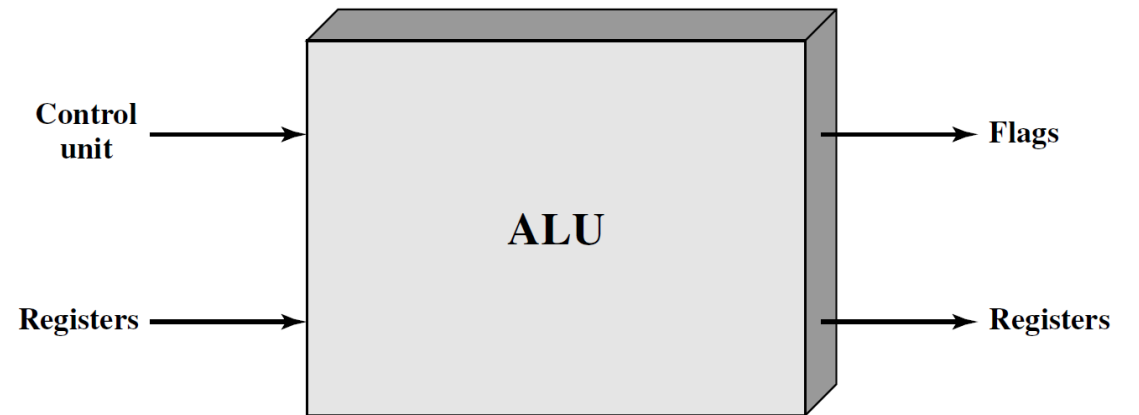


# ARITMETIČKO-LOGIČKA JEDINICA (ALU)

- Osnovni zadatak ovog dela procesora je izvršavanje osnovnih aritmetičkih i logičkih operacija.
- Funkcionisanje ALU je dirigovano i kontrolisano od strane upravljačke jedinice.
- Podaci (operandi) koji su predmet obrade se preko MBR prenose iz operativne memorije u AC.
- Po okončanju procesa obrade rezultat se upisuje u AC, a zatim se ili koristi za sledeću operaciju ili distribuira na odgovarajuću adresu u operativnoj memoriji.

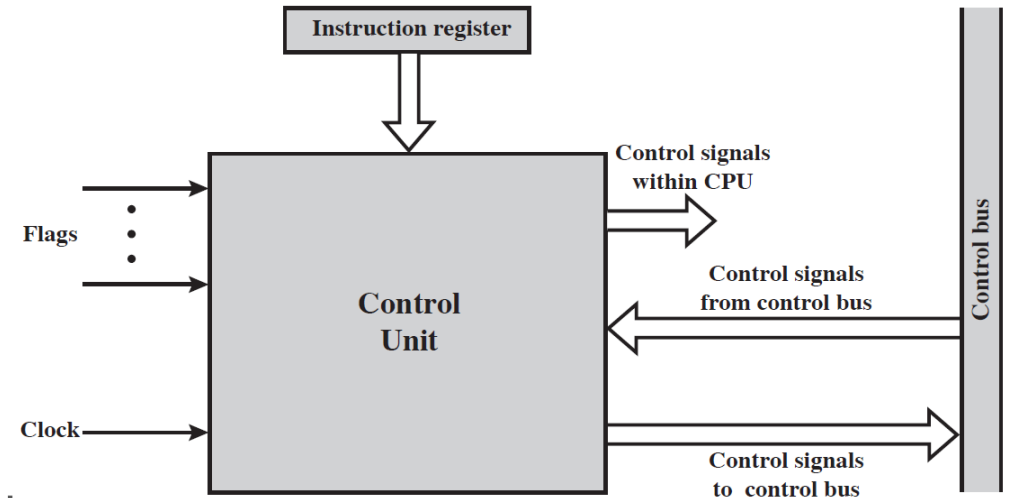


# UPRAVLJAČKO-KONTROLNA JEDINICA

- Glavni supervizor svih procesa koji se odvijaju u CPU.

Ulazi:

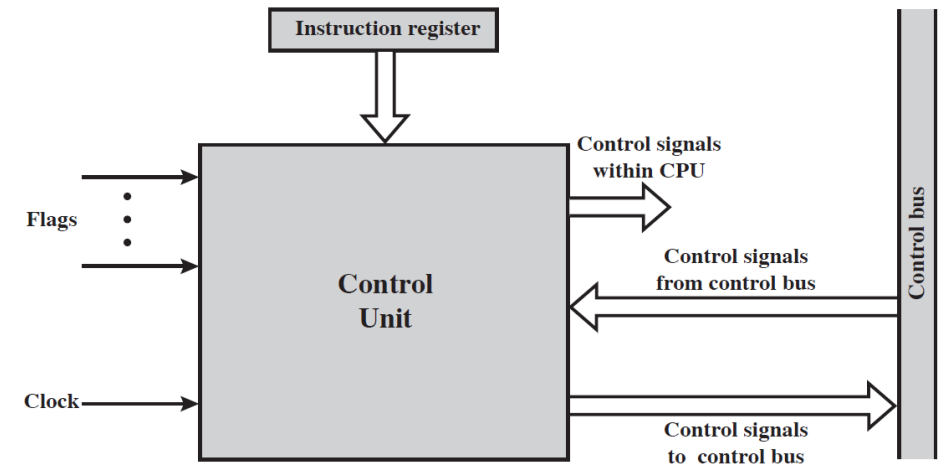
- **Takt sata:** Jedna mikro-operacija (ili više istovremenih mikro-operacija) se izvršava na jedan takt sata.
- **Instrukcioni registar:** Operacioni kod i način adresiranja trenutne instrukcije upućuju na to koje mikro-operacije treba da budu izvršene tokom ciklusa instrukcije.
- **Flegovi:** Potrebni su da bi kontrolna jedinica utvrdila u kom je statusu procesor i kakav je rezultat prethodnih ALU operacija.
- **Kontrolni signali sa magistrale kontrole**



