

Verificare finală: Arhitectura Sistemelor de Calcul (ASC) 2020

Întrebarea 1 (adunare numere naturale)

1. Se dau două numere naturale în format binar, $a = 10101111$ și $b = 01101100$. Calculați rezultatul $c = a + b$ (atât în binar, cât și în zecimal, octal și hexazecimal prefix 0x)
2. Se dau două numere naturale în format binar, $a = 00011011$ și $b = 01101100$. Calculați rezultatul $c = a + b$ (atât în binar, cât și în zecimal, octal și hexazecimal prefix 0x)
3. Se dau două numere naturale în format binar, $a = 11011011$ și $b = 11111100$. Calculați rezultatul $c = a + b$ (atât în binar, cât și în zecimal, octal și hexazecimal prefix 0x)
4. Se dau două numere naturale în format binar, $a = 11111011$ și $b = 10001100$. Calculați rezultatul $c = a + b$ (atât în binar, cât și în zecimal, octal și hexazecimal prefix 0x)
5. Se dau două numere naturale în format binar, $a = 10111111$ și $b = 11111100$. Calculați rezultatul $c = a + b$ (atât în binar, cât și în zecimal, octal și hexazecimal prefix 0x)

Întrebarea 2 (înmulțire numere naturale)

1. Se dau două numere naturale în format binar, $a = 10101111$ și $b = 01101100$. Calculați rezultatul $c = a \times b$ (atât în binar, cât și în zecimal, octal și hexazecimal prefix 0x)
2. Se dau două numere naturale în format binar, $a = 00011011$ și $b = 01101100$. Calculați rezultatul $c = a \times b$ (atât în binar, cât și în zecimal, octal și hexazecimal prefix 0x)
3. Se dau două numere naturale în format binar, $a = 11011011$ și $b = 11111100$. Calculați rezultatul $c = a \times b$ (atât în binar, cât și în zecimal, octal și hexazecimal prefix 0x)
4. Se dau două numere naturale în format binar, $a = 11111011$ și $b = 10001100$. Calculați rezultatul $c = a \times b$ (atât în binar, cât și în zecimal, octal și hexazecimal prefix 0x)
5. Se dau două numere naturale în format binar, $a = 10111111$ și $b = 11111100$. Calculați rezultatul $c = a \times b$ (atât în binar, cât și în zecimal, octal și hexazecimal prefix 0x)

Întrebarea 3 (adunare numere întregi)

1. Se dau două numere întregi în format binar pe 8 biți, $a = 10101111$ și $b = 01101100$. Calculați rezultatul $c = a + b$ (atât în binar, cât și în zecimal și hexazecimal prefix 0x)
2. Se dau două numere întregi în format binar pe 8 biți, $a = 11110000$ și $b = 11001100$. Calculați rezultatul $c = a + b$ (atât în binar, cât și în zecimal și hexazecimal prefix 0x)
3. Se dau două numere întregi în format binar pe 8 biți, $a = 11011011$ și $b = 11111100$. Calculați rezultatul $c = a + b$ (atât în binar, cât și în zecimal și hexazecimal prefix 0x)
4. Se dau două numere întregi în format binar pe 8 biți, $a = 11111011$ și $b = 10001100$. Calculați rezultatul $c = a + b$ (atât în binar, cât și în zecimal și hexazecimal prefix 0x)
5. Se dau două numere întregi în format binar pe 8 biți, $a = 10111111$ și $b = 11111100$. Calculați rezultatul $c = a + b$ (atât în binar, cât și în zecimal și hexazecimal prefix 0x)

Întrebarea 4 (înmulțire numere întregi)

