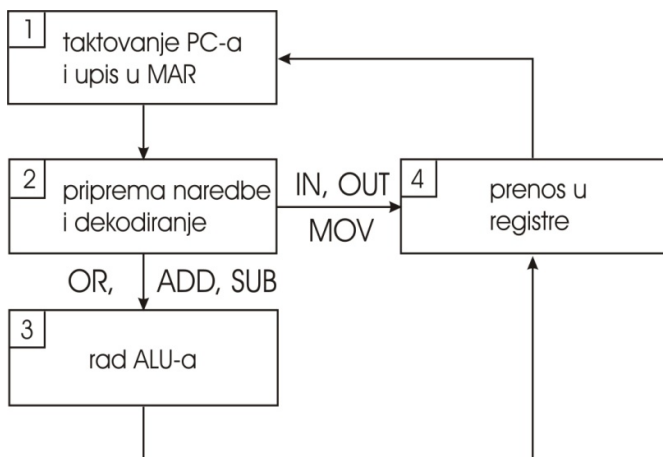


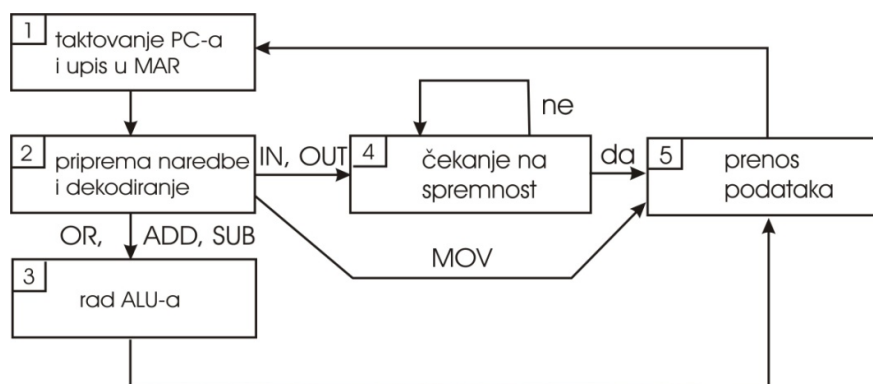
Prvi kolokvijum iz Računarskih sistema

1.) Za hipotetički računar projektovati dijagram prelaza sa 4 stanja i dijagram prelaza sa 5 stanja. Objasniti razlike u hardveru vezane za brojač stanja i dekođer stanja. (10 poena)

Rešenje:



Za dijagram sa 4 stanja brojač stanja se realizuje kao dvostepeni brojač, a “load” signal obezbeđuje prelaz iz 2 u 4.



Mašina kod koje postoji stanje čekanja na spremnost

S obzirom da dijagram prikazuje 5 različitih stanja, **brojač stanja** se realizuje kao trostepeni brojač po modulu 5. Kod starog dijagrama “load” je bio aktivan samo kada se prelazilo iz 2→4 za naredbe koje nisu koristile ALU. Ovde “load” ima složenu funkciju. Postavlja brojač stanja:

iz 2→4 (IN, OUT)

iz 2→5 (MOV, NOP)

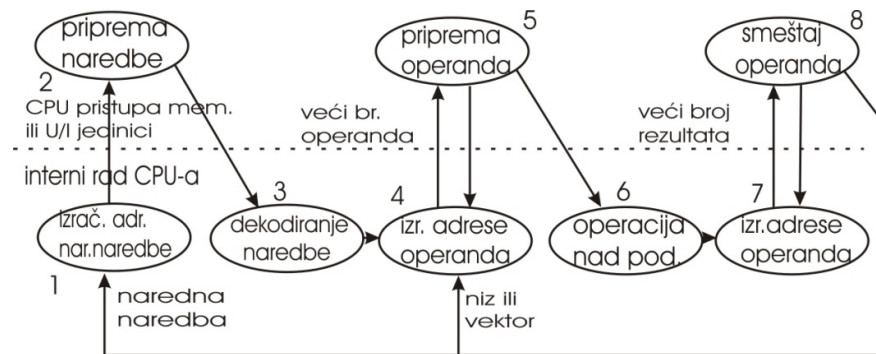
iz 3→5 (ADD, SUB, OR, ...).