# UPRAVLJAČKO-KONTROLNA JEDINICA

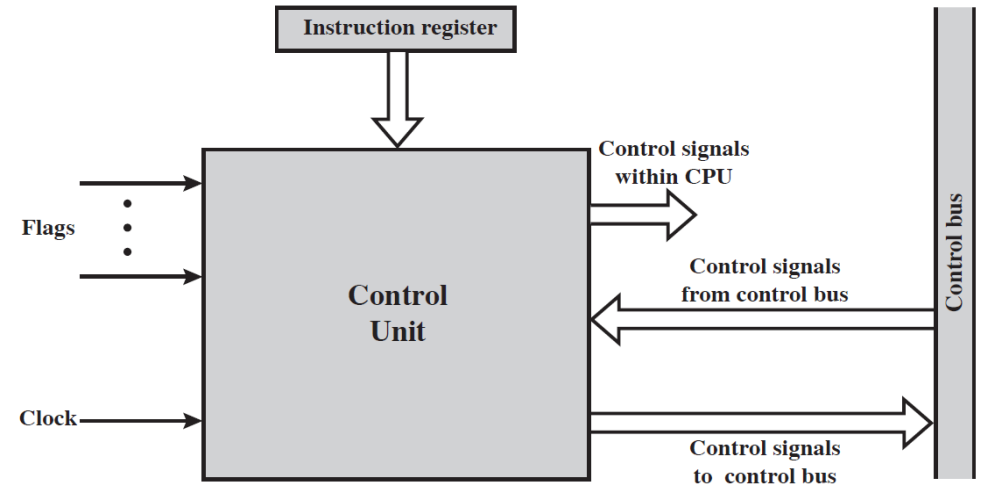
Izlaz iz upravljačko-kontrolne jedinice su

- **Kontrolni signali unutar procesora:**
  - Signali koji zahtevaju prenošenje podataka između registara
  - Signali koji aktiviraju određene ALU funkcije.
- **Kontrolni signali za magistralu kontrole:**
  - Signali koji aktiviraju određen put za podatke (ka/iz memorije)
  - Signali koje dolaze/odlaze sa nekog od spoljašnjih interfejsa ili neke od sistemskih magistrala.



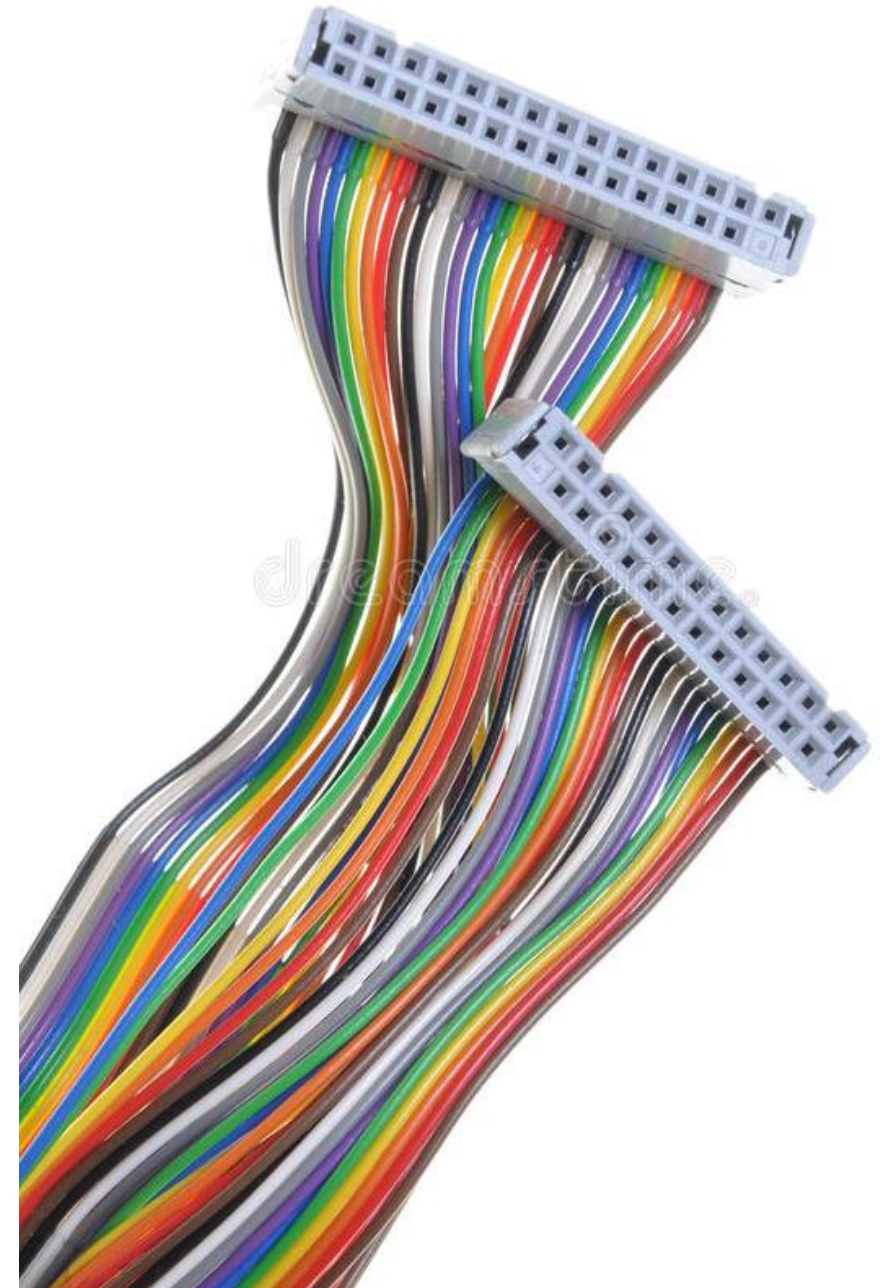
# UPRAVLJAČKO-KONTROLNA JEDINICA

- Osnovne uloge ovog dela računara su:
  - Unos podataka i programskih instrukcija iz operativne memorije
  - Upravljanje prenosom podataka između aritmetičko logičke jedinice i operativne memorije
  - Praćenje izvršenja aritmetičkih i logičkih operacija i donošenje odluka na osnovu rezultata
  - Upravljanje i praćenje rada ulazno-izlaznih jedinica



