COMPUTER ARCHITECTURE

1 Introduction

© 2022, Rozman, Škraba, FRI

1. Introduction :

© 2022, Rozman, Škraba, FRI

1. Introduction :

 © 2022, Rozman, Škraba, FRI

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

RA VSP 2022/23

- MS Teams
  - Team enter code: kygg8n7

- Office hours: currently on Tuesdays at 18:15 in R2.40, 50

  Possible changes will be posted to the web classroom. Announce over email.
  Announce: email or https://calendly.com/rrozman/govorilne (experimental)

Team CA

Tutors

Žiga Pušnik
ziga.pusnik@fri...
.

Anamari Orehar
ao6477@student.uni-lj.si

Kristian Šurbek
ks5453@student.uni-lj.si

Andrej Sušnik
as1767@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

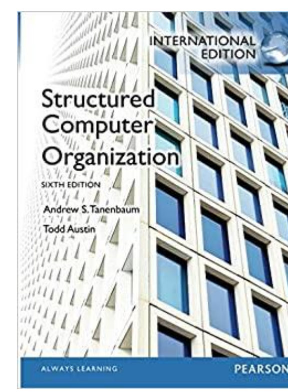
  - MS Teams (chat, lecture notes)
  - Common (shared) notes – Gdocs

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

Computer Architecture - Crowd-sourced Shared Notes

Računalniška arhitektura - Deljeni zapiski za skupno dopolnjevanje

  - Basic, includes more comprehensive 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

© 2022, Rozman, Škraba, FRI

Grades STUDIS

Povpr.ocena/max·[št.odg./vsi]¤	2021/22-Nosilec¤	2020/21-Nosilec¤	2019/20-Nosilec¤
Predmet¤	4.53/5·[192/196]¤	4.65/5·[154/161]¤	4.54/5·[149/150]¤
Izvajalec¤	4.75/5·[192/196]¤	4.74/5·[154/161]¤	4.79/5·[149/150]¤

Important :

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

Surveys (2018/21) - highlights:

- Kahoot! I learned a lot from kahoot and the RAwall.
- OneNote notebooks with notes, useful and interesting exercises, Kahoots, good pdfs
- Current news were often mentioned in the lectures, which encouraged critical and independent thinking
- Great emphasis is placed on understanding, and not on learning by heart
- The lecturer's energy, practical (life) examples
- A good learning system for foreign students
- ... the assistants and especially the professor can see that they don't care about our knowledge. Because of this, the quality of lectures and exercises is known.

To Improve:

- Theory sometimes becomes difficult to understand, it is difficult to follow new concepts and ideas.
- Lecture material <> exercise
- "Not perfect, which is characteristic of everyone"

© 2022, Rozman, Škraba, FRI

What‘s new in 2022:

- Live lectures and lab sessions (one lab session in English)
  - 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 : kygg8n7

- Important :
  - be active
  - cooperate, talk, ask, comment, …
  - all major documents are translated into English
  - testing of real-time Slovene-English translation – project ON
  - please help us on English documents (typos, missing content, …)

© 2022, Rozman, Škraba, FRI

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

≡ FRI Učilnica FRI 22/23 Slovenščina (sl)

Računalniška arhitektura

Nadzorna plošča / Moji predmeti / ra

Splošno - General

- Splošne informacije - General info
- Forum novic - News Forum
- Forum - RA - vprašanja in odgovori - Q&A
- Wiki RA
- RAWall - Padlet komunikacijski kanal za predavanja

Izpitni roki - Exams 2022/2023

- Izpitni roki pravila - Exams Rules
- Primeri prejšnjih izpitov - Selected previous exams

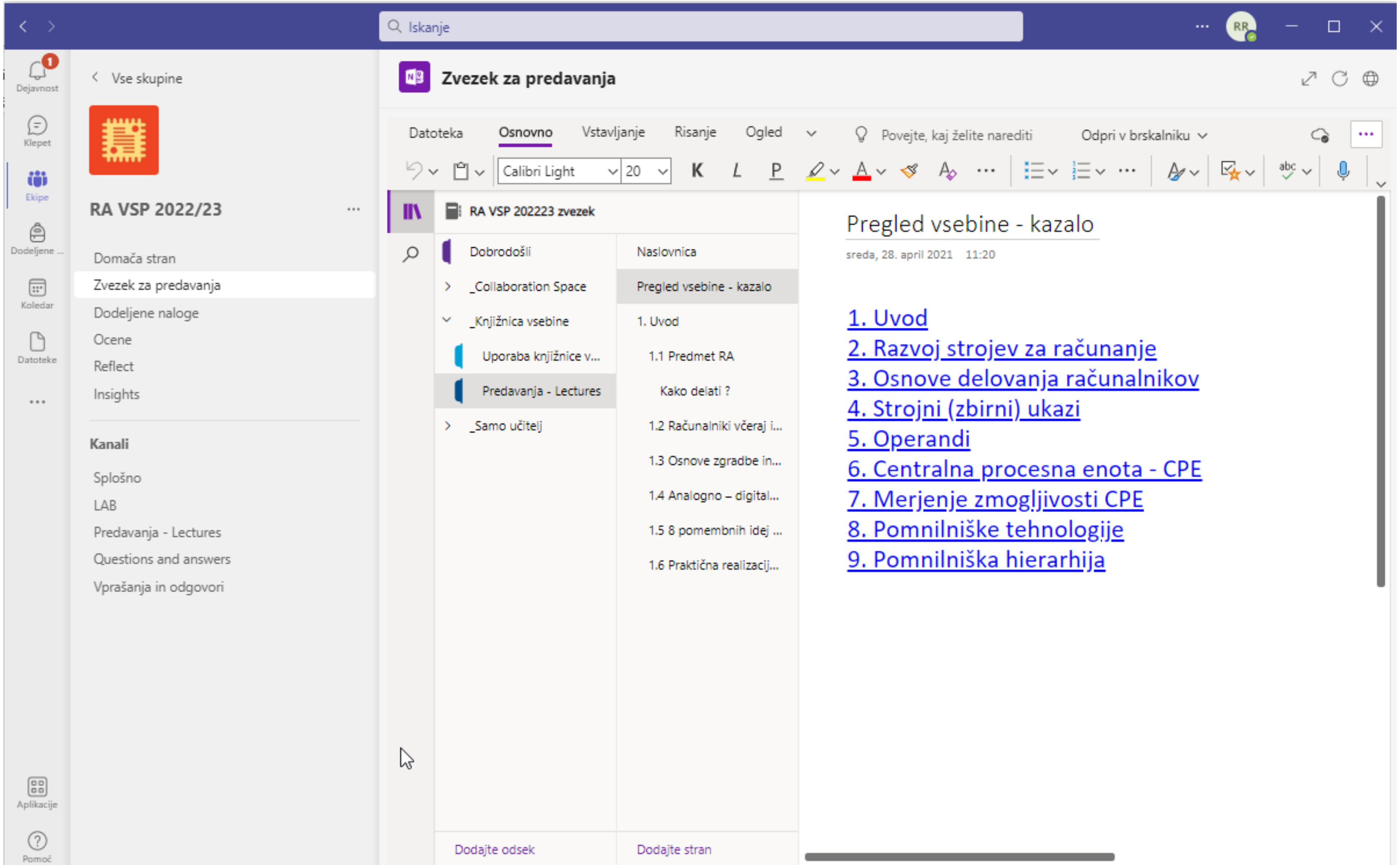
Predavanja/Lectures 2022/23

------------------------- **Arhiv: Predavanja/Lectures 2021/22** ---------------------------------

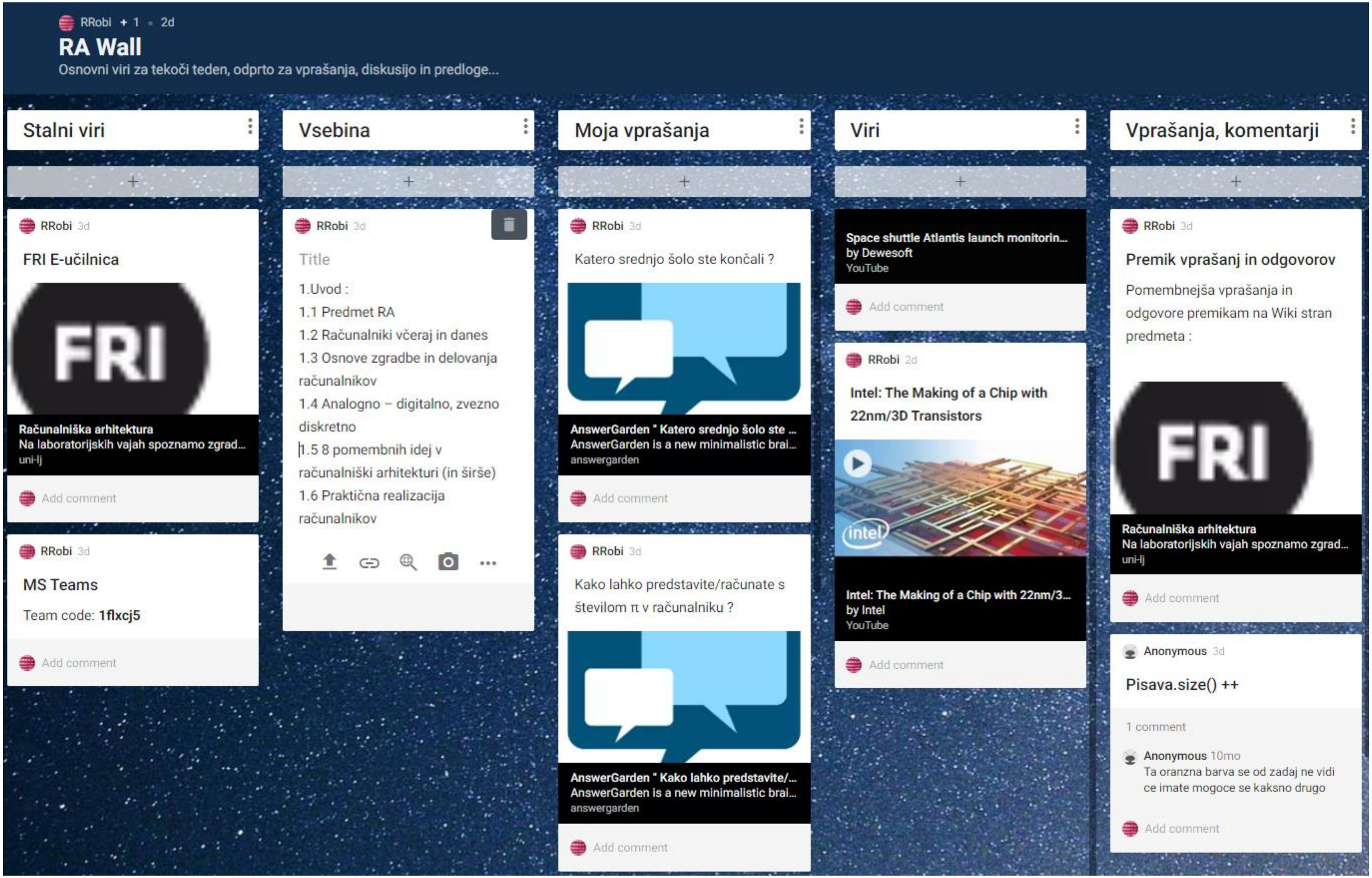
- RA-1 Uvod Uploaded 6/10/21, 23.35
  - RA-1 Introduction Uploaded 6/10/21, 23.36
  - RAM_pomnilnik_demo.circ Uploaded 6/10/21, 23.37
  - Pomen poznavanja računalniške arhitekture, Miha_Krajnc Uploaded 14/10/21, 02.31

© 2022, Rozman, Škraba, FRI

MS Teams: chat, OneNote notebook



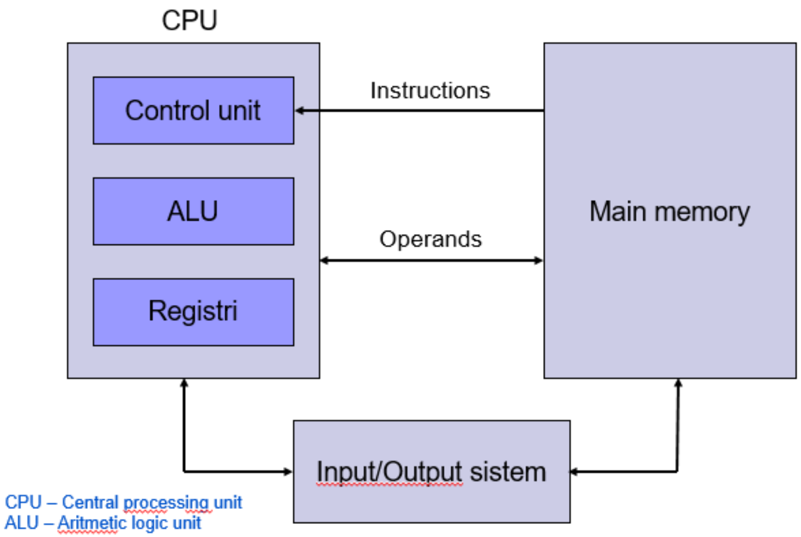
Chapter related content https://padlet.com/rawall/RAWall



What will you learn on the course of Computer Architecture?

Lectures, Lab sessions

© 2022, Rozman, Škraba, FRI



Lectures content:

© 2022, Rozman, Škraba, FRI

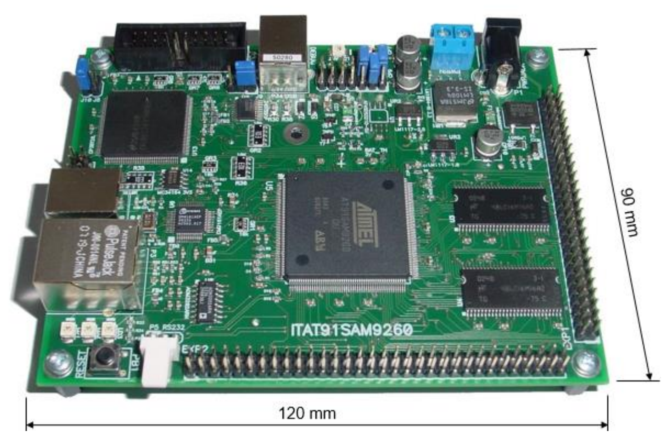
Laboratory work contents:

- Learn the basics of computer architecture from a practical point of view
- Understand the inner workings of computers (ARM) by programming 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

Computer STM32H750-DK

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



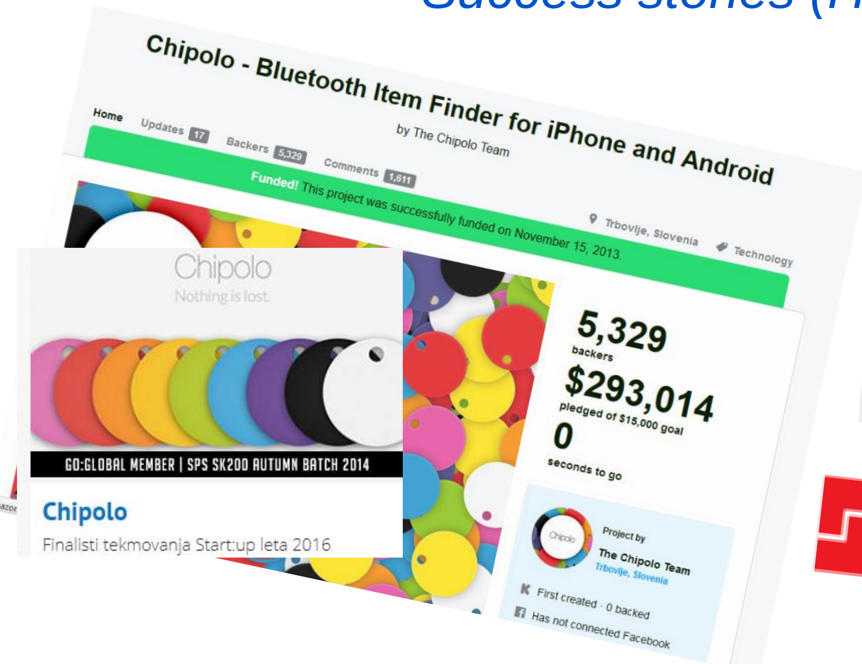
© 2022, Rozman, Škraba, FRI

Why is Computer Architecture important ?

