MAŠINSKE NAREDBE I KARAKTERISTIKE

- Aktivnosti CPU-a odredjuju naredbe koje on izvršava.
- Naredbe zovemo mašinske naredbe.
- Skup naredbi CPU-a predstavlja ukupni broj različitih naredbi koje on može da izvrši.
- Sastavni elementi mašinske naredbe su:
  a) operacioni kod (opkod)
  b) obraćanje izvornom operandu
  c) obraćanje odredišnom operandu
  d) obraćanje narednoj naredbi.

- Izvorni i odredišni operand mogu biti smešteni u:
  - memoriji,
  - registrima CPU-a, i
  - U/I uredjajima.

Predstavljanje naredbi:

- U samom računaru svaka naredba se predstavlja kao sekvenca bitova i deli se na polja.
- Izgled (layout) naredbe zove se format naredbe.

Format naredbi

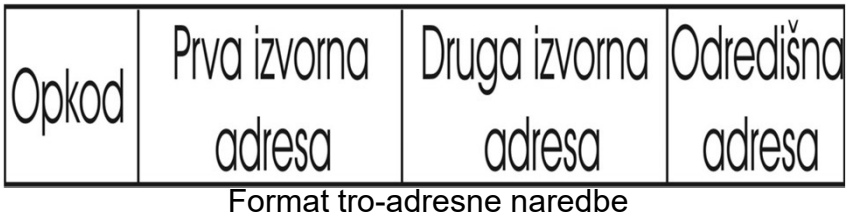
Da bi naredba sa dva izvorišna operanda i jednim odredišnim mogla da se izvrši potrebne su četiri specifikacije:

a) tip operacije (opkod)

b) adresa prvog operanda

c) adresa drugog operanda

d) adresa rezultata.

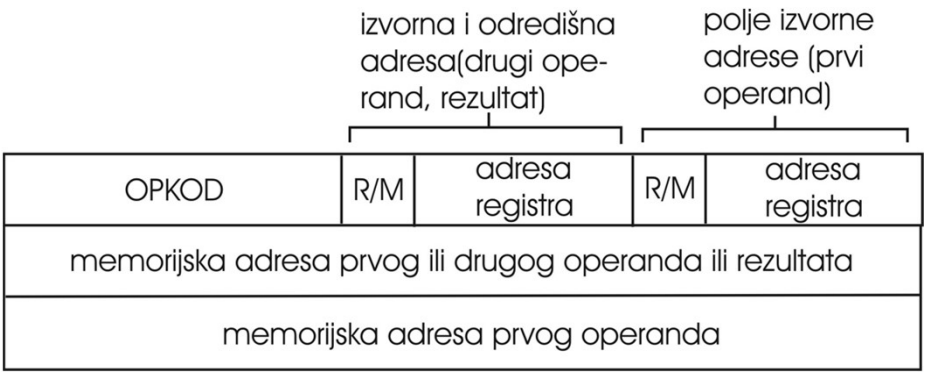


Format tro-adresne naredbe

Npr. 8 + 3 x 16 = 56

Ako procesor čita 16-bitne podatke to znači da su za čitanje ovakve naredbe neophodna četiri ciklusa

Postoji li način da se smanji obim naredbe,
ili konkretnije broj adresa?
Jedan način je da se koristi registarsko adresiranje,
A drugi da jedan operand prvo bude izvorišni, a potom odredišni.



Tipičan dvoadresni format naredbi

- Tipovi naredbi koje su relativne u odnosu na specifikaciju izvornog i odredišnog operanda:
    1. naredbe tipa registar-u-registar,
    2. naredbe tipa registar-u-memoriju,
    3. naredbe tipa memorija-u-registar
    4. naredbe tipa memorija-u-memoriju.