1. Se dau două numere întregi în format binar pe 8 biți, $a = 10101111$ și $b = 01101100$. Calculați rezultatul $c = a \times b$ (atât în binar, cât și în zecimal și hexazecimal prefix 0x)
2. Se dau două numere întregi în format binar pe 8 biți, $a = 11110000$ și $b = 11001100$. Calculați rezultatul $c = a \times b$ (atât în binar, cât și în zecimal și hexazecimal prefix 0x)
3. Se dau două numere întregi în format binar pe 8 biți, $a = 11011011$ și $b = 01111100$. Calculați rezultatul $c = a \times b$ (atât în binar, cât și în zecimal și hexazecimal prefix 0x)
4. Se dau două numere întregi în format binar pe 8 biți, $a = 01111011$ și $b = 10001100$. Calculați rezultatul $c = a \times b$ (atât în binar, cât și în zecimal și hexazecimal prefix 0x)
5. Se dau două numere întregi în format binar pe 8 biți, $a = 10111111$ și $b = 01111100$. Calculați rezultatul $c = a \times b$ (atât în binar, cât și în zecimal și hexazecimal prefix 0x)

Întrebarea 5 (reprezentare, cifre)

1. Dacă cifra "15" are reprezentarea "F" atunci cifra "19" ar avea reprezentarea:
2. Dacă cifra "10" are reprezentarea "A" atunci cifra "22" ar avea reprezentarea:
3. Dacă cifra "12" are reprezentarea "C" atunci cifra "25" ar avea reprezentarea:
4. Dacă cifra "15" are reprezentarea "F" atunci cifra "26" ar avea reprezentarea:
5. Dacă cifra "13" are reprezentarea "D" atunci cifra "19" ar avea reprezentarea:

Întrebarea 6 (entropia)

1. Calculați entropia următorului set de evenimente care are probabilitățile: $\{1/2, 1/4, 1/8, 1/8\}$
2. Calculați entropia următorului set de evenimente care are probabilitățile: $\{1/3, 1/6, 1/6, 1/12, 1/12, 1/12, 1/12\}$
3. Calculați entropia următorului set de evenimente care are probabilitățile: $\{1/4, 1/4, 1/4, 1/8, 1/8\}$
4. Calculați entropia următorului set de evenimente care are probabilitățile: $\{1/2, 1/4, 1/8, 1/16, 1/16\}$
5. Calculați entropia următorului set de evenimente care are probabilitățile: $\{1/4, 1/4, 1/4, 1/16, 1/16, 1/16, 1/16\}$

Întrebarea 7 (simplificări logice)

1. Simplificați următoarea expresie logică $A = !Y + !X + XY!Z$
2. Simplificați următoarea expresie logică $A = (Y + Z)(X + Z)(!X + Y)$
3. Simplificați următoarea expresie logică $A = XZ + YZ + !XY$
4. Simplificați următoarea expresie logică $A = X^2(!X + !Y)$
5. Simplificați următoarea expresie logică $A = !Y + 0 + !X + X!Z Y$

Întrebarea 8 (counter pe 3 biți, scădere)

1. Fie un counter binar pe 3 biți. Presupunem că avem operația de decrementare (sărim de la elementul curent în la precedentul) adică trecem de la cei 3 biți $q_2^{(t)} q_1^{(t)} q_0^{(t)}$ la $q_2^{(t+1)} q_1^{(t+1)} q_0^{(t+1)}$. Scrieți expresia logică (simplificată maxim) pentru $q_2^{(t+1)}$ în funcție de $q_2^{(t)}$, $q_1^{(t)}$ și $q_0^{(t)}$.

2. Fie un counter binar pe 3 biți. Presupunem că avem operația de decrementare (sărim de la elementul curent în la precedentul) adică trecem de la cei 3 biți $q_2^{(t)} q_1^{(t)} q_0^{(t)}$ la $q_2^{(t+1)} q_1^{(t+1)} q_0^{(t+1)}$. Scrieți expresia logică (simplificată maxim) pentru $q_1^{(t+1)}$ în funcție de $q_2^{(t)}$, $q_1^{(t)}$ și $q_0^{(t)}$.
3. Fie un counter binar pe 3 biți. Presupunem că avem operația de