- 4 questions and
- 4 answers

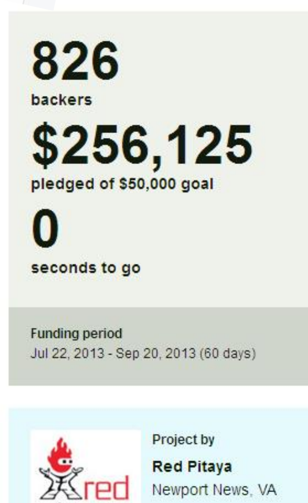
1. Complete products (HW+SW)

Success stories (HW+SW)



COSYLAB

iSYSTEM



2. Why Comp.Arch., HW ?

- Because knowledge on computer architecture and operation leads to more efficient programming (programs).
  - Case 1: 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];
```

 © 2022, Rozman, Škraba, FRI

2. Why Comp.Arch., HW ?

- Because knowledge on computer architecture and operation leads to more efficient programming (programs).
  - Case 2: 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?“

© 2022, Rozman, Škraba, FRI

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

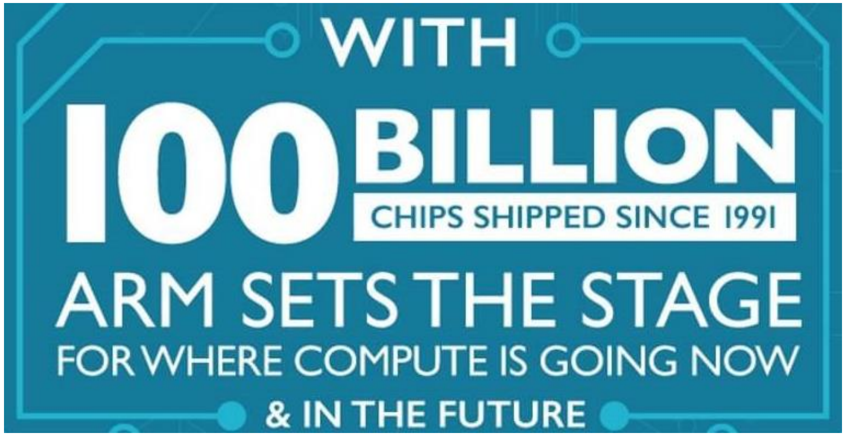
4. Why the ARM architecture?

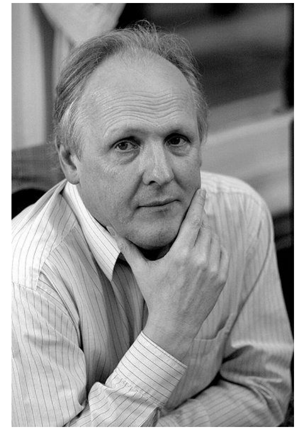
Because ??? …“

 © 2022, Rozman, Škraba, FRI

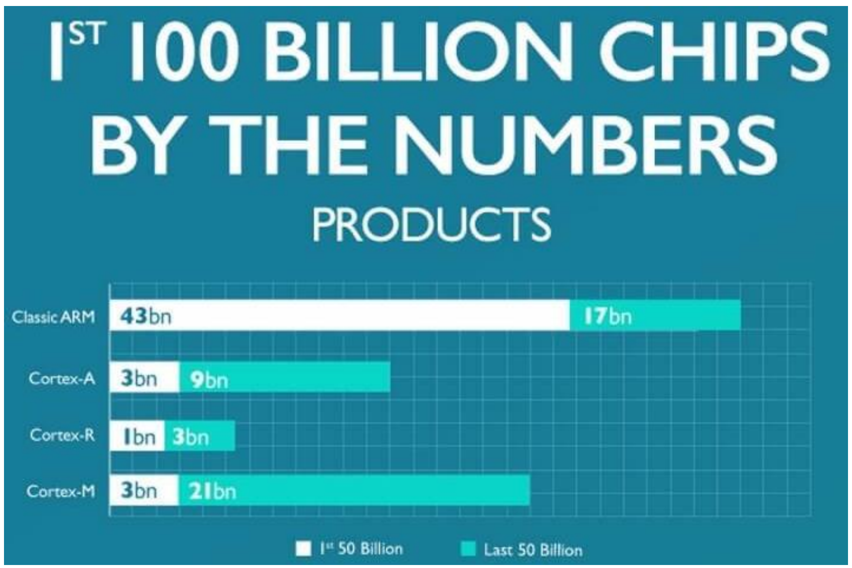
4. Why the ARM architecture?

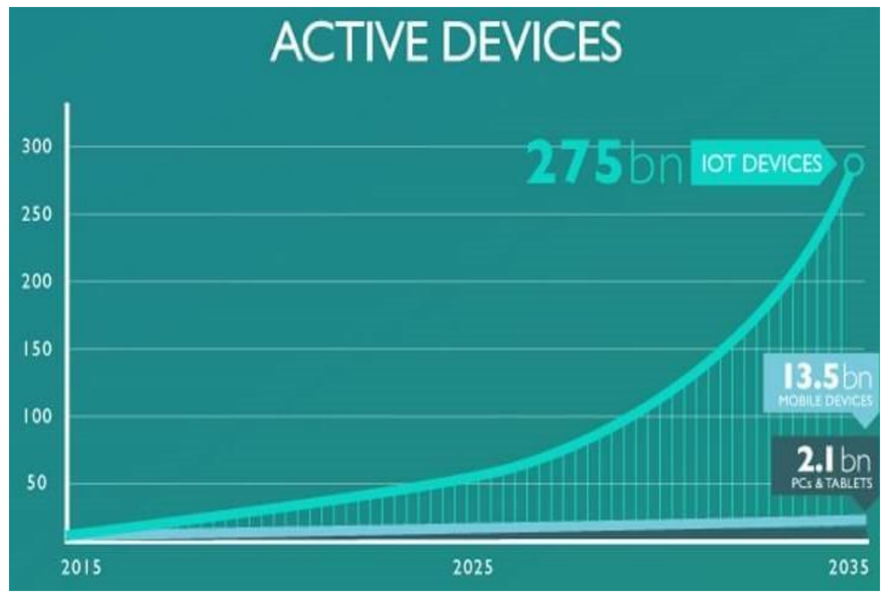
„Steve Furber on FRI“





principal designer of the [BBC Micro]() and the [ARM]() [32-bit]() [RISC]() [microprocessor]().[15]





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

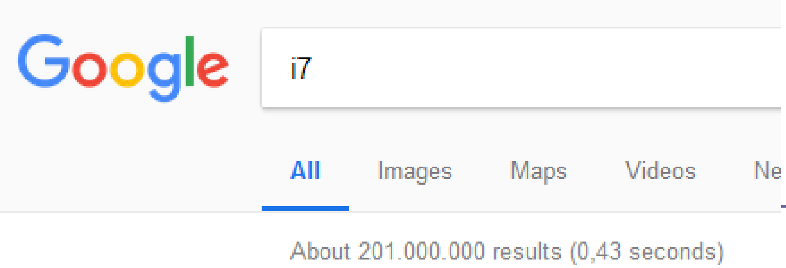
© 2022, Rozman, Škraba, FRI

1. Introduction :

- 1.1 CA course
- 1.2 About computers
- 1.3 Basic computer structure and operation
- 1.4 Analog – digital, continuous-discrete
- 1.5 8 important ideas in computer architecture (and in general)
- 1.6 Computer realization

 © 2022, Rozman, Škraba, FRI

- 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)



© 2022, Rozman, Škraba, FRI

- 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).

© 2022, Rozman, Škraba, FRI

- 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*)

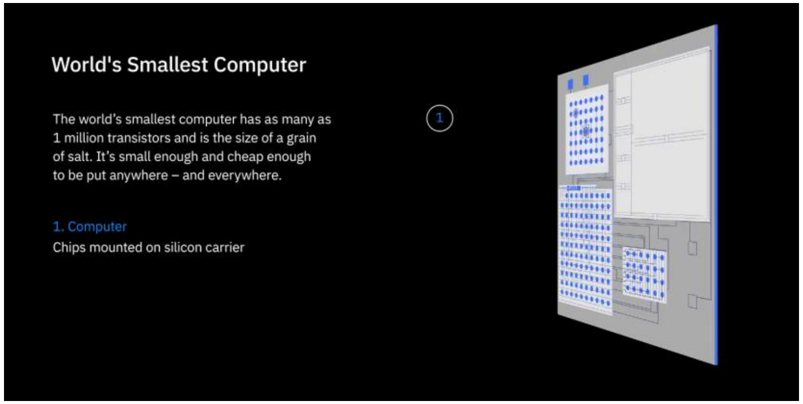
© 2022, Rozman, Škraba, FRI
https://www.top500.org/lists/2019/06/

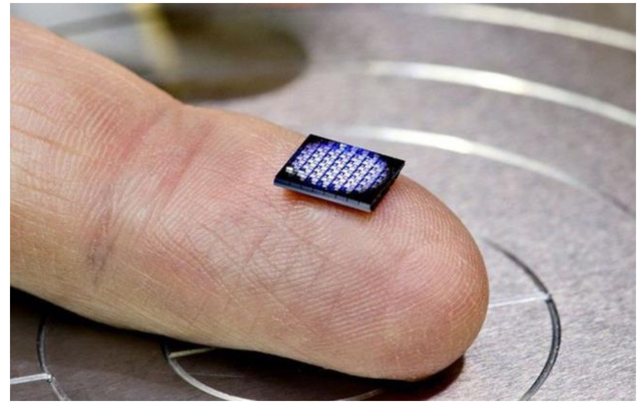
- Currently the most powerful computer in the world :
  - SUPERCOMPUTER FUGAKU in Kobe, Japan
  - 7 630 848 cores
  - Performance 537 212 TFLOPS
