

# Grada računala

izv. prof. dr. sc. Tomislav Hrkać

Fakultet elektrotehnike i računarstva

email: [tomislav.hrkac@fer.hr](mailto:tomislav.hrkac@fer.hr)

# Građa računala – sadržaj predmeta

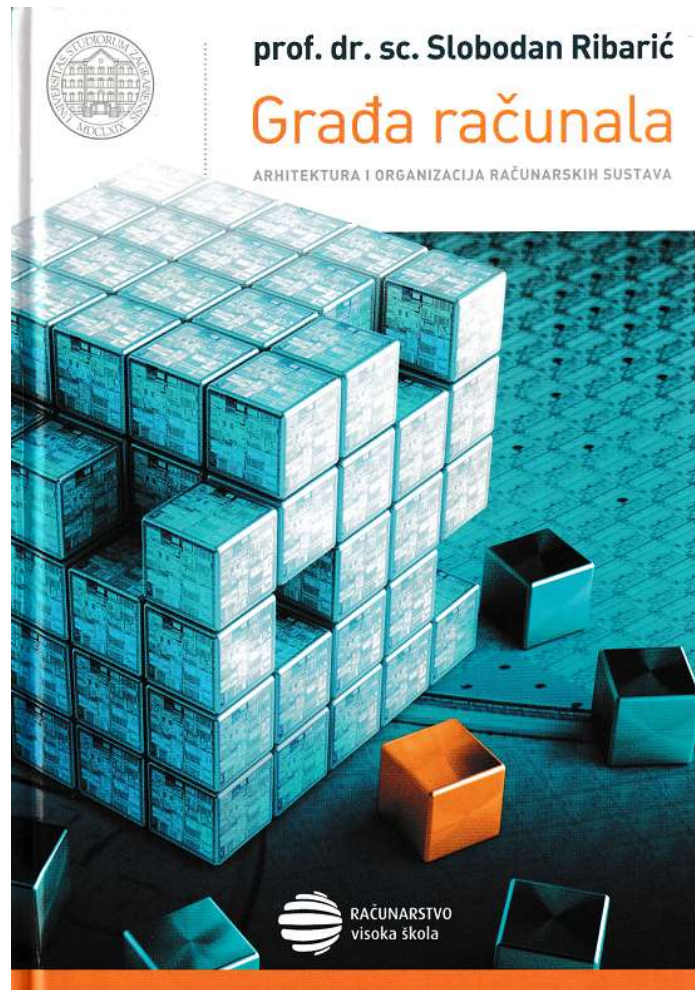
1. Uvodno predavanje (Arhitektura, organizacija i građa računarskog sustava, Flynnova klasifikacija arhitekture)
2. Turingov stroj
3. Von Neumanov model računala
4. Pojednostavljeni modeli CISC i RISC procesora
5. Performansa računala
6. Zbirni jezik i prevodioci za zbirni jezik
7. Aritmetičko-logička jedinica

# Grada računala – sadržaj predmeta

- 8. Upravljačka jedinica  
(mikroprogramska i sklopovska izvedba)
- 9. Memorijski sustav
- 12. Priručna (cache) memorija
- 13. Virtualna memorija
- 14. U/I podsustav
- 15. Protočnost
- 16. Višeprosesorski sustavi, višejezgreni i grafički procesori

# Osnovna literatura (1)

1. S. Ribarić, *Građa računala, Arhitektura i organizacija računarskih sustava*, Algebra, Zagreb, 2011. (542 stranice)



# Osnovna literatura (2)

2. S. Ribarić, *Zbirka riješenih zadataka iz Građe računala, Arhitekture i organizacije računarskih sustava*, Merkur A.B.D. 2017. (399 stranica)



## Dodatna literatura

3. D. A. Patterson, J. L. Hennessy, *Computer Organization & Design, The Hardware/Software Interface*, Morgan Kaufmann Pub., Second edition, 2014.
4. J. L. Hennessy, D. A. Patterson, *Computer Architecture, A Quantitative Approach*, Morgan Kaufmann Pub., Fifth edition, 2011.
5. A. S. Tanenbaum, *Structured Computer Organization*, Prentice-Hall, 2006.
6. S. Ribarić, *Naprednije arhitekture mikroprocesora*, Element, Zagreb, 2.izdanje, 1997.
7. S. Ribarić, *Arhitektura RISC i CISC računala*, Školska knjiga, Zagreb, 1996.

# Organizacija predmeta

## Predavanja

- 2 sata tjedno
- pohađanje nije obavezno
- prezentacije dostupne na stranicama predmeta

## Vježbe

- 2 sata tjedno
- pravila u dogovoru s asistentom

## Domaće zadaće

- 4 zadaće u semestru, gradivo vježbi

## Projektni zadatak

# Pravila polaganja

Dokument „Pravila ocjenjivanja...”



# Arhitektura, organizacija i građa računarskog sustava

(Građa računala, str. 18-29)

- Definicija arhitekture računarskog sustava

Izraz *arhitektura računala* (engl. *computer architecture*) pojavio se šezdesetih godina prošlog stoljeća u jednoj od vodećih svjetskih tvrtki na području IT (informacijske tehnologije), IBM-u (International Business Machines).

Upotrebljavao se za opisivanje programskog modela računala iz serije IBM 360 na razini *asemblera* (engl. *assembly language*), tj. Zbirnog jezika koji je vrlo blizak *strojnom jeziku*.

**Strojni jezik** (engl. *machine language*) sastoji se od instrukcija koje su razumljive sklopovlju računala i koje su kodirane sljedovima 0 i 1.

**Zbirni jezik** (ili asemblerski jezik) je jezik kojim se strojne instrukcije predočavaju obično dvo- ili troslovčanim oznakama, koje se nazivaju *mnemonici*, te je time omogućen prikaz programa za ljude u čitljivijem obliku.

### **Primjer 1.8. (Građa računala, str. 19)**

Instrukcija kojom se zbrajaju dva broja bit će u zbirnom jeziku predočena

*add r3, r2, r1*

gdje mnemonik *add* specificira operaciju zbrajanja (troslovčana oznaka nas asocira na zbrajanje; *add* u engleskom znači zbroji), a *r1* i *r2* određuju dva izvora podataka (registre), a odredište rezultata određeno je kao registar *r3*.

Ova će se instrukcija prevesti, na primjer, u strojnu instrukciju oblika:

00100000110001000001000000000000

U ovom je slučaju strojna instrukcija niz od 32 bita.

Dva su računala imala istu arhitekturu ako su imala jednak skup strojnih instrukcija. Na ovaj način opisana arhitektura računala obično se označava kao **ISA (*Instruction Set Architecture*)** i sadržava opise elemenata kao što su skup strojnih instrukcija, tipovi podataka koji su izravno podržani sklopovljem procesora, načini adresiranja te registri “vidljivi” programeru na razini instrukcija u zbirnom jeziku.

S vremenom se pojam *arhitektura računala* proširio tako da se ona nešto općenitije može definirati kao **slika računarskog sustava kojom se sustav predočava programeru u strojnom jeziku**, ali i programeru koji piše prevodioc (engl. *compiler*) za više programske jezike kojim se naredbe u višim programskim jezicima prevode u strojne instrukcije. Odnosno, program napisan u višem programskom jeziku (npr. u C; C++ ili Pascalu) prevodi se u slijed strojnih instrukcija koji se naziva *objektni (binarni) program* (engl. *Object (binary) program*).

Neki znanstvenici pod *arhitekturom računala* podrazumijevaju **algoritme** koji se rabe u osnovnim funkcijskim jedinicama računarskog sustava (upravljačka jedinica, aritmetičko-logička jedinica, ulazno-izlazna jedinica, memorija) bez obzira jesu li algoritmi ostvareni sklopovljem ili programima (dualizam!).

