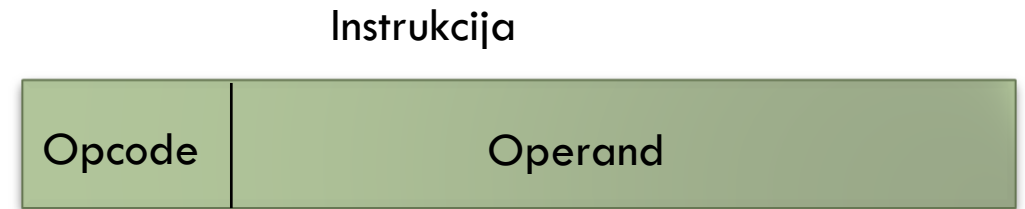


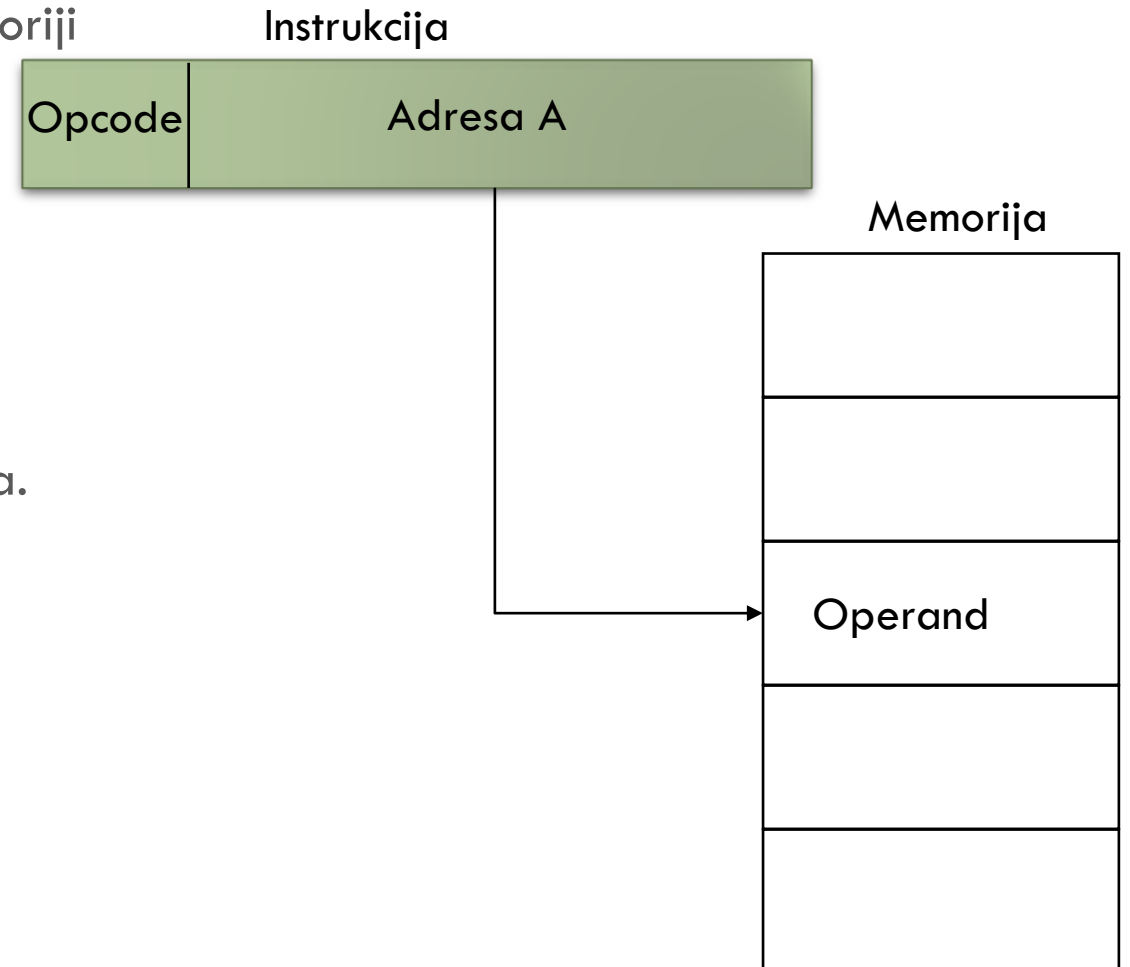
NEPOSREDNO ADRESIRANJE

- Operand je deo instrukcije
- Operand = address field
- Na primer: ADD 5
 - Dodaj 5 sadržaju akumulatora
 - 5 je operand
- Nema obraćanja memoriji za podatke
- Brzina
- Nedostatak je u tome što je veličina broja koji predstavlja operand ograničena na veličinu polja za adresu, koja je u većini instrukcija mala u poređenju sa dužinom reči.



