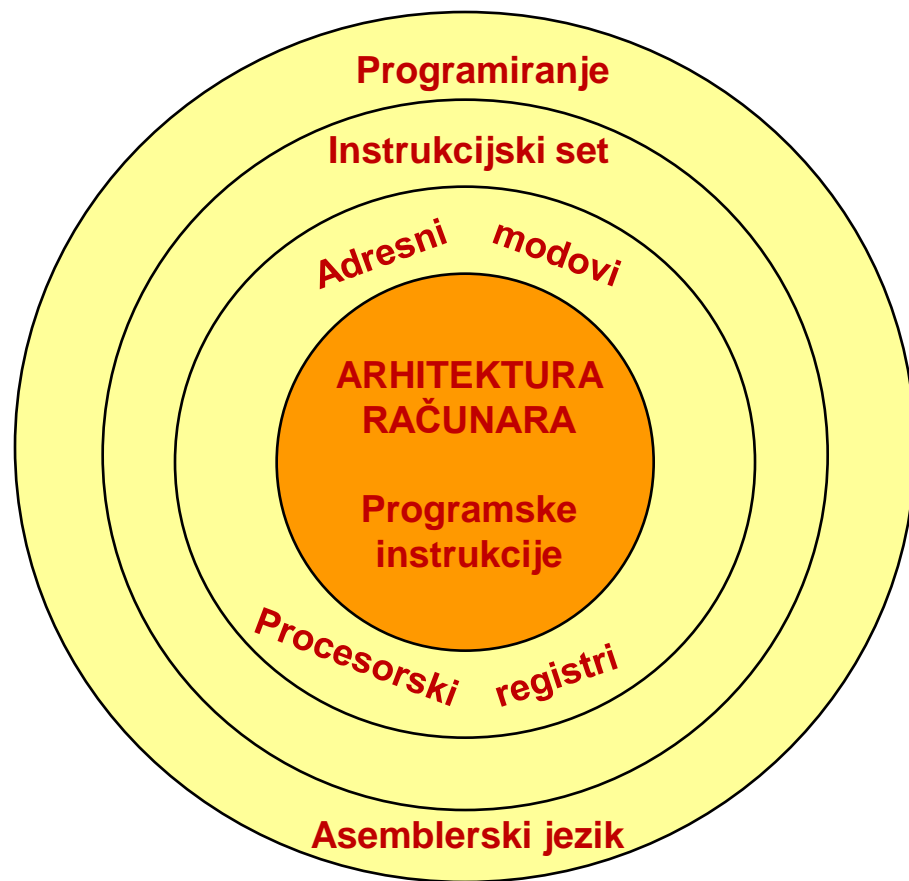


Programske instrukcije



TEME

- ✓ Osnovni pojmovi
- ✓ Formati instrukcija
 - ✓ 3-adresni format
 - ✓ 2-adresni format
 - ✓ 1-adresni format
 - ✓ 0-adresni format

Osnovni pojmovi

Arhitektura računara obuhvata sve ono što treba znati o računaru da bi za njega mogli da se pišu i na njemu izvršavaju programi. To je skup atributa sistema koji je vidljiv za programera.

Organizacija računara podrazumeva različite načine realizacije računara. To su atributi sistema transparentni za programera.

Atributi

Arhitektura računara:

- tipovi podataka
- formati instrukcija
- set instrukcija
- načini adresiranja
- programski dostupni registri
- I/O mehanizam
- mehanizam prekida...

Organizacija računara:

- kontrolni signali
- veze računara sa periferijama
- korišćene memorijske tehnologije
- tehnike realizacije upravljačkih jedinica
- tehnike realizacije keš memorija...

Tipovi podataka

- Neoznačeni celi brojevi
 - Drugi komplement
 - Pokretni zarez
 - Alfaniumerički podaci...
- ☐ Isti binarni niz može da predstavlja podatke različitih tipova.

