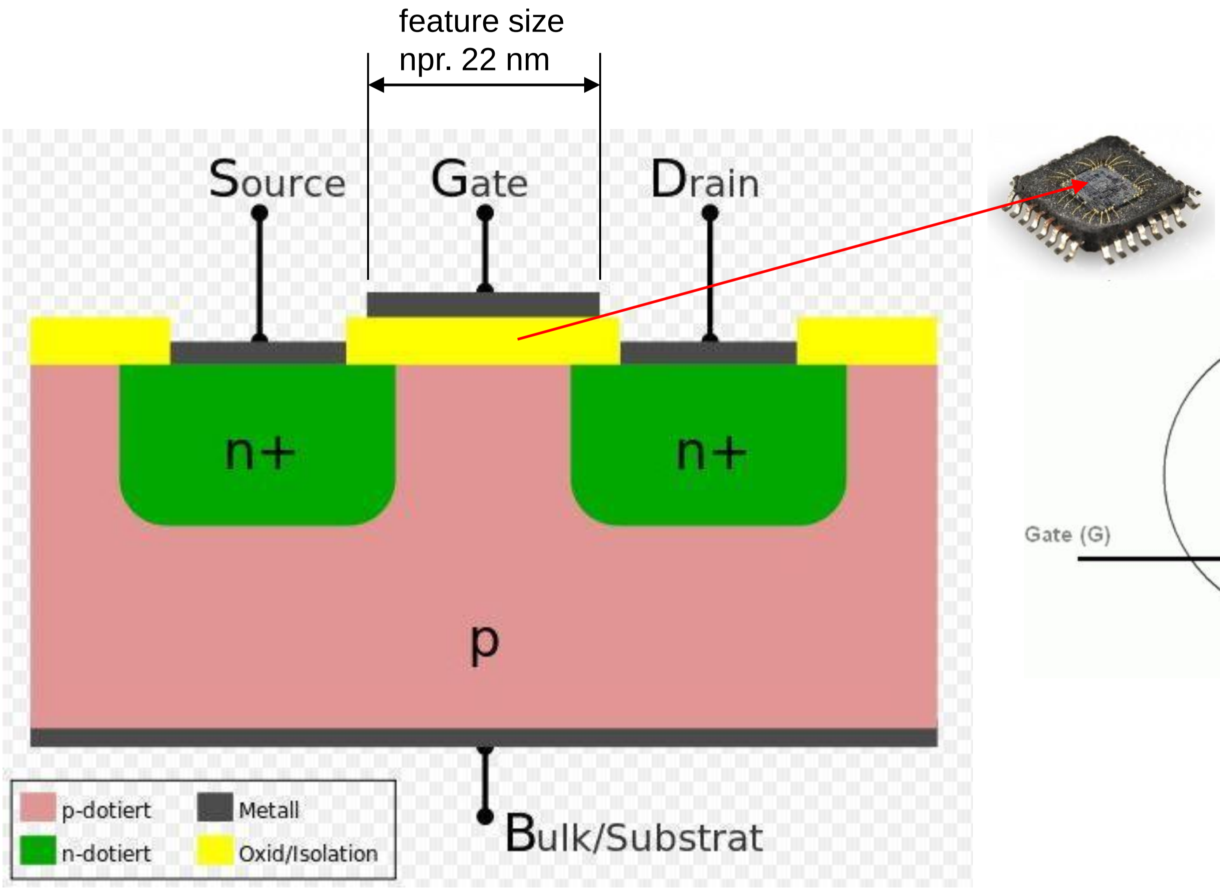


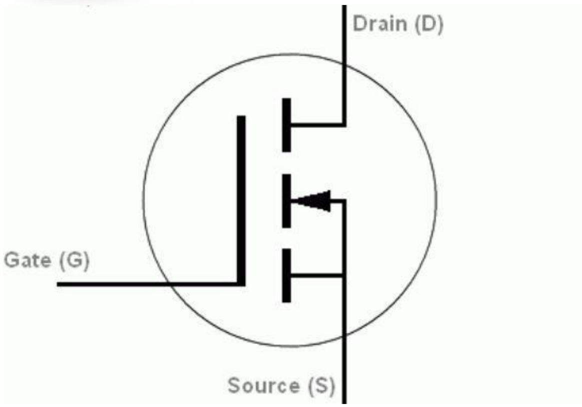


A1	B1	X1
0	0	1
0	1	1
1	0	1
1	1	0

Transistors as a part of the integrated circuit

