

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]

a) [4] NOR instrukcija nije deo RISC V instrukcijskog seta. Istu funkcionalnost moguće je ostvariti sa postojećim instrukcijama. Napisati kratak asemblerski program koji ostvaruje funkcionalnost:

$$s3 = s4 \text{ NOR } s5$$

b) Deo koda napisan u višem programskom jeziku napisati u assembleru RISC V. Podrazumevati da su označani brojevi g i h u registrima s0 i s1. Jasno komentarisati kod.

b1) [3] if (g > h)
 g = g + 1;
 else
 h = h - 1;

b2) [3] if (g <= h)
 g = 0;
 else
 h = 0;

2. [20] Nacrtati realizaciju dela jednociklusnog RISC V procesora koji izvodi sledeću instrukciju.

Address	Instruction	Type	Fields					Machine Language	
0x1004	sw x6, 8(x9)	S	imm ₁₁ s	rs2	rs1	fb	imm ₄ ,o	op	0064A423
	[x9] = 0x 0000 2004		0000000	00110	01001	010	01000		
	[0x2000] = 0x 0000 000A								

Definisati potrebne signale koje treba da generiše kontrolna jedinica kao i njihov vremenski redosled i razmak.

3. [20] Namenski sistem koristi procesor baziran na 32bitnoj RISC-V arhitekturi instrukcijskog seta. Poznato je da je memorija povezana sa procesorom preko 32bitne magistrale koja ima odvojene putanje za podatke i za adrese. Inicijalni sadržaj dela memorije namenske platforme dat je u tabeli 3.1. Nakon dekodovanja sadržaja dela memorije uspešno su dekodovane neke asemblerske instrukcije predstavljene u okviru *Dissassembly 3.1*.

Dissassembly 3.1

```

0x00: addi x2, x0, 48
0x04: -----
0x08: addi x3, x0, 4
0x0C: -----
0x10: addi x5, x4, 1
0x14: -----
0x18: addi x1, x1, 1
0x1C: addi x2, x2, -1
0x20: addi x3, x3, -1
0x24: bne x3, x0, -24
0x28: -----

```

Tabela 3.1

Adresa	Sadržaj				Adresa	Sadržaj			
0x00	0x93	0x00	0x00	0x03	0x20	0x93	0x81	0xf1	0xff
0x04	0x13	0x01	0x70	0x03	0x24	0xe3	0x94	0x01	0xfe
0x08	0x93	0x01	0x40	0x00	0x28	0x6f	0x00	0x00	0x00
0x0C	0x03	0x82	0x00	0x00	0x2C	0x00	0x00	0x00	0x00
0x10	0x93	0x42	0xf2	0x0f	0x30	0x0a	0x0b	0x0c	0x0d
0x14	0x23	0x00	0x51	0x00					
0x18	0x93	0x80	0x10	0x00					
0x1C	0x13	0x01	0xf1	0xff					
					...				

Ako je nakon sistemskog reseta, PC registar CPUa inicijalizovan na vrednost 0, popuniti tabele 3.2, 3.3 i 3.4 (pogledati formular za odgovore) za svaku od faza izvršavanja instrukcije.

Napomena: Sve adrese i vrednosti date u tabeli 3.1 su predstavljene u heksadecimalnom brojnom sistemu. Ukoliko ispred brojnih vrednosti **operanada** instrukcija definisanih u okviru *Dissassembly 3.1* postoji prefiks 0x smatrati da su te brojne vrednosti date u heksadecimalnom brojnom sistemu dok se u suprotnom može smatrati da su vrednosti date u decimalnom brojnom sistemu. Izvršavanje programa se analizira dok se instrukcija sa iste memorijske lokacije ne izvrši uzastopno dva puta. Broj redova u okviru tabela datih na formularima za odgovore je **proizvoljan** i **ne mora** odgovarati broju koraka potrebnih za potpunu analizu izvršavanja programa.

2. KOLOKVIJUM

4. Vrednosti parametara hijerarhijski organizovanog memorijskog dela sistema, sa jednim stepenom hijerarhije, su:
- kapacitet glavne memorije (MC) = 256B;
 - vreme pristupa glavnoj memoriji ($T_{Penalty}$) = $180T_{CLK}$;
 - veličina bloka u kešu (BS) = 8B;
 - adresibilna jedinica (AUS) = 1B;
 - kapacitet keš memorije (CC) = 32B;
 - vreme pristupa keš memoriji (T_{HIT}) = $5T_{CLK}$;
 - keš memorija je organizovana kao **2 way set asocijativni keš** čiji kontroler primenjuje *write back – write allocate* politiku upisa;
 - keš memorija je integrisana u sistem koristeći *look through* topologiju
 - inicijalni sadržaj glavne memorije definisan je tabelom 3.1. Smatrati da su memorijske lokacije, koje su u tabeli 3.1 označene sa --, inicijalizovane na vrednosti 0xFF.

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):

$$M_w[0x10]=0x02, M_r[0x013], M_w[10]=0x03, M_w[9] = 0x18, M_r[0x05], M_r[0x11], M_w[17] = 0xCD, M_w[0x10] = 0x04.$$

- a) [12] U tabeli 4.2 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 4.1 predstaviti sadržaj dela glavne memorije nakon završetka transakcije na magistrali.
- b) [4] Koliko iznosi *hit rate*? Izračunati koliko iznosi AMAT.
- c) [4] Pod pretpostavkom da je magistrala između procesora i keš memorije, kao keš memorije i glavne memorije, širine 2B (sistemska magistrala), koliko zahteva se generiše između procesora i keš memorije (NCC\$) a koliko između keš memorije i glavne memorije (NC\$MM)?

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 dodan u keš memoriju.

-
5. [15] Definirati vremenski dijagram rada sinhrono 16bitne magistrale (koja ima upravljanje bajtovima i mogućnost upisa na neporavnate adrese) kada se upisuje 16bitni podatak na neparnu adresu. Definirati sve vremenske parametre koji su bitni za projektanta sistema.

-
6. [15] Nacrtati opštenamenski prekidni kontroler sa 8 ulaza koji prekide sa aktivnim usponskom ili silaznom ivicom, što može da se programira. Takođe obezbediti mogućnost da se posle prihvaćenog prekida automatski resetuje prekidni zahtev ili se resetuje od strane procesora što može da se programira. Napomena; maskiranje prekidnih zahteva je uvek obavezno u kontroleru prekida.