E. C. Joseph arhitekturu računala promatra **funkcionalno usmjerenom**, poput arhitekture u modernom građevinarstvu.

Ciljeva oblikovanja:

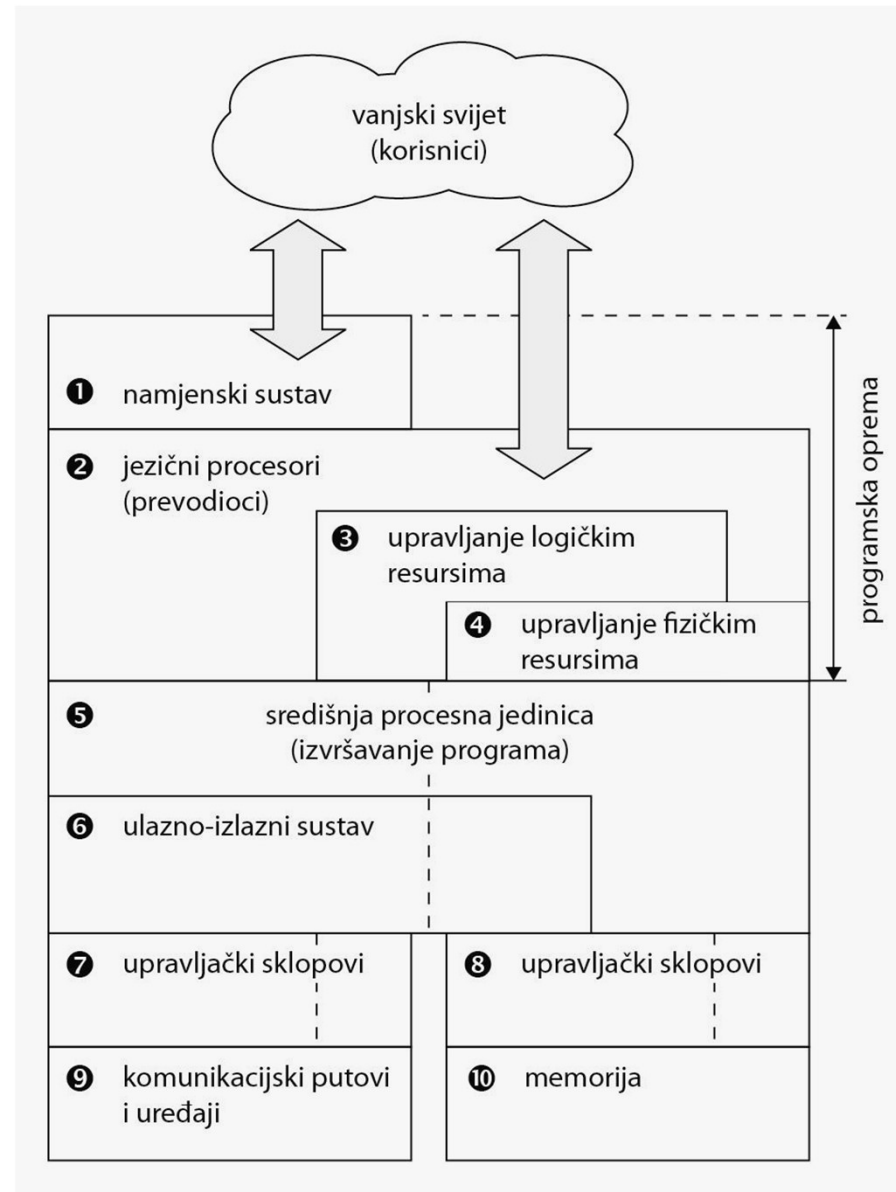
- **povećanje propusnosti** (engl. *throughput*), odnosno povećanje procesne moći ili performanse sustava,
- **prilagodljivosti** (engl. *flexibility*) – mogućnost uporabe sustava na različitim područjima ljudske djelatnosti,
- **pouzdanosti** (engl. *reliability*) – što je moguće veći vremenski razmak između dva ispada iz rada uslijed sklopovske ili programske pogreške i
- **raspoloživosti** (engl. *availability*) – što dulje vrijeme u kojem je sustav raspoloživ korisniku (vrijeme u kojem računarski sustav ispravno djeluje, a ne nalazi se u postupku ispitivanja ili redovitog održavanja).

Jedan od vrlo važnih ciljeva arhitekture računala, posebno s gledišta proizvođača računarskih sustava, jest i postizanje što je moguće **niže cijene** sustava.

Točnije, postizanje što je moguće većeg **omjera između performanse računarskog sustava i njegove cijene**.

- **ekološko zbrinjavanje računalnog otpada**, odnosno izbor materijala za izgradnju računala koji je ekološki razgradljiv ili se može reciklirati.

Arhitektura računarskog sustava može se promatrati kao distribucija funkcija po zadanim **razinama višeslojnog modela**, gdje svaka razina predstavlja određenu apstraktnu predodžbu računala.



Prva, najviša razina naziva se *arhitektura sustava* i predstavlja sliku sustava s motrišta korisnika koji preko namjenskog sustava koristi računalo.

Namjenski sustav čine namjenski programi koje obično isporučuje proizvođač zajedno s računalom ili programi koji se mogu nabaviti od nezavisnih isporučitelja (“programskih kuća”).

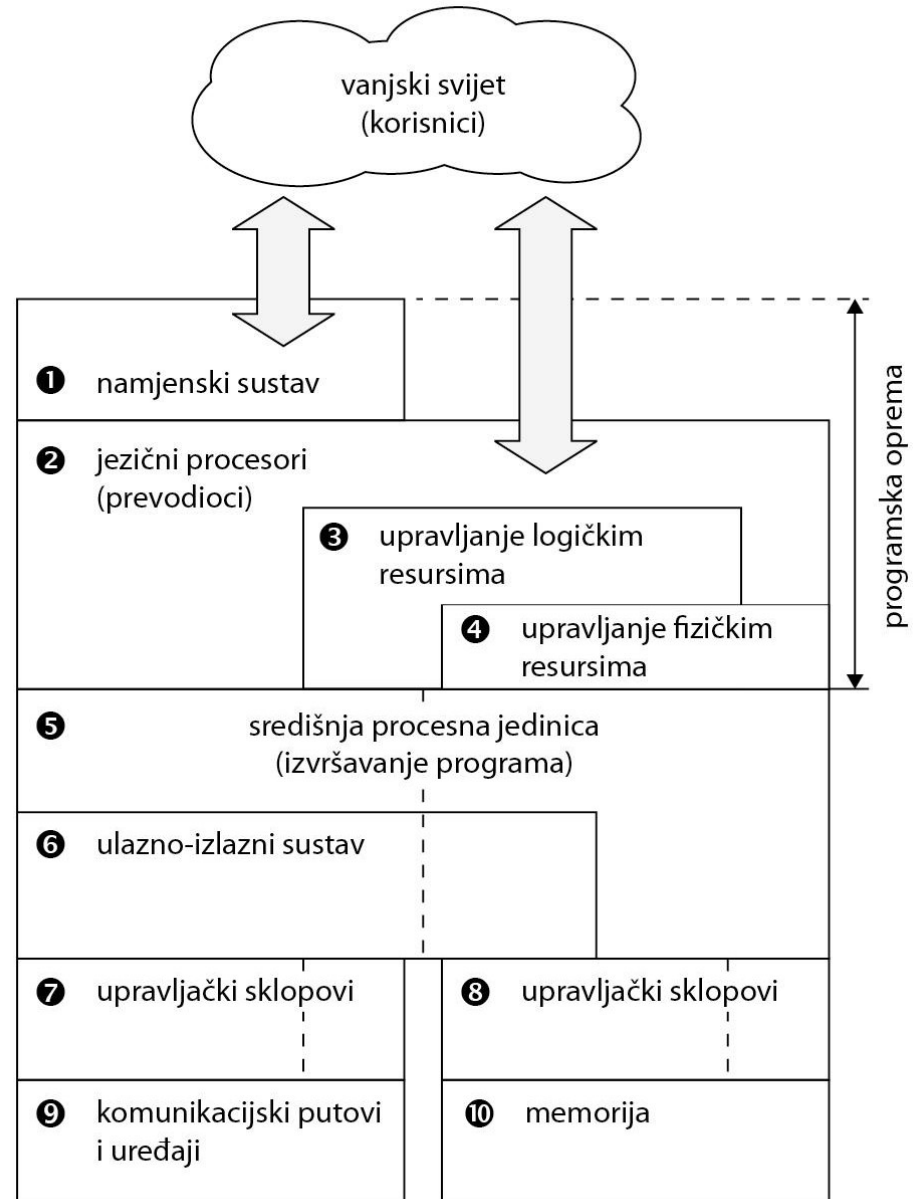
Tipični **namjenski programi** jesu oni za **pretraživanje i sortiranje podataka**, **pretraživanje teksta**, programi kojima se rješavaju problemi s područja matematike, tehnike i poslovanja itd.