2.) Nacrtati dijagram stanja za tok izvršenja naredbe. Sva stanja označiti brojevima. Za nizovnu instrukciju koja obavlja sabiranje:

a) dva vektora sa 4 elemenata zabeležiti redosled stanja kroz koja se prolazi da bi se izvršila instrukcija.

b) četiri vektora sa 3 elemenata zabeležiti redosled stanja kroz koja se prolazi da bi se izvršila instrukcija. (10 poena)

Rešenje:



a) 1,2,3,4,5,4,5,6,7,8,4,5,4,5,6,7,8,4,5,4,5,6,7,8,4,5,4,5,6,7,8

b) 1,2,3,4,5,4,5,4,5,4,5,6,7,8,4,5,4,5,4,5,4,5,6,7,8,4,5,4,5,4,5,4,5,6,7,8

3.) Projektovati procesnu jedinicu koja je organizovana oko dve magistrale:

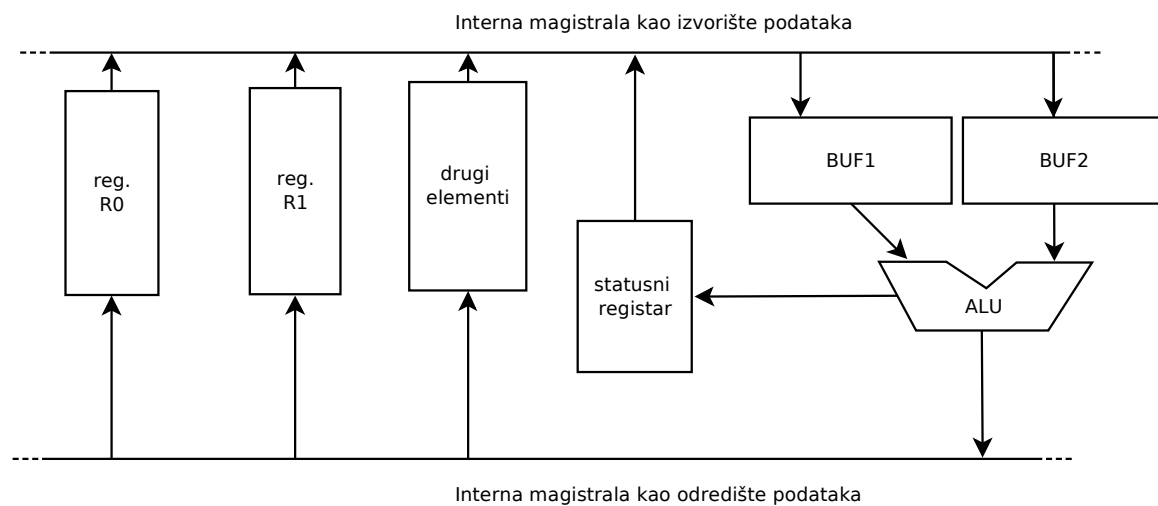
a) Tako da se na putu prvog operanda nalazi memorijski element BUF1, a na putu drugog operanda BUF2;

b) Tako da se na putu prvog operanda nalaze memorijski elementi BUF1 i BUF2, a na putu drugog operanda nema memorijskih elemenata.

Za svaku od arhitektura pod a) i b) napisati od kojih se mikroinstrukcija sastoji operacija

$(R0) = (R0) + (R1)$. Tamo gde postoji mogućnost hazarda navesti mikrooperaciju koja dovodi do hazarda. (10 poena)

Rešenje:



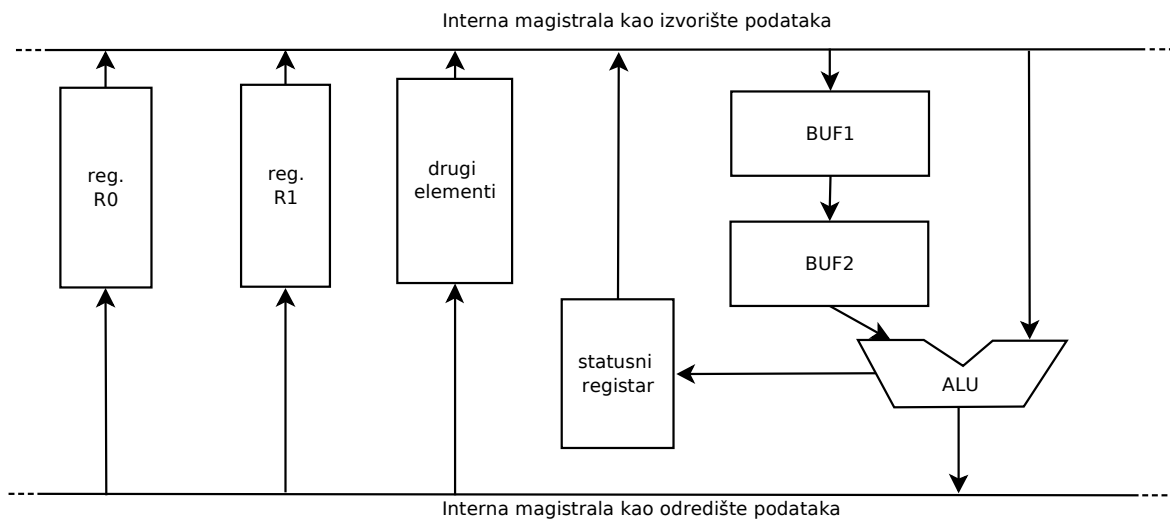
$(R0) = (R0) + (R1)$

$(BUF1) \leftarrow (R0)$

$(BUF2) \leftarrow (R1)$

$(R0) \leftarrow (BUF1) + (BUF2)$

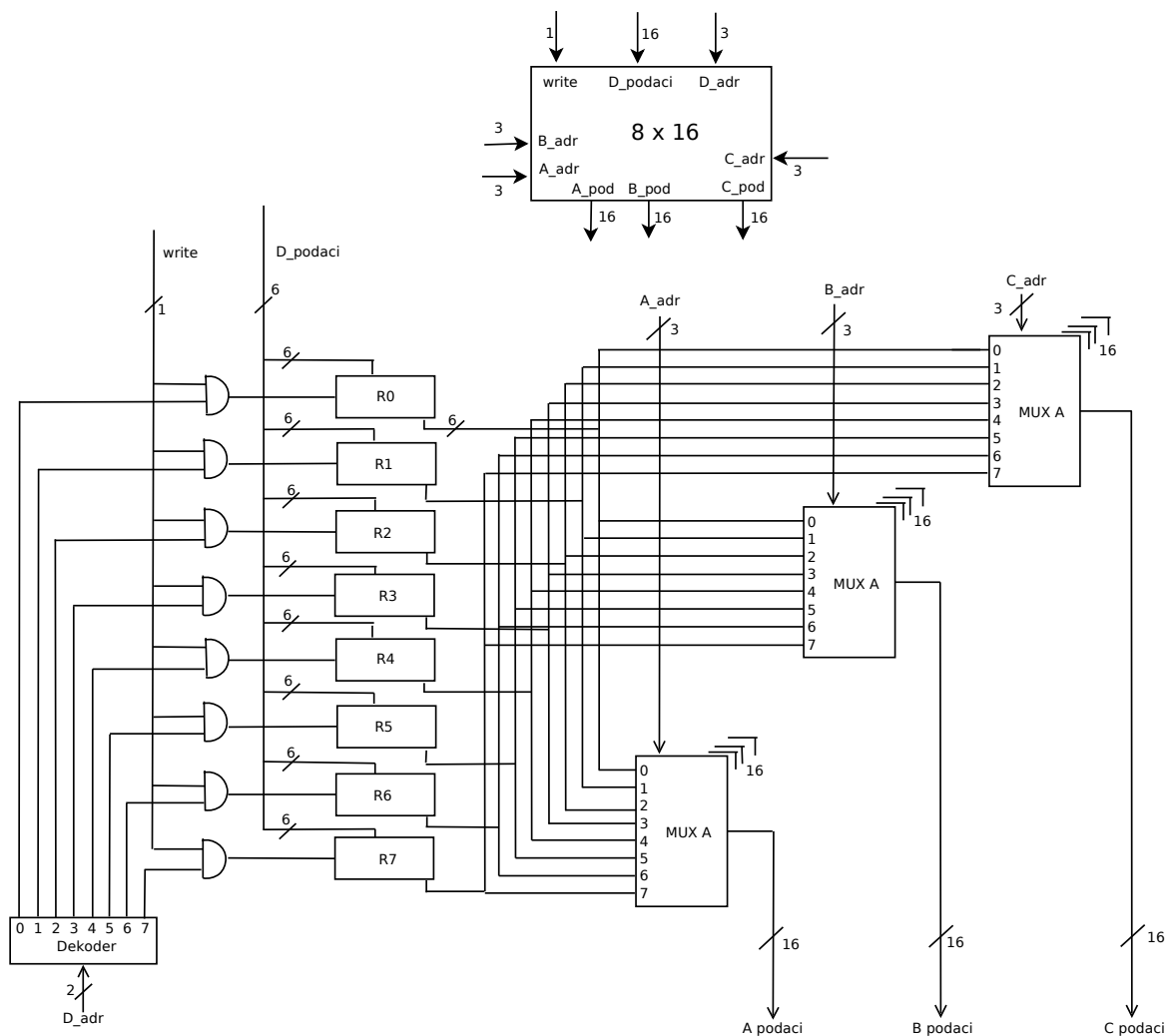
Ne postoji mogućnost hazarda.