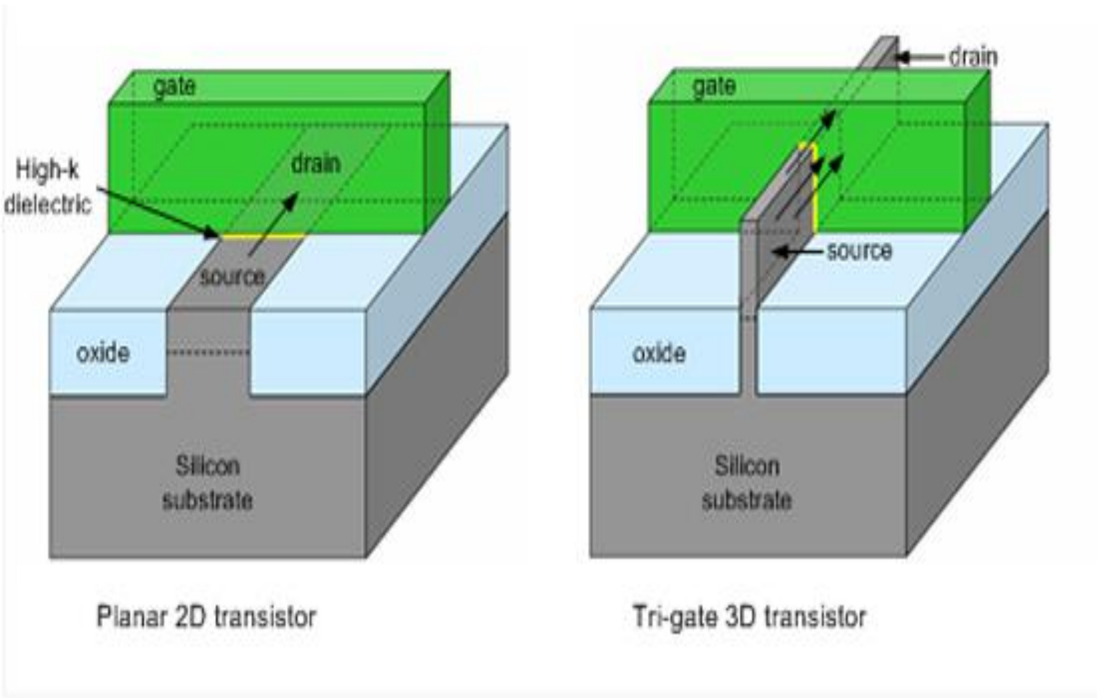






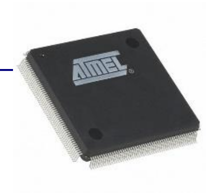
Transistors evolution in modern circuits:

- From 2D to 3D -> less space, higher density !!!

