

JMBAG

IME I PREZIME

BROJ BODOVA

Građa računala

Drugi jesenski ispitni rok - 8. rujna 2026. godine

ZADATAK 1.

(10 bodova)

Na raspolaganju su dva identična memorijska RAM modula kapaciteta 1 GB, sa sljedećim priključcima: $A_0 - A_{29}$, $D_0 - D_7$, $R/\overline{W}$, E i $\overline{E}$. Dodatno, na raspolaganju je ROM modul kapaciteta 64KB s priključcima $A_0 - A_{15}$, $D_0 - D_7$ i $\overline{E}$.

- (a) Skicirajte shemu spajanja ovih triju modula na računalo s 32-bitnom adresnom i 8-bitnom podatkovnom sabirnicom, tako da se RAM javlja u kontinuiranom adresnom prostoru s početnom adresom 40000000_{16} , a ROM na najvišim mogućim adresama.
- (b) Ako bismo željeli proširiti kapacitet RAM-a računala iz zadatka (a) dodavanjem samo jednog dodatnog RAM modula u adresni prostor između postojećeg RAM-a i ROM-a, koliki bi bio maksimalni kapacitet takvog dodatnog RAM modula? Koji bi raspon adresa on zauzima, uz pretpostavku da su te adrese u kontinuitetu s postojećim RAM-om iz zadatka (a)?

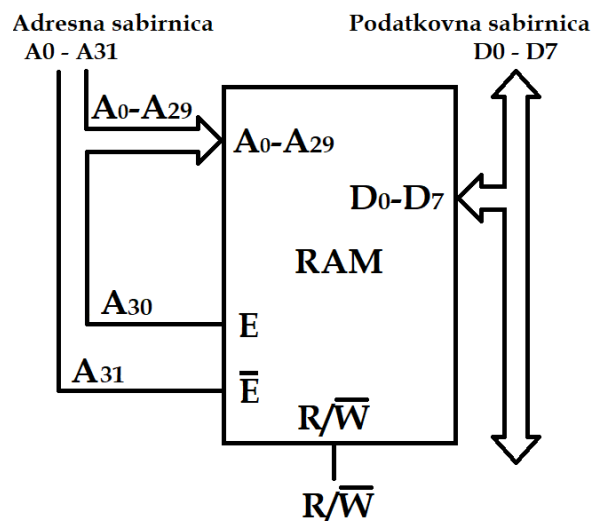
Rješenje. $1\text{ GB} = 2^{30}\text{ B}$, $64\text{ KB} = 2^6 \cdot 2^{10}\text{ B} = 2^{16}\text{ B}$, ukupni adresni prostor 2^{32} B

- (a) Adresni prostor prvog RAM modula ide:

od $4000\ 0000_{16} = 0100\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_2$

do $7FFF\ FFFF_{16} = 0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_2$.

Prema tome nižih 30 bitova adresne sabirnice ide na priključke modula $A_0 - A_{29}$, a dva najviša bita adrese su $A_{31} = 0$, $A_{30} = 1$, pa je najelegantnije rješenje spojiti A_{31} na $\overline{E}$, a A_{30} na E . Shema priključenja prvog RAM modula nalazi se na slici desno:

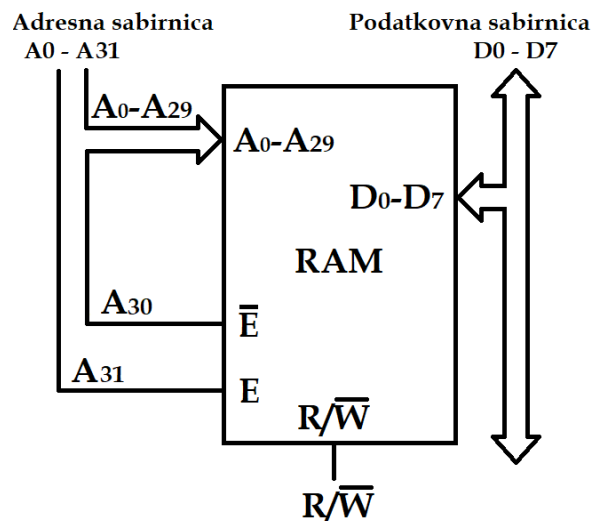


Drugi RAM modul se kontinuirano nastavlja na adresni prostor prvoga, pa se dakle rasprostire:

od $8000\ 0000_{16} = 1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_2$

do $BFFF\ FFFF_{16} = 1011\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_2$.

