

Neki računar ima naredbe dužine 1Bajt. Bit 7 naredbe služi za naznačavanje vrste adresiranja. Bit 6 naredbe služi za naznačavanje registara koji se u adresiranju koriste. Preostali bitovi predstavljaju kod operacije.

- a) Koliki je najveći broj različitih naredbi?
- b) Koliko načina adresiranja ovaj računar može da koristi?
- c) Koliko registara se može koristiti u adresiranju?

Neki računar ima naredbe dužine 1Bajt. Bit 7 naredbe služi za naznačavanje vrste adresiranja. Bit 6 naredbe služi za naznačavanje registara koji se u adresiranju koriste. Preostali bitovi predstavljaju kod operacije.

- a) Koliki je najveći broj različitih naredbi?
- b) Koliko načina adresiranja ovaj računar može da koristiti?
- c) Koliko registara se može koristiti u adresiranju?

7 6 5	4 3 2 1 0	Drugi bajt naredbe
način adresiranja	kod operacije	adresni deo

Neki računar ima naredbe dužine 1Bajt. Bit 7 naredbe služi za naznačavanje vrste adresiranja. Bit 6 naredbe služi za naznačavanje registara koji se u adresiranju koriste. Preostali bitovi predstavljaju kod operacije.

- a) Koliki je najveći broj različitih naredbi?
- b) Koliko načina adresiranja ovaj računar može da koristi?
- c) Koliko registara se može koristiti u adresiranju?

7 6 5	4 3 2 1 0	Drugi bajt naredbe
način adresiranja	kod operacije	adresni deo

Broj različitih tipova naredbi B_N određen je dužinom binarne reči koja predstavlja kod operacije

a) $B_N = 2^6$ različitih naredbi = 64 različite naredbe

Neki računar ima naredbe dužine 1Bajt. Bit 7 naredbe služi za naznačavanje vrste adresiranja. Bit 6 naredbe služi za naznačavanje registara koji se u adresiranju koriste. Preostali bitovi predstavljaju kod operacije.

- Koliki je najveći broj različitih naredbi?
- Koliko načina adresiranja ovaj računar može da koristi?
- Koliko registara se može koristiti u adresiranju?

7 6 5	4 3 2 1 0	Drugi bajt naredbe
način adresiranja	kod operacije	adresni deo

a) $B_N = 2^6$ različitih naredbi = 64 različite naredbe

b) $N_a = 2^1$ načina adresiranja = 2 načina adresiranja

c) Kako bit 6 naznačava regstar koji se koristi za adresovanje, moguće je ovim bitom naznačiti dva takva registra, napr.: bit 6 = 0 - koristi se registar X0, bit 6 = 1 - koristi se registar X1

Koliko bitova sadrži svaka ćelija operativne memorije ako se zna da kapacitet ove memorije iznosi 3MB, odnosno 1Mreč? Napisati najveći heksadecimalni broj koji se može upisati u ćeliju ove memorije?

Koliko bitova sadrži svaka ćelija operativne memorije ako se zna da kapacitet ove memorije iznosi 3MB, odnosno 1Mreč? Napisati najveći heksadecimalni broj koji se može upisati u ćeliju ove memorije?

$$C_M = 1\text{Mreč} = 2^{20}\text{reči}$$

$$C_M = 3\text{ MB} = 3 \times 2^{20}\text{ B} = 2^{20} \times 3\text{ B} = 2^{20}\text{ reči}$$

Koliko bitova sadrži svaka ćelija operativne memorije ako se zna da kapacitet ove memorije iznosi 3MB, odnosno 1Mreč? Napisati najveći heksadecimalni broj koji se može upisati u ćeliju ove memorije?

$$C_M = 1\text{Mreč} = 2^{20}\text{reči}$$

$$C_M = 3\text{ MB} = 3 \times 2^{20}\text{ B} = 2^{20} \times 3\text{ B} = 2^{20}\text{ reči}$$

odakle sledi da ćelije operativne memorije sadrže 3B. Najveći heksadecimalni broj koji se može upisati u ćeliju ove memorije je FFFFFFFF.

. U jednodresnom računaru koristi se memorijsko indirektno adresiranje. Dužina naredbe je 3 Bajta. Sve adrese i operandi su dužine 2 Bajta. Iz memorije se čita i u memoriju se upisuje bajt po bajt. Koliki je broj pristupa memoriji u fazi čitanja naredbe, određivanja adrese i čitanja operanda?

