

---

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Multiprocesorski sistemi (13S114MUPS)

*Nastavnik:* dr Milo Tomašević, red. prof.

*Asistent:* doc. dr Marko Mišić

*Ispitni rok:* Treći kolokvijum (januar 2019.)

*Datum:* 22.01.2019.

*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>	_____ /15
<i>Zadatak 2</i>	_____ /15	<i>Zadatak 6</i>	_____ /10
<i>Zadatak 3</i>	_____ /15	<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 ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**.

\* popunjava student.

---

1. [15] Objasniti strukturu kataloga, funkcionisanje, prednosti i nedostatke *Dir; B* protokola.

2. [15] Objasniti motivaciju za tehniku smanjivanja visine u *directory* protokolima. Ukratko objasniti funkcionisanje ove tehnike, prednosti i nedostatke.

3. [15] Nacrtati i objasniti strukturu dvonivoskog hijerarhijskog sistema sa distribuiranom memorijom. Ukratko objasniti održavanje koherencije.

4. [15] Šta su direktne interkonekzione mreže? Definisati najvažnije parametre ovih mreža i njihove poželjne vrednosti.

5. [15] Zadato CUDA jezgro predstavlja jednu varijantu redukcije na nivou bloka. Detaljno objasniti odsustvo sinhronizacije u poslednjih nekoliko koraka i na koji način se taj mehanizam sprovodi.

```
template <unsigned int blockSize>
__global__ void reduce(int *g_idata, int *g_odata, unsigned int n) {
    extern __shared__ int sdata[];
    unsigned int tid = threadIdx.x;
    unsigned int i = blockIdx.x*(blockSize*2) + tid;
    unsigned int gridSize = blockSize*2*gridDim.x;
    sdata[tid] = 0;
    while (i < n) {
        sdata[tid] += g_idata[i] + g_idata[i+blockSize];
        i += gridSize;
    }
    __syncthreads();
    if (blockSize >= 512) { if (tid < 256) {
        sdata[tid] += sdata[tid + 256]; } __syncthreads(); }
    if (blockSize >= 256) { if (tid < 128) {
        sdata[tid] += sdata[tid + 128]; } __syncthreads(); }
    if (blockSize >= 128) { if (tid < 64) {
        sdata[tid] += sdata[tid + 64]; } __syncthreads(); }
    if (tid < 32) {
        if (blockSize >= 64) sdata[tid] += sdata[tid + 32];
        if (blockSize >= 32) sdata[tid] += sdata[tid + 16];
        if (blockSize >= 16) sdata[tid] += sdata[tid + 8];
        if (blockSize >= 8) sdata[tid] += sdata[tid + 4];
        if (blockSize >= 4) sdata[tid] += sdata[tid + 2];
        if (blockSize >= 2) sdata[tid] += sdata[tid + 1];
    }
    if (tid == 0) g_odata[blockIdx.x] = sdata[0];
}
```

6. **[10]** Na koji način se vrši alokacija registara prilikom izvršavanja koda na grafičkom procesoru i kako to utiče na performanse izvršavanja koda?

7. [15] Koristeći CUDA tehnologiju paralelizovati funkciju koja računa procentualnu razliku dve slike istih dimenzija. Slika je definisana kao niz čija svaka tri uzastopna elementa predstavljaju RGB komponente slike. Obratiti pažnju na efikasnost paralelizacije i koristiti deljenu memoriju. Napisati deo koda za centralni procesor koji vrši pozivanje implementiranog jezgra.

```
#define RED_C 0
#define GREEN_C 1
#define BLUE_C 2
#define GET_PIXEL(IMG, X, Y) ((IMG)->buf[ (Y) * (IMG)->width + (X) ])
...
image im1, im2, totalDiff = 0.0;
unsigned int x, y;
...
for(x=0; x < im1->width; x++) {
    for(y=0; y < im1->width; y++) {
        totalDiff += fabs(GET_PIXEL(im1, x, y)[RED_C] -
                           GET_PIXEL(im2, x, y)[RED_C] ) / 255.0;
        totalDiff += fabs(GET_PIXEL(im1, x, y)[GREEN_C] -
                           GET_PIXEL(im2, x, y)[GREEN_C] ) / 255.0;
        totalDiff += fabs(GET_PIXEL(im1, x, y)[BLUE_C] -
                           GET_PIXEL(im2, x, y)[BLUE_C] ) / 255.0;
    }
}
```