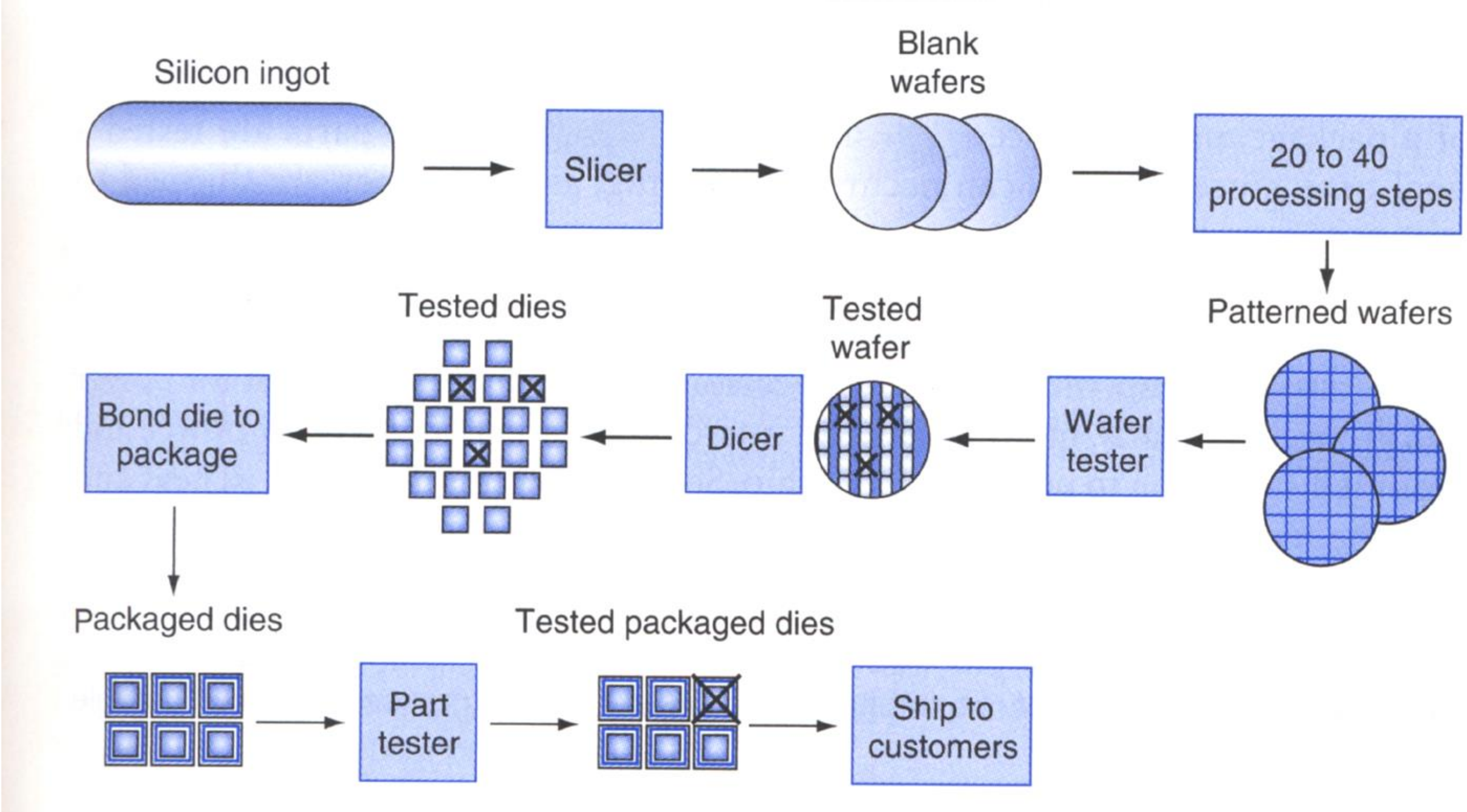


(horizontal) 3D tranzistor (vertical)

Si atom's diameter is 0.24nm!!!

