



RAČUNALNIŠKA ARHITEKTURA

1 Uvod

1. Uvod :

- ❑ 1.1 Predmet RA
- ❑ 1.2 Računalniki včeraj in danes
- ❑ 1.3 Osnove zgradbe in delovanja računalnikov
- ❑ 1.4 Analogno – digitalno, zvezno diskretno
- ❑ 1.5 8 pomembnih idej v računalniški arhitekturi (in širše)
- ❑ 1.6 Praktična realizacija računalnikov

1. Uvod :

- 1.1 Predmet RA
- 1.2 Računalniki včeraj in danes
- 1.3 Osnove zgradbe in delovanja računalnikov
- 1.4 Analogno – digitalno, zvezno diskretno
- 1.5 8 pomembnih idej v računalniški arhitekturi (in širše)
- 1.6 Praktična realizacija računalnikov

Računalniška arhitektura – 1.1 Predmet RA

- Spletne strani: <http://ucilnica.fri.uni-lj.si>
<https://padlet.com/rawall/RAWall>
- MS Teams
 - Koda za vstop: 277pdvh
- Govorilne ure: Po razporedu v R2.40,50 ali MS Teams (na daljavo)
Občasne spremembe bodo pravočasno objavljene na učilnici. Najavite se po emailu.



Ekipa RA

Asistenta

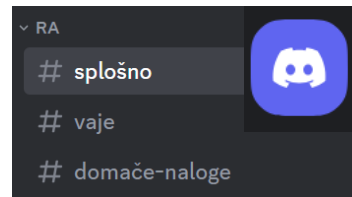


Žiga Pušnik
ziga.pusnik@fri...



Romanela Lajić
romanela.lajic@fr...

Tutorji



<https://discord.gg/nmzjQU7me7>



Robert Rozman
rozman@fri.uni-lj.si



■ Literatura (knjige na voljo tudi v knjižnici FRI)::

- Vsebina predavanj, lab. vaj in prosojnic

- <http://ucilnica.fri.uni-lj.si>

- MS Teams (komunikacija, zapiski s table)

- Skupni (deljeni) zapiski – Gdocs

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

 Computer Architecture - Crowd-sourced Shared Notes

 Računalniška Arhitektura - Deljeni zapiski za skupno dopolnjevanje

- Osnovna, bolj obširna:

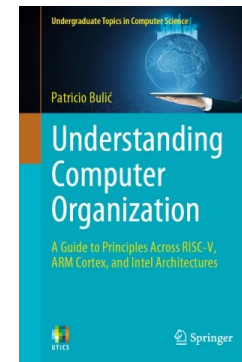
- Dušan Kodek: ARHITEKTURA IN ORGANIZACIJA RAČUNALNIŠKIH SISTEMOV, Bi-TIM, 2008

- Dodatna (samo določeni deli – 5 izvodov v knjižnici FRI):

- P. Bulić: Understanding Computer Organization

- Dodatna (samo določeni deli – 4 izvodi v knjižnici FRI):

- Andrew S. Tanenbaum: STRUCTURED COMPUTER ORGANIZATION, Sixth Edition Pearson Prentice Hall, 2013



Izhodišča :

- Ni neumnih vprašanj, so samo tisti, ki ne vprašajo
- Vedno dobrodošli
- Vsi se trudimo

Povpr.ocena/max-[št.odg./vsj]†	2022/23-Nosilec†	2021/22-Nosilec†	2020/21-Nosilec†	2019/20-Nosilec†	2018/19-Nosilec†
Predmet†	4.56/5-[173/184]†	4.53/5-[192/196]†	4.65/5-[154/161]†	4.54/5-[149/150]†	4.66/5-[138/164]†
Izvajalec†	4.72/5-[173/184]†	4.75/5-[192/196]†	4.74/5-[154/161]†	4.79/5-[149/150]†	4.77/5-[138/164]†

Mnenja slušateljev (2018/23) - izbor:

- Dobro:
 - Kahoot! Veliko sem odnesel od kahoota ter RA walla z vprašanji.
 - OneNote zvezki z zapiski, uporabne in zanimive vaje, Kahooti, dobri pdfji
 - Na predavanjih so bila velikokrat omenjena aktualna vprašanja ali novice, spodbuja se kritično in samostojno razmišljanje
 - Velik poudarek da na razumevanje, in ne na učenje na pamet
 - Energija predavatelja, praktični (življenjski) primeri
 - Good learning system for foreign students
 - ... asistentom in še posebej profesorju se vidi da jim ni vseeno za naše znanje. Zaradi tega se pozna kvaliteta predavanj in vaj.
- Izboljšati, pojasniti:
 - Teorija kdaj postane težko razumljiva, težko je slediti novim pojmom in idejam.
 - Snov predavanja <> vaje
 - „Ni popoln, kar je za vsakega značilno“

Izhodišča 2024:

- Živa izvedba predavanj in vaj

- <https://padlet.com/rawall/RAWall>, Slido
 - Vprašanja, izzivi, povezave, ...

- Orodja :

- e-učilnica <http://ucilnica.fri.uni-lj.si>
- MS Teams (tabla, komunikacija)



- Izhodišča:

- karseda aktivno (v živo)
- sodelovanje, pogovor, vprašanja, komentarji, ...
- nova platforma STM32H750-DK – dve praktični LAB vaji

What's new in 2023:

- Live lectures and lab sessions (one lab session in English – **day ?**)

- <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 : **y331dds**

- Important :

- be active

- cooperate, talk, ask, comment, ...

- all major documents are translated to English

- occasional testing of realtime Slovene-English translation – project ON

- please help us on English documents (typos, missing content, ...)



<https://dis-slovarcek.ijs.si/>

Na Univerzi v Ljubljani letos študira dobrih 40 tisoč študentov. Devet odstotkov jih prihaja iz tujine.

What's new in 2024:

This year, more than 40,000 students are studying at the University of Ljubljana. Nine percent of them come from abroad.

■ Live lecture recognition to Slovene and translation to English text

- testing of real-time Slovene-English translation – [project ON](#)

■ Access link :

- <https://on.uni-lj.si>
- using your official student e-mail identity (*@student.uni-lj.si).

occasional testing of realtime Slovene-English translation – [project ON](#)

■ Please fill introductory survey :

- <https://1ka.arnes.si/ON-2022>

■ Further instructions will follow

■ Contact:

- tjasa.jelovsek@fri.uni-lj.si

ONLINE NOTES (ON)

RAZVOJ SISTEMA ZA AVTOMATIZIRANO PREVAJANJE SLOVENSКИH PREDAVANJ V TUJE JEZIKE

Projekt **Online Notes (ON)** je namenjen razvoju sistema za avtomatizirano prevajanje slovenskih predavanj v tuje jezike. Razvoj poteka pod vodstvom Fakultete za računalništvo in informatiko Univerze v Ljubljani v sodelovanju s Centrom za jezikovne vire in tehnologije Univerze v Ljubljani. Cilj projekta je povečati dostopnost predavanj za tuje študente doma in v tujini ter za ciljne skupine s senzornimi oviranostmi.

Rezultati projekta bodo koristni za več skupin: prevodi v živo bodo tujim študentom olajšali spremljanje predavanj, ki potekajo v slovenskem jeziku; prepisi v slovenskem jeziku bodo povečali dostopnost gluhih in naglušnih; izvlečki in posnetki predavanj, ki bodo dosegljivi preko posebnega portala, pa bodo odlično dopolnjevali elektronska gradiva za študij.

<https://www.cjvt.si/infrastruktura-podpora/online-notes/>




Računalniška arhitektura

[Predmet](#) [Nastavitve](#) [Sodelujoči](#) [Ocene](#) [Poročila](#) [Več](#) ▾

▼ Splošno - General

 [Anketa o predmetu Računalniška arhitektura \(2023/24\)](#)

 Skrito za udeležence

 [Splošne informacije - General info](#)

MS Teams ekipe: RA VSP 2024/25 - izvedba predmeta (koda: 277pdvh), LAPSy Embedded Academy - stalni FRI HW portal (koda: ty5qjm9)

 [Discord Hardware FRI - skupnost za HW predmete](#)

 [Forum novic - News Forum](#)

 [Forum - RA - vprašanja in odgovori - Q&A](#)

MS Teams: Komunikacija, OneNote zvezek

The screenshot displays the Microsoft OneNote application interface. The top bar includes a search bar with the text "Iskanje (Ctrl+E)" and window control buttons. The left sidebar shows the "Vse skupine" (All Teams) section with a list of teams, including "RA VSP 2024/25". The main content area is titled "Class Notebook" and features a ribbon menu with tabs for "Datoteka", "Osnovno", "Vstavljanje", "Risanje", "Ogled", and "Pomoč". The "Osnovno" tab is active, showing a toolbar with various editing tools. The notebook content is organized into sections and pages. The current page is titled "Kako biti uspešen?" and contains handwritten notes in green, blue, and black ink. The notes include:

- **RADOVEDNOST**
- **RAZUMEVANJE = ZNANJE**
- **AKTIVNO SODELOVANJE**

The notes are accompanied by a large blue bracket on the right side and a double-headed arrow below the third bullet point. The bottom of the page shows a cursor and a horizontal line.

Vsebina po poglavjih

<https://padlet.com/rawall/RAWall>

The screenshot shows a Padlet board with the following content:

- Stalni viri:**
 - RAWall 3yr: Padlet lahko uporabite za svoje potrebe. S prijavo spodaj nam prinesete free padlet.
 - RRobi 3yr: FRI E-učilnica. Računalniška arhitektura.
 - RRobi 3yr: MS Teams. Team code: y331dds.
- Vsebina:**
 - RRobi 3yr: 1.Uvod :
 - 1.1 Predmet RA
 - 1.2 Računalniki včeraj in danes
 - 1.3 Osnove zgradbe in delovanja računalnikov
 - 1.4 Analogno – digitalno, zvezno diskretno
 - 1.5 8 pomembnih idej v računalniški arhitekturi (in širše)
 - 1.6 Praktična realizacija računalnikov
- Moja vprašanja:**
 - RAWall 1yr: Občutki ?
 - RRobi 2yr: VEGA HPC na top500.org
 - RRobi 3yr: Dewesoft & Atlantis
 - RRobi 3yr: Katero srednjo šolo ste končali ? (klik na sliko)
- Viri:**
 - RRobi 2yr: VEGA HPC na top500.org. The List. VEGA HPC GPU - BullSequana XH2000, AMD EPYC 7H12 64C 2.6GHz, NVIDIA A100, Infiniband HDR.
 - RRobi 3yr: Dewesoft & Atlantis. Space shuttle Atlantis launch monitoring with Dewesoft X software.
- Vprašanja, komentarji:**
 - RRobi 3yr: Premik vprašanj in odgovorov. FRI. Pomembnejša vprašanja in odgovore premikam na Wiki stran predmeta :

Tedenska vsebina – 2024/25

Možnost anonimnih vprašanj, aktivnega sodelovanja, izrekanje mnenj

Join at
slido.com
#RA1

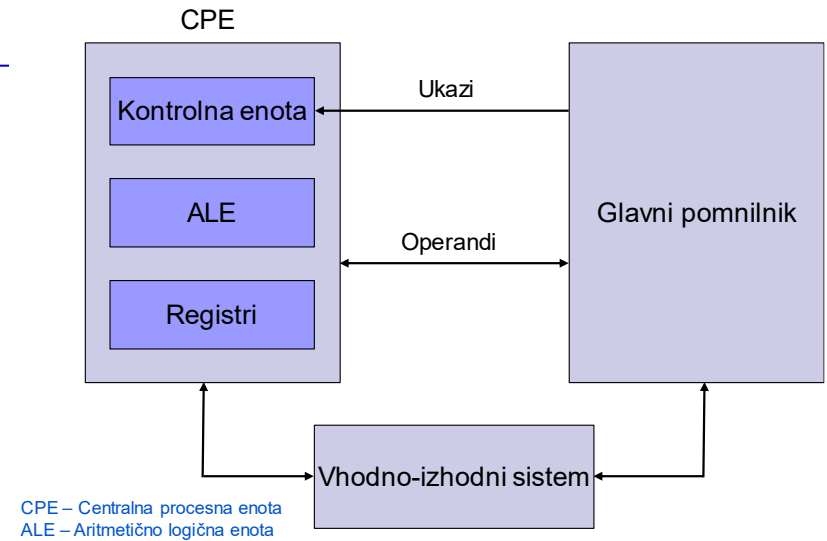


Kaj boste lahko izvedeli pri predmetu
Računalniška arhitektura?

Predavanja, vaje

Vsebina predavanj:

- RA-1 Uvod
- RA-2 Razvoj strojev za računanje
- RA-3 Osnovni principi delovanja
- RA-4 Strojni ukazi
- RA-5 Predstavitev informacije v računalniku
- RA-6 Zgradba in delovanje CPE
- RA-7 Merjenje zmogljivosti CPE
- RA-8 Pomnilnik
- RA-9 Pomnilniška hierarhija



Vsebina laboratorijskih vaj:

- Spoznati **osnove računalniške arhitekture** s praktičnega vidika
- **Razumeti delovanje računalnika (ARM)** s programiranjem v zbirnem jeziku
- **Podrobnejši vpogled:**
 - **v delovanje računalnika**
 - **v izvajanje programov na računalniku**

Nadgradnje znanja -> predmeta Organizacija računalnikov (OR), Vhodno izhodne naprave (VIN) in ostali sorodni predmeti v višjih letnikih.

Računalnik STM32H750-DK



*Dve praktični,
zanimivi LAB vaji*

Možnost sposoje

```
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite(PI_13, HIGH); // turn the LED on (HIGH is the voltage level)  
  digitalWrite(PJ_2, LOW); // turn the LED off by making the voltage LOW  
  
  delay(1000); // wait for a second  
  i++;  
  
  digitalWrite(PI_13, LOW); // turn the LED off by making the voltage LOW  
  digitalWrite(PJ_2, HIGH); // turn the LED on (HIGH is the voltage level)  
  
  delay(1000); // wait for a second  
  
  Serial3.print("Hello World: ");  
  Serial3.println(i);  
}
```

- Računalnik FRI-SMS
 - Mikrokrmilnik AT91SAM9260 iz družine mikrokrmilnikov ARM9



Zakaj je računalniška arhitektura pomembna ?

- pet vprašanj
- in ...
- pet odgovorov

1. Produkti z aparaturno in programsko opremo

Primeri uspešnih projektov/podjetij (HW+SW)



Make your home healthier,
your office more productive

Uncover the simple solutions. With just a small, stylish, cordless
and connected Cube in each room.

Get Your Cubes Now!

Winter 2013 batch available!

Chipolo - Bluetooth Item Finder for iPhone and Android
by The Chipolo Team

Home Updates **17** Backers **5,329** Comments **1,611**

Funded! This project was successfully funded on November 15, 2013

Trbovlje, Slovenia Technology

Chipolo
Nothing is lost.

GO:GLOBAL MEMBER | SPS SK200 AUTUMN BATCH 2014

Chipolo
Finalisti tekmovanja Start:up leta 2016

5,329 backers
\$293,014 pledged of \$15,000 goal
0 seconds to go

Project by
The Chipolo Team
Trbovlje, Slovenia

First created - 0 backed
Has not connected Facebook



COSYLAB



OPEN INSTRUMENTS
FOR EVERYONE



826 backers
\$256,125 pledged of \$50,000 goal
0 seconds to go

Funding period
Jul 22, 2013 - Sep 20, 2013 (60 days)



Project by
Red Pitaya
Newport News, VA



2. Učinkovitejše programiranje

- Ker **poznavanje arhitekture in delovanja računalnikov** lahko vodi v **učinkovitejše programiranje (programe)**.
 - Primer 1: optimizacija programa za hitrejše delovanje ob upoštevanju delovanja predpomnilnikov

```
/* 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. Učinkovitejše programiranje

- Ker **poznavanje arhitekture in delovanja računalnikov** lahko vodi v **učinkovitejše programiranje** (programe).
 - Primer 2: optimizacija programa za hitrejšo delovanje ob upoštevanju vzporednosti delovanja - paralelnosti

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];  
}
```

Spodnja rešitev je skoraj **4-krat hitrejša** !

Vir: „Pomen poznavanja računalniške arhitekture“,
avtor Miha Krajnc (e-učilnica).

3. Zakaj še zbirnik ?

Zbirnik oziroma strojni jezik je edini jezik, ki ga računalnik razume in zna izvajati

„kdo pa to še zna ?“

3. Zakaj še zbirnik ?

„ker se je „vljudno“ naučiti domačega jezika, kulture ...“

[Dejan Črnica, Dewesoft]:

Past Meetup

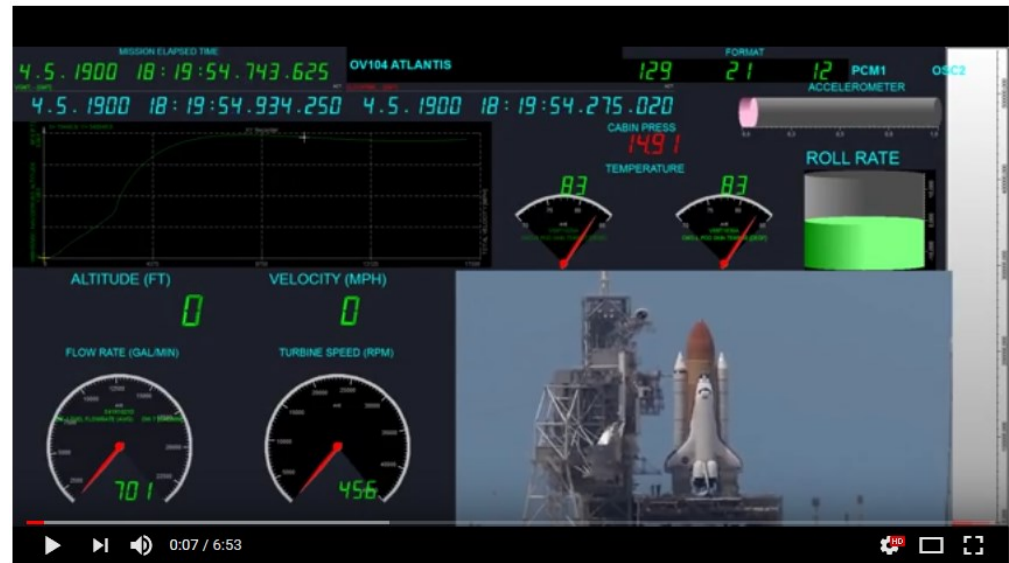
Code optimization on modern processors [Dejan Črnica, Dewesoft]

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.

„pri nas v podjetju razvijalci „govorijo“ v zbirniku...“

„s poznavanjem sistemov in zbirnika lahko pohitrimo kodo tudi **64x** !!!...“

Dejan Črnica Dejan Črnica 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

https://www.youtube.com/watch?v=R8QmL1pyUSo&ab_channel=Dewesoft

3. Zakaj še zbirnik ?

Kje je pomembno znanje arhitekture?

- Če programer ve kako deluje prevajalnik in zbirnik, lahko lažje in hitreje reši napake v kodi.
- Omogoči pisanje **hitrejših programov.**
- Programerji razumejo relativno ceno operacij(**CPI**) in **učinke različnega načina pisanja programa**

Zakaj potem ne programiramo v zbirniku?

V bistvu bomo programirali z posebnimi metodami, ki se neposredno prevedejo v zbirne ukaze.

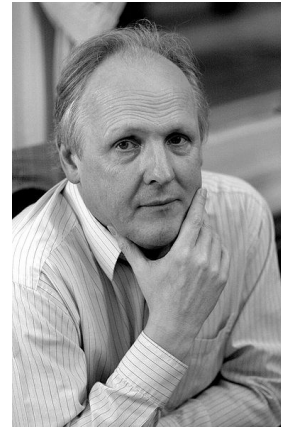
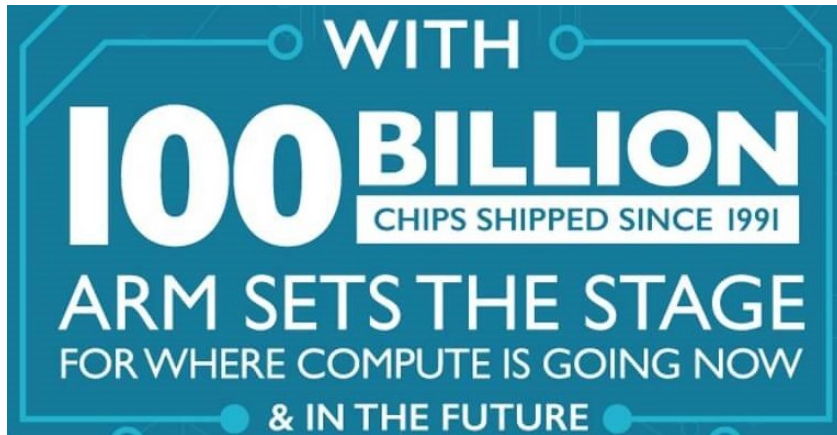
Dodatno gradivo (e-učilnica): „Pomen poznavanja računalniške arhitekture“, avtor Miha Krajnc.

4. Zakaj ravno ARM arhitektura?

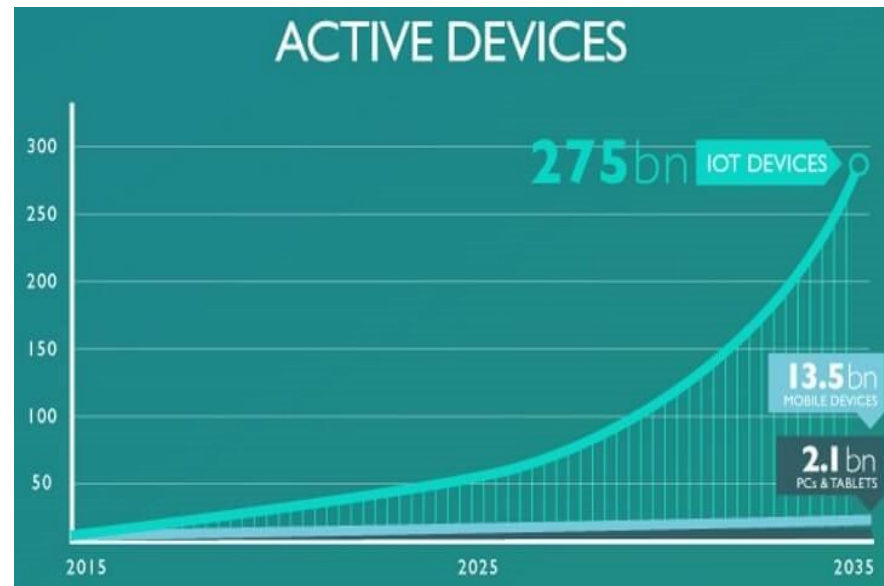
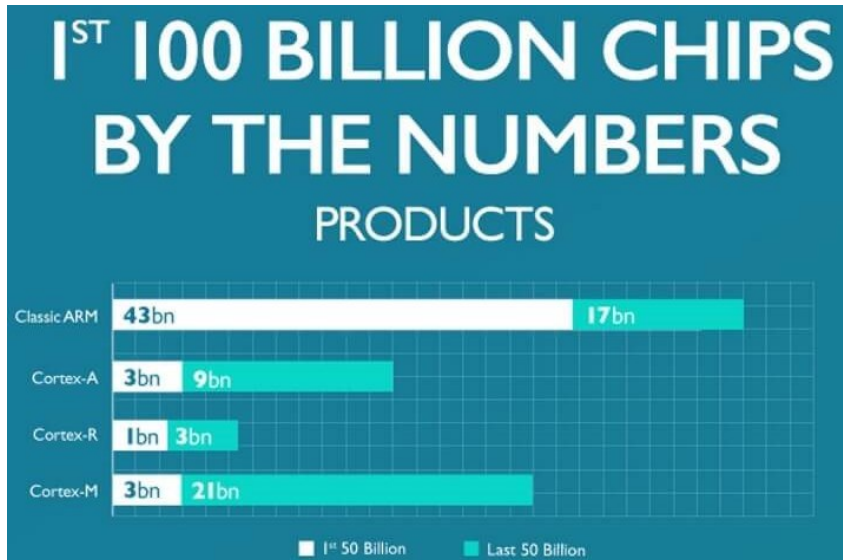
Ker ??? ...“

4. Zakaj ravno ARM arhitektura?

„Steve Furber na 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>

5. „Črna škatla“ tudi za izkušene inženirje ?

H AS THE COMPUTER BECOME A BLACK BOX, EVEN TO EXPERIENCED ELECTRICAL ENGINEERS?

Will we be forever reliant upon large, opaque organizations to build them for us? Absolutely not, we say. And to prove our point, we built our very own laptop, from the circuit boards on up.

Admittedly, we did not delude ourselves that we could build a laptop that would be faster, smaller, or cheaper than those of Apple, Dell, or HP. However, we did set out to build a machine powerful and convenient enough to use every day. Fortunately, our dream inspired enough people to crowdfund the effort. Our laptop, which we call Novena, started shipping to backers in January 2015.



RA – rečemo
„ne !!!“

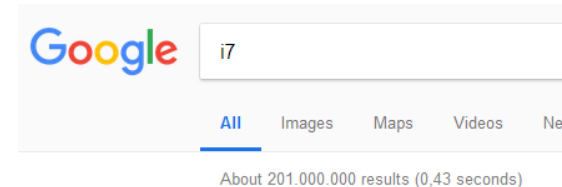
<https://spectrum.ieee.org/novena-a-laptop-with-no-secrets>

1. Uvod :

- 1.1 Predmet RA
- 1.2 Računalniki včeraj in danes
- 1.3 Osnove zgradbe in delovanja računalnikov
- 1.4 Analogno – digitalno, zvezno diskretno
- 1.5 8 pomembnih idej v računalniški arhitekturi (in širše)
- 1.6 Praktična realizacija računalnikov

Razvoj in uporaba računalnikov

- informacijska revolucija
- izredno hiter razvoj v zadnjih 25 letih
- aplikacije, ki so bile do nedavna „nemogoče“, so postale vsakdanje:
 - Računalniki v avtomobilih (avtonomna vožnja)
 - Mobilna telefonija
 - Analiza DNK (projekt Človeški genom)
 - Svetovni splet
 - Iskalniki (iskalnik Google: i7 \Rightarrow \approx 200.000.000 zadetkov v nekaj desetinkah sekunde)
 - AI - Digitalni Asistenti (ChatGPT, generativna UI, ...)



- Med računalniki so

- v izvedbi velike razlike:

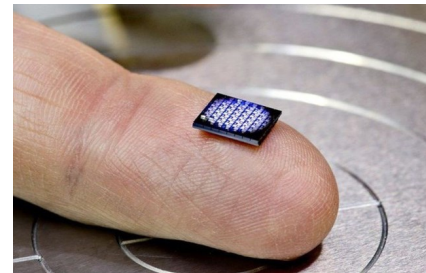
- Superračunalnik

- Enostaven računalnik na enem čipu

- v zgradbi so razlike precej manjše



Slabosti ?
Prednosti ?



- Z vsakim, tudi najenostavnejšim, pa lahko izračunamo vse, kar se da izračunati (je izračunljivo).

■ Najzmogljivejši računalnik na svetu (junij 2021, danes 2. mesto):

- SUPERCOMPUTER FUGAKU v mestu Kobe, Japonska
- 7 630 848 jeder
- Zmogljivost 537 212 TFLOPS
- Poraba energije 29 899 kW (HE Medvode 26 700 kW)



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

<https://www.top500.org/lists/top500/2023/06/>

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



Potreba po superračunalniških kapacitetah



- Molekulske simulacije: primer klasične molekulske dinamike (MD)
- velikost simuliranega sistema: > 100 000 atomov
- vsak atom ima x, y, z koordinate in V_x, V_y, V_z hitrosti \rightarrow 600 000 parametrov
- sisteme simuliramo ~ 1 mikrosekundo s časovnim korakom 2 femtosekunde: 500 000 000 korakov
- 600 000 parametrov * 500 000 000 korakov = **300 tisoč milijard** izračunov



superračunalnik

$\sim 500 \times$



domači računalnik



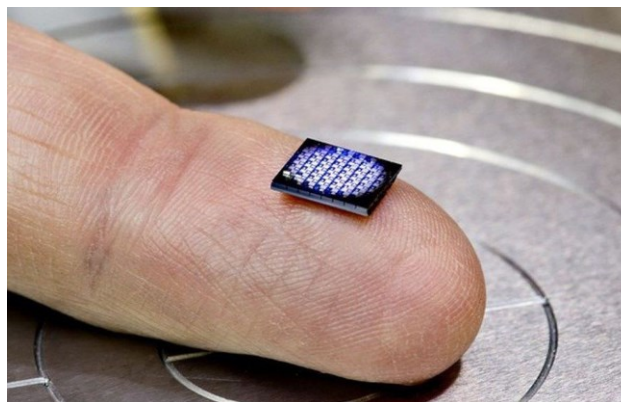
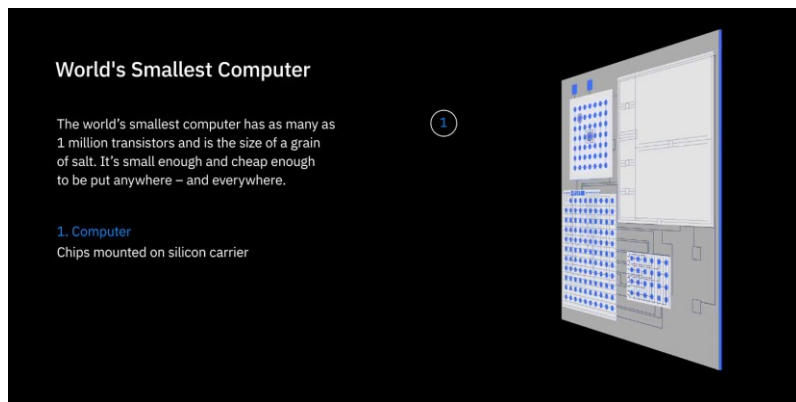
Powered by Zoom

Slovensko nacionalno superračunalniško omrežje

SLING je konzorcij za razvoj omrežja grid/HPC in upravljanje razpršenih računskih infrastruktur v Sloveniji.

<https://www.sling.si/>

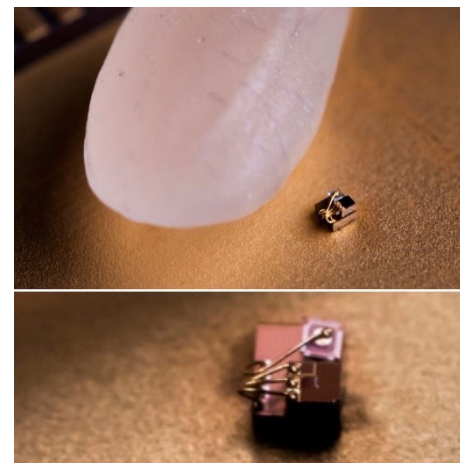
■ Eden najmanjših računalnikov na svetu (I. 2018):



<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.

- + Nizka poraba
- Manjša zmogljivost



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

■ Mikro - računalnik STM32H750-DK (nekje vmes)

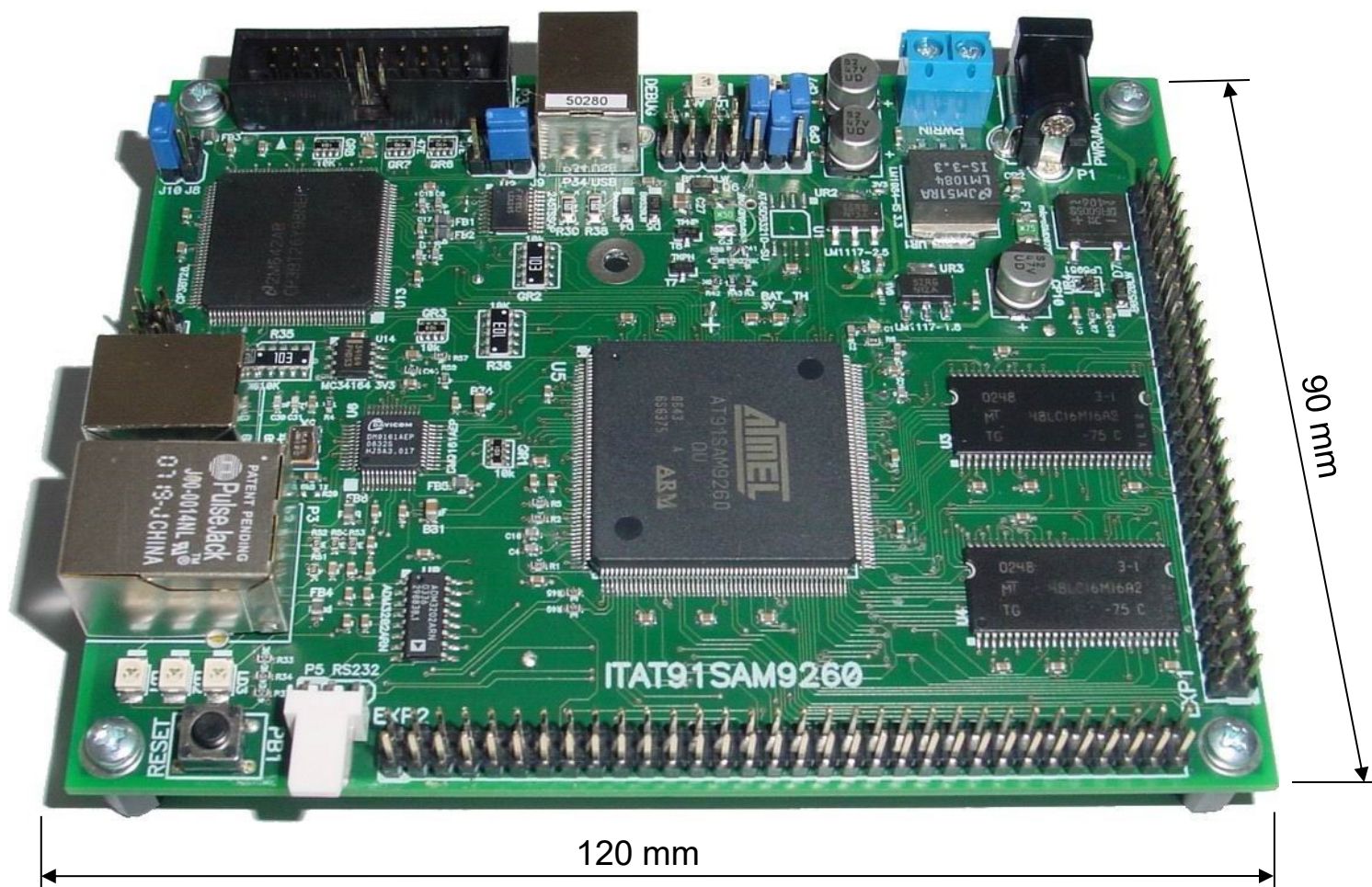
- Mikrokrmilnik STM32H750XB iz družine mikrokrmilnikov ARM-Cortex M7



System LDO, USB and backup regulators POR/PDR/PVD/BOR Multi-power domains Xtal oscillators 32 kHz + 4 ~48 MHz Internal RC oscillators 32 kHz + 4, 48 & 64 MHz 3x PLL Clock control RTC/AWU 1x SysTick timer 2x watchdogs (independent and window) 82/140/168 I/Os Cyclic redundancy check (CRC) Unique ID	Chrom-ART Accelerator™ JPEG Codec Acceleration Cache I/D 16+16 Kbytes Arm® Cortex® -M7 480 MHz	128-Kbyte Flash memory RAM 1056KB incl. 64KB ITCM FMC/SRAM/NOR/NAND/SDRAM Dual Quad-SPI 1024-byte + 4-Kbyte backup SRAM
Control 2x 16-bit motor control PWM synchronized AC timer 10x 16-bit timers 2x 32-bit timers 5x Low-power timer 16-bit High res. timer	Floating point unit (DP-FPU) Nested vector interrupt controller (NVIC) JTAG/SW debug/ETM Memory Protection Unit (MPU) ROP, PC-ROP anti-tamper	Connectivity TFT LCD controller HDMI-CEC 6x SPI, 3x I ² S, 4x I ² C Camera interface Ethernet MAC 10/100 with IEEE 1588 MDIO slave 2x FDCAN (Flexible Data rate) 1x USB 2.0 OTG FS/HS 1x USB 2.0 OTG FS 2x SDMMC 4x USART + 4 UART LIN, smartcard, IrDA, modem control 1x Low-power UART 4x SAI (Serial audio interface) SPDIF input x4 DFSDM (8 inputs/4 filters) SWP (Single Wire Protocol)
Crypto/Hash processor 3DES, AES 256, GCM, CCM SHA-1, SHA-256, MD5, HMAC Security services SFI and SB-SFU	AXI and Multi-AHB bus matrix 4x DMA True random number generator (RNG)	Analog 2x 12-bit, 2-channel DACs 3 x 16-bit ADC (up to 3.6 Msps) 20 channels/up to 2 MSPS Temperature sensor 2x COMP 2x OpAmp

- V današnjem času se je delitev računalnikov poenostavila na tri kategorije po funkcionalnosti :
 - Osebni računalniki (namizni, tablični, . . .)
 - Strežniki
 - Med strežniki so velike razlike v ceni in zmogljivosti
 - Malo zmogljivejši namizni računalnik na spodnjem nivoju
 - Superračunalnik s terabajti glavnega pomnilnika in petabajti zunanlega pomnilnika na zgornjem nivoju
 - Vgrajeni (embedded = vsebovani) računalniki
 - Najštevilčnejša skupina računalnikov
 - Mikroprocesorji (ali mikrokrmilniki) v avtomobilih, mobilnih telefonih, igralnih konzolah, gospodinjskih aparatih, avdio in video napravah, ...

- Mikro - računalnik FRI-SMS (nekje vmes)
 - Mikrokrmilnik AT91SAM9260 iz družine mikrokrmilnikov ARM9

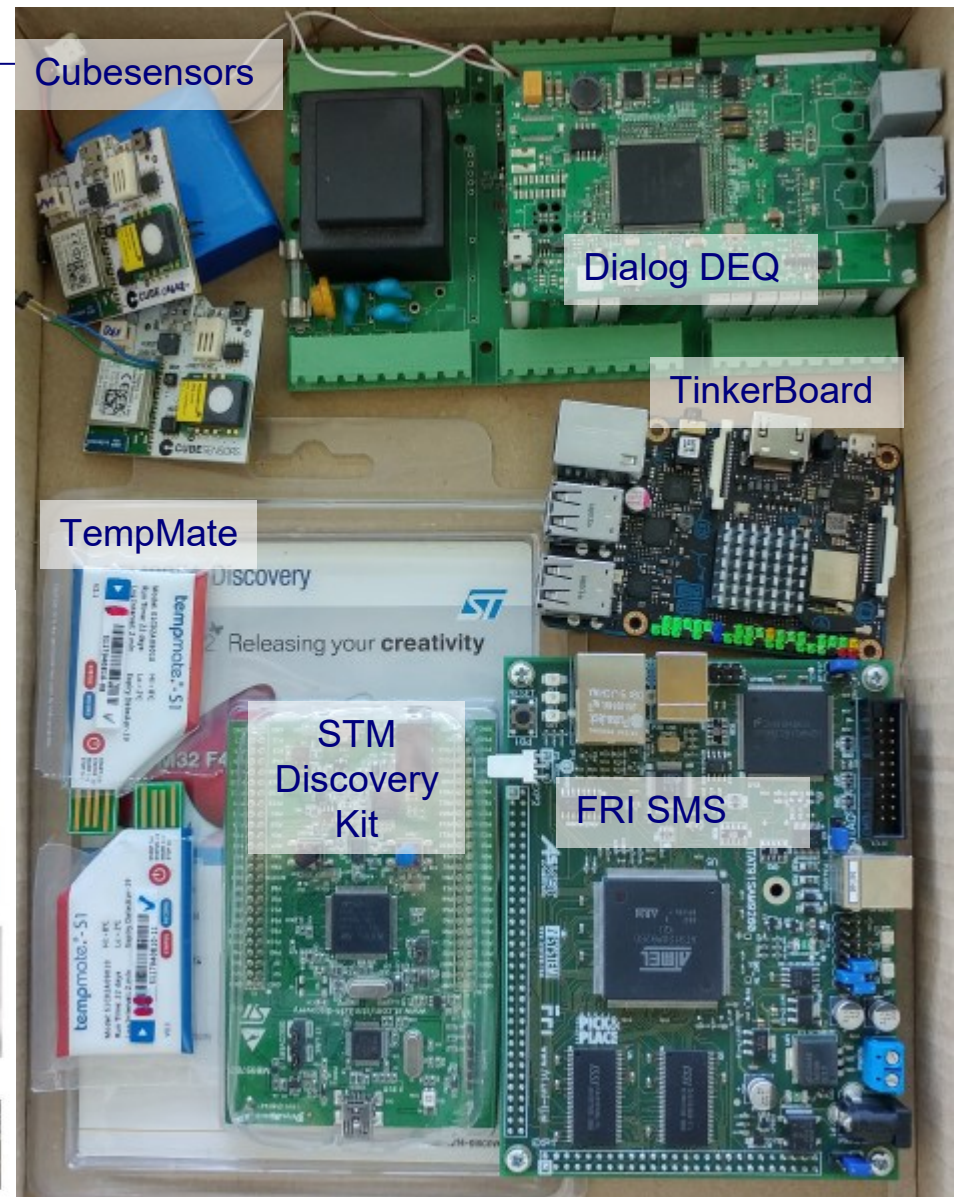


Vgrajeni (embedded) računalniki (konkretni primeri)

Vsi (na sliki desno) so zasnovani
na ARM arhitekturi.

Vsebolj zanimiva tudi RISC-V
arhitektura.

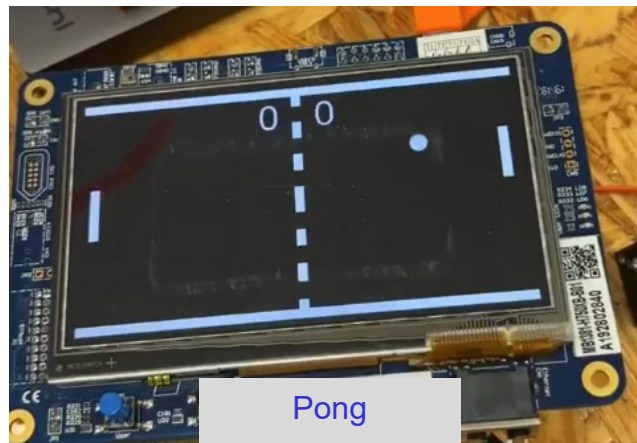
Examples of Embedded Systems



Študentski projekti na STM32H750 plošči



Pacman



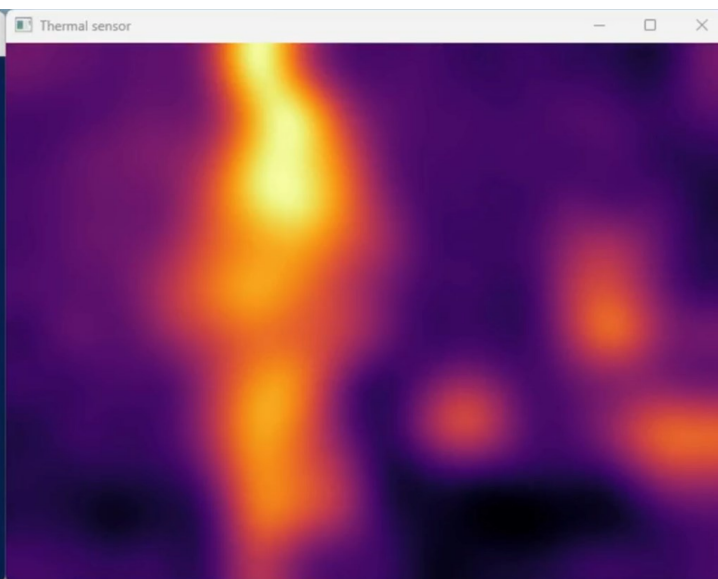
Pong



Termo kamera

```
Windows PowerShell
PS C:\Users\Jakob\git\LIR2> python3 .\classifier.py models/convolutional_model.pt -s COM4 -d
Person present: True
```

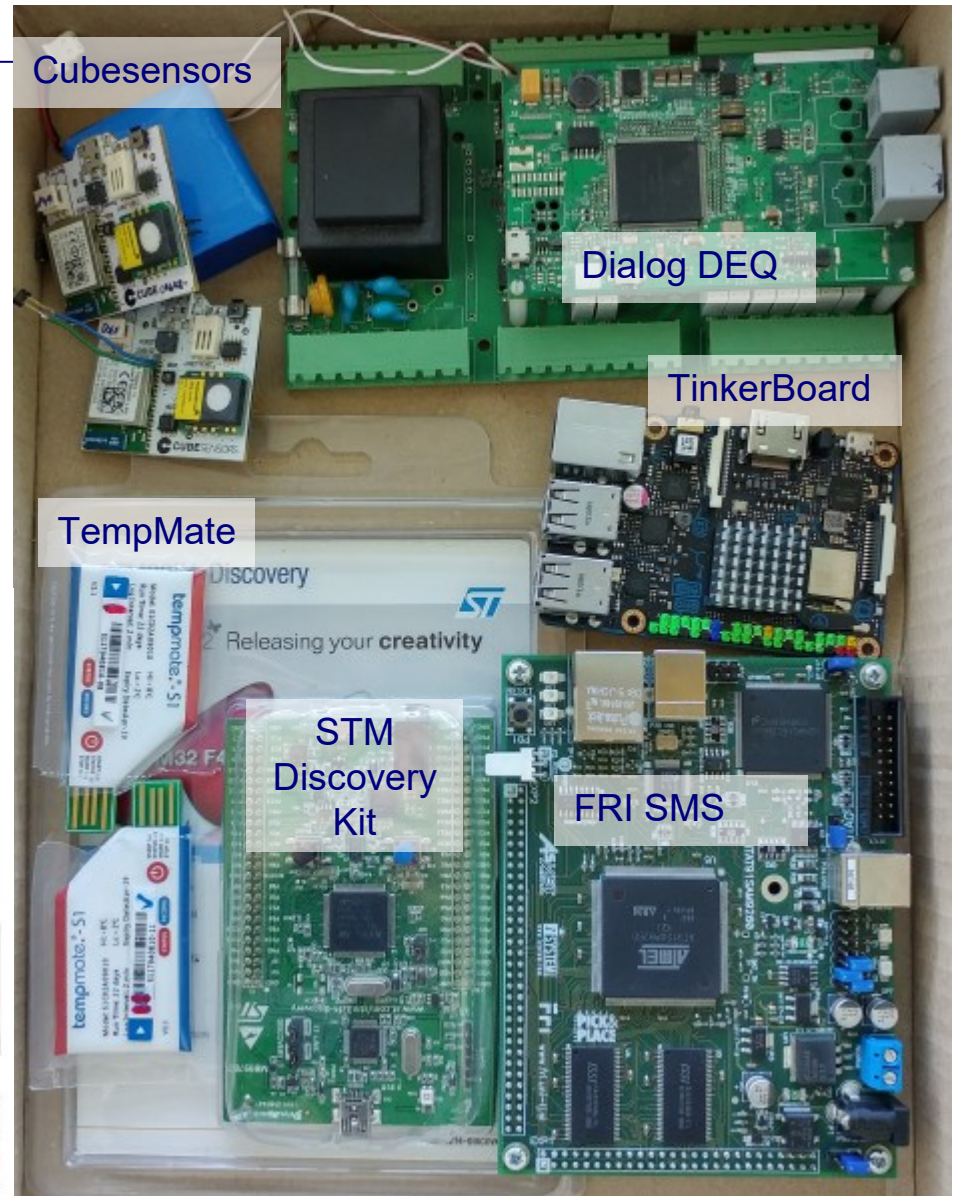
Termo kamera – detekcija človeka



Vgrajeni (embedded)
računalniki
(konkretni primeri)

Vsi (na sliki desno) so
zasnovani na ARM arhitekturi.

Examples of Embedded Systems



1. Uvod :

- ❑ 1.1 Predmet RA
- ❑ 1.2 Računalniki včeraj in danes
- ❑ 1.3 Osnove zgradbe in delovanja računalnikov
- ❑ 1.4 Analogno – digitalno, zvezno diskretno
- ❑ 1.5 8 pomembnih idej v računalniški arhitekturi (in širše)
- ❑ 1.6 Praktična realizacija računalnikov

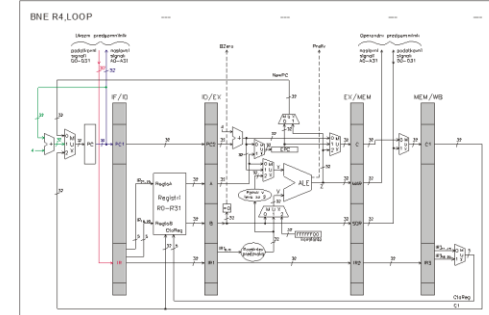
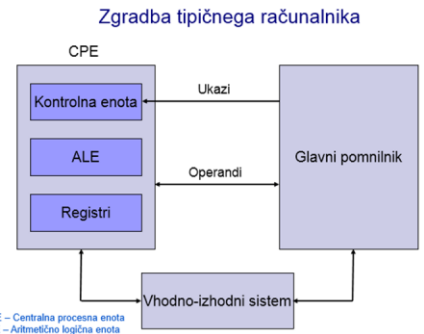
Osnovni definiciji

Def: Arhitektura računalnika je

- obravnava za programerja vidnih lastnosti računalnika na način, ki je neodvisen od njegove logične in fizične realizacije [Kodek]
 - „... to, kar vidi programer na nivoju strojnega jezika ...“

Def: Organizacija računalnika (tudi mikroarhitektura) :

- obravnava logično zgradbo in lastnosti sestavnih delov računalnika in njihovih medsebojnih povezav [Kodek]
 - „ ... je arhitektura posameznih delov ...“
 - „ ... je bližje aparturnemu (HW) nivoju ...“



Neka arhitektura se lahko realizira z različnimi vrstami organizacije in obratno.

Delovanje (digitalnih) računalnikov

- Računalniška arhitektura je tudi zgradba računalnika, kot jo vidi programer, ki programira v strojnem jeziku.
- Strojni jezik je jezik, ki ga sestavljajo ukazi, ki jih računalnik direktno izvaja. Te ukaze imenujemo strojni ukazi.
- Strojni ukazi so ukazi, ki so “vgrajeni” v računalnike. Računalniki različnih proizvajalcev lahko imajo različne strojne ukaze.

Računalnik „razume“ izključno strojne ukaze !!!

Kaj dela računalnik? (Kako deluje?)

- A. Samo izvaja ukaze
- B. Razmišlja in ukrepa
- C. Se uči iz primerov
- D. Govori

Kaj dela računalnik? (Kako deluje?)

Izvaja ukaze !

- Digitalni računalnik je stroj za reševanje problemov tako da izvaja ukaze, ki jih vanj vnašajo ljudje.
- Zaporedje ukazov, ki določajo kako naj stroj izvede določeno nalogo, imenujemo program.
- Elektronsko vezje v računalniku prepoznava in direktno izvaja samo omejeno število strojnih ukazov, v katere se morajo pred izvajanjem spremeniti vsi programi.
- Različni procesorji imajo različne strojne ukaze.

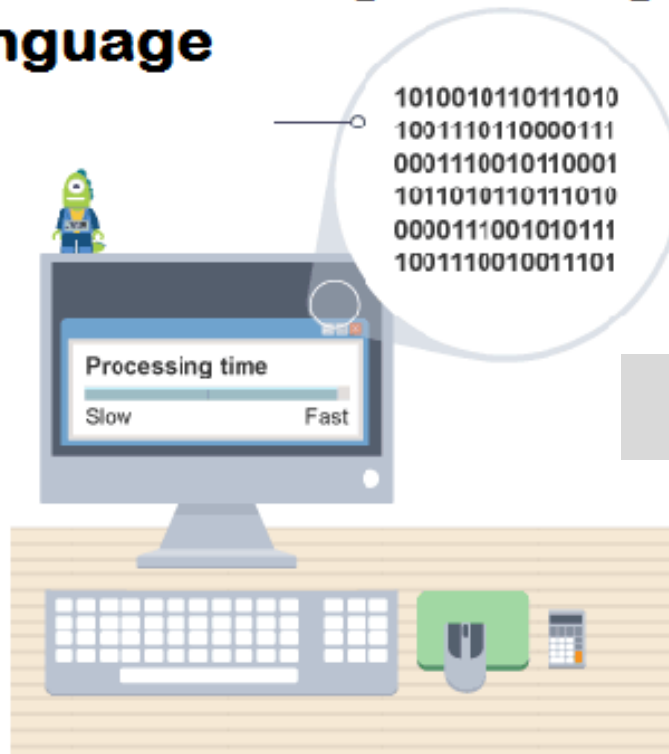
- Ti osnovni ukazi (strojni ukazi) so zelo enostavni kot npr.:
 - Seštej dve števili
 - Testiraj ali je število enako nič
 - Kopiraj podatek iz enega dela računalnikovega pomnilnika v drugi del.
- Vsak program, ki je napisan z nekimi drugačnimi ukazi (npr. z ukazi jezika Java, C++, VisualBasic,...) je zato treba spremeniti (prevesti) v te osnovne ukaze.

Utrditev razumevanja :

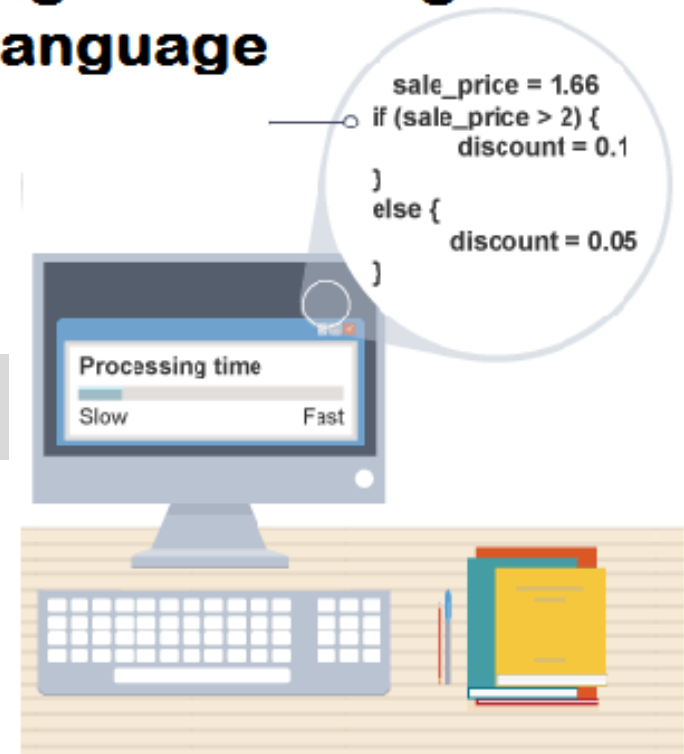
strojni jezik

<-> višjenivojski programski jeziki?

Low Level Programming Language



High Level Programming Language

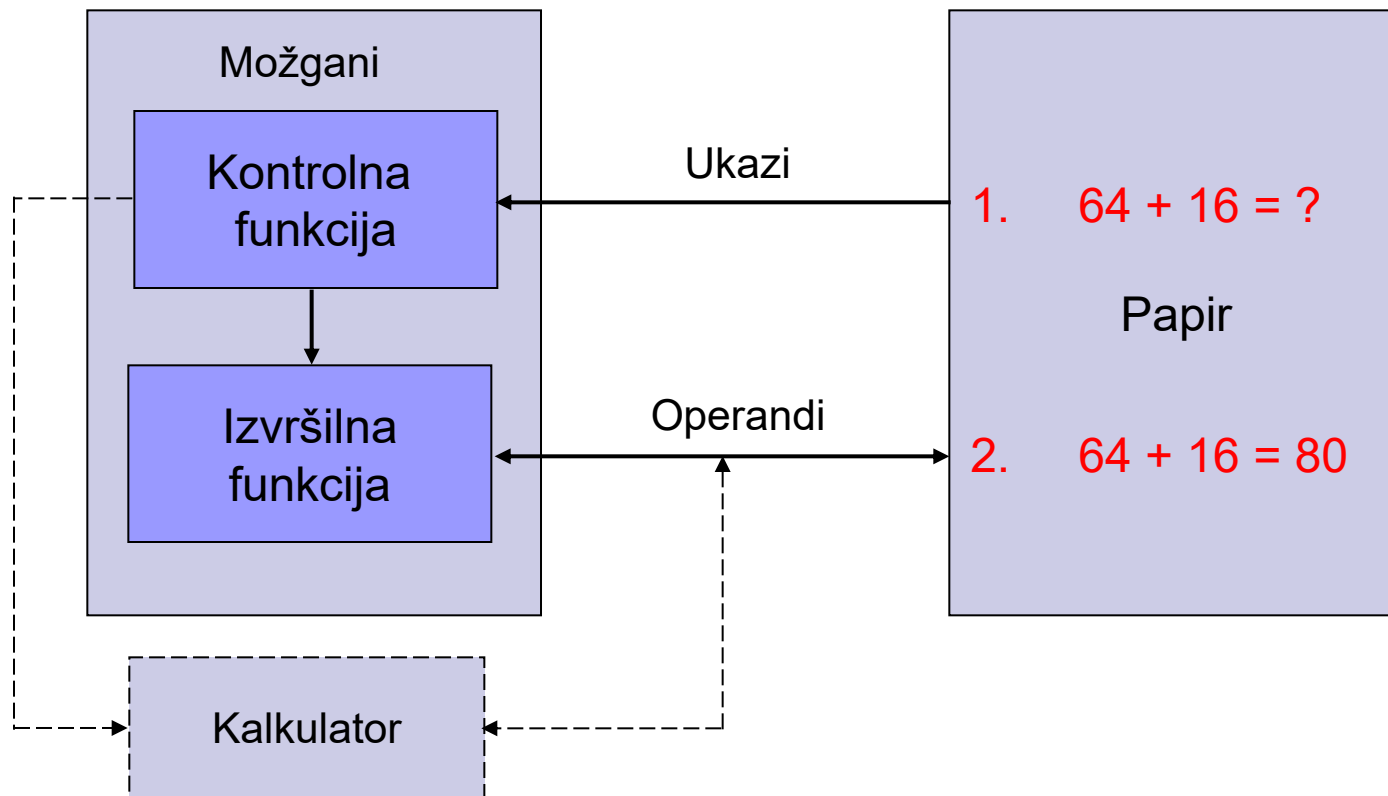


Prevaljalnik
Interpreter

Prenosljivost vs hitrost ?

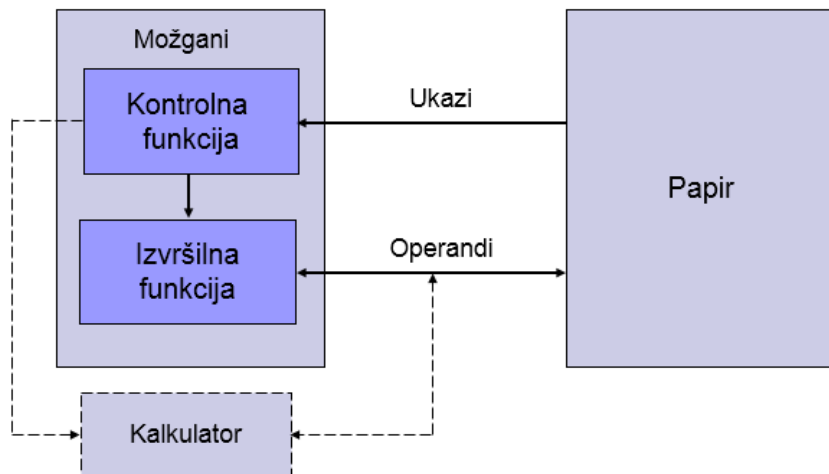
Povezava med ročnim in strojnim računanjem

Ročno računanje

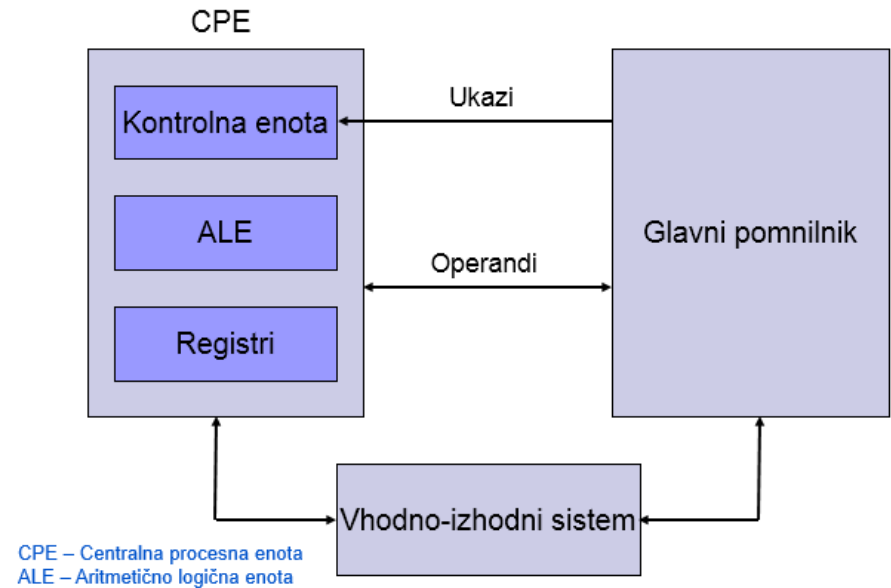


Primerjava modelov računanja

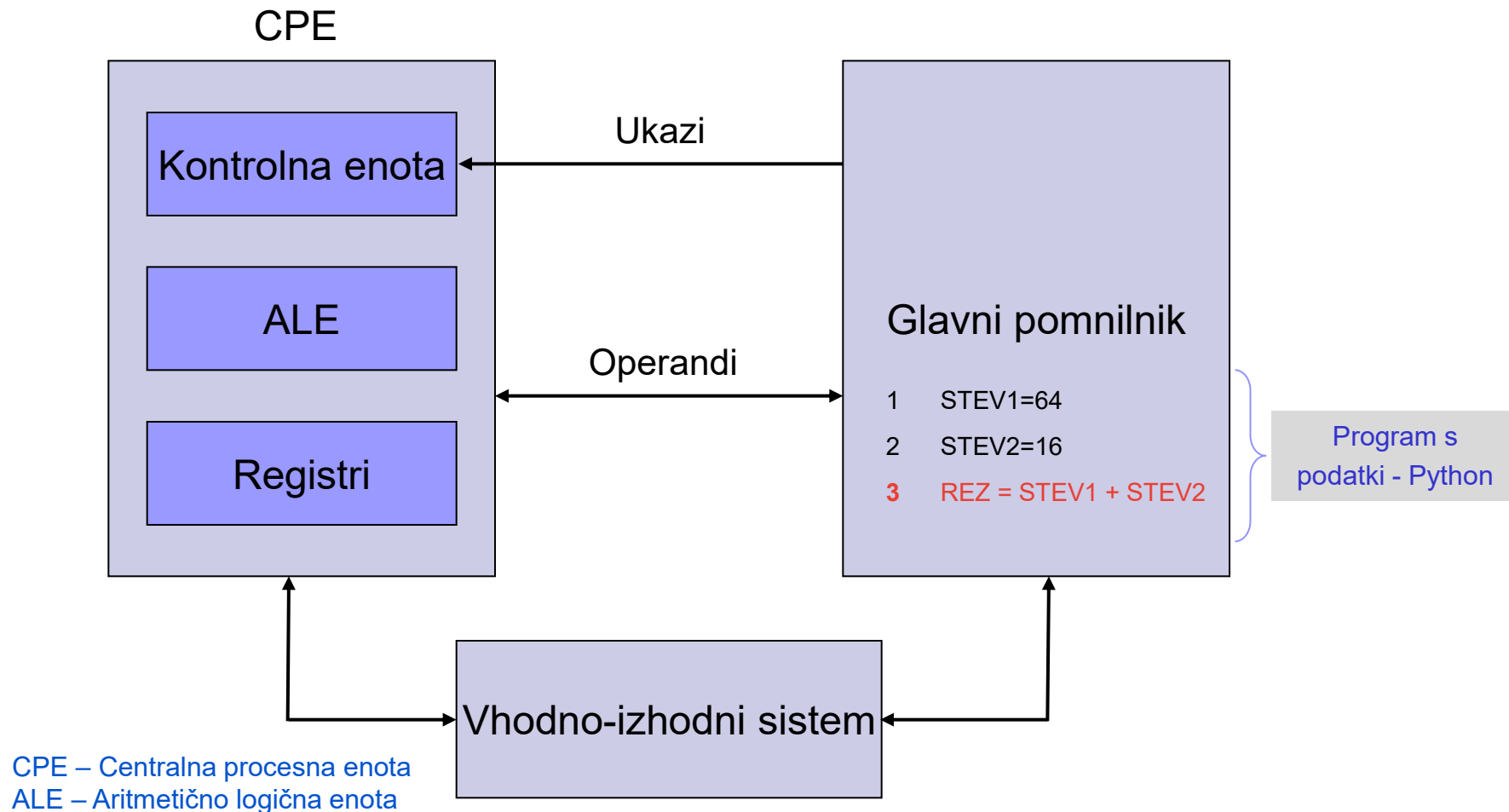
Ročno računanje

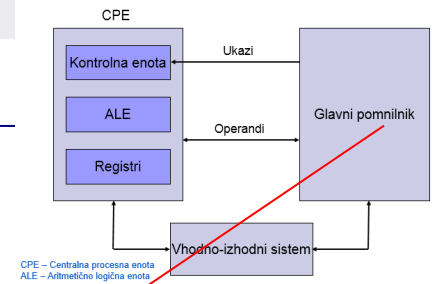


Zgradba tipičnega računalnika

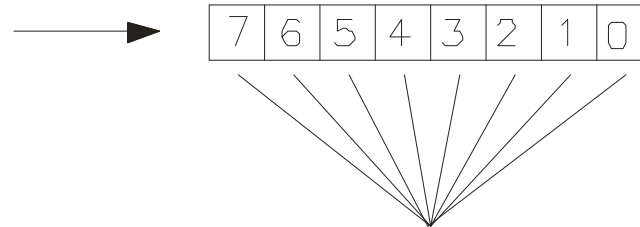
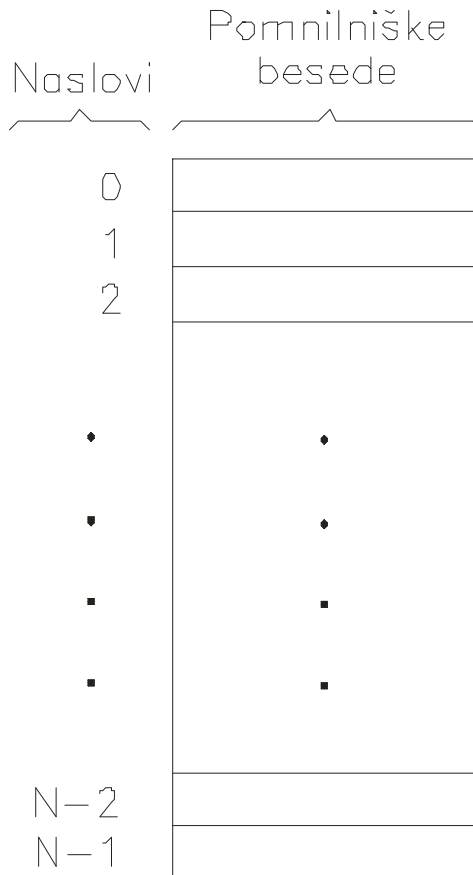


Zgradba tipičnega računalnika



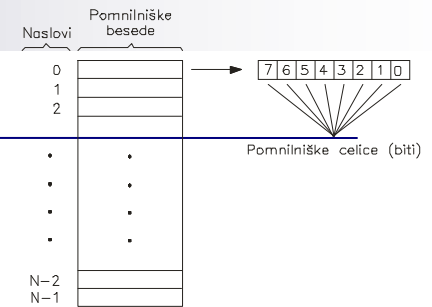


Pomnilnik



Pomnilniške celice (biti)





Pomnilnik

Demonstracija – Logisim EVO

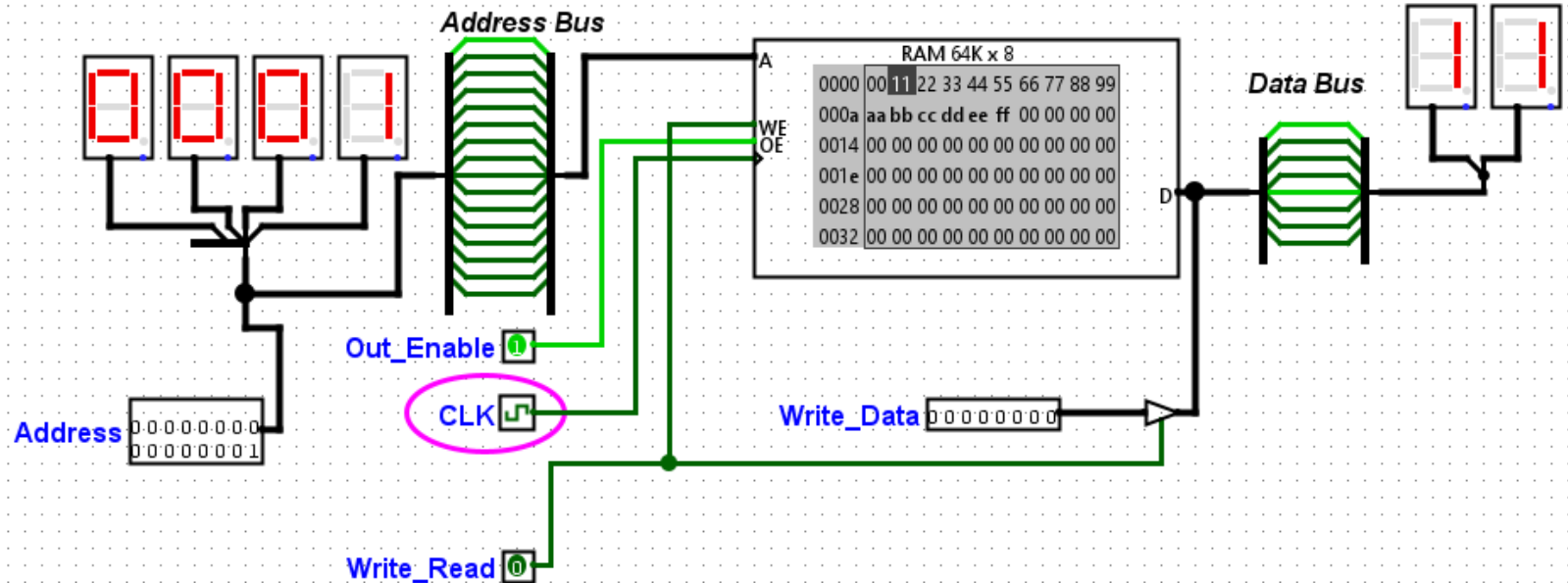
Primer delovanja pomnilnika RAM

Branje :- nastavi naslov (Adress Bus)

- Out_Enable = 1
- Write_Read = 0
- CLK: 1 cikel (dva klika)

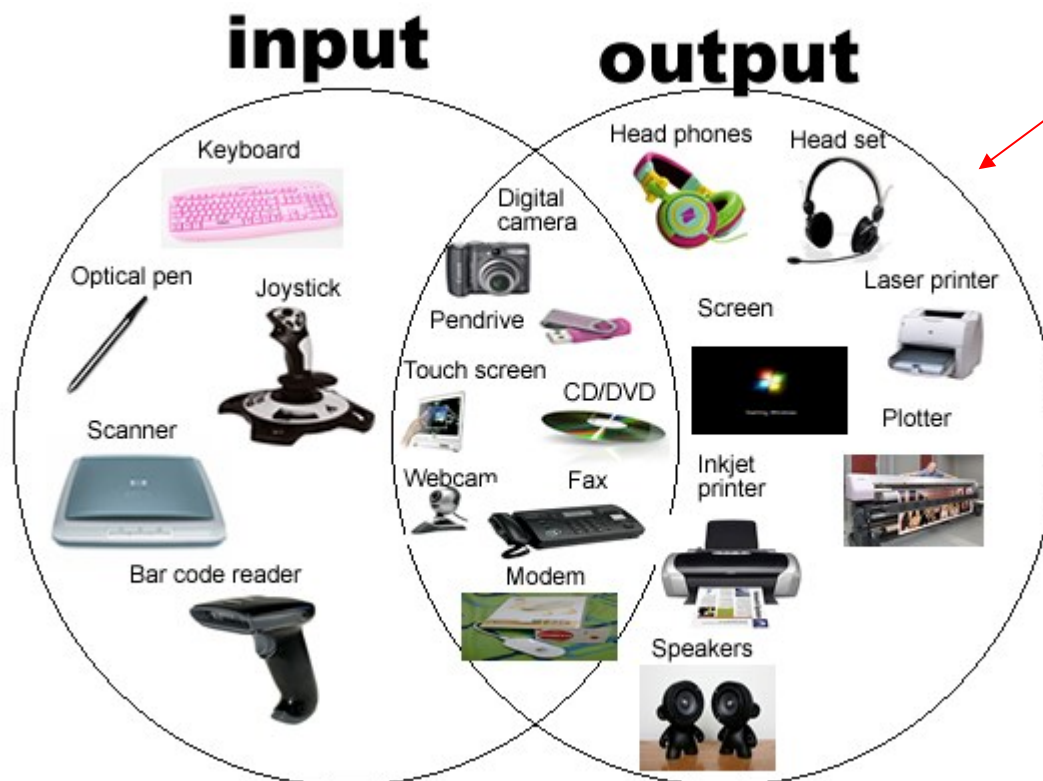
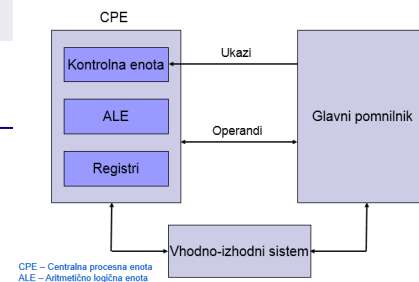
Pisanje :- nastavi naslov (Adress Bus)

- Write_Data = podatek za vpis
- Out_Enable = 0
- Write_Read = 1
- CLK: 1 cikel (dva klika)

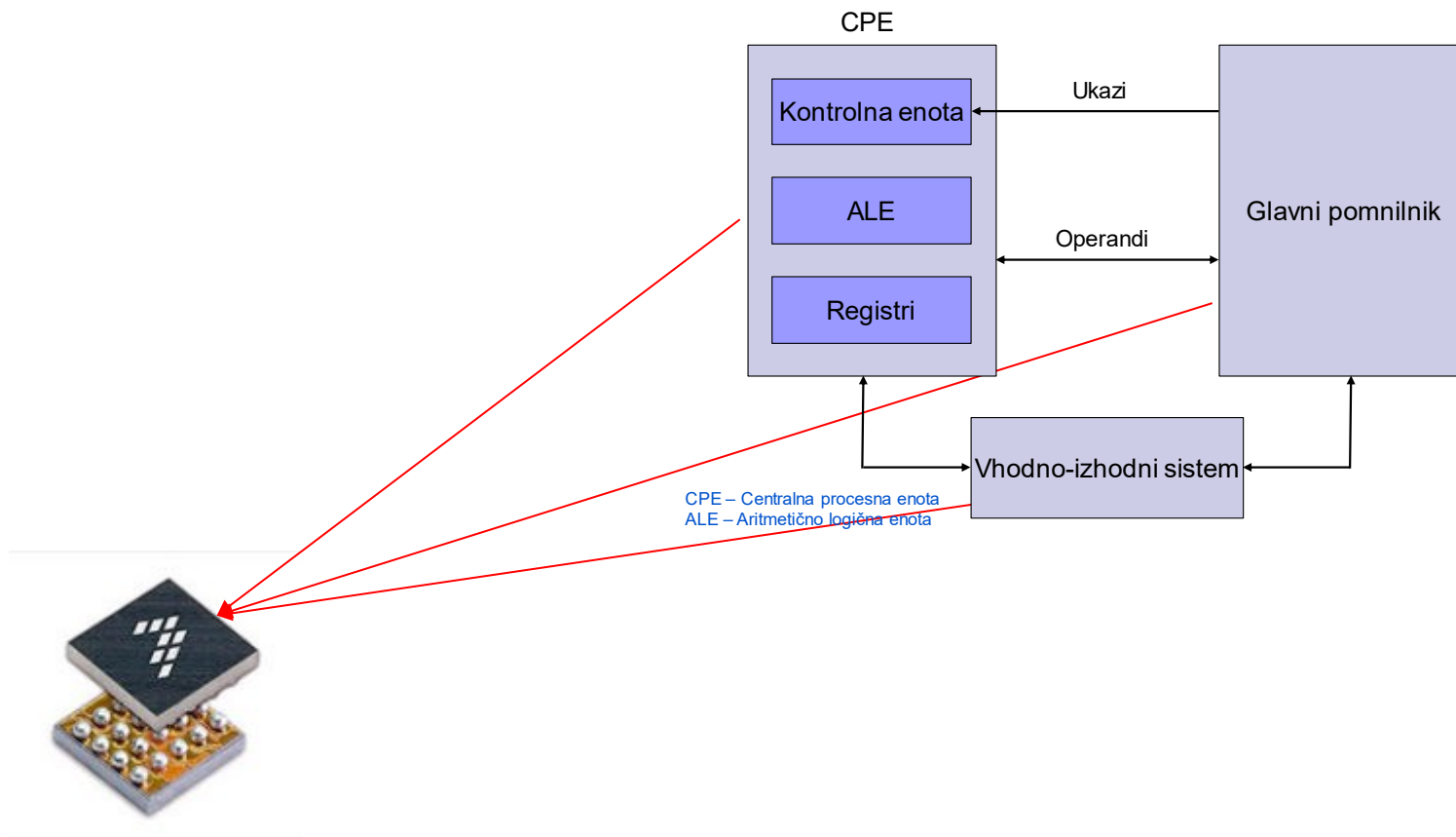


RAM_pomnilnik_demo_EVO.circ

Vhodno-izhodni sistem (naprave)

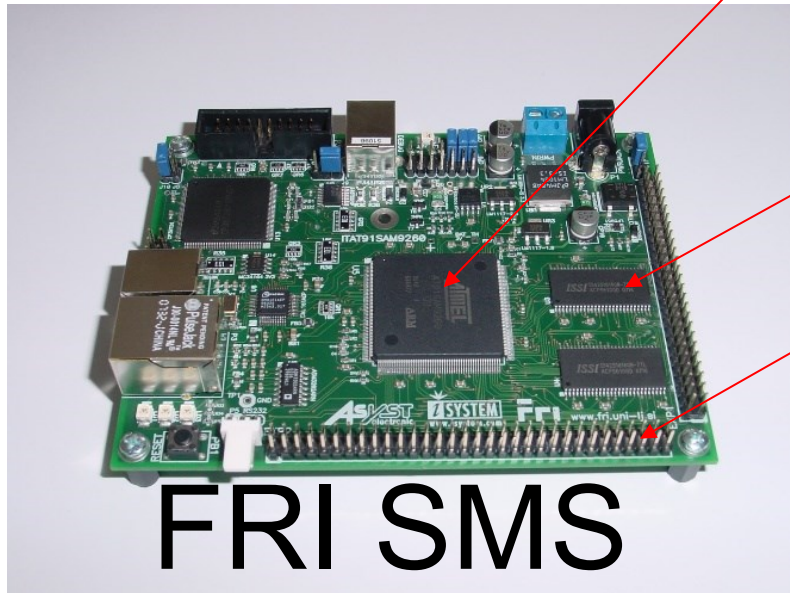


Zgradba tipičnega vgrajenega računalnika

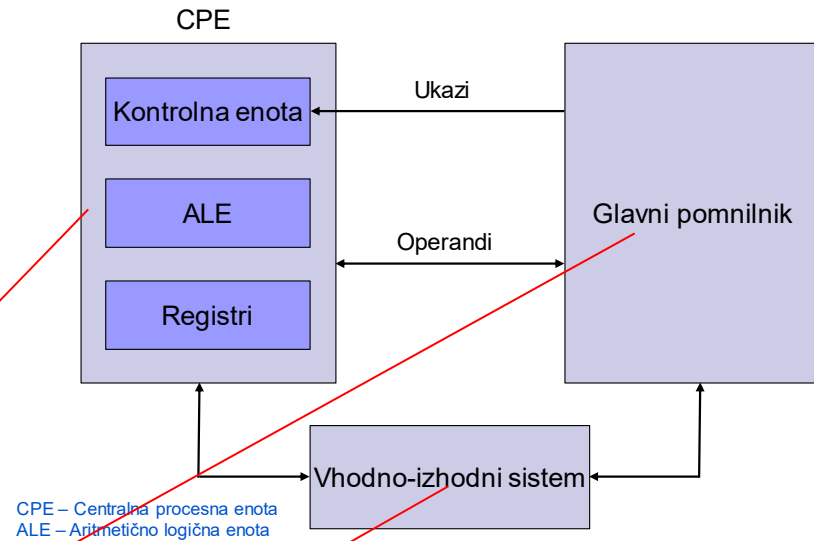


Mikrokontrolnik

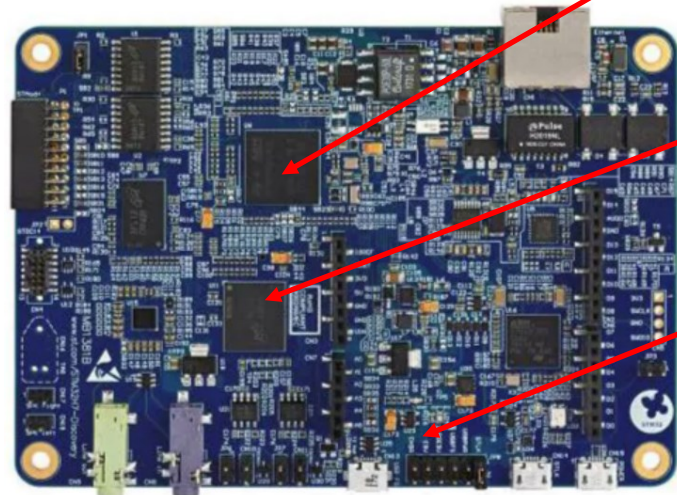
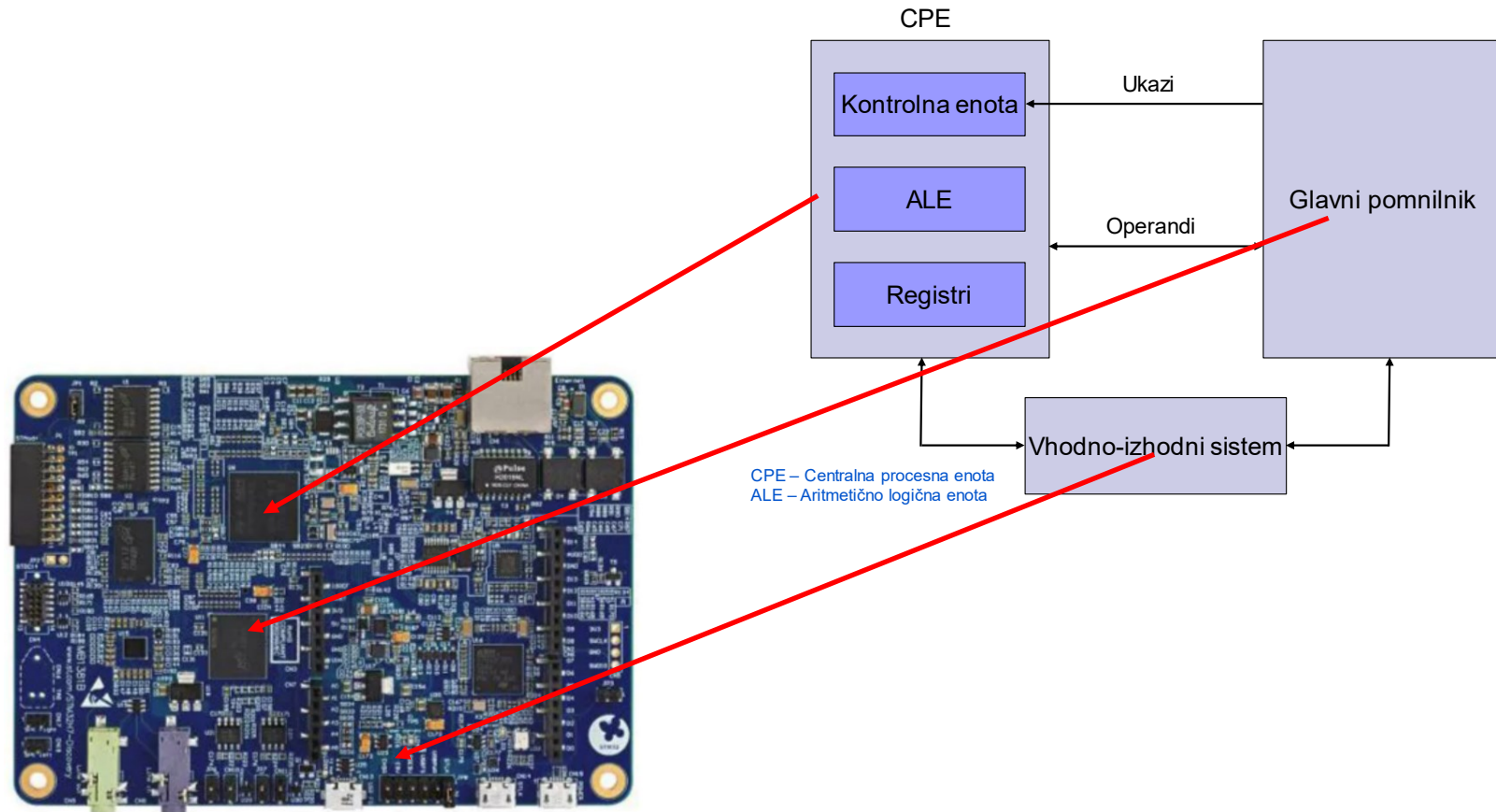
Zgradba tipičnega manjšega (mikro) računalnika



FRI SMS

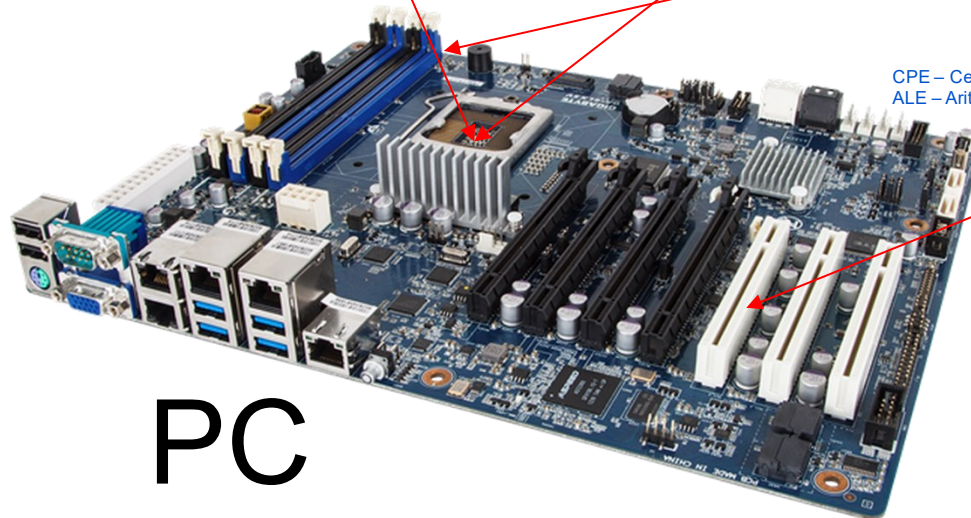


Zgradba tipičnega manjšega (mikro) računalnika

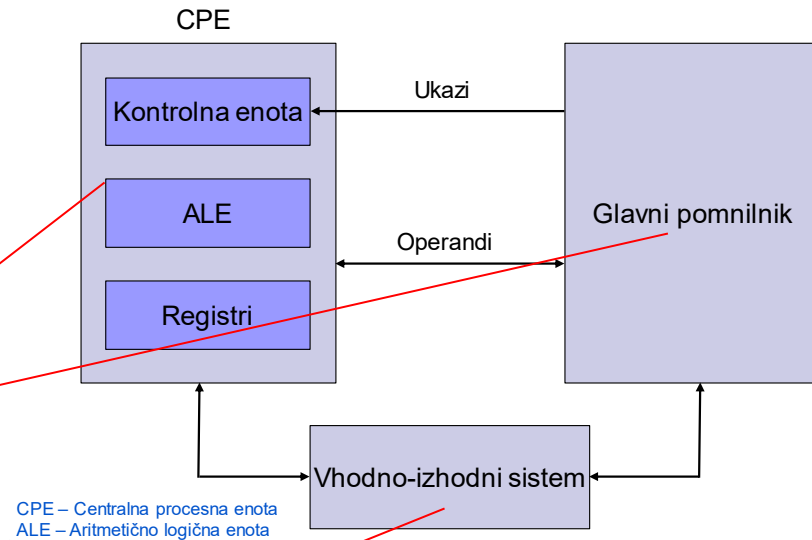


STM32H750-DK

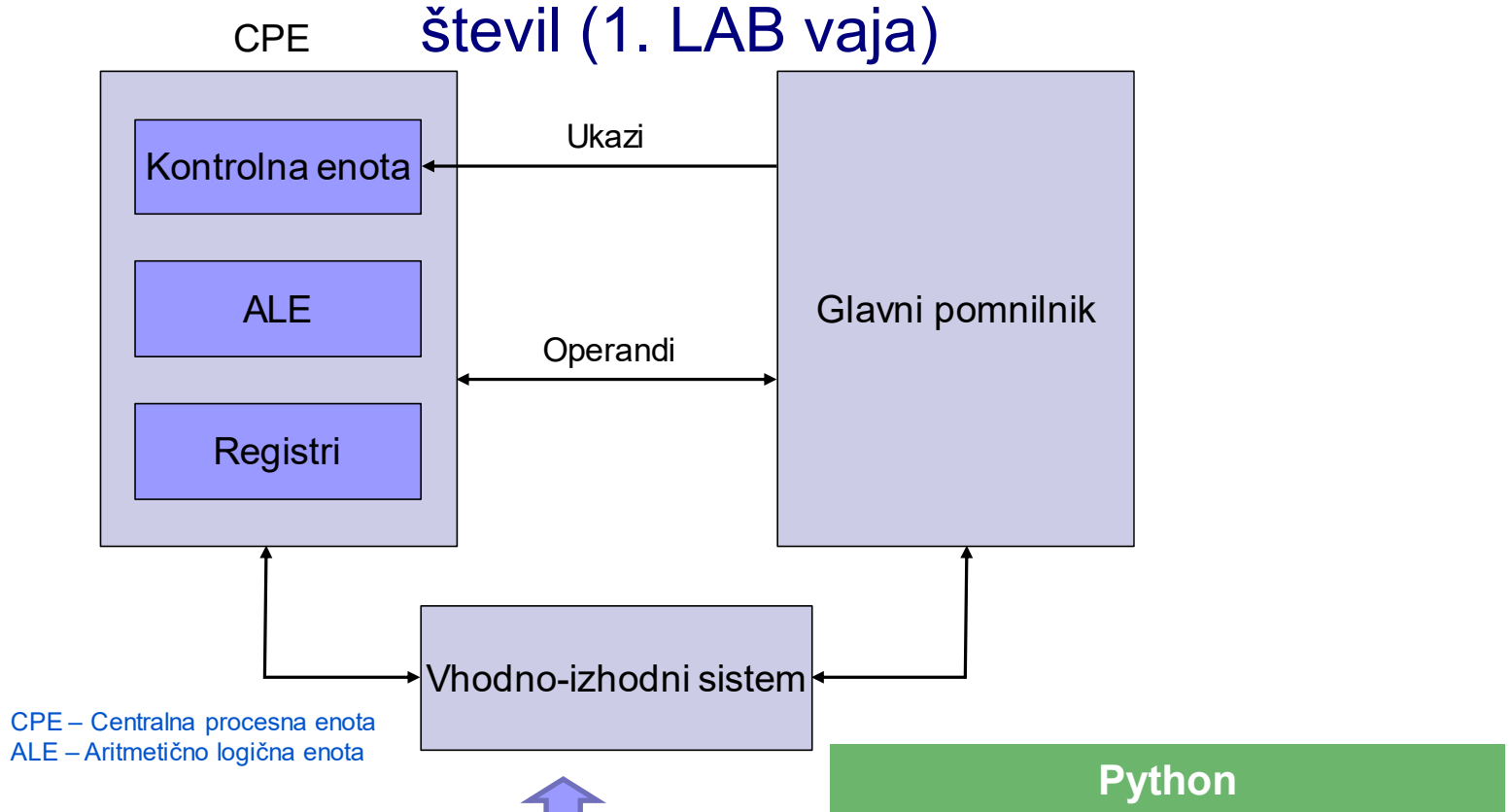
Zgradba tipičnega namiznega računalnika



PC



Zgradba tipičnega računalnika in program vsote dveh števil (1. LAB vaja)



CPE – Centralna procesna enota
ALE – Aritmetično logična enota

Python

- 1 STEV1=64
- 2 STEV2=16
- 3 **REZ = STEV1 + STEV2**

Zbirni jezik	Opis ukaza	Strojni jezik
<code>adr r0, stev1</code>	$R0 \leftarrow \text{nasl. stev1}$	0xE24F0014
<code>ldr r1, [r0]</code>	$R1 \leftarrow M[R0]$	0xE5901000
<code>adr r0, stev2</code>	$R0 \leftarrow \text{nasl. stev2}$	0xE24F0018
<code>ldr r2, [r0]</code>	$R2 \leftarrow M[R0]$	0xE5902000
<code>add r3, r2, r1</code>	$R3 \leftarrow R1 + R2$	0xE0823001
<code>adr r0, rez</code>	$R0 \leftarrow \text{nasl. rez}$	0xE24F0020
<code>str r3, [r0]</code>	$M[R0] \leftarrow R3$	0xE5803000

Program vsote dveh števil (1. LAB vaja)

Python

Zbirnik

<http://goo.gl/YXQ5qN>

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



```
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
```

Spremenljivke v pomnilniku

Frames	Objects
Global frame	
STEV1	64
STEV2	16
REZ	80

<https://cpulator.01xz.net/?sys=arm&loadasm=share/s9brA0e.s>

Zgradba tipičnega računalnika in program vsote dveh števil (1. LAB vaja)

Python

- 1 STEV1=64
- 2 STEV2=16
- 3 **REZ = STEV1 + STEV2**



Zbirnik

```

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

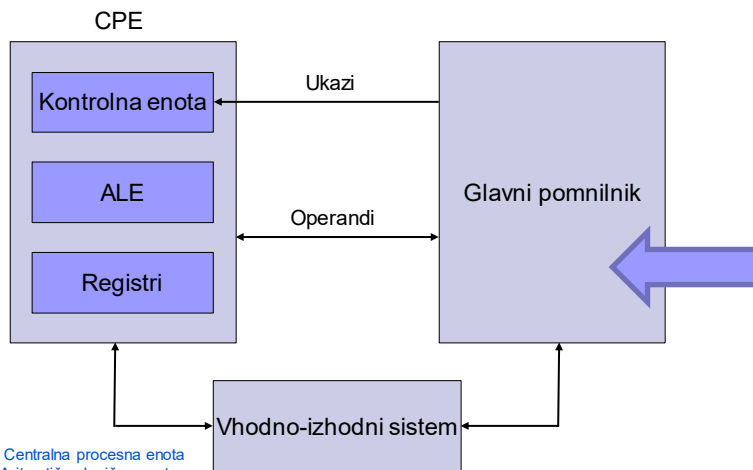
add r3,r1,r2 // r1 + r2 -> r3

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



Strojni jezik

Address	Memory contents and ASCII
00000010	00000000 00000000 00000000 00000000
00000020	00000010 00000040 00000000 e24f0014
00000030	e5901000 e24f0018 e5902000 e0813002
00000040	e24f0020 e5803000 aaaaaaaaa aaaaaaaaa



CPE – Centralna procesna enota
ALE – Aritmetično logična enota

Python (zgled: REZ = STEV1 + STEV2)

Seštevanje spremenljivk v Pythonu.

<http://goo.gl/YXQ5qN>

Python 2.7

```
1 STEV1=0x40
2 STEV2=0x10
3 REZ = STEV1 + STEV2
→ 4 print (" STEV1 = " + hex(STEV1) + "\n+STEV2 = " + hex(STE
```

Frames

Objects

Print output (drag lower right corner to resize)

Global frame

STEV1	64
-------	----

STEV2	16
-------	----

REZ	80
-----	----

```
STEV1 = 0x40
```

```
+STEV2 = 0x10
```

```
-----
```

```
REZ = 0x50
```

Program vsote dveh števil (1. LAB vaja)

Zbirnik

```
.org 0x20
STEV1: .word 0x10 // 32-bitna spr.
STEV2: .word 0x40 // 32-bitna spr.
VSOTA: .word 0 // 32-bitna spr.

.global _start
_start:

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

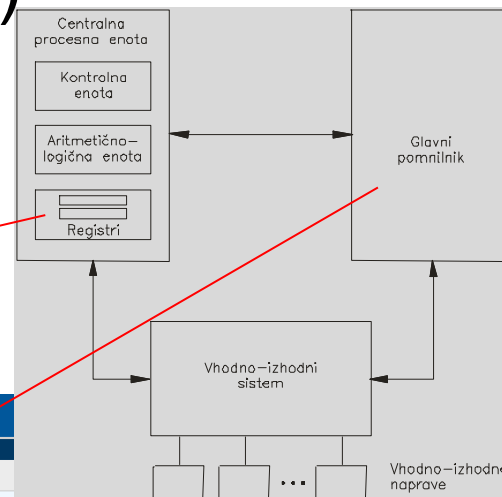
    adr r0, STEV2 // Naslov od STEV2 -> r0
    ldr r2, [r0] // Vsebina iz naslova v r0 -> r2

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

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

Zbirnik (zglede: REZ = STEV1 + STEV2)

CPUlator – spletni simulator



CPUlator

The screenshot shows the CPUlator web simulator interface with the following sections:

- Registers:** A list of registers (r0-r15, sp, lr, pc, cpsr, spsr) with their current values. The PC register is highlighted at 0000002c.
- Disassembly (Ctrl-D):** A table showing the current instruction stream. The instruction at address 0000002c is `ldr r0, [r0, #0x20]`, which is highlighted in red. The instruction at 0000002d is `ldr r1, [r0]`.
- Q Memory (Ctrl-M):** A table showing memory contents and ASCII values. The memory at address 00000020 is highlighted in blue, containing the value 00000014.

This is a zoomed-in view of the Q Memory (Ctrl-M) section. It shows a search bar with the address `20` entered. Below it is a table of memory contents:

Address	Memory contents and ASCII
00000010	00000000 00000000 00000000 00000000
00000020	00000014 00000040 00000000 e24f0014
00000030	e5901000 e24f0018 e5902000 e0813002
00000040	e24f0020 e5803000 aaaaaaaaaa aaaaaaaaaa

1. Uvod :

- ❑ 1.1 Predmet RA
- ❑ 1.2 Računalniki včeraj in danes
- ❑ 1.3 Osnove zgradbe in delovanja računalnikov
- ❑ 1.4 Analogno – digitalno, zvezno - diskretno
- ❑ 1.5 8 pomembnih idej v računalniški arhitekturi (in širše)
- ❑ 1.6 Praktična realizacija računalnikov

Primer 1: Ročna ura

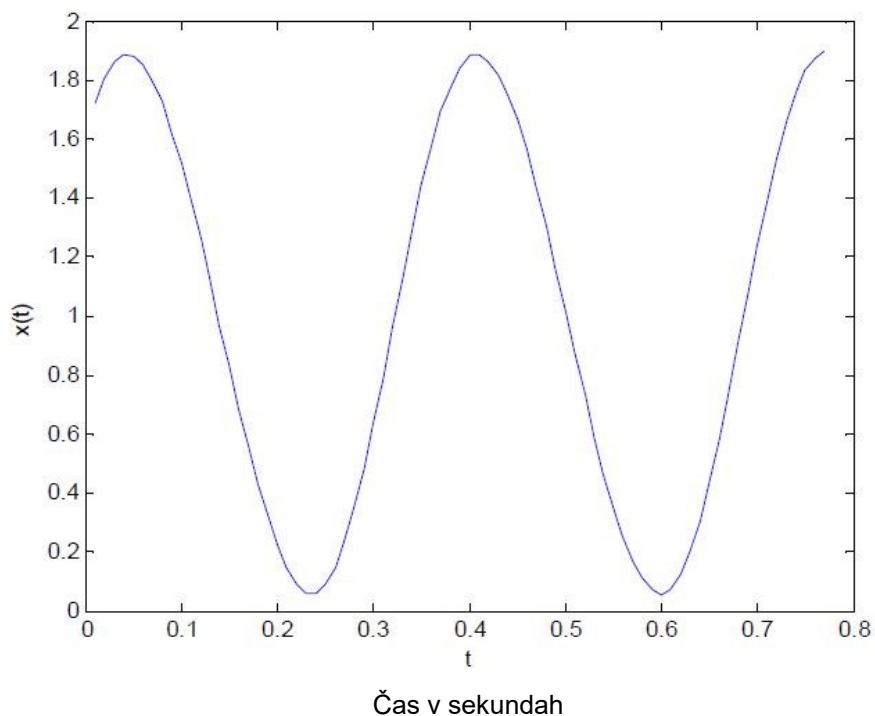
Analogno –
- zvezna predstavitev



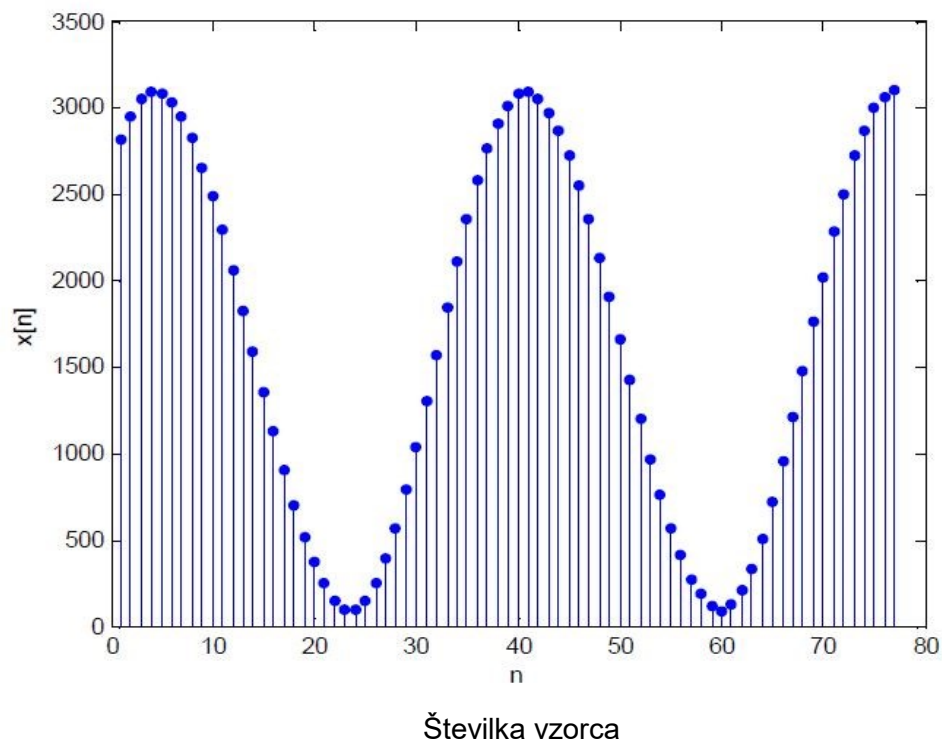
Digitalno –
- diskretna predstavitev



Analogno – - zvezna predstavitev

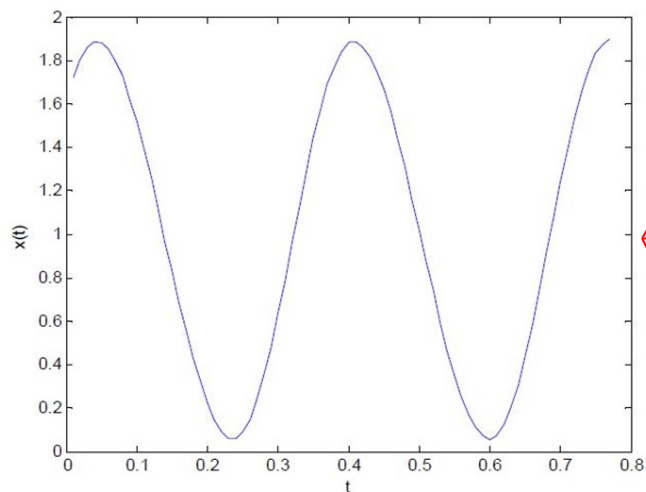


Digitalno – - diskretna predstavitev

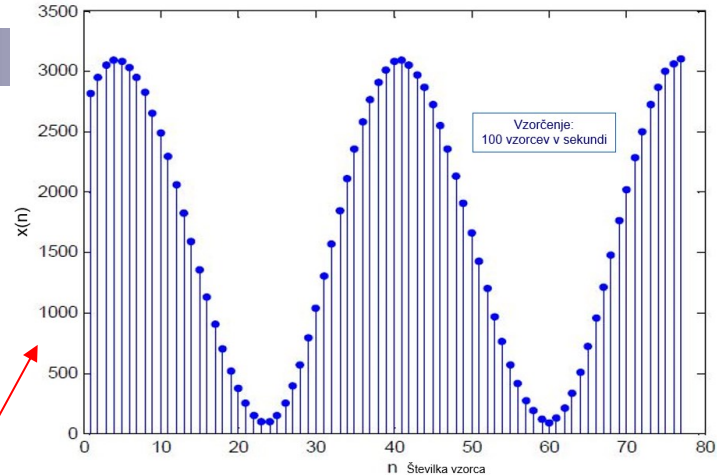
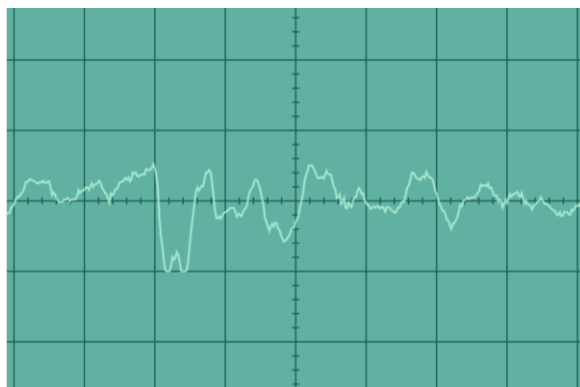


Vzorčenje: 100 vzorcev v sekundi

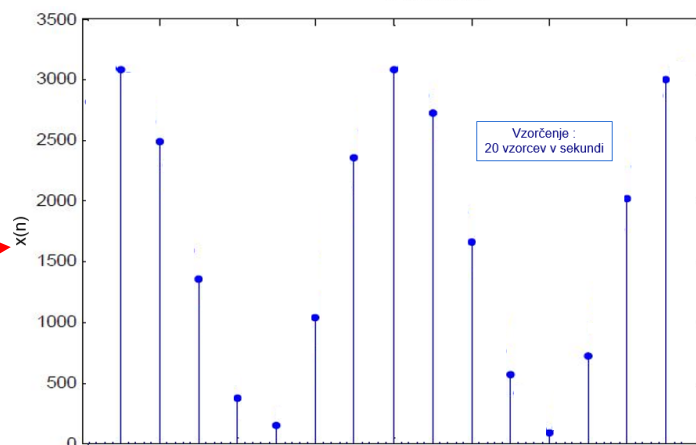
Primer 2: Vzorčenje signala



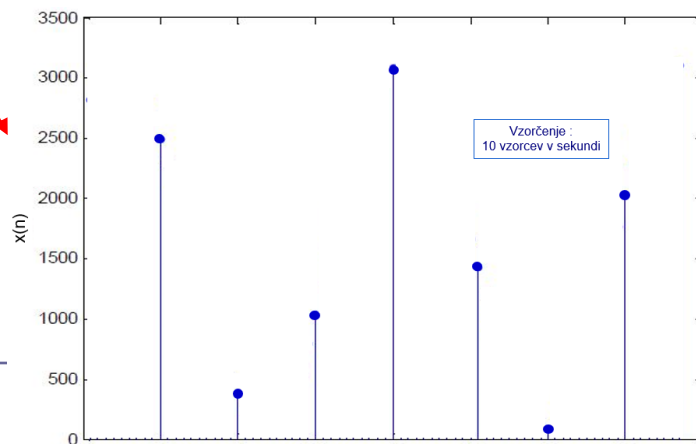
Čas v sekundah



100 vzorcev/sek



20 vzorcev/sek



10 vzorcev/sek

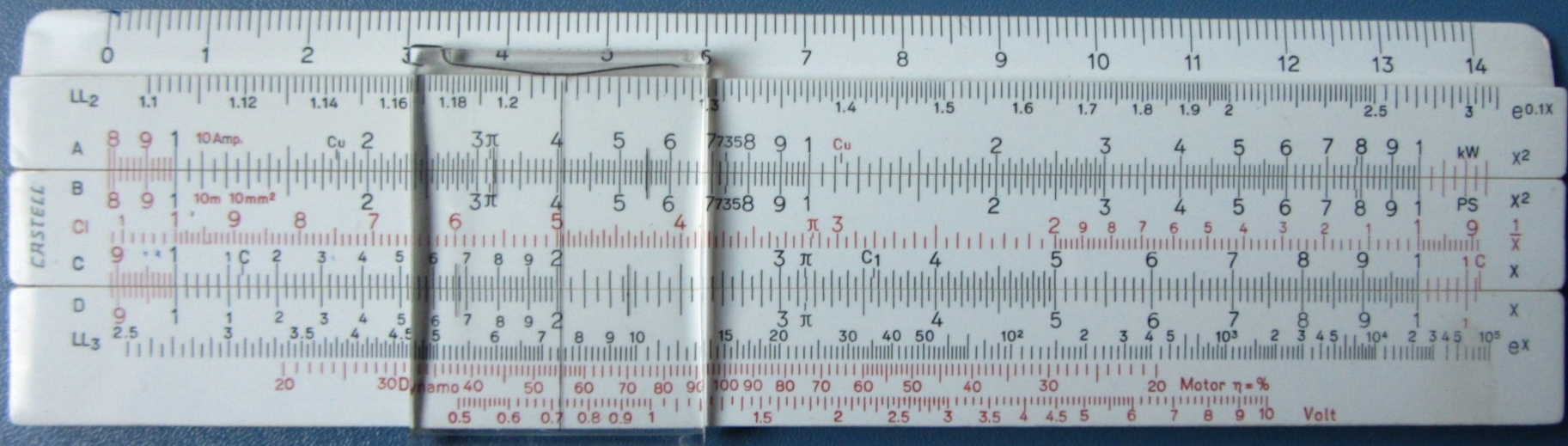


Analogno računanje – zvezna predstavitev števil

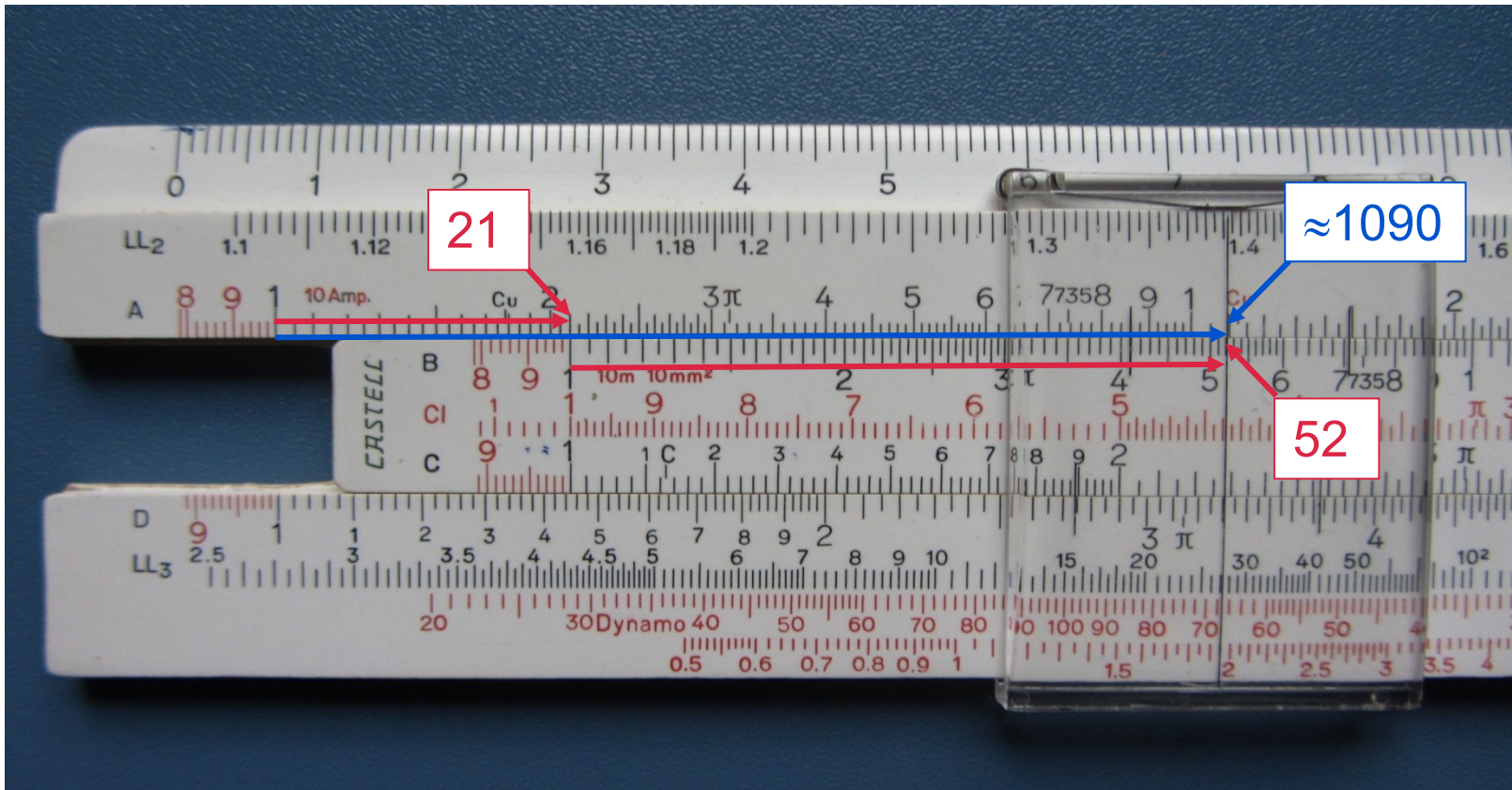
Digitalno računanje – diskretna predstavitev števil
končno število bitov oziroma „končna beseda“

- **Analogno računanje** poteka s predstavitvijo števil z neko drugo fizikalno veličino:

- Z razdaljo \Rightarrow Logaritmično računalno
- Ideja: $\log_{10}(a \cdot b) = \log_{10} a + \log_{10} b$



- Primer množenja npr. 21×52 z logaritmičnim računalom:



$$21 \times 52 \approx 1090$$

Odčitani rezultat

$$21 \times 52 = 1092$$

Točen rezultat

- Z napetostjo \Rightarrow Analogni ojačevalec



- Z napetostjo \Rightarrow Analogni računalnik



Diskretno računanje

- s kroglicami



- s števki od 0 do 9

Digitalno računanje

- s števka 0 in 1



- dvojiški številski sistem:

- osnova številskega sistema je 2
- števki 0 in 1

- dvojiška številka angl. **binary digit = bit**

- bit = ena od dveh števk (0 ali 1) dvojiškega številskega sistema

- **digitalni računalnik temelji na dvojiškem sistemu**

1. Uvod :

- ❑ 1.1 Predmet RA
- ❑ 1.2 Računalniki včeraj in danes
- ❑ 1.3 Osnove zgradbe in delovanja računalnikov
- ❑ 1.4 Analogno – digitalno, zvezno diskretno
- ❑ 1.5 8 pomembnih idej v računalniški arhitekturi (in širše)
- ❑ 1.6 Praktična realizacija računalnikov

8 pomembnih idej v računalniški arhitekturi (in širše) [Patt]

1. Moorov zakon

- Viri v integriranih vezjih se podvojijo na 18-24 mesecev

2. Abstrakcija kot poenostavitev

- Načrtovanje aparature in programske opreme, programski jeziki, podprogrami, ...

3. Pohitriti pogoste postopke

- Najbolj se splača pohitriti pogosto uporabljane koncepte

4. Višja zmogljivost s pomočjo paralelizma

- Glede na razvoj tehnologije je to edina pot

8 pomembnih idej v računalniški arhitekturi (in širše) [Patt]

5. Zmogljivost s cevovodnim procesiranjem

- Učinkovit, transparenten način pohitritve delovanja CPE

6. Zmogljivost s predvidevanjem

- „Raje delaj ob neki predpostavki, kot samo čakaj“

7. Hierarhični model pomnilnika

- Učinkovit kompromis med hitrostjo in ceno pomnilnikov v računalnikih

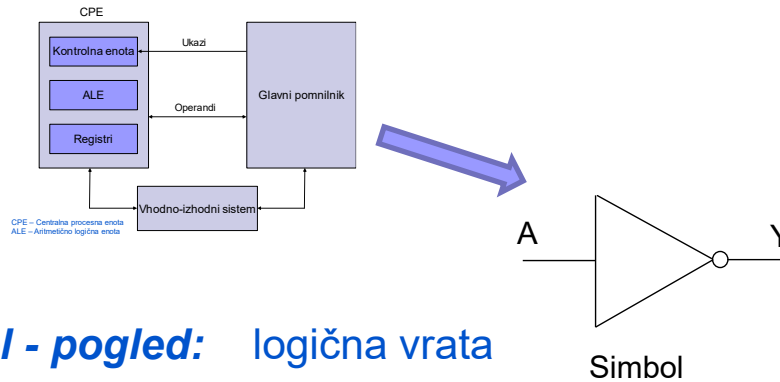
8. Zanesljivost z redundanco

- Strošek rezervnega sistema morda manjši od škode nedelovanja

1. Uvod :

- ❑ 1.1 Predmet RA
- ❑ 1.2 Računalniki včeraj in danes
- ❑ 1.3 Osnove zgradbe in delovanja računalnikov
- ❑ 1.4 Analogno – digitalno, zvezno diskretno
- ❑ 1.5 8 pomembnih idej v računalniški arhitekturi (in širše)
- ❑ 1.6 Praktična realizacija računalnikov

Teoretični model <-> Praktična realizacija



Teoretični model - pogled: logična vrata

logični nivoji 0,1

Matematični ideal

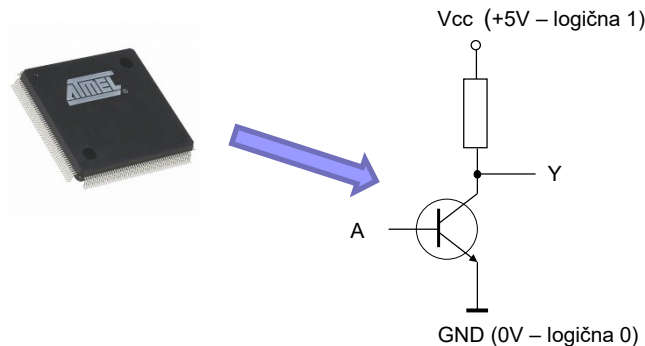
Elektronska realizacija : elektronsko vezje

napetostni nivoji $\approx 0V, \approx 3.3 (5) V$

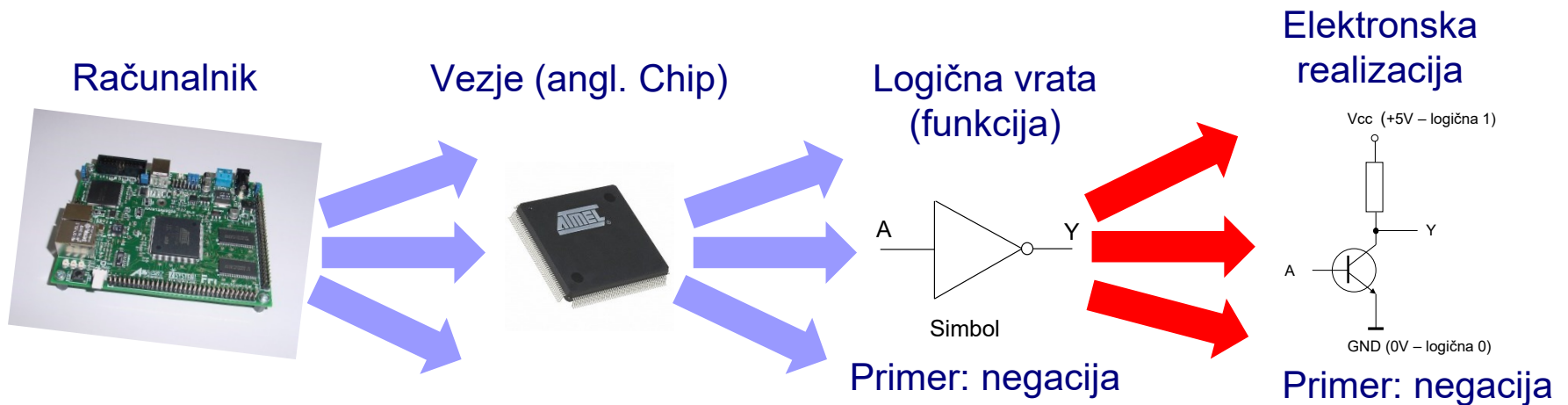
Elektronska realizacija

Slabosti :

- zvezne napetosti
- časovne zakasnitve
- motnje



Fizična zgradba računalnikov



Informacije (ukazi in operandi) so v računalniku predstavljene v dvojiški obliki, s pomočjo električnih signalov

- Dve stanji (simbola) 0 in 1 sta predstavljeni z dvema nivojema električne napetosti.

- Stanje 0** predstavljeno z nizko napetostjo ($\approx 0V$)
- Stanje 1** predstavljeno z visoko napetostjo (do +5V)

- Enostavna realizacija s stikalom – primer luči:

- Stanje 0** nizka napetost stikalo odprto
- Stanje 1** visoka napetost stikalo sklenjeno



- Eno stikalo je lahko v dveh stanjih, v stanju 0 ali 1.
- Količina informacije, ki jo eno tako stikalo predstavlja ali hrani, je 1 bit.
- Osnovno celico pomnilnika si lahko predstavljamo kot tako stikalo, ki navzven izkazuje svoje stanje in vanjo lahko shranimo 1 bit informacije. (0 ali 1)
- Če želimo shraniti več informacije, ne samo en bit, potrebujemo več celic.

Realizacije stikala v razvoju digitalnih računalnikov – razvoj tehnologije

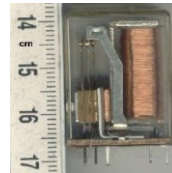
Predpone merskih enot

Oznaka	Ime	Vrednost	Zapis s potenco (znanstveni zapis)
p	piko	0,000 000 000 001	10^{-12}
n	nano	0,000 000 001	10^{-9}
μ	mikro	0,000 001	10^{-6}
m	mili	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}

Realizacija stikala kot osnovnega gradnika – povzetek :

□ Elektromehansko stikalo

- 1939: Rele,



čas preklopa 1-10ms

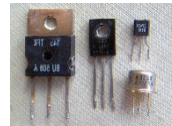
□ Elektronsko stikalo

- 1945-1955: Elektronka,



čas preklopa ~ 5μs

- 1955: Tranzistor → ,



čas preklopa ~10ns

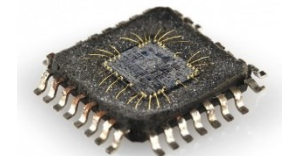
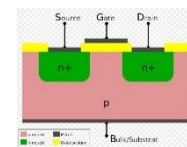
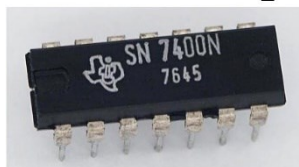
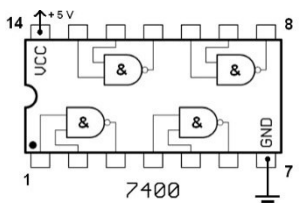
- 1958: Integrirano vezje - čip,

čas preklopa 2-10ns

- 1980: VLSI integrirana vezja

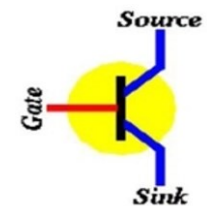
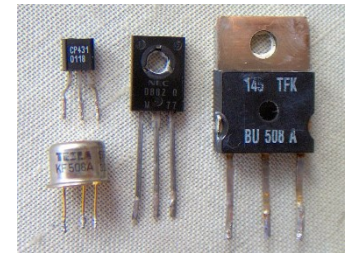
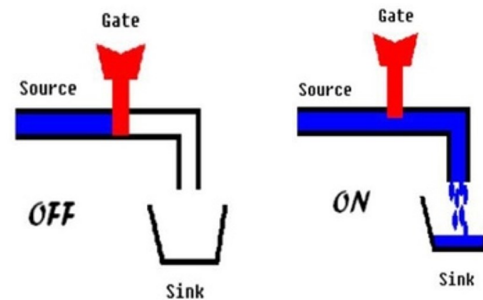
čas preklopa < 0.1ns

- Very Large Scale Integration

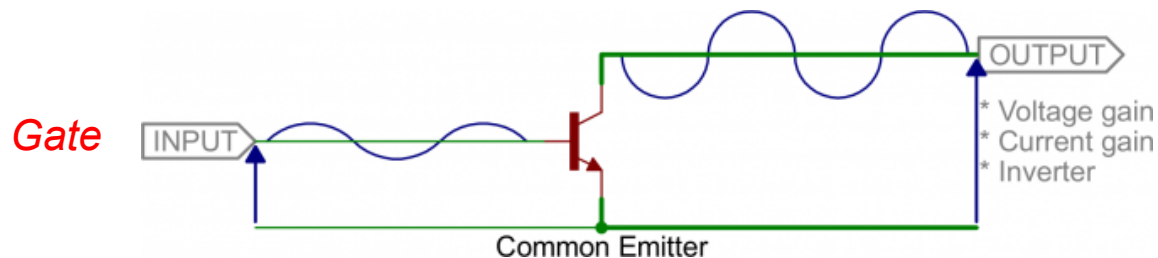


Tranzistor se lahko uporabi kot :

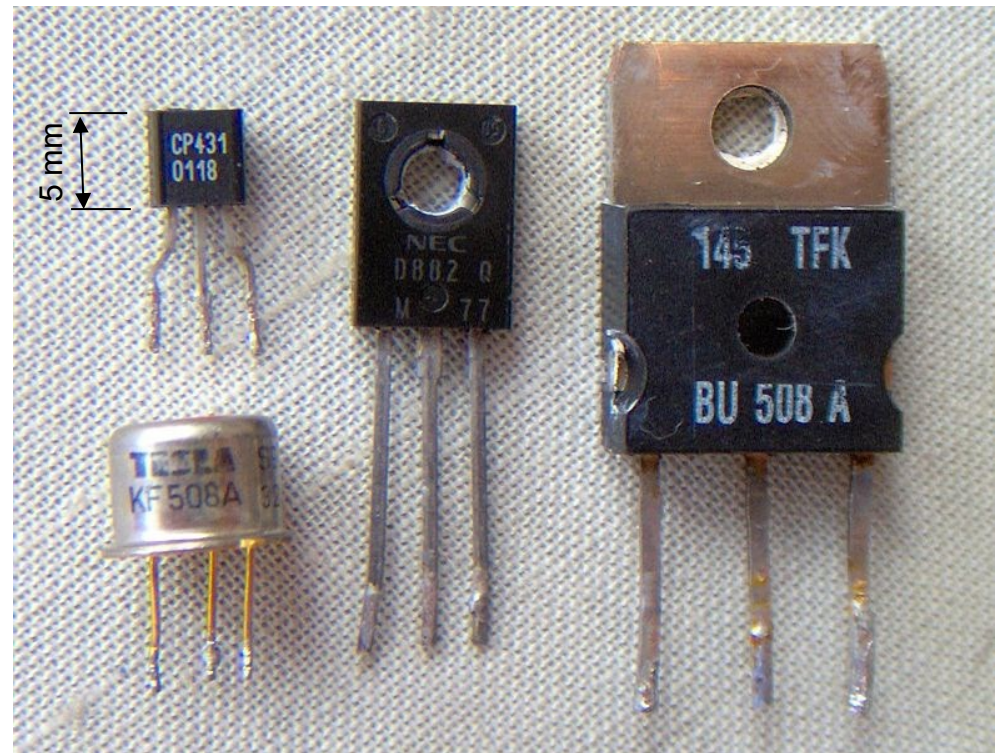
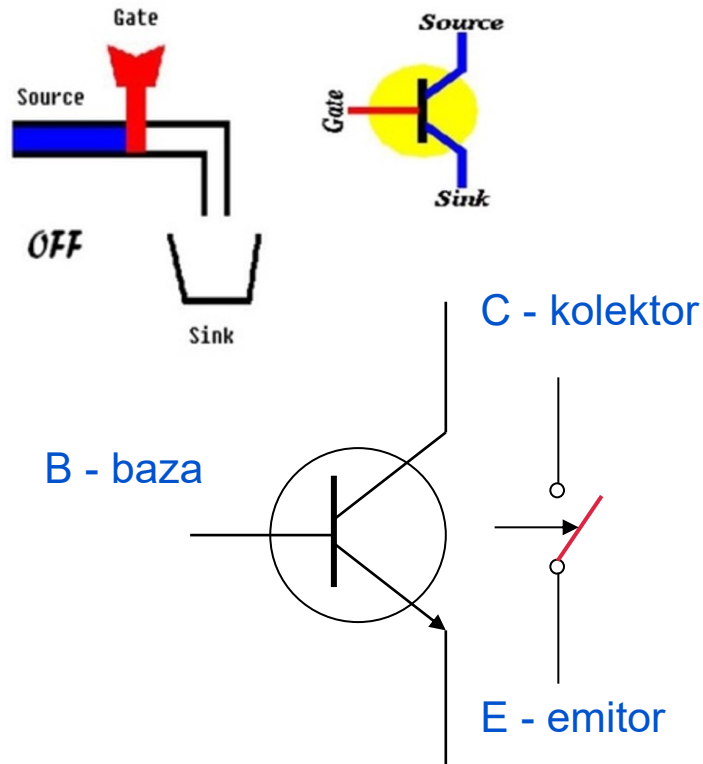
- Stikalo
 - v digitalnih vezjih



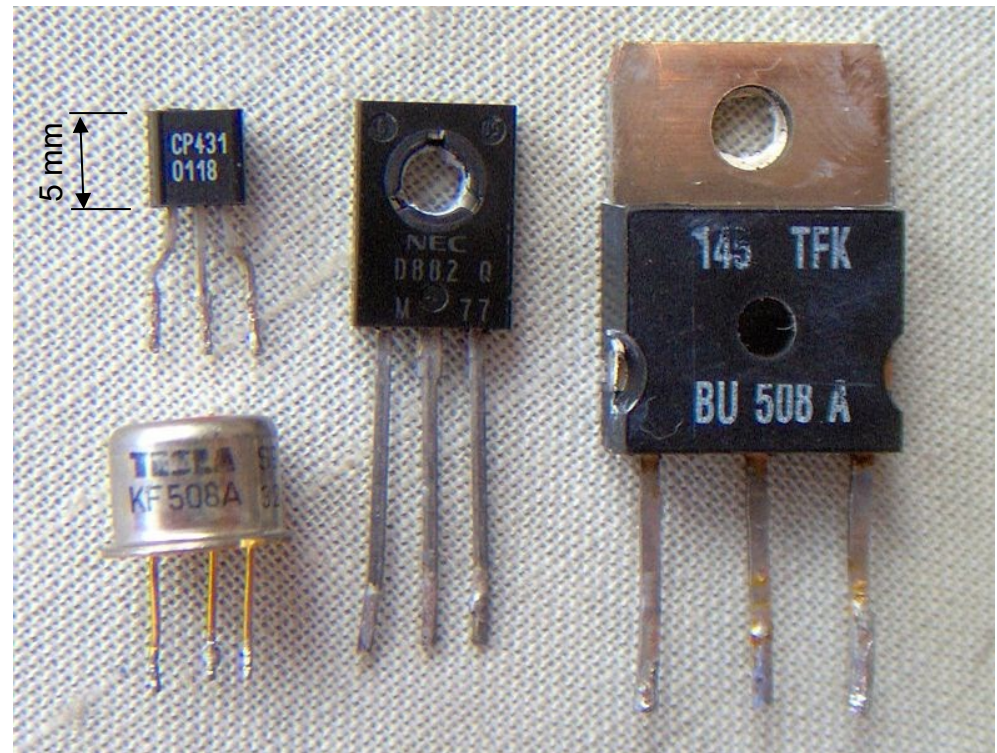
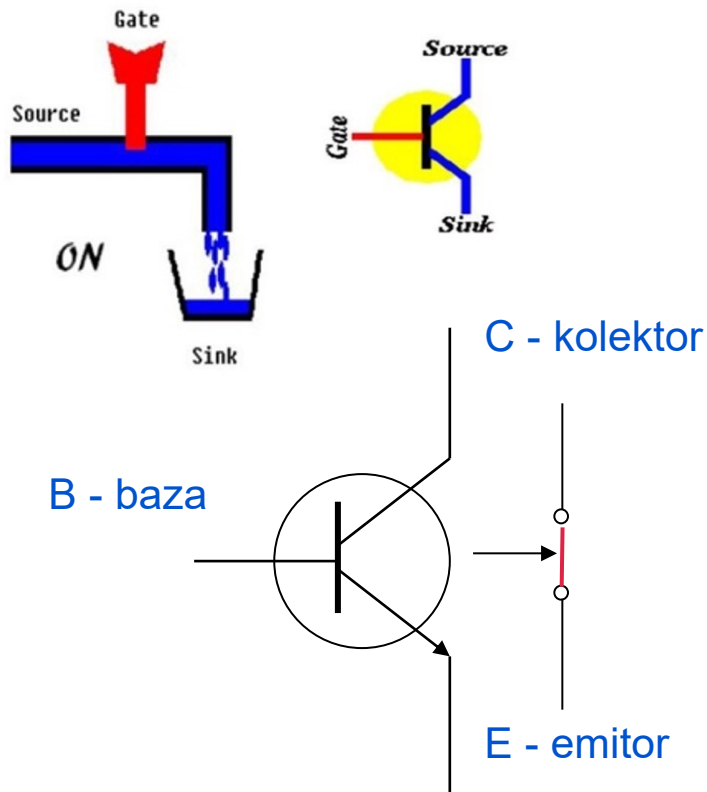
- Ojačevalnik signala
 - V elektronskih vezjih (ojačevalci)



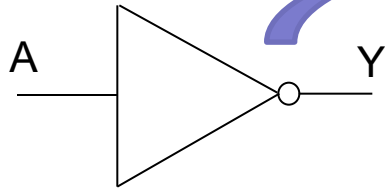
□ Delovanje tranzistorja kot stikala – izklop - OFF



□ Delovanje tranzistorja kot stikala – vklop - ON



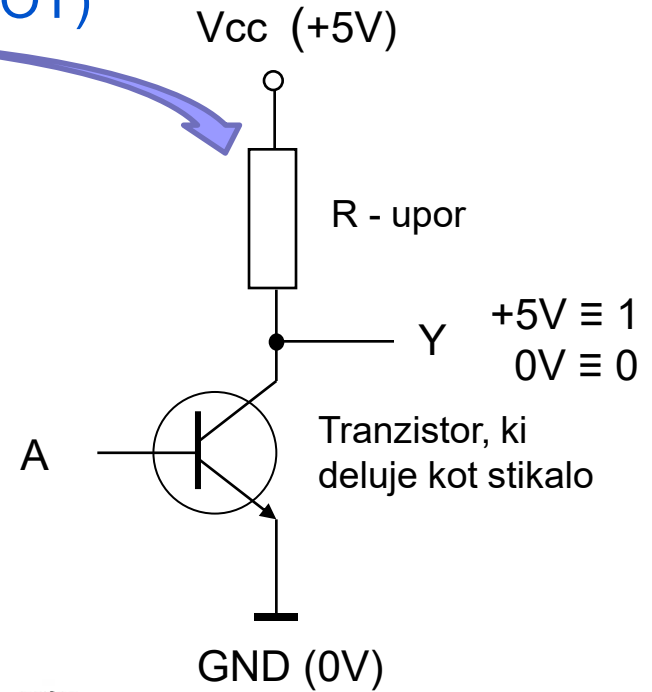
Realizacija logične funkcije NEGACIJA (NOT)



Simbol

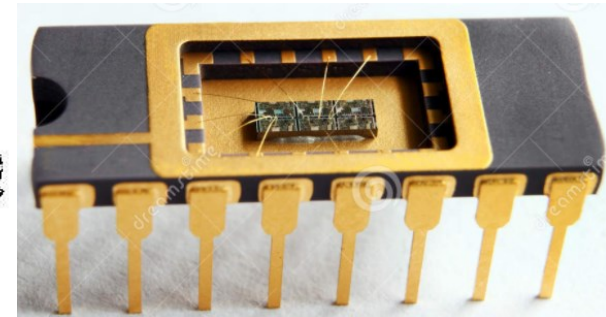
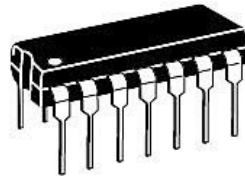
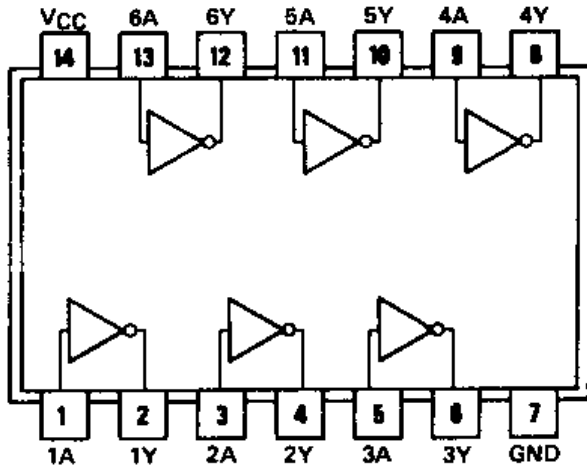
A	Y
0	1
1	0

Pravilnostna tabela

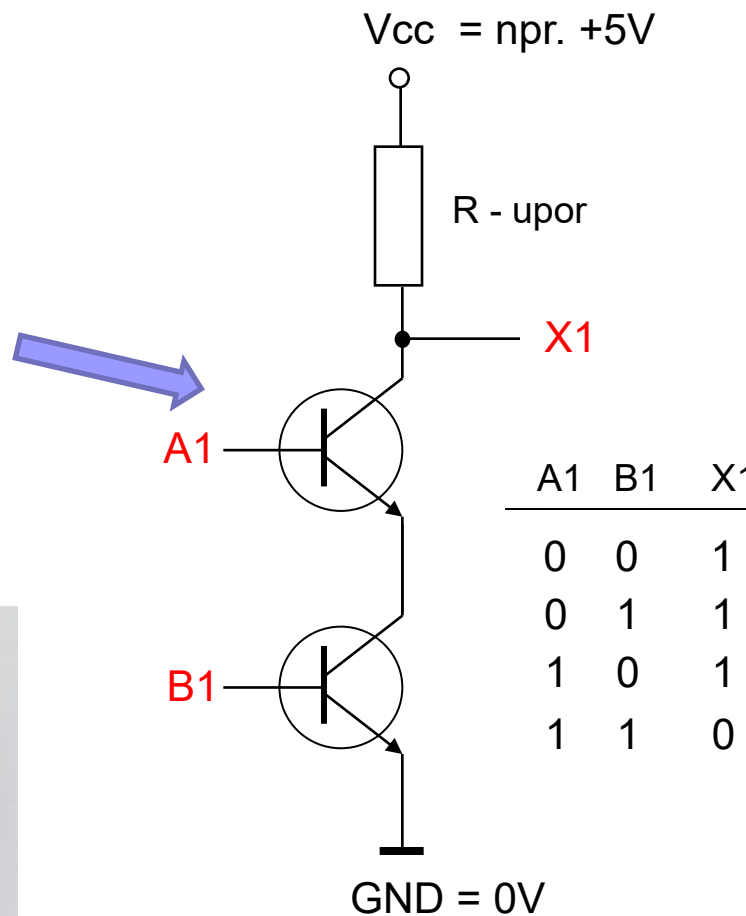
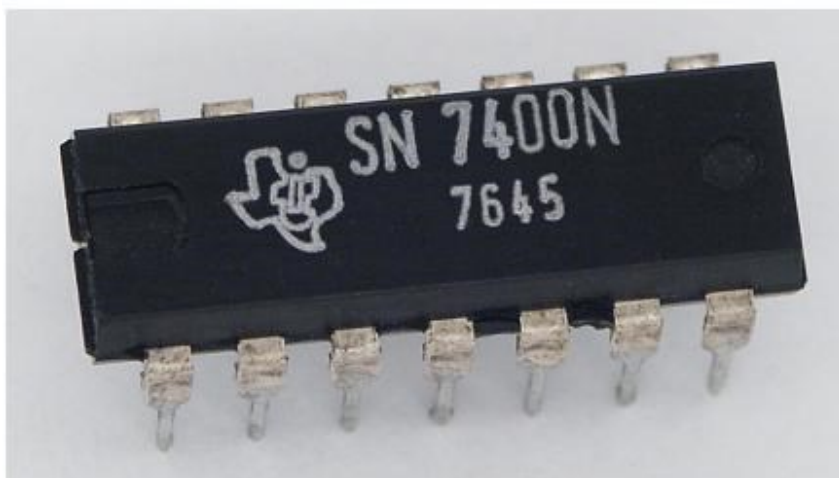
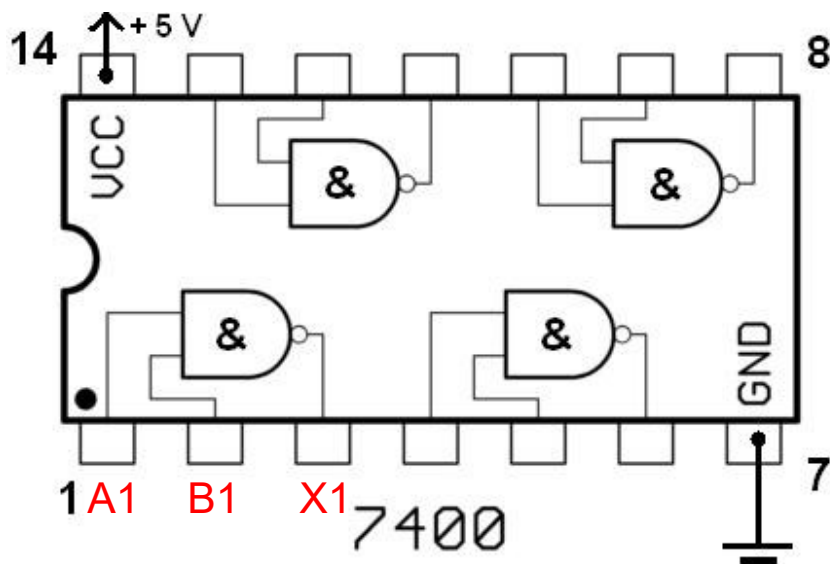


IC (Integrated Circuit) s 6 negatorji

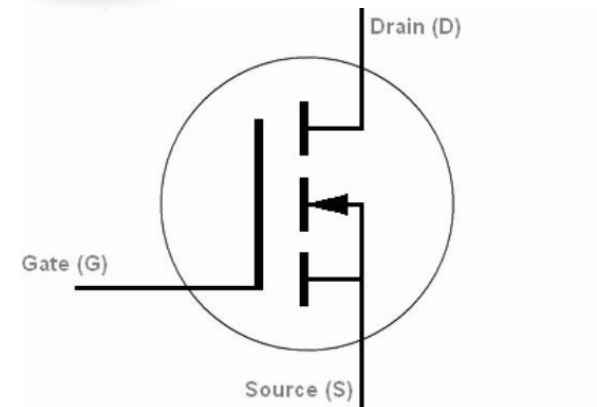
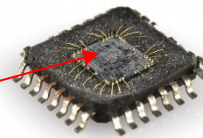
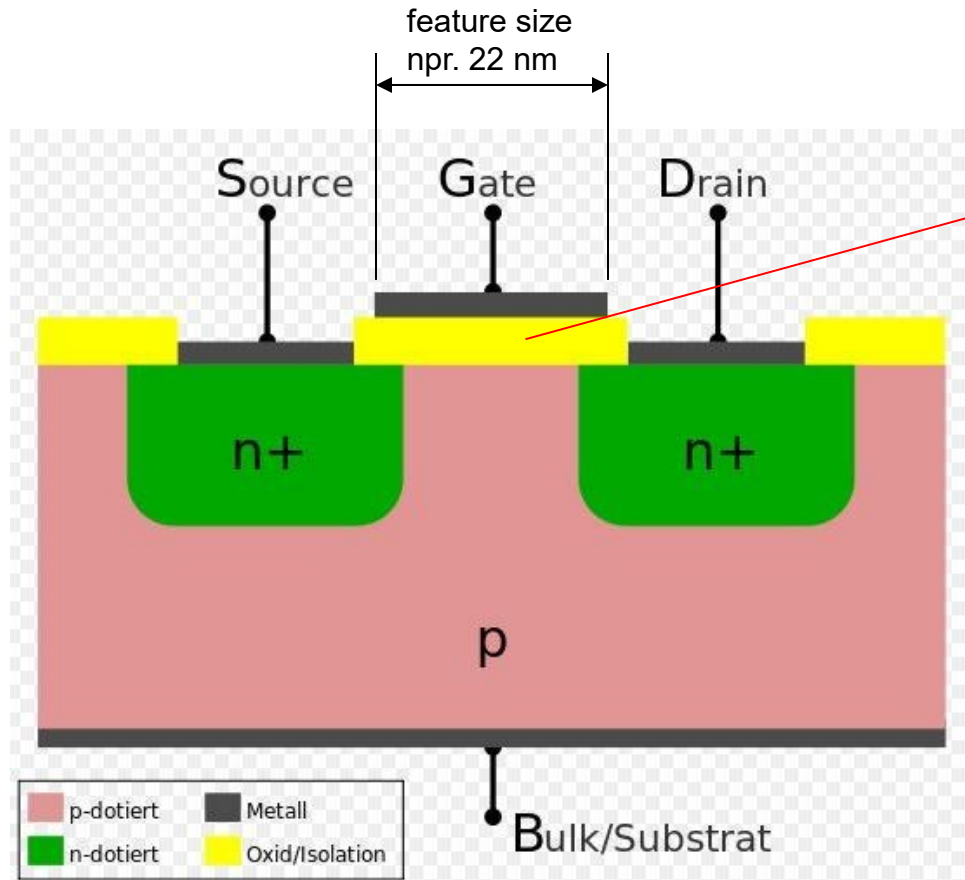
7406



Realizacija logične funkcije NAND (Negirana konjunkcija)

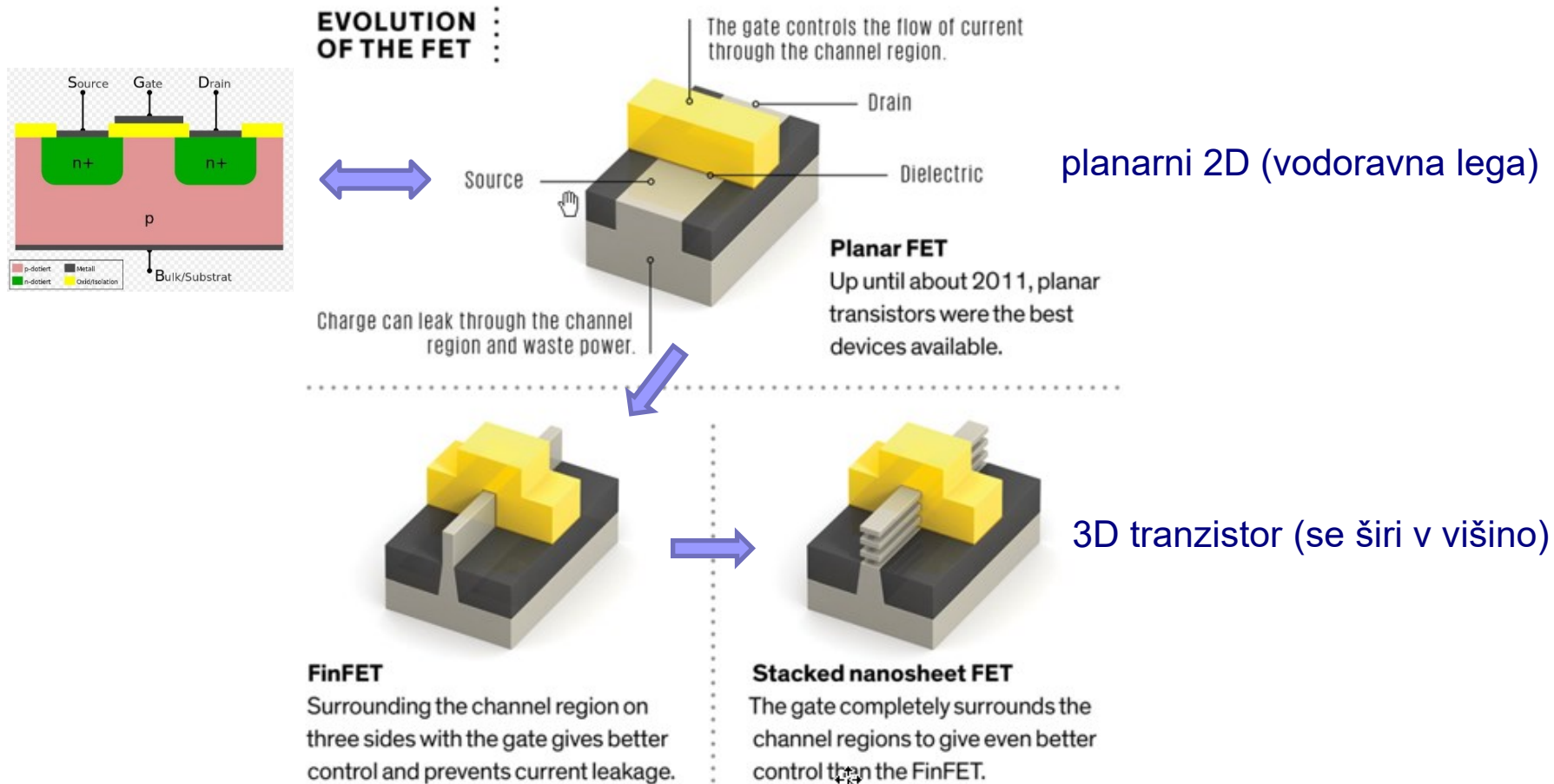


Tranzistor kot del integriranega vezja VLSI



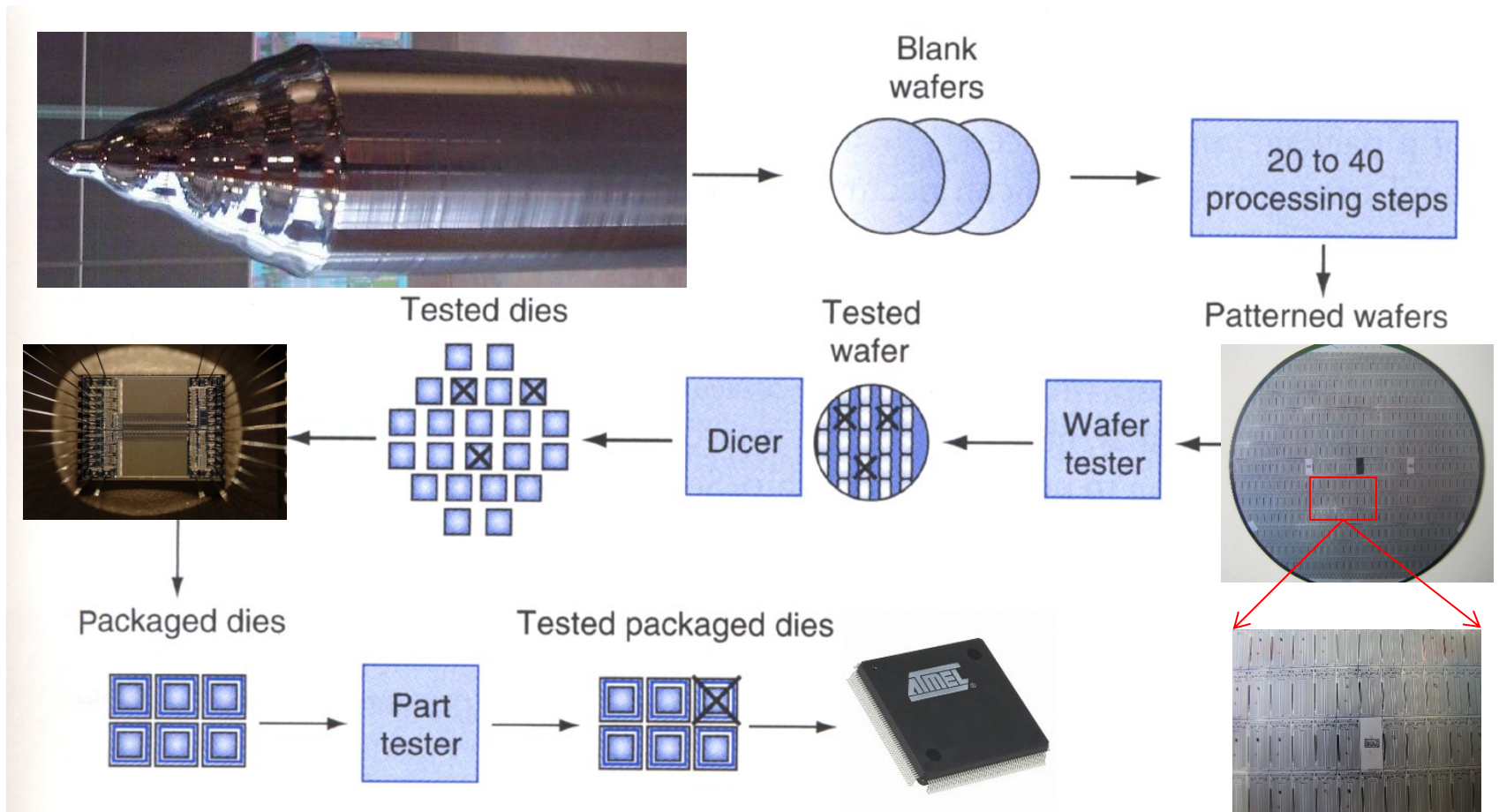
Razvoj tranzistorjev v najsodobnejših vezjih:

- prehod iz vodoravne (2D) v navpično obliko (3D) -> manjša površina, večja gostota !!!



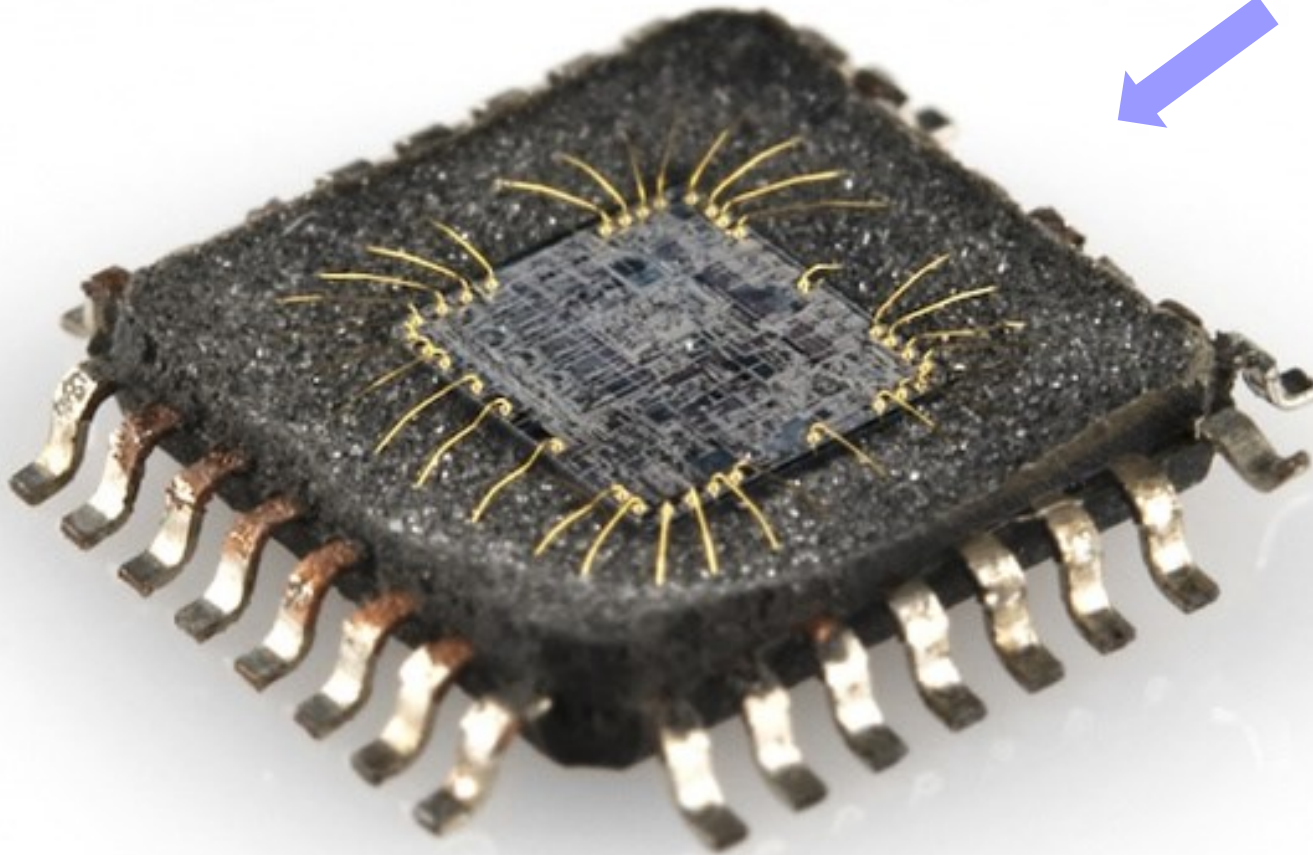
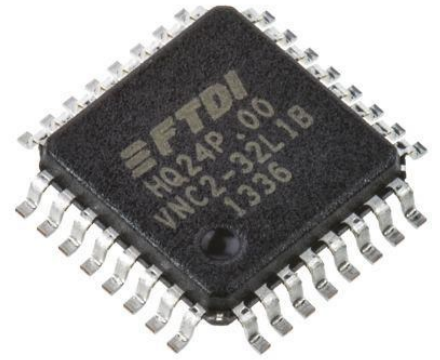
Premer Si atoma je 0.24nm!!!

Postopek izdelave VLSI čipa



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

VLSI čip - notranjost



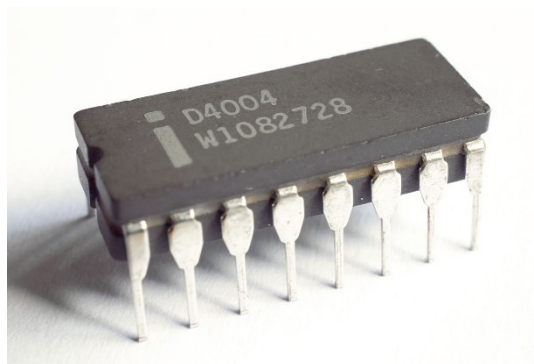
- ? nm proces („feature size ? nm“)
 - Parameter **feature size** pri integriranih vezjih v največji meri določa število tranzistorjev na integriranem vezju in tudi njihove lastnosti
 - Določa **najmanjšo možno velikost** kateregakoli objekta na integriranem vezju
 - Objekt je lahko del tranzistorja, povezovalna žica, presledek med dvema objektoma. Celoten tranzistor je običajno večji
 - Število tranzistorjev na čipu je odvisno od površine, ki jo zaseda tranzistor, zato se **število tranzistorjev povečuje s kvadratom zmanjševanja parametra feature size**

■ problemi sodobnih VLSI tehnologij

- hitrost preklopa tranzistorjev narašča zelo počasi
- hitreje narašča gostota tranzistorjev (elementov) -> PARALELIZEM
 - hitrost naraščanja gostote je vse bolj omejena
- dimenzije elementov vedno manjše -> TEŽAVE
 - sprošča se odvečna toplota -> ODVAJANJE, HLAJENJE
 - manjša odpornost na motnje

Primer 1:

- prvi procesor na enem čipu **Intel 4004** (leto 1971)
 - **2.250** tranzistorjev na ploščici 3,2 x 4,2 mm
 - **10 μm** proces (feature size 10 μm = 10×10^{-6} m = 0,00001 m, človeški las ima premer približno 100 μm)
 - **16** kontaktov
 - Izvedba ukaza 10,8 μs (= 0,0000108 s) ali 21,6 μs
 - Poraba **1,0 W**
 - Cena (preračunana na današnja razmerja) \$26

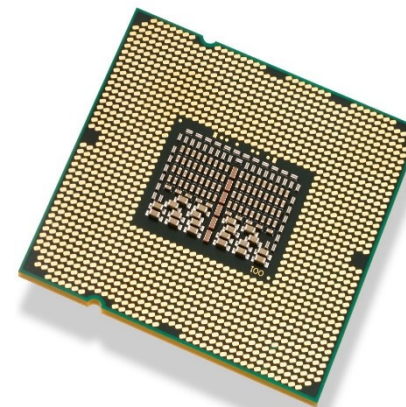
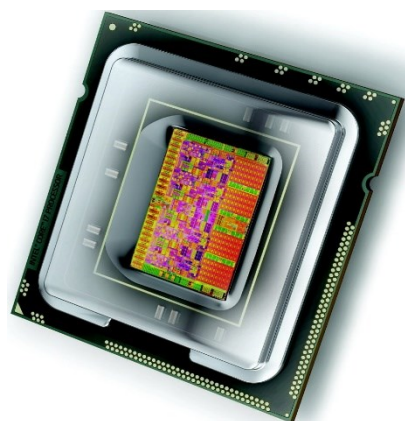
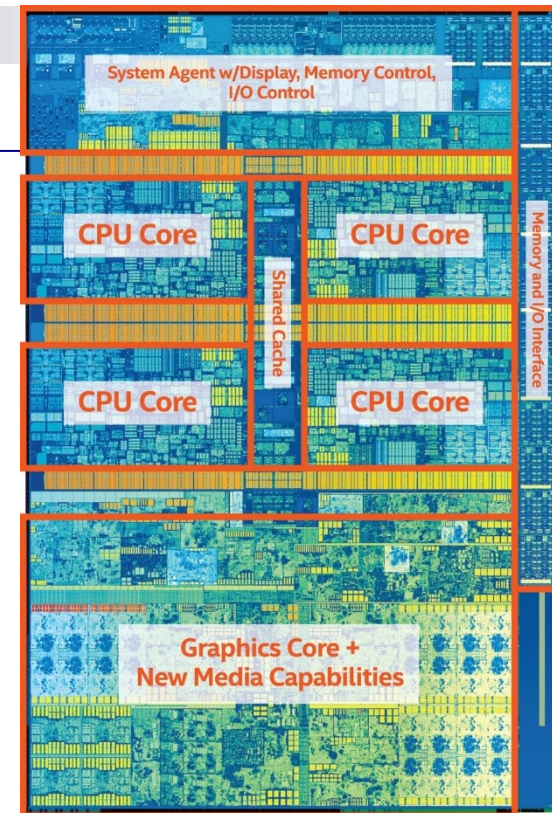


Primer 2:

■ Procesor Intel i7 7700

(mikroarhitektura Kaby Lake 7. generacija leto 2017):

- Število transistorjev - Intel tega podatka ne objavlja več
- **14 nm** proces ($14\text{nm} = 14 \times 10^{-9} \text{m} = 0,000000014 \text{m}$)
- Velikost čipa - Intel tega podatka ne objavlja več
- **4 jedra** (4 procesorji, 8 niti), grafični procesor
- **1155 kontaktov**
- Poraba (TDP) **65 W**
- Priporočena cena (Intel) 303 \$ - 312 \$

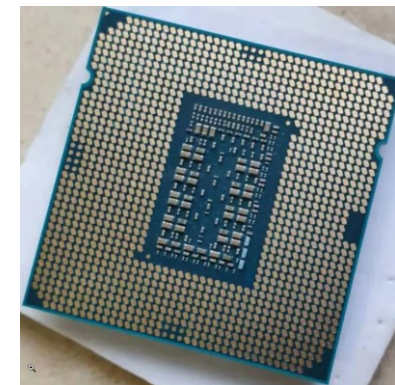


Primer 3:

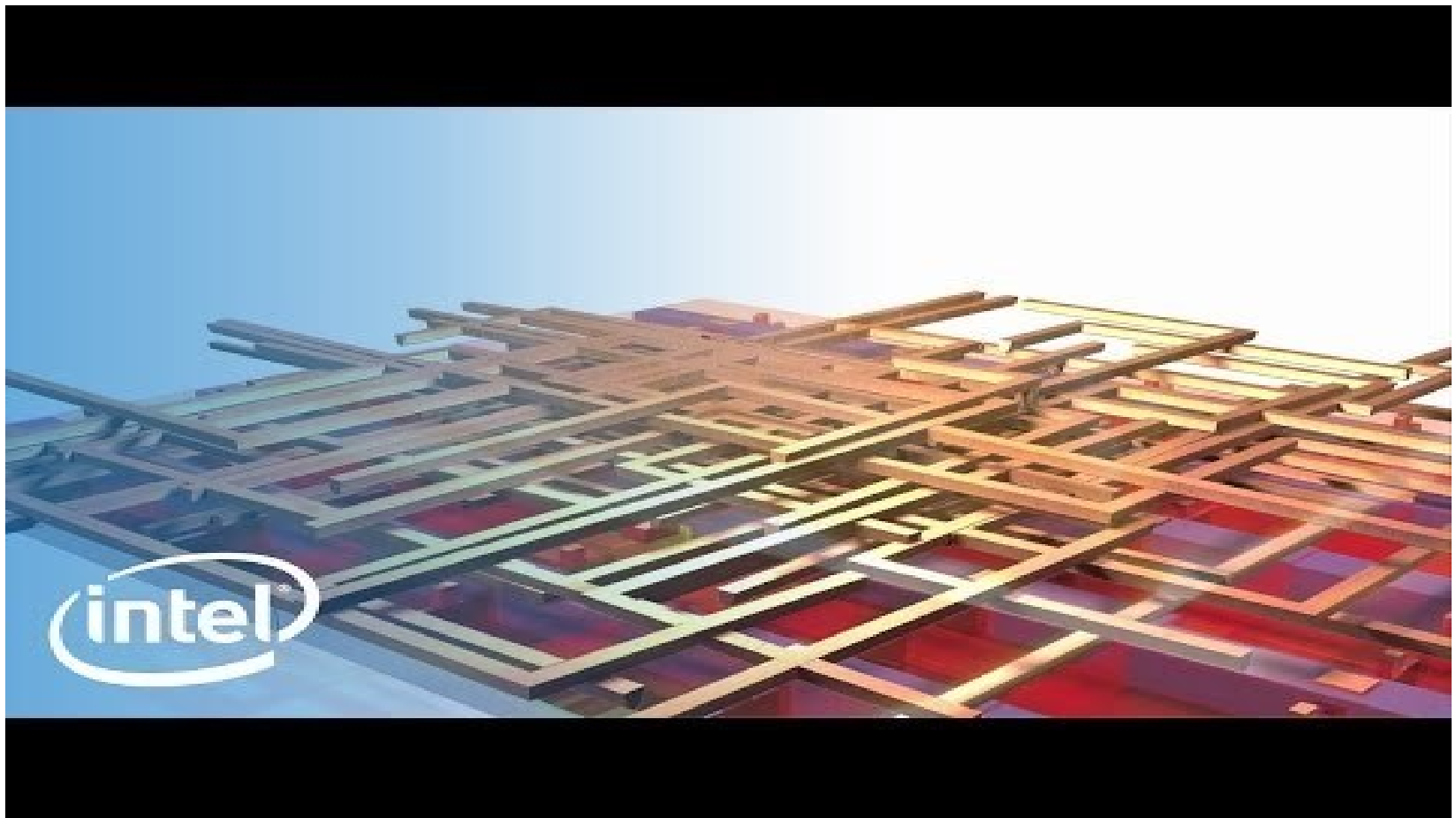
■ Procesor Intel i9-11900

(mikroarhitektura Rocket Lake 11. generacija leto 2021):

- Število transistorjev - Intel tega podatka ne objavlja več
- **14 nm** proces ($14\text{nm} = 14 \times 10^{-9} \text{m} = 0,000000014 \text{m}$)
- Velikost čipa - Intel tega podatka ne objavlja več
- 8 jeder (16 niti), grafični procesor
- **1200 priključkov**
- Poraba (TDP) **65 W**
- Priporočena cena (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

Sprotni kvizi na predavanjih (po zaključkih predavanj):

- Vsakič se zabeležijo „stopničke“
- Uporaba vpisne številke – sledimo vašo uspešnost
 - Lahko zviša končno oceno, ne more je znižat...
- Možnost vprašanj, aktivnega sodelovanja

RA-1 | My Space #RA1
26 Oct – 10 Nov 2022 Engage Your event has finished Share Present

Live polls Audience Q&A **Analytics**

Slido overview

Share Analytics