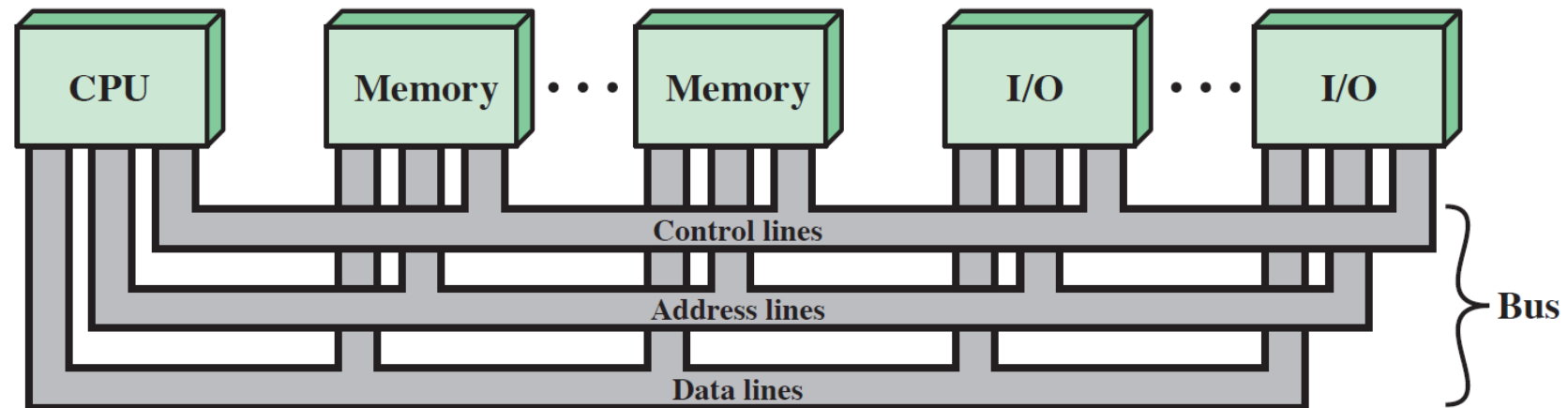
# MAGISTRALA (BUS)

- Magistrala je komunikacioni put koji povezuje dva ili više uređaja.
- Više uređaja se povezuje na magistralu, a signal koji emituje jedan uređaj dostupan je za prijem svim ostalim uređajima priključenim na magistralu.
- Bus se sastoji od više komunikacionih puteva ili linija. Svaka linija može da prenosi električne impulse koji predstavljaju binarne 1 i 0.
- Prenos podataka između operativne memorije i CPU, ulaznih uređaja i operativne memorije kao i između operativne memorije i ekranskih uređaja se vrši posredstvom magistrale.



# MAGISTRALA (BUS)

- Linije magistrale se mogu klasifikovati u tri grupe:
- Data bus – magistrala podataka
- Address bus – magistrala adresa
- Control bus – magistrala kontrole

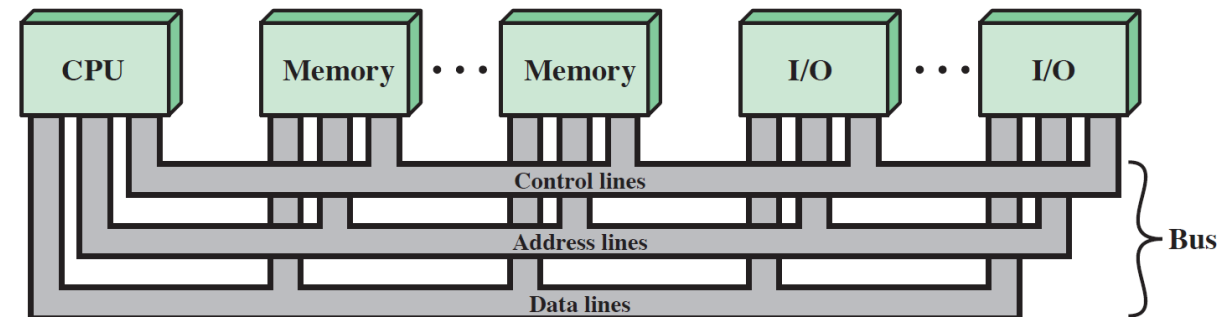


# DATA BUS - PODATKOVNA MAGISTRALA

- Linije podataka predstavljaju putanju za prenos podataka među sistemskim modulima.
- Ove linije, kolektivno, nazivaju se data bus.
- Podatkovna magistrala se može sastojati od 32, 64, 128 ili više odvojenih linija, a broj linija se naziva širina.
- Pošto svaka linija može da nosi samo jedan po jedan bit, broj linija određuje koliko bitova može biti preneto u isto vreme.
- Širina magistrale podataka je ključni faktor u određivanju ukupnih performansi sistema.
- Na primer, ako je data bus širine 32 bita a svaka instrukcija je 64 bita dugačka, onda procesor mora dva puta pristupiti memorijskom modulu tokom svakog ciklusa instrukcija.