SUB	Y,A,B	Y ← A - B	troadresne naredbe
MUL	T,D,E	T ← D * E	
ADD	T,T,C	T ← T + C	
DIV	Y,Y,T	Y ← Y / T	
MOVE	Y,A	Y ← A	
SUB	Y,B	Y ← Y – B	
MOVE	T,D	T ← D	dvo-adresne naredbe
MUL	T,E	T ← T * E	
ADD	T,C	T ← T + C	
DIV	Y,T	Y ← Y / T	
LOAD	D	ACC ← D	
MUL	E	ACC ← ACC * E	
ADD	C	ACC ← ACC + C	
STORE	Y	Y ← ACC	
LOAD	A	ACC ← A	jedno-adresne naredbe
SUB	B	ACC ← ACC – B	
DIV	Y	ACC ← ACC / Y	
STORE	Y	Y ← ACC	

- Napomene:

  a) tro-adresni format naredbi nije pogodan jer zahteva relativno dugi format naredbi

  b) kod dvo-adresnih naredbi jedna od adresa mora biti i odredišna i izvorišna. Dvo-adresnim naredbama pojednostavljuje se format i obim naredbi, ali se uvode ograničenja

  c) kod jedno-adresnih naredbi druga adresa mora biti implicitna.

  d) postoje i nul-adresne naredbe (za manipulaciju sa magacinom)

Projektovanje skupa naredbi:

To je jedan od najinteresantnijih i najviše analizaran aspekt kod projektovanja računara.

- Najvažniji aspekti:

a) repertoar operacija – koje su to operacije, koliko ih ima i koje su složenosti

b) tipovi podataka – sa kojim tipovima podataka se operiše

c) format naredbi – odredjuje dužinu naredbi, broj adresa, obim različitih polja ...

d) registri – odredjuje broj CPU- ovih registara kojima se može pristupati od strane naredbi i način njihovog korišćenja

e) adresiranje – odredjuje način ili načine rada pomoću kojih se specificira adresa operanda.

Tipovi operanada

- **Brojevi** – kod svih mašinskih jezika operiše se sa numeričkim podacima. Kod računara se uglavnom koriste tri tipa podataka:
  - celobrojne vrednosti
  - brojevi u pokretnom zarezu
  - decimalni brojevi.
- **Znakovi** – veoma čest oblik podataka je tekstualni ili znakovni niz. Znakovi se predstavljaju kao sekvence bitova pomoću nekog koda (ASCII, EBCDIC...).
- **Logički podaci**

Tipovi operacija

Operacije možemo kategorizirati na sledeći način:

a) prenos podataka,
b) aritmetičke
c) logičke
d) konverzione
e) U/I
f) sistemsko upravljačke
g) prenos upravljanja.

Prenos podataka	Prenos podataka iz jednog registra u drugi Ako je u prenos□ulkjučena memorija: - odredjuje se adresa memorijske lok. - obavlja se virtuelno-aktuelno-memo-rijska adresna transformacija - proverava keš memoriju - inicira čitanje / upis memorije
Aritmetičke	Moguć prenos podataka pre ili posle: - obavljanja ALU funkcije - postavljanja uslovnih kodova ili markera
Logičke	Iste aktivnosti kao i aritmetičke
Konverzija	Slične su aritmetičkim i logičkim naredbama. Može da postoji specijalna logika za obavljanje konverzije

Prenos upravljanja	Ažuriraju PC. Kod naredbe za poziv i povratak iz potprograma (CALL i RET) upravljaju prenosom parametara i povezivanjem
Ulaz-izlaz	Koriste komande za U/I module. Kada je U/I memorijsko-preslikanog tipa odredjuju adresu memorijskog