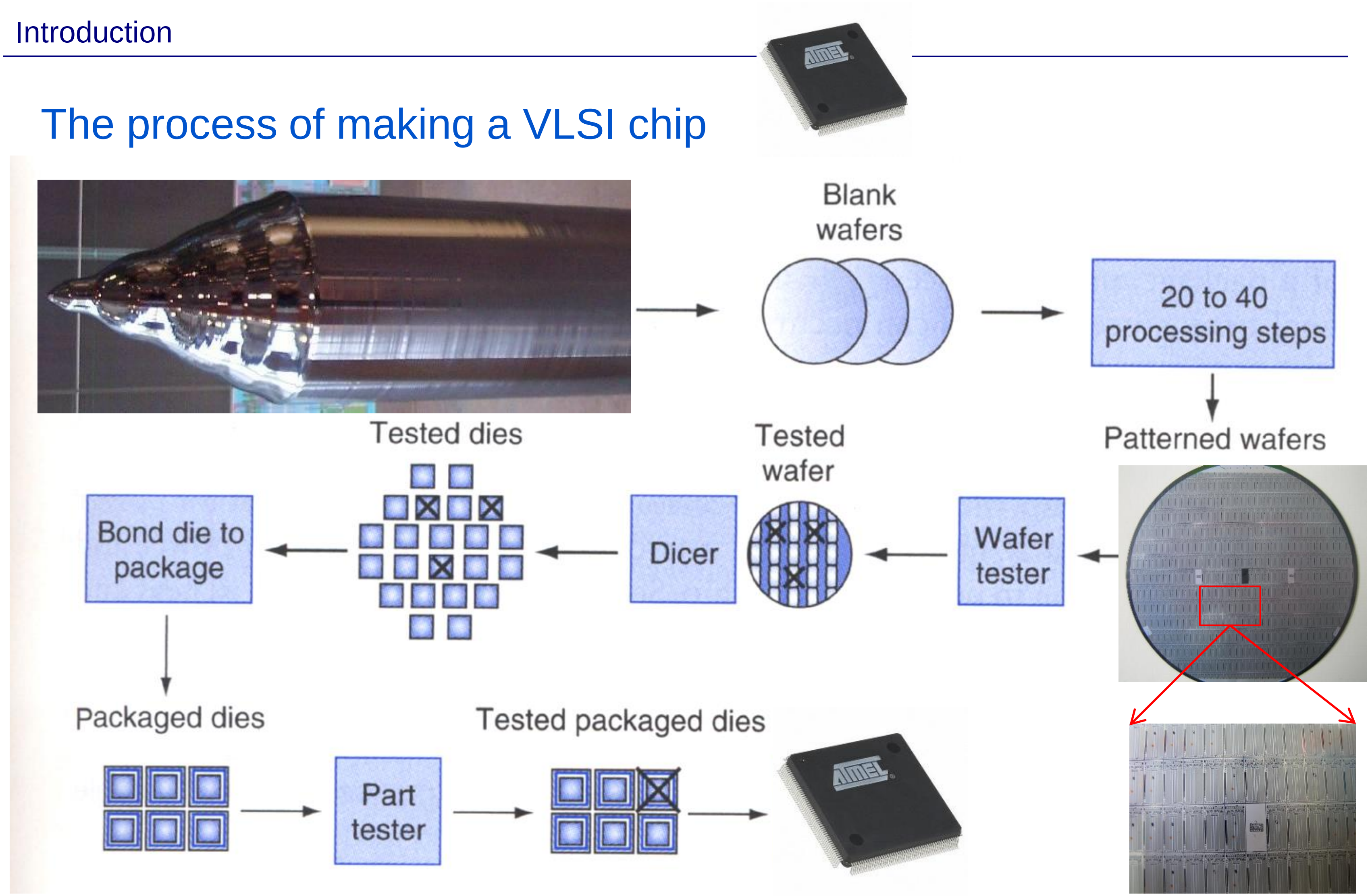


The process of making a VLSI chip



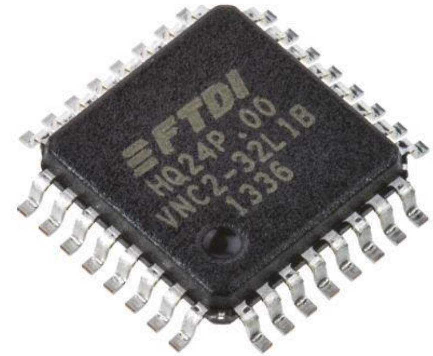
David A. Patterson, John L. Hennesy:
Computer Organization and Design, Fourth Edition