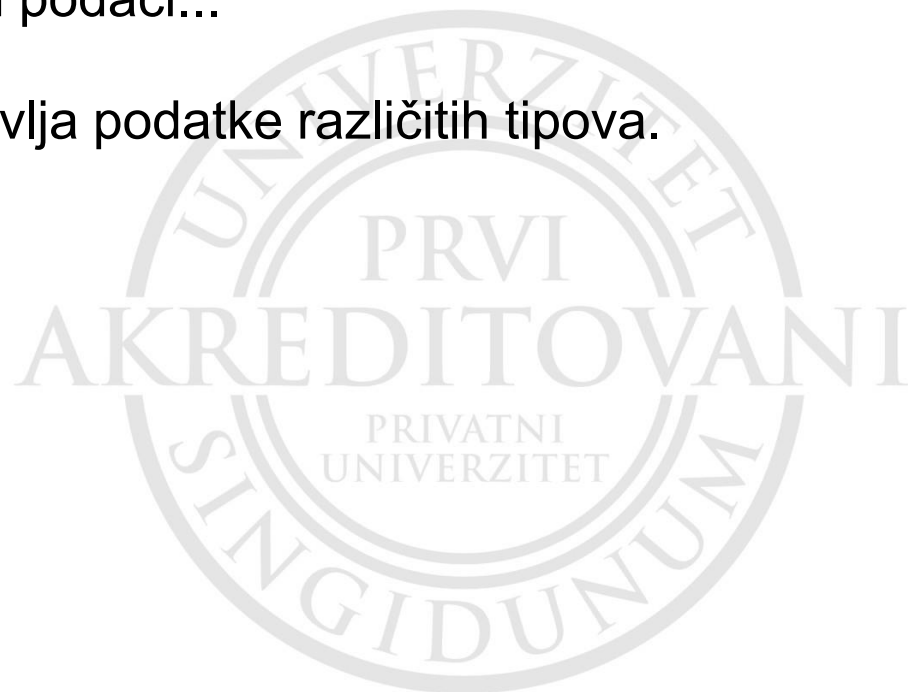
Primer 1

100011

neoznačeni ceo broj → 35

ceo broj sa znakom → -3

drugi komplement → -29



Tipovi podataka

- ❑ Podaci bilo kog tipa mogu imati različite dužine: 8, 16, 32, 64 bita.
- ❑ Ako je dužina podatka veća od dužine memorijske lokacije, podatak se smešta u više sukcesivnih lokacija.
Kao adresa podatka koristi se adresa najniže lokacije.

Podatak

Bajt4	Bajt3	Bajt2	Bajt1
-------	-------	-------	-------

Memorija (dužina lokacije 1B)

50	Bajt4	50	Bajt1
51	Bajt3	51	Bajt2
52	Bajt2	52	Bajt3
53	Bajt1	53	Bajt4

big-endian

little-endian

- ❑ Procesor mora da bude realizovan tako da kad se zada adresa 50 očita 4 bajta sa adresa 50, 51, 52 i 53.

Formati instrukcija

- ❑ Svaka instrukcija predstavlja niz bitova (0 i 1) određene dužine.
- ❑ Format instrukcije definiše značenje bitova u instrukciji.

Formatom instrukcije se specificiraju:

- operacija koju treba izvršiti
- tip podatka nad kojima se izvršava operacija
- izvorišni i odredišni operandi

Primer 2

Instrukcija: 01000001011100000000000000000000

OC	SRC1	SRC2	DST
01000001	01110000	00000000	00000000

Dužina instrukcije

- ❑ Dužina formata instrukcije zavisi od:
 - veličine memorije
 - memorijske organizacije
 - magistrale
 - složenosti i brzine procesora.
- ❑ Odluka o dužini instrukcije utiče na fleksibilnost računara sa stanovišta programera.
- ❑ Programer želi raznovrsne, moćne instrukcije da bi mu programi bili kraći.
- ❑ Pravi se *trade-off* između želje programera za moćnim instrukcijama i prostora potrebnog za njihov smeštaj.

Operacija i tip operanda

- ❑ Operacija predstavlja jednu iz skupa operacija_koji *ALU* unutar procesora može da izvrši.
- ❑ Tip operanda određuje iz koliko susednih memorijskih lokacija treba pročitati operand i kako ga interpretirati.
- ❑ Informacija o operaciji i tipu podatka specificira se poljem u formatu instrukcije koje se naziva **kod operacije**.
- ❑ Ako se ista operacija primenjuje nad različitim tipovima podataka, mora se koristiti veći broj kodova operacija.

Primer 3

Ako procesor podržava:

- neoznačene cele brojeve dužine 8 i 16 bitova i
- brojeve u pokretnom zarezu dužine 32 i 64 bita,

za operaciju množenja potrebna su 4 koda operacije.

Operandi

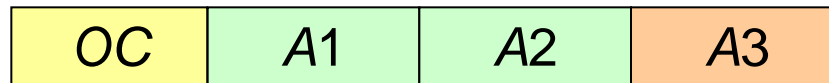
- ❑ U formatu instrukcije može da postoji 0, 1, 2 ili 3 polja kojima se specificiraju operandi.
- ❑ Prema broju polja za operande, procesori mogu biti troadresni, dvoadresni, jednoadresni i nulaadresni.
- ❑ Procesor jednog tipa ne mora da koristi samo instrukcije tog tipa, već može i instrukcije sa manje operanada.
- ❑ Procesor mora da zna kako da interpretira sva polja u formatu.