Tabela: Akcije CPU-a za različite tipove operacija

- **Naredbe za prenos podataka** – najosnovniji tip mašinskih naredbi. Da bi se ove naredbe izvršile neophodno je:

  a) da se specificira lokacija izvornog i odredišnog operanda. Svaka lokacija može biti memorijska, registar ili vrh magacina,

  b) da se naznači dužina podataka sa kojim se manipuliše,

  c) da se specificira način adresiranja za svaki operand

- **Aritmetičke naredbe** – najveći broj mašina poseduje osnovne aritmetičke operacije kao što su: sabiranje, oduzimanje, množenje i deljenje.
- Često u ovu grupu spadaju i jedno-operandske naredbe:
    a) apsolutna vrednost
    b) negacija
    c) inkrementacija
    d) dekrementacija.
- Aritmetičke naredbe obavlja ALU ili specijalno projektovani hardver, dobija se rezultat i postavljaju se markeri čije se stanje u daljem programskom toku može testirati

- **Logičke naredbe** – su operacije pomoću kojih se manipuliše sa individualnim bitovima podataka, a zasnivaju se na Boolean-operacijama. Osnovne operacije iz ove grupe su NOT, OR, XOR, i AND. Tu spadaju još i naredbe za logičko i aritmetičko pomeranje ulevo i udesno, kao i rotiranje ulevo i udesno.

- **Naredbe za konverziju** – su operacije koje menjaju format ili operišu nad formatom podataka. Tipičan primer je konverzija iz decimalnog u binarni format.

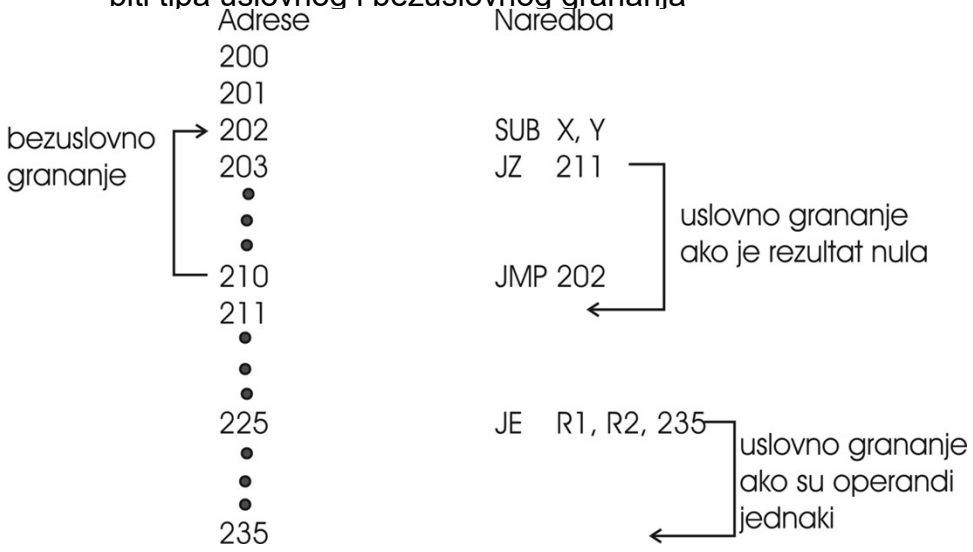
- **Ulazno/izlazne naredbe** – da bi se izvršila U/I komanda CPU se koristi adresom pomoću koje specificira odredjeni U/I modul. Postoje četiri tipa U/I komadi: upravljačka, test, čitanje i upis.

- **Naredbe sistemskog upravljanja** – u opštem slučaju to su privilegovane naredbe i mogu se izvršavati samo dok je procesor u odredjenom privilegovanom stanju, ili se izvršavaju od strane programa koji je smešten u privilegovanoj memorijskoj oblasti. Tipično, ove neredbe su rezervisane za korišćenje od strane operativnog sistema.