. U jednoadresnom računarstvu koristi se memorijsko indirektno adresiranje. Dužina naredbe je 3 Bajta. Sve adrese i operandi su dužine 2 Bajta. Iz memorije se čita i u memoriju se upisuje bajt po bajt. Koliki je broj pristupa memoriji u fazi čitanja naredbe, određivanja adrese i čitanja operanda?

Kako se iz memorije čita bajt po bajt, u fazi čitanja naredbe memoriji se pristupa 3 puta, jer je naredba dugačka 3 bajta.

. U jednodresnom računaru koristi se memorijsko indirektno adresiranje. Dužina naredbe je 3 Bajta. Sve adrese i operandi su dužine 2 Bajta. Iz memorije se čita i u memoriju se upisuje bajt po bajt. Koliki je broj pristupa memoriji u fazi čitanja naredbe, određivanja adrese i čitanja operanda?

Kako se iz memorije čita bajt po bajt, u fazi čitanja naredbe memoriji se pristupa 3 puta, jer je naredba dugačka 3 bajta.

Adresni deo naredbe sadrži dvobajtnu adresu adrese operanda, pa se memoriji pristupa 2 puta u fazi određivanja dvobajtna adrese.

. U jednodresnom računaru koristi se memorijsko indirektno adresiranje. Dužina naredbe je 3 Bajta. Sve adrese i operandi su dužine 2 Bajta. Iz memorije se čita i u memoriju se upisuje bajt po bajt. Koliki je broj pristupa memoriji u fazi čitanja naredbe, određivanja adrese i čitanja operanda?

Kako se iz memorije čita bajt po bajt, u fazi čitanja naredbe memoriji se pristupa 3 puta, jer je naredba dugačka 3 bajta.

Adresni deo naredbe sadrži dvobajtnu adresu adrese operanda, pa se memoriji pristupa 2 puta u fazi određivanja dvobajtna adrese.

Operand je dužine 2 bajta pa se u fazi čitanja operanda memoriji pristupa 2 puta

. U jednodresnom računaru koristi se memorijsko indirektno adresiranje. Dužina naredbe je 3 Bajta. Sve adrese i operandi su dužine 2 Bajta. Iz memorije se čita i u memoriju se upisuje bajt po bajt. Koliki je broj pristupa memoriji u fazi čitanja naredbe, određivanja adrese i čitanja operanda?

Kako se iz memorije čita bajt po bajt, u fazi čitanja naredbe memoriji se pristupa 3 puta, jer je naredba dugačka 3 bajta.

Adresni deo naredbe sadrži dvobajtnu adresu adrese operanda, pa se memoriji pristupa 2 puta u fazi određivanja dvobajtna adrese.

Operand je dužine 2 bajta pa se u fazi čitanja operanda memoriji pristupa 2 puta

Kod nekog jednodresnog računara prvi bajt naredbe sadrži kod operacije (bitovi od 5 do 0) i kod načina adresovanja (bitovi 7 i 6). Svi operandi su jednobajtni i prikazani su u drugom komplementu. Sve adrese su dvobajtni i smeštaju se niži bajt na nižoj, a viši bajt na višoj adresi. Stek raste od viših ka nižim adresama u memoriji, a SP ukazuje na prvu slobodnu lokaciju na steku. Sve naredbe koriste jednobajtni akumulator u kome ostaje rezultat. Sadržaj dela memorije je:

Lokacija:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Sadržaj:	0E	00	C0	5C	00	00	B8	30	FE	FF	00	FA	55	A3	AF	0E

adresiranje	kod	dužina naredbe	operacija	kod
memorijsko direktno	00	3 bajta	ADD	011100
memorijsko indirektno	01	3 bajta	SUB	111000
neposredno	10	2 bajta	PUSH	111110
bezaadresna naredba	11	1 bajt	POP	000000

a) Ako je sadržaj brojača naredbi $PC = 2$ i pokazivača steka $SP = 0E$ odrediti vrednost sadržaja akumulatora i pokazivača steka po izvršenju prve naredbe.

b) Koliki je broj pristupa memoriji u fazi čitanja naredbe, određivanja adrese, čitanja operanda i izvršenja za izvršenu naredbu.

