

Arhitektura računala 2

1. međuispit, problemski dio (60% bodova)

1. Zadana je ispisna memorija (ROM) kapaciteta 8192 bajta. Pored adresnih i podatkovnih izvoda, memorija ima i jedan ulaz za omogućavanje E. Prikazati shemu spajanja zadane memorije na računalni sustav s 16-bitnom adresnom i 8-bitnom podatkovnom sabirnicom. Memoriju treba spojiti koristeći potpuno dekodiranje adresa tako da memorija zauzima adresni prostor \$c000 do \$dfff.
Što bi se dogodilo u takvom sustavu kad bi procesor izveo instrukciju sta \$cdef? (12 bodova)
2. Procesor MC68000 poziva potprogram instrukcijom \$1000: jsr \$2000. Pretpostaviti da operacijski kod odnosno operand instrukcije zauzimaju ukupno 6 bajta. Neka se parametri potprograma ne prenose preko stoga, te neka potprogram koristi dvije memorijske lokalne varijable širine po jedan bajt.
Neposredno nakon izvršavanja instrukcije potprograma \$2100: move d0,d1 procesor prima zahtjev za prekid te se poziva prekidni potprogram na adresi \$ff0a0.
Skicirati stanja stogova procesora prije poziva potprograma, neposredno prije primanja prekida, tijekom obrade prekida, nakon povratka iz prekidnog potprograma, te nakon povratka u glavni program. (12 bodova)
3. Za pojednostavljeni model procesora prikazati stanje na sabirnicama tijekom izvođenja programa:
\$0010: inc \$13
\$0012: lda \$80

Neka je sadržaj radne memorije kako slijedi:

Neposredno nakon izvršavanja instrukcije potprograma \$2100
za prekid te se poziva prekidni potprogram na adresi \$ff0a0.

Skicirati stanja stogova procesora prije poziva potprograma, neposredno prije primanja prekida, tijekom obrade prekida, nakon povratka iz prekidnog potprograma, te nakon povratka u glavni program. (12 bodova)

3. Za pojednostavljeni model procesora prikazati stanje na sabirnicama tijekom izvođenja programa:

```
$0010: inc $13  
$0012: lda $80
```

Neka je sadržaj radne memorije kako slijedi:

```
$007f: $00  
$0080: $01  
$0081: $02
```

Neka su operacijski kodovi za instrukcije inc i lda s izravnim adresiranjem nulte stranice \$23 i \$18. (12 bodova)

4. Preinačite model osaminstrukcijskog procesora sa slike na način da dodate instrukciju uvjetnog indirektnog skoka jmpni. Instrukcija jmpni prihvaća grananje ako je rezultat posljednje aritmetičke operacije negativan, a odredišna adresa grananja se pribavlja s adrese specificirane operandom instrukcije.

Odredite jednadžbe signala koji upravljaju pribavljanjem i izvođenjem instrukcije jmpni. (12 bodova)

Za model mikroprogramiranog procesora koji je zadan slikom i formatom mikroriječi, napisati mikroprogram i odrediti sadržaj mikroprogramske memorije za fazu izvrši instrukcije djnz (decrement B and jump relative to A if not zero).

Instrukcija djnz prvo dekrementira registar B. Ako rezultat nije jednak nuli, instrukcija grana na adresu $PC+A$, a inače se izvodi sljedeća instrukcija u nizu.

Neka je operacijski kod instrukcije \$ab, te neka mikroprogram za fazu PRIBAVI ima početnu adresu

Neka je sadržaj radne memorije kako slijedi:

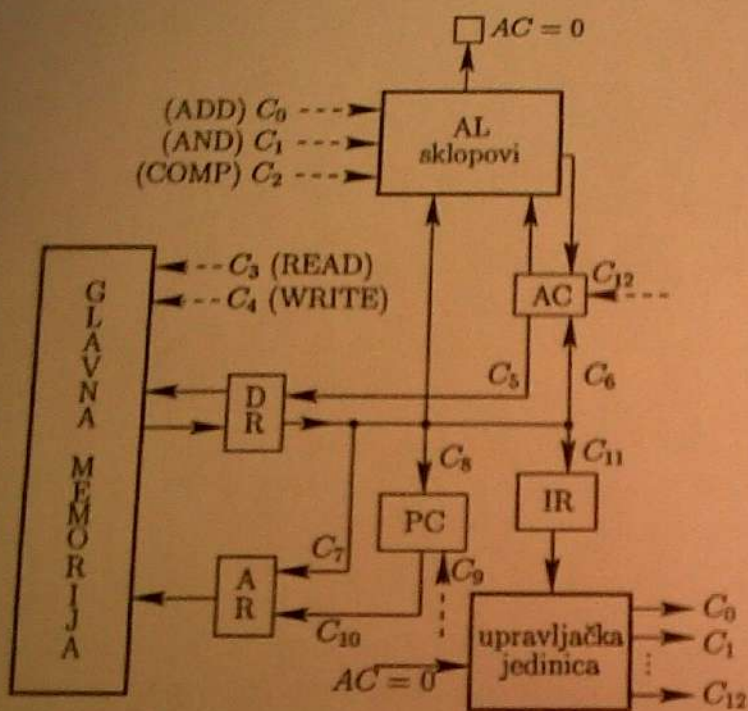
\$007f: \$00

\$0080: \$01

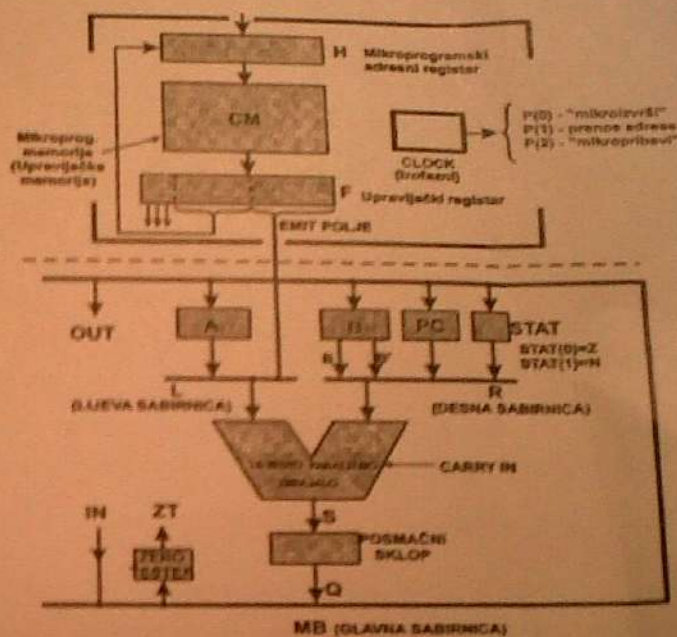
\$0081: \$02

Neka su operacijski kodovi za instrukcije inc i lda s izravnim adresiranjem nulte stranice \$23 i \$18. (12 bodova)

4. Preinačite model osaminstrukcijskog procesora sa slike na način da dodate instrukciju uvjetnog indirektnog skoka jmpni. Instrukcija jmpni prihvaća grananje ako je rezultat posljednje aritmetičke operacije negativan, a odredišna adresa grananja se pribavlja s adrese specificirane operandom instrukcije. Odredite jednadžbe signala koji upravljaju pribavljanjem i izvođenjem instrukcije jmpni. (12 bodova)
5. Za model mikroprogramiranog procesora koji je zadan slikom i formatom mikroriječi, napisati mikroprogram i odrediti sadržaj mikroprogramske memorije za fazu izvrši instrukcije djnz (decrement B and jump relative to A if not zero). Instrukcija djnz prvo dekrementira registar B. Ako rezultat nije jednak nuli, instrukcija grana na adresu PC+A, a inače se izvodi sljedeća instrukcija u nizu. Neka je operacijski kod instrukcije \$ab, te neka mikroprogram za fazu PRIBAVI ima početnu adresu \$f0. (12 bodova)



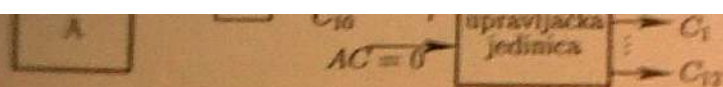
Slika uz zadatak 4: organizacija osaminstrukcijskog procesora



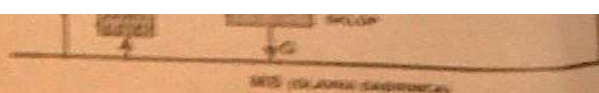
Slika uz zadatak 5: organizacija mikroprogramiranog procesora

31	29	26	24	22	19	17	15	13	7	0
CA	CB	COP	CSH	CMB	CAB	CBB	CST	CNA	CEM	

CA	CB	COP	CSH
00 ... nema prijenosa	000 ... nema prijenosa	00 ... suma uz C=0	00 ... MB ← S
01 ... L ← [0, F(CEM)]	001 ... R ← B	01 ... suma uz C=1	01 ... MB ← shr S
10 ... L ← [F(CEM), 0]	010 ... R ← B*	10 ... ne koristi se	10 ... MB ← shl S



Slika uz zadatak 4: organizacija osaminstrukcijskog procesora



Slika uz zadatak 5: organizacija mikroprogramiranog procesora

31	29	26	24	22	19	17	15	13	7	0
CA	CB	COP	CSH	CMB	CAB	CBB	CST	CNA	CEM	

CA	CB	COP	CSH
00 ... nema prijenosa	000 ... nema prijenosa	00 ... suma uz C=0	00 ... MB ← S
01 ... L ← [0, F(CEM)]	001 ... R ← B	01 ... suma uz C=1	01 ... MB ← shr S
10 ... L ← [F(CEM), 0]	010 ... R ← B*	10 ... ne koristi se	10 ... MB ← shl S
11 ... L ← A	011 ... R ← PC	11 ... ne koristi se	11 ... MB ← IN
	100 ... R ← SR		
	101 ... R ← 0		
CMB	CAB	CBB	CST
000 ... nema prijenosa	00 ... H(1) ← 0	00 ... H(0) ← 0	00 ... SR se ne mijenja
001 ... A ← MB	01 ... H(1) ← 1	01 ... H(0) ← 1	01 ... SR(0) ← ZT
010 ... B ← MB	10 ... H(1) ← SR(1)	10 ... H(0) ← SR(1)	10 ... SR(1) ← MB(15)
011 ... PC ← MB	11 ... H(1) ← SR(0)	11 ... H(0) ← MB(15)	11 ... SR(0) ← ZT
100 ... SR ← MB			SR(1) ← MB(15)
101 ... OUT ← MB			

Slika uz zadatak 5: format mikroinstrukcijske riječi