Prema tome nižih 30 bitova adresne sabirnice ide na priključke modula A_0-A_{29} , a dva najviša bita adrese su $A_{31} = 1$, $A_{30} = 0$, pa je najelegantnije rješenje spojiti A_{31} na E , a A_{30} na $\bar{E}$. Shema priključenja drugog RAM modula nalazi se na slici desno:

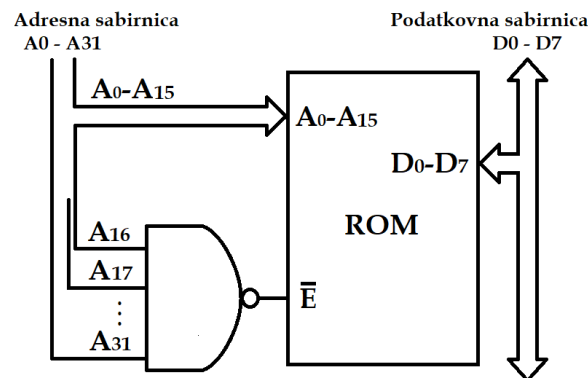


Obzirom da se ROM javlja na najvišim mogućim adresama dobivamo da se on javlja na adresama

od $FFFF\ 0000_{16} = 1111\ 1111\ 1111\ 1111\ 0000\ 0000\ 0000\ 0000_2$

do $FFFF\ FFFF_{16} = 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_2$.

Prema tome nižih 16 bitova adresne sabirnice ide na priključke modula A_0-A_{15} , a 16 najviših bitova adrese spajamo na jedan sklop NI čiji izlaz spajamo na $\bar{E}$. Shema priključenja ROM modula nalazi se na slici desno:



- (b) Slobodni adresni prostor između postojećeg RAM-a i ROM-a je $C000\ 000_{16} - FFFE\ FFFF_{16}$. Veličina tog prostora je $FFFE\ FFFF_{16} - BFFF\ FFFF_{16} = 3FFF\ 0000_{16}$. Vrijedi $2^{29} = 2000\ 0000_{16} < 3FFF\ 0000_{16} < 4000\ 0000_{16} = 2^{30}$ pa zaključujemo da je maksimalni kapacitet takvog dodanog RAM-a $2^{29}\text{ B} = 2^9 \cdot 2^{20}\text{ B} = 512\text{ MB}$. On bi zauzimao adrese $C000\ 0000_{16} - DFFF\ FFFF_{16}$.

■

JMBAG

IME I PREZIME

BROJ BODOVA

Građa računala

Drugi jesenski ispitni rok - 8. rujna 2026. godine

ZADATAK 2.

(10 bodova)

Računalo temeljeno na pojednostavljenom modelu 8-bitnog mikroprocesora izvršava sljedeći programski odsječak koji započinje na adresi 0100_{16} :

```
LDA #$1A      ;napuni akumulator A s (usputnom) konstantom 1A
STA $0AAA     ;pohrani sadržaj akumulatora A na memorijsku lokaciju 0AAA
```

Operacijski kod instrukcije LDA je 86_{16} , a operacijski kod instrukcije STA je $B7_{16}$.