Operandi mogu biti:

- ✓ **eksplicitno definisani** (imaju adresno polje u formatu instrukcije)
- ✓ **implicitno definisani** (nemaju adresno polje u formatu instrukcije)

3-adresni format

Troadresna instrukcija:



Interpretacija formata:

- 2 adresna polja predstavljaju adrese izvorišnih operanada
- 1 adresno polje predstavlja adresu odredišnog operanda

Primer 4

Moguće su različite interpretacije:

procesor *P*: A1 i A2 su adrese izvorišnih, a A3 adresa odredišnog operanda

procesor *Q*: A1 je adresa odredišnog, a A2 i A3 adrese izvorišnih operanada.

- ✓ Dobra strana formata: jednom instrukcijom se izračunava izraz
- ✓ Loša strana formata: velika dužina instrukcije

3-adresni format

Primer 5

Procena dužine 3-adresne instrukcije:

- za adresiranje jedne lokacije operativne memorije kapaciteta nekoliko giga lokacija, potrebno je oko 30 bitova ($1\text{G} = 10^9 \approx 2^{30}$)
- tri adresna polja u formatu instrukcije zahtevaju oko 90 bitova
- pretpostavimo da je polje sa kodom operacije dužine 8 bitova
- dužina troadresnog formata instrukcije je oko 100 bitova
- ako se u memorijsku lokaciju može smestiti 1 bajt, za smeštaj jedne instrukcije u memoriji mora se utrošiti bar 12 lokacija

Problemi:

- za smeštaj programa potreban je jako veliki prostor
- mnogo vremena se gubi na obraćanja OM (12 puta), što usporava izvršavanje

2-adresni format

Dvoadresna instrukcija:

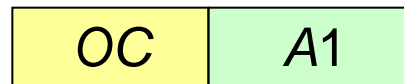
OC	A1	A2
----	----	----

Interpretacija formata:

- oba adresna polja predstavljaju adrese izvorišnih operanada
- odredišna lokacija se ne definiše eksplicitno, već se jedna od izvorišnih lokacija koristi i kao odredišna (za svaki procesor se posebno definiše koja je to lokacija)
- ✓ Dobra strana formata: instrukcija je kraća
- ✓ Loša strana formata: potrebno je više instrukcija za izračunavanje izraza

1-adresni format

Jednoadresna instrukcija:

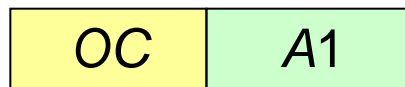


Interpretacija formata:

- adresno polje predstavlja adresu izvorišnog operanda
- u procesoru postoji poseban registar – **akumulator** koji implicitno služi za smeštaj jednog izvorišnog i jednog odredišnog operanda
- ✓ Dobra strana formata: instrukcija je kratka
- ✓ Loša strana formata: potrebno je još više instrukcija za izračunavanje izraza