  - Power consumption 29 899 kW (*Hydro PowerPlant Medvode 26 700 kW*)

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

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

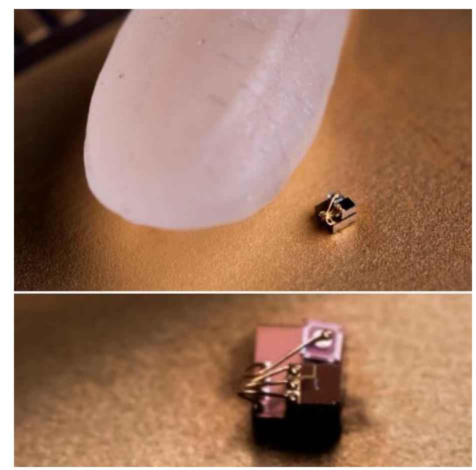
- Currently the most miniature computer in the world (year 2018):   ?





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

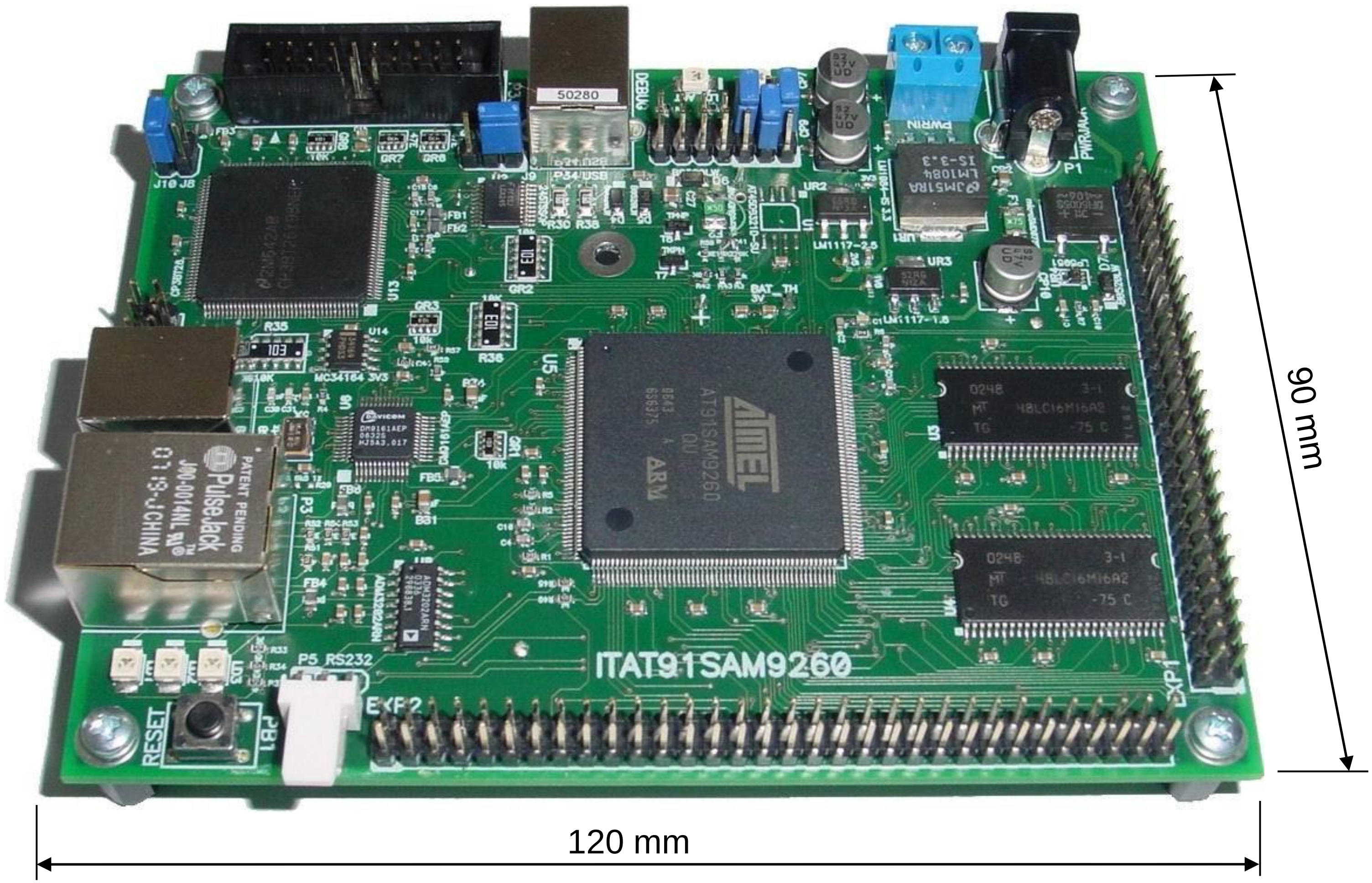
The university on Thursday said its engineers have produced a computer that's 0.3 mm x 0.3 mm -- it would be dwarfed by a grain of rice. While it drew comparisons to IBM's own 1mm x 1mm computer, Michigan's team said the creation is about more than just size.



- 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



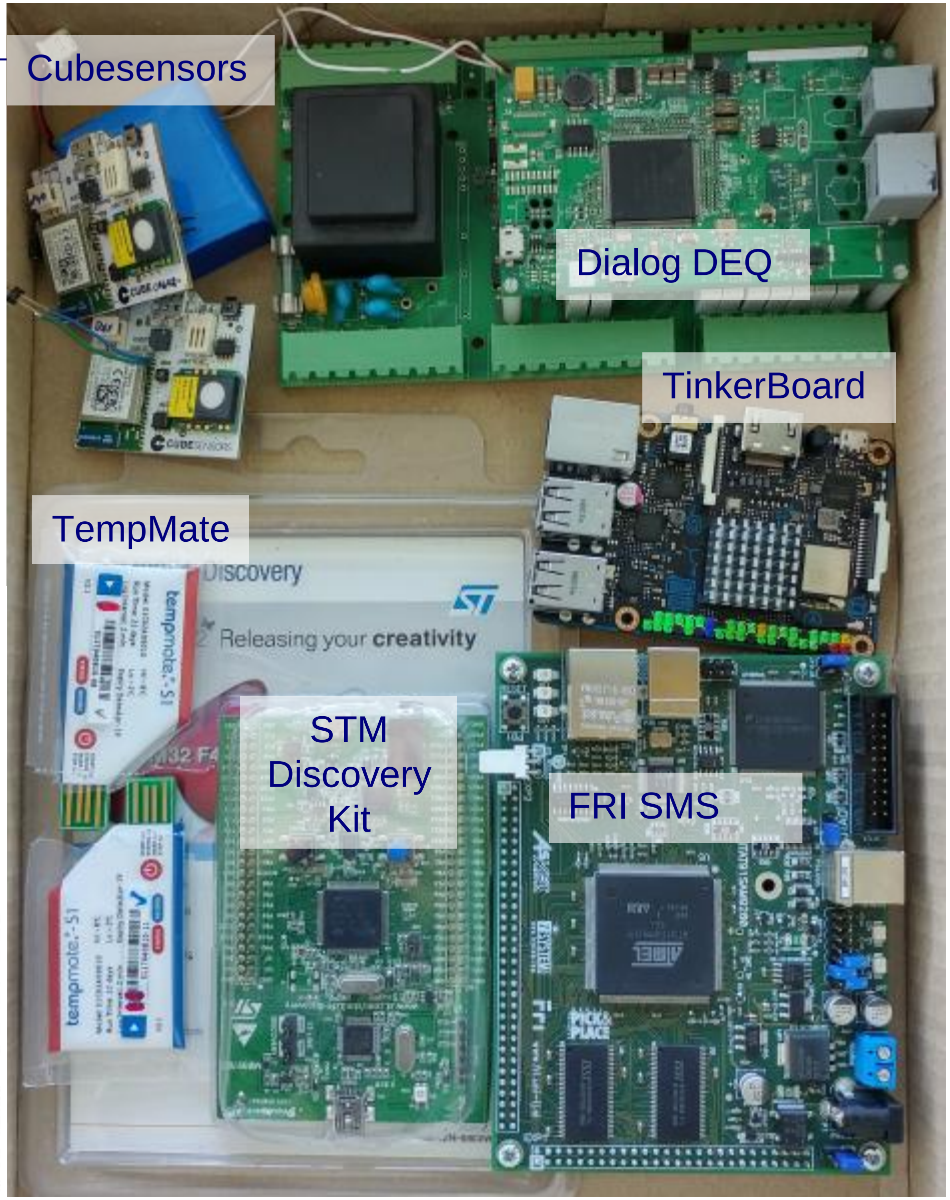
© 2022, Rozman, Škraba, FRI

- Nowadays, computers can be attributed into three functional categories:
  - Personal Computers (laptop, tablet, . . .)
  - Servers
    - There are significant 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, …

 © 2022, Rozman, Škraba, FRI

Embedded computers (practical examples)

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



 © 2022, Rozman, Škraba, FRI

1. Introduction :

- 1.1 CA course
- 1.2 About computers
- 1.3 Basic computer structure and operation
- 1.4 Analog – digital, continuous-discrete
- 1.5 8 important ideas in computer architecture (and in general)
- 1.6 Computer realization

© 2022, Rozman, Škraba, FRI

Basic definitions

Def: Computer architecture is

- consideration of the programmer‘s 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.

 © 2022, Rozman, Škraba, FRI

Operation of (digital) computers

- Computer architecture is also the 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 built into computers. Computers from different manufacturers can generally have different machine code instructions.

Computer „understands“ own machine instructions only !!!

 © 2022, Rozman, Škraba, FRI

What is the computer doing ?