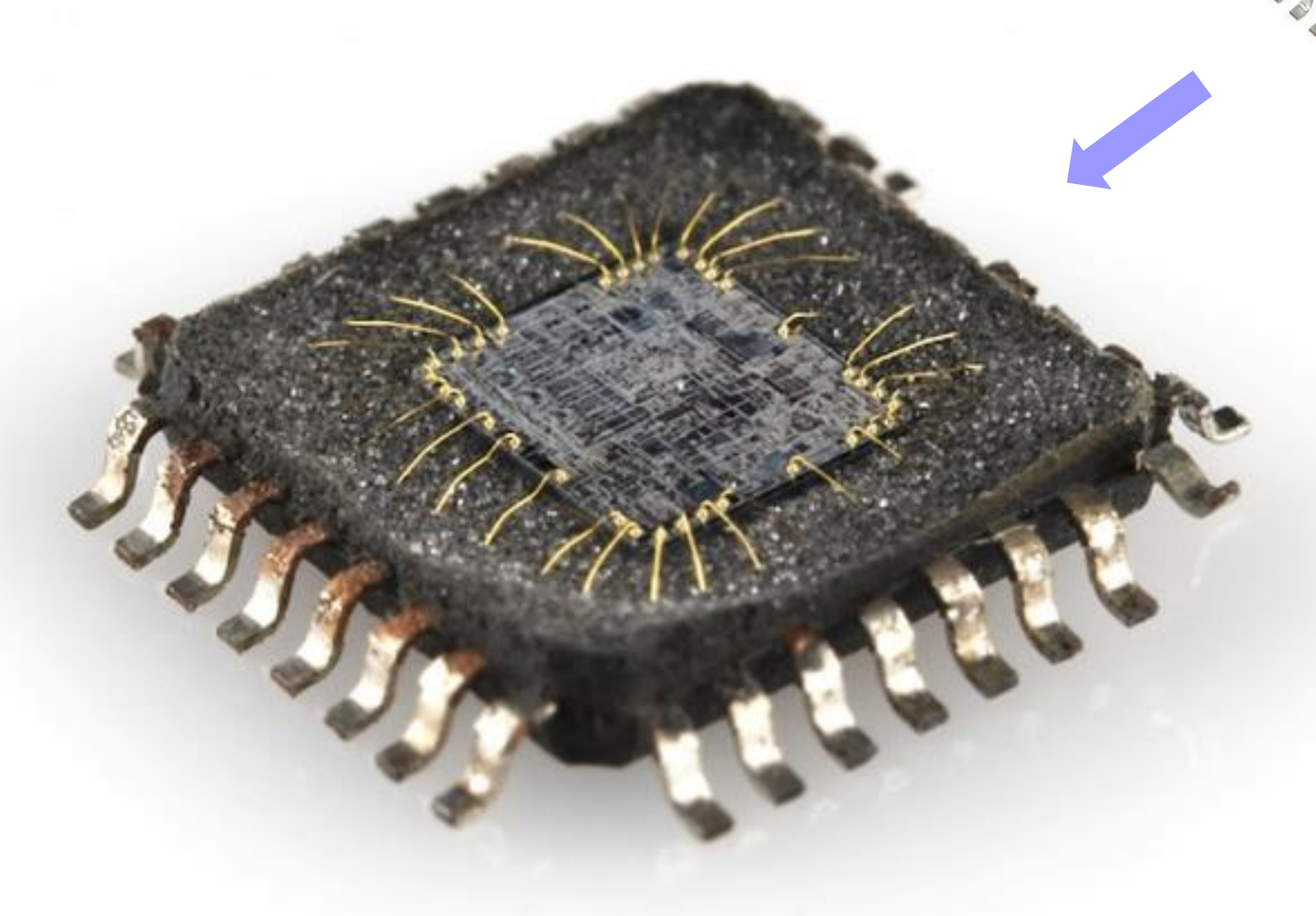
The process of making a VLSI chip



David A. Patterson, John L. Hennesy:
Computer Organization and Design, Fourth Edition

VLSI chip - inside





- ? nm process (feature size ? nm)

  - The parameter ***feature size*** in integrated circuits mostly determines the number of transistors on the integrated circuit and its properties.

  - Determines the smallest size of any object on the integrated circuit.

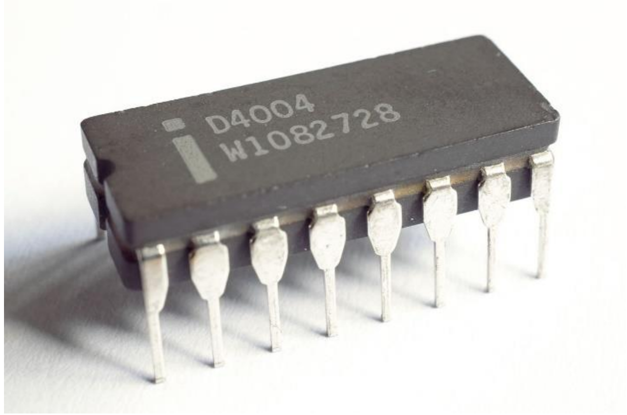
  - The object can be a part of the transistor, connection wire, space between two objects. The whole transistor is normally bigger.