$(R0) = (R0) + (R1)$
 $(BUF1) \leftarrow (R0)$
 $(BUF2) \leftarrow (BUF1)$
 $(R0) = (BUF2) + (R1)$
 $Nd(R1) \leftarrow (BUF2) + (R1)$

4.) Projektovati RF polje 8x32 kod koga postoji mogućnost upisa podatka i čitanja tri podatka istovremeno. Prikazati projektovani blok sa svim potrebnim ulazima i izlazima, dati unutrašnju strukturu i navesti koliko kojih komponenti je upotrebljeno za projektovanje. (10 poena)

Rešenje:



Potrebne komponente:
 8 registara dužine 32 bita
 1 dekoder 1 od 8
 8 dvoulaznih I kola
 $3 \times 32 = 96$ multipleksera 8 u 1

5.) Popuniti tablicu maksimalnim brojem čipova zadatog kapaciteta koji se mogu povezati u memorijski podsistem primenom linearnog izbora čipa na adresnu magistralu sa zadatim brojem adresnih linija. (10 poena)

Broj linija adresne magistrale \ Kapacitet čipa	1K	8K	32K
16			
32			
64			

Rešenje:

Broj linija adresne magistrale \ Kapacitet čipa	1K	8K	32K
16	6	3	1
32	22	19	17
64	54	51	49

6.) Segment sa podacima definisan je na sledeći način:

```
podaci segment
    A1 db 5,6,7
    B1 db 4 dup(?)
    A2 dw 8B2h
    B2 dw 2 dup(?)
    A3 db 0A5h,65
    B3 dd 256256h
podaci ends
```

a) Nacrtati raspored bajtova u memoriji u segmentu `podaci` i označiti na slici iznose pomeraja lokacija na koje ukazuju navedeni simboli. Vrednosti u memorijskim lokacijama navesti u heksadekadnom brojnem sistemu. Koja je vrednost u registru `CS` ako se zna da se segment programa nalazi odmah nakon segmenta `podaci`, i da je fizička adresa podatka `B1` `A233h`? Obrazložiti odgovor. (12 poena)

b) Ukoliko je u registar `DS` upisana adresa početka segmenta `podaci`, navesti vrednosti i poreklo operanada svake od narednih instrukcija, kao i sve efekte koje instrukcija proizvodi : `MOV [DI], AX; MOV [A2+3], DI; LEA A3, AX; SUB AX, offset A2`. Pretpostaviti da se svaka instrukcija izvršava nad istim početnim stanjem memorije i registara, a da je trenutna vrednost u registrima: `AX=0ABDCh`, `DI=000Eh`. Ukoliko neka instrukcija ne može da se prevede, navesti razloge za to. (10 poena)

c) Napisati sadržaj segmenta sa instrukcijama koji će imati sledeće efekte, redom:

- Sabira podatke `A1` i `A2`, vodeći računa o eventualnom izlaznom prenosu.
- Dobijeni zbir deli podatkom `A3`; podešava vrednost podatka `B1` tako da ima vrednost celog dela količnika prethodnog deljenja.
- Ostatak pri prethodnom deljenju množi podatkom `A3`; dobijeni proizvod smešta u memoriju u promenljivu `B2`.