# ADDRESS BUS — ADRESNA MAGISTRALA

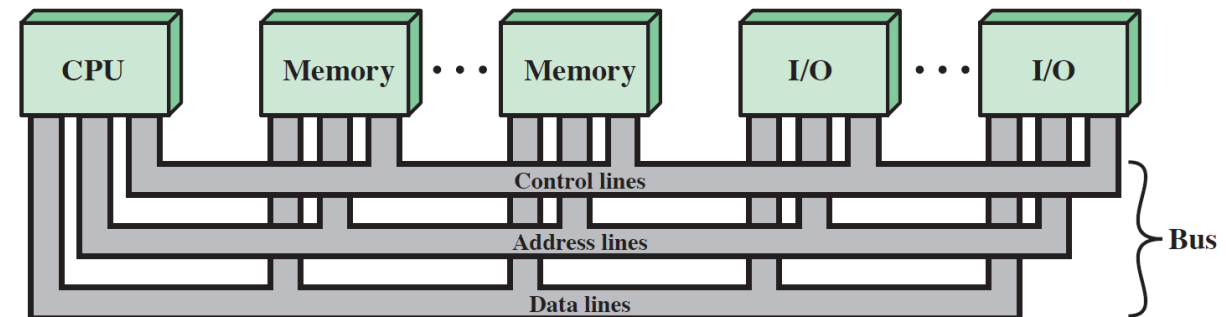
- Linije adresa koriste se za označavanje izvora ili odredišta podataka na magistrali podataka.
- Ako procesor treba da čita podatke iz memorije, on stavlja adresu željene reči na liniju adresa.
- Širina adresne magistrale određuje maksimalni kapacitet memorije sistema.





# CONTROL BUS – MAGISTRALA KONTROLE

- Kontrolne linije se koriste za kontrolu pristupa i korišćenja magistrale podataka i magistrale adresa.
- Kontrolni signali prenose komande i informacije o vremenu između sistemskih modula.
- Komandni signali navode operacije koje treba izvršiti:
  - Čitanje iz/pisanje u memoriju,
  - Čitanje sa/ispis na I/O komponentu,
  - Zahtev/odobrenje za upotrebu magistrale,...



# NAČIN FUNKCIONISANJA MAGISTRALE

- Ako jedan modul želi da pošalje podatke preko magistrale drugom modulu, mora da uradi dve stvari:
  - pribavi dozvolu za upotrebu magistrale, i
  - prenese podatke preko magistrale.
- Ako jedan modul treba da traži podatke iz drugog modula, on mora:
  - dobiti dozvolu za korišćenje magistrale,
  - preneti zahtev drugom modulu preko odgovarajućih linija za kontrolu i adresu,
  - čekati da drugi modul pošalje podatke.

# POZIVI POTPROGRAMA

Određivanje adrese potprograma koga treba pozvati



Određivanje i priprema argumenata potprograma koji se poziva



Čuvanje svih relevantnih parametara trenutnog stanja programa koji se izvršava



Specificiranje adrese na koju se vrši povratak iz potprograma



Izvršavanje instrukcije skoka kojom se izvršavanje prenosi na naredbu potprograma



Izvršavanje potprograma



Izvršavanje naredbe povratka na zahtevanu adresu u pozivajućem programu

# ČUVANJE TRENUTNOG STANJA PROGRAMA KOJI SE IZVRŠAVA

- Sadržaj opštih registara i PSW
- Vrednosti ovih registara se mogu čuvati na steku ili u posebnom prostoru koji definiše program koji poziva potprogram.
- Čuvanje tekućeg stanja može da se izvrši instrukcijom koja poziva potprogram:

PSW – Program Status Word  
Registar procesora koji sadrži status programa koji se trenutno izvršava: adresu naredne instrukcije, razne indikatore, uslovne kodove, itd.

1. Instrukcija čuva trenutno stanje pozivajućeg programa i upisuje novu vrednost u PSW, ubacujući u brojač instrukcija adresu naredbe pozvanog potprograma.
2. Kada se izvrši naredba povratka iz potprograma, stare vrednosti PSW-a i opštih registara se restauriraju i u brojač instrukcija se upisuje adresa povratka, tj. adresa naredne instrukcije programa koju treba izvršiti.

# POVRATAK NA ZAHTEVANU ADRESU U POZIVAJUĆEM PROGRAMU

- Adresa povratka može da se čuva:
  - u nekom od opštih registara,
  - na početku samog potprograma,
  - na steku.
- Ako se čuva u nekom od opštih registara, tada se
  - Taj registar ne sme da se koristi u potprogramu,
  - Svi potprogrami koriste taj registar za adresu povratka.

# ODREĐIVANJE I PRIPREMA ARGUMENATA POTPROGRAMA KOJI SE POZIVA

Argumenti i rezultati potprograma se mogu preneti u opštem slučaju na tri načina:

- Preko opštih registara.
  - Ovaj način je pogodan samo ako je broj argumenata i rezultata relativno mali.
- Upisivanjem u određenu zonu memorije čija se adresa prenosi preko opštih registara.
  - Ovaj način je pogodan za veći broj argumenata
- Putem steka, zajedno sa adresom povratka.

# MAŠINSKI JEZIK

- Mašinski jezik je skup instrukcija procesora koje poseduju semantičku interpretaciju u hardveru.
- Sve mašinske instrukcije i njihovi operandi predstavljaju nizove binarnih cifara.
- Adrese koje se javljaju u programima su apsolutne.
- Program može da radi korektno, samo ako je u memoriju smešten upravo od adrese navedene u programu.
- Programer mora da zna sve binarne ekvivalente mašinskih instrukcija, i da korektno prevede dekadne vrednosti brojeva u binarni sistem

# MAŠINSKI JEZIK

Primer:

- $A=B+C*D$
- Program počinje na adresi 51 heksadekadno, a prostor na kome su smeštene promenljive počinje na adresi 80.
- Svaka promenljiva zauzima 2 bajta.
- $A=0$ ,  $B=10$ ,  $C=20$ ,  $D=30$  dekadno
- Adresibilna jedinica je bajt.
- Dužina instrukcije dva bajta:
  - Prvi polubajt je op kod
  - Drugi polubajt je oznaka registra
  - Preostali bajt je adresa.