  - The number of transistors on the chip depends on the size of the transistor. The number of transistors is increasing quadraticly according to the reduction of the parameter feature size

- Problems in contemporary VLSI technologies
  - Switching speed of transistor is slowly progressing
  - Density of transistors is increasing faster -> PARALLELISM
  - Reduction of elements‘ dimensions -> TROUBLE (heating, noises)
  - Excess heating dissipation -> COOLING
  - Density increase is more and more limited

Case 1:

- The first processor on a chip Intel 4004 (year 1971)
  - 2.250 transistors on a die size 3,2 x 4,2 mm
  - 10 µm process (feature size 10 µm = 10×10^{-6} m = 0,00001 m, human hair is approximately 100 µm thick)
  - 16 connectors (pins)
  - Instruction execution time 10,8 µs (= 0,0000108 s) or 21,6 µs
  - Power consumption 1,0 W
  - Price (according to nowadays standards) $26

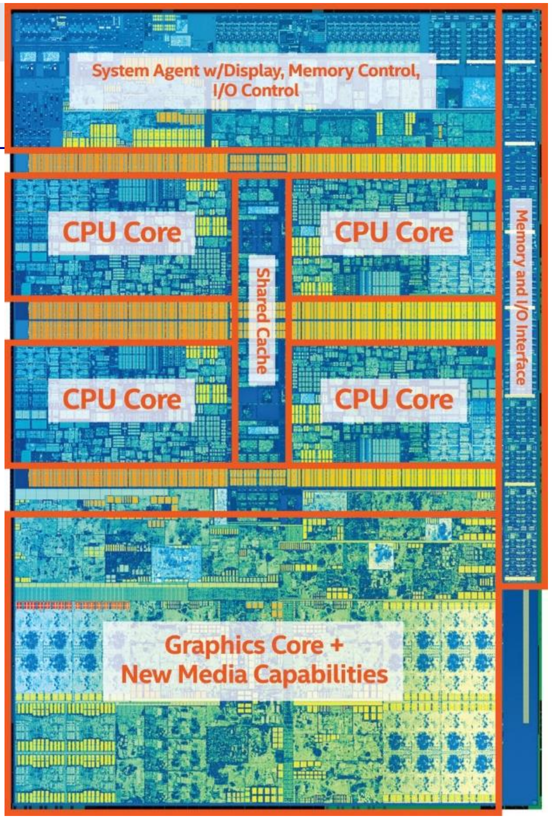


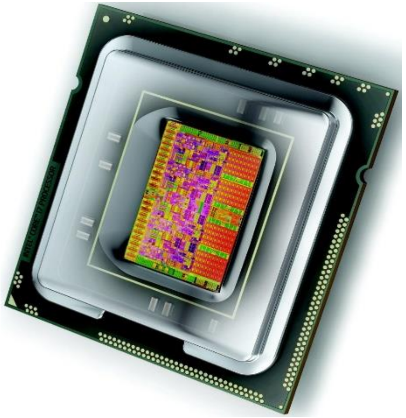
Case 2:

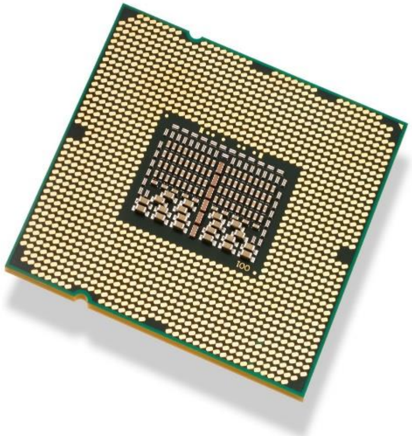
- Processor Intel i7 7700

  (microarchitecture Kaby Lake 7th generation year 2017):

  - Number of transistors - Intel doesn't disclose this number
  - 14 nm process (14nm = 14×10^{-9} m = 0,000000014 m)
  - Size of the chip - Intel doesn't disclose this information
  - 4 cores (4 processors, 8 threads), graphical processor
  - 1155 connectors (pins)
  - Power consumption (TDP) 65 W
  - Recommended Price (Intel) 303 $ - 312 $







