
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 2018.)

Datum: 23.01.2018.

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 motivaciju i logiku adaptivnih protokola. Koja su dodatna stanja u protokolu EDWP i koja je njihova semantika? Objasniti kako se vrši invalidacija u ovom protoklu.

2. [15] Objasniti zaključke analize načina deljenja podataka u tipičnim paralelnim aplikacijama, kao i konsekvence ove analize na strukturu kataloga u *directory* protokolima.

3. [15] Objasniti kako je organizovan katalog u tehnici *kombinovanog kataloga*. Opisati osnovne akcije.

4. [15] Kojoj grupi interkonekcionih mreža pripada *Butterfly*? Objasniti način povezivanja i nacrtati ovu mrežu za 8 ulaza i 8 izlaza. Nacrtati put poruke između ulaza 7 i izlaza 3. Objasniti da li je mreža blokirajuća.

5. [15] Iz zadanog CUDA jezgra koje određuje histogram zadanog niza su uklonjeni određeni sinhronizacioni konstrukti. Na koji način se sve može postići sinhronizacija u okviru CUDA jezgra? Obrazložiti i dodati na odgovarajuća mesta u kodu sinhronizacione konstrukte, kako bi se kod u prilogu korektno izvršavao.

```
__global__ void histo_kernel(const char *input, int *bins,
                             int num_elements, int num_bins) {
    int tid = blockIdx.x * blockDim.x + threadIdx.x;

    extern __shared__ unsigned int bins_s[];

    for (int binIdx = threadIdx.x; binIdx < num_bins; binIdx += blockDim.x)
        bins_s[binIdx] = 0;

    for (int i = tid; i < num_elements; i += blockDim.x * gridDim.x)
        atomicAdd(&(bins_s[(unsigned int)input[i]]), 1);

    for (int binIdx = threadIdx.x; binIdx < num_bins; binIdx += blockDim.x)
        atomicAdd(&(bins[binIdx]), bins_s[binIdx]);
}
```

6. [10] Na koji način je organizovana memorijska hijerarhija grafičkog procesora i kako to utiče na performanse izvršavanja koda?

7. [15] Koristeći CUDA tehnologiju paralelizovati funkciju koja je data u prilogu. Funkcija vrši konvoluciju dva zadata signala. Prilikom paralelizacije koristiti deljenu memoriju. Obratiti pažnju na efikasnost paralelizacije. Napisati deo koda za centralni procesor koji vrši pozivanje implementiranog jezgra.

```
void convolve(double *Signal, size_t SignalLen,
             double *Kernel, size_t KernelLen, double *Result) {
    size_t i, kmin, kmax, k;
    for (i = 0; i < SignalLen + KernelLen - 1; i++) {
        Result[i] = 0;
        kmin = (i >= KernelLen - 1) ? i - (KernelLen - 1) : 0;
        kmax = (i < SignalLen - 1) ? i : SignalLen - 1;
        for (k = kmin; k <= kmax; k++)
            Result[i] += Signal[k] * Kernel[i - k];
    }
}
```