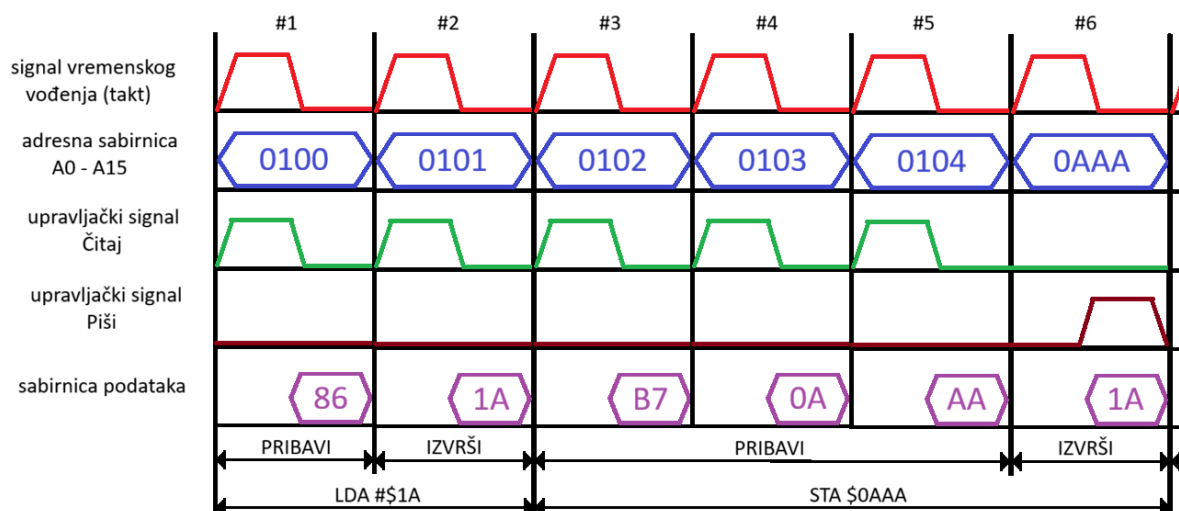
- Skicirajte sadržaj relevantnog dijela memorije.
- Nacrtajte stanje na vanjskim sabirnicama tijekom izvođenja programskog odsječka.
- Odredite konačne sadržaje registara PC, IR, A i DC nakon izvođenja navedenog programskog odsječka.

Rješenje.

(a)

adresa	sadržaj	
0100	B6	LDA #\$1A
0101	1A	
0102	B7	
0103	0A	STA \$0AAA
0104	AA	

(b)



(c) (PC) = 0105, (IR) = B7, (A) = 1A, (DC) = 0AAA

Napomena. U podzadatku 2.b) moglo bi se filozofirati oko toga kad prestaje faza pribavi prve instrukcije i počinje faza izvrši; naime, konstanta 1A je isto dio same instrukcije, i dok se ona čita, to bi po nekoj logici trebala biti i dalje faza pribavi. No gdje je onda faza izvrši i kako ona izgleda za prvu instrukciju? To ovisi o izvedbi samog procesora, a budući da je "pojednostavljeni model" izmišljeni procesor za koji nismo razradili te detalje, tu se javlja nejednoznačnost. Jedna mogućnost je da procesor već pri čitanju konstante odmah smješta tu konstantu u akumulator A; tada bi se faza izvrši zapravo preklapala s fazom pribavi i bilo bi kao što je nacrtano. Druga mogućnost je da procesor u fazi pribavi zapamti tu konstantu 1A u nekom privremenom registru, ali ne još u akumulatoru. Onda ju u fazi izvrši prebacuje iz privremenog registra u akumulator, što troši 1 takt, ali se ne vidi na vanjskoj sabirnici (jer nema komunikacije s memorijom), pa je procesor odspojen od sabirnice, slično kao u primjeru s predavanja.

Građa računala

Drugi jesenski ispitni rok - 8. rujna 2026. godine

ZADATAK 3.

(10 bodova)

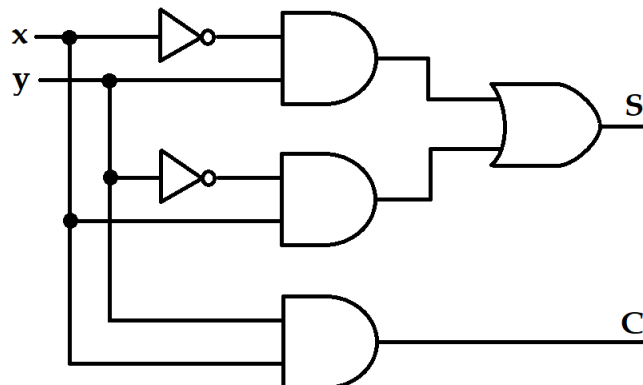
- Ukratko objasnite pojmove poluzbrajala i potpunog zbrajala.
- Napišite logičke jednačbe za poluzbrajalo te nacrtajte logičku shemu sklopa poluzbrajala ostvarenog osnovnim logičkim sklopovima.
- Shematski prikažite kako se pomoću dvaju poluzbrajala i sklopa logičko ILI ostvaruje potpuno zbrajalo.