(How does it work ?)

$\boxed{\text{Executing instructions !}}$

© 2022, Rozman, Škraba, FRI

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

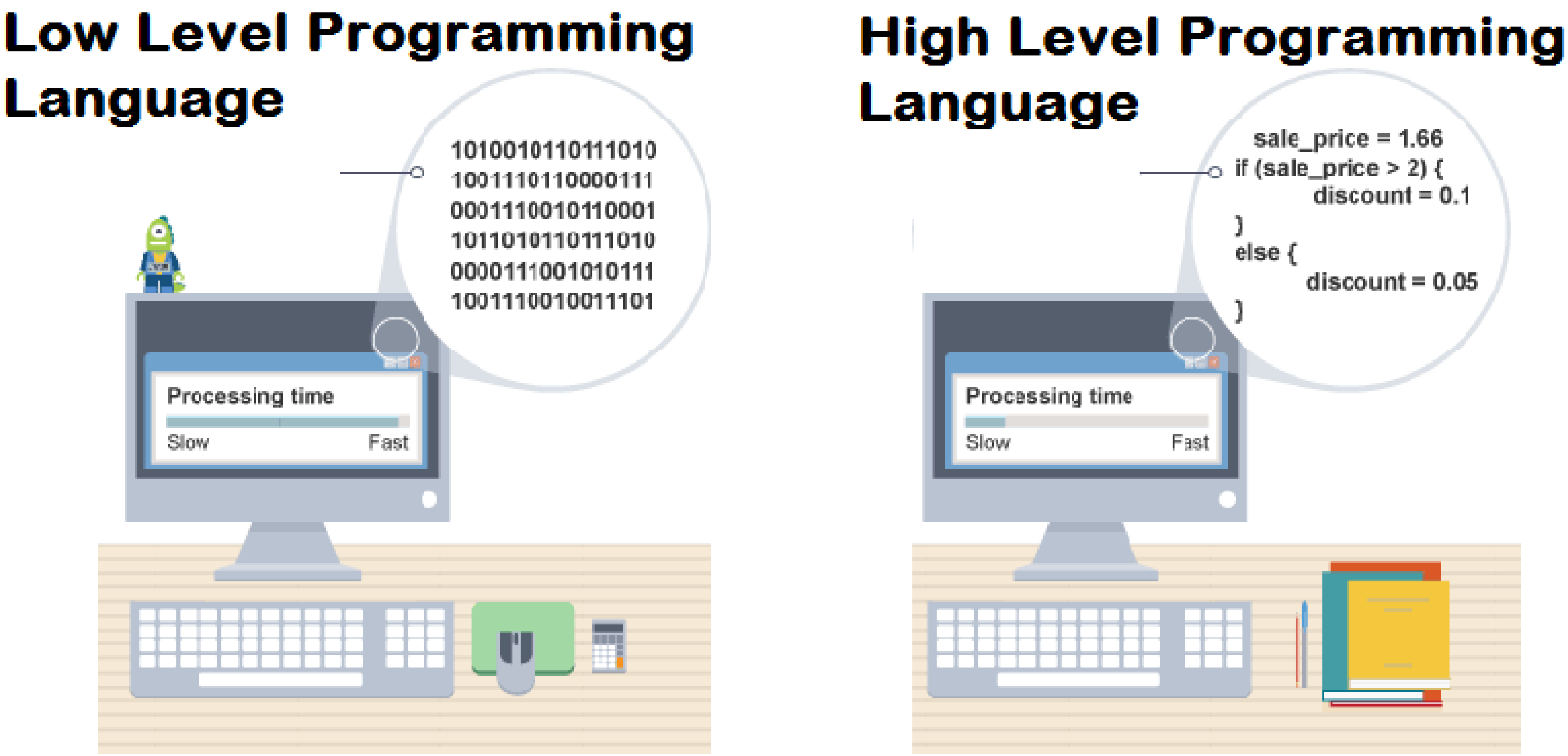
© 2022, Rozman, Škraba, FRI

- 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.

 © 2022, Rozman, Škraba, FRI

machine langauge <-> high-level languages ?

- Portability vs speed

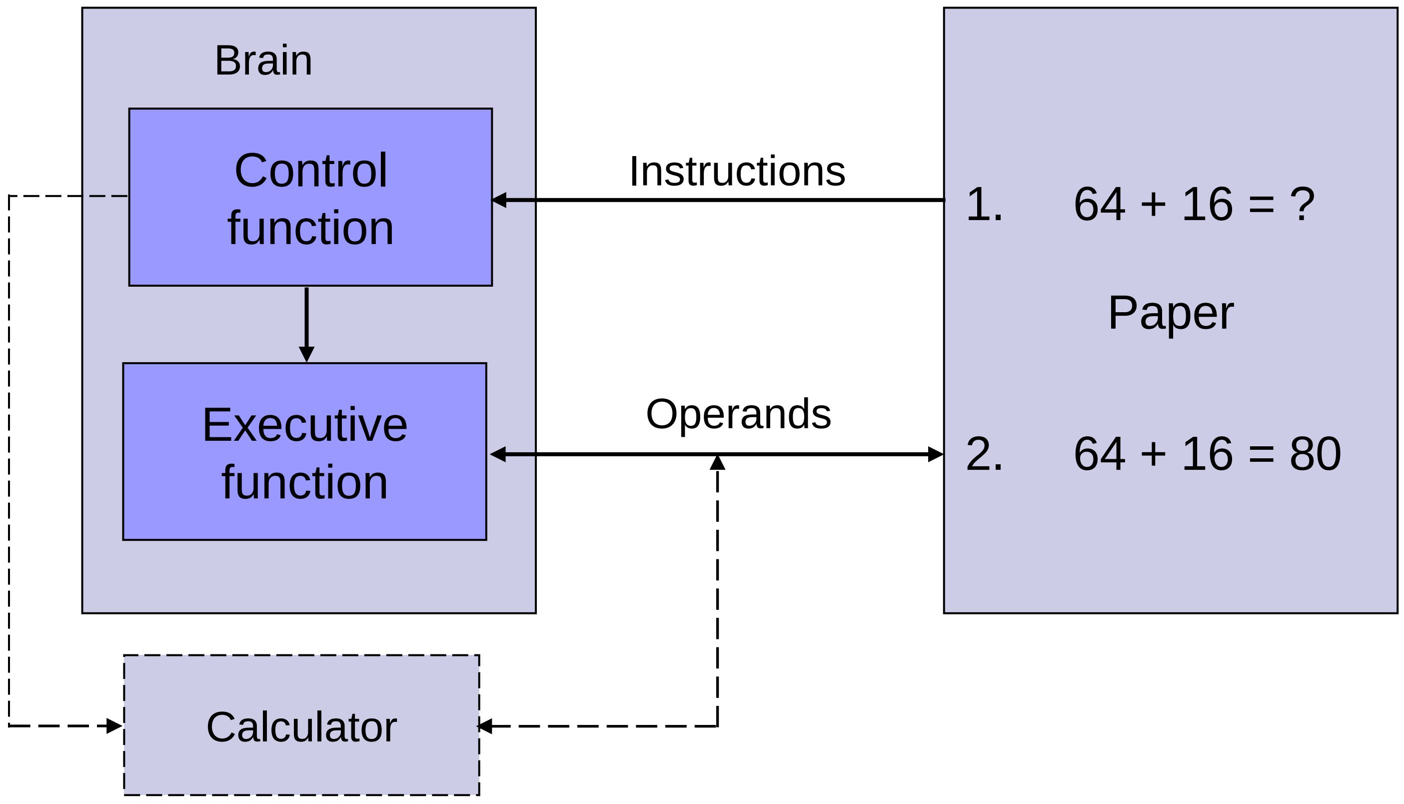


© 2022, Rozman, Škraba, FRI

The relation between manual and machine model of computing

 © 2022, Rozman, Škraba, FRI

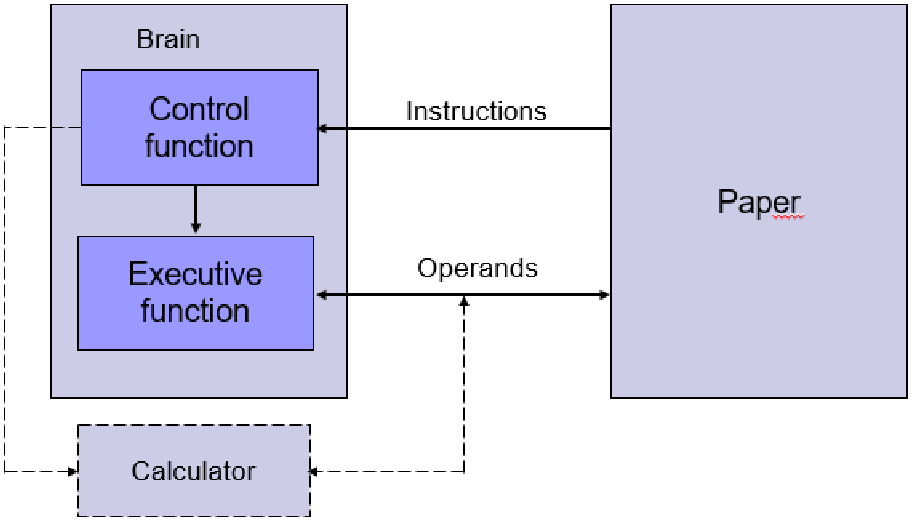
Manual computing



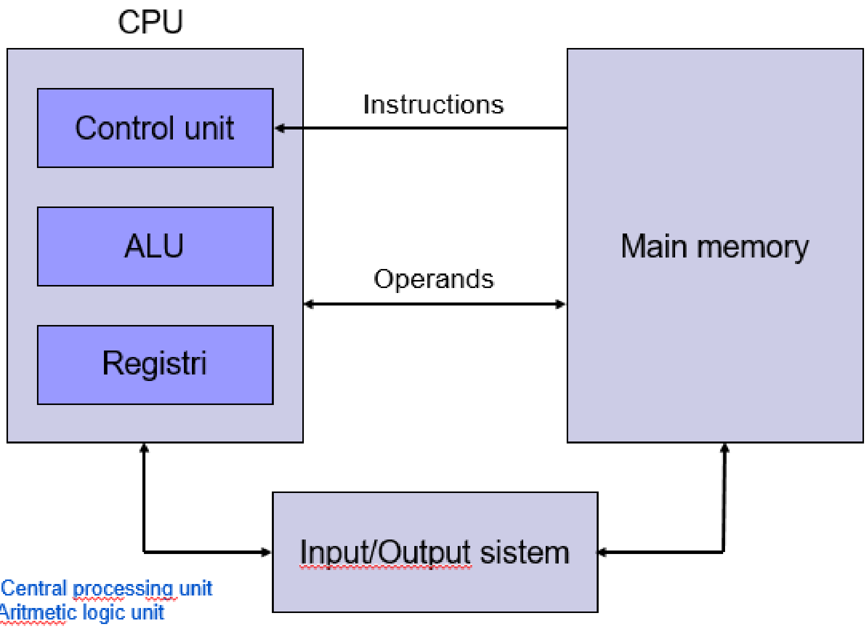
© 2022, Rozman, Škraba, FRI

Comparison between models of computing

Manual computing



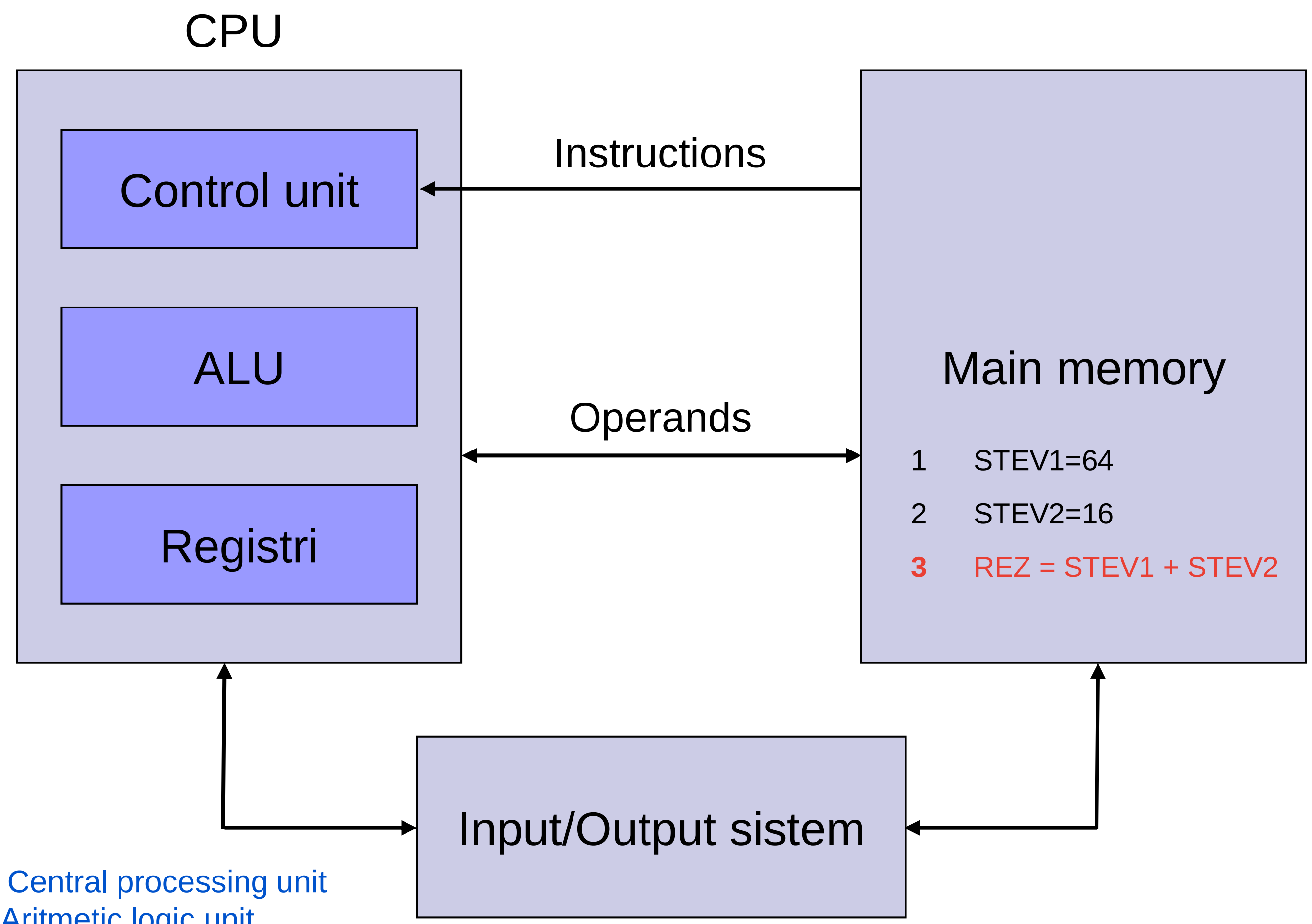
Structure of a typical computer



CPU – Central processing unit
ALU – Aritmetic logic unit

© 2022, Rozman, Škraba, FRI

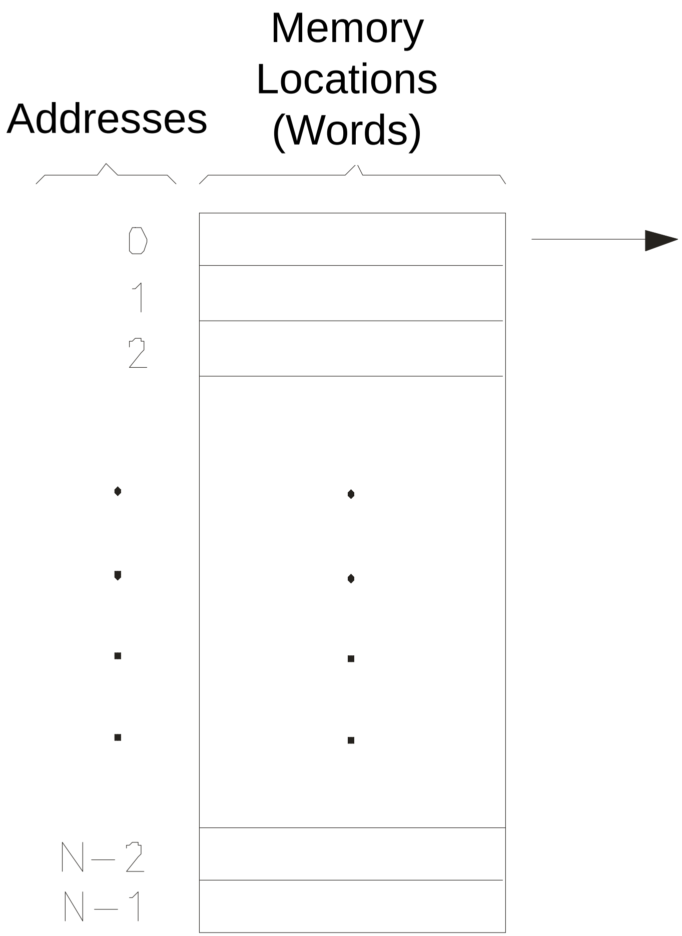
Structure of a typical computer

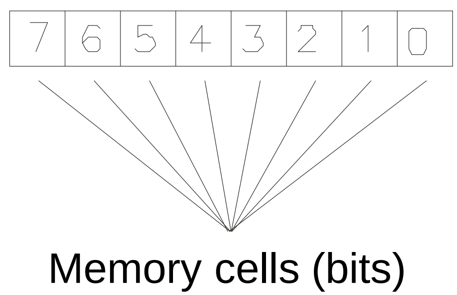


CPU – Central processing unit
ALU – Aritmetic logic unit

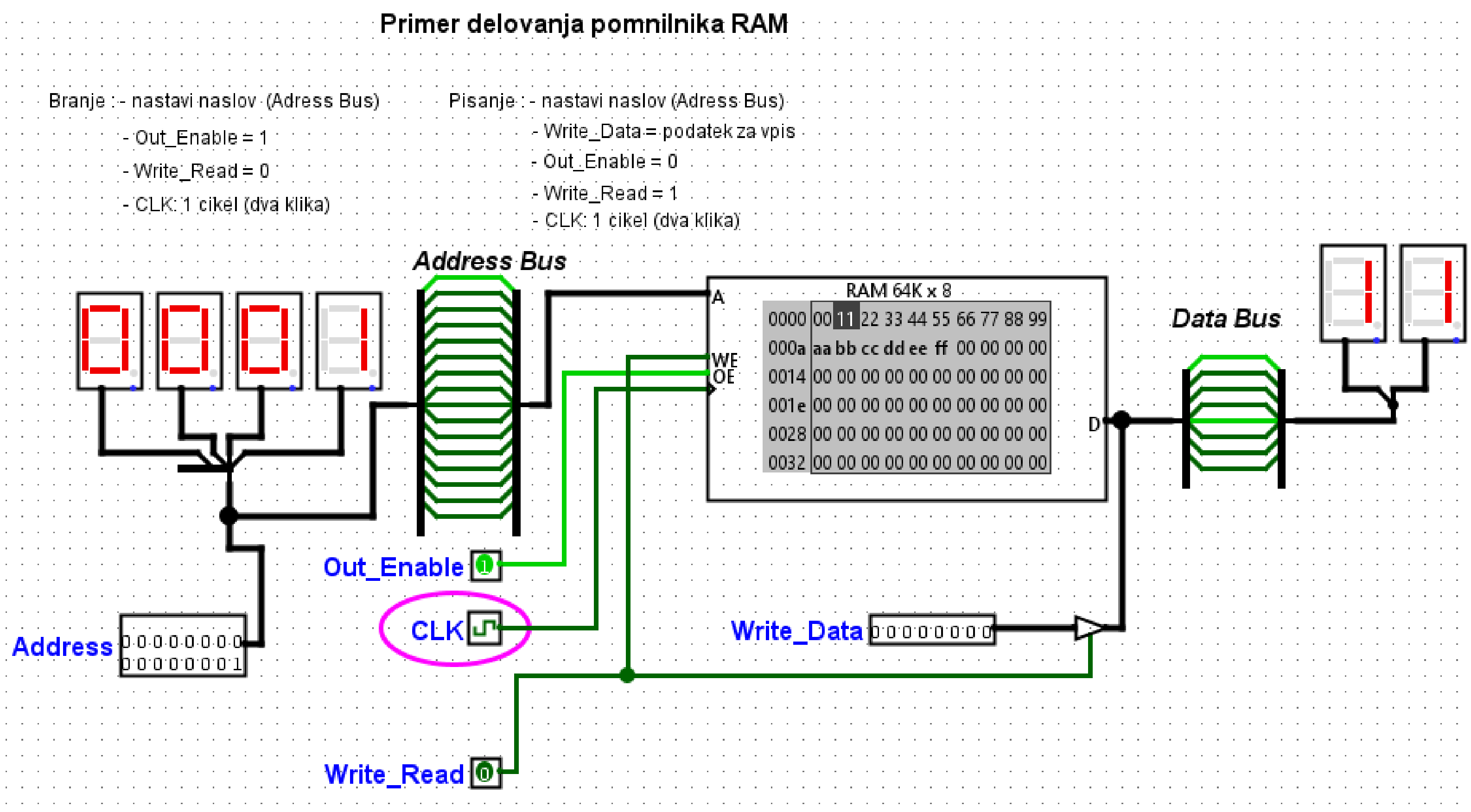
© 2022, Rozman, Škraba, FRI

Memory


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

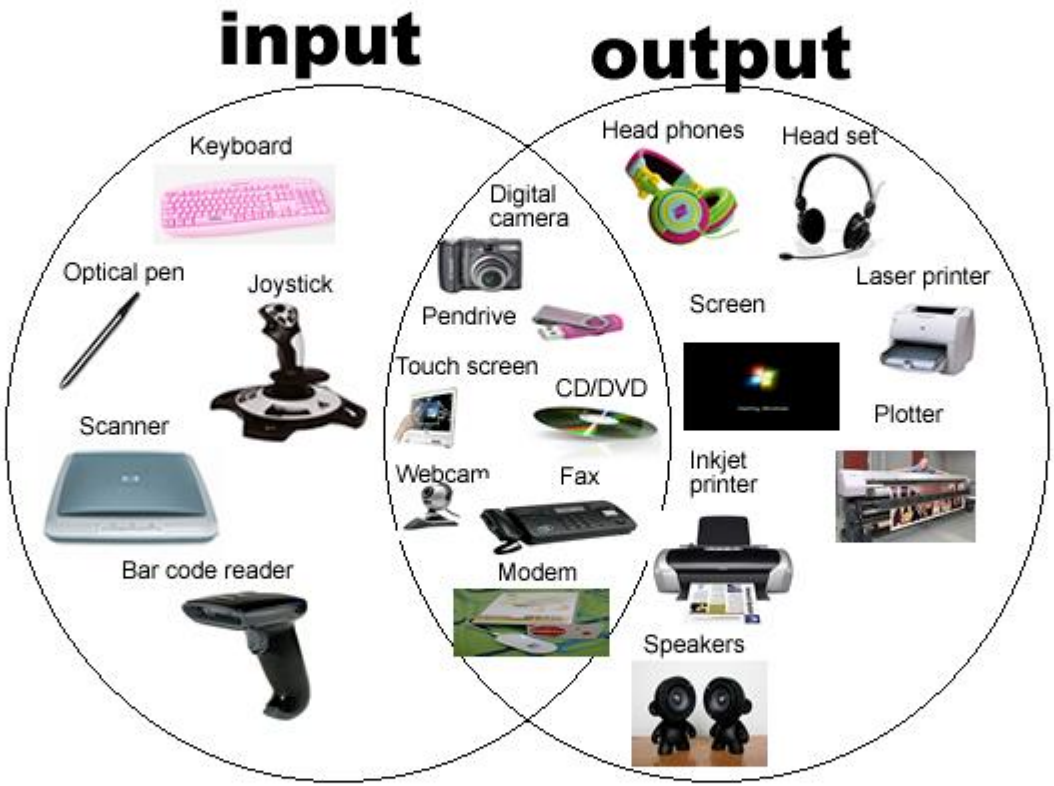

7 6 5 4 3 2 1 0
Memory cells (bits)

Memory
Logisim EVO Demo



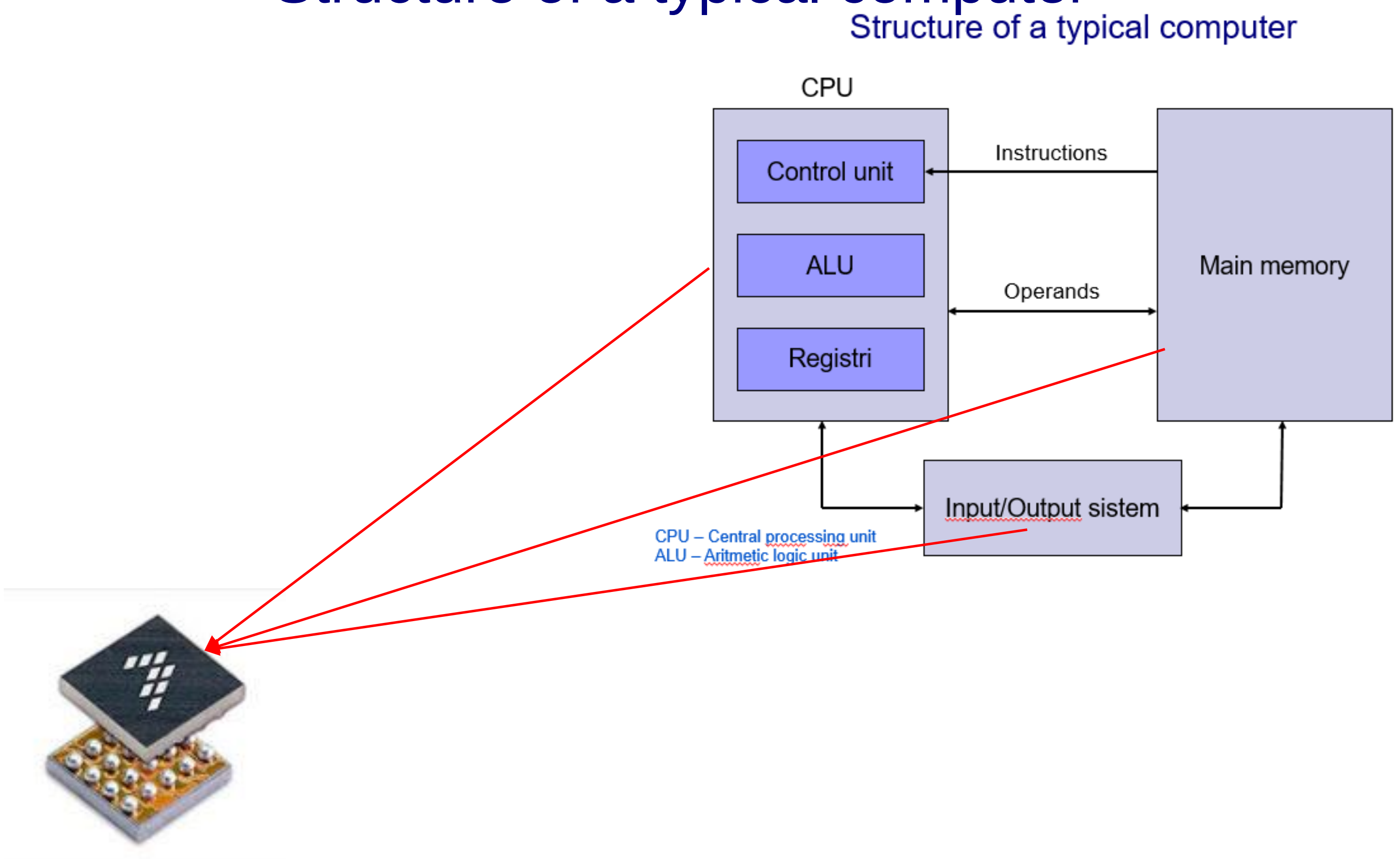
RAM_pomnilnik_demo_EVO.circ

Input-Output system (devices)



 © 2022, Rozman, Škraba, FRI

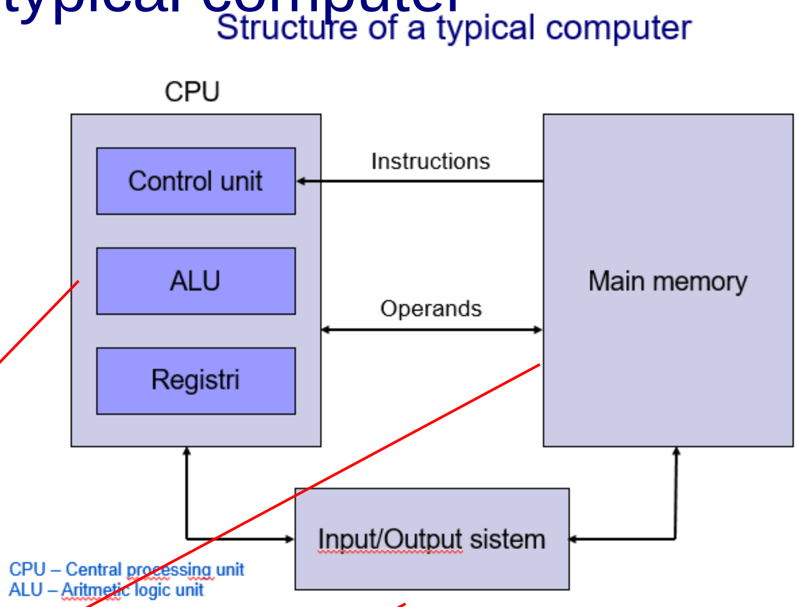
Structure of a typical computer

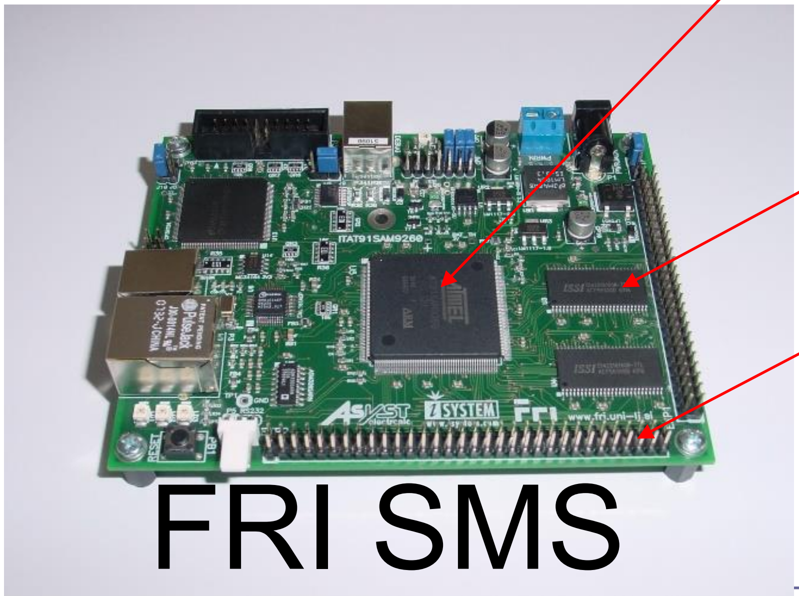


Microcontroller

© 2022, Rozman, Škraba, FRI

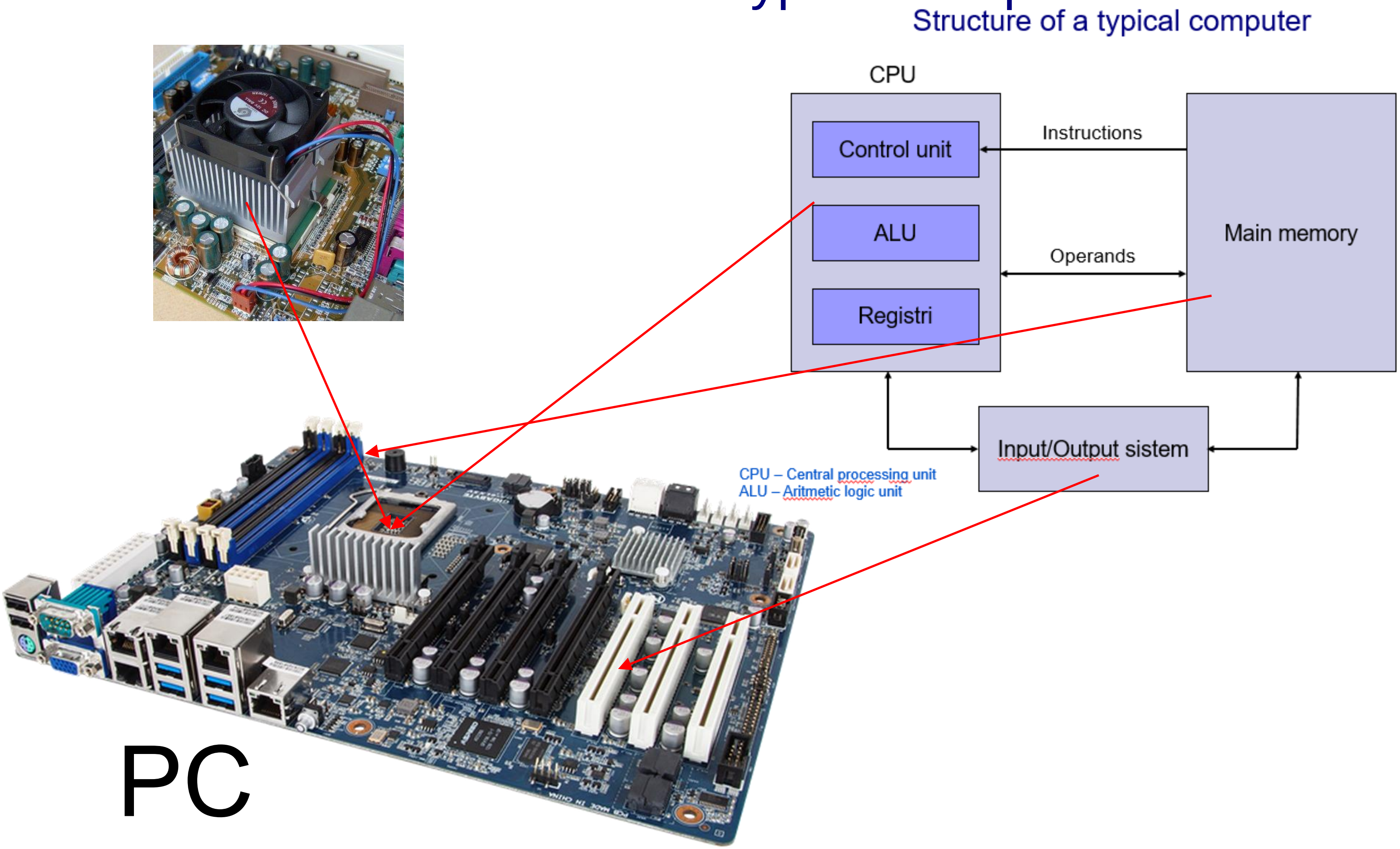
Structure of a typical computer





© 2022, Rozman, Škraba, FRI

Structure of a typical computer



 © 2022, Rozman, Škraba, FRI

Structure of a typical computer and addition of two numbers LAB -1

Python

1 STEV1=64

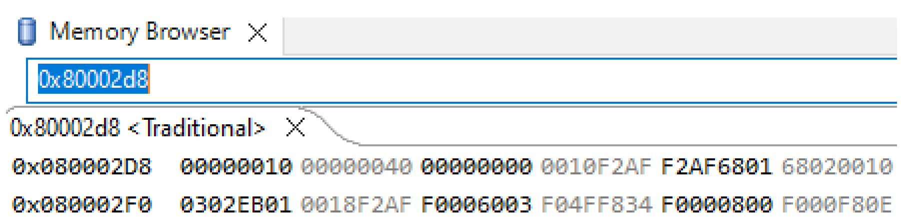
2 STEV2=16

3 REZ = STEV1 + STEV2

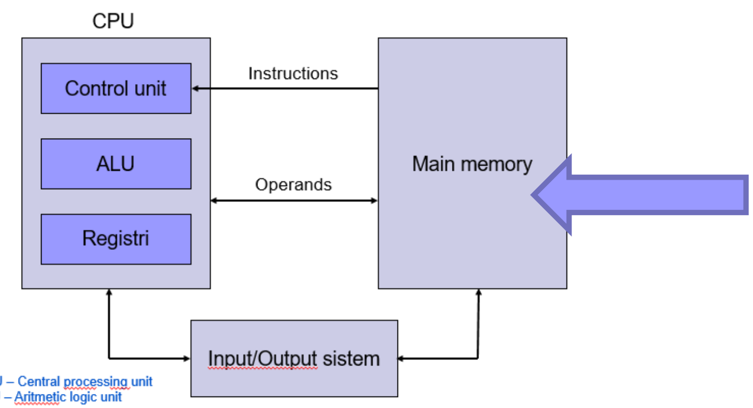
Assembler