Međurezultate čuvati u registrima, ne rezervisati dodatne memorijske lokacije. Opisati u komentarima ulogu svake instrukcije u sprovođenju traženog izračunavanja. Podatke `A1`, `A2` i `A3` smatrati neoznačenim brojevima. (18 poena)

NAPOMENA: rešenja bez obrazloženja ili komentara neće biti bodovana.

Rešenje:

a)

		pomeraj	fizička adresa
A1 →	5h	0	
	6h		
	7h		
B1 →	<neinic.>	3	0A233h
	<neinic.>		
	<neinic.>		
A2 →	B2h	7	
	8h		
B2 →	<neinic.>	9	
	<neinic.>		
	<neinic.>		
	<neinic.>		
A3 →	A5h	13	
	41h		
B3 →	56h	15	A240h
	62h		
	25h		
	0h		A242h

Ako je B1 na A233h, brojanjem se može ustanoviti da je poslednja rezervisana lokacija u segmentu podataka na fiz. adresi 0A242h. Segment koji sledi može najranije da počne na adresi 0A250h. U segmentnom registru se ne čuvaju najniža četiri bita fizičke adrese, stoga je CS=0A25h.

b)

```
MOV [DI], AX;
```

Na lokaciju koja ima pomeraj 14 (DI=000Eh), i na narednu lokaciju se upisuje vrednost AX=0ABDCh. Na pomeraju 14 je bio podatak 41h, umesto njega se upisuje DCh; na adresi 15 se umesto podatka 56h upisuje ABh.

```
MOV [A2+3], DI;
```

Na dve lokacije počev od pomeraja 16 (pomeraj simbola A3 je 13) se upisuje sadržaj registra DI=000Eh. Na pomeraju 16 je bio podatak 62h, umesto njega se upisuje 0Eh; na pomeraju 17 je bio podatak 25h, umesto njega se upisuje 00h.

```
LEA A3, AX;
```

Prvi operand je simbol A3, vrednost mu je A5h a nalazi se na adresi DS:13. Drugi operand je registar AX, vrednost mu je ABDCh. Instrukcija ne može da se prevede, drugi operand instrukcije LEA mora da bude memorijski operand.

```
SUB AX, offset A2
```

Prvi operand je AX=0ABDCh; drugi operand je vrednost pomeraja simbola A2, što iznosi 7. Instrukcija SUB oduzima vrednosti operandu i rezultat smešta u prvi operand. Vrednost 0ABDCh-7=ABD5h će se smestiti u registar AX. Instrukcija utiče na statusne flegove, vrednosti flegova nakon izvršenja ove instrukcije će biti: CF=0, (priznaje se i rešenje bez ostalih flegova) PF= 0, AF=0, ZF=0, SF=1, OF=0.

c)

```
code segment
assume cs:code, ds:data
```

start:

```
MOV AX, data
MOV DS, AX

MOV AX, word ptr A1 ; niža reč podatka A1
MOV DI, byte ptr A1+2 ; treći bajt A1
MOV DH, 0 ; proširuje se 0 da prihvati izl. prenos

ADD AX, A2 ; A2+niža reč A1
ADC DX, 0 ; treći bajt A1 + izl prenos iz preth sab., izl prenos u DH

DIV word ptr A3 ; zbir preth sabiranja iz DX:AX se deli 16b podatkom s
adrese A3 (mora word ptr jer je A3 dekl. sa db)
MOV word ptr B1, AX ; ceo deo količnika iz AX kopira se u nižu reč B1
MOV word ptr B1+2, 0 ; B1 je 4B, pa se viša reč proširuje sa 0

MOV AX, DX ; ostatak pri deljenju iz DX kopira se u AX
MUL word ptr A3 ; množi AX sa A3

MOV B2, AX ; niži deo proizvoda u nižu reč B2
MOV B2+2, DX ; viši deo proizvoda u višu reč B2

code ends
end start
```

DOZVOLJENA LITERATURA Za vreme ispita može se koristiti "Spisak instrukcija za μ P i APX86" ukoliko nema dopisanih delova.

Vreme izrade: 120 minuta

PREDMETNI NASTAVNIK