

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. Data je rekurzivna funkcija (ponovo ista iz januara posto studenti nisu sagledali suštinu, pa je sada u drugačijoj formi)

```
int fib(int n) {  
    int a,b,c;  
    if (n < 2) return n;  
    a= fib(n-1);  
    b= fib(n-2);  
    c=a+b;  
    return c;  
}
```

a) [9] Napisati datu funkciju korišćenjem asemblerskog jezika za RISC V procesor i koristeći standarde za prenos argumenta preko registara, čuvanja, odnosno korišćenje registara procesora. Jasno komentarisati kod. Podrazumevati da je int promenljiva dužine 32 bita.

b) [6] Da li je moguće izvesti funkciju bez posledica za bilo koju vrednost n? Koje su eventualne posledice? Ako postoje modifikovati asemblerski kod da do njih ne dolazi (nisu potrebne apsolutno tačne vrednosti nekih konstanti, ako su potrebne). Ako je za stek procesora dozvoljeno maksimalno koristiti 4KiB, koja je maksimalna vrednost za promenljivu n?

2.a) [3] Kodovati instrukciju RISC V procesora

sw x5,16(x3)

Jasno, tabelarno, prikazati sve bite u instrukciji, kao i njihovo značenje. Na primer

.....

...	...	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
...	...	rd						OP					
...	...	0	0	1	0	1	0	0	1	0	0	1	1

b) [12] Nacrtati realizaciju dela višeciklusnog RISC V procesora koji izvodi instrukciju iz tačke a). Označiti sve signale na šemi. Jasno definisati (i objasniti šta rade) potrebne signale koje treba da generiše kontrolna jedinica, kao i njihov vremenski redosled.

3. [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 3.1*. 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 3.1. Sadržaj nekih memorijskih lokacija na kojima su smeštene instrukcije date sa *Dissassembly 3.1* nije poznat i označen je sa x.

Dissassembly 3.1

```
0x00: -----  
0x04: addi x2, x0, 55  
0x08: -----  
0x0C: lb x4, 0(x1)  
0x10: addi x5, x4, 1  
0x14: -----  
0x18: addi x1, x1, 1  
0x1C: addi x2, x2, -1  
0x20: addi x3, x3, -1  
0x24: -----  
0x28: jal x0, 0
```

Tabela 3.1

Adresa	Sadržaj				Adresa	Sadržaj			
0x00	93	00	00	03	0x28	6f	00	00	00
0x04	13	01	70	03	0x2c	00	00	00	00
0x08	93	01	40	00	0x30	AA	BB	CC	DD
0x0C	03	82	00	00	0x34	00	00	00	00
0x10	93	02	12	00	0x38	00	00	00	00
0x14	23	00	51	00	0x3c	6f	00	00	00
0x18	93	80	10	00	0x40	ff	ff	ff	ff
0x1C	13	01	f1	ff	0x44	ff	ff	ff	ff
0x20	93	81	f1	ff	0x48	ff	ff	ff	ff
0x24	e3	94	01	fe	0x4c	ff	ff	ff	ff

Ako je nakon sistemskog reseta, PC registar CPUa inicijalizovan na vrednost 0 popuniti tabele 3.2, 3.3 i 3.4 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. [20] Vrednosti parametara hijerarhijski organizovanog memorijskog dela sistema, sa jednim stepenom hijerarhije, su:

- kapacitet glavne memorije (MC) = 1024B;
- vreme pristupa glavnoj memoriji ($T_{Penalty}$) = $120T_{CLK}$;
- veličina bloka u kešu (BS) = 16B;
- adresibilna jedinica (AUS) = 1B;
- kapacitet keš memorije (CC) = 64B;
- vreme pristupa keš memoriji (T_{HIT}) = $5T_{CLK}$;
- keš memorija je organizovana kao **direktno mapirani keš** čiji kontroler primenjuje *write back – write no allocate* politiku upisa;
- keš memorija je integrisana u sistem koristeći *look through* topologiju
- inicijalni sadržaj glavne memorije definisan je tabelom 3.1.

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):

$M_R[0x03]$, $M_W[0x0D] = 0xA2$, $M_R[0x33]$, $M_W[0x3B] = 0xC4$, $M_R[0x43]$, $M_R[0x0B]$, $M_W[0x3D] = 0xE1$, $M_R[0x47]$.

a) [10] 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 ukoliko dolazi do promene sadržaja memorije.

b) [1] Koliko iznosi *hit rate*? Izračunati koliko iznosi AMAT.

c) [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}) = $10T_{CLK}$; *miss rate* na L2 nivou ($r_{MISS-L2}$) = 5%; vreme pristupa keš memoriji na L3 nivou (T_{HIT-L3}) = $20T_{CLK}$; *global miss rate* na L3 nivou ($r_{MISS GLOBAL-L3}$) = 0.5%.

d) [5] Ukoliko je širina magistrale između procesora i keš memorije jednaka 8B, a između keš memorije i glavne memorije 4B, koliko iznosi ukupan broj generisanih transakcija na svakoj od magistrala (NT\$ - broj transakcija između procesora i keš memorije; NT\$MM – broj transakcija između keš memorije i glavne memorije)? Prodiskutovati mogućnost smanjenja ukupnog broja generisanih transakcija između keš memorije i glavne memorije.

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.

5. [15] Dati su vremenski dijagrami rada jedne sinhronne paralelne magistrale, kao i vremenski parametri.

Izračunati:

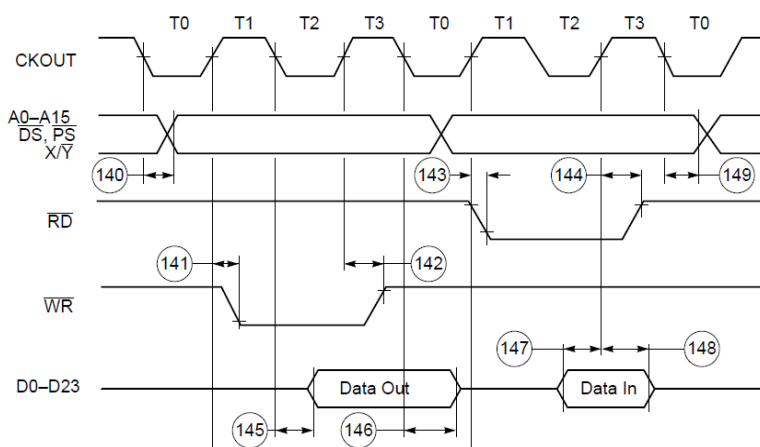
a) [8] maksimalno dozvoljena setap i hold vremena prihvatnog registra u koji se upisuje podatak usponskom ivicom signala upisa.

b) [7] maksimalno dozvoljeno vreme otvaranja trostatičkih bafera u ciklusu čitanja.

Rezultate prikazati sa brojevnim oznakama sa dijagrama i za svaki broj naglasiti da li se uzima minimalno ili maksimalno vreme.

Na primer

... = 143_{min}^+ ...



6. [15] Nacrtati opštenamenski prekidni kontroler sa 8 ulaza koji može da prihvati prekide sa aktivnom usponskom ili silaznom ivicom što može da se programira. Posle svakog prihvaćenog prekida zahtev za prekidom se automatski resetuje. **NAPOMENA:** Procesor treba da ima mogućnost čitanja i da li postoji prekid, bez obzira da li je maskiran ili ne, i da li je sa aktivnom usponskom ili silaznom ivicom i koji prekidi su maskirani.