- Prenesi sadržaj sa lokacije 84 u opšti registar 1. (LOAD =1, ADD=2, MUL=3, STORE=4)
- Pomnoži sadržaj lokacije 86 sa sadržajem opšteg registra 1.
- Saberi sadržaj lokacije 82 sa sadržajem opšteg registra 1.
- Dobijeni zbir upiši na lokaciju 80.

Adresa	Sadržaj (binarno)			
51	0001	0001	1000	0100
53	0011	0001	1000	0110
55	0010	0001	1000	0010
57	0100	0001	1000	0000
80	0000	0000	0000	0000
82	0000	0000	0000	1010
84	0000	0000	0001	0100
86	0000	0000	0001	1110



# ASEMBLERSKI JEZIK

- Umesto binarnih kodova, za označavanje operacija se koriste mnemonici.
- Koriste se relativne adrese.
- Sva mesta na koja se program referiše (grananja i skokovi) označavaju se labelama.
- U svim naredbama u kojima se referiše memorija korišćene su simboličke umesto apsolutnih adresa.
- Povećana produktivnost programera i veća čitljivost programa.
- Asembler - program koji vrši prevođenje sa asemblerskog na mašinski jezik, i povezuje simboličke sa pravim adresama.

Primer:

- Prenesi sadržaj sa lokacije 84 u opšti registar 1.
- Pomnoži sadržaj lokacije 86 sa sadržajem opšteg registra 1.
- Saberi sadržaj lokacije 82 sa sadržajem opšteg registra 1.
- Dobijeni zbir upiši na lokaciju 80.

Labele	Simb. kod	Operandi
START	LOAD	1, C
	MUL	1, D
	ADD	1, B
	STORE	1, A
A	INT	0
B	INT	10
C	INT	20
D	INT	30

# MAŠINSKI I ASEMBLERSKI JEZICI

## Mašinski jezik

- Mašinski jezik je skup instrukcija procesora koje poseduju semantičku interpretaciju u hardveru.
- Sve mašinske instrukcije i njihovi operandi predstavljaju nizove binarnih cifara.
- Adrese koje se javljaju u programima su apsolutne.
- Program može da radi korektno, samo ako je u memoriju smešten upravo od adrese navedene u programu.

## Asemblerski jezik

- Umesto binarnih kodova, za označavanje operacija se koriste mnemonici.
- Koriste se relativne adrese.
- Sva mesta na koja se program referiše (grananja i skokovi) označavaju se labelama.
- U svim naredbama u kojima se referiše memorija korišćene su simboličke umesto apsolutnih adresa.
- Povećana produktivnost programera i veća čitljivost programa.
- Asembler - program koji vrši prevođenje sa asemblerskog na mašinski jezik

- **Mašinski jezik**

Binarno predstavljanje računarskog programa koji računar zapravo čita i tumači. Program u mašinskom kodu sastoji se od niza mašinskih instrukcija (eventualno razdvojenih sa podacima).

- **Skup instrukcija (Instruction Set)**

Skup svih mogućih uputstava za određeni računar; to jest, skup instrukcija mašinskog jezika koje određeni procesor razume.

- **Asemblerski jezik**

Simbolička reprezentacija mašinskog jezika za određeni procesor, dopunjena dodatnim iskazima koji olakšavaju pisanje programa i koje pružaju uputstva assembleru.

- **Assembler**

Program koji prevodi assemblerski jezik u mašinski kod.

# SADRŽAJ

- Deo 1. Memorija
- Deo 2. Softver - uloga i kategorizacija
- Deo 3. Rasterska i vektorska grafika

# MEMORIJA

*Memorija je svaki uređaj koji je u stanju da podatke sačuva u formatu koji neka mašina može da prepozna*

*James A. Sean*

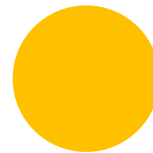
- Među delovima računarskog sistema, memorija računara ispoljava najširi spektar tipova, tehnologija koje se koriste za njenu izradu, načina organizovanja, performansi i cena.
- Ne postoji jedna vrsta memorije koja optimalno zadovoljava sve memorijske zahteve računarskog sistema. Kao posledica toga, tipičan računarski sistem je opremljen hijerarhijom memorijskih podsistema, od kojih su neki unutrašnji (direktno dostupni procesoru), a neki spoljni (dostupni procesoru preko U/I kontrolera).

Podela memorije se može izvršiti na razne načine, zavisno od karakteristike memorije u odnosu na koju se podela vrši



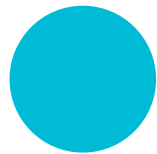
### Lokacija

Interna  
Eksterna



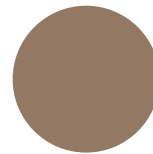
### Stalnost zapisa

Privremena/trajna  
Izbrisiva/neizbrisiva



### Kapacitet

Broj reči  
Broj bajtova



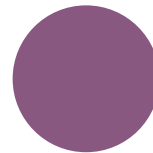
### Fizički tip

Poluprovodničke,  
Optičke,  
Magnetne,  
Magnetno-optičke



### Način pristupa

Sekvencijalni  
Direktni  
Slučajni  
Asocijativni



### Jedinica prenosa

Reč  
Blok

# LOKACIJA MEMORIJE

U odnosu na centralni procesor

- **Unutrašnje memorije**  
smeštene unutar samog procesora ili u njegovoj neposrednoj blizini sa kojim čine centralnu procesorsku jedinicu
- **Spoljašnje memorije**  
locirane dalje od procesora sa kojim komuniciraju preko U/I kontrolera i koje pripadaju kategoriji perifernih uređaja

# KAPACITET MEMORIJE

- **Reč**
  - “prirodna” jedinica organizacije memorije.
  - dužina reči = broj bitova koji se koriste za predstavljanje celog broja, ili dužina instrukcije koja se obrađuje u procesoru.
- **Adresibilna jedinica** je reč, dok neki sistemi dozvoljavaju adresiranje na nivou bajta.
  - $2^m$  adresibilnih jedinica se može adresirati sa svim adresama od m bitova.