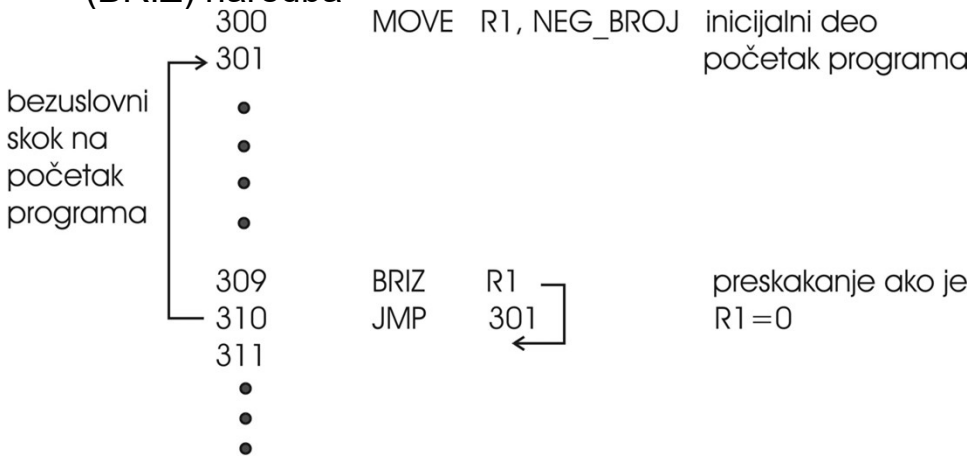
- **Naredbe za promenu toka programskog izvršenja** – CPU ažurira programski brojač i postavlja ga na vrednost adrese neke naredbe u programu. Najvažniji razlozi za promenu toka programskog izvršenja su:
  1. kad postoji potreba da se jedna ili više naredbi izvršava veći broj puta (petlja),
  2. skoro kod svih programa postoji deo gde se donosi neka odluka. U zavisnosti od rezultata testa vrši se grananje na jednu ili drugu stranu programa,
  3. programi srednjeg ili većeg obima razlažu se na manje celine koje se zatim pozivaju na izvršenje.

- Ove naredbe možemo podeliti na:
- a) naredbe grananja
- b) naredbe preskakanja
- c) naredbe za poziv potprograma.

- a) **Naredbe grananja (skoka)**
- Kod ovih naredbi jedan od operanada je adresa naredbe koja će se kao naredna izvršavati. Mogu biti tipa uslovnog i bezuslovnog grananja



- b) **Naredbe preskakanja**
- Pojam preskakanja ukazuje da se izvršenje jedne naredbe u programskom toku mora preskočiti.
- Tipičan primer je Increment- and- Skip- if- Zero (BRIZ) naredba