```
STEV1: .word 0x10    // 32-bitna spr.
STEV2: .word 0x40    // 32-bitna spr.
VSOTA: .word 0       // 32-bitna spr.

 adr r0, STEV1    // Naslov od STEV1 -> r0
 ldr r1, [r0]     // Vsebina iz naslova v r0 -> r1

 adr r0, STEV2    // Naslov od STEV2 -> r0
 ldr r2, [r0]// Vsebina iz naslova v r0 -> r2

 add r3,r1,r2     // r1 + r2 -> r3

 adr r0,VSOTA// Naslov od VSOTA -> r0
 str r3,[r0]// iz registra r3 -> na naslov v r0
```

Machine language



Structure of a typical computer



CPU – Central processing unit
ALU – Aritmetic logic unit

© 2022, Rozman, Škraba, FRI

addition of two numbers(1. LAB session)

Python

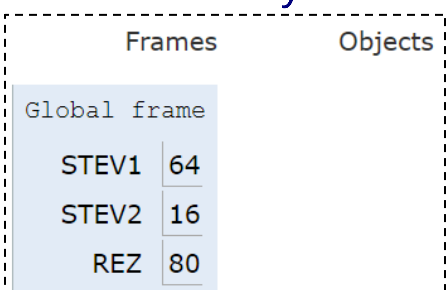
http://goo.gl/YXQ5qN

1 STEV1=64

2 STEV2=16

3 REZ = STEV1 + STEV2

Variables in memory



Assembler

```
STEV1: .word 0x10   // 32-bitna spr.
STEV2: .word 0x40   // 32-bitna spr.
VSOTA: .word 0      // 32-bitna spr.

 adr r0, STEV1    // Naslov od STEV1 -> r0
 ldr r1, [r0]     // Vsebina iz naslova v r0 -> r1

 adr r0, STEV2    // Naslov od STEV2 -> r0
 ldr r2, [r0]// Vsebina iz naslova v r0 -> r2

 add r3,r1,r2     // r1 + r2 -> r3

 adr r0,VSOTA// Naslov od VSOTA -> r0
 str r3,[r0]// iz registra r3 -> na naslov v r0
```

https://cpulator.01xz.net/?sys=arm&loadasm=share/sBe6EPC.s

© 2022, Rozman, Škraba, FRI

Python (case: REZ = STEV1 + STEV2)

Adding variables in Python.

http://goo.gl/YXQ5qN

Python 2.7

```
STEV1=0x40
STEV2=0x10
REZ = STEV1 + STEV2
print (" STEV1 = " + hex(STEV1) + "\n+STEV2 = " + hex(STE
```

Frames

Objects

Global frame

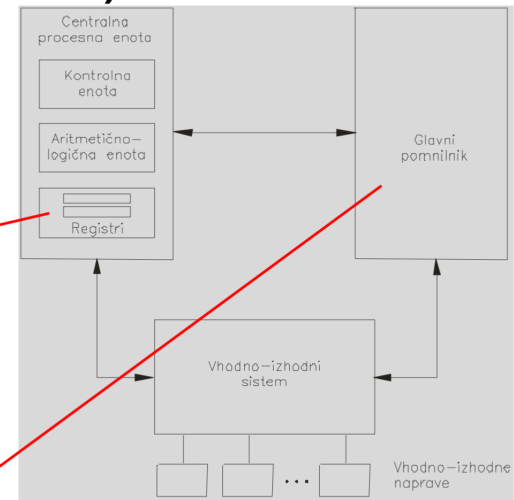
STEV1	64
STEV2	16
REZ	80

Print output (drag lower right corner to resize)