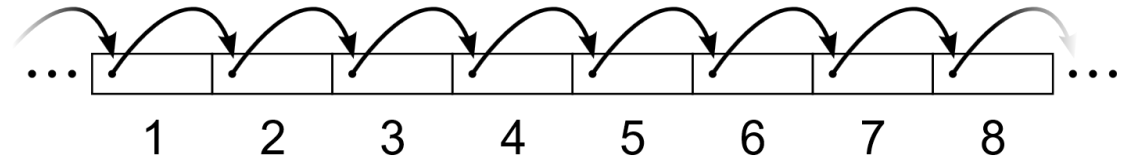
# JEDINICA PRENOSA

- **Unutrašnja memorija**
  - Broj bitova koji su pročitani iz ili su upisani u memoriju u datom trenutku. Jedinica transfera ne mora biti jednaka jednoj reči ili adresibilnoj jedinici.
- **Spoljašnja memorija**
  - Podaci se često prebacuju u mnogo većim jedinicama od reči, a one se nazivaju blokovi.

# NAČIN PRISTUPA

- **Sekvencijalni**

- podaci organizovani u slogove
- jedan po jedan slog
- vreme pristupa dosta varira
- Magnetna traka



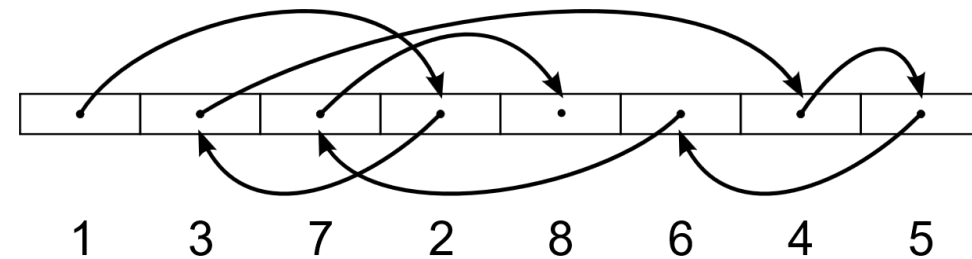
- **Direktan**

- Slogovi su organizovani u blokove
- Blok ima jedinstvenu adresu i pristupa mu se direktno.
- Slogu u okviru bloka se pristupa sekvencijalno.
- Vreme pristupa je promenljivo, ali kraće nego kod sekvencijalnog pristupa.
- Hard disk

# NAČIN PRISTUPA

- **Slučajni**

- do željenog podatka se pristupa direktno na osnovu adrese u memoriji na kojoj je podatak zapisan.
- termin slučajni (eng. *random*) znači da vreme pristupa ne zavisi od prethodnih pristupa



- **Asocijativni**

- podacima se pristupa ne na osnovu adrese nego na osnovu njihovog sadržaja.
- vrsta memorije sa slučajnim pristupom.

# PERFORMANCE

- **Vreme pristupa (*latency*)**- interval koji započinje činom iniciranja komunikacije sa memorijom, a završava se pronalaženjem podatka u okviru memorijske lokacije
- **Vreme memorijskog ciklusa** - zbir vremena pristupa i dodatnog vremena potrebnog za ponovni pristup memoriji.
- **Brzina prenosa** - količina podataka koji se mogu pročitati ili upisati u memoriju u jedinici vremena.

# FIZIČKI TIP MEMORIJE

- Poluprovodničke memorije
  - RAM
- Memorije sa magnetnom površinom
  - Disk i traka
- Optičke memorije
  - CD, DVD

# FIZIČKE KARAKTERISTIKE

- Stalnost zapisa.
  - Privremena memorija - podaci se gube kada je električna energija isključena.
  - Trajna memorija - podaci koji su jednom sačuvani, ostaju nepromenjeni sve dok ih neko namerno ne promeni
  - Magnetne memorije su trajne. Poluprovodničke memorije mogu biti privremene ili trajne.
- Izbrisivost
  - Neizbrisiva memorija se ne može menjati, osim uništavanjem.
  - Poluprovodnička memorija ovog tipa je poznata kao memorija samo za čitanje (ROM).