Rješenje.

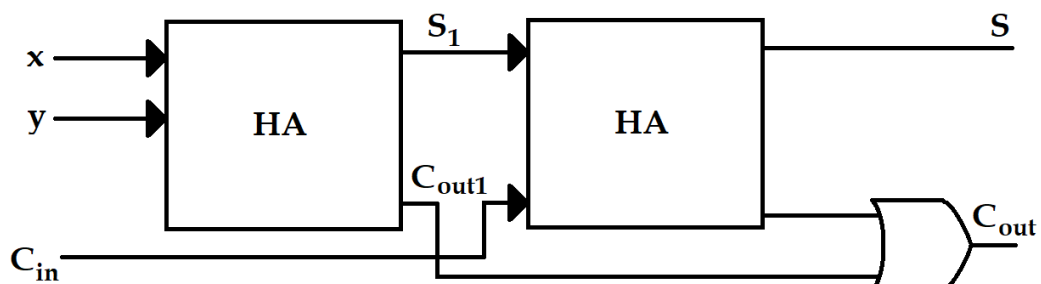
- Poluzbrajalo je kombinaijski sklop za zbrajanje dvaju binarnih znamenaka. Potpuno zbrajalo je sklop koji zbraja tri binarne znamenke: dvije znamenke na bitovnoj poziciji i jednu binarnu znamenku koja predstavlja bit prijenosa s manje značajne bitovne pozicije.
- Za ulaz x , ulaz y , izlaz S (suma) i bit prijenosa C :

$$S = \bar{x} \cdot y + x \cdot \bar{y}$$

$$C = x \cdot y$$



- Ostvarivanje potpunog zbrajala pomoću dvaju poluzbrajala (HA) i sklopa logičko ILI:



JMBAG

IME I PREZIME

BROJ BODOVA

Građa računala

Drugi jesenski ispitni rok - 8. rujna 2026. godine

ZADATAK 4.

(10 bodova)

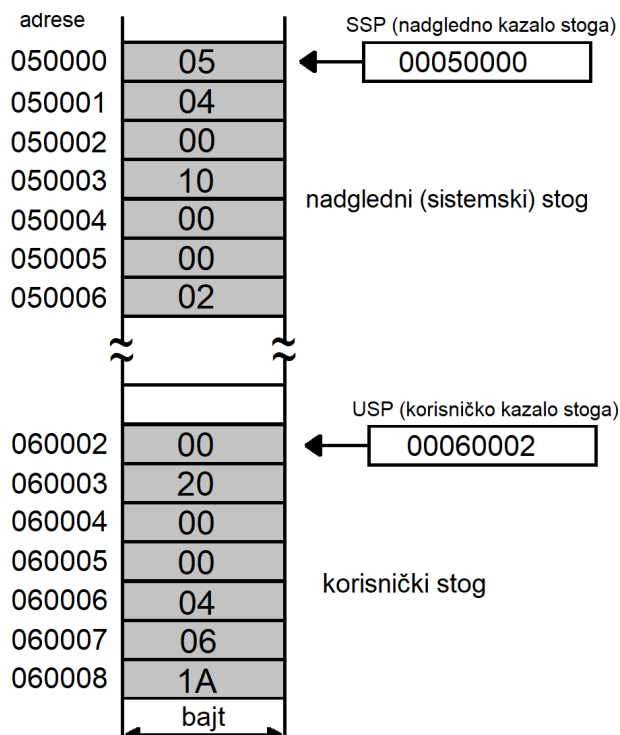
Slika desno prikazuje stanje stogova za računalo na bazi procesora MC68000. Nadgledni stog sadrži pohranjen minimalni kontekst, a korisnički stog sadrži povratnu adresu iz potprograma. Procesor posluhuje prekidni potprogram, s tim da je nevješti programer povlaštenu instrukciju RTE (kao zadnju instrukciju prekidnog programa) zamijenio programskim odsječkom:

```
MOVE.W (A7)+, SR
```

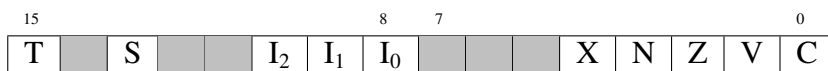
```
RTS
```

Odgovorite na sljedeća pitanja. Svaki odgovor kratko obrazložite!