```
 STEV1 = 0x40
+STEV2 = 0x10
-------------
   REZ = 0x50
```

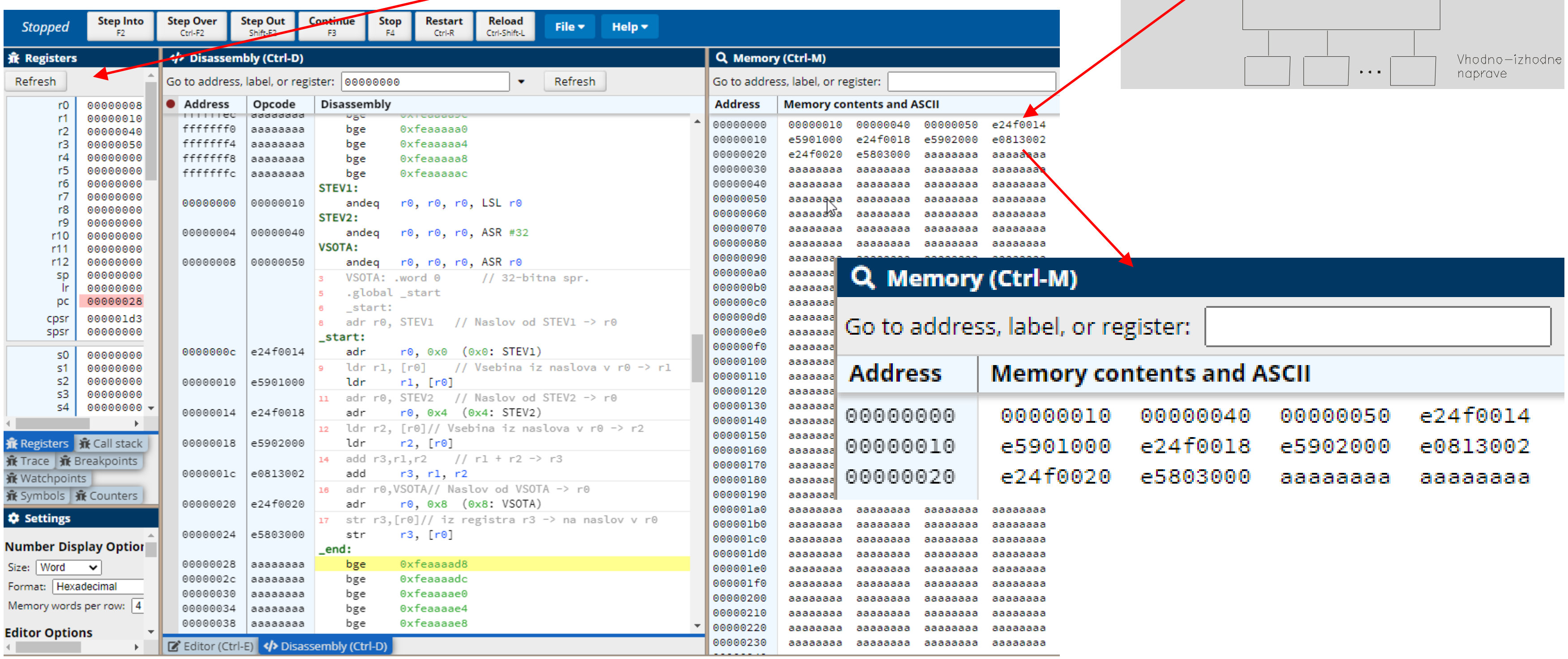

© 2022, Rozman, Škraba, FRI

Assembler (case: REZ = STEV1 + STEV2)



CPULator – online simulator

[CPULator]()



1. Introduction :

- 1.1 CA course
- 1.2 About computers
- 1.3 Basic computer structure and operation
- 1.4 Analog – digital, continuous-discrete
- 1.5 8 important ideas in computer architecture (and in general)
- 1.6 Computer realization

© 2022, Rozman, Škraba, FRI

Comparisons:

Analog – digital

Continuous – discrete

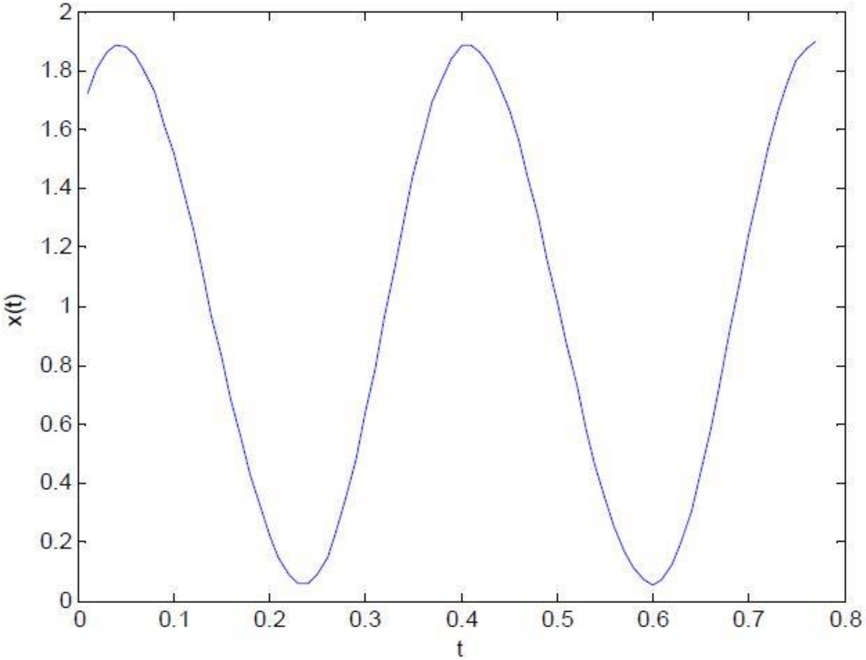
© 2022, Rozman, Škraba, FRI

Analog –
- continuous representation

Digital –
- discrete representation

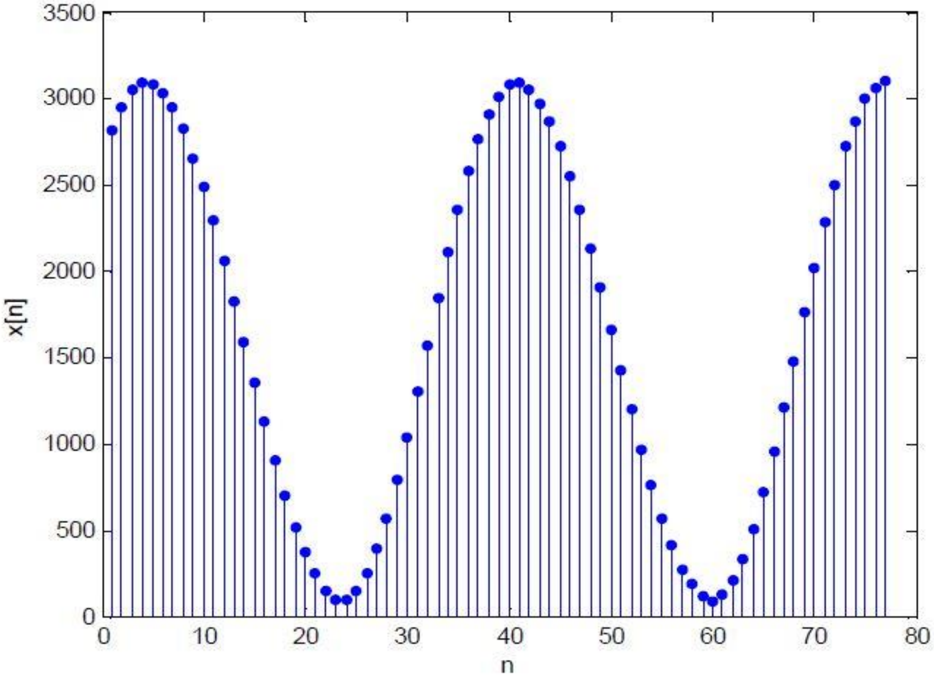
© 2022, Rozman, Škraba, FRI

Analog –
- continuous representation



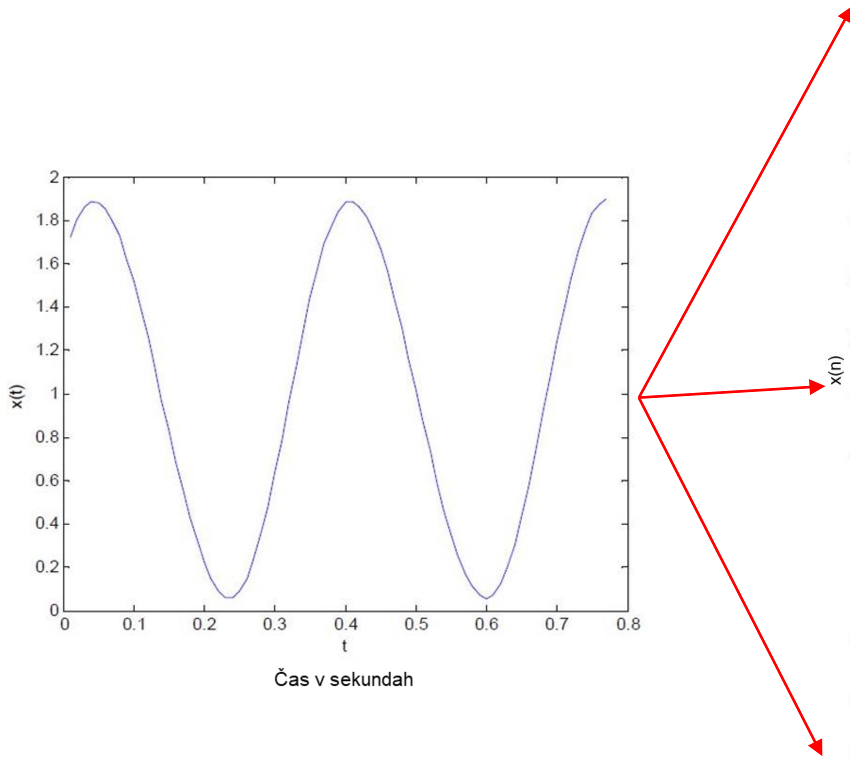
Time in seconds

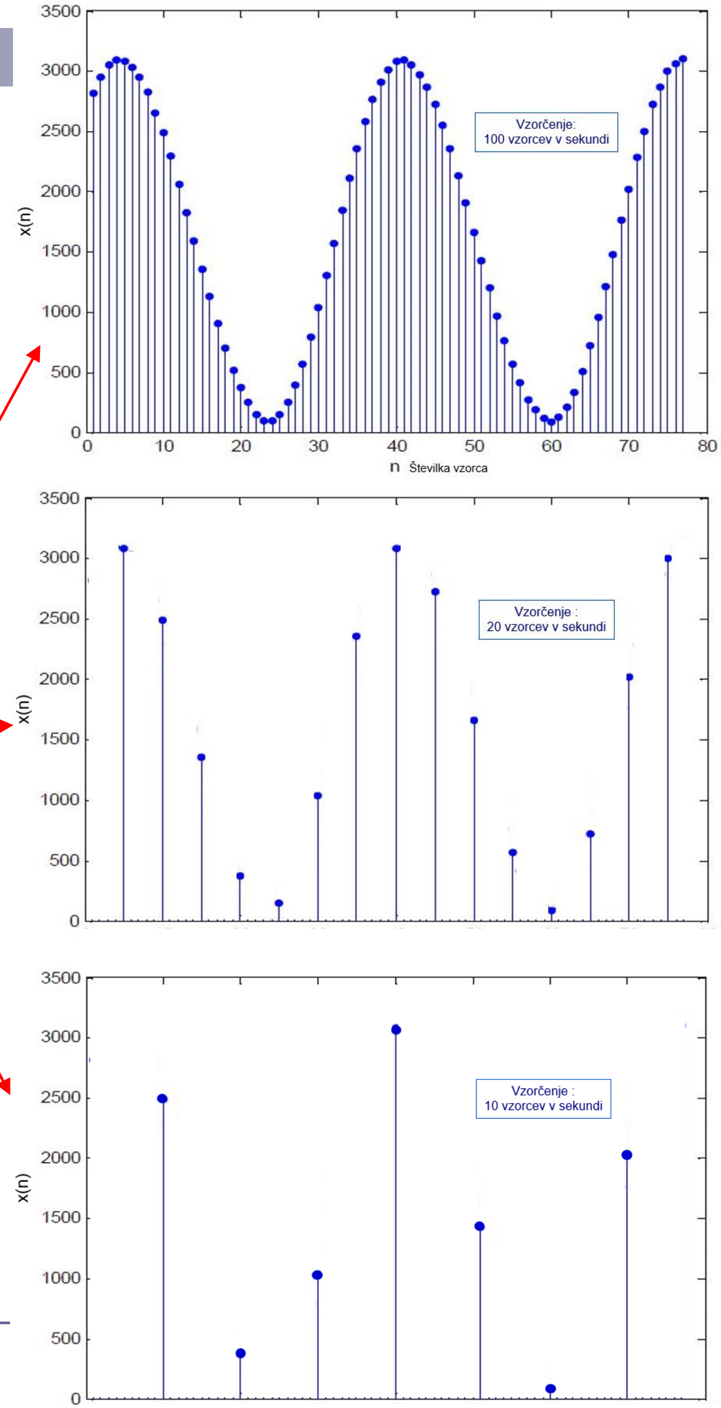
Digital –
- discrete representation



Sample number
100 samples per second

Case 2: Signal sampling





100 samples/sec

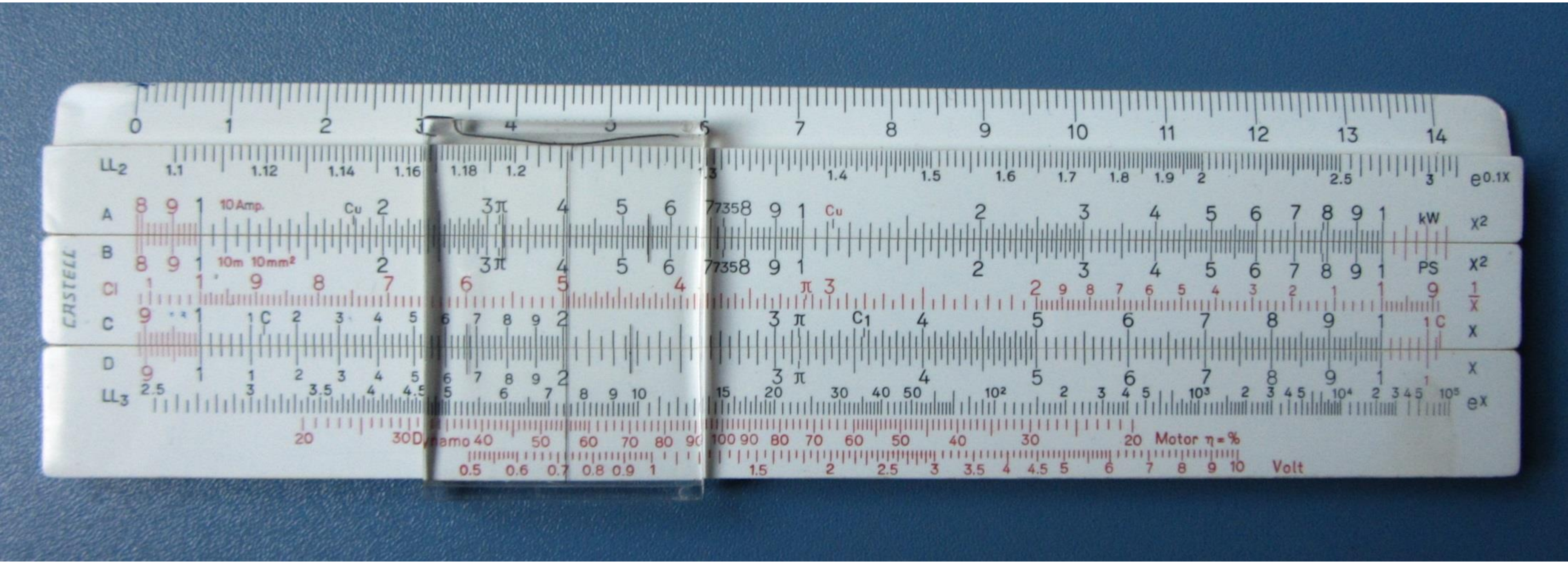
20 samples/sec

10 samples/sec

Analog computing – continuous presentation of numbers

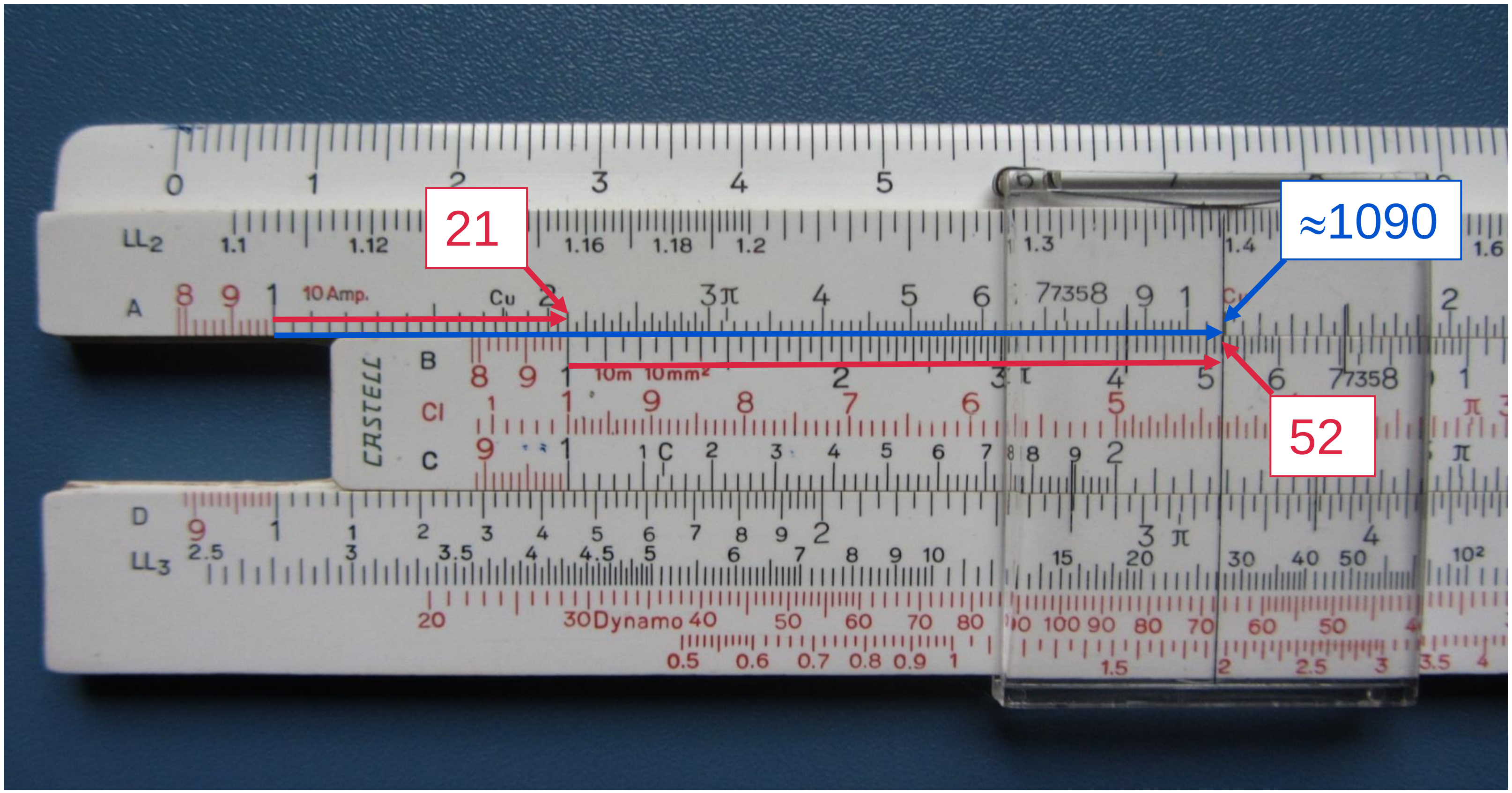
Digital computing – discrete presentation of numbers

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



© 2022, Rozman, Škraba, FRI

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



21 x 52 ≈ 1090
Measured result

21 x 52 = 1092
Exact result

© 2022, Rozman, Škraba, FRI

- using continuous Voltage ⇒ Analog amplifier and computer

 © 2022, Rozman, Škraba, FRI

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

 © 2022, Rozman, Škraba, FRI

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 a binary numeral system

© 2022, Rozman, Škraba, FRI

1. Introduction :

- 1.1 CA course
- 1.2 About computers
- 1.3 Basic computer structure and operation
- 1.4 Analog – digital, continuous-discrete
- 1.5 8 important ideas in computer architecture (and in general)
- 1.6 Computer realization

© 2022, Rozman, Škraba, FRI

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 than just do nothing - wait"
- Memory hierarchy
  - Compromise between memory speed and cost
- Reliability with redundancy
  - Cost of the backup system may be lower than the cost of failure

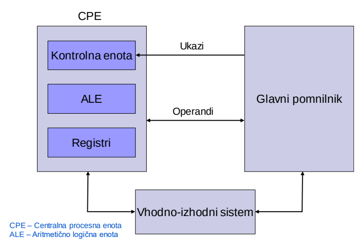
 © 2022, Rozman, Škraba, FRI

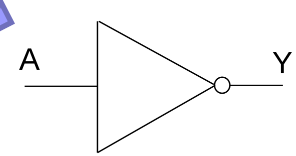
1. Introduction :

- 1.1 CA course
- 1.2 About computers
- 1.3 Basic computer structure and operation
- 1.4 Analog – digital, continuous-discrete