# OGRANIČENJA DIZAJNA RAČUNARSKE MEMORIJE

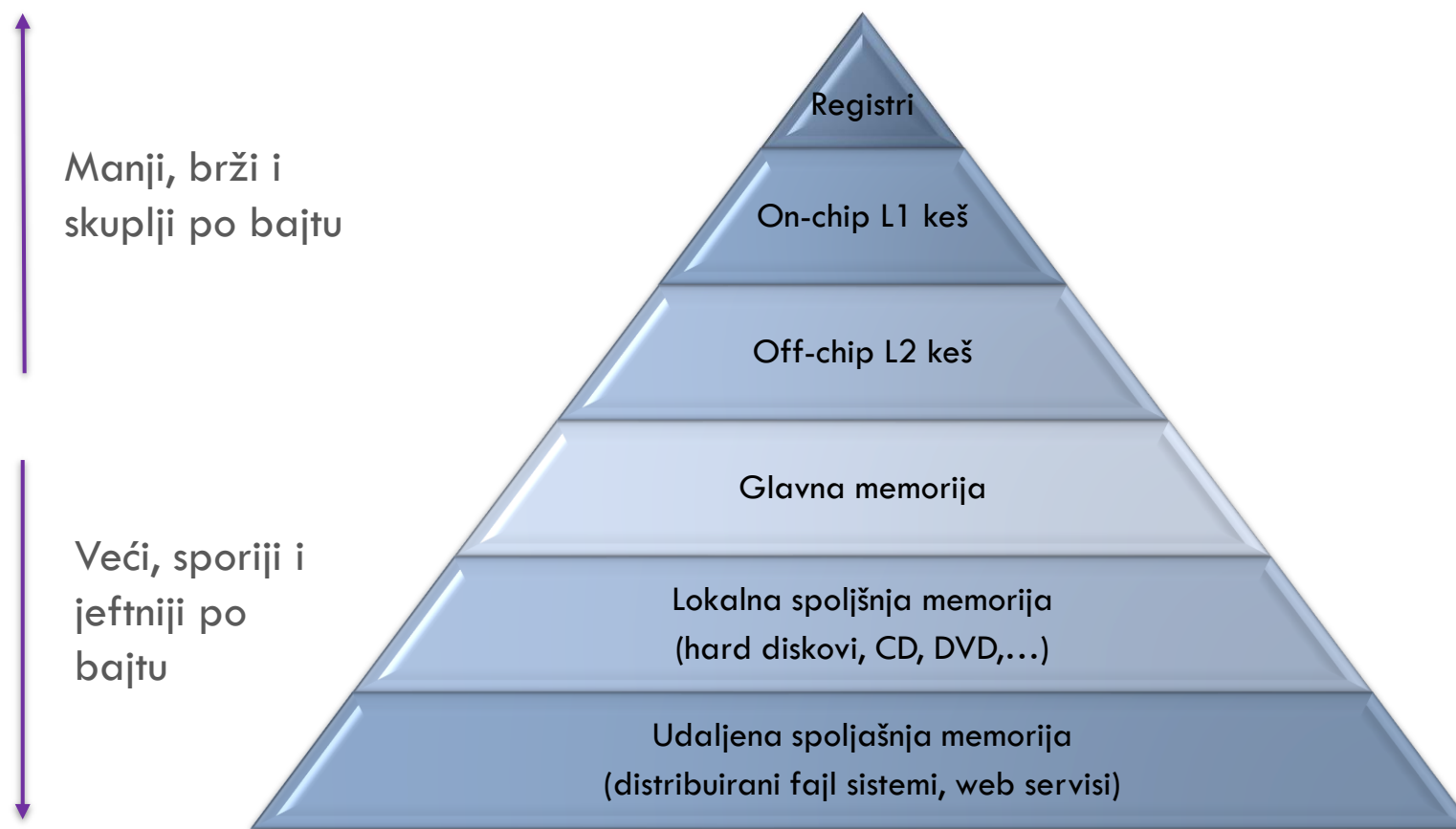
- Kog je kapaciteta?
- Koliko je brza?
- Koliko košta?

# OGRANIČENJA DIZAJNA RAČUNARSKE MEMORIJE

- Kog je kapaciteta?
  - Koliko je brza?
  - Koliko košta?
- 
- Što je brži pristup, veća je cena po bitu
  - Što je veći kapacitet, niža je cena po bitu
  - Što je veći kapacitet, sporiji je pristup