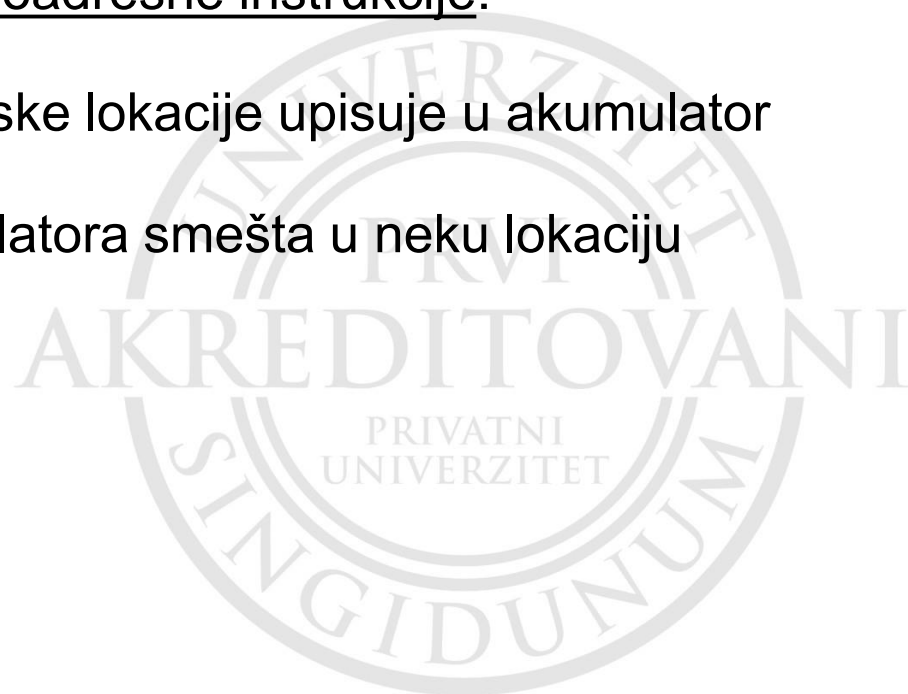
1-adresni format

Jednoadresna instrukcija:



Jednoadresni procesori imaju dve jednoadresne instrukcije:

- **LOAD** kojom se sadržaj memorijske lokacije upisuje u akumulator
- **STORE** kojom se sadržaj akumulatora smešta u neku lokaciju



0-adresni format

Nulaadresna instrukcija:

OC

Interpretacija formata:

- nema eksplicitno definisanih izvorišnih i odredišnih operanada
- kod ovih procesora, vrh **steka** je implicitno izvoršite za oba operanda i implicitno odredište za rezultat
- ✓ Dobra strana formata: najkraća instrukcija
- ✓ Loša strana formata: potrebno je najviše instrukcija za izračunavanje izraza (u odnosu na ostale formate)

0-adresni format

Nulaadresna instrukcija:

OC

OC	A1
----	----

Nulaadresni procesori imaju dve jednoadresne instrukcije (izuzeci):

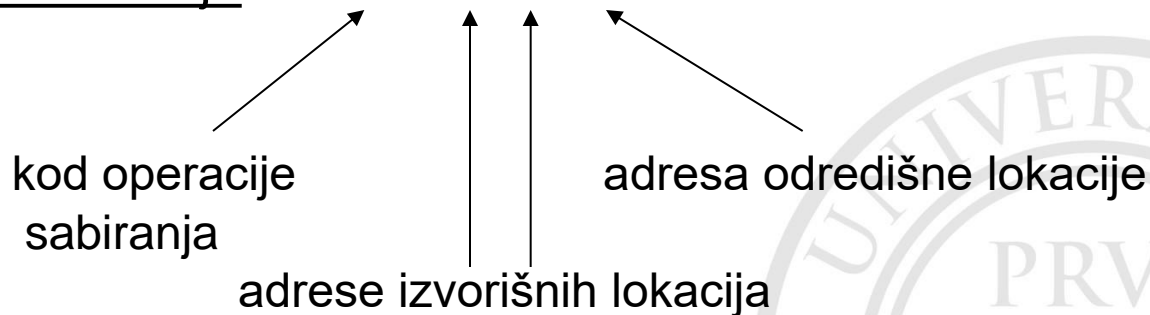
- **PUSH** kojom se sadržaj neke lokacije smešta na vrh steka
- **POP** kojom se sadržaj steka smešta u neku lokaciju