- 1.5 8 important ideas in computer architecture (and in general)
- 1.6 Computer realization

© 2022, Rozman, Škraba, FRI

Theoretical model <-> Practical realization





Simbol

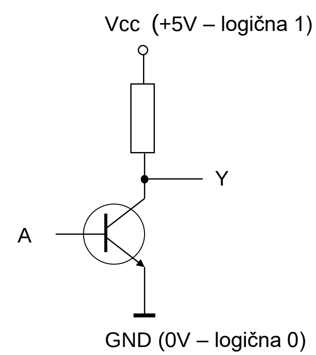
mathematical (logical) view: logic gate

logical level 0,1

Mathematical ideal

---

electrical realization : electrical circuit



voltage levels ≈0V, ≈3.3 (5) V

Electrical realization

Cons :

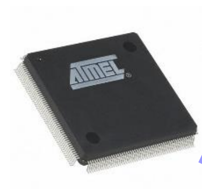
- *analogue voltagesi*
- *time-delays*
- *noises*

© 2022, Rozman, Škraba, FRI

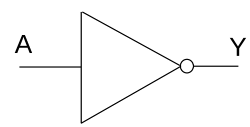
Physical structure of the computer

Computer

Circuits (Chips)



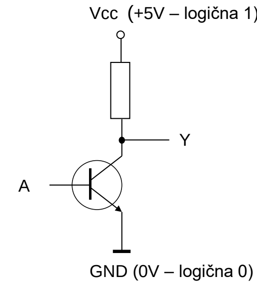
Logic gate (function)



Simbol

Case: negation

Elektronic implementation



Case: negation

© 2022, Rozman, Škraba, FRI

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 (up to +5V)

- Simple realization with a switch - example:
  - **State 0** – switch open – low voltage
  - **State 1** – switch closed – high voltage

© 2022, Rozman, Škraba, FRI

- One switch can be in two states, state 0 or 1.
- Such a 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 than only 1 bit of information, we need more cells.

© 2022, Rozman, Škraba, FRI

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

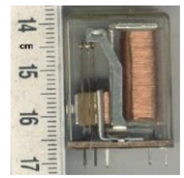
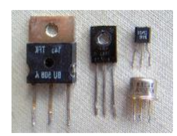
© 2022, Rozman, Škraba, FRI

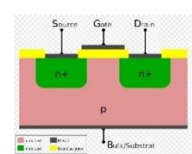
Prefixes for units of measurement

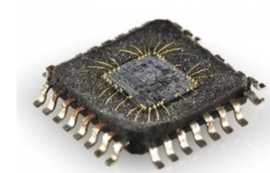
Abbreviation	Name	Value	Exponent (scientific notation)
p	pico	0,000 000 000 001	10^{-12}
n	nano	0,000 000 001	10^{-9}
µ	micro	0,000 001	10^{-6}
m	milli	0,001	10^{-3}
K	kilo	1 000	10^{3}
M	mega	1 000 000	10^{6}
G	giga	1 000 000 000	10^{9}
T	tera	1 000 000 000 000	10^{12}

 © 2022, Rozman, Škraba, FRI

Realization of switches as the basic building block - summary:

- Electro-mechanical 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

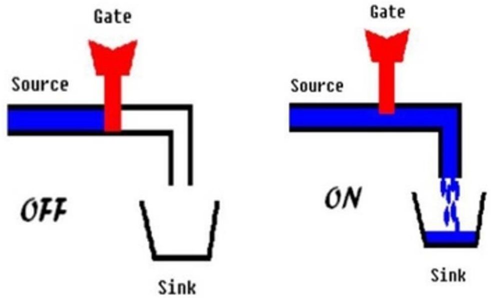


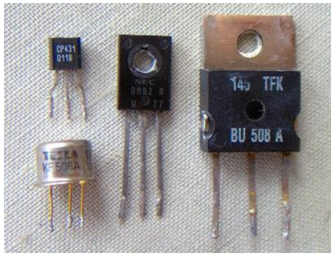


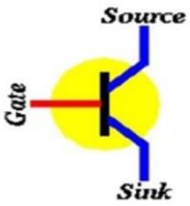
© 2022, Rozman, Škraba, FRI

Transistor can be used as :

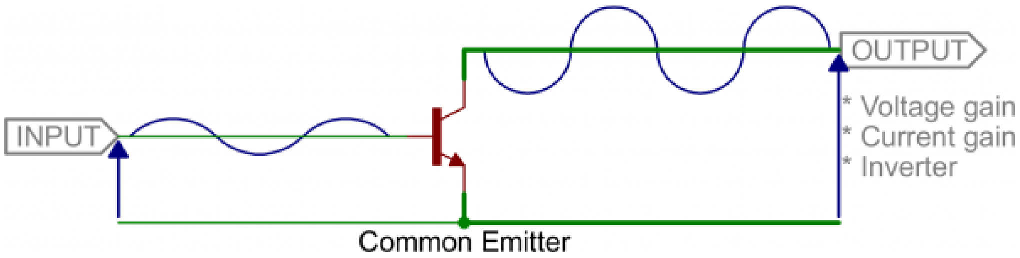
- Switch
  - In digital circuits





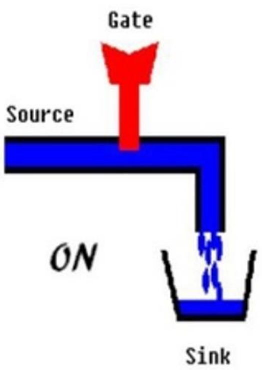


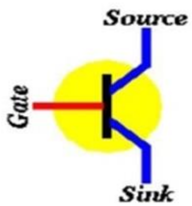
- Signal amplifier
  - Electronic circuits (amplifiers)

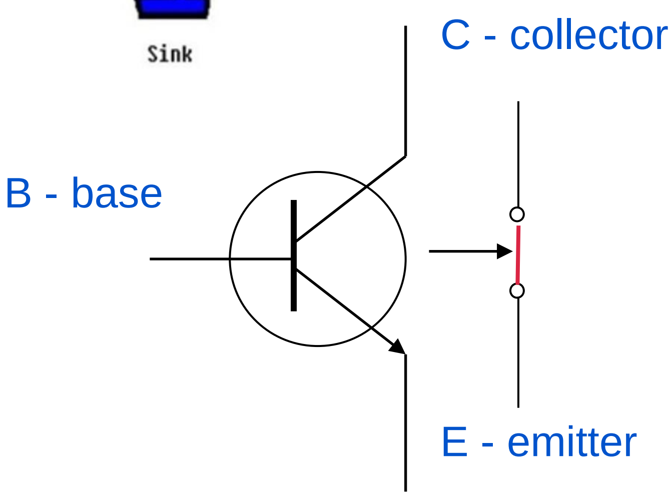


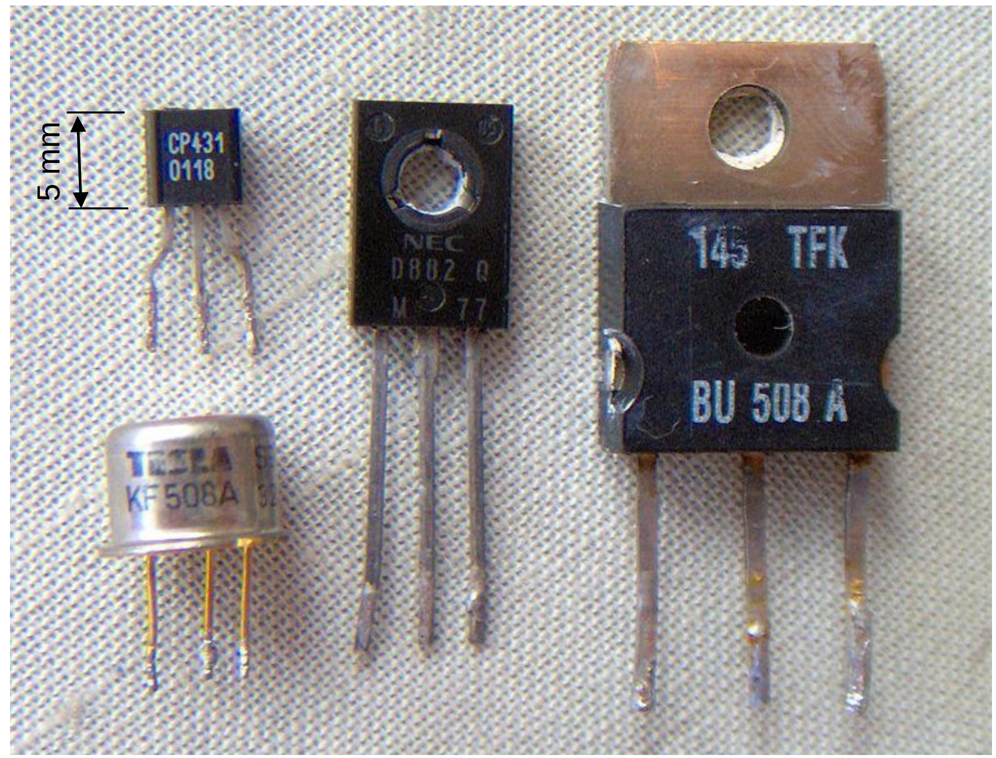
 © 2022, Rozman, Škraba, FRI

- Transistor as switch - ON



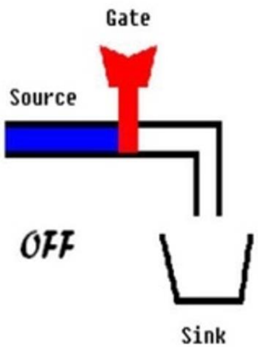


