
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Multiprocesorski sistemi
(13E114MUPS)
Nastavnik: dr Milo Tomašević, red. prof.
Asistent: dr Marko Mišić, dipl. ing.
Ispitni rok: Jul 2 2017.
Datum: 26.06.2017.

Kandidat:* _____

Broj Indeksa:* _____

*Ispit traje 180 minuta, prvih sat vremena nije dozvoljeno napuštanje ispita.
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /5	<i>Zadatak 6</i>	_____ /10
<i>Zadatak 2</i>	_____ /10	<i>Zadatak 7</i>	_____ /10
<i>Zadatak 3</i>	_____ /10	<i>Zadatak 8</i>	_____ /10
<i>Zadatak 4</i>	_____ /10	<i>Zadatak 9</i>	_____ /15
<i>Zadatak 5</i>	_____ /10	<i>Zadatak 10</i>	_____ /10

Ukupno na ispitu: _____ /100

Napomena: Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**.

* popunjava student.

1. [5] Objasniti tehnološke trendove u oblasti memorija.

2. [10] Objasniti *dataflow* programski model i osnovne karakteristike odgovarajuće arhitekture.

3. **[10]** Nacrtati dijagram stanja i objasniti akcije MSI protokola.

4. **[10]** Uporedno prikazati prednosti i nedostatke strategija poništavanja i ažuriranja u protokolima. Koji je osnovni kriterijum za izbor u pogledu osobina aplikacije?

5. **[10]** Objasniti organizaciju kataloga i opisati akcije protokola sa dinamičkim pointerima *Dir* *DP*.

6. **[10]** Opisati transakcije koje se u oba smera obavljaju između dva nivoa keš memorije u protokolu koji koristi inkluziju.

7. [10] Objasniti kakve su i šta rade MPI rutine *MPI_Reduce* and *MPI_Allreduce* i navesti razliku između njih. Napisati deo koda koji korišćenjem *MPI_Reduce* i dodatnog koda simulira ponašanje *MPI_Allreduce* rutine.

8. [10] Kada i kako se može koristiti sinhronizacija na barijeri kod OpenMP programskog modela? Na kojim mestima se nalazi implicitna barijera?

9. [15] Korišćenjem CUDA tehnologije paralelizovati funkciju u prilogu koja predstavlja jedan korak simulacije poznate igre *Game of Life*. Koristiti 2D organizaciju jezgra. Obratiti pažnju na efikasnost paralelizacije.

```
void evolve(void *u, int w, int h) {
    unsigned (*univ)[w] = u, new[h][w];
    for (int x = 0; x < w; x++)
        for (int y = 0; y < h; y++) { int n = 0;
            for (int y1 = y - 1; y1 <= y + 1; y1++)
                for (int x1 = x - 1; x1 <= x + 1; x1++)
                    if (univ[(y1 + h) % h][(x1 + w) % w]) n++;
            if (univ[y][x]) n--;
            new[y][x] = (n == 3 || (n == 2 && univ[y][x]));
        }
    for (int x = 0; x < w; x++)
        for (int y = 0; y < h; y++) univ[y][x] = new[y][x];
}
```

10. [10] Dat je multiprocesorski sistem sa 4 identična procesora, koji koristi WTI *write-no-allocate* protokol za održavanje koherencije keš memorije. Svaka keš memorija ima po 2 ulaza, koji su veličine jedne reči. Preslikavanje je **asocijativno** Početne vrednosti podataka su 0. Svaki upis uvećava vrednost izmenjenog podatka za 1. Na početku su sve keš memorije prazne. Data je sledeća sekvenca pristupa memoriji:

1. P0,R,A0	3. P2,W,A0	5. P1,W,A2	7. P0,R,A0
2. P0,W,A2	4. P1,R,A2	6. P1,W,A1	8. P0,W,A1

Napisati stanja koherencije u svim procesorima i stanje memorije posle svake promene i skicirati opisani sistem u trenutku 8. [8 poena]

Da li procesori pristupaju memoriji i kada? Za svaki pristup navesti razlog. [2 poena]

Trenutak 1												Memorija	
P0			P1			P2			P3			A0	

Pristupi memoriji:

Trenutak 2												Memorija	
P0			P1			P2			P3			A0	

Pristupi memoriji:

Trenutak 3												Memorija	
P0			P1			P2			P3			A0	

Pristupi memoriji:

Trenutak 4												Memorija	
P0			P1			P2			P3			A0	

Pristupi memoriji:

Trenutak 5

P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:

Trenutak 6

P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:

Trenutak 7

P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:

Trenutak 8

P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:
