
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Multiprocesorski sistemi (13S114MUPS)
Nastavnik: dr Milo Tomašević, red. prof.
Asistenti: doc. dr Marko Mišić; Pavle Divović, dipl.ing.
Ispitni rok: Treći kolokvijum (januar 2023.)
Datum: 17.01.2023.

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] Opisati šta se dešava pri izbacivanju nekog bloka iz keš memorije kod *directory* protokola. Posebno diskutovati alternative pri izbacivanju bloka čiji je sadržaj isti kao u memoriji. Identifikovati situacije i protokole za koje su pojedine alternative pogodne.

2. [15] Na kojoj činjenici su zasnovane metode za smanjivanje visine kataloga u *directory* protokolima? Objasniti kako izgleda struktura koja se koristi za katalog i opisati operacije za alokaciju i dealokaciju ulaza. Kakve su specifičnosti ove strukture u odnosu na uobičajeno korišćenje u procesoru?

3. [15] Objasniti princip organizacije višestepene sprežne mreže tipa *Omega*, kao i način rutiranja poruka. Objasniti da li je mreža blokirajuća. Ako jeste, nacrtati u mreži 8x8 primer rutiranja dve poruke koje izazivaju blokiranje i označiti mesto gde dolazi do blokiranja.
4. [15] Kojoj grupi interkonekcionih mreža pripada mreža tipa stabla? Objasniti strukturu ove mreže i nacrtati je za $n = 8$. Napisati izraze za vrednosti parametara mreže u opštem slučaju. Koji je njen glavni problem i kako se prevazilazi?

5. [15] Šta je to *burst* sekcija i kako ona utiče na pristup memoriji kod grafičkih procesora? Na primeru dela koda u prilogu koji vrši množenje matrice, navesti i objasniti kod kojih pristupa se koriste prednosti pristupa u transakciji, a kod kojih ne. Odgovor ilustrovati slikom.

```
__global__ void MatrixMulKernel
(float* Md, float* Nd, float* Pd, int Width) {
    int Row = blockIdx.y * blockDim.y + threadIdx.y;
    int Col = blockIdx.x * blockDim.x + threadIdx.x;
    float Pvalue = 0;
    for (int k = 0; k < Width; ++k)
        Pvalue += Md[Row * Width + k] * Nd[k * Width + Col];
    Pd[Row * Width + Col] = Pvalue;
}
```

6. [10] Nacrtati i objasniti tipičnu redukcionu šemu na grafičkom procesoru. Navesti broj koraka i operacija koji se pritom izvršava i uporediti sa klasičnom sekvencijalnom redukcijom. Zašto sekvencijalna redukcija nije pogodna na grafičkom procesoru?

7. [15] Koristeći CUDA tehnologiju paralelizovati funkciju koja računa vrednosti gradijenta zadate slike. Koristiti 2D organizaciju jezgra. Obratiti pažnju na efikasnost paralelizacije i koristiti deljenu memoriju. Smatrati da su sve promenljive inicijalizovane.

```
enum orientation : unsigned char {
    None, Vertical, Horizontal
}
double *grad ( size_t rows, size_t cols, int img[],
               short gradImg[], char dirImg[] ) {
    for (size_t i = 1; i < rows - 1; i++) {
        for (size_t j = 1; j < cols - 1; j++) {
            size_t index = i * cols + j;

            int com1 = img[index + cols + 1] - img[index - cols - 1];
            int com2 = img[index - cols + 1] - img[index + cols - 1];

            int gx = com1 + com2 + img[index + 1] - img[index - 1];
            int gy = com1 - com2 + img[index + cols] - img[index - cols];

            int sum = hypot(gx, gy);

            gradImg[index] = sum;
            if (sum >= THRESHOLD) {
                dirImg[index] = abs(gx) >= abs(gy) ? Vertical : Horizontal;
            }
        }
    }
}
```