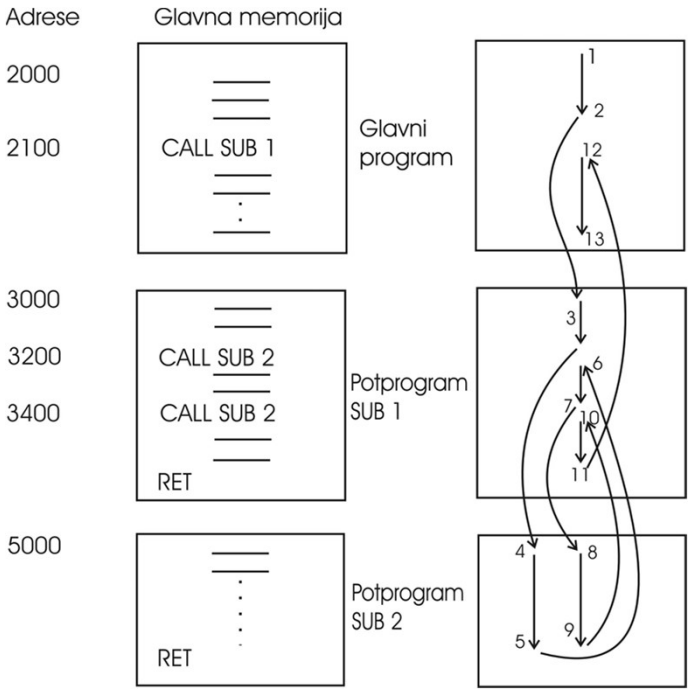
- **Naredbe za poziv potprograma**

- Uglavnom postoje dva razloga za korišćenje potprograma, a to su:
- 1.ekonomičnost – jedan isti deo programa
- može se pozivati više puta što ima za
- efekat da je program kratak (ušteda
- memorije)
- 2. modularnost – potprogramima je
- obezbedjeno da se veliki programski
- zadaci podele u manje celine, što ima za
- efekat da su programi pregledniji i da se
- lakše mogu testirati.

1. Prelazak na potprogram
2. Prenos parametara
3. Povratak u glavni program

- Mehanizam za poziv potprograma uključuje dve
- osnovne naredbe:
- 1. naredbu tipa poziv (CALL)
- 2. naredbu povratka (RET).

CPU mora u trenutku poziva da sačuva povratnu adresu da bi se nakon povratka iz potprograma produžilo sa izvršenjem sa onog mesta gde je izvršen poziv. Moguće ih je sačuvati u registrima, na početku potprograma i na vrhu magacina.



Adresni načini rada

- Termin kojim se opisuju različiti metodi koje CPU koristi za adresiranje podataka.
- Adresno polje kod tipičnog formata naredbi je veoma ograničeno, a želja programera je da se može obratiti velikom broju lokacija.
- Tipični načini adresiranja kod 16-bitnih procesora su:
- a) neposredno
- b) direktno
- c) indirektno
- d) registarsko
- e) registarsko indirektno
- f) adresiranje sa razmeštajem
- g) adresiranje preko magacina.

a) neposredno

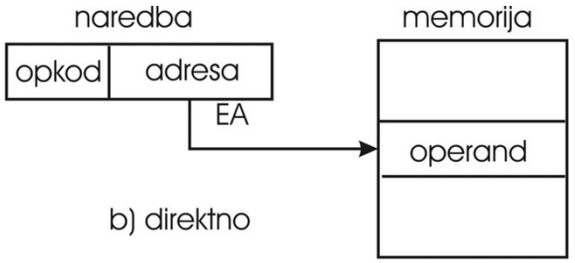
Neposredno adresiranje – operand predstavlja deo naredbe.

OPERAND = A

Koristi se kod definisanja konstanti ili za postavljanje početne vrednosti promenljivih.

Ne postoji dodatno obraćanje, sem onog koje je potrebno za pribavljanje naredbe.

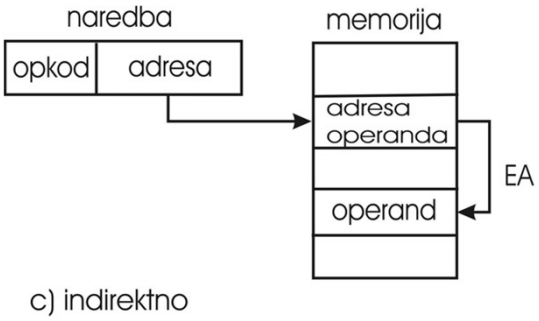
Nedostatak: obim broja ograničen je obimom adresnog polja.



Direktno adresiranje – adresno polje sadrži efektivnu adresu operanda.

$$EA = A$$

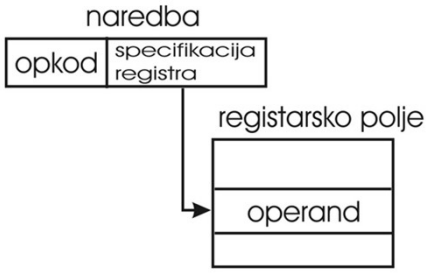
Jedno obraćanje memoriji
Ne postoji potreba za posebnim izračunavanjem adrese
Nedostatak: manipuliše se sa ograničenim adresnim prostorom



Indirektno adresiranje – naredba poseduje adresno polje kojim se može poslužiti kod obraćanja memorijskoj adresi koja sadrži adresu operanda

$$EA = (A)$$

Ako je dužina adresnog polja N može se adresirati 2^N različitih memorijskih lokacija.
Zahtevaju se dva memorijska obraćanja, prvim se pribavlja adresa, a drugim operand.

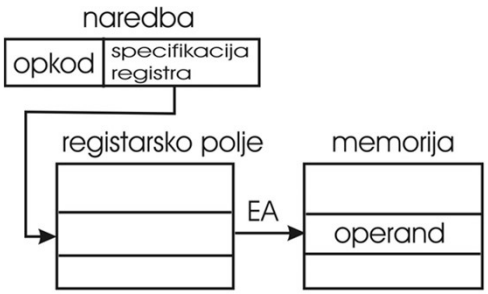


d) registarsko

Registarsko adresiranje – razlika u odnosu na direktno adresiranje je u tome što se adresnim poljem specificira registar, a ne adresa u glavnoj memoriji.

EA= R

Prednosti:
Adresno polje naredbe je malo
Nije potrebno dodatno obraćanje memoriji
Nedostatak:
Adresni prostor je ograničen brojem registara CPU-a

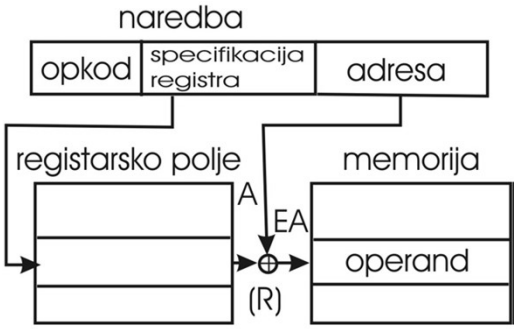


e) registarsko indirektno

Registarsko indirektno adresiranje – analogno indirektnom adresiranju sa tom razlikom što se adresno polje naredbe odnosi na specifikaciju registra, a ne specifikaciju memorijske lokacije.

EA= (R)

Prednosti i nedostaci slični kao kod indirektnog. Postoji samo jedno dodatno obraćanje memoriji.



f) adresiranje sa razmeštajem

Adresiranje sa razmeštajem – moćan način jer kombinuje mogućnosti direktnog i registarsko indirektnog adresiranja.

EA= A + (R).

Neophodno je da naredba ima dva adresna polja pri čemu je jedno eksplicitno. Vrednost koja se nalazi u jednom adresnom polju (vrednost A) se koristi direktno. Drugo adresno polje se zasniva na opkodu i odnosi se na registar čiji se sadržaj sabira sa A da bi se dobila efektivna adresa.

- Tri najčešće korišćenja načina adresiranja sa razmeštajem su:
  a) relativno,
  b) bazno registarsko
  c) indeksno adresiranje.

  a) **Relativno adresiranje**

  Registar kome se implicitno obraćamo je PC. Adresi tekuće naredbe dodaje se adresno polje i formira EA.

  Adresno polje se obično tretira kao broj u dvojnom komplemnetu.

  Na ovaj način ef. adresa je razmeštaj koji je relativan u odnosu na adresu naredbe.

  b) **Bazno registarsko adresiranje**

  Registar kome se obraćamo sadrži memorijsku adresu, a adresno polje sadrži razmeštaj (neoznačen celi broj) u odnosu na tu adresu.

c) **Indeksiranje**

Adresnim poljem se obraćamo adresi glavne memorije, a registar kome se obraćamo sadrži pozitivni razmeštaj u odnosu na tu adresu.