Lokacija: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
 Sadržaj: 0E 00 C0 5C 00 00 B8 30 FE FF 00 FA 55 A3 AF 0E

adresiranje	kod	dužina naredbe	operacija	kod
memorijsko direktno	00	3 bajta	ADD	011100
memorijsko indirektno	01	3 bajta	SUB	111000
neposredno	10	2 bajta	PUSH	111110
bezaadresna naredba	11	1 bajt	POP	000000

a) $(PC) = 02_{\text{xeq}}$, pa sadržaj lokacije sa adresom 02 predstavlja prvi bajt naredbe

$$(02) = C0_{\text{xeq}} = 11000000_2,$$

na osnovu čega se zaključuje da se radi o bezaadresnoj naredbi POP (pogledati u tablici kodove načina adresiranja i kodove operacije). Rezultat izvršenja ove naredbe je smeštanje sadržaja prve pune lokacije steka u akumulator. Izgled steka pre izvršenja ove naredbe je

Lokacija:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Sadržaj:	0E	00	C0	5C	00	00	B8	30	FE	FF	00	FA	55	A3	AF	0E

a) $(PC) = 02_{\text{xeq}}$, pa sadržaj lokacije sa adresom 02 predstavlja prvi bajt naredbe

$$(02) = C0_{\text{xeq}} = 11000000_2,$$

na osnovu čega se zaključuje da se radi o bezadresnoj naredbi POP (pogledati u tablici kodove načina adresiranja i kodove operacije). Rezultat izvršenja ove naredbe je smeštanje sadržaja prve pune lokacije steka u akumulator. Izgled steka pre izvršenja ove naredbe je

SP		0E
	0E	0F
	////////	10
	////////	11
		

Lokacija:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Sadržaj:	0E	00	C0	5C	00	00	B8	30	FE	FF	00	FA	55	A3	AF	0E

a) $(PC) = 02_{\text{xeq}}$, pa sadržaj lokacije sa adresom 02 predstavlja prvi bajt naredbe

$$(02) = C0_{\text{xeq}} = 11000000_2,$$

na osnovu čega se zaključuje da se radi o bezadresnoj naredbi POP (pogledati u tablici kodove načina adresiranja i kodove operacije). Rezultat izvršenja ove naredbe je smeštanje sadržaja prve pune lokacije steka u akumulator. Izgled steka pre izvršenja ove naredbe je

SP		0E
	0E	0F
	////////	10
	////////	11
		

pa će sadržaj akumulatora po izvršenju POP naredbe biti jednak sadržaju lokacije 0F, tj

$$(Acc) = 0F_{\text{xeq}}$$

dok će SP opet ukazivati na prvu slobodnu lokaciju na steku, tj.

$$(SP) = 0F_{\text{xeq}}$$

Lokacija:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Sadržaj:	0E	00	C0	5C	00	00	B8	30	FE	FF	00	FA	55	A3	AF	0E

a) $(PC) = 02_{\text{xeq}}$, pa sadržaj lokacije sa adresom 02 predstavlja prvi bajt naredbe

$$(02) = C0_{\text{xeq}} = 11000000_2,$$

na osnovu čega se zaključuje da se radi o bezadresnoj naredbi POP (pogledati u tablici kodove načina adresiranja i kodove operacije). Rezultat izvršenja ove naredbe je smeštanje sadržaja prve pune lokacije steka u akumulator. Izgled steka pre izvršenja ove naredbe je

SP		0E
	0E	0F
	////////	10
	////////	11
		

pa će sadržaj akumulatora po izvršenju POP naredbe biti jednak sadržaju lokacije 0F, tj

$$(Acc) = 0F_{\text{xeq}}$$

dok će SP opet ukazivati na prvu slobodnu lokaciju na steku, tj.

$$(SP) = 0F_{\text{xeq}}$$

b) Pošto je izvršena naredba jednobajtna bezadresna naredba (vidi tablicu), broj pristupa memoriji u fazi čitanja naredbe je 1, nema pristupa memoriji u fazi određivanja adrese i čitanja operanda, a u fazi izvršenja memoriji se pristupa 1 put (kada se očitava prva puna lokacija steka na adresi 0F)

Kod nekog jednoadresnog računara prvi bajt naredbe sadrži kod operacije (bitovi od 7 do 2) i kod načina adresovanja (bitovi 1 i 0). Svi operandi su jednobajtni i prikazani su u drugom komplementu. Sve adrese su dvobajtna i smeštaju se niži bajt na nižoj, a viši bajt na višoj adresi. Stek raste od nižih ka višim adresama u memoriji, a SP ukazuje na prvu slobodnu lokaciju na steku. Sve naredbe koriste jednobajtni akumulator u kome ostaje rezultat. Sadržaj dela memorije je:

Lokacija:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Sadržaj:	0E	00	03	55	00	00	AA	C7	FF	FF	00	FA	55	AF	AF	0E

adresovanje	kod	dužina naredbe	operacija	kod
memorijsko direktno	00	3 bajta	ADD	010101
memorijsko indirektno	01	3 bajta	SUB	101010
neposredno	10	2 bajta	PUSH	111111
bezaadresna naredba	11	1 bajt	POP	000000

a) Ako je sadržaj brojača naredbi PC =02 i pokazivača steka SP = 0C odrediti decimalnu vrednost sadržaja akumulatora i SP-a kao i sadržaj indikatora N, Z, V i C po izvršenju svake od 4 sukcesivne naredbe. PUSH i POP ne menjaju sadržaj indikatora.

b) Koliki je broj pristupa memoriji u fazi čitanja naredbe, određivanja adrese, čitanja operanda za svaku od 4 izvršene naredbe.

a) 1. naredba

$$(PC) = 02_{\text{xeq}}$$

$$(02) = 03_{\text{xeq}} = 00000011_2$$

pa je ova naredba bezadresna naredba POP. Rezultat izvršenja ove naredbe je smeštanje sadržaja prve pune lokacije steka u akumulator. Izgled steka pre izvršenja ove naredbe je

SP		0C
	FA	0B
	00	0A
	FF	09
		

pa će sadržaj akumulatora po izvršenju POP naredbe biti jednak sadržaju lokacije 0B, tj.

$$(Acc) = FA_{\text{xeq}} = 11111010 = - (00000101 + 1) = -6_{10}$$

dok će SP opet ukazivati na prvu slobodnu lokaciju na steku, tj.

$$(SP) = 0B_{\text{xeq}} = 11_{10}$$

Kako bezadresne naredbe ne menjaju sadržaj indikatora, ako su pre izvršenja ove naredbe svi bili resetovani, njihov sadržaj i po izvršenju ove naredbe ostaje 0, tj

$$C = Z = N = V = 0$$

2. naredba

$(PC) = 03_{\text{xeq}}$ jer je reč o bezadresnoj naredbi, dužine 1B, pa je $PC = PC + 1$

$(03) = 55_{\text{xeq}} = 01010101_2$

Ovaj binarni sadržaj je prvi bajt 2-ge naredbe, na osnovu kog se zaključuje da se radi o 3-bajtnoj naredbi sabiranja, čiji je mnemonik ADD. 2-gi i 3-ći bajt ove naredbe sadrže adresu adrese drugog operanda (prvi operand je u akumulatoru), jer je način adresiranja memorijsko indirektno (vidi tablicu). Niži bajt adrese adese je sadržaj memorijske lokacije 04 i taj sadržaj je 00, pa je niži bajt adrese sadržaj lokacije 00 i taj sadržaj je 0E. Dakle, na lokaciji sa adresom 0E nalazi se drugi operand O_2 za operaciju sabiranja

$$O_2 = (0E) = AF_{\text{xeq}}$$

Rezultat sabiranja prvog operanda i drugog operanda ostaje u akumulatoru, a sabiranje se izvodi u komplementu dvojke, tj.

$$(Acc) = (Acc) + O_2 = FA + AF = 11111010 + 10101111 = 10101001_2 = A9_{\text{xeq}} = 10101001 = -(01010110 + 1) = -87_{10}$$

Pošto je ADD aritmetička naredba, ona utiče na promenu vrednosti indikatora, tj. biće

$C = 1$, jer postoji prenos pri sabiranju u najstarijem razredu,

$N = 1$, jer je najstarija cifra rezultata 1,

$Z = 0$, jer $F_{n-1} + F_{n-2} + \dots + F_0 \neq 0$

$V = 0$, jer $A_{n-1}B_{n-1}\overline{F}_{n-1} + \overline{A}_{n-1}\overline{B}_{n-1}F_{n-1} = 0$

Pošto je ADD aritmetička naredba, ona ne utiče na promenu sadržaja SP, pa je i posle izvršenja 2. naredbe $(SP) = 0B_{\text{xeq}}$.