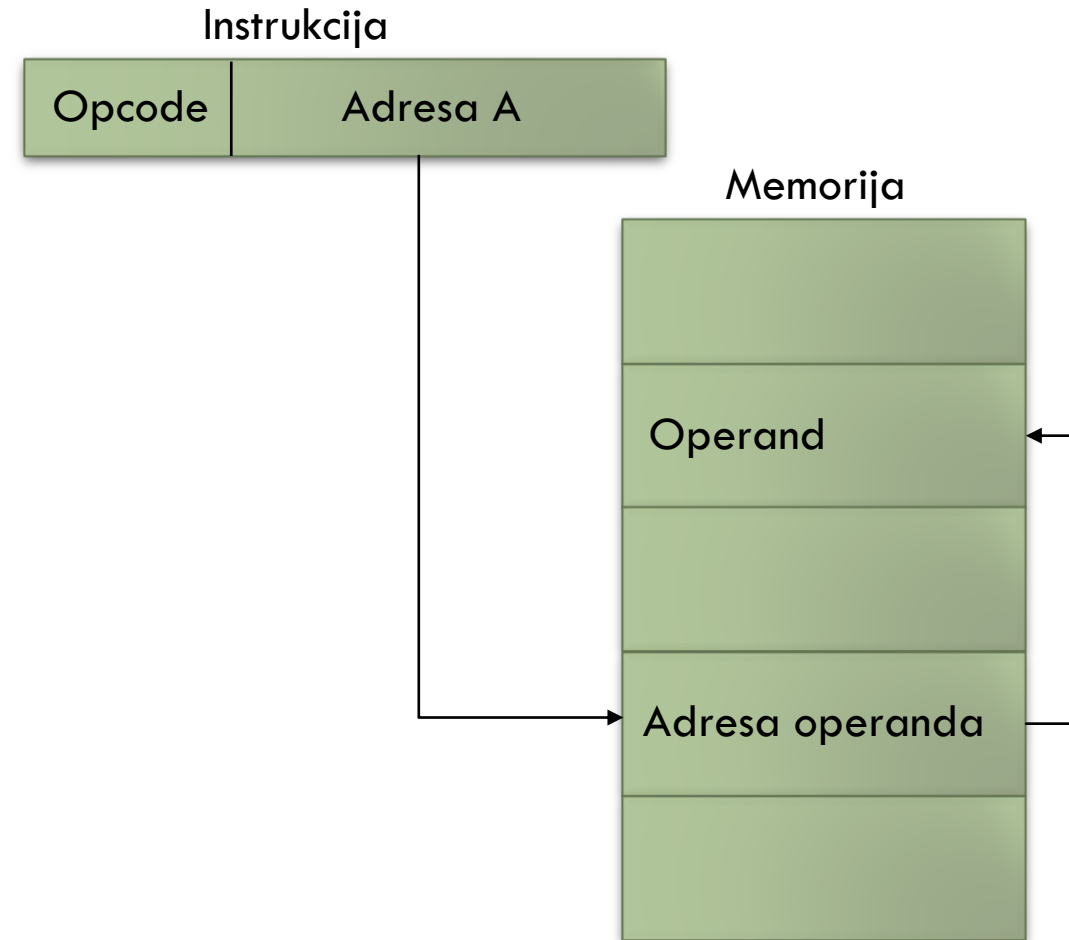
DIREKTNO ADRESIRANJE

- Polje za adresu sadrži pravu adresu operanda u memoriji
- Effective address (EA) = address field (A)
- Npr. ADD 10
 - Dodaj sadržaj memorijske ćelije sa adresom 10 sadržaju akumulatora
 - U memoriji na adresi 10 se nalazi operand
- Jedno referenciranje memorije radi pristupa podacima.
- Nema dodatnih izračunavanja efektivne adrese.
- Ograničen prostor adresa.



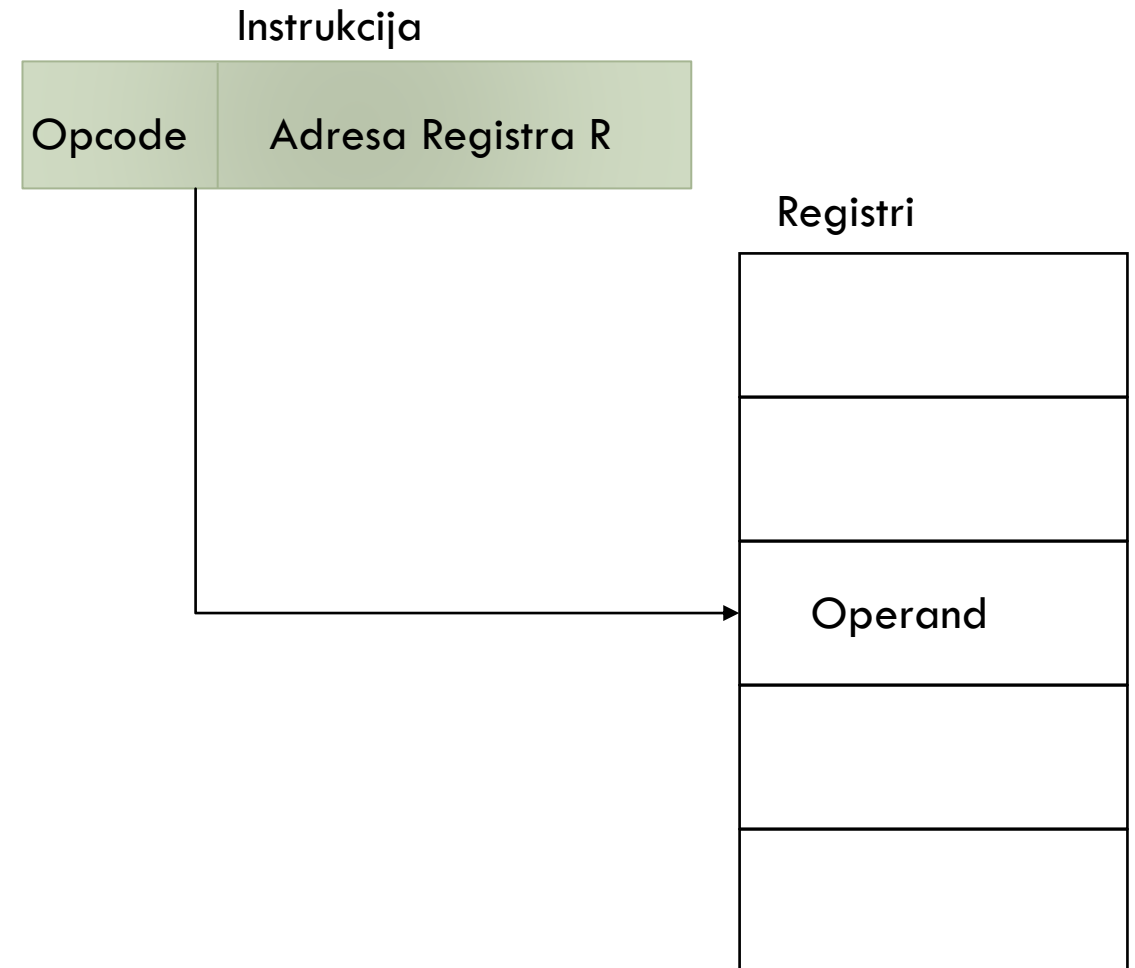
INDIREKTNO ADRESIRANJE

- Polje za adresu sadrži adresu memorijske lokacije, na kojoj se nalazi adresa operanda.
- $EA = (A)$
 - Pogledaj sadržaj na adresi A, nađi adresu (A) i sadržaj te adrese će biti pravi operand
- Na primer ADD (A)
 - Sadržaj akumulatora saberi sa sadržajem memorijske ćelije čija se adresa nalazi na lokaciji na koju upućuje sadržaj memorijske ćelije sa adrese A
- Veći adresni prostor:
 - 2^n gde je n = dužina reči
- Dvostruko referenciranje memorije radi pristupa podatku.
- Sporije

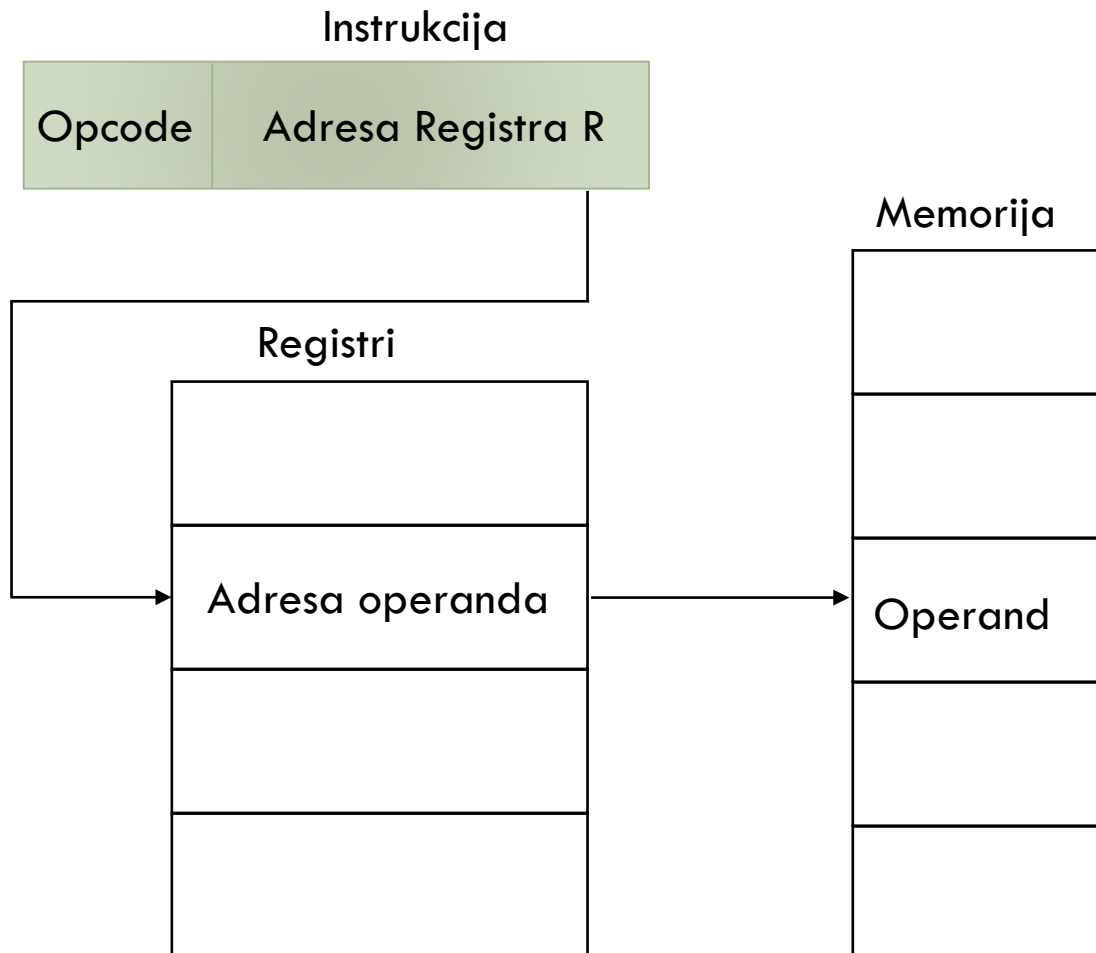


REGISTARSKO ADRESIRANJE

- Operand se nalazi u registru imenovanom u adresnom polju
- $EA = R$
- Ograničen broj registara
- Nema pristupanja memoriji, te se otuda dobija na brzini.
- Veoma ograničen adresni prostor



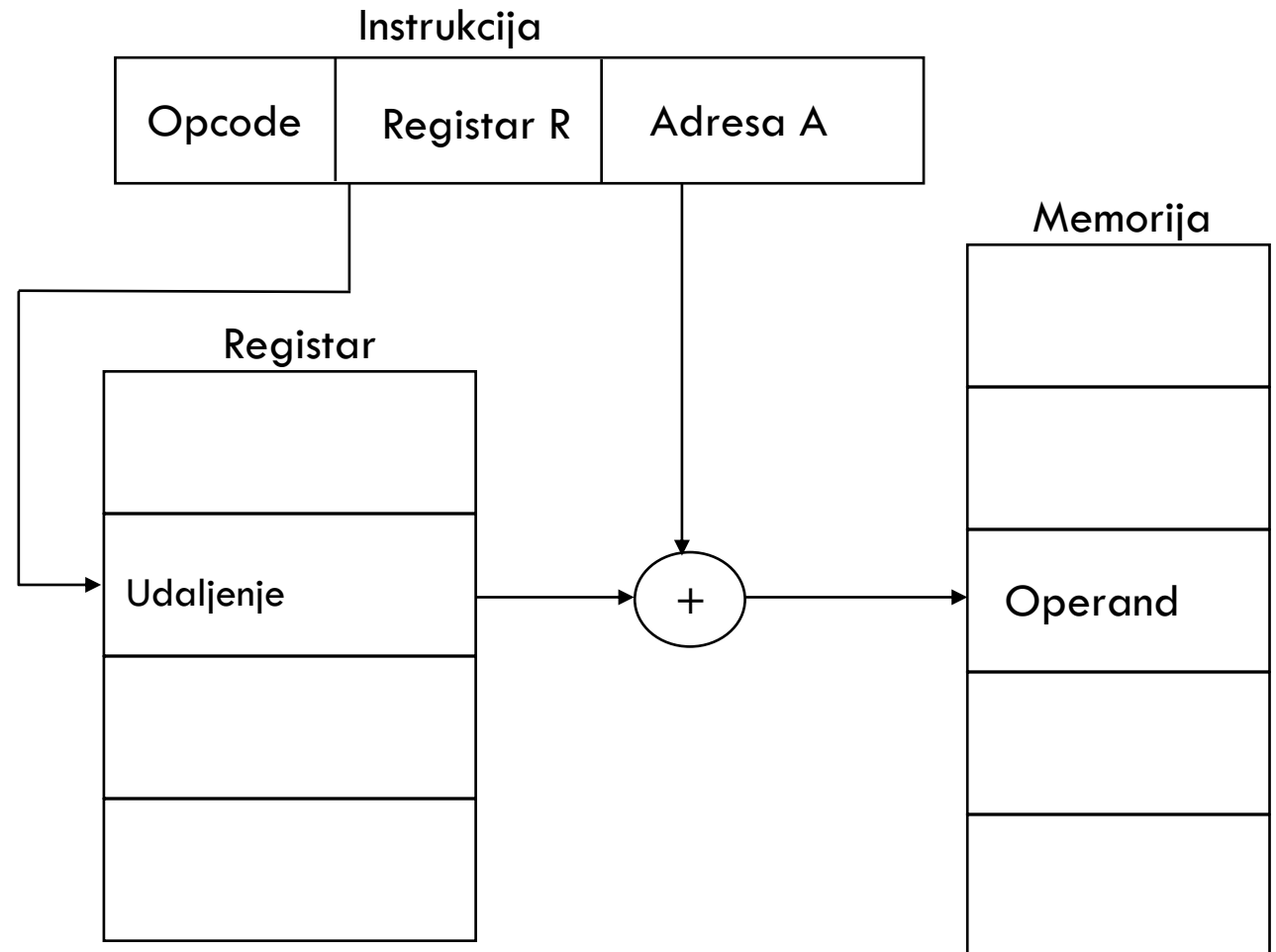
REGISTARSKO INDIREKTNO ADRESIRANJE



- $EA = (R)$
- Operand je u memorijskoj ćeliji čija se adresa nalazi u registru R.
- Veliki adresni prostor (2^n).
- Jedno pristupanje memoriji manje nego kod indirektnog adresiranja.

ADRESIRANJE SA UDALJENJEM

- Instrukcija (pored operacionog koda) sadrži reference na dva adresna polja.
- Jedno adresno polje se koristi kao kod direktnog adresiranja. Njegov sadržaj označćemo sa A.
- Drugo adresno polje sadrži oznaku registra čiji se sadržaj sabira sa A radi dobijanja stvarne adrese.
- A predstavlja udaljenje (eng. *displacement*) od početne adrese navedene u registru.
- $EA = A + (R)$



PODVRSTE ADRESIRANJA SA UDALJENJEM

- Relativno adresiranje
- Adresiranje upotrebom baznog registra
- Indeksno adresiranje

RELATIVNO ADRESIRANJE

- Jedna verzija adresiranja sa udaljenjem kod koga se jedno adresno polje implicitno odnosi na brojač instrukcija, PC
- $R = \text{Program counter, PC}$
- $EA = A + (PC)$
- Uzmi operanda iz memorijske ćelije čija je adresa za A udaljena od adrese koja se nalazi u brojaču instrukcija.

ADRESIRANJE KORIŠĆENJEM BAZNOG REGISTRA

- Referencirani registar sadrži adresu u glavnoj memoriji
- Adresno polje sadrži udaljenje od te adrese.
- Referencirani registar može biti eksplicitno naveden ili se podrazumevati.
- Može se upotrebiti kod segmentirane memorije, tako da registar sadrži bazičnu (početnu) adresu segmenata, dok nam adresno polje govori koliko treba da se udaljimo od početka segmenta da bismo došli do traženog podatka.

INDEKSNO ADRESIRANJE

- Adresno polje referencira adresu u glavnoj memoriji
- Referencirani registar sadrži pozitivno udaljenje od te adrese.
- Efikasan mehanizam za sprovođenje iterativnih operacija.

$$EA = (A) + (R)$$

$$(R) \leftarrow (R) + 1$$

NAČINI ADRESIRANJA OPERANADA U MAŠINSKIM INSTRUKCIJAMA

Neposredno	Operand se nalazi u adresnom delu instrukcije
Direktno	Stvarna adresa operanda se uključuje direktno u instrukciju
Indirektno	Instrukcija sadrži adresu lokacije u memoriji na kojoj se nalazi adresa operanda
Registarsko	Instrukcija sadrži oznaku registra koji sadrži operand
Registarsko indirektno	Instrukcija sadrži oznaku registra koji sadrži memorijsku adresu operanda
Adresiranje sa udaljenjem	Adresni deo instrukcije se dodaje sadržaju eksplicitno ili implicitno imenovanog registra da bi se dobila efektivna adresa operanda

Adresa	Memorija		
200	LOAD	Mode	PC = 200
201	Address = 500		
202	Next instruction		R1 = 400
			XR = 100
399	450		AC
400	700		
500	800		
600	900		
702	325		
800	300		

Primer: polje Mode određuje tip adresiranja, i ako je potrebno sadrži referencu na registar

Tip adresiranja	Efektivna adresa operanda	Sadržaj akumulatora
Direktno	500	800
Neposredno	201	500
Indirektno	800	300
Registarsko	/	400
Registarsko indirektno	400	700
Relativno (PC)	702	325
Sa indeks registrom	600	900

PREGLED TIPOVA ADRESIRANJA U MAŠINSKIM INSTRUKCIJAMA

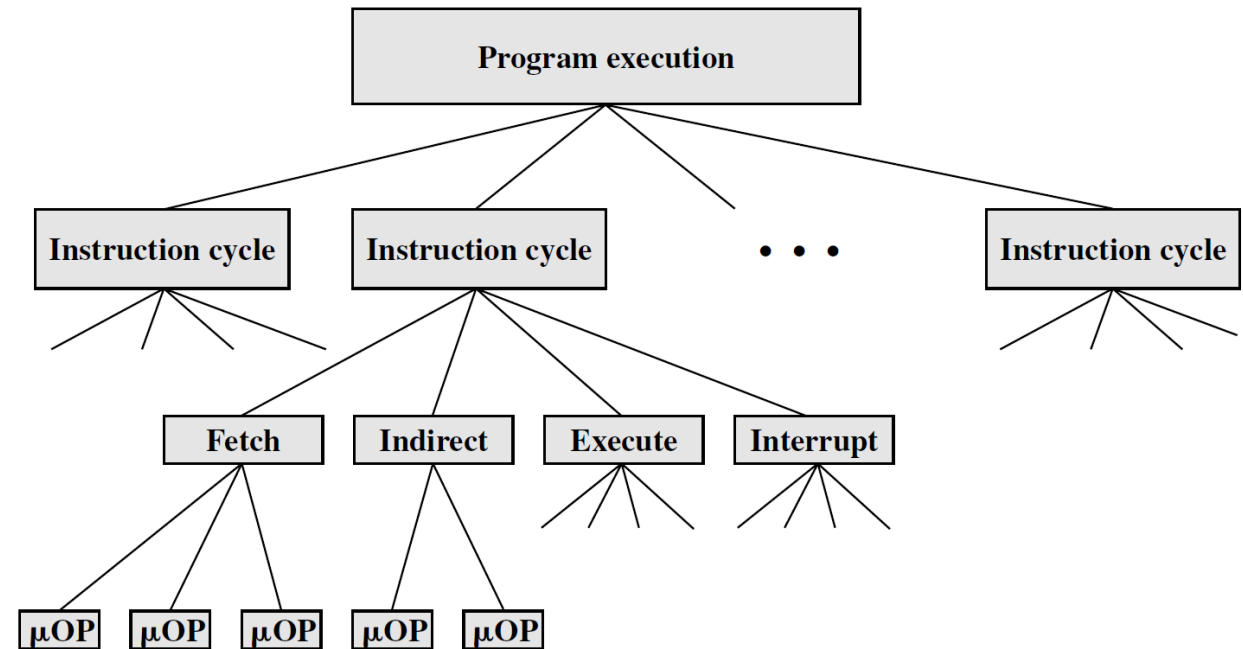
Mode	Algorithm	Principal Advantage	Principal Disadvantage
Neposredno	$\text{Operand} = A$	Nema obraćanja memoriji za operand	Ograničena veličina operanda
Direktno	$EA = A$	Jednostavno	Ograničen opseg adresa
Indirektno	$EA = (A)$	Veliki adresni prostor	Višestruko obraćanje memoriji
Registarsko	$EA = R$	Nema obraćanja memoriji za operand	Ograničen opseg adresa
Registarsko indirektno	$EA = (R)$	Veliki opseg adresa	Dodatno referenciranje na memorijsku adresu
Adresiranje sa udaljenjem	$EA = A + (R)$	Fleksibilnost	Kompleksnost

ČASOVNIK

- Sistemski časovnik reguliše sve operacije koje obavlja procesor, kao što su preuzimanje instrukcije, dekodiranje instrukcije, izvođenje aritmetičke operacije itd.
- Brzina izvršenja instrukcija je određena ciklusom časovnika - vreme koje protekne između dva impulsa (takta) časovnika.
- Što je veći broj impulsa časovnika u sekundi, to će procesor brže obrađivati podatke.
- Brzina časovnika se meri u hercima Hz, pri čemu 1 MHz označava milion taktova u sekundi, dok 1 GHz označava 10^9 taktova u sekundi.

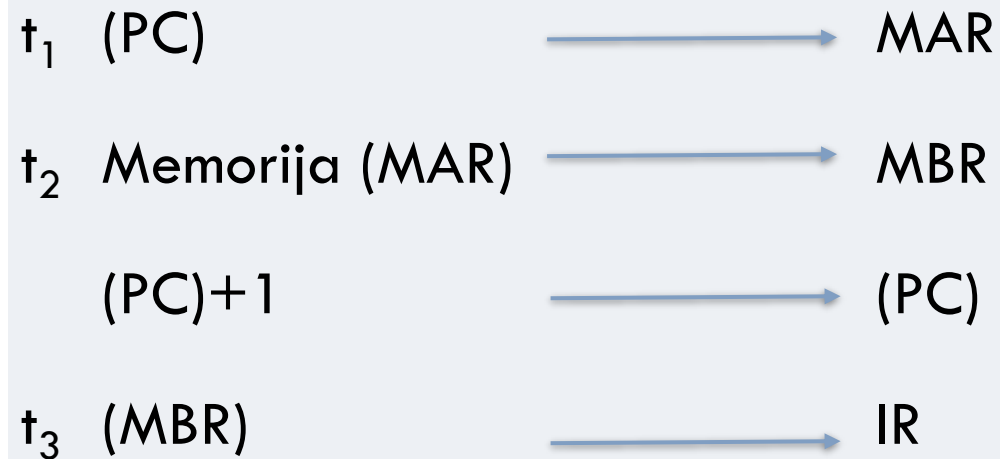
PROCES REALIZACIJE INSTRUKCIJE

- Svaka od faza instrukcionog ciklusa se može dalje razložiti na mikrooperacije.
- Mikrooperacije se prema funkciji koju obavljaju dele u sledeće kategorije:
 - Prenos podataka između registara
 - Prenos podataka iz registara ka spoljašnjem interfejsu
 - Prenos podataka od spoljašnjeg interfejsa ka registrima
 - Izvršavanje AL operacija kod kojih se U/I nalaze u registrima procesora.



PROCES REALIZACIJE INSTRUKCIJE

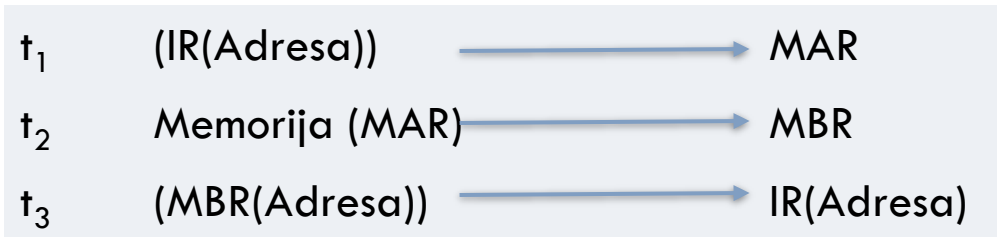
FAZA DOHVATANJA



- Jednostavni ciklus dohvatanja instrukcije se zapravo sastoji od tri koraka i četiri mikrooperacije.
- Svaka mikro operacija uključuje premeštanje podataka u ili iz registra. Sve dok se ta premeštanja ne mešaju jedna sa drugima, neka od njih mogu se odvijati tokom jednog koraka, štedeći vreme.
- Svaki takt sata definiše jedan vremenski korak za izvršavanje jedne ili više mikro-operacija.

PROCES REALIZACIJE INSTRUKCIJE

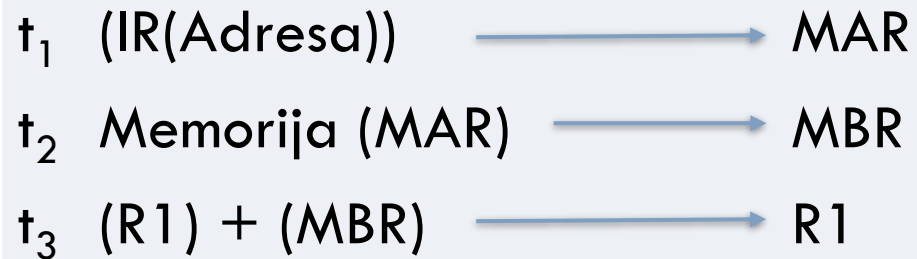
FAZA INDIREKCIJE



- Ako je instrukcija jednoadresna i koristi indirektno adresiranje, tada se ciklus indirekcije izvršava na sledeći način:
- Adresno polje instrukcije se prebacuje u MAR.
- Sa adrese koja je u MAR-u se iz memorije preuzima direktna adresa operanda i ona sada postaje adresno polje instrukcije.
- IR je sada u istom stanju u kome bi bio da je umesto indirektnog korišćeno direktno adresiranje, i spreman je za izvršenje instrukcije.

PROCES REALIZACIJE INSTRUKCIJE

FAZA IZVRŠENJA



- Primer: izvršenje instrukcije sabiranja ADD R1 L u kojoj se sadržaj opšteg registra R1 uvećava za sadržaj memorijske lokacije L:
- IR sadrži instrukciju ADD. Adresni deo instrukcije se učitava MAR.
- Operand se sa memorijske adrese koja se nalazi u MAR-u učitava u MBR.
- ALU sabira sadržaj MBR-a sa sadržajem R1 i upisuje u R1.

PROCES REALIZACIJE INSTRUKCIJE

FAZA OBRADJE PREKIDA

- Vrste prekida i načini njihove obrade zavise od konkretne mašine. Ono što se mora preduzeti je čuvanje trenutnog stanja programa koji se izvršava i upisivanje adrese rutine koja vrši obradu prekida u brojač instrukcija. Trenutno stanje obuhvata adresu instrukcije programa koji se trenutno izvršava kao i sadržaje opštih registara i PSW registra.