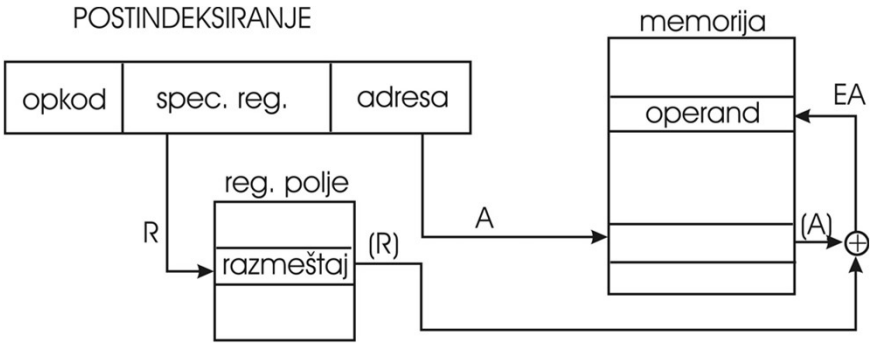
Važna karakteristika indeksiranja je ta da obezbedjuje mehanizam za obavljanje iterativnih operacija.

autoindeksiranje

EA = A + (R)

(R) ← (R) + 1

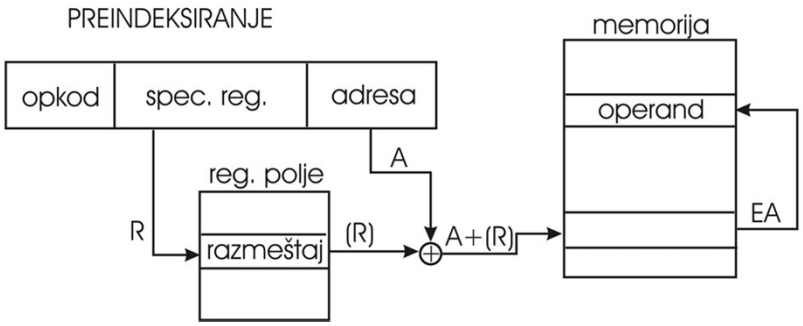
Kod indeksiranja postoje dve mogućnosti u zavisnosti od toga da li se indeksiranje obavlja pre ili posle indirekcije:



postindeksiranje

EA= (A) + (R)

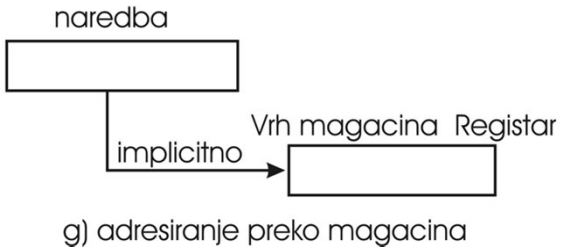
Sadržaj adresnog polja se koristi za pristup memorijskoj lokaciji koja sadrži direktnu adresu. Ova adresa se zatim indeksira od strane vrednosti registra.



Preindeksiranje

EA= (A+ (R))

Adresa se izračunava kao kod indeksiranja. U ovom slučaju izračunata adresa ne sadrži operand nego adresu operanda.



g) adresiranje preko magacina

Adresiranje magacina – spada u klasu implicitnog adresiranja. Mašinske naredbe ne sadrže polje za obraćanje memoriji, nego implicitno manipulišu sa vrhom magacina.

Način rada	Algoritam	Glavna prednost	Glavni nedostatak
neposredno adresiranje	operand=A	ne postoji obraćanje memoriji	ograničeni iznos operanda
direktno adresiranje	EA= A	jednostavnost	ograničeni adresni prostor
indirektno adresiranje	EA=(A)	veliki adresni prostor	veći broj obraćanja memoriji
registarsko adresiranje	EA=R	ne postoji obraćanje memoriji	ograničeni adresni prostor
registarsko indirektno	EA=(R)	veliki adresni prostor	dodatno obraćanje memoriji
adresiranje sa razmeštajem	EA=A+(R)	fleksibilno	kompleksno
magacin	EA=vrh magacina	ne postoji obraćanje memoriji	ograničena primena