

1. Trajanje ispita 180 minuta.
2. Odgovori se daju u vežbanci ili na formularu.
3. Na naslovnoj strani obavezno zaokružiti redne brojeve zadataka koji su rađeni.

1. KOLOKVIJUM

1. [10] Nacrtati realizaciju memorijskog podsistema u računaru sa 32-bitnom asinhronom magistralom (asinhroni pristup je kontrolisan signalom ACK sa aktivnom ivicom) koja ima 20-bitnu adresnu magistralu i mogućnost upravljanja bajtovima. Veličina potrebne memorije je 64k x 8 i nalazi se u najvišem delu memorijskog prostora. Na raspolaganju su 4 standardne memorijske komponente SRAM tipa veličine 16k x 8. U slučaju 8-bitnog pristupa za A0=A1=0, podatak ide po delu magistrale D7..D0, za A0=1 A1=0 po delu magistrale D15..D8, A0=0 A1=1 po delu magistrale D23..D16 i za A0=1 A1=1 po delu magistrale D31..D24. U slučaju 16-bitnog pristupa A0=0 i za A1=0, podatak ide po delu magistrale D15..D0, a za A1=1 podatak ide po delu magistrale D31..D16. Memorija je dovoljno brza.

2. [10] Nacrtati opštenamenski prekidni kontroler sa 8 ulaza koji prihvata prekida sa aktivnim nivoom logičke jedinice. Posle svakog prihvaćenog prekida svi prekidi istog i nižeg prioriteta se automatski maskiraju.

3. [10] Nacrtati vremenski dijagram rada uopštene 16 bitne **sinhrone** magistrale koja **ima upravljanje bajtovima u izlaznom 16 bitnom** ciklusu na **neparnu** memorijsku lokaciju. Označiti sve relevantne vremenske parametre na dijagramu (t1, t2, sa naznakom da li je min ili max t1(max)....). Naznačiti koji vremenski parametri su bitni za projektanta perifera koji se priključuju na magistralu

4. [20] Namenski sistem koristi procesor baziran na 32bitnoj RISC-V arhitekturi instrukcijskog seta na kome se izvršava program P čiji je *dissassembly* asemblerskog koda predstavljen sa *Dissassembly 4.1*. Memorija sistema je povezana sa procesorom koristeći 32bitnu magistralu sa odvojenim linijama za podatke i adrese. Inicijalni sadržaj dela memorije namenske platforme dat je u tabeli 4.1. Sadržaj nekih memorijskih lokacija na kojima su smeštene instrukcije date sa *Dissassembly 4.1* nije poznat i označen je sa x.

Dissassembly 4.1

0x00:addi x1, x0, 64	0x34:addi x1, x1, 4
0x04:addi x2, x0, 170	0x38:sw x3, 0(x1)
0x08:slli x3, x2, 24	0x3C:jal x0, 0
0x0C:addi x2, x0, 0xBB	
0x10:slli x2, x2, 16	
0x14:or x3, x2, x3	
0x18:addi x2, x0, 204	
0x1C:slli x2, x2, 8	
0x20:or x3, x3, x2	
0x24:addi x2, x0, 221	
0x28:or x3, x3, x2	
0x2C:lw x2, 0(x1)	
0x30:xor x3, x3, x2	

Tabela 4.1

Adresa	Sadržaj				Adresa	Sadržaj			
0x00	93	00	00	04	0x28	b3	e1	21	00
0x04	<u>x</u>	<u>x</u>	<u>x</u>	<u>x</u>	0x2c	<u>x</u>	<u>x</u>	<u>x</u>	<u>x</u>
0x08	93	11	81	01	0x30	b3	c1	21	00
0x0c	13	01	b0	0b	0x34	<u>x</u>	<u>x</u>	<u>x</u>	<u>x</u>
0x10	<u>x</u>	<u>x</u>	<u>x</u>	<u>x</u>	0x38	23	a0	30	00
0x14	<u>x</u>	<u>x</u>	<u>x</u>	<u>x</u>	0x3c	6f	00	00	00
0x18	13	01	c0	0c	0x40	ff	ff	ff	ff
0x1c	<u>x</u>	<u>x</u>	<u>x</u>	<u>x</u>	0x44	00	00	00	00
0x20	b3	e1	21	00	0x48	00	00	00	00
0x24	<u>x</u>	<u>x</u>	<u>x</u>	<u>x</u>	0x4c	00	00	00	00

Ako je nakon sistemskog reseta, PC registar CPUa inicijalizovan na vrednost 0:

- a) [12] popuniti tabelu 4.2 za svaku od instrukcija definisanih sa *Dissassembly 4.1*. Koje vrednosti se nalaze u memoriji na kolacijama 0x44-0x47 nakon izvršenja programa P?
- b) [8] Koristeći tabelu 4.3 odrediti sadržaj nepoznatih memorijskih lokacija.