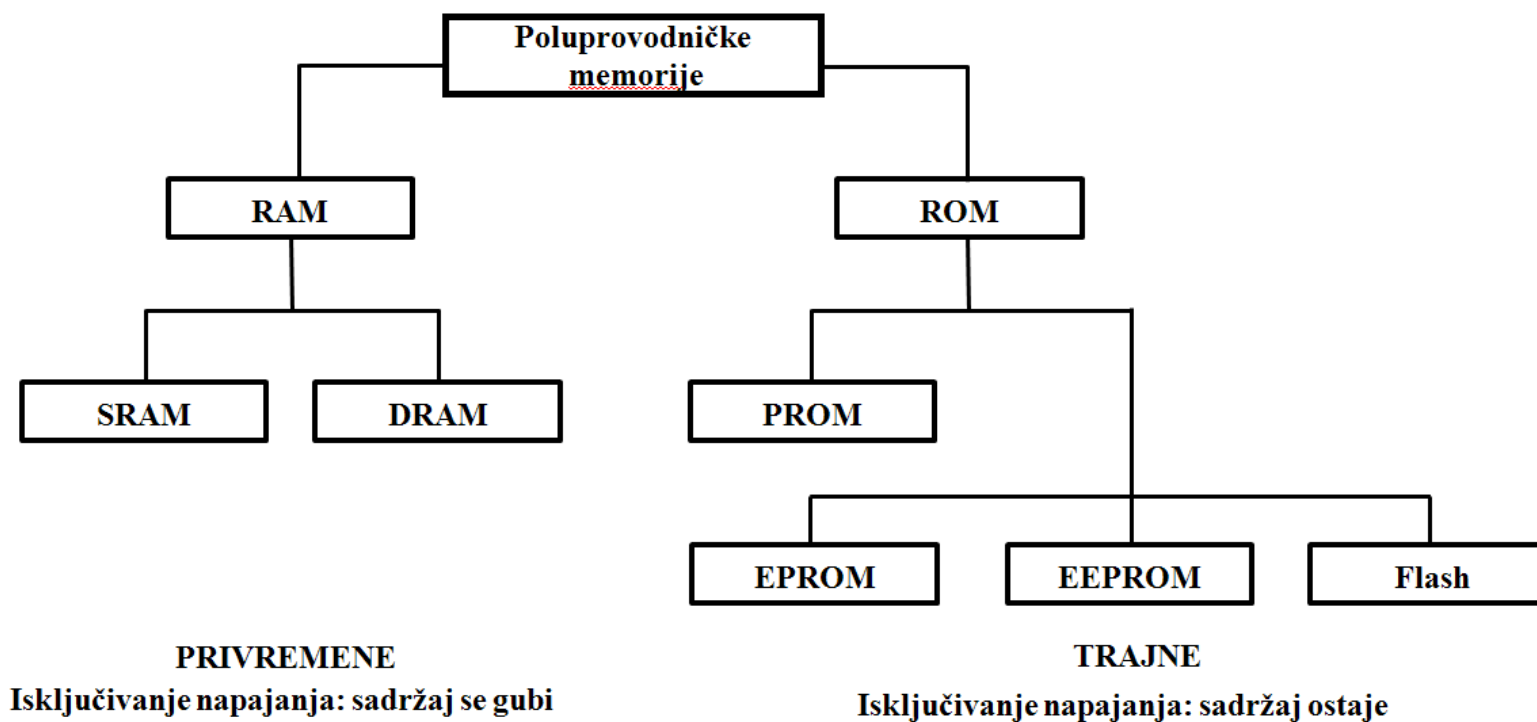
# HIJERARHIJA MEMORIJA



# POLUPROVODNIČKA UNUTRAŠNJA MEMORIJA

- Osnovna komponenta poluprovodničke memorije je memorijska ćelija.
- Ćelije mogu imati jedno od dva stabilna (ili polustabilna) – 0 i 1.
- U njih se može upisivati (makar jednom).
- Njihovo stanje se može očitati.

# POLUPROVODNIČKE MEMORIJE



# TIPOVI POLUPROVODNIČKE UNUTRAŠNJA MEMORIJA

Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Electrically	Volatile
Read-only memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile
Programmable ROM (PROM)			Electrically	
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level		
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		

# RAM

- Pojedinačnim rečima u memoriji se pristupa direktno.
- Čitanje je i upisivanje podataka je jednostavno i brzo i obavlja se korišćenjem električnih signala.
- RAM je nepostojana memorija.
- Dve vrste: DRAM i SRAM

# SRAM vs DRAM

## DRAM

- Bitovi se čuvaju u vidu električnog naboja kondenzatora
- Naboj se tokom vremena smanjuje
- Potrebno je osvežavanje
- Jednostavna konstrukcija
- Manja po bitu
- Jeftinija
- Sporija
- Koristi se u glavnoj memoriji
- U biti je analogna
  - Nivo naboja određuje vrednost

## SRAM

- Bitovi se čuvaju kao on/off prekidači
- Nema gubljenja naboja
- Nije potrebno osvežavanje
- Složenija konstrukcija
- Veće po bitu
- Skuplje
- Brže
- Koriste se za keš
- Digitalna
  - Koristi flip-flopove

# ROM — READ ONLY MEMORY

- ROM je stalna, neizmenjiva memorija računara.
- Sadržaj ROM memorije je “hard-wired” u strujnim kolima i ostaje prisutan i nakon gašenja računara.
- Tehnologija za izradu ROM ne dozvoljava ikakvu modifikaciju njenog sadržaja do fizičke zamene čipa.
- Utiskivanje podataka je relativno skupo.
- Nema mesta za greške.
- Koristi se za čuvanje BIOS-a.

## TERMINOLOGY NOTE

A ROM or EEPROM chip along with its instructions is commonly referred to as firmware; a sort of melding of hardware and software.

# KEŠ

Keš (*Cache*) - Interna memorija procesora koja premošćava veliku razliku u brzini prenosa podataka između spregnutih komponenti.

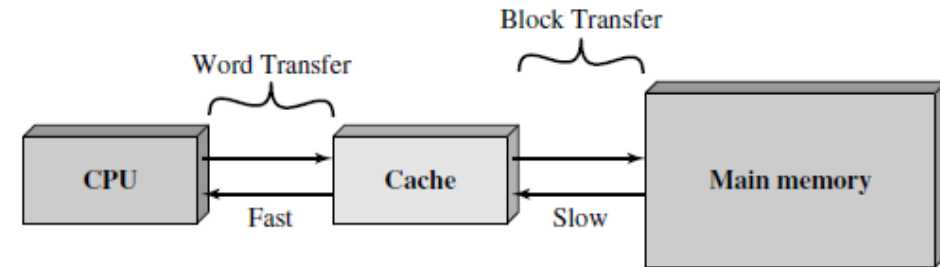
Keš sadrži kopije delova glavne memorije

Nivoi keša

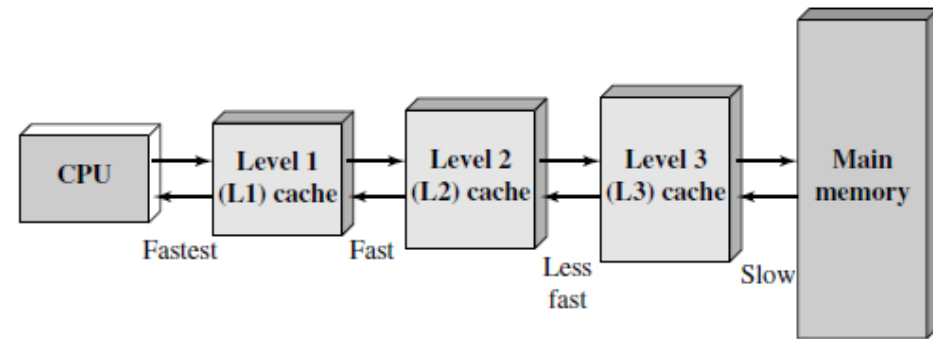
L1 - Unutar CPU čipa

L2 - Van CPU, ali u paketu s njim

L3 - Ponekad na matičnoj ploči



(a) Single cache



(b) Three-level cache organization



# ORGANIZACIJA KEŠ MEMORIJE

- Keš se sastoji od  $m$  blokova koji se nazivaju linije.
- Svaka linija sadrži  $K$  reči i dodatnu oznaku od nekoliko bita.
- Reč sadrži i kontrolni bit koji govori o tome da li je linija bila izmenjena od kako je poslednji put učitana u keš.
- Oznaka identifikuje blok memorije koji je trenutno smešten u liniju keša.