Operacija sabiranja

$$A + B \rightarrow C$$

Zadatak: izračunati dati izraz u procesoru sa 3-adresnim instrukcijama

Troadresna instrukcija: **ADD A B C**

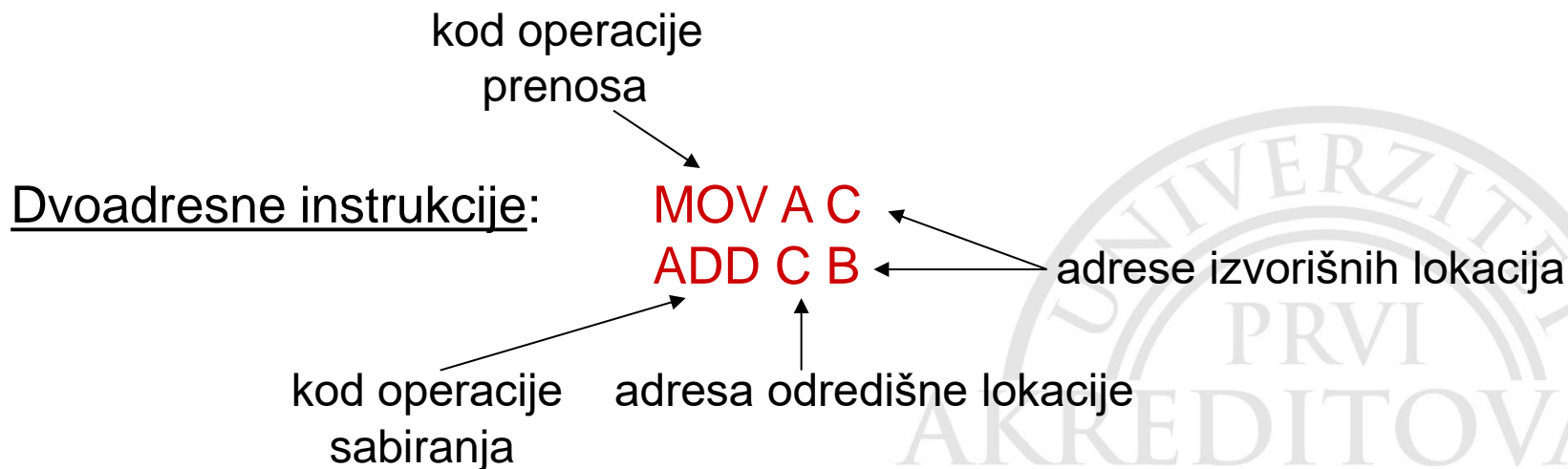


Izvršavanje instrukcije: sadržaji memorijskih lokacija čije su adrese *A* i *B* sabraju se i rezultat se smešta u memorijsku lokaciju čija je adresa *C*

Operacija sabiranja

$$A + B \rightarrow C$$

Zadatak: izračunati dati izraz u procesoru sa 2-adresnim instrukcijama



Izvršavanje instrukcije: iz lokacije sa adresom *A* čita se operand i smešta u lokaciju sa adresom *C*; zatim se iz lokacija sa adresama *C* i *B* čitaju sadržaji i sabiraju, a rezultat smešta u lokaciju sa adresom *C*

Operacija sabiranja

$$A + B \rightarrow C$$

Zadatak: izračunati dati izraz u procesoru sa 1-adresnim instrukcijama

Jednoadresne instrukcije:

OC prenosa u akumulator	→	LOAD A	←	adrese operanada
OC sabiranja	→	ADD B	←	
OC prenosa iz akumulatora	→	STORE C	←	

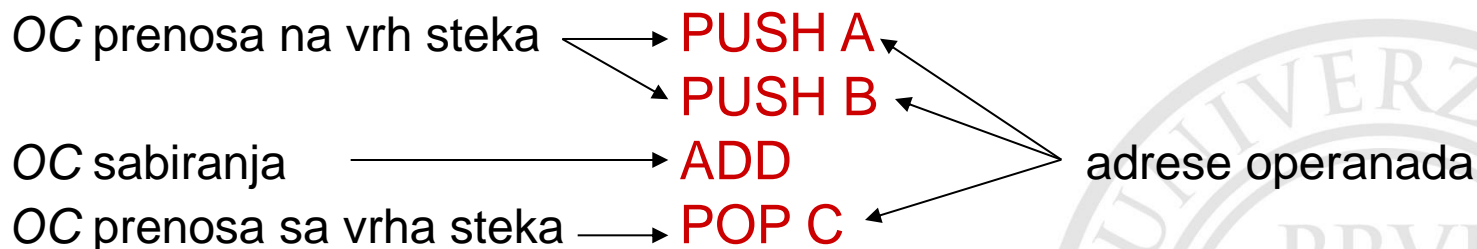
Izvršavanje instrukcije: iz lokacije sa adresom *A* čita se operand i smešta u akumulator; zatim se čita sadržaj iz lokacije sa adresom *B* i sabira sa sadržajem akumulatora, a zbir smešta u akumulator; na kraju se sadržaj akumulatora prenosi u lokaciju sa adresom *C*

Operacija sabiranja

$$A + B \rightarrow C$$

Zadatak: izračunati dati izraz u procesoru sa 0-adresnim instrukcijama

Nulaadresne instrukcije:



Izvršavanje instrukcije: iz lokacije sa adresom A čita se operand i smešta na vrh steka; iz lokacije sa adresom B čita se sadržaj i smešta na vrh steka; zatim se čitaju dva operanda sa vrha steka, sabiraju i rezultat stavlja na vrh steka; na kraju, sadržaj sa vrha steka se upisuje u lokaciju sa adresom C

Operacija sabiranja

Zaključci:

- ☐ Nakon izračunavanja izraza, sadržaji na memorijskim lokacijama sa adresama A , B i C su isti bez obzira na korišćenu varijantu procesora.
- ☐ U svim varijantama se na drugačiji način formira traženi zbir u lokaciji sa adresom C .
- ☐ Varijante imaju različite memorijske i vremenske zahteve.