Na nešto nižoj razini apstrakcije računalo se korisniku predstavlja preko **viših programskih jezika**.

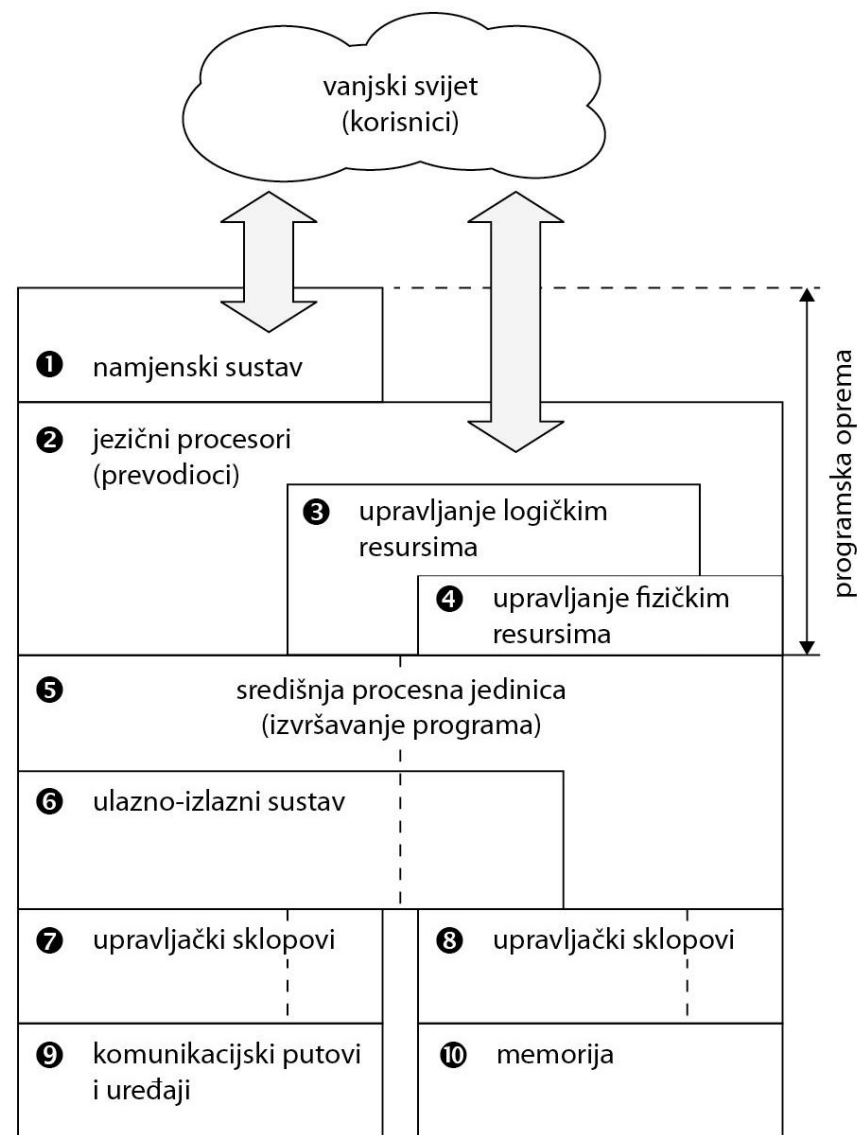
U tom se slučaju komunikacija, odnosno interakcija između korisnika (programera) i računala ostvaruje pomoću jezičnih procesora koji mogu podržavati različite vrste jezika (viši programski jezici kao što su C, C++, Pascal, FORTRAN, Basic, Logo i sl., upravljački jezici, jezici za oblikovanje i rukovanje bazama podataka).



Razine 1. – 4. odnose se na funkcije koje su (obično) ostvarene programski s tim da su **jezični procesori**, **upravljanje logičkim resursima** i **upravljanje fizičkim resursima** - razine koje odgovaraju funkcijama **operacijskog sustava računala**.



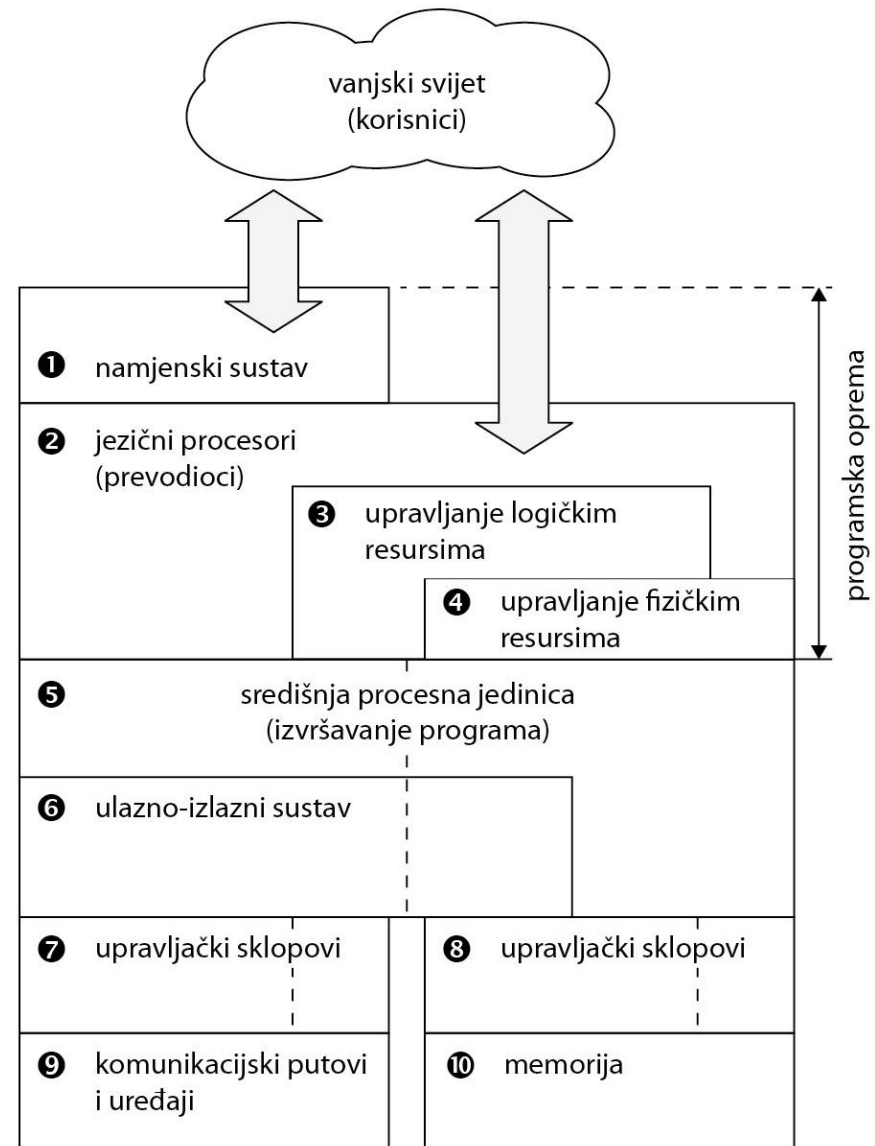
**Upravljanje logičkim resursima** obuhvaća funkcije kao što su upravljanje virtualnom memorijom, upravljanje bazama podataka, upravljanje postupcima obrade u mrežama računala, dok **upravljanje fizičkim resursima** podrazumijeva funkcije rukovanja memorijom (glavnom ili radnom, sekundarnom), upravljanje procesorom (njegovo dodjeljivanje, odnosno dodjeljivanje vremena procesora različitim programima) i upravljanje drugim uređajima računarskog sustava.



