
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Multiprocesorski sistemi (13S114MUPS)

Nastavnik: dr Marko Mišić, vanr. prof

Asistent: Matija Dodović, mast. ing.

Ispitni rok: Treći kolokvijum (februar 2026.)

Datum: 14.02.2026.

Kandidat:* _____

Broj Indeksa:* _____

*Kolokvijum traje 90 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 i šematski prikazati jedan *streaming multiprocessor* u okviru grafičkog procesora. Na koji način variranje broja SM-ova u okviru različitih grafičkih procesora omogućava skalabilno izvršavanje blokova niti?

2. [15] Nacrtati i objasniti direktnu interkonekcionu mrežu tipa prstena sa 6 čvorova. Diskutovati osnovne parametre ovakve mreže.

3. [15] Šta predstavlja *warp* niti i koja je uloga *warp*-ova prilikom sakrivanja kašnjenja kod pristupa globalnoj memoriji grafičkog procesora?
4. [15] Nacrtati i objasniti paralelnu memorijsku arhitekturu globalne memorije grafičkog procesora. Koje tehnike u implementaciji memorijske arhitekture se koriste radi povećanja propusnog opsega?

5. [15] Posmatra se jezgro u prilogu koje vrši obrtanje redosleda elemenata zadatog jednodimenzionalnog niza na grafičkom procesoru.

```
__global__ void reverseArrayBlock(int *d_out, int *d_in){
    extern __shared__ int s_data[];
    int inOffset = blockDim.x * blockIdx.x;
    int in = inOffset + threadIdx.x;
    s_data[blockDim.x - 1 - threadIdx.x] = d_in[in];
    __syncthreads();
    int outOffset = blockDim.x * (gridDim.x - 1 - blockIdx.x);
    int out = outOffset + threadIdx.x;
    d_out[out] = s_data[threadIdx.x];
}
```

- a) [5] Odrediti i komentarisati aritmetički intenzitet ove operacije. Da li je jezgro *compute-bound* ili *memory-bound*?
- b) [5] Komentarisati upotrebu deljene memorije u kontekstu povećanja efikasnosti ovog jezgra.
- c) [5] Ukoliko se jezgro poziva za niz dužine 100000 elemenata na GPU arhitekturi *Ada Lovelace* (najviše 1024 niti po bloku, najviše 1536 niti po SM, najviše 32 bloka na SM), odrediti i objasniti optimalnu izvršnu konfiguraciju za poziv jezgra i napisati odgovarajući poziv.

6. [10] Šematski prikazati i objasniti ulogu *tiling* tehnike prilikom implementacije jezgra za množenje matrica na grafičkom procesoru.

7. [15] Koristeći CUDA tehnologiju, paralelizovati deo koda u prilogu koji vrši određenu obradu nad nizom tačaka u prostoru. Svaka tačka je opisana strukturom koja sadrži tri realna polja (x, y i z koordinate tačke). Jezgro treba da formira niz realnih brojeva koji sadrži rastojanja između svake tačke u nizu tačaka i zadate referentne tačke. Obratiti pažnju na efikasnost paralelizacije i koristiti deljenu memoriju, gde je moguće i donosi korist. Smatrati da su sve promenljive inicijalizovane, a memorija alocirana.

```
#include <math.h>
typedef struct {
    float x, y, z;
} Point;
void distance(Point *points, float *dist, Point ref, int n)
{
    for (int i = 0; i < n; i++) {
        float dx = points[i].x - ref.x;
        float dy = points[i].y - ref.y;
        float dz = points[i].z - ref.z;

        dist[i] = sqrtf(dx*dx + dy*dy + dz*dz);
    }
}
```

