

---

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Multiprocesorski sistemi (SI4MPS)  
*Nastavnik:* dr Milo Tomašević, vanr. prof.  
*Asistent:* dipl. ing. Marko Mišić  
*Ispitni rok:* Treći kolokvijum (januar 2014.)  
*Datum:* 14.01.2014.

*Kandidat\*:* \_\_\_\_\_

*Broj Indeksa\*:* \_\_\_\_\_

*Kolokvijum traje 105 minuta, prvih sat vremena nije dozvoljeno napuštanje kolokvijuma.  
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /15	<i>Zadatak 5</i>	_____ /10
<i>Zadatak 2</i>	_____ /10	<i>Zadatak 6</i>	_____ /15
<i>Zadatak 3</i>	_____ /20	<i>Zadatak 7</i>	_____ /15
<i>Zadatak 4</i>	_____ /15		

**Ukupno na kolokvijumu:** \_\_\_\_\_ /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 ponudene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**.

\* popunjava student.

1. [15] Objasniti o čemu mora da se vodi računa pri razvoju paralelnog softvera da bi se smanjio *overhead* pri održanju koherencije i poboljšale performanse.

2. [10] Objasniti zašto primena isključivo invalidacione ili ažurirajuće strategije najčešće nije optimalno rešenje. Objasniti kako se to prevazilazi.

3. [20] Šta se postiže primenom “*cache-based*” protokola? Objasniti organizaciju kataloga i izračunati njegovu veličinu ako je  $m$  – veličina memorije,  $c$  – veličina keš memorije po procesoru,  $b$  - veličina bloka i  $n$  – broj procesora. Precizno opisati šta se dešava kod *read miss*-a i *write hit*-a.

4. [15] a) Definisati pojam inkluzije u keš hijerarhiji. Objasniti šta se dobija primenom inkluzije. b) U slučaju da se održava inkluzija, objasniti koje sve se transakcije i kako obavljaju između dva nivoa keš memorije.

5. [15] Šta predstavlja *loop unrolling* tehnika za optimizaciju petlji? Na primeru `reductionSum` jezgra pokazati kako ona može da doprinese poboljšanju performansi koda koji se izvršava na grafičkom procesoru.

```
__global__ void reductionSum (int* devA, int* blockResults, int n) {
    extern __shared__ int sharedData[];
    unsigned int tid = threadIdx.x;
    unsigned int i = blockIdx.x * blockDim.x + threadIdx.x;
    if (i < n) sharedData[tid] = devA[i];
    else sharedData[tid] = 0;
    __syncthreads();
    for (unsigned int s = blockDim.x/2; s > 0; s >>= 1) {
        if (tid < s) {
            sharedData[tid] += sharedData[tid + s];
        }
        __syncthreads();
    }
    if (tid == 0) blockResults[blockIdx.x] = sharedData[0];
}
```

6. [10] Na koji način se na grafičkom procesoru sakrivaju kašnjenja koja nastaju prilikom pristupa sporoj, globalnoj memoriji uređaja? Koja razlika tu postoji u odnosu na centralni procesor?

7. [15] Napisati jezgro CUDA programa koje vrši obradu nad dvodimenzionalnom kvadratnom matricom realnih brojeva. Jezgro treba da formira novu matricu na osnovu postojeće tako da svaki element novoformirane matrice ima vrednost:

$$Out_{x,y} = C_1 \cdot In_{x,y} + C_2 \cdot (In_{x-1,y} + In_{x+1,y} + In_{x,y-1} + In_{x,y+1}) / 4$$

Smatrati da svaki element matrice ima najviše četiri suseda. Smatrati da su matrice alocirane unapred. Koristiti 2D organizaciju jezgra. Prilikom rešavanja zadatka koristiti deljenu memoriju za smeštanje međurezultata i voditi računa da se ostvari maksimalan paralelizam.

```
__global__ void filter (float* in, float* out, int n, float C1, float C2);
```