Razine 5. – 10. odnose se na funkcije koje su ostvarene sklopovski.

Razina 5. koja se odnosi na funkciju izvršavanja programa obično se naziva *arhitektura procesora* ili *arhitektura središnje procesne jedinice*, dok se razine 6. – 9. odnose na funkcije koje se klasificiraju u *arhitekturu ulazno-izlaznog sustava i računalnih mreža*.

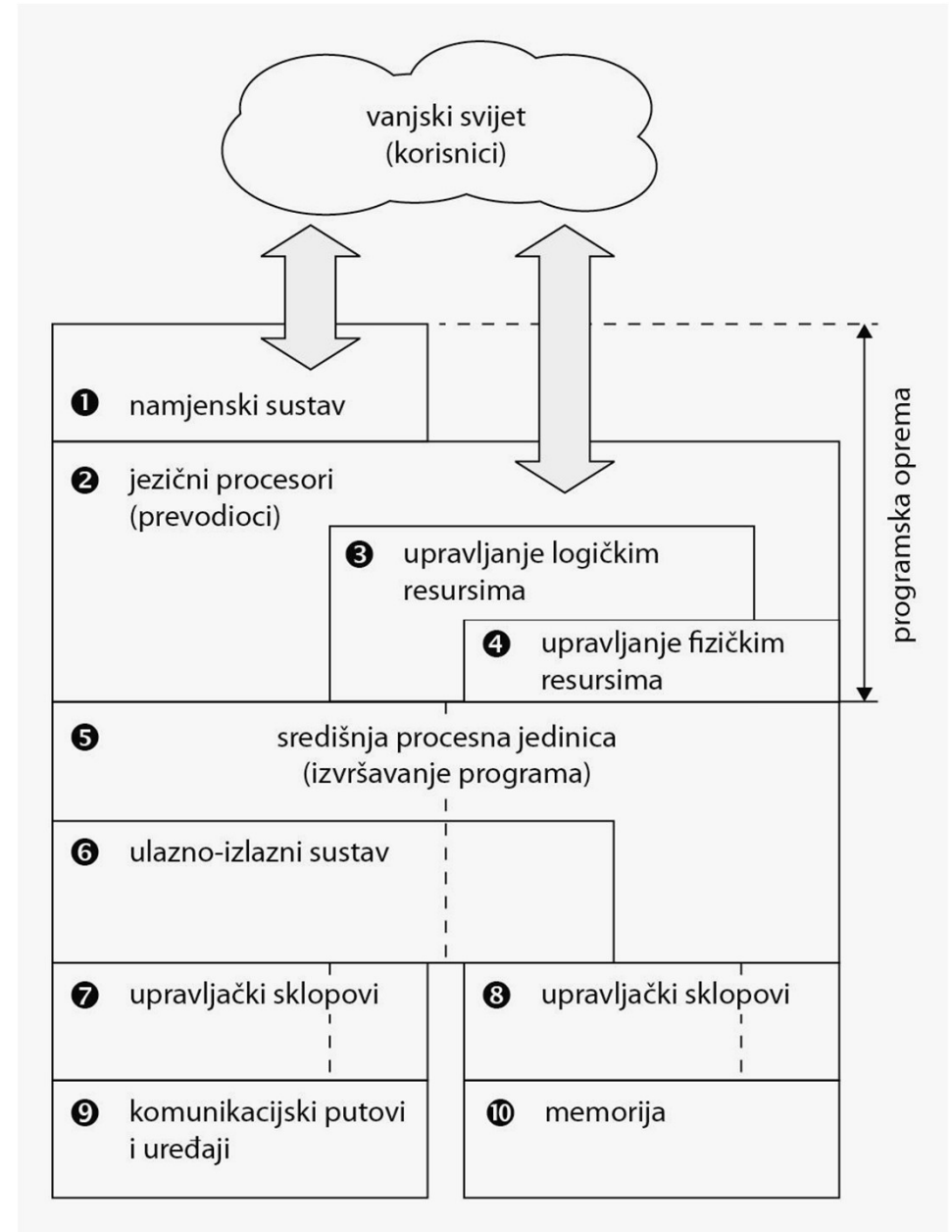
Funkcije u razinama 8. i 10. odnose se na *arhitekturu memorijskog sustava*.



Crtkane linije određuju funkcije koje su zaista realizirane **sklopovljem** i one koje su ostvarene “programski”, odnosno **mikroprogramom**.

Naime, na području arhitekture računala može se govoriti o tzv. **dualizmu sklopovske i programske opreme** (kao što se u modernoj fizici govori o dualizmu mase i energije).

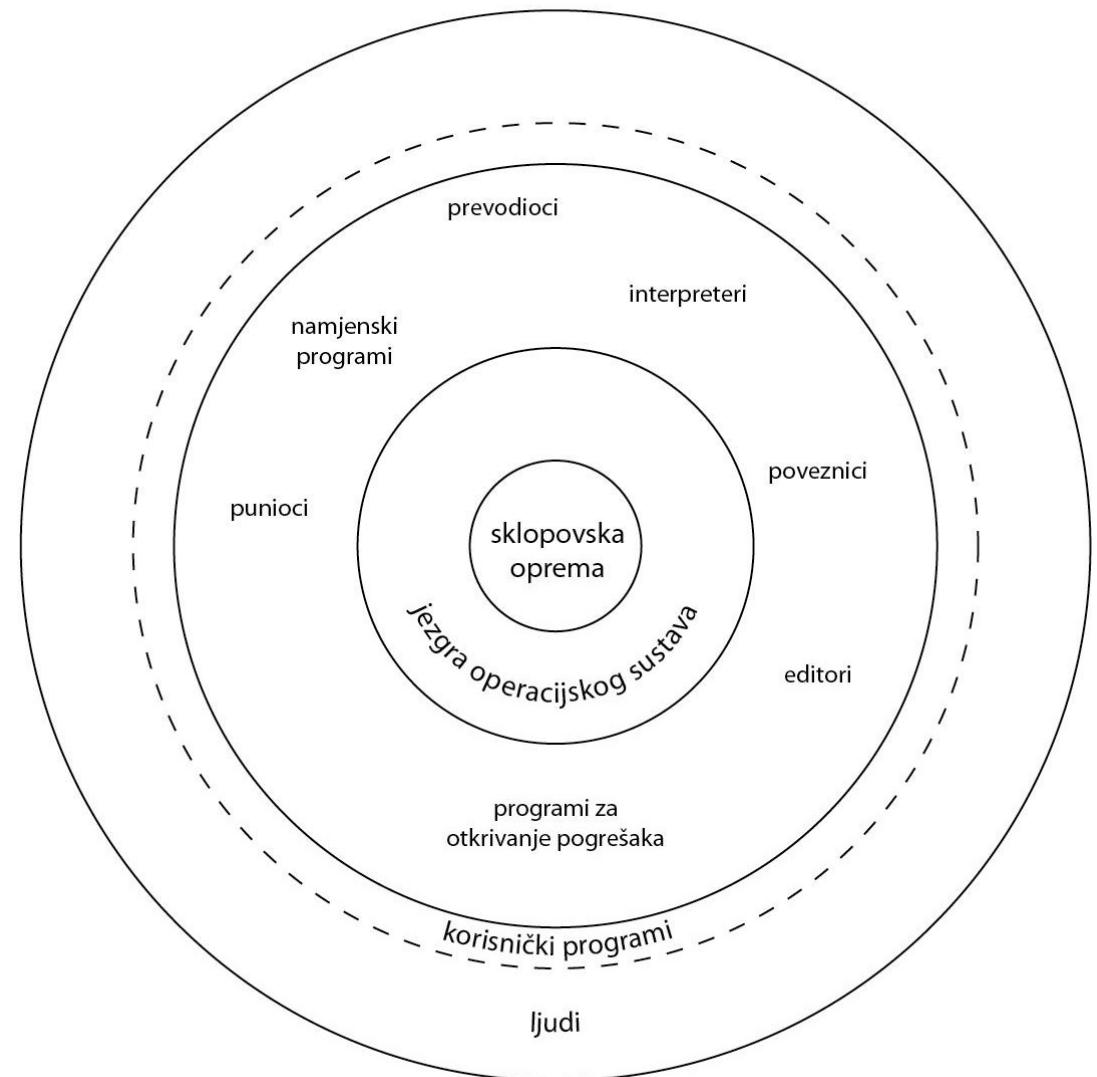
Pojednostavljeno, dualizam sklopovske i programske opreme očituje se u tome da se skoro sve što je realizirano u sklopovlju može ostvariti i programski, i obratno.