Napomena: Sve adrese i vrednosti date u tabeli 4.1 su predstavljene u heksadecimalnom brojnem sistemu. Ukoliko ispred brojnih vrednosti **operanada** instrukcija definisanih u okviru *Dissassembly 4.1* postoji prefiks 0x smatrati da su te brojne vrednosti date u heksadecimalnom brojnem sistemu dok se u suprotnom može smatrati da su vrednosti date u decimalnom brojnem sistemu. Izvršavanje programa se analizira dok se instrukcija sa iste memorijske lokacije ne izvrši uzastopno više od 2 puta.

2. KOLOKVIJUM

5. Vrednosti parametara hijerarhijski organizovanog memorijskog dela sistema, sa jednim stepenom hijerarhije, su:

- kapacitet glavne memorije (MC) = 256B;
- vreme pristupa glavnoj memoriji ($T_{Penalty}$) = $200T_{CLK}$;
- veličina bloka u kešu (BS) = 4B;
- adresibilna jedinica (AUS) = 1B;
- kapacitet keš memorije (CC) = 16B;
- vreme pristupa keš memoriji (T_{HIT}) = $4T_{CLK}$;
- keš memorija je organizovana kao **direktno mapirani keš** čiji kontroler primenjuje *write back* – *write allocate* polisu upisa;
- keš memorija je integrisana u sistem koristeći *look through* topologiju
- inicijalni sadržaj glavne memorije definisan je tabelom 4.1.

Za program **P** koji se izvršava na ovoj namenskoj platformi poznato je da sekvencijalno pristupa sledećim adresama (**R** u indeksu označava čitanje sa memorijske lokacije definisane u uglastim zagradama dok **W** u indeksu označava upis podatka sa desne strane znaka = na memorijsku lokaciju u uglastim zagradama):

$$M_R[0x19], M_R[0x0c], M_R[26], M_W[3] = 0xAB, M_W[0x1F] = 0xBC, M_W[0] = 0xCD, M_W[45] = 0xDA.$$

a) [10] U tabeli 5.2 najpre kreirati okvir tabele koji ilustruje organizaciju keša a zatim predstaviti sadržaj kreirane tabele za svaki od adresnih ciklusa generisanih od strane procesora. U tabeli 5.1 predstaviti sadržaj dela glavne memorije nakon završetka transakcije na magistrali.

b) [3] Koliko iznosi *hit rate*? Izračunati koliko iznosi AMAT.

c) [3] Ukoliko je magistrala za podatke široka 32bita, koliko iznosi broj ciklusa (N_{WBWA}) na magistrali između keš kontrolera i glavne memorije? Koliko iznosi broj ciklusa (N_{WBWNA}) na magistrali između keš kontrolera i glavne memorije ukoliko se umesto *write back* – *write allocate* polise upisa koristi *write through* – *no write allocates* polisa upisa? Obrazložiti odgovor!

d) [4] Ukoliko se hijerarhijski organizovana memorija iz tačke a) proširi sa još dva dodatna stepena hijerarhije (L2 i L3) izračunati AMAT u tom slučaju ukoliko je poznato: vreme pristupa keš memoriji na L2 nivou (T_{HIT-L2}) = $5T_{CLK}$; *miss rate* na L2 nivou ($r_{MISS-L2}$) = 10%; vreme pristupa keš memoriji na L3 nivou (T_{HIT-L3}) = $23T_{CLK}$; *global miss rate* na L3 nivou ($r_{MISS-GLOBAL-L3}$) = 0.3%.

Napomena: Ukoliko ispred brojnih vrednosti postoji prefiks 0x smatrati da su te brojne vrednosti date u heksadecimalnom brojnem sistemu dok se u suprotnom može smatrati da su vrednosti date u decimalnom brojnem sistemu. Ukoliko je potrebno izvršiti zamenu bloka u keš memoriji, iz keš memorije se izbacuje onaj blok koji je prvi dodat u keš memoriju.

6. [10] Napisati kratak asemblerski program koji ostvaruje funkcionalnost:

```
int i;
int temp[100];

for (i = 0; i < 100; i = i + 1)
    temp[i] = temp[i] * 128;
```

Podrazumeva se da je niz temp inicijalizovan i da se u registru t3 nalazi bazna adresa niza. Int je 16 bitni podatak.

7. [10] Deo koda napisan u višem programskom jeziku napisati u assembleru RISC V. Podrazumevatrati da su označani brojevi button i amt u registrima s0 i s1. Jasno komentarisati kod.

```
switch (button) {
    case 1: amt = 20; break;
    case 2: amt = 50; break;
    case 3: amt = 100; break;
    default: amt = 0;
}
```

8. [10] Opisati hazarde koji mogu da nastanu prilikom izvođenja procesorskih instrukcija sa *pipeline*-om, kao i načine njihovog razrešenja.