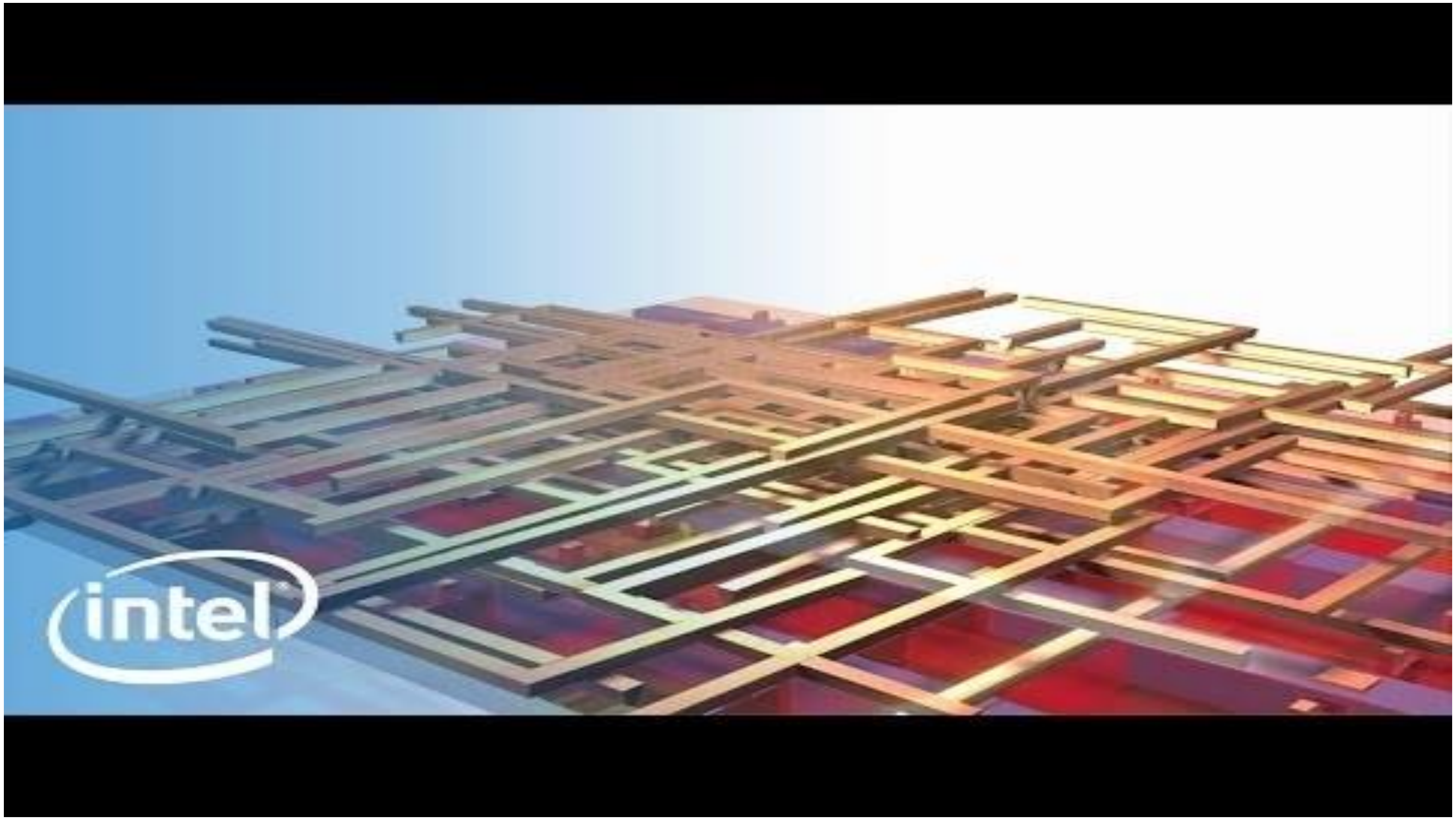
Case 3:

- Processor Intel i9-11900

  (microarchitecture Rocket Lake 11th generation year 2021):

  - Number of transistors - Intel doesn't disclose this number
  - 14 nm process (14nm = 14×10^{-9} m = 0,000000014 m)
  - Size of the chip - Intel doesn't disclose this information
  - 8 cores (16 threads), graphical processor
  - 1200 connectors (pins)
  - Power consumption (TDP) 65 W
  - Recommended Price (Intel) 439 $ - 449 $

Intel: The Making of a Chip with 22nm/3D Transistors (Youtube Video)



https://www.youtube.com/watch?v=d9SWNLZvA8g&ab_channel=Intel