Definiciju arhitekture računarskog sustava temeljit ćemo na hijerarhijskom modelu sustava:

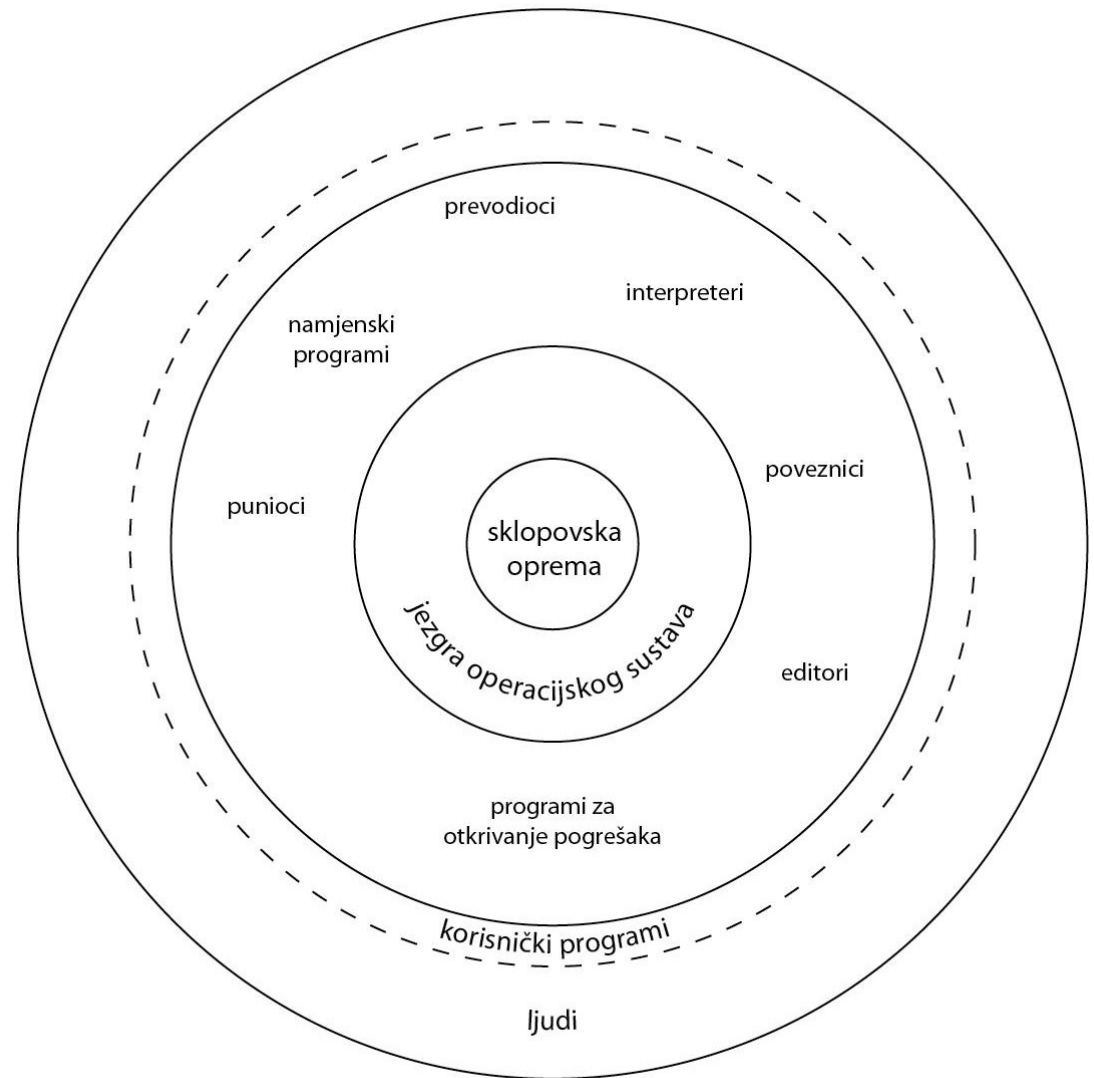
Najnižu razinu, predočenu jezgrom prstenaste strukture, čini **sklopovlje**, odnosno sklopovska oprema ili **hardver**.

Općenito, pod tim se razumijeva sve materijalno od čega je računalo sastavljeno: svi **mehanički dijelovi**, **električki** i **elektronički** dijelovi.

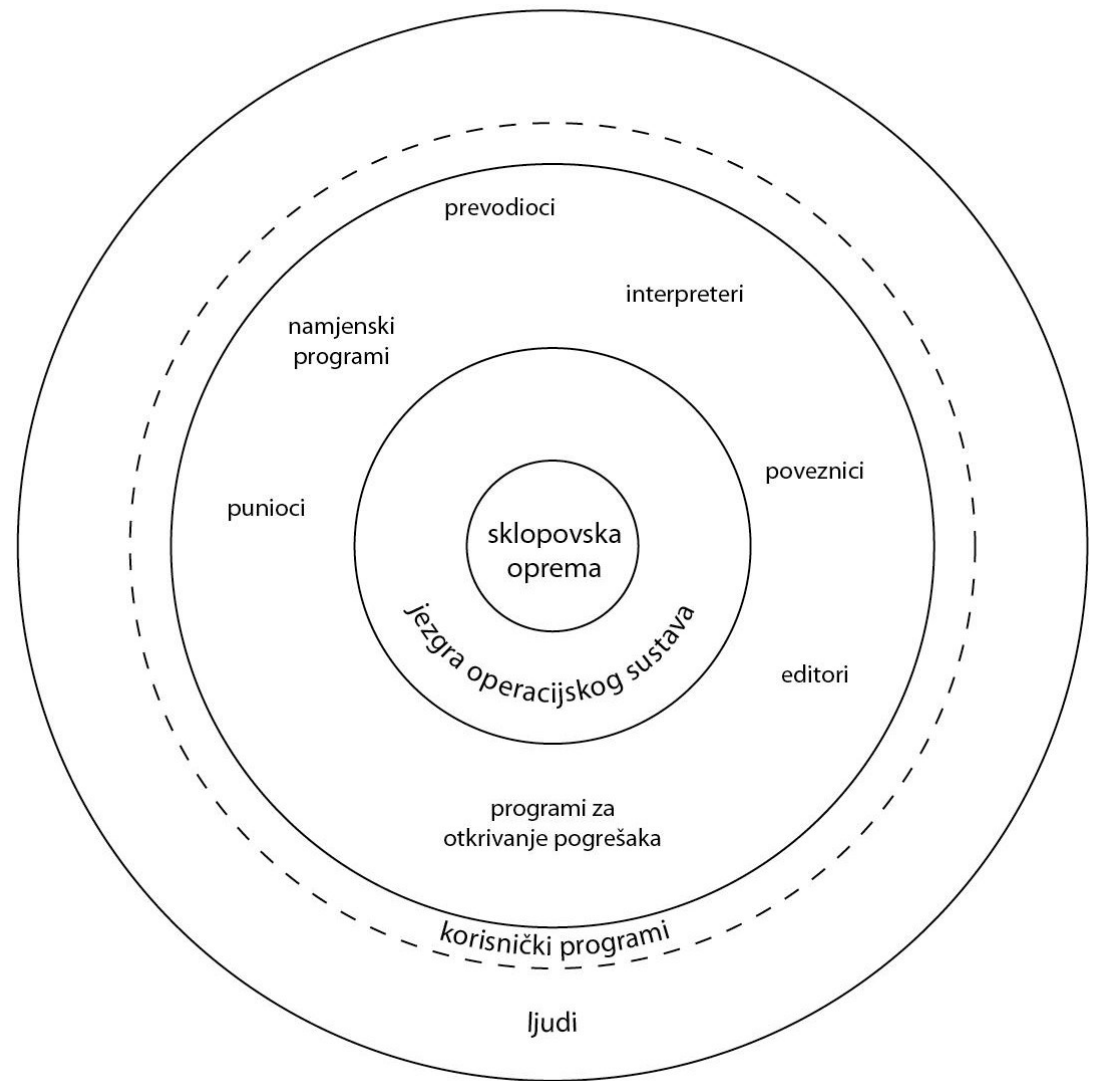


U užem smislu, pod sklopovskom ćemo opremom podrazumijevati elektronićke sastavne dijelove: sklopove realizirane u tehnologiji visokog (LSI – *Large Scale Integration*) i vrlo visokog stupnja integracije (VLSI – *Very Large Scale Integration*).

Sljedeću razinu u modelu čini *jezgra operacijskog sustava*. Zadatak je jezgre operacijskog sustava da **rukovodi i dodjeljuje osnovne resurse računala** (procesor, odnosno vrijeme procesora, memoriju, ulazno-izlazne uređaje, sistemske programe i datoteke) na temelju potreba zadatka, procesa i dretvi (engl. *thread*) različitih (korisničkih) programa koji se istodobno izvršavaju u računarskom sustavu.



Sljedeću višu razinu modela čine preostali dijelovi operacijskog sustava kao što su **prevodioci** za različite više programske jezike (C, C++, Pascal, FORTRAN, Basic, Prolog, Lisp, ...), **interpreteri** (engl. *interpreter*), **punioci** (engl. *loader*), **poveznici** (engl. *linker*), **uređivači teksta** (editori; engl. *editor*), **programi za otkrivanje pogrešaka** (engl. *debugger*) i već spomenuti namjenski programi.





*Interpreter* je program koji analizira naredbu po naredbu programa i za razliku od prevodioca, izvršava je umjesto da generira objektni program koji će se kasnije izvršiti.

*Punilac* jest uslužni program kojim se izvršljivi program tj. objektni binarni program smješta u glavnu memoriju i time biva spreman za izvođenje.

*Poveznik* je uslužni program za povezivanje kojim se različiti prevedeni programi ili programski moduli povezuju u oblik pogodan za izvršavanje.

Sastavnice arhitekture računarskog sustava:

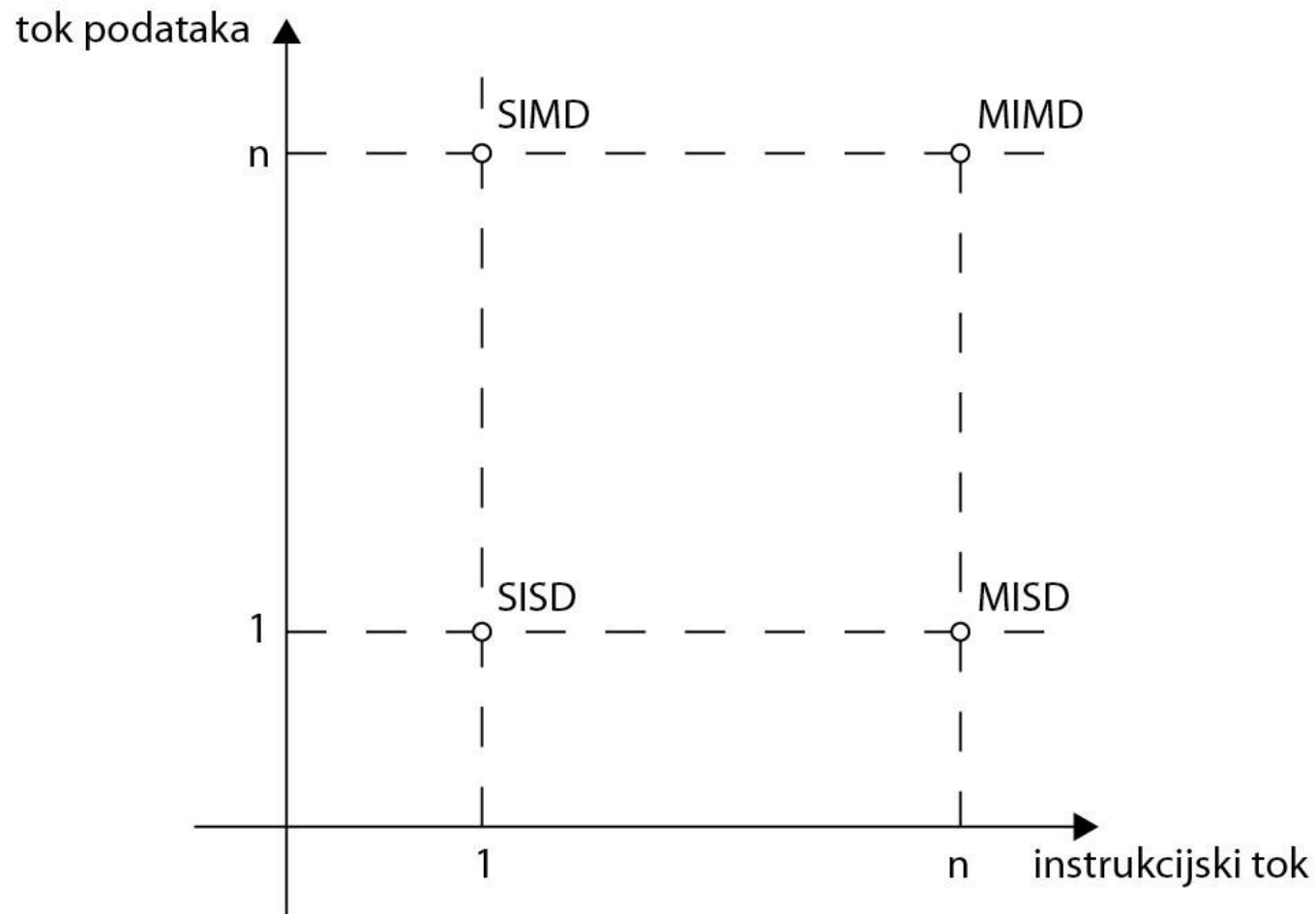
- sklopovlje (engl. hardware)
- programska oprema (engl. software)
- sučelje čovjek – stroj (engl. humanware)

**Arhitektura računarskog sustava** je dio računarske znanosti koji se odnosi na oblikovanje računarskih sustava radi ostvarivanja osnovnih ciljeva kao što su veća performansa, prilagodljivost, pouzdanost i raspoloživost uz što je moguće nižu cijenu, odnosno, što je moguće veći omjer performansa/cijena.

To se postiže uporabom niza tehnika, postupaka i zahvata u svim hijerarhijskim razinama – od sklopovlja, programske oprema pa sve do razine interakcije čovjeka i stroja (engl. humanware).

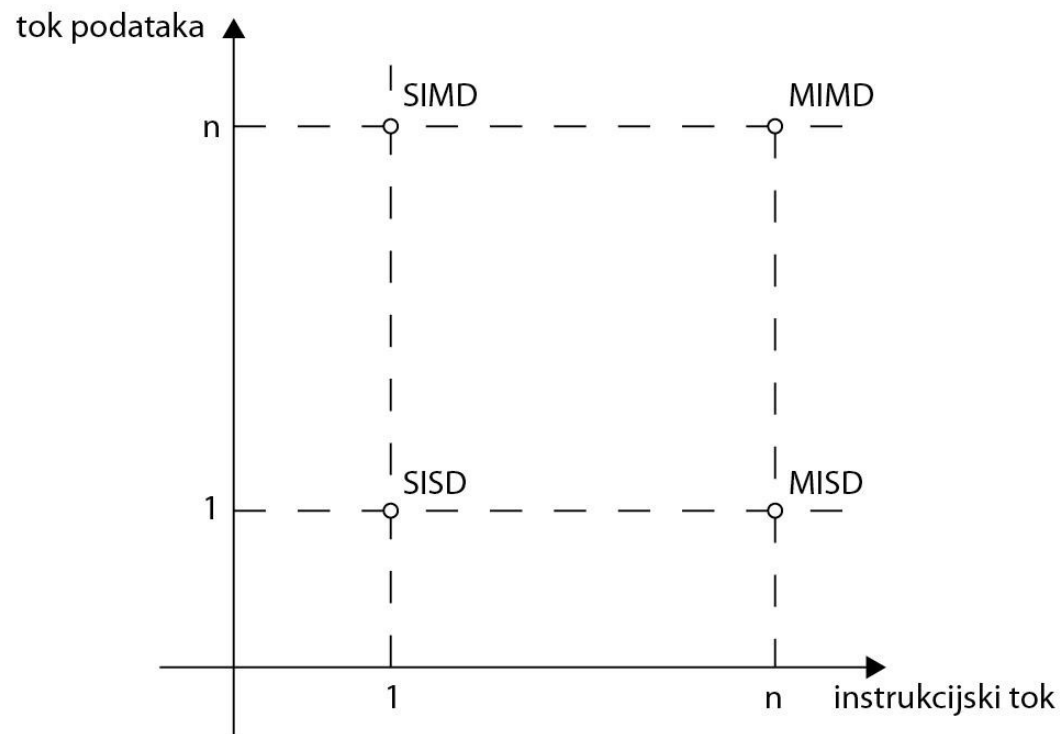
## Flynnova klasifikacija (Građa računala, str. 25 – 29)

Algoritam se stroju predočava programom, a njegovo je izvršavanje definirano **slijedom instrukcija**. Na određenoj razini apstrakcije, dinamički promatrano, tijekom izvođenja programa možemo govoriti o ***toku podataka (DS)*** i ***instrukcijskom toku (IS)*** koji se uspostavljaju između funkcijskih jedinica stroja.



**SISD** (***S**ingle **I**nstruction Stream **S**ingle **D**ata Stream*) – računalo s jednostrukim instrukcijskim tokom i jednostrukim tokom podataka. Arhitektura SISD predstavlja arhitekturu sekvencijalnog računala temeljenog na von Neumannovom računskom modelu

**SIMD** (***S**ingle **I**nstruction **S**team **M**ultiple **D**ata **S**team*) – računalo s jednostrukim instrukcijskim tokom i višestrukim tokom podataka. U ovu se kategoriju svrstavaju paralelna računala (nazivaju se i matrična računala) koja se obično sastoje od velikog broja procesora ili procesnih elemenata koji istodobno izvršavaju istu instrukciju na različitim podacima.

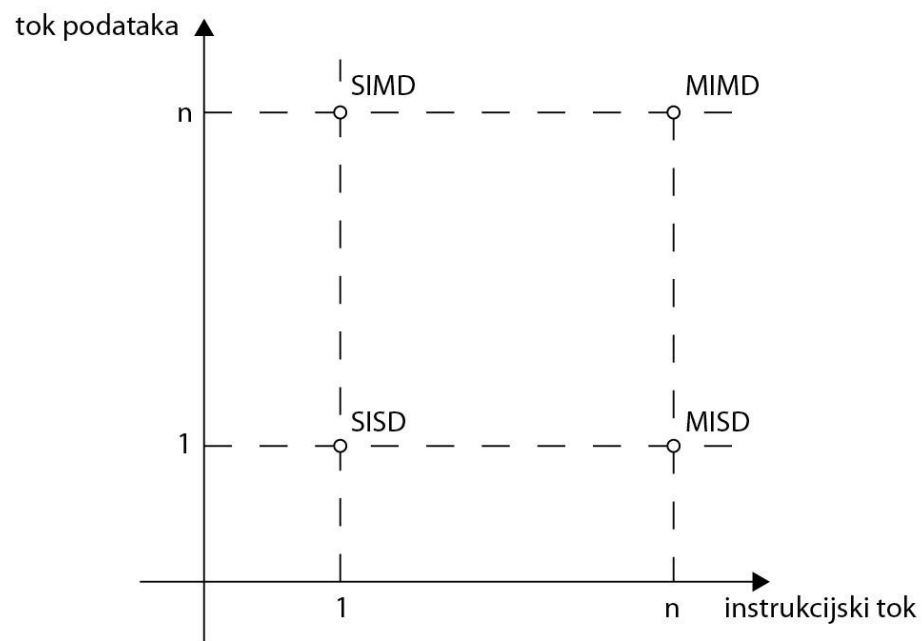


**MISD** (***M**ultiple **I**nstruction **S**team **S**ingle **D**ata **S**team*) – računalo s jednostrukim instrukcijskim tokom i višestrukim tokom podataka. Teorijski strogo gledano, računala ovog tipa arhitekture ne mogu se fizički realizirati. Nemoguće je, naime, ostvariti da se istodobno više različitih instrukcija izvršava na istim podacima (slično kao što u fizici makro svijeta nije moguće da se dva različita tijela istodobno nalaze na istom mjestu).

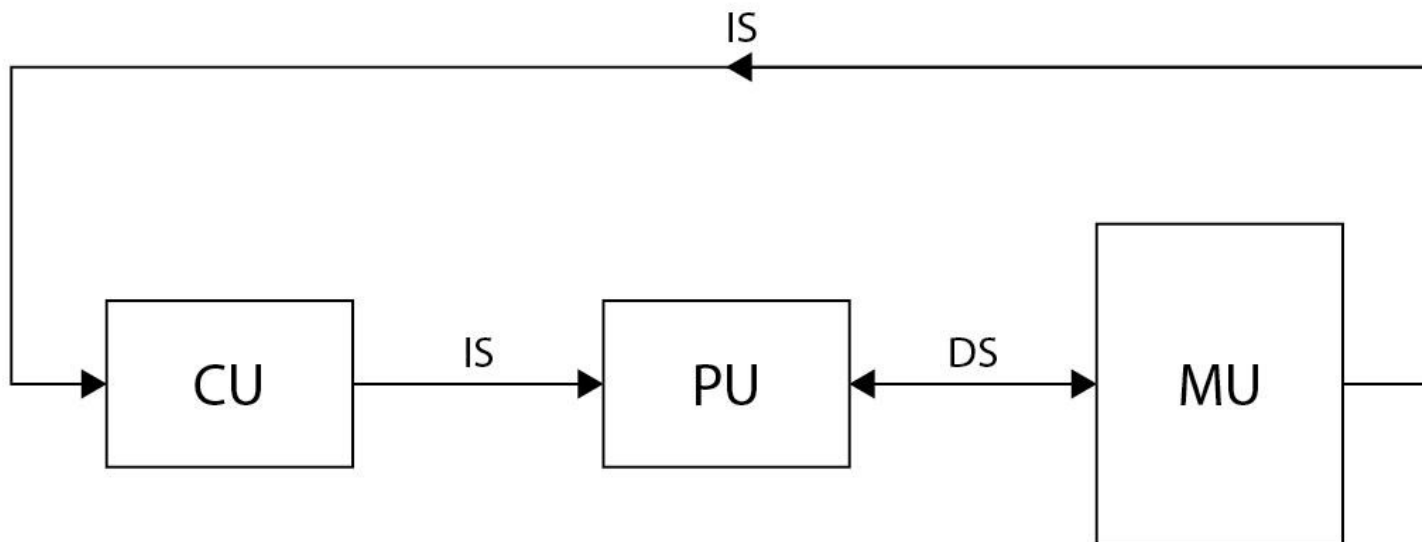
No dogovorno u ovu kategoriju uvrštavamo **protočna** (engl. *pipeline*) računala i računarske sustave koji se temelje na **sistoličkim poljima**.

**MIMD** (***M**ultiple **I**nstruction Stream **M**ultiple **D**ata Stream*) – računalo s višestrukim instrukcijskim tokom i višestrukim tokom podataka.