- U kojem se stanju (načinu rada) nalazio procesor neposredno prije prihvatanja prekida?
- Koja je razina prekida bila definirana u procesoru prije negoli se dogodio prekid?
- Odredite sadržaje registra SR i PC nakon izvođenja (korektne) instrukcije RTE.
- Odredite sadržaje registra SR i PC nakon izvođenja programskog odsječka kojim je zamijenjena instrukcija RTE. Kratko komentirajte posljedice dobivenog rezultata.



Organizacija statusnog registra SR prikazana je donjom slikom:



Rješenje.

- Minimalni kontekst pohranjen na nadglednom stogu sadrži pohranjen sadržaj statusnog registra

$$SR = 0504_{16} = 0000\ 0101\ 0000\ 0100_2$$

Vidimo da zastavica S nije postavljena ($S=0$), što znači da je procesor bio u korisničkom načinu rada.

- (b) Prema pohranjenom sadržaju statusnog registra koji smo odredili u (a) podzadatku, vidimo da je vrijednost zastavica $I_2I_1I_0 = 101_2$. Prema tome je definirana razina prekida (prije nego se dogodio prekid) bila $101_2 = 5$.
- (c) Instrukcija RTE bi skinula 6 bajtova s nadglednog stoga, pri čemu viša dva bajta stavlja u SR, a preostala četiri u PC. Tako bismo dobili sljedeće sadržaje traženih registara: $SR = 0504_{16}$, $PC = 00100000_{16}$.
- (d) Instrukcija `MOVE.W (A7)+, SR` skida dva bajta s nadglednog stoga i stavlja ih u statusni registar SR. Kao što je dobiveno u (a) podzadatku, time se zastavica S postavlja na 0, pa procesor prelazi u korisnički način rada. Zbog toga instrukcija RTS skida 4 bajta s korisničkog (a ne nadglednog) stoga i stavlja ih u PC. Tako dobivamo da PC sadrži adresu 00200000_{16} umjesto željene adrese 00100000_{16} s nadglednog stoga. Posljedica je da će program „zalutati.”

■

Grada računala

Drugi jesenski ispitni rok - 8. rujna 2026. godine

ZADATAK 5.

(10 bodova)

Navedite i ukratko opišite tri osnovna načina organizacije priručne memorije prema načinu smještanja blokova u bločne priključke.

Rješenje.

- **priručna memorija s potpuno asocijativnim preslikavanjem**

- priručni blok ili linija veličine b riječi iz glavne memorije može se smjestiti u bilo koji slobodni bločni priključak priručne memorije. Adresa kojom se referencira memorija sastoji se od dva dijela: k značajnijih bitova adrese koji čine značku i w manje značajnih bitova adrese koji služe za izbor jedne od b riječi u izabranom bloku. Na temelju značke utvrđuje se nalazi li se referencirani blok u priručnoj memoriji.

- **priručna memorija s izravnim preslikavanjem**

- svaki blok glavne memorije može se smjestiti u samo jedan točno određeni bločni priključak priručne memorije. Pravilo koje određuje koji je to bločni priključak je sljedeće: blok $B_{M2}(i)$ iz glavne memorije smješta se u bločni priključak $B_{M1}(j)$ priručne memorije pri čemu vrijedi

$$j = i \text{ (modulo } B_{M1}\text{)},$$

gdje je B_{M1} ukupan broj bločnih priključaka priručne memorije, j indeks bločnog priključka u priručnoj memoriji te i indeks bloka u glavnoj memoriji.

- **priručna memorija sa skupnim asocijativnim preslikavanjem**

- bločni priključci priručne memorije organizirani su u skupine tako da mehanizam preslikavanja dopušta da se blok iz glavne memorije priključuje u bilo koji bločni priključak koji pripada odgovarajućoj skupini. Pretpostavimo da priručna memorija ima B_s skupina bločnih priključaka. Svaka se skupina sastoji od k bločnih priključaka. Ukupno je $B_s \times k$ bločnih priključaka u priručnoj memoriji. Blok iz glavne memorije $B_{M2}(i)$ može se priključiti u bilo koji od k slobodnih bločnih priključaka skupine j , pri čemu je

$$j = i \text{ (modulo } B_s\text{)}.$$

■