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

- Spletne strani: http://ucilnica.fri.uni-lj.si
  https://padlet.com/rawall/RAWall
- MS Teams
  - Koda za vstop: kygg8n7
- Govorilne ure: trenutno torek ob 18:15 v R2.40,50
  Občasne spremembe bodo pravočasno objavljene na učilnici

Ekipa RA

Tutorji

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

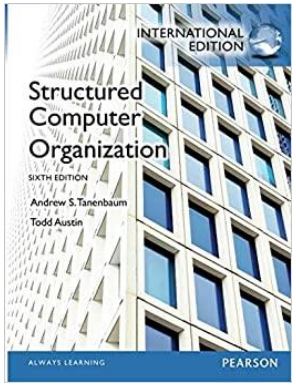
- Literatura (knjige na voljo tudi v knjižnici FRI)::
  - Vsebina predavanj, lab. vaj in prosojnic
    - http://ucilnica.fri.uni-lj.si
  - MS Teams (komunikacija, zapiski s table)
  - Skupni (deljeni) zapiski – Gdocs

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



Ocene STUDIS

Izhodišča :

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

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]¤

Mnenja slušateljev (2018/21) - 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“

Novosti - posebnosti 2022:

- Živa izvedba predavanj in vaj
  - https://padlet.com/rawall/RAWall
    - 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

What‘s new in 2022:

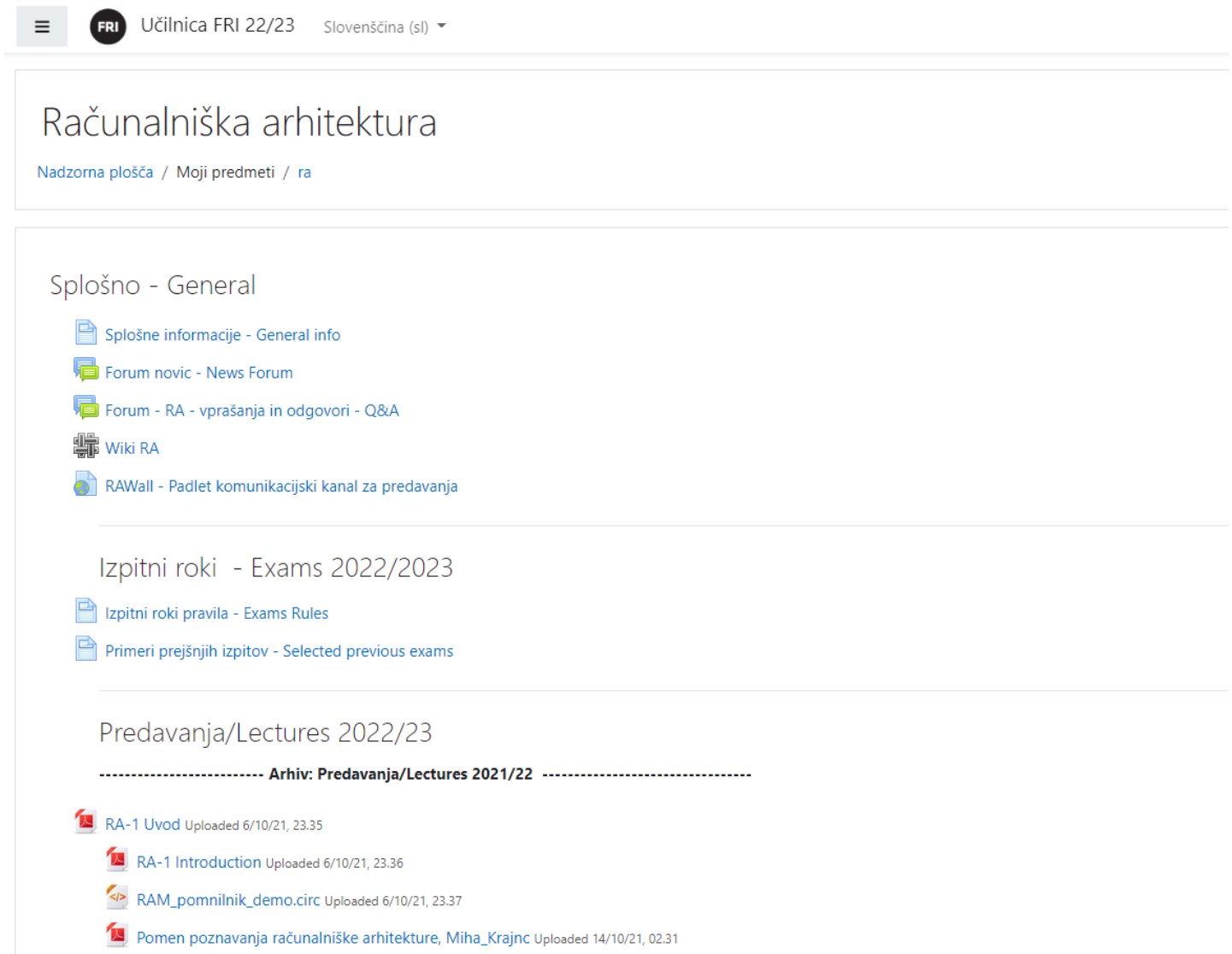
- Live lectures and lab sessions (one lab session in English)
  - https://padlet.com/rawall/RAWall
    - Questions, challenges, links, …
- Platforms :
  - e-classrom http://ucilnica.fri.uni-lj.si

    Computer Architecture - Crowd-sourced Shared Notes
  - MS Teams (board notes, communication)
    - Team entry code : kygg8n7

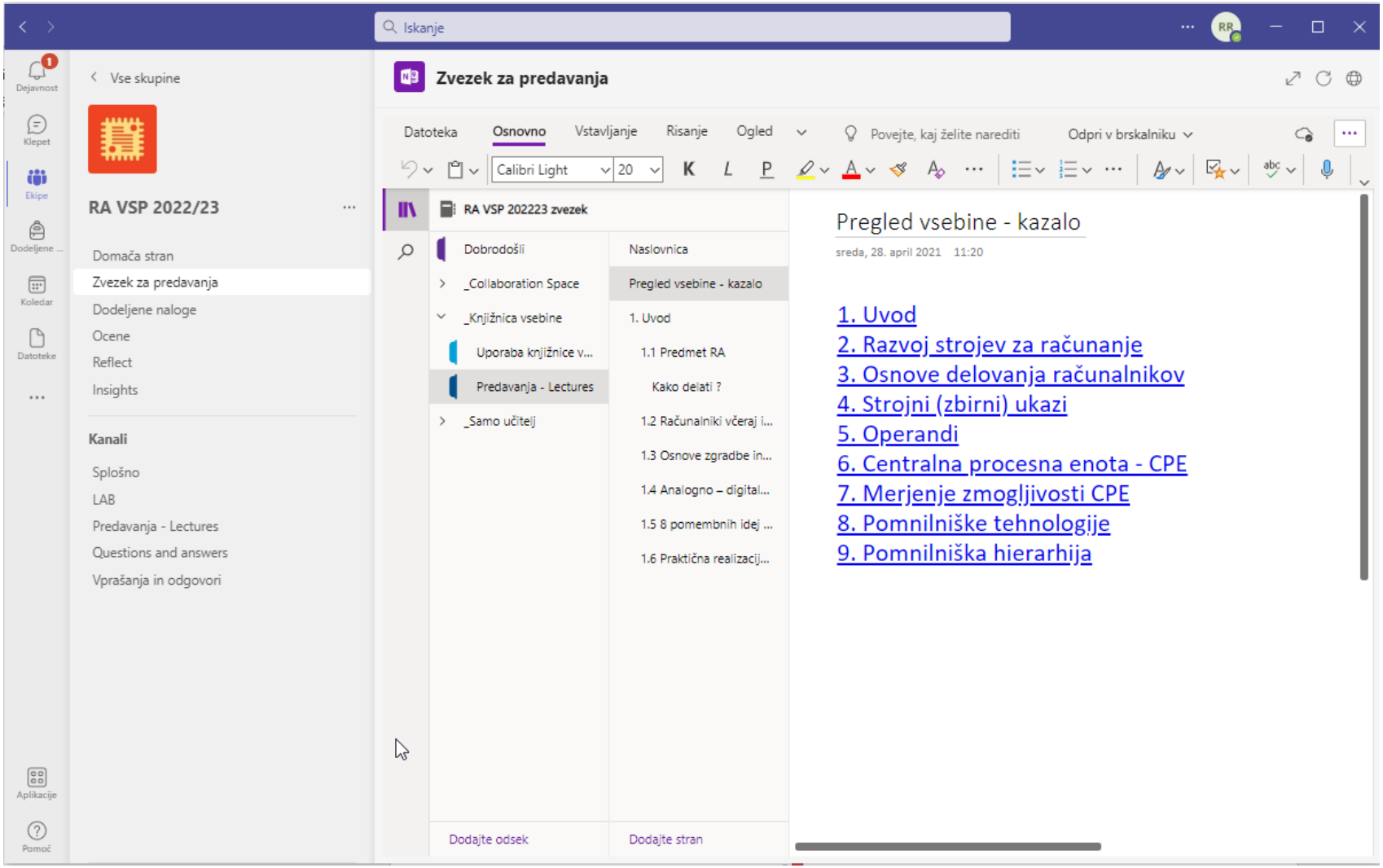
- Important :
  - be active
  - cooperate, talk, ask, comment, …
  - all major documents are translated to English
  - testing of realtime Slovene-English translation – project ON
  - please help us on English documents (typos, missing content, …)

E-učilnica : vsebina predmeta RA

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

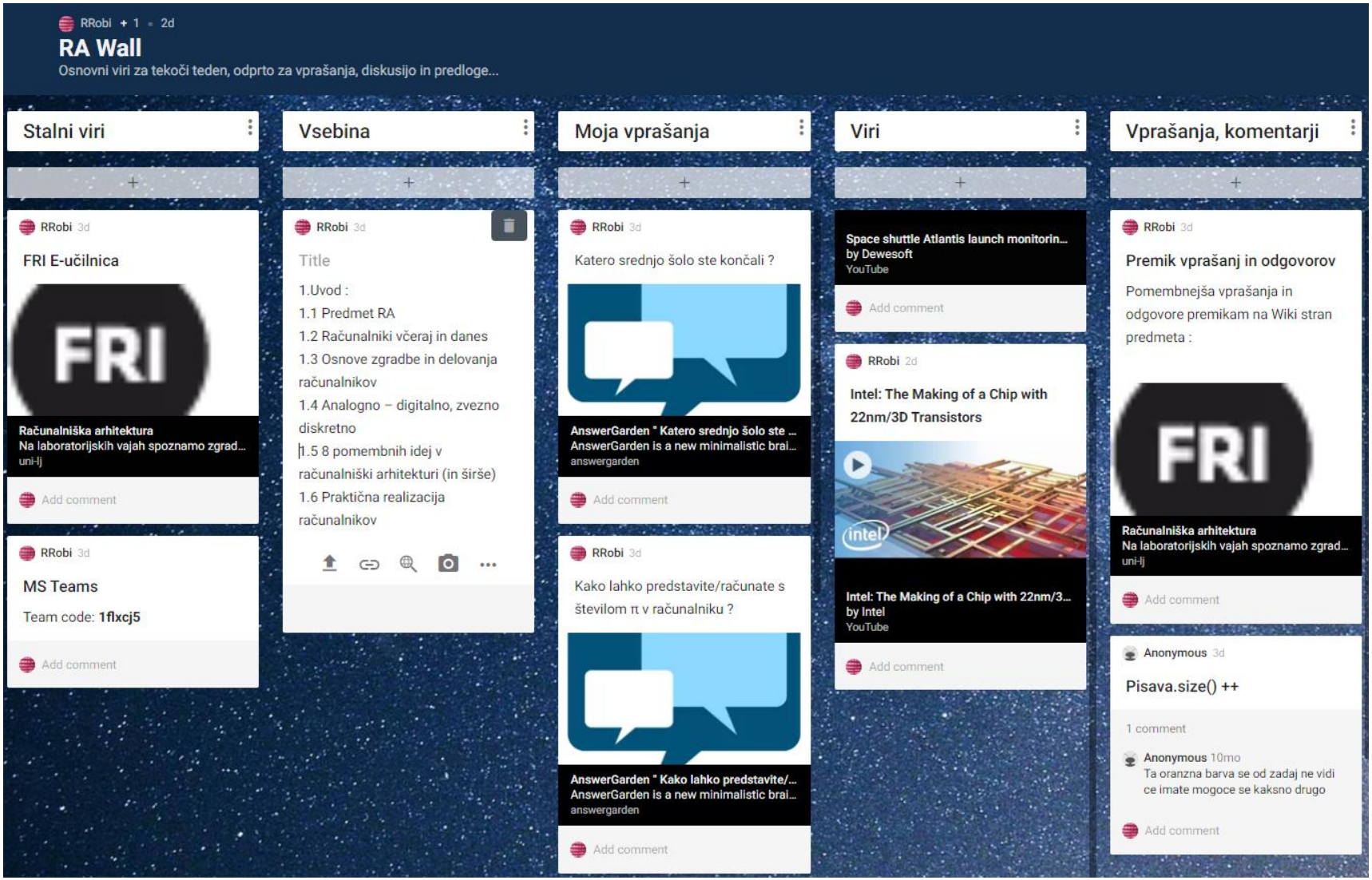


MS Teams: Komunikacija, OneNote zvezek



Tedenska vsebina

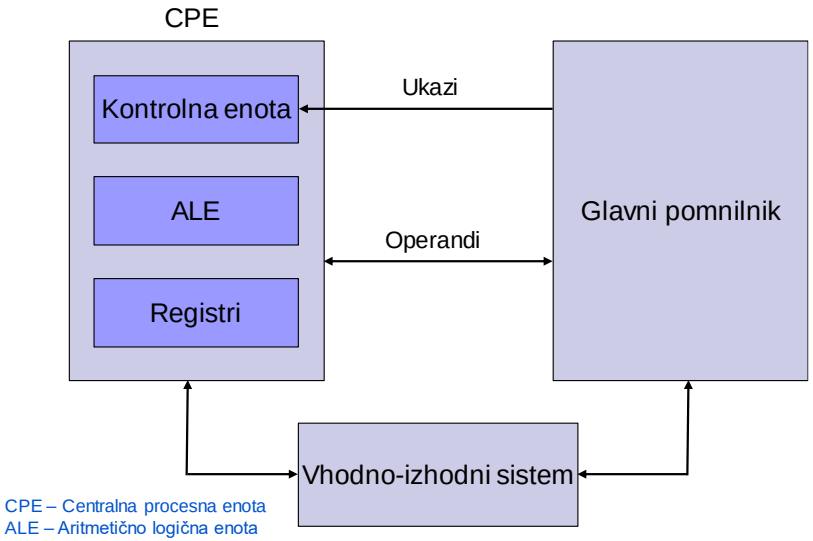
https://padlet.com/rawall/RAWall



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

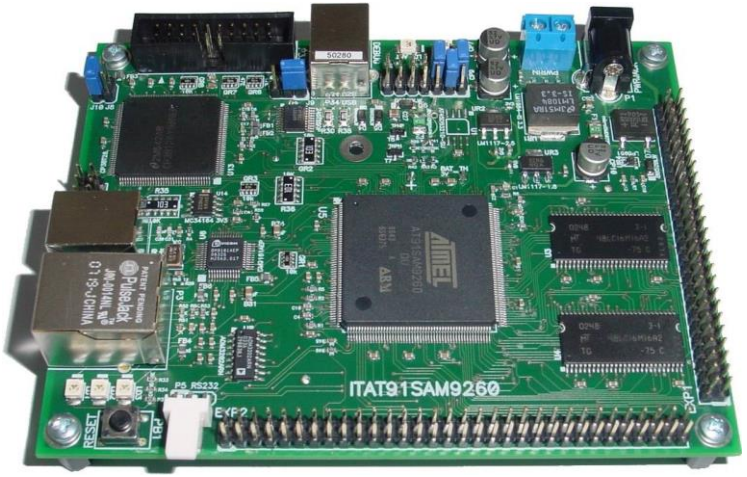
Vsebina laboratorijskih vaj:

- 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

Nadgradnja -> predmet Organizacija računalnikov, Vhodno izhodne naprave in ostali sorodni predmeti

Računalnik STM32H750-DK

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



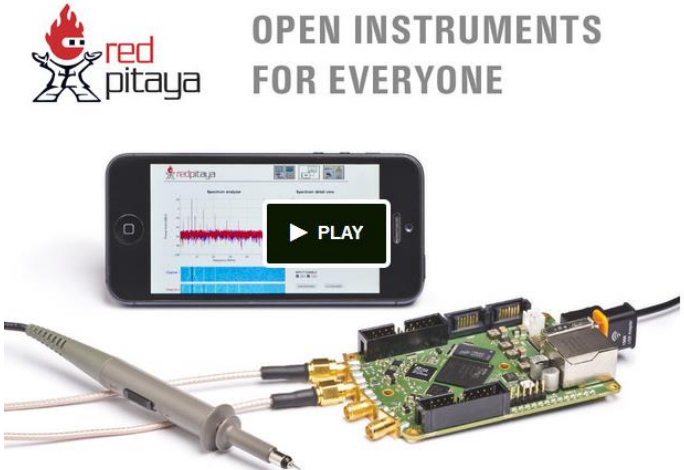
Zakaj je računalniška arhitektura pomembna ?

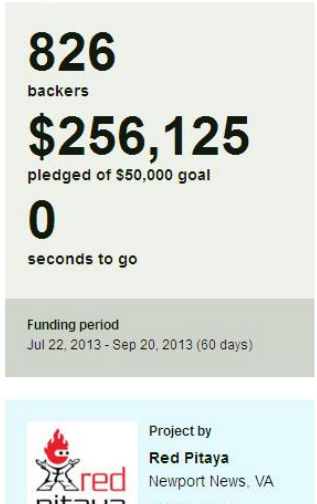
- 4 vprašanja in
- 4 odgovori

1. Produkti z aparaturno in programsko opremo

Primeri uspešnih projektov/podjetij (HW+SW)

COSYLAB





, FRI

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 hitrejše 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 Črnila, Dewesoft]:

Past Meetup

Code optimization on modern processors [Dejan Črnila, 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 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

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.

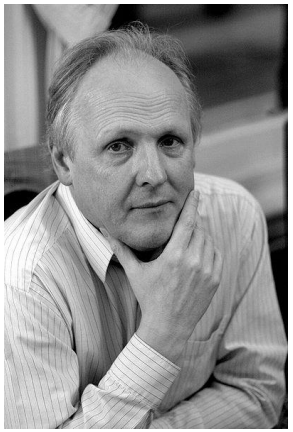
Dodatno gradivo (e-učilnica): „Pomen poznavanja računalniške arhitekture“, avtor Miha Krajnc.

4. Zakaj ravno ARM arhitektura?

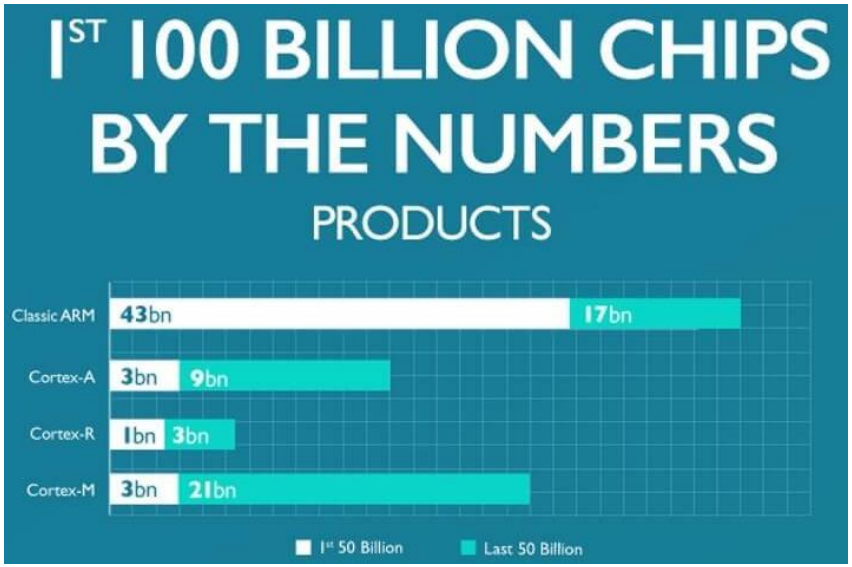
Ker ??? …“

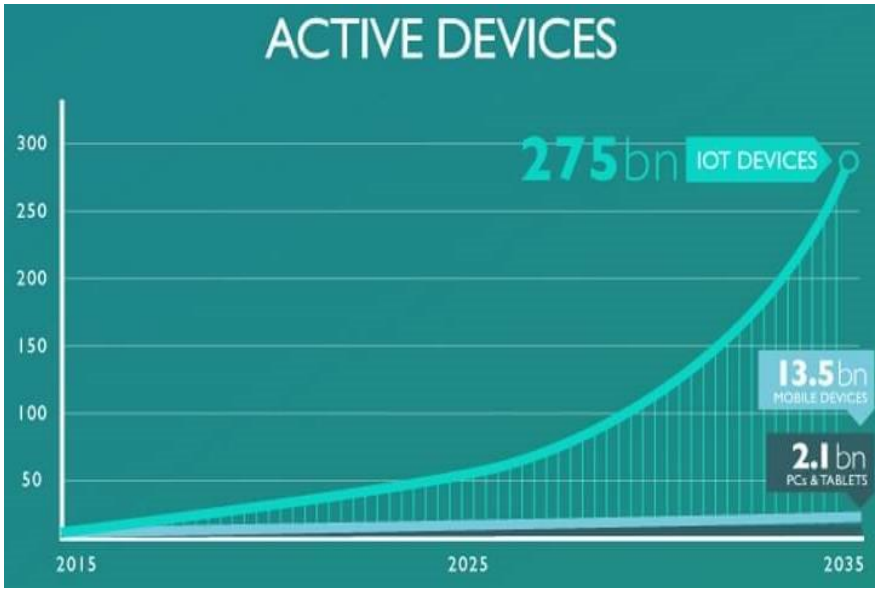
4. Zakaj ravno ARM arhitektura?

„Steve Furber na FRI“



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





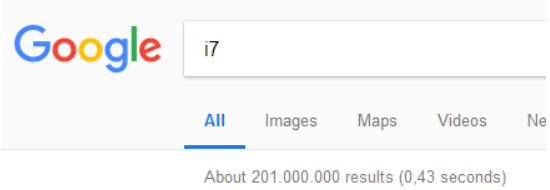
https://community.arm.com/processors/b/blog/posts/inside-the-numbers-100-billion-arm-based-chips-1345571105

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 (tretja revolucija v naši civilizaciji)
- 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)
  - Digitalni Asistenti



- Med računalniki so
  - v izvedbi velike razlike:
    - Superračunalnik
    - Enostaven računalnik na enem čipu
  - v zgradbi so razlike precej manjše

Slabosti ?
Prednosti ?

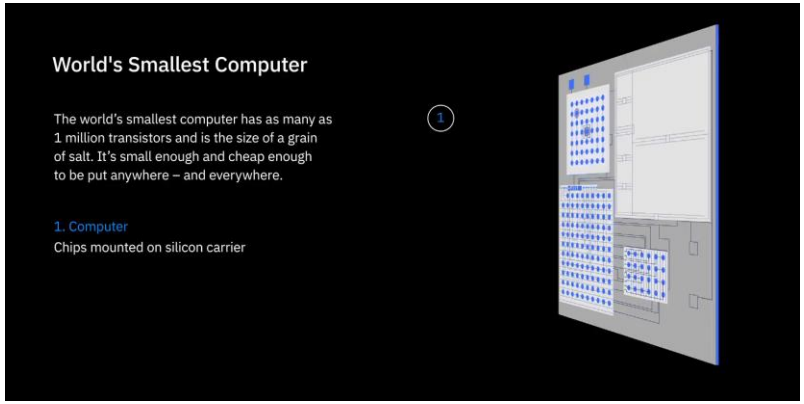
- Z vsakim, tudi najenostavnejšim, pa lahko izračunamo vse, kar se da izračunati (je izračunljivo).

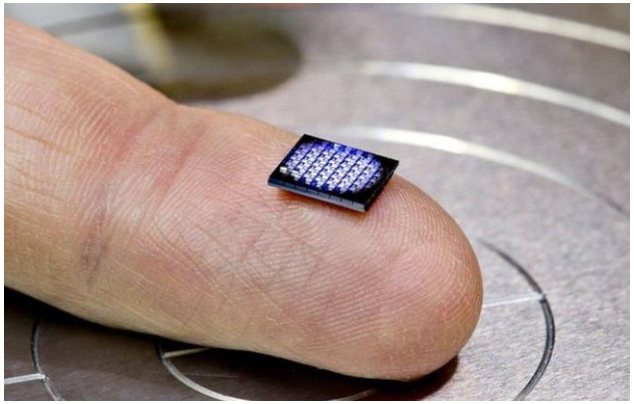
- Trenutno najzmogljivejši računalnik na svetu :
  - SUPERCOMPUTER FUGAKU v mestu Kobe, Japonska
  - 7 630 848 jeder
  - Zmogljivost 537 212 TFLOPS
  - Poraba energije 29 899 kW (*HE Medvode 26 700 kW*)

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

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

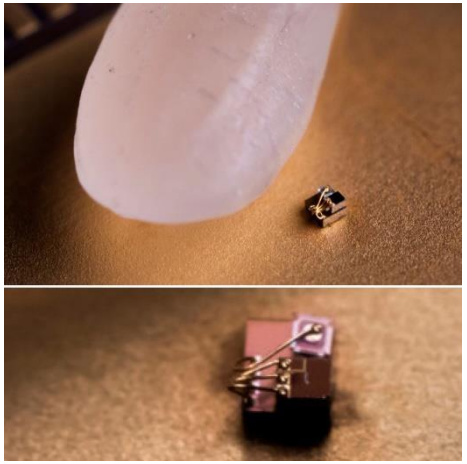
- Eden najmanjših računalnikov na svetu (l. 2018):





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

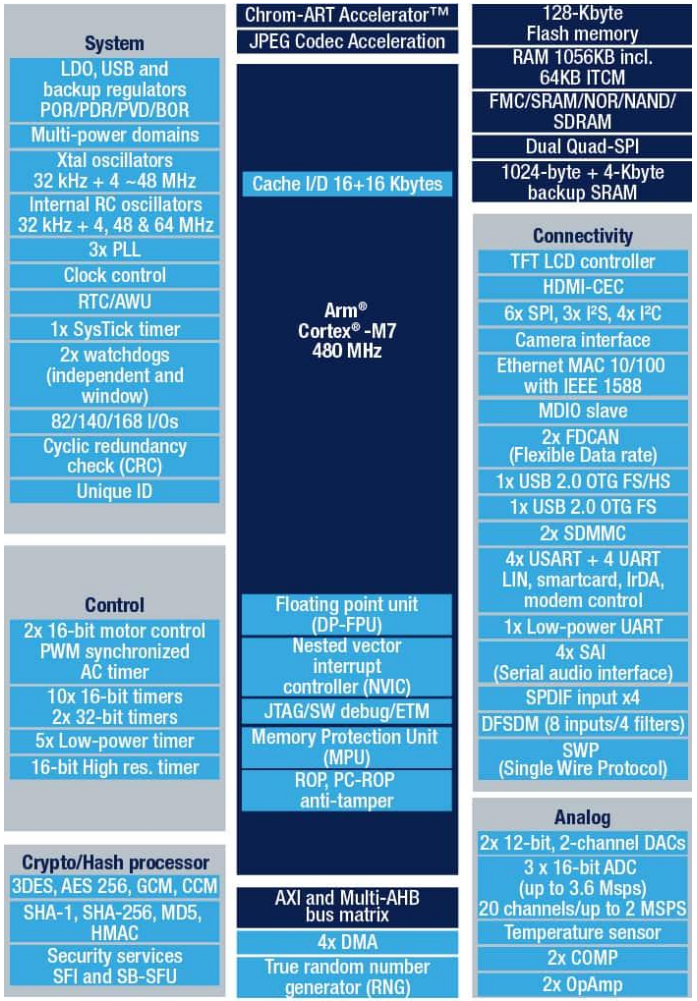
The university on Thursday said its engineers have produced a computer that's 0.3 mm x 0.3 mm -- it would be dwarfed by a grain of rice. While it drew comparisons to IBM's own 1mm x 1mm computer, Michigan's team said the creation is about more than just size.



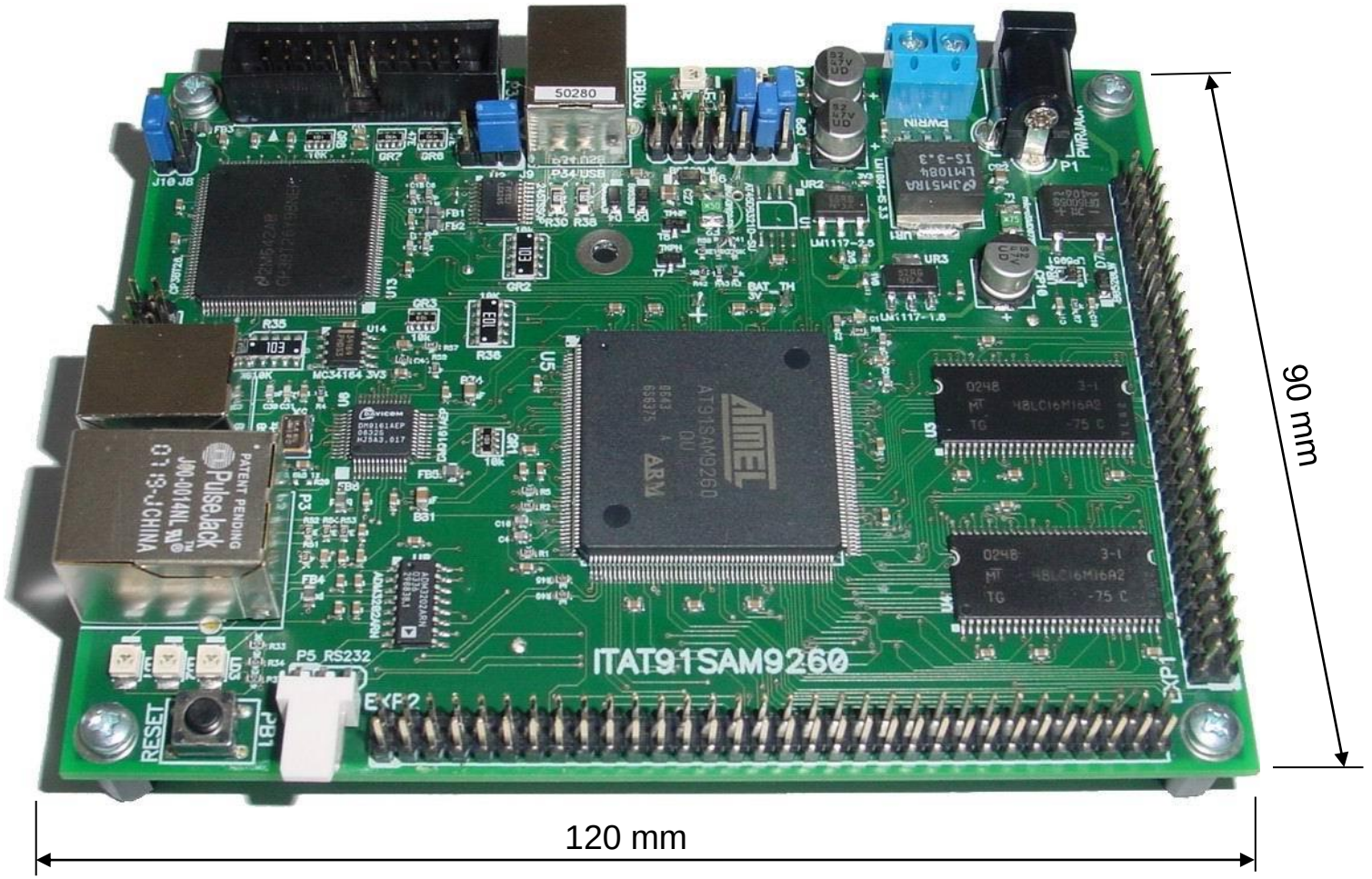
- + Nizka poraba
- - Manjša zmogljivost

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



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

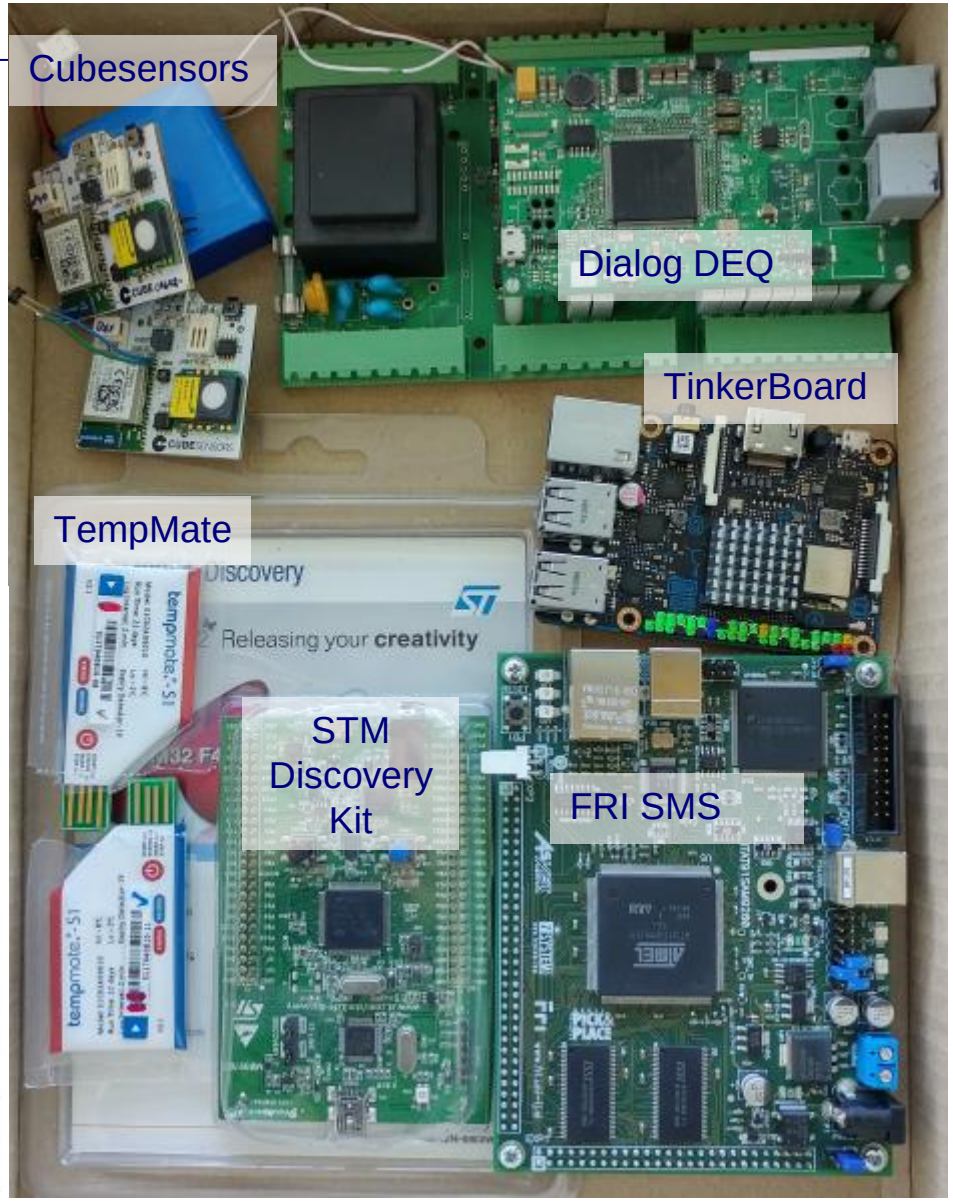


- V današnjem času se je delitev računalnikov poenostavila na tri kategorije po funkcionalnosti :
  - Osebni računalniki (namizni, tablični, . . .)
  - Strežniki
    - Med strežniki so velike razlike v ceni in zmogljivosti
      - Malo zmogljivejši namizni računalnik na spodnjem nivoju
      - Superračunalnik s terabajti glavnega pomnilnika in petabajti zunanjega 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, …

Vgrajeni (embedded) računalniki

(konkretni primeri)

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



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 aparaturnemu (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

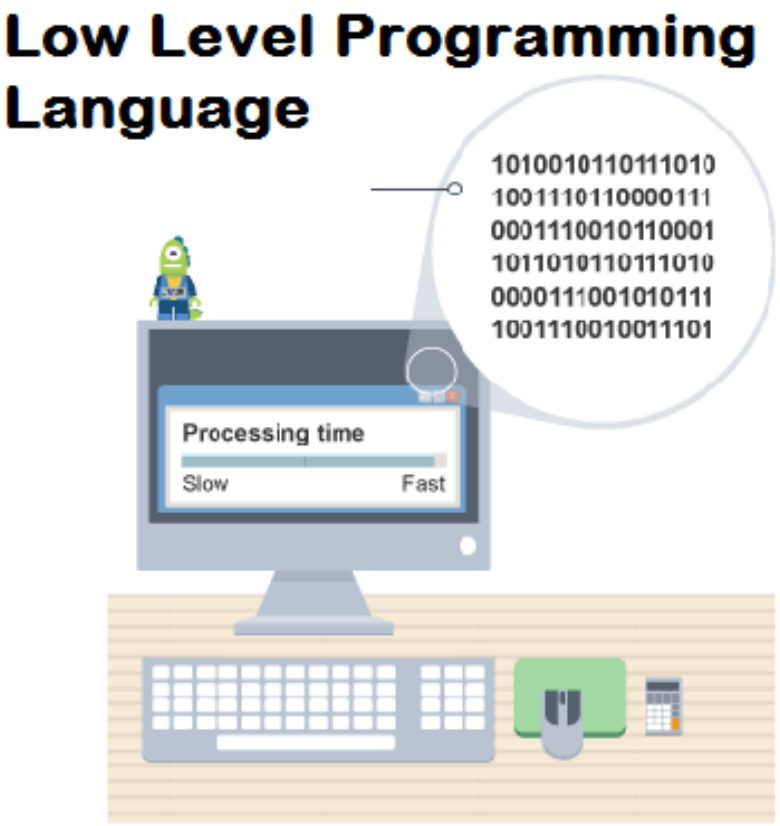
Kaj dela računalnik? (Kako deluje?)

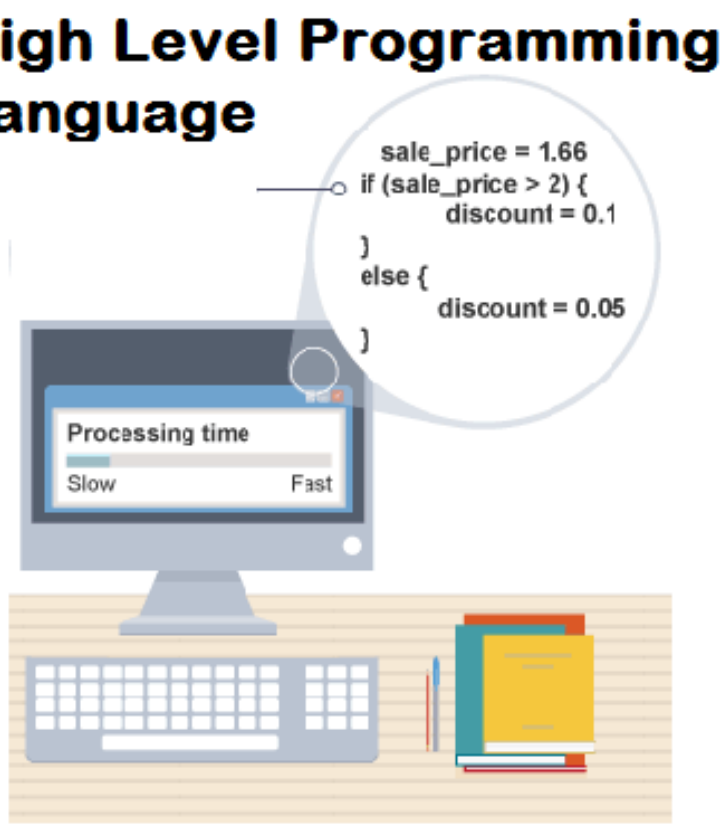
Izvaja ukaze !

- 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.

strojni jezik <-> višjenivojski programski jeziki?

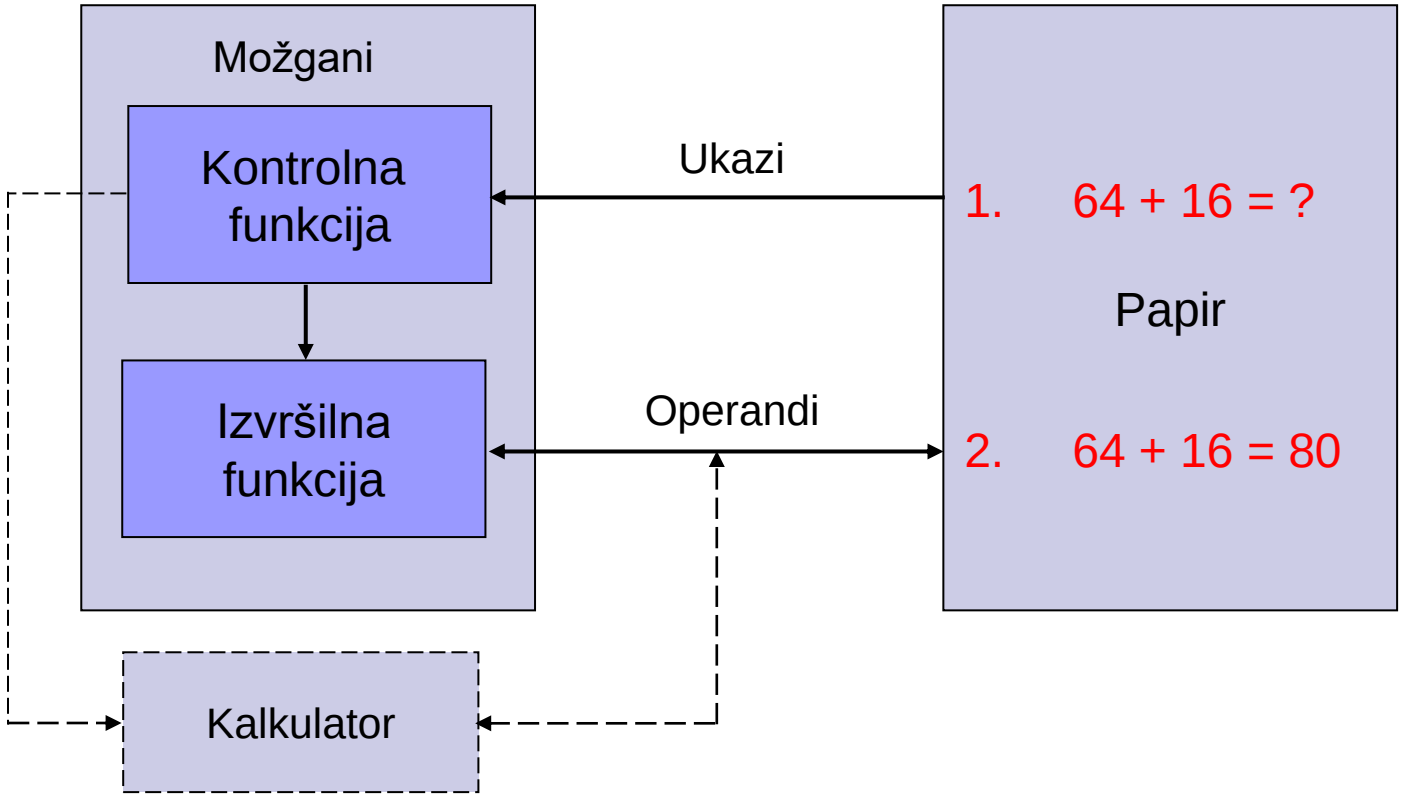




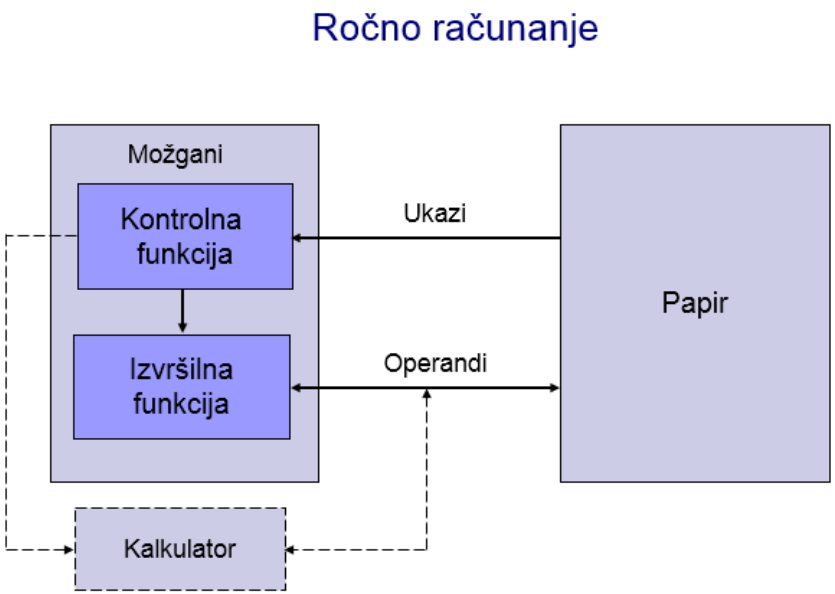
Prenosljivost vs hitrost ?

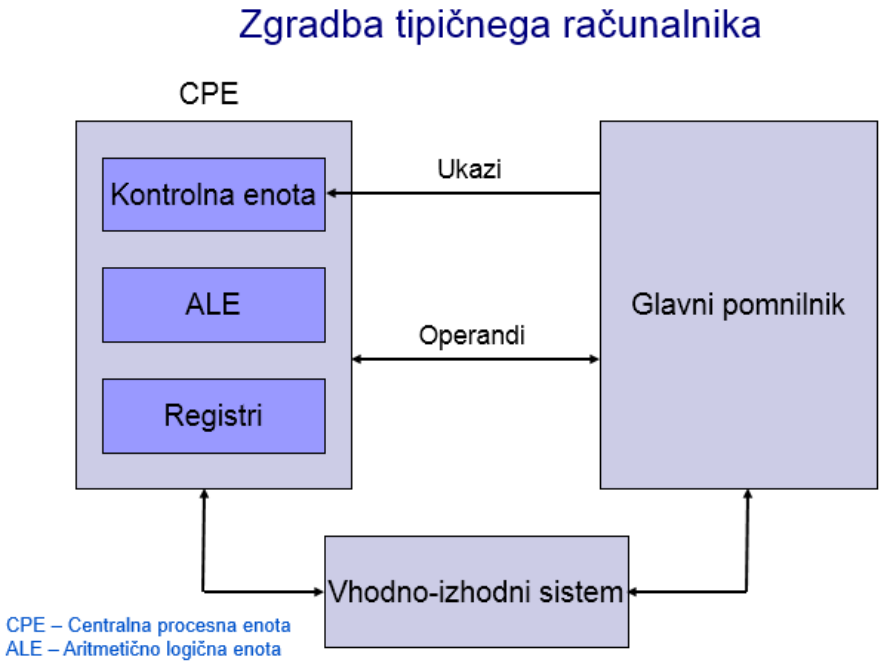
Povezava med ročnim in strojnim računanjem

Ročno računanje

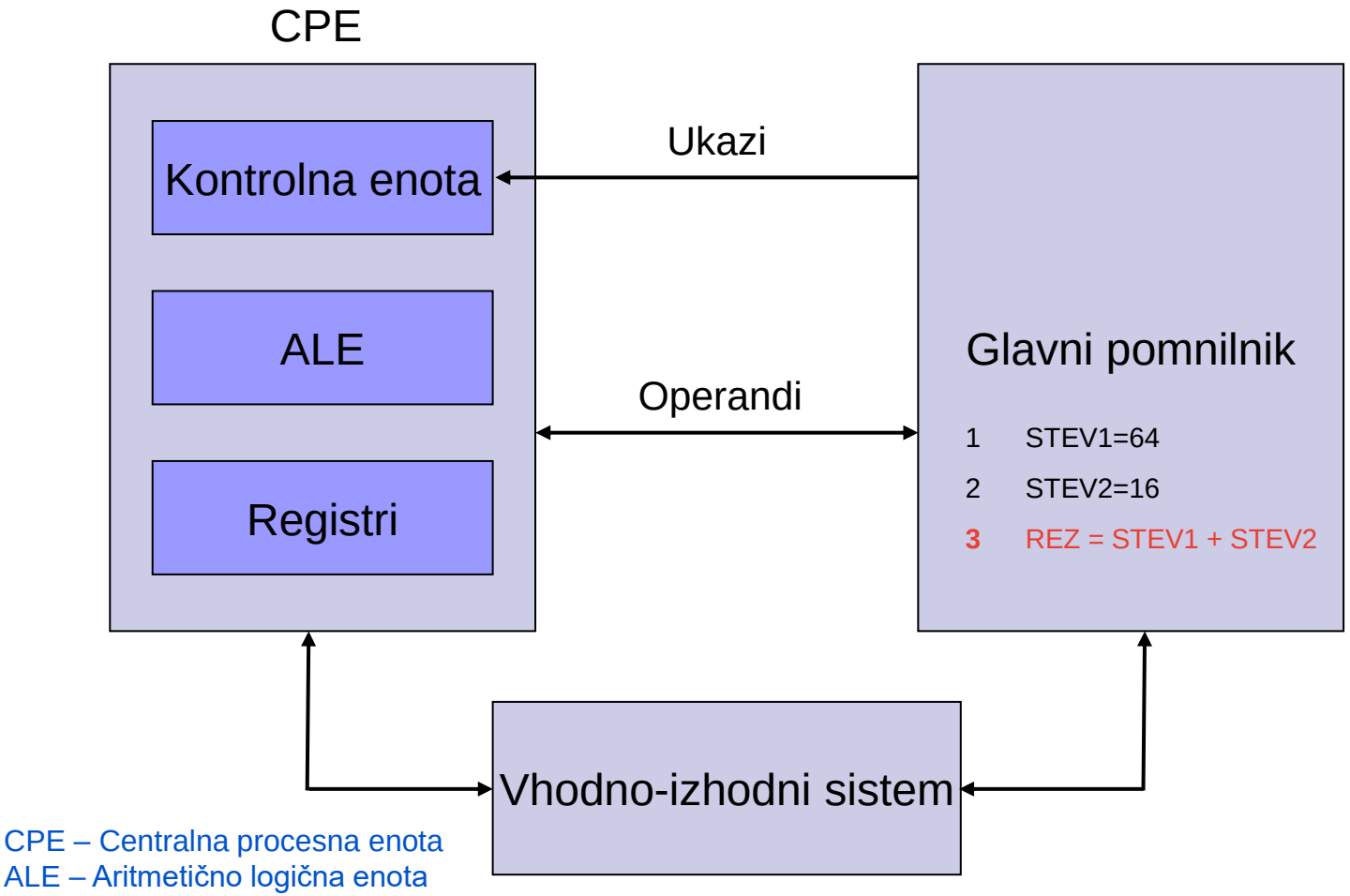


Primerjava modelov računanja

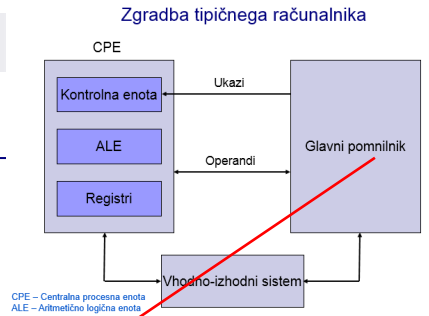




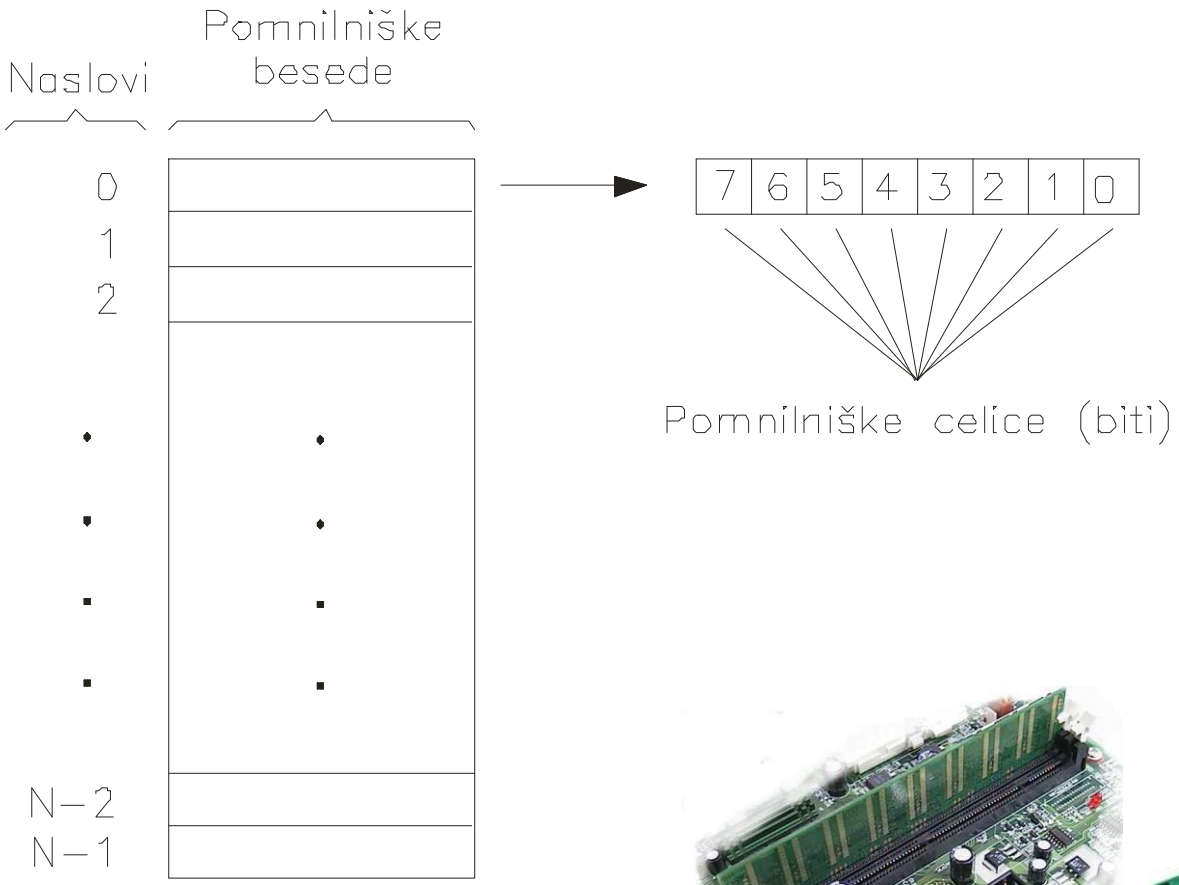
Zgradba tipičnega računalnika

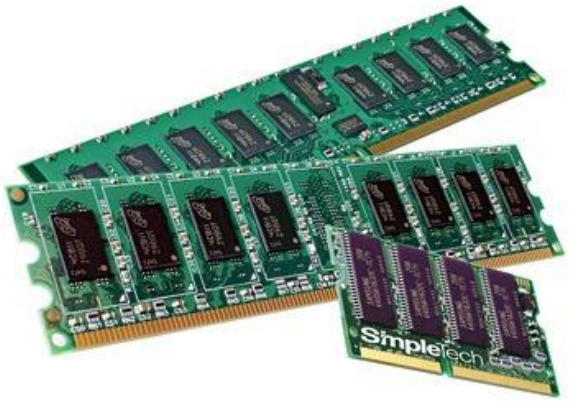


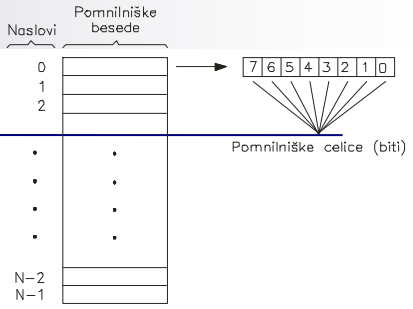
CPE – Centralna procesna enota
ALE – Aritmetično logična enota


Zgradba tipičnega računalnika
CPE
Kontrolna enota
ALE
Registri
Ukazi
Operandi
Glavni pomnilnik
Vhodno-izhodni sistem
CPE – Centralna procesna enota
ALE – Aritmetično logična enota

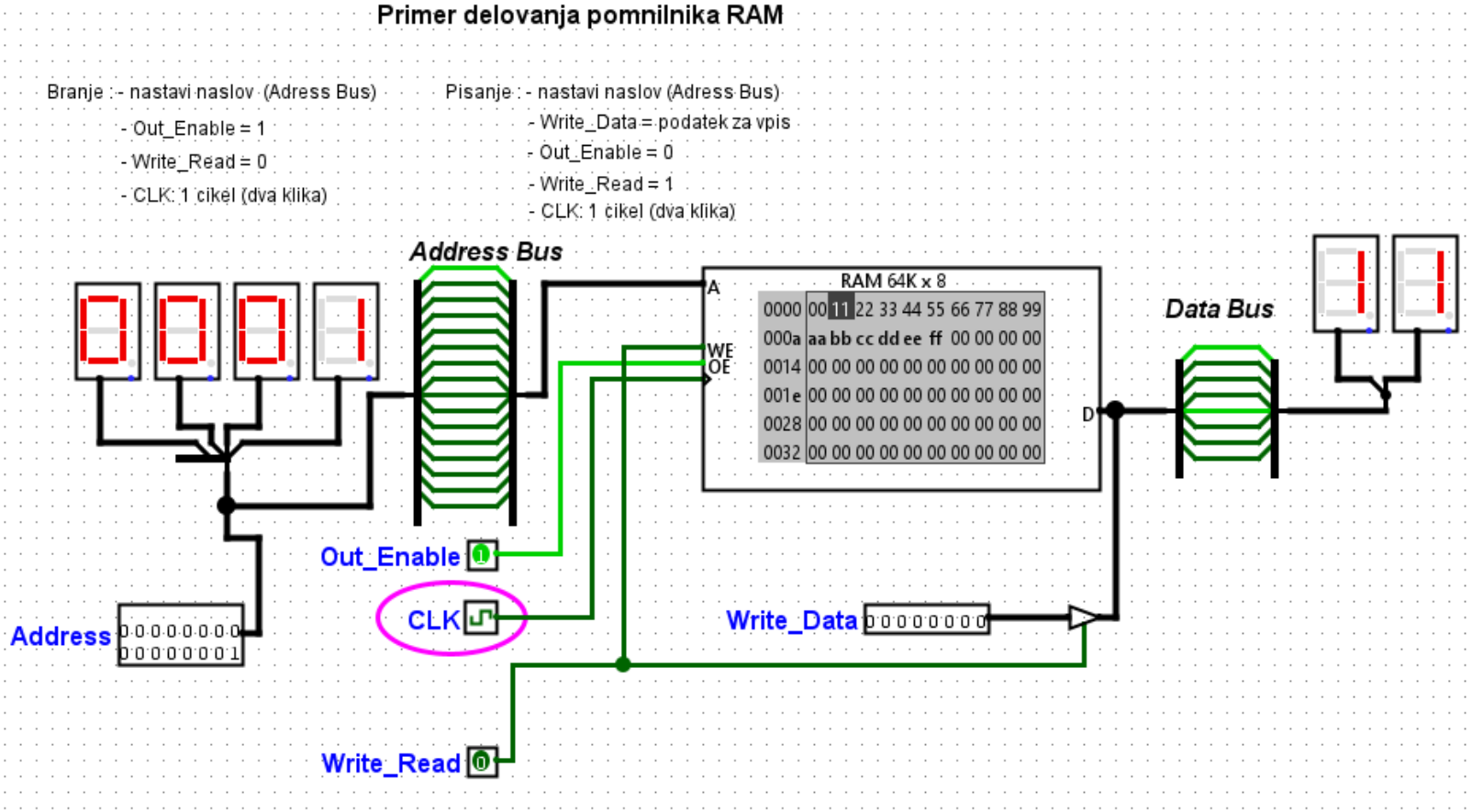
Pomnilnik


Naslovi
Pomnilniške besede
0
1
2
N–2
N–1
7 6 5 4 3 2 1 0
Pomnilniške celice (biti)

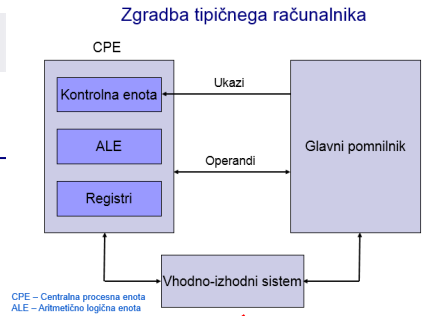

Simpletech



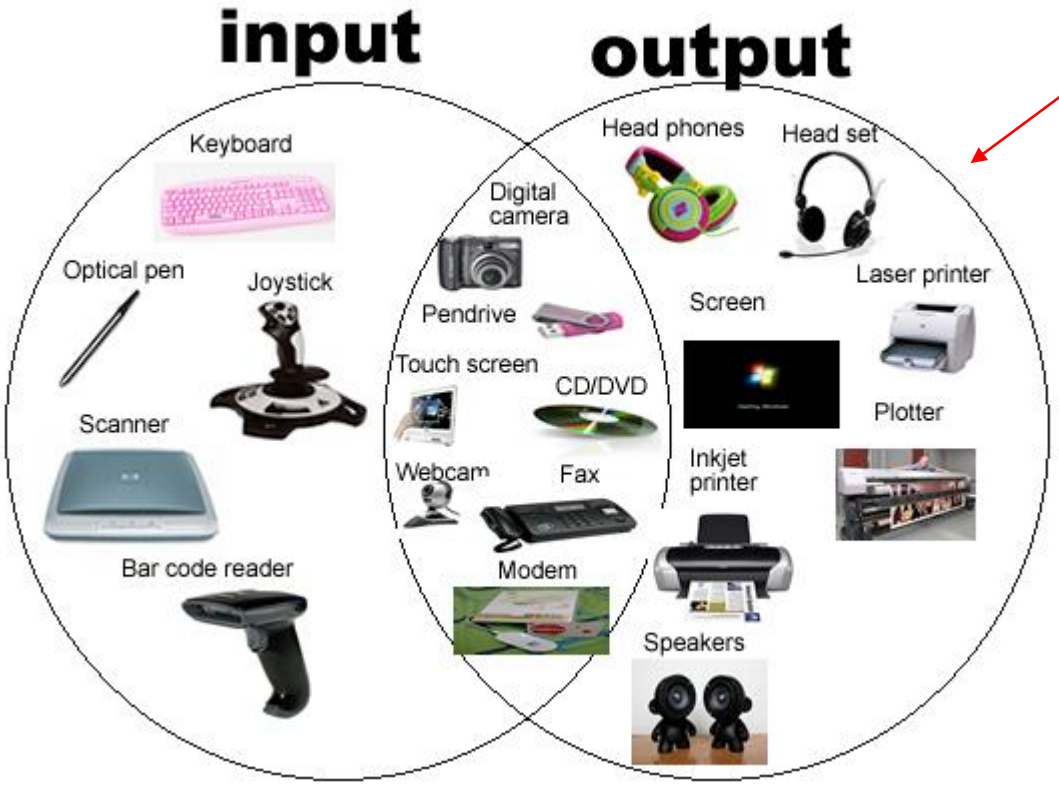
Pomnilnik
Demonstracija – Logisim EVO



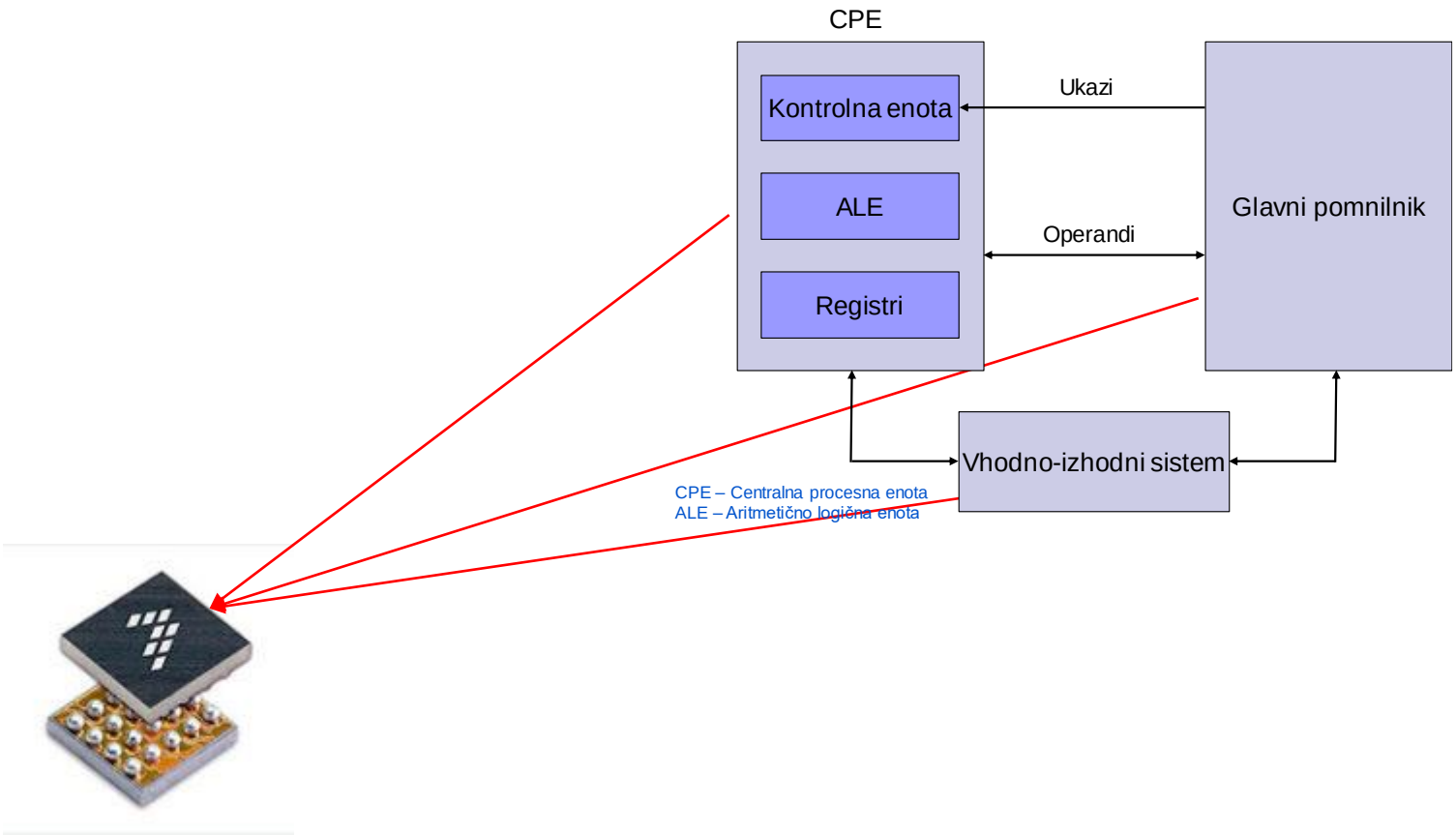
RAM_pomnilnik_demo_EVO.circ



Vhodno-izhodni sistem (naprave)

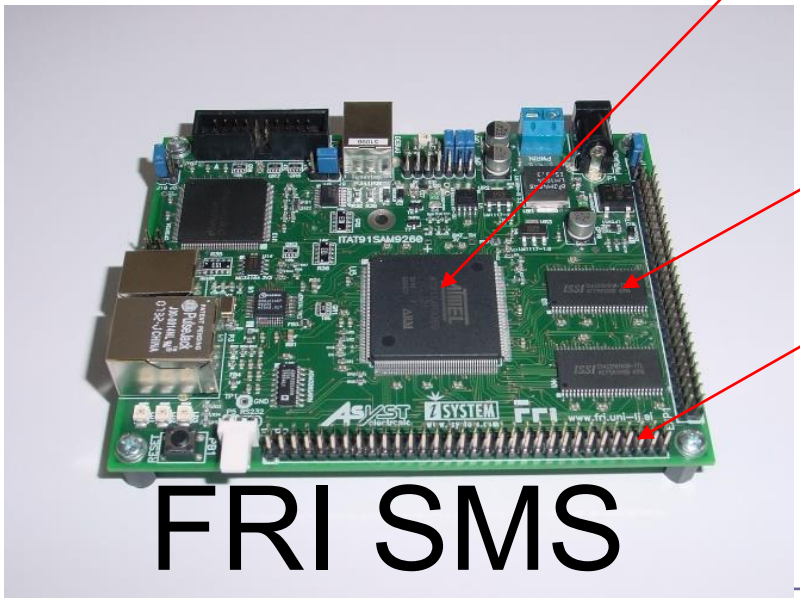


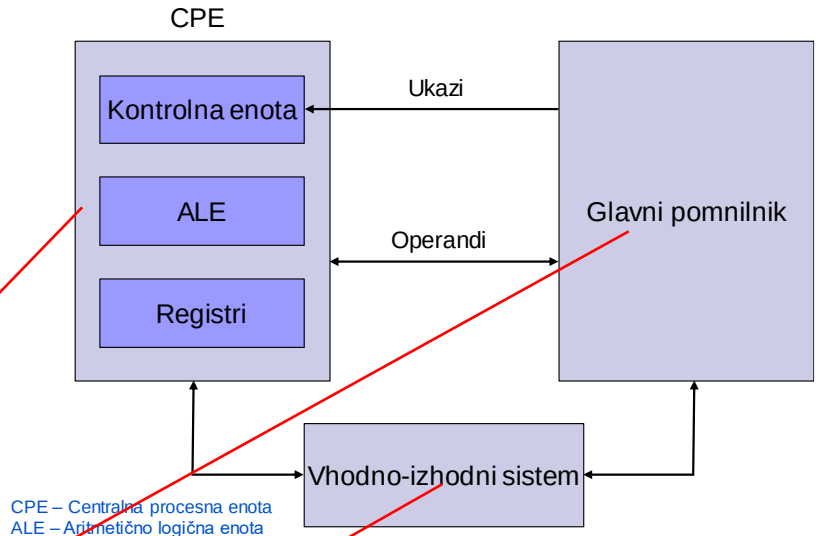
Zgradba tipičnega vgrajenega računalnika



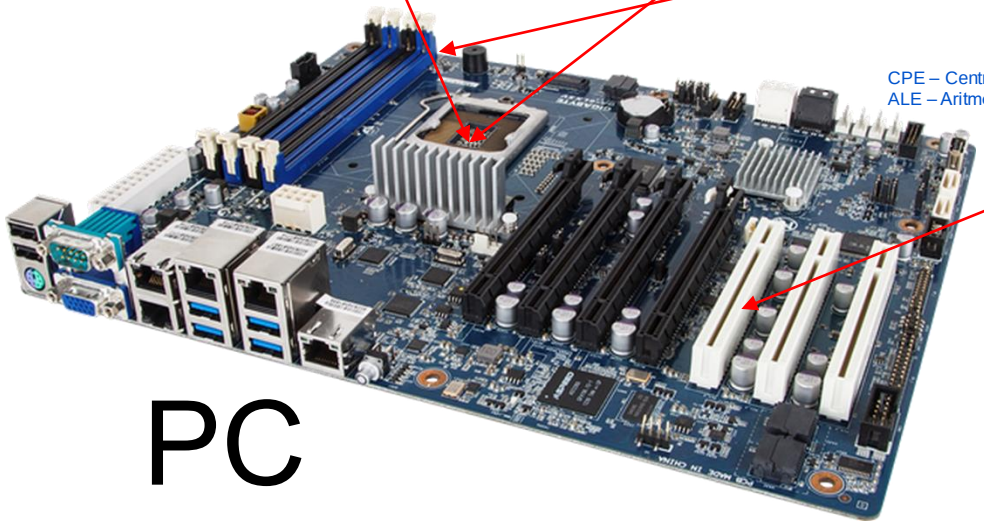
Mikrokrmilnik

Zgradba tipičnega manjšega (mikro) računalnika

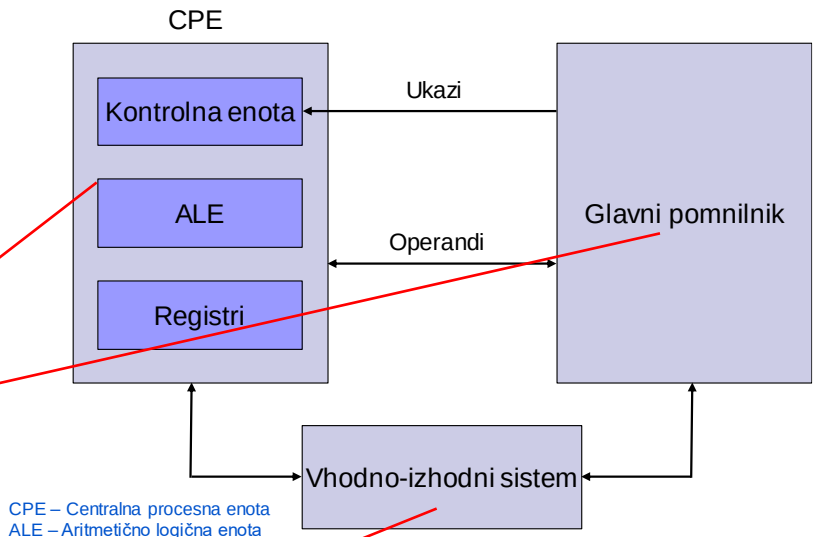




Zgradba tipičnega namiznega računalnika



PC



CPE – Centralna procesna enota
ALE – Aritmetično logična enota

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

Python

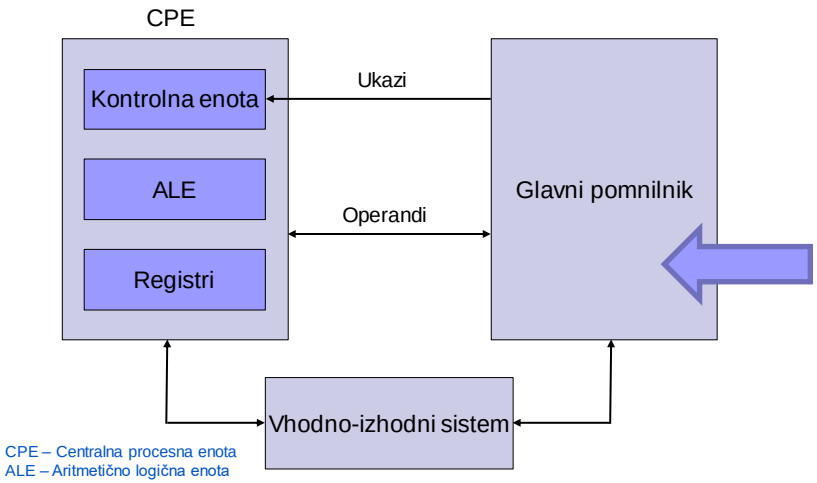
1 STEV1=64

2 STEV2=16

3 REZ = STEV1 + STEV2

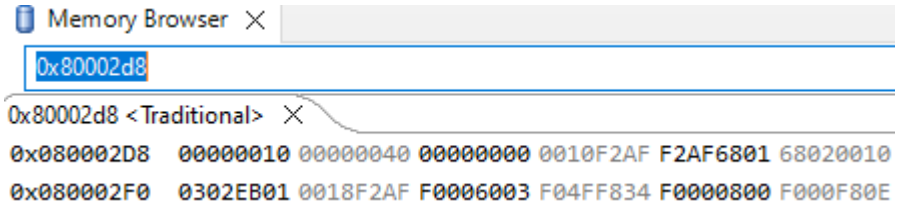
Zbirnik

```
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
```



CPE – Centralna procesna enota
ALE – Aritmetično logična enota

Strojni jezik



Program vsote dveh števil (1. LAB vaja)

Python

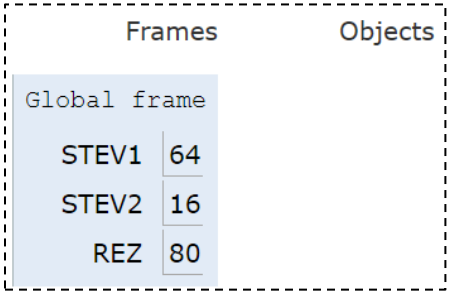
http://goo.gl/YXQ5qN

1 STEV1=64

2 STEV2=16

3 REZ = STEV1 + STEV2

Spremenljivke v pomnilniku



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
```

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

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

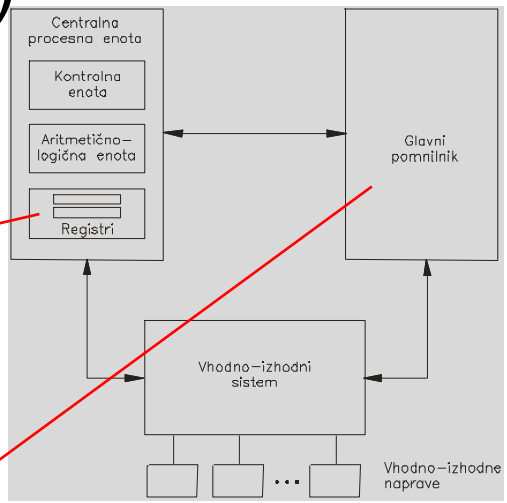
Global frame	
STEV1	64
STEV2	16
REZ	80

Print output (drag lower right corner to resize)

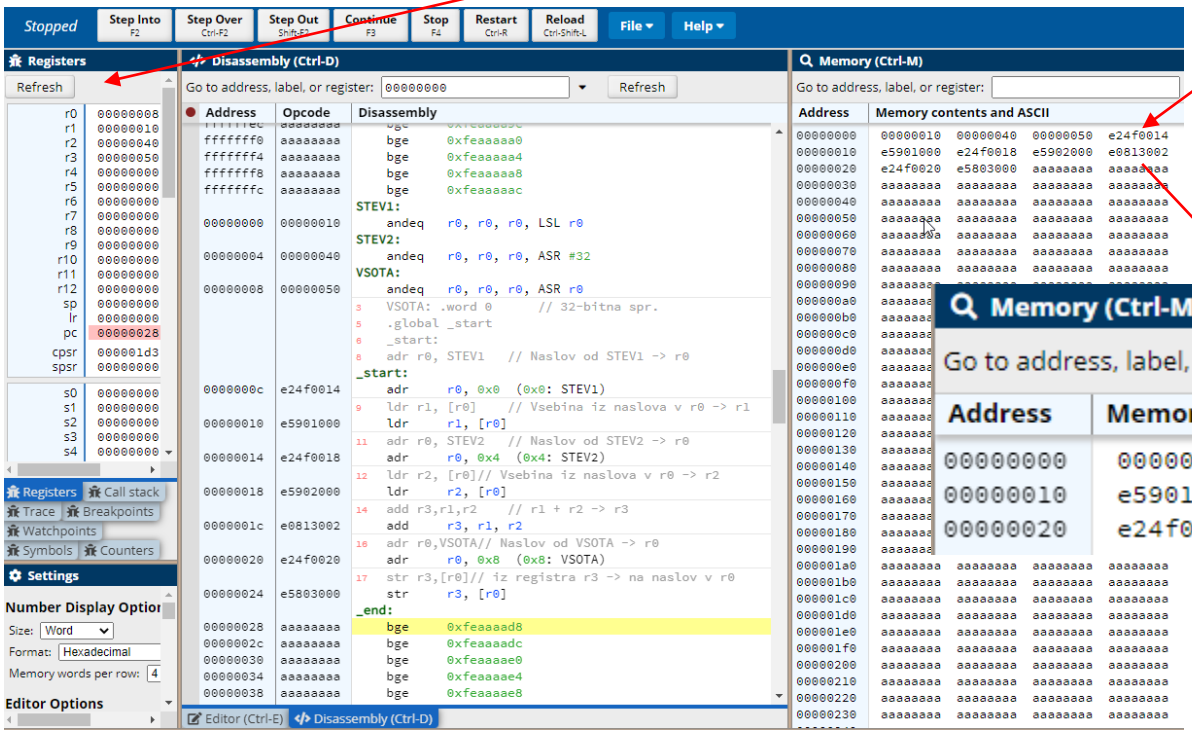
```
 STEV1 = 0x40
+STEV2 = 0x10
-------------
   REZ = 0x50
```

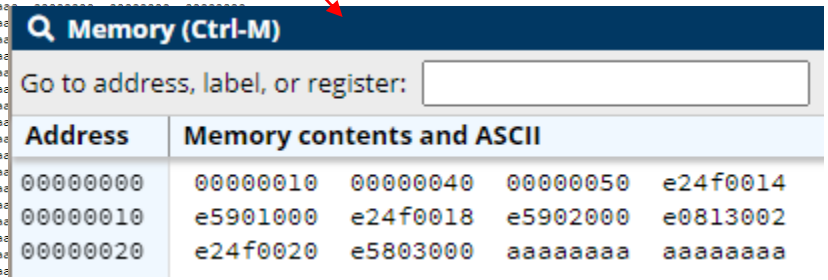
Zbirnik (zgled: REZ = STEV1 + STEV2)

CPULator – spletni simulator



CPULator





1. Uvod :

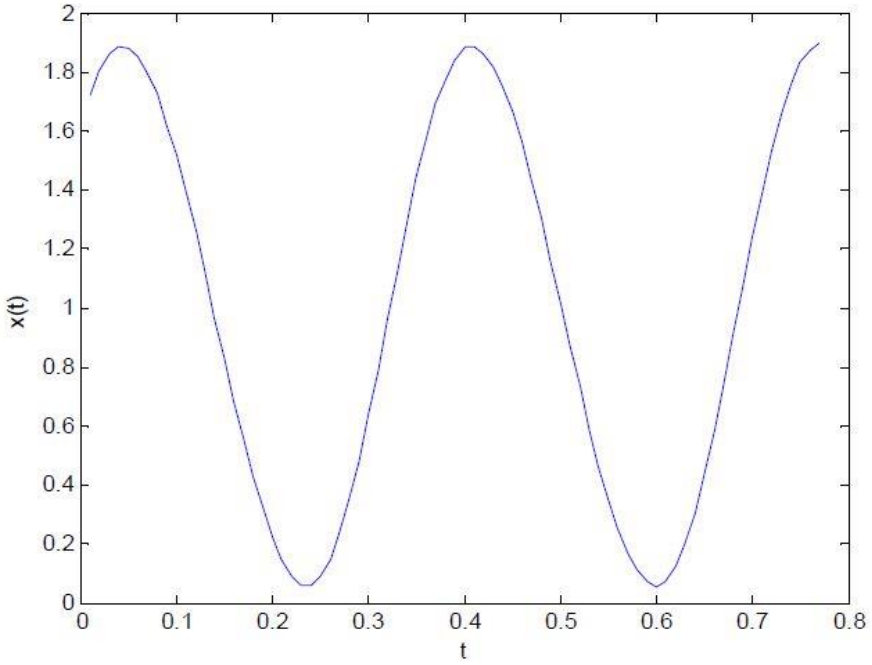
- 1.1 Predmet RA
- 1.2 Računalniki včeraj in danes
- 1.3 Osnove zgradbe in delovanja računalnikov
- 1.4 Analogno – digitalno, zvezno - diskretno
- 1.5 8 pomembnih idej v računalniški arhitekturi (in širše)
- 1.6 Praktična realizacija računalnikov

Primer 1: Ročna ura

Analogno –
- zvezna predstavitev

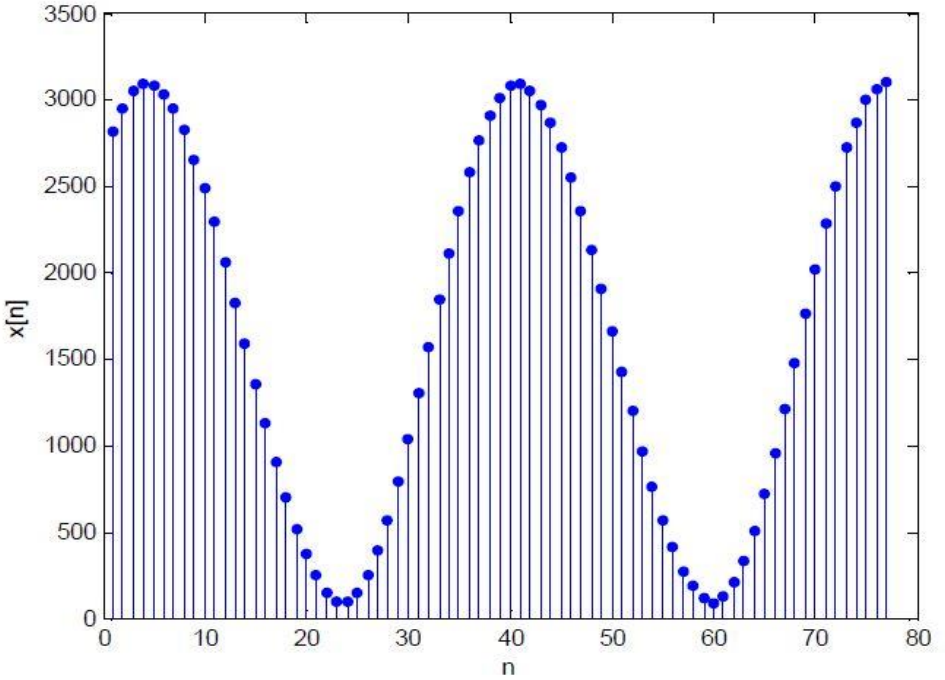
Digitalno –
- diskretna predstavitev

Analogno –
- zvezna predstavitev



Čas v sekundah

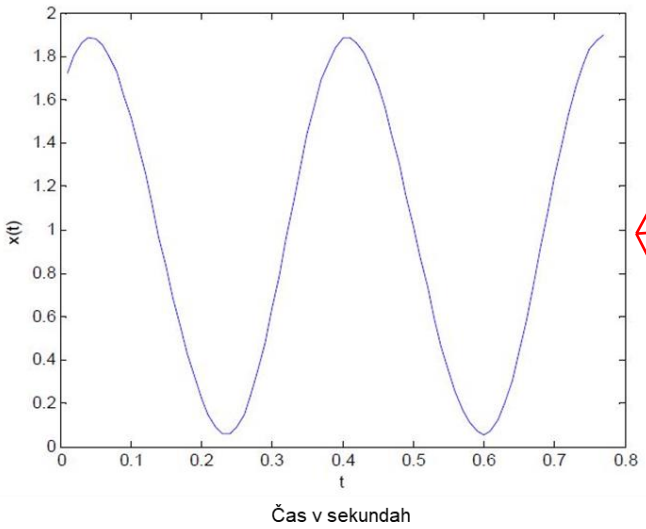
Digitalno –
- diskretna predstavitev

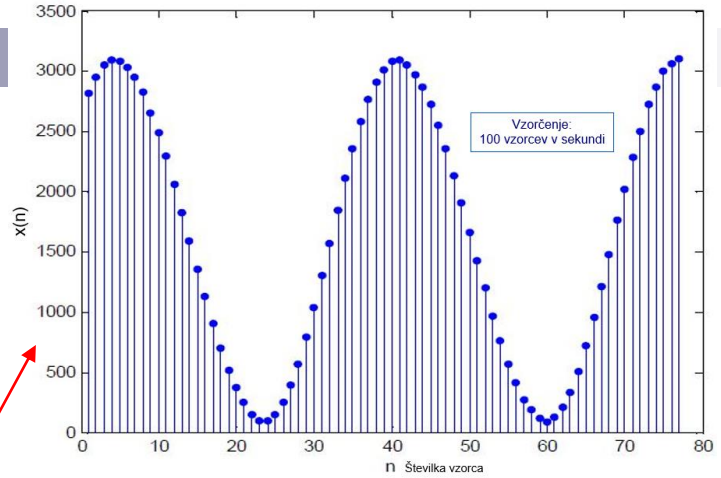


Številka vzorca

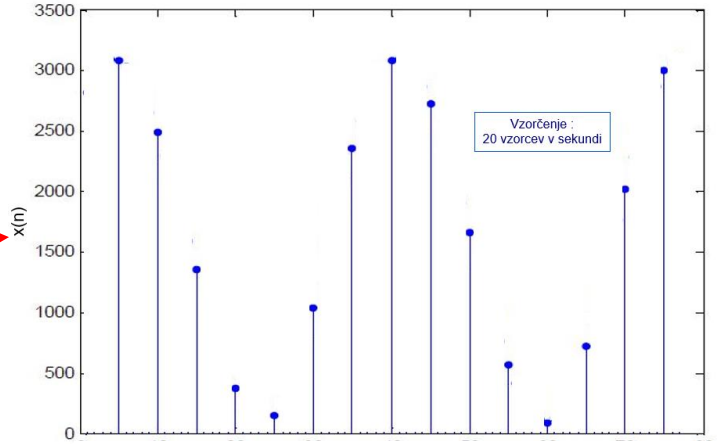
Vzorčenje: 100 vzorcev v sekundi

Primer 2: Vzorčenje signala

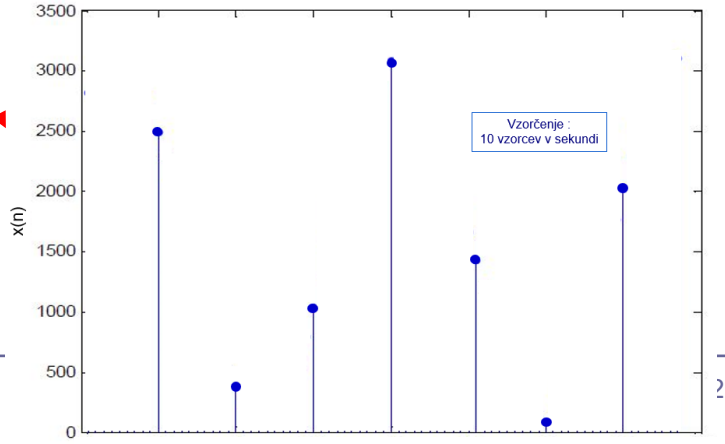




100 vzorcev/sek



20 vzorcev/sek

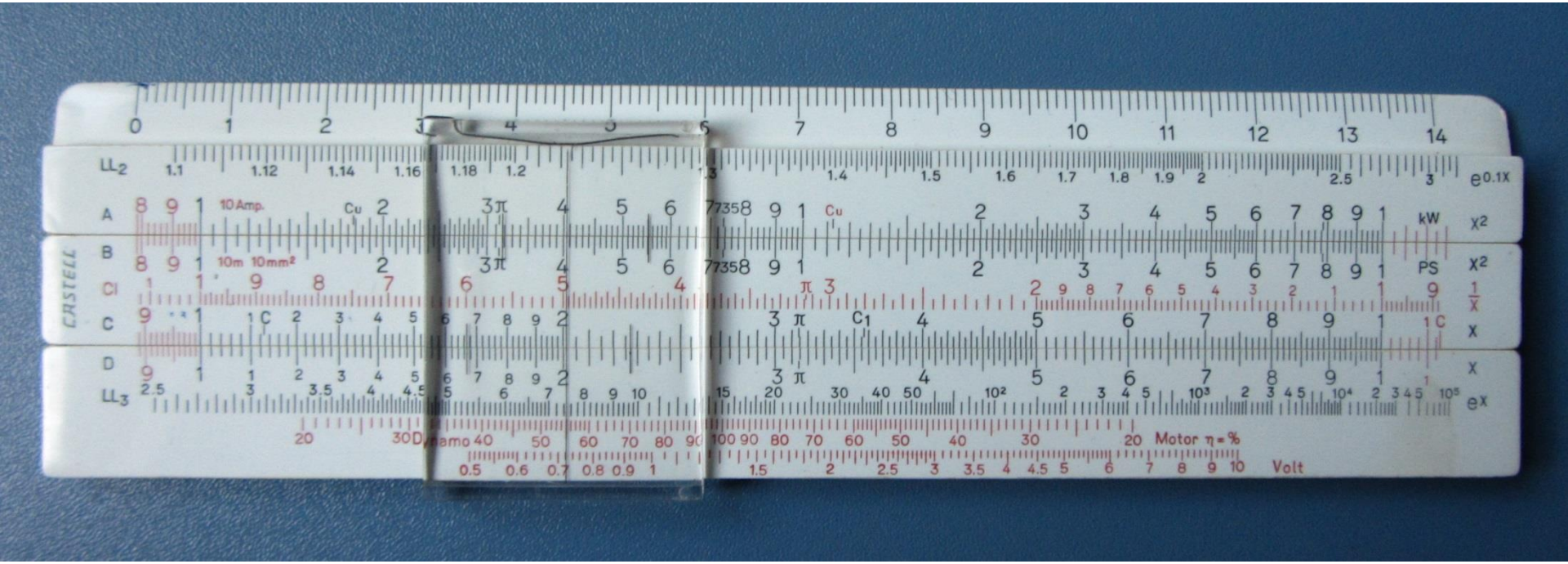


10 vzorcev/sek

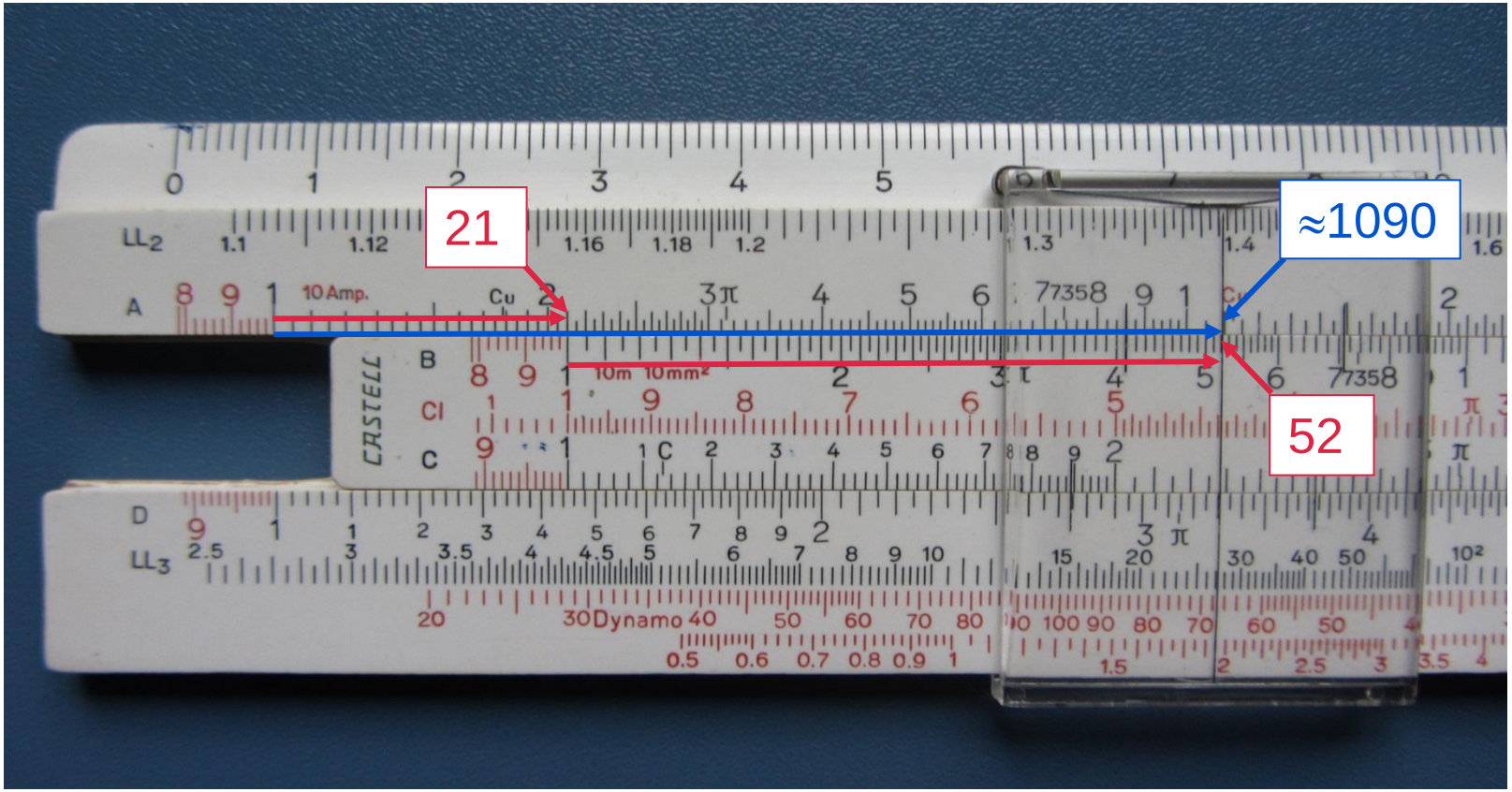
Analogno računanje – zvezna predstavitev števil

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 računalo
  - Ideja: $\log_{10}(a \cdot b) = \log_{10} a + \log_{10} b$



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



21 x 52 ≈ 1090

Odčitan rezultat

21 x 52 = 1092

Točen rezultat

- Z napetostjo ⇒ Analogni ojačevalec

- Z napetostjo ⇒ Analogni računalnik

Diskretno računanje

- s kroglicami

- s števkami od 0 do 9

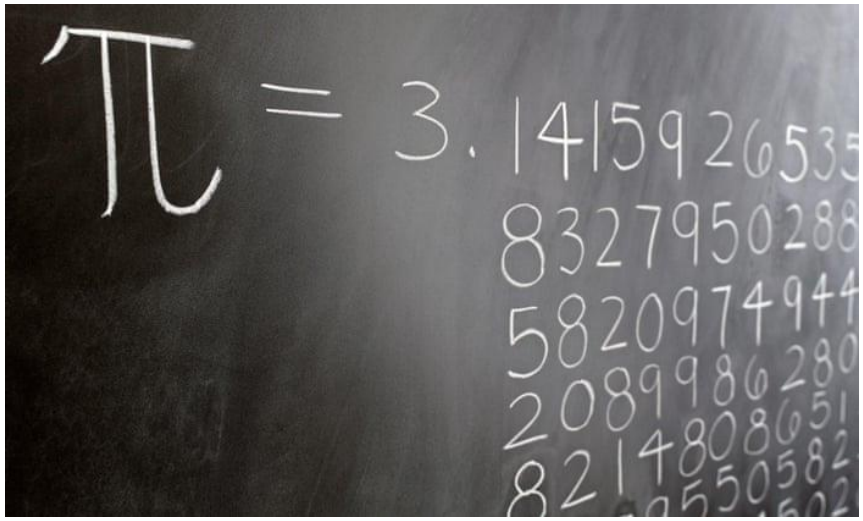
Digitalno računanje

- s števkama 0 in 1

- dvojiški številski sistem:
  - osnova številskega sistema je 2
  - števki 0 in 1
- dvojiška števka angl. binary digit = bit
- bit = ena od dveh števk (0 ali 1) dvojiškega številskega sistema
- digitalni računalnik temelji na dvojiškem sistemu

- števila predstavimo s končnim številom bitov

- Kako lahko predstavite/računate s številom π v računalniku ?



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 aparaturne 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

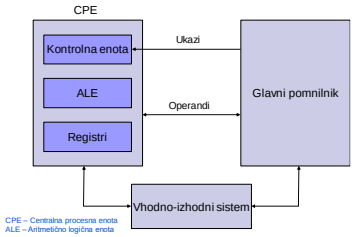
8 pomembnih idej v računalniški arhitekturi (in širše) [Patt]

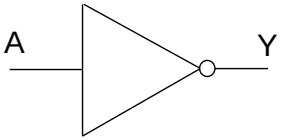
5. Zmogljivost s cevovodnim procesiranjem
   - Učinkovit, transparenten način pohitritve delovanja CPE
6. Zmogljivost s predvidevanjem
   - „Raje delaj ob neki predpostavki, kot samo čakaj“
7. Hierarhični model pomnilnika
   - Učinkovit kompromis med hitrostjo in ceno pomnilnikov v računalnikih
8. Zanesljivost z redudanco
   - 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





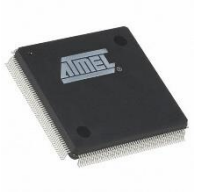
Simbol

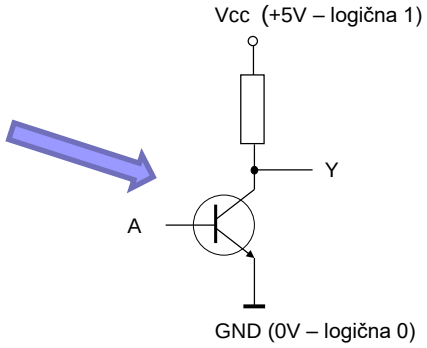
Matematični ideal

Teoretični model - pogled: logična vrata — logični nivoji 0,1

Elektronska realizacija : elektronsko vezje — napetostni nivoji ≈0V, ≈3.3 (5) V

Elektronska realizacija





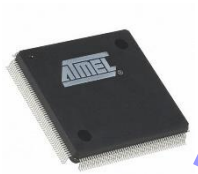
Slabosti :

- *zvezne napetosti*
- *časovne zakasnitve*
- *motnje*

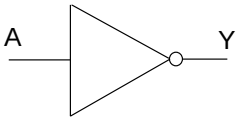
Fizična zgradba računalnikov

Računalnik

Vezje (angl. Chip)



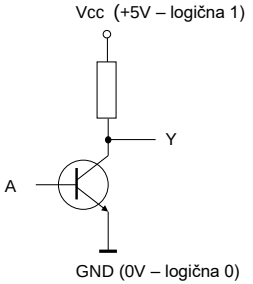
Logična vrata (funkcija)



Simbol

Primer: negacija

Elektronska realizacija



Primer: negacija

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 (≈ 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.

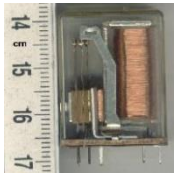
Realizacije stikala v razvoju digitalnih računalnikov – razvoj tehnologije

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 :

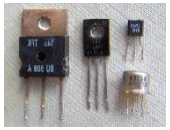
- Elektromehansko stikalo



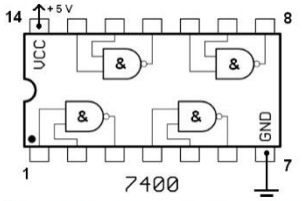
  - 1939: Rele, čas preklopa 1-10ms

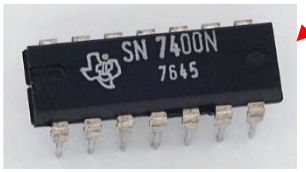
- Elektronsko stikalo

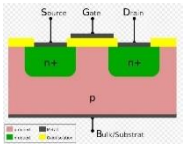
  - 1945-1955: Elektronka, čas preklopa ~ 5μs

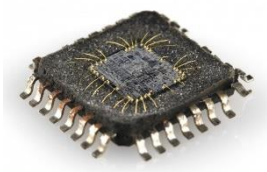


  - 1955: Tranzistor → , čas preklopa ~10ns
    - 1958: Integrirano vezje - čip, čas preklopa 2-10ns
    - 1980: VLSI integrirana vezja čas preklopa < 0.1ns
      - Very Large Scale Integration

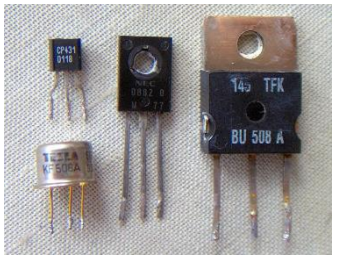




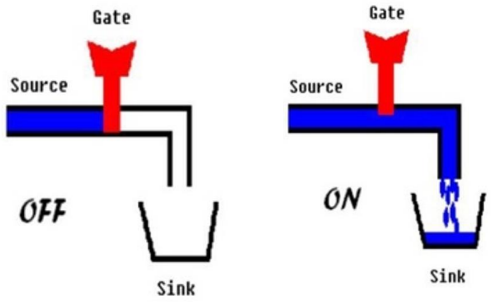


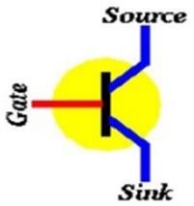


Tranzistor se lahko uporabi kot :

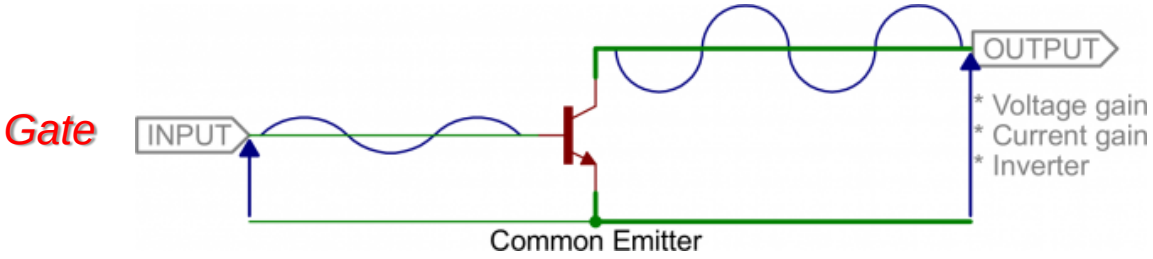


- Stikalo
  - v digitalnih vezjih

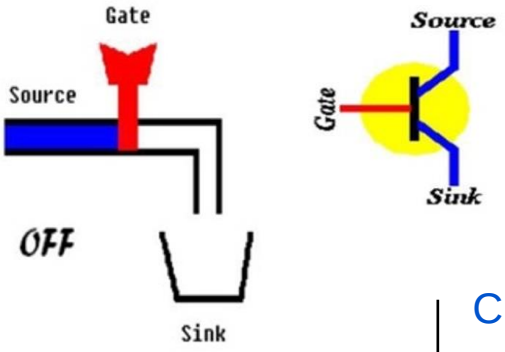


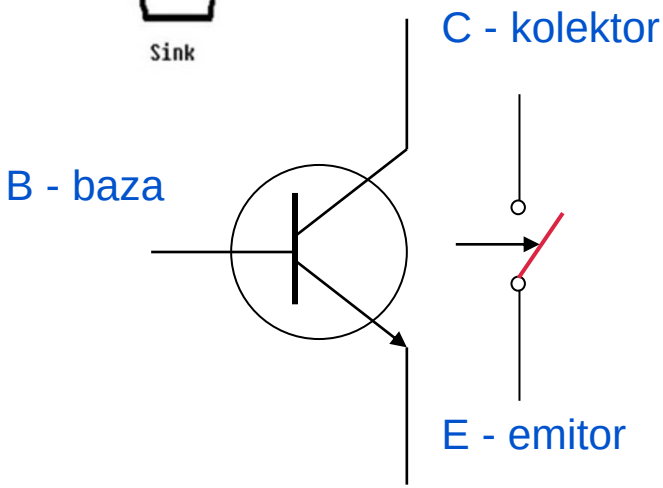


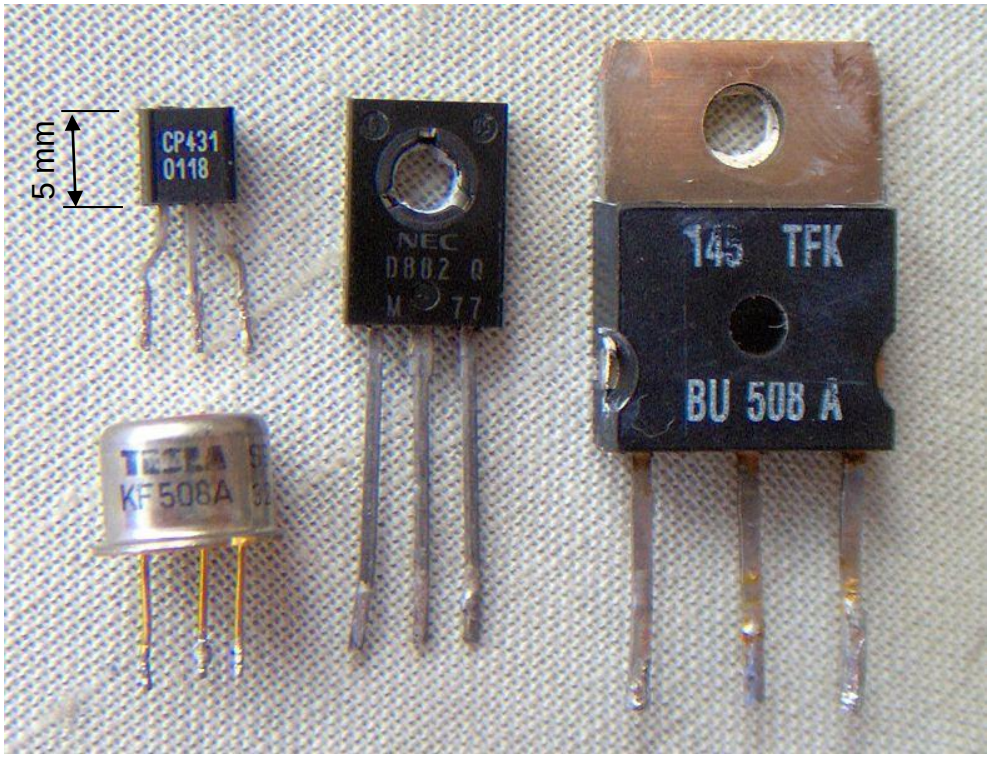
- Ojačevalnik signala
  - V elektronskih vezjih (ojačevalci)



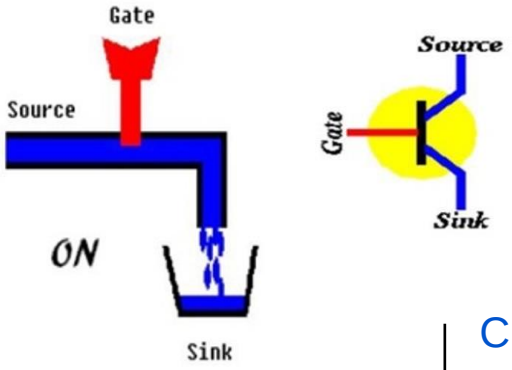
- Delovanje tranzistorja kot stikala – izklop - OFF

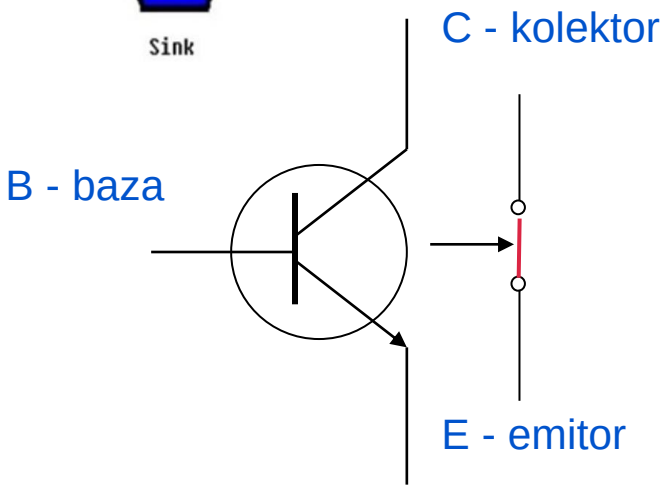


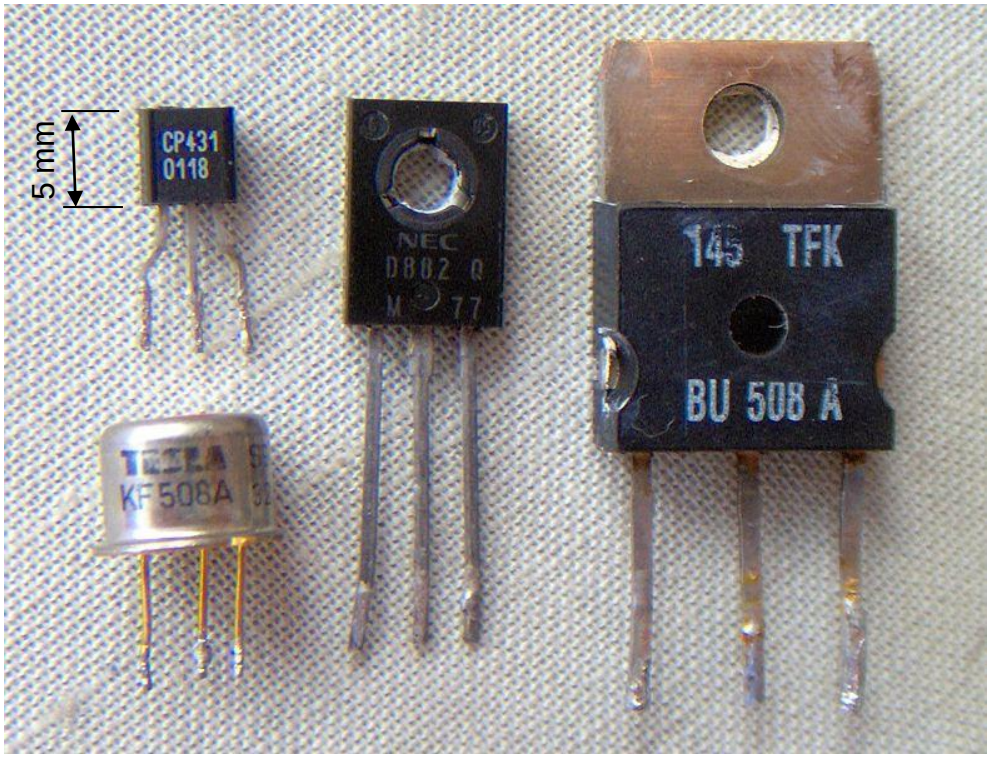




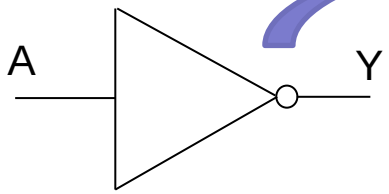
- Delovanje tranzistorja kot stikala – vklop - ON







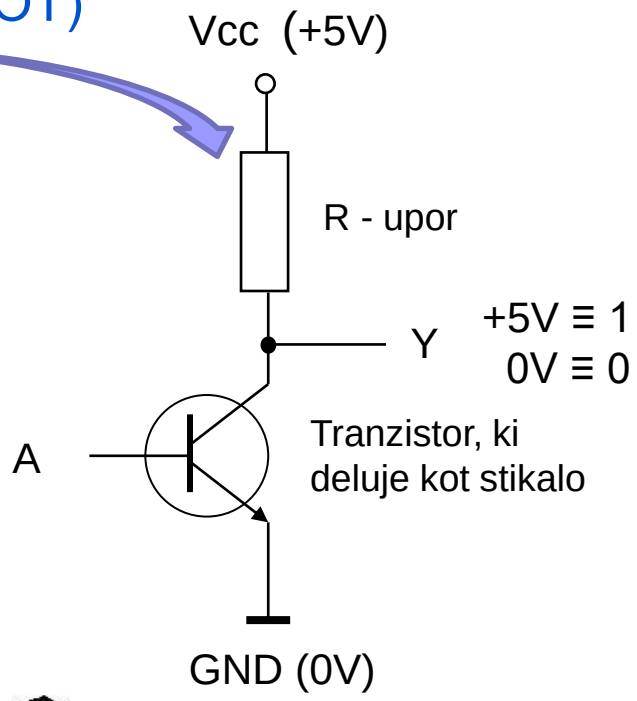
Realizacija logične funkcije NEGACIJA (NOT)



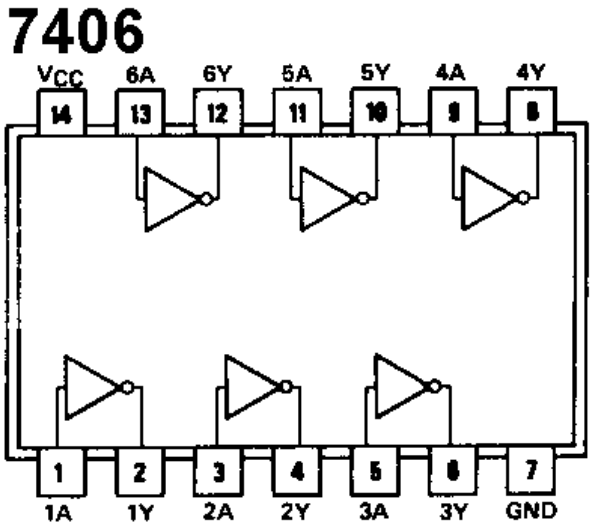
Simbol

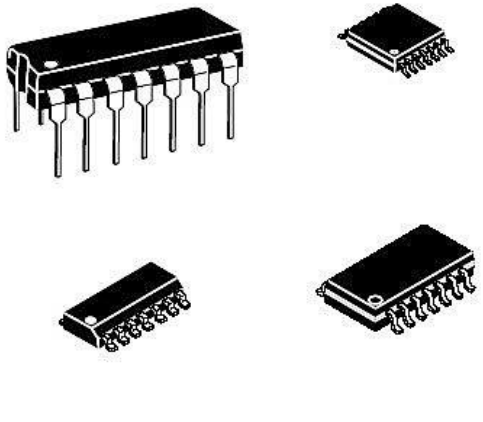
A	Y
0	1
1	0

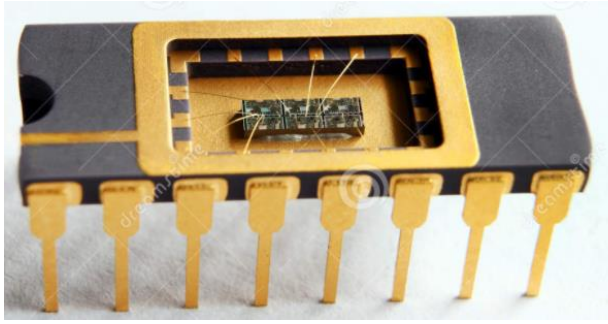
Pravilnostna tabela



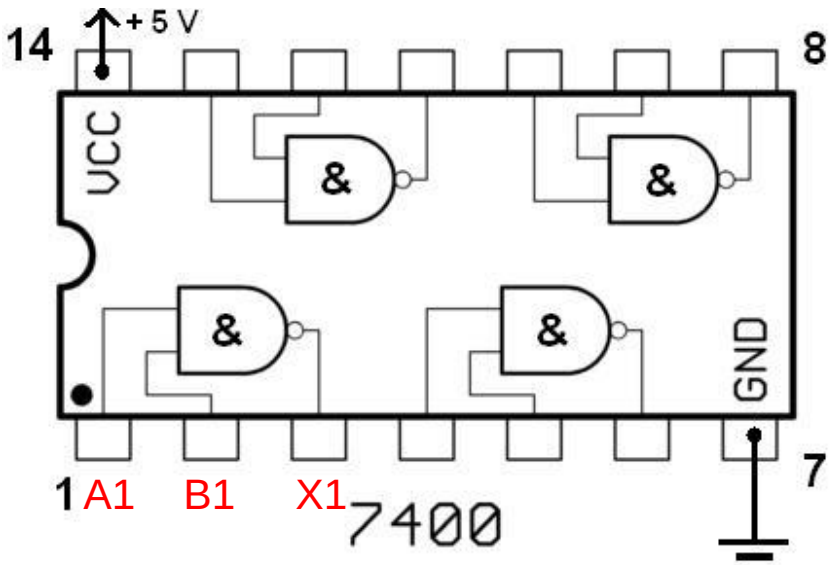
IC (Integrated Circuit) s 6 negatorji

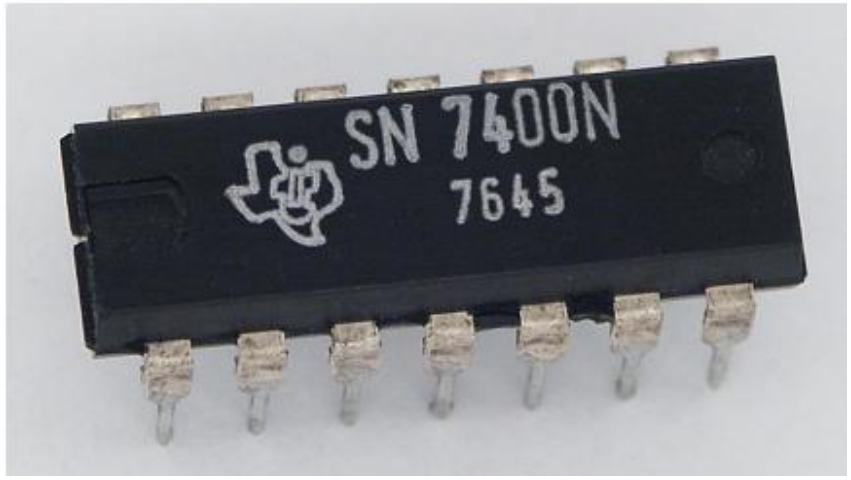


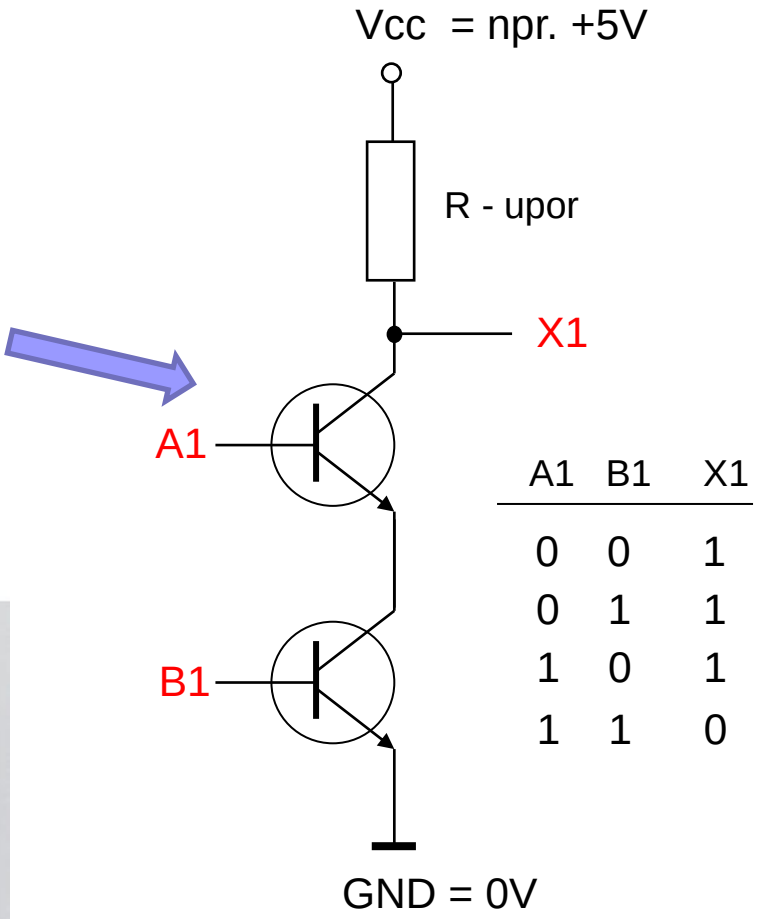




Realizacija logične funkcije NAND (Negirana konjunkcija)

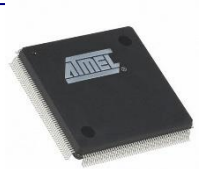


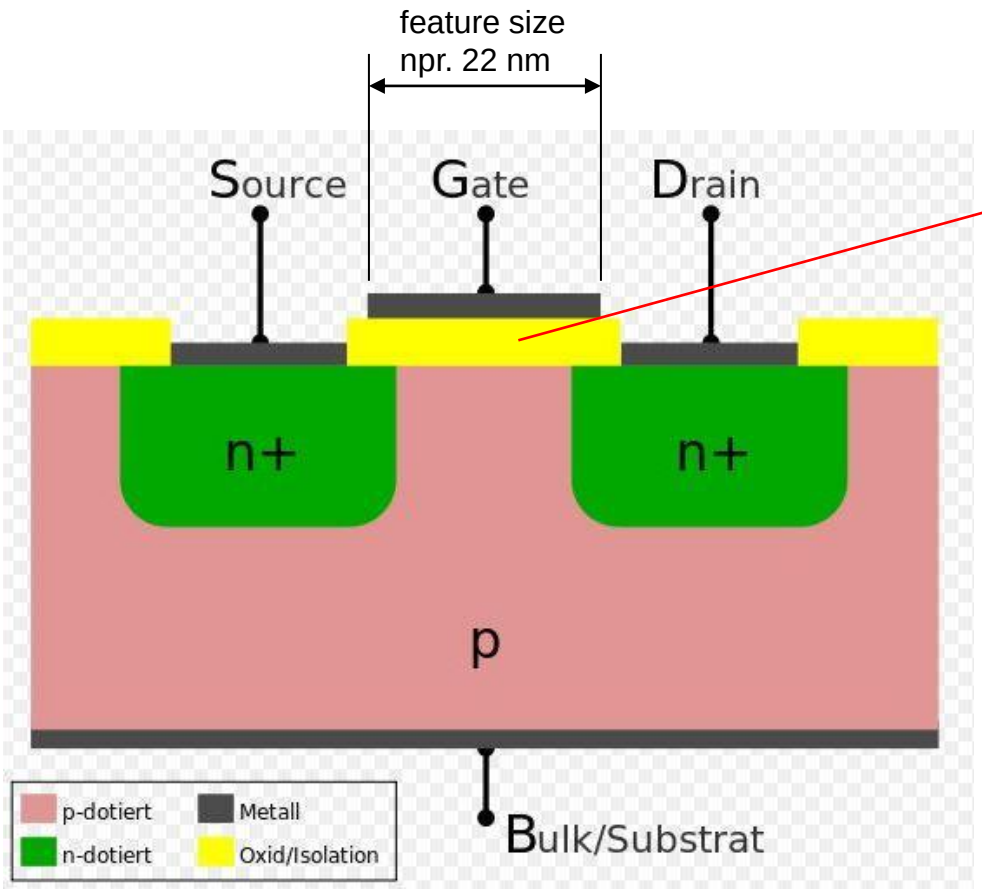


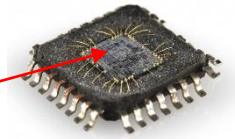


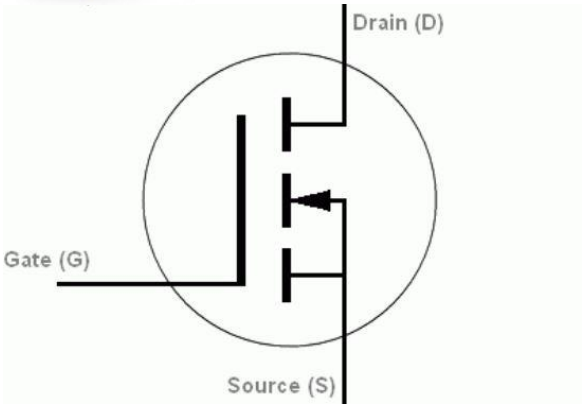
A1	B1	X1
0	0	1
0	1	1
1	0	1
1	1	0

Tranzistor kot del integriranega vezja VLSI



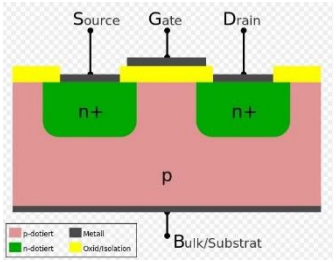


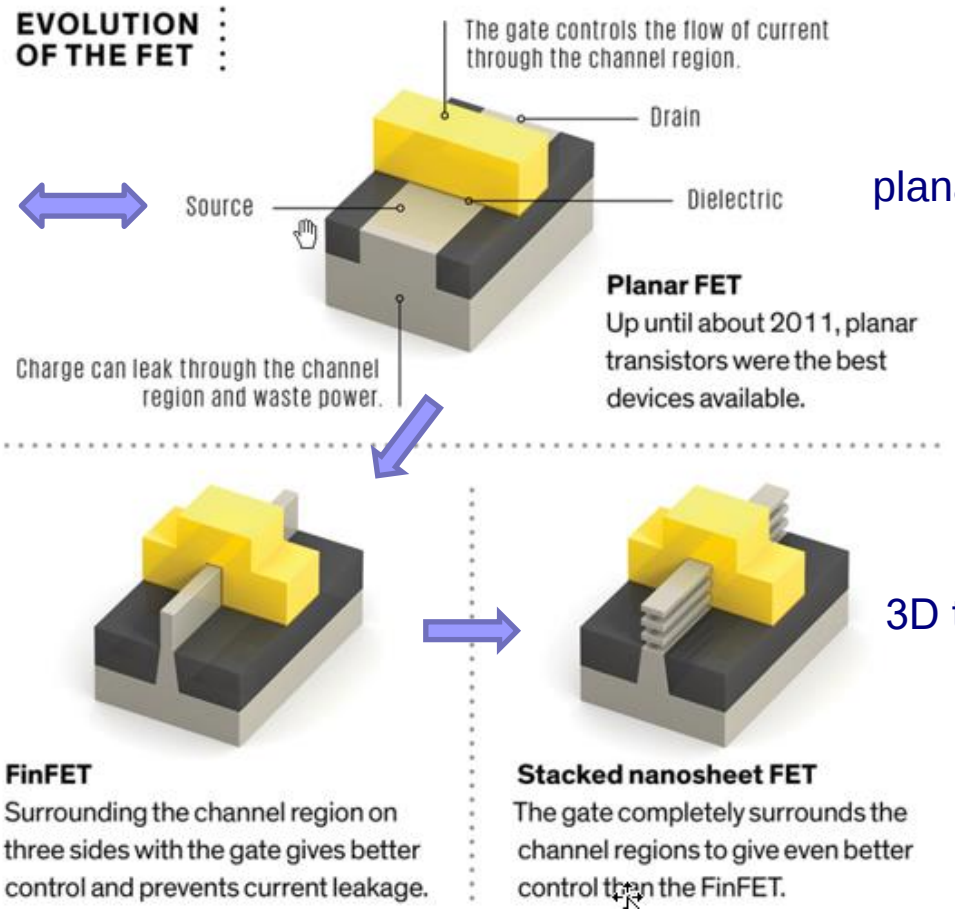




Razvoj tranzistorjev v najsodobnejših vezjih:

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



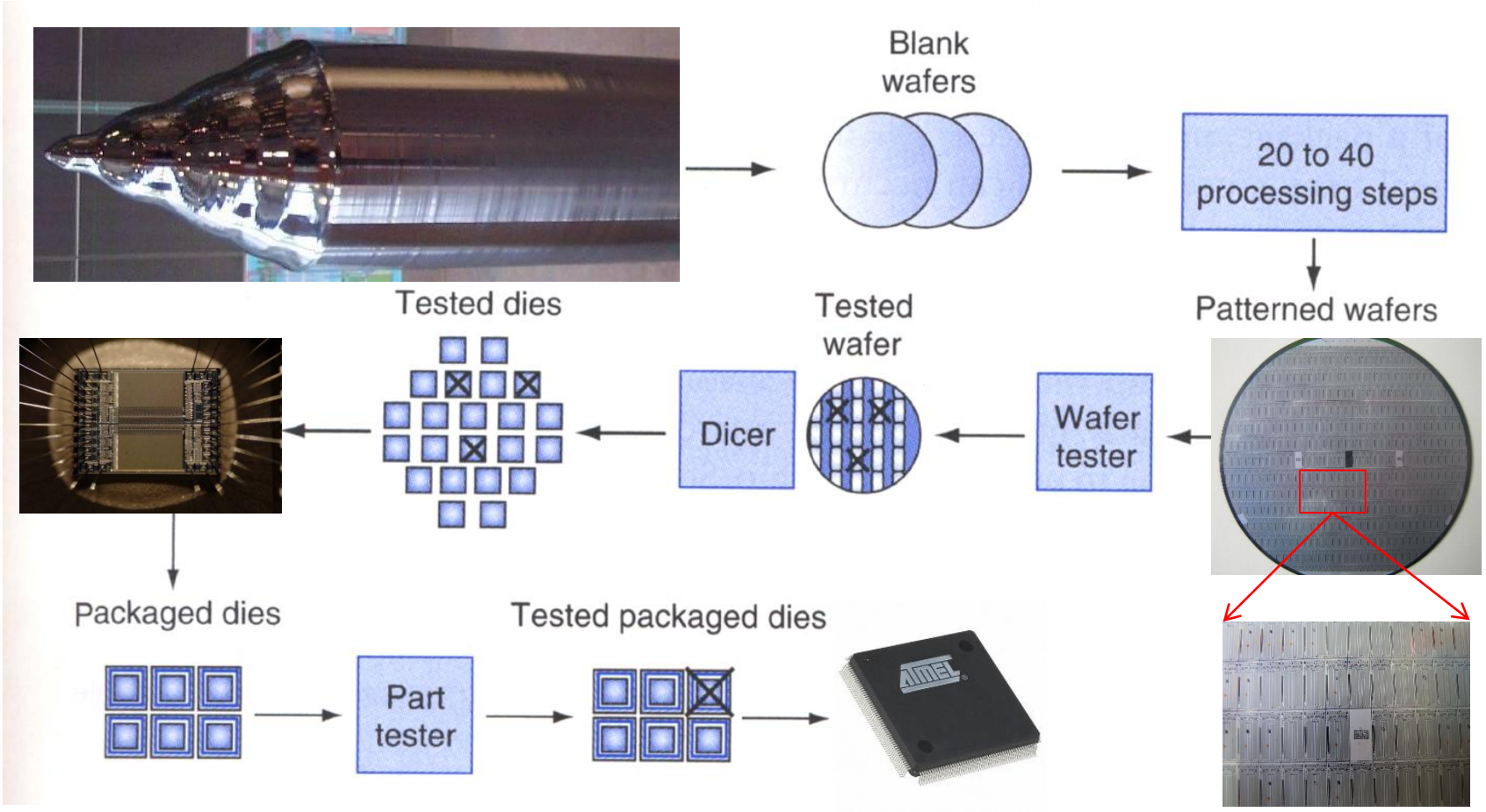


planarni 2D (vodoravna lega)

3D tranzistor (se širi v višino)

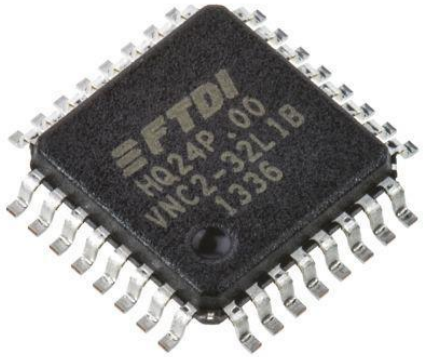
Premer Si atoma je 0.24nm!!!

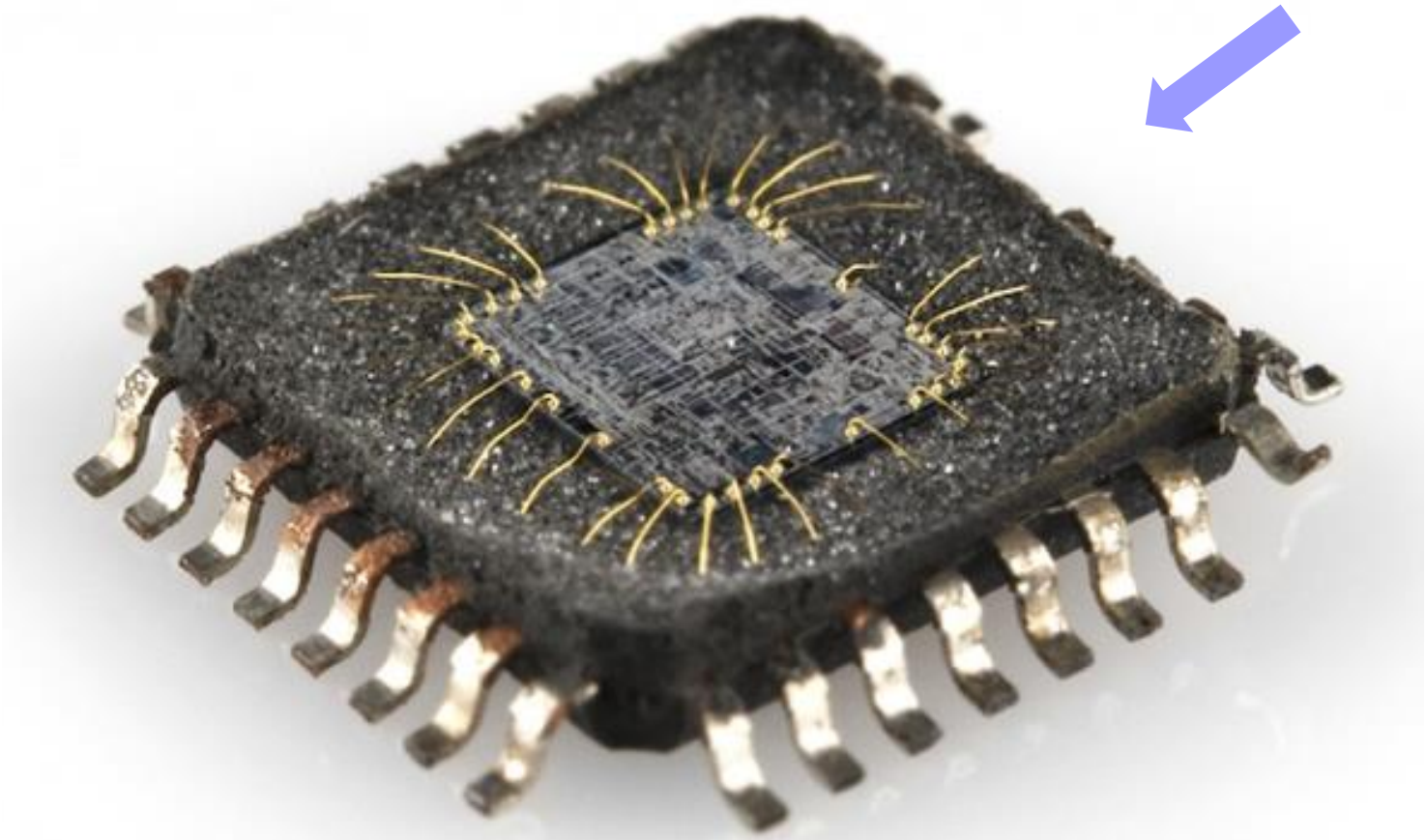
Postopek izdelave VLSI čipa



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

VLSI čip - notranjost



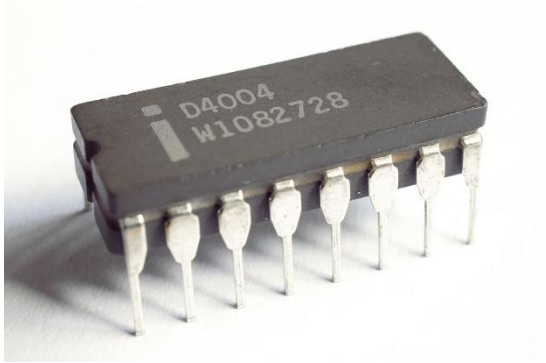


- ? nm proces („feature size ? nm“)
  - Parameter ***feature size*** pri integriranih vezjih v največji meri določa število tranzistorjev na integriranem vezju in tudi njihove lastnosti
  - Določa najmanjšo možno velikost kateregakoli objekta na integriranem vezju
  - Objekt je lahko del tranzistorja, 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 (leto1971)
  - 2.250 tranzistorjev na ploščici 3,2 x 4,2 mm
  - 10 µm proces (feature size 10 µm = $10x10^{-6}$ 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

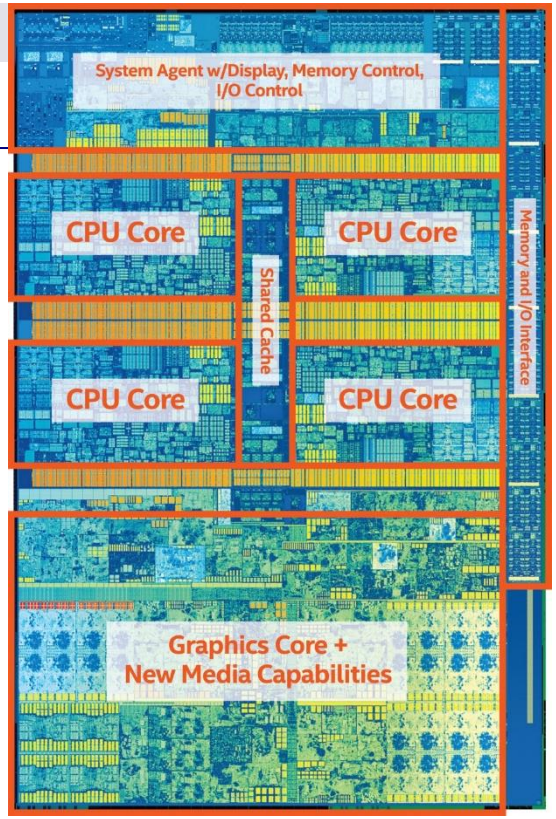


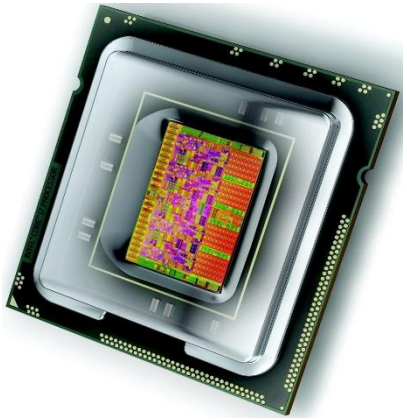
Primer 2:

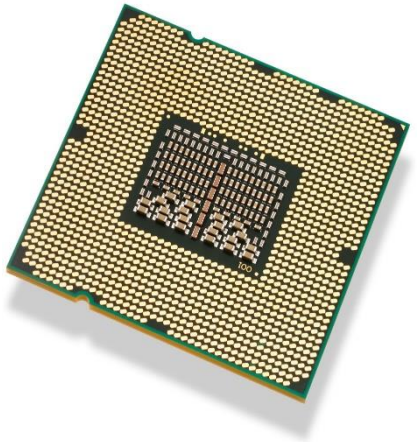
- Procesor Intel i7 7700

  (mikroarhitektura Kaby Lake 7. generacija leto 2017):

  - Število transistorjev - Intel tega podatka ne objavlja več
  - 14 nm proces (14nm = 14×10^{-9} m = 0,000000014 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 $







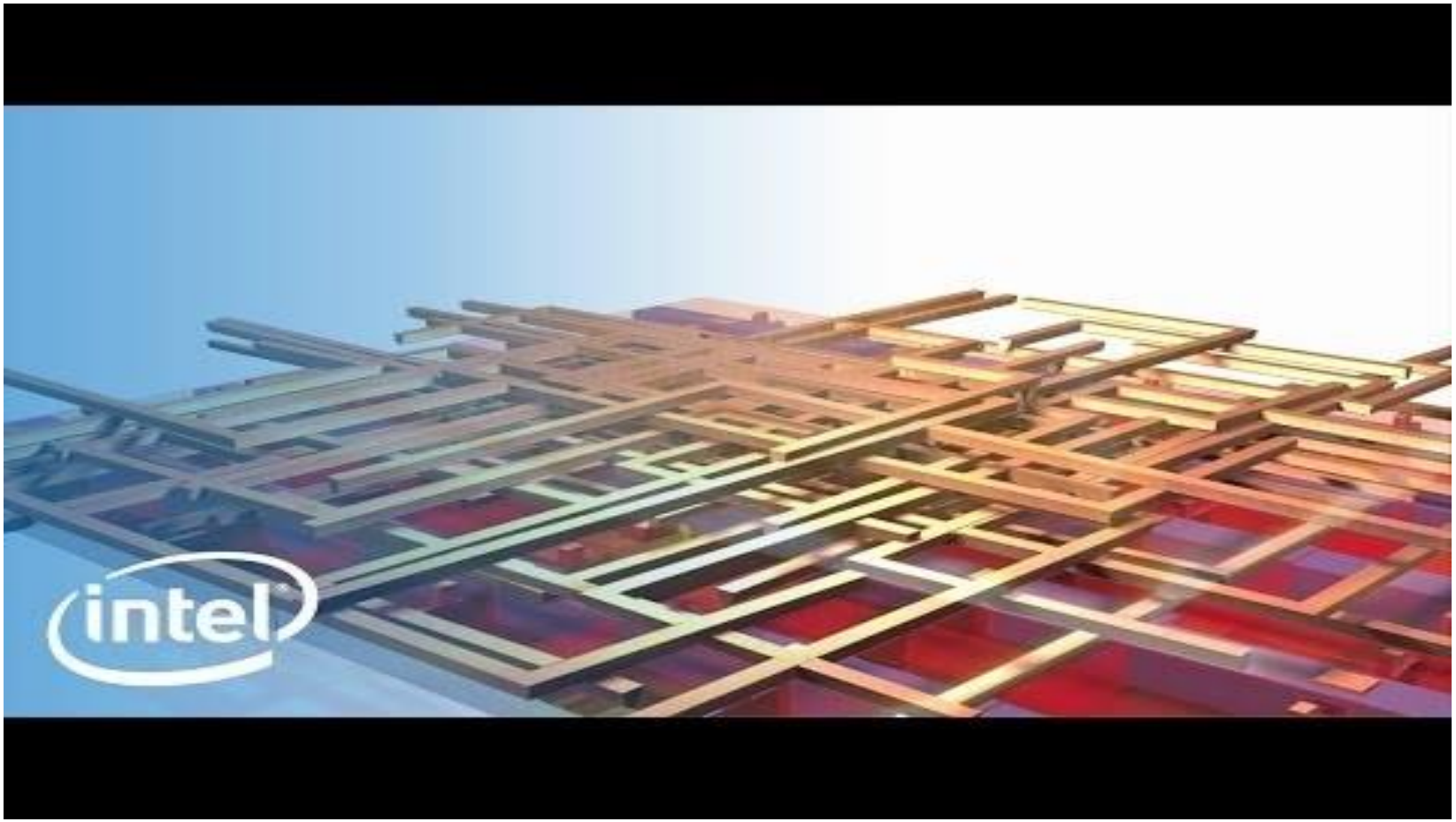
Primer 3:

- Procesor Intel i9-11900

  (mikroarhitektura Rocket Lake 11. generacija leto 2021):

  - Število transistorjev - Intel tega podatka ne objavlja več
  - 14 nm proces (14nm = 14×10^{-9} m = 0,000000014 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