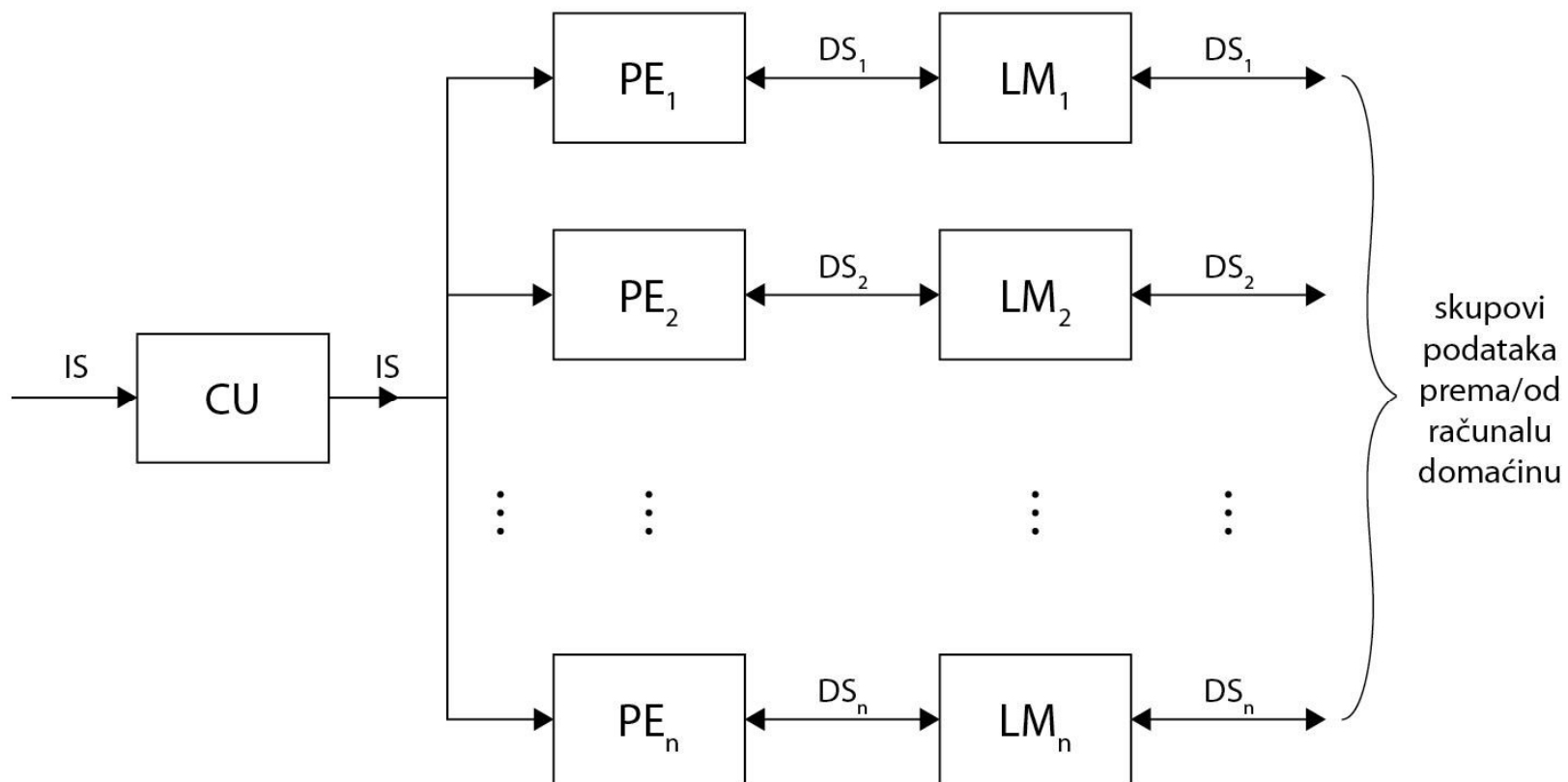
**Multiprocesorski sustavi**, odnosno **paralelni računarski sustavi** s dva i više procesora približno jednakih performansi, pri čemu svaki od njih ima pristup zajedničkoj memorijskoj jedinici i svi dijele ulazno-izlazne jedinice, a pritom djeluju pod jednim operacijskim sustavom predstavljaju ovaj tip arhitekture.



SISD

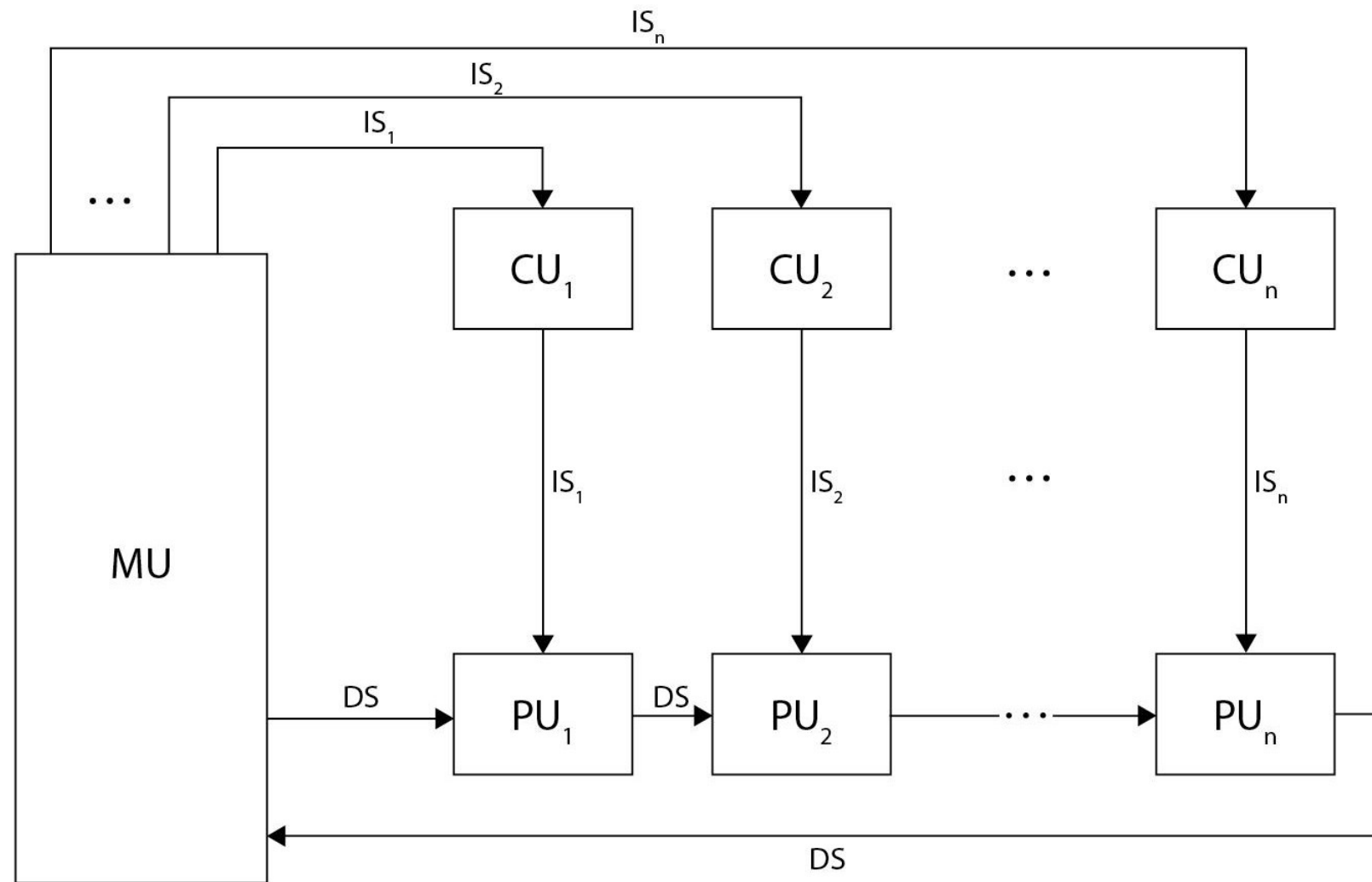


## SIMD





# MISD



# MIMD