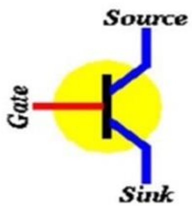


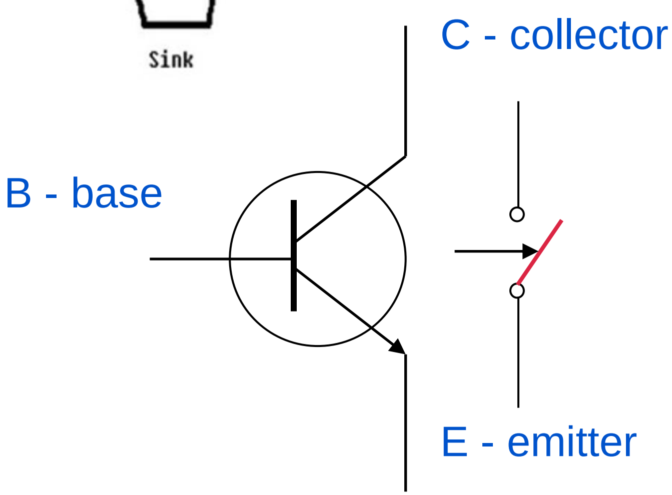


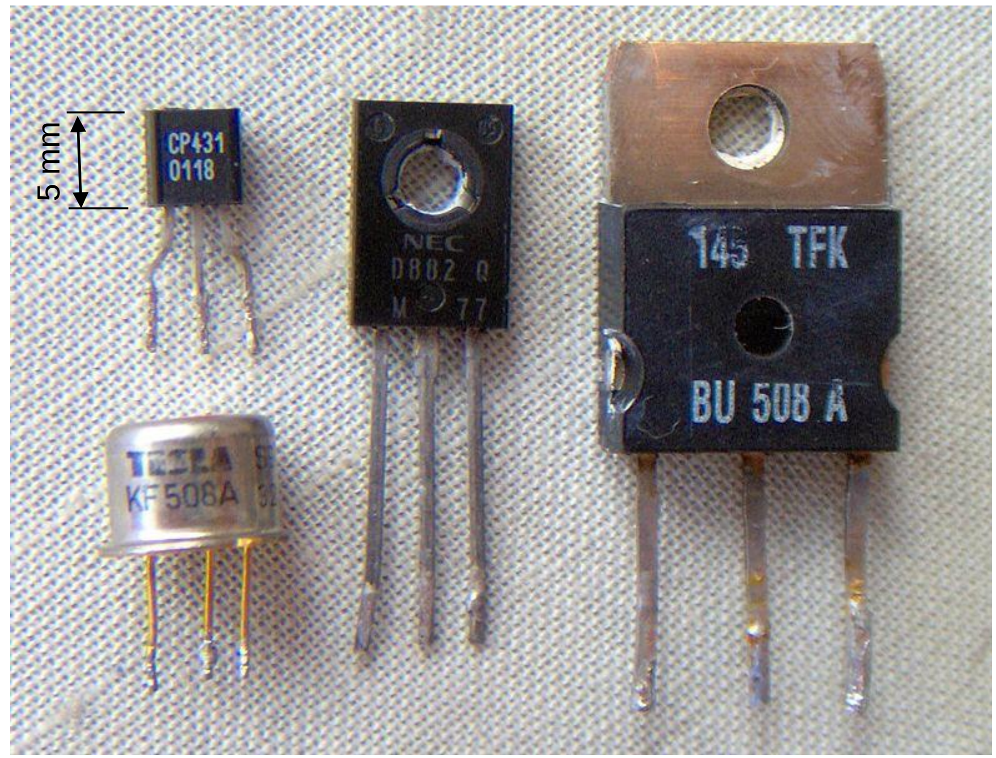
© 2022, Rozman, Škraba, FRI

- Transistor as switch - OFF



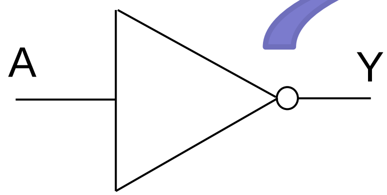






© 2022, Rozman, Škraba, FRI

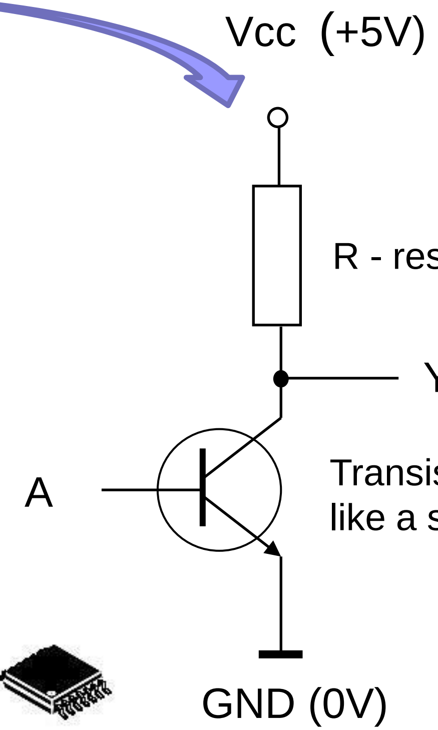
Realization of the logical function NEGATION (NOT)



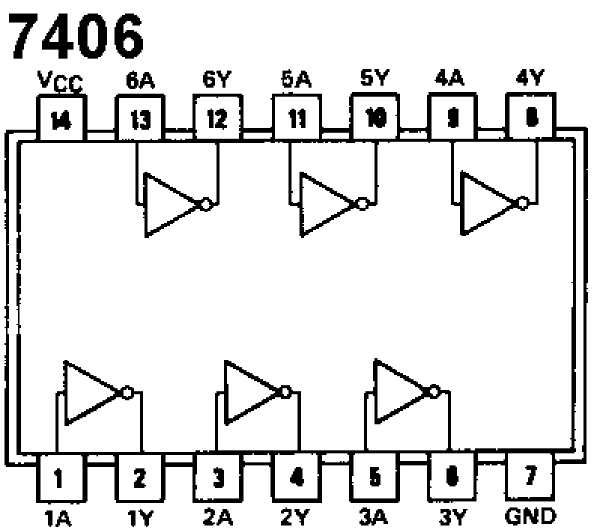
Symbol

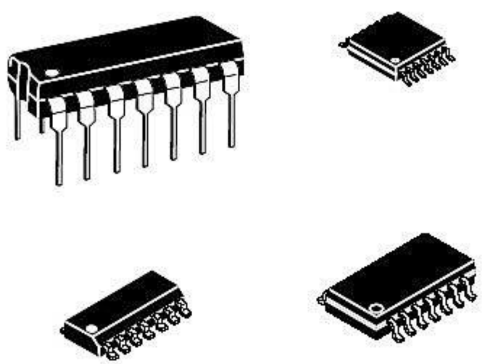
A	Y
0	1
1	0

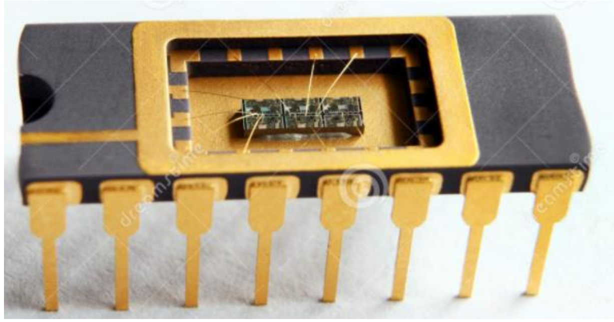
Truth table



IC (Integrated Circuit) with 6 negators

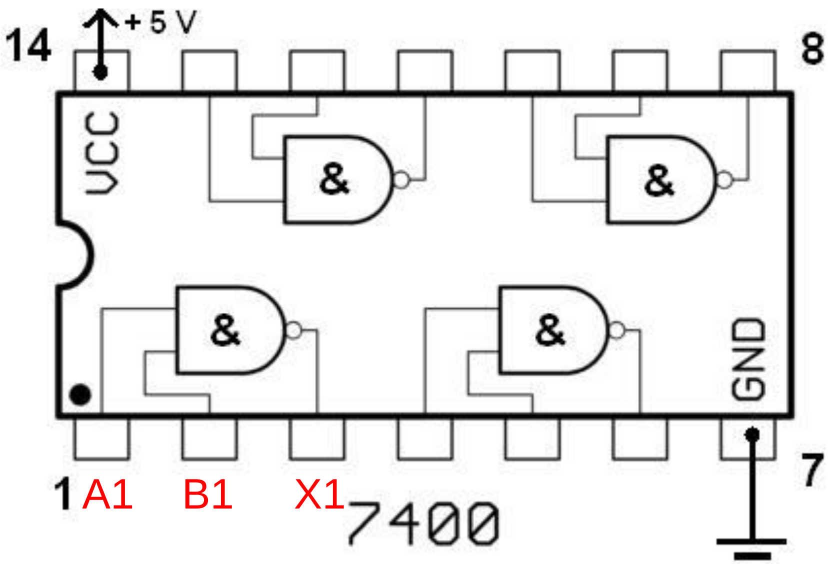


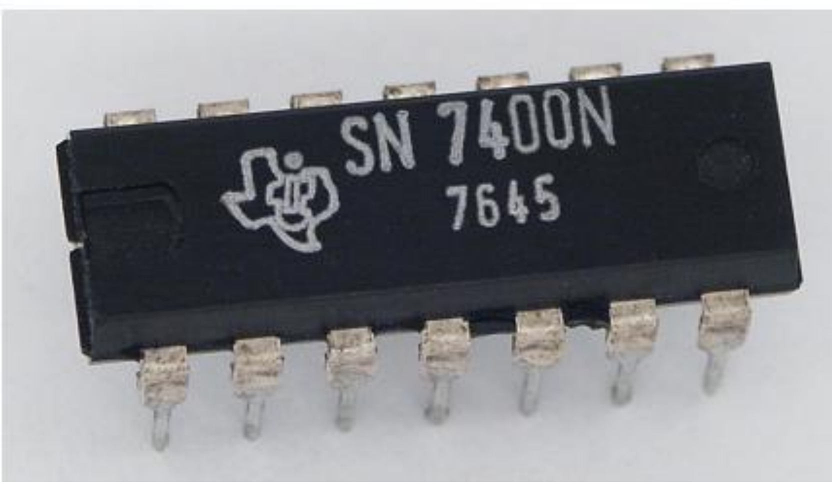


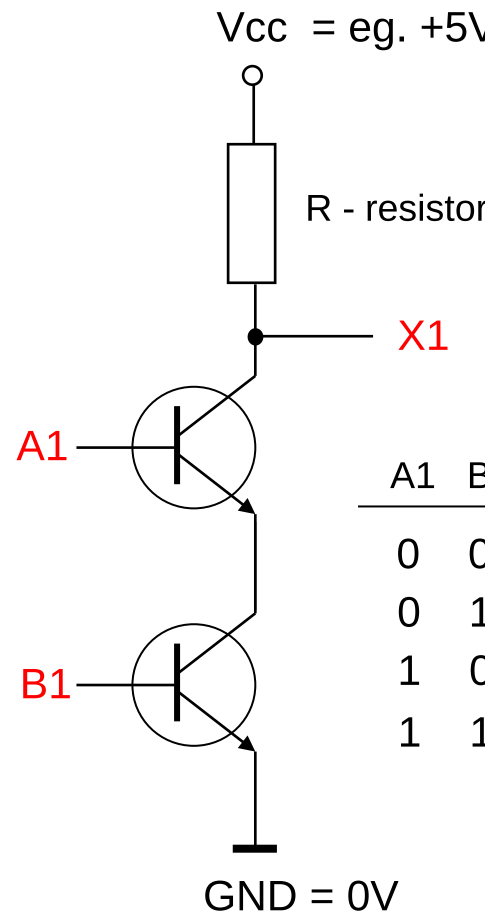


© 2022, Rozman, Škraba, FRI

Realization of the logical function NAND (Negated conjunction)



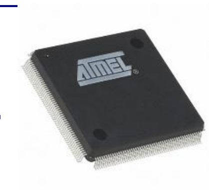


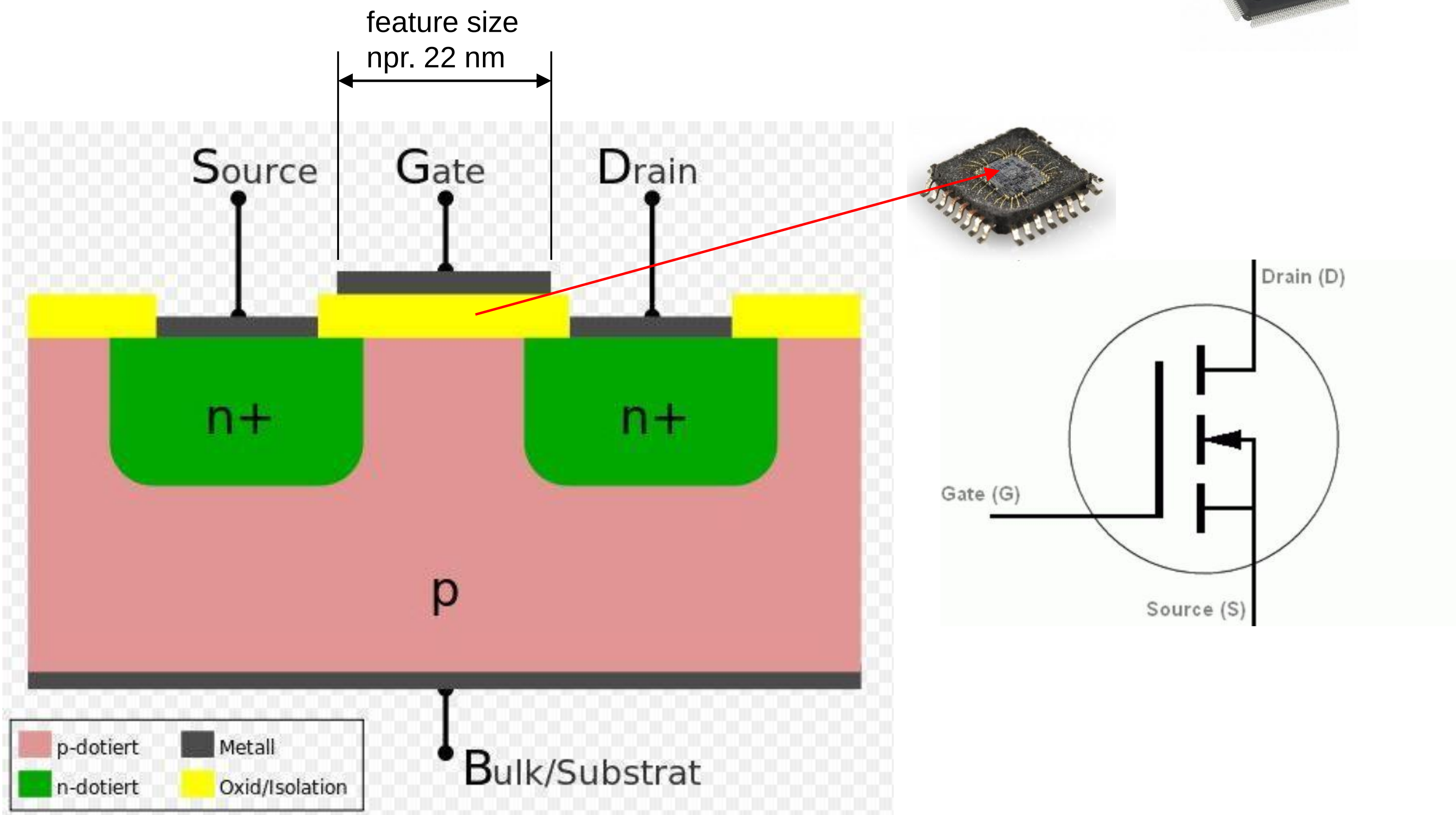


A1	B1	X1
0	0	1
0	1	1
1	0	1
1	1	0

© 2022, Rozman, Škraba, FRI

Transistors as a part of the integrated circuit



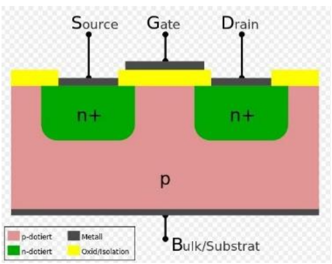


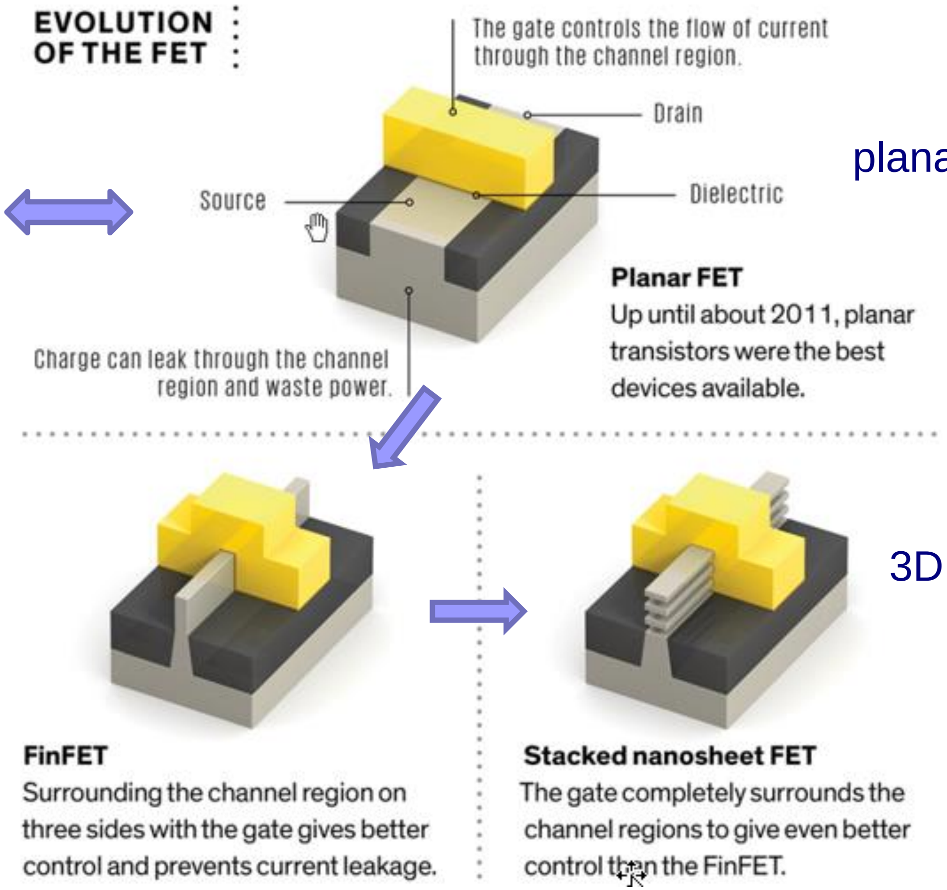
 © 2022, Rozman, Škraba, FRI

Introduction

Transistors evolution in modern circuits:

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



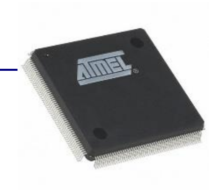


planar 2D (horizontal)

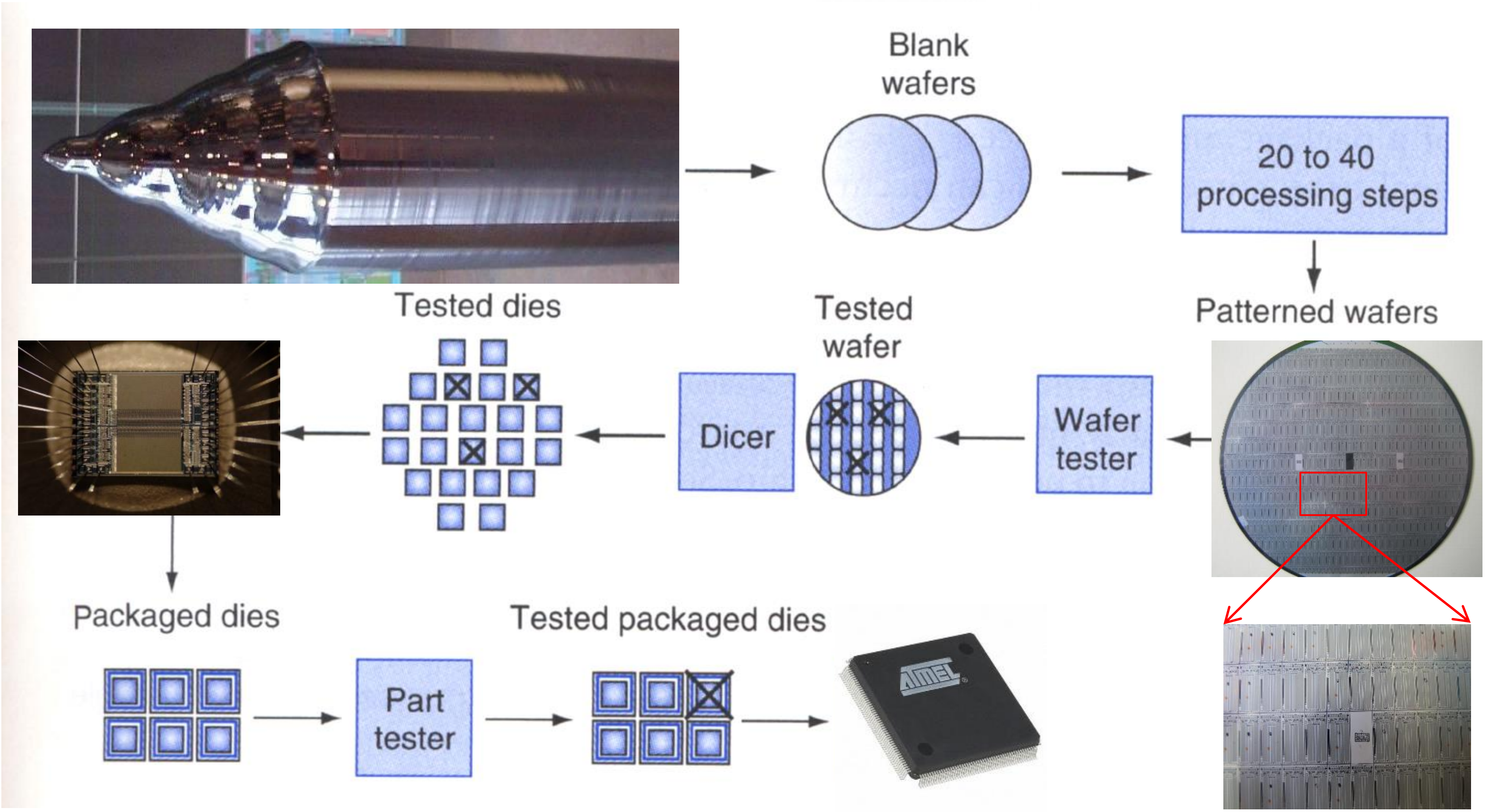
3D transistor (vertical)

Si atom's diameter is 0.24nm!!!

© 2022, Rozman, Škraba, FRI



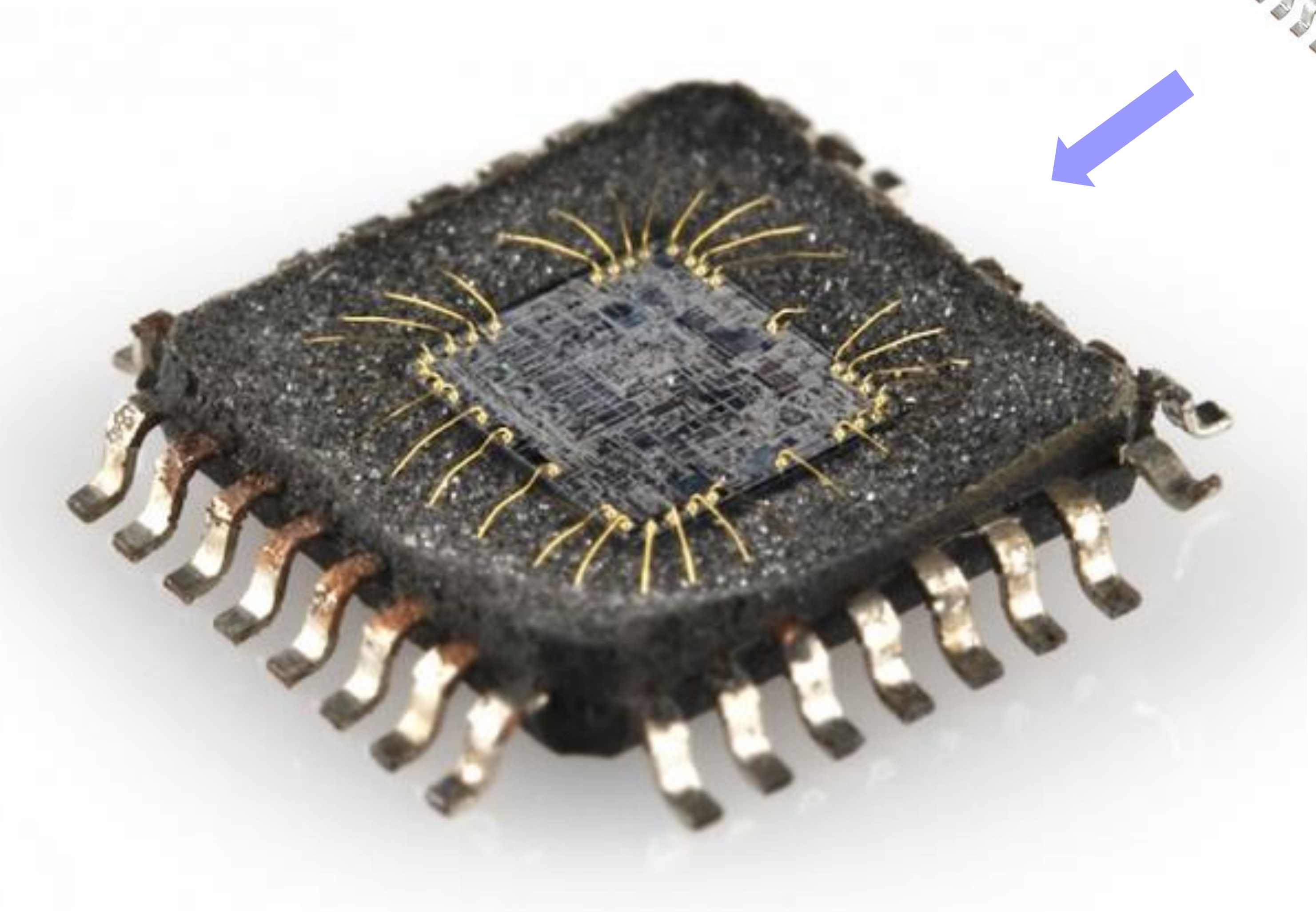
The process of making a VLSI chip

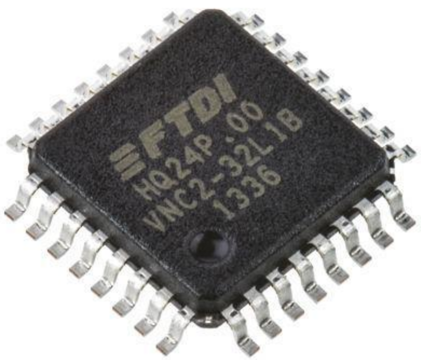


David A. Patterson, John L. Hennesy:
Computer Organization and Design, Fourth Edition

© 2022, Rozman, Škraba, FRI

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

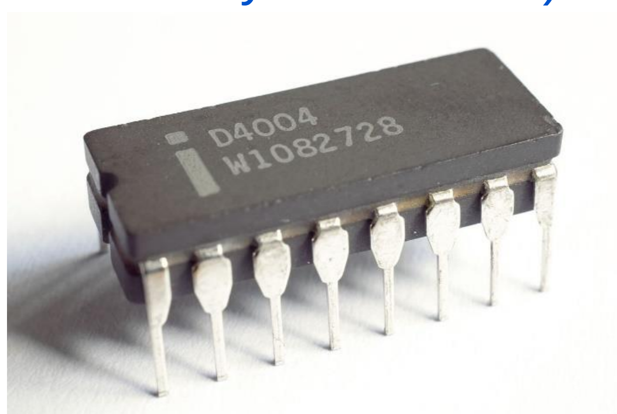
© 2022, Rozman, Škraba, FRI

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

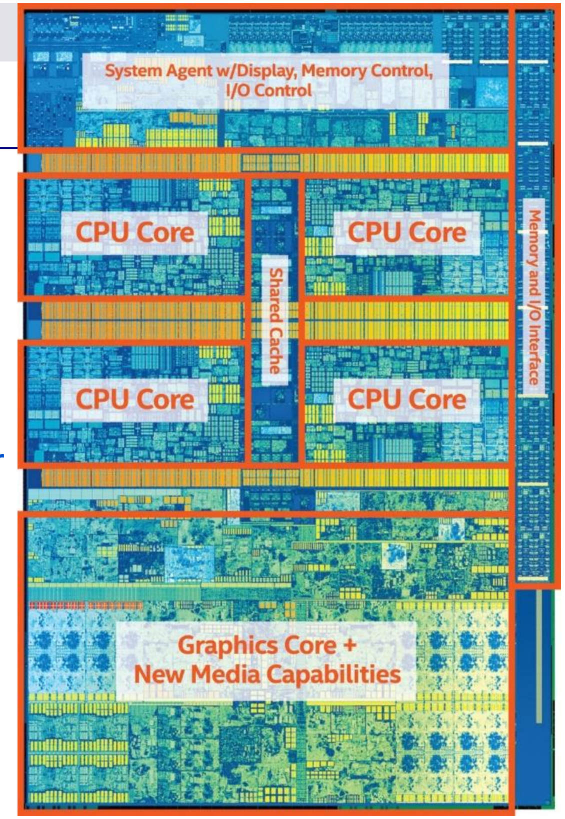
© 2022, Rozman, Škraba, FRI

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 = 10x10^{-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



 © 2022, Rozman, Škraba, FRI

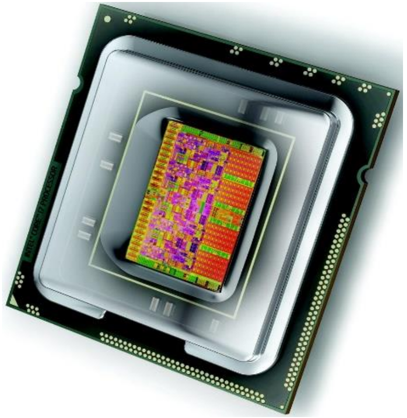


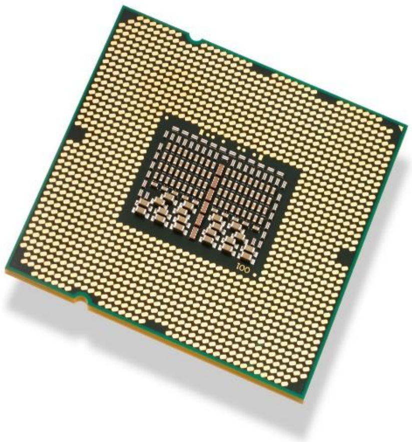
Case 2:

- **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 $





© 2022, Rozman, Škraba, FRI

Case 3:

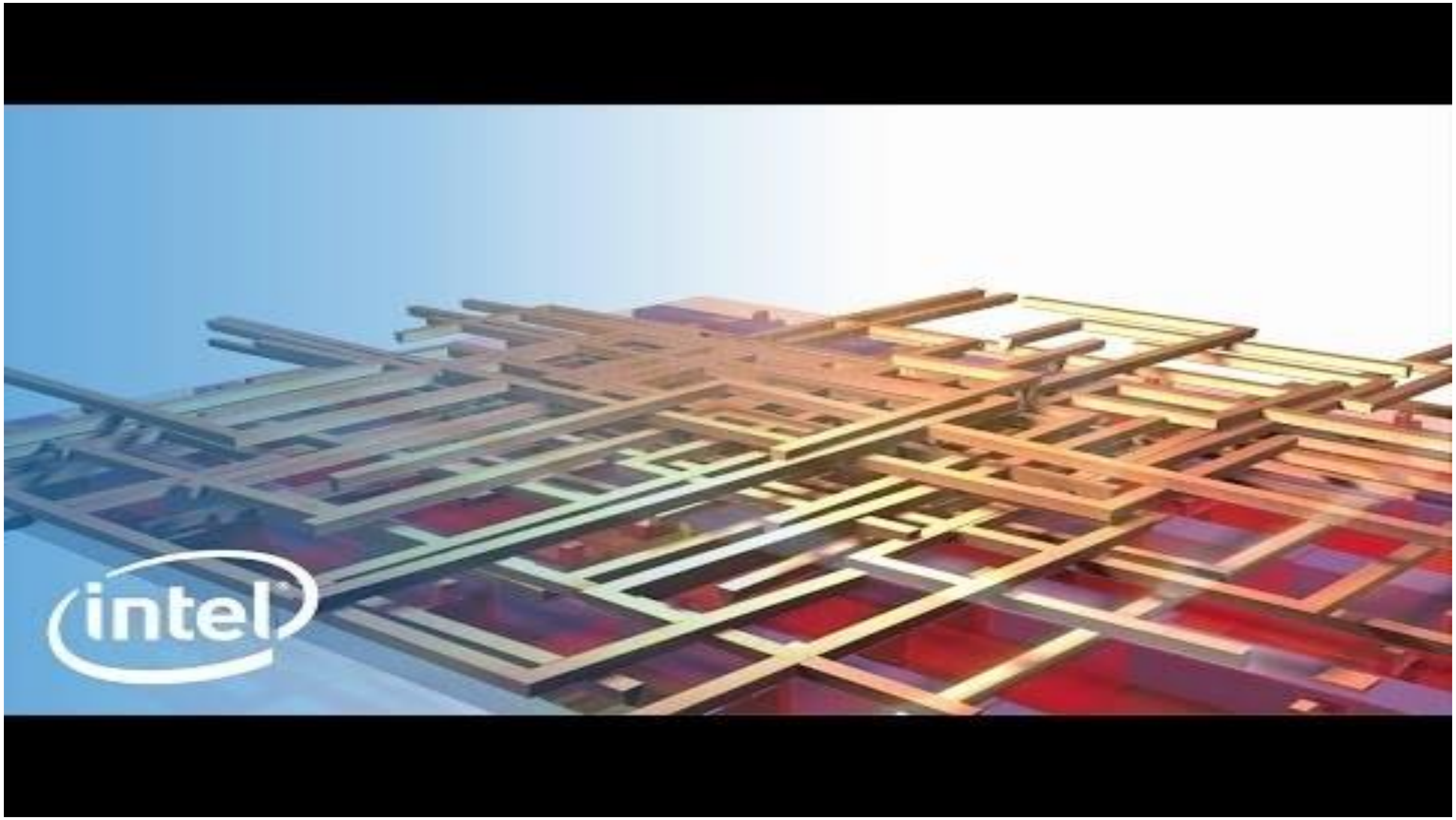
- Processor Intel i9-11900

  (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 $

 © 2022, Rozman, Škraba, FRI

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



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

© 2022, Rozman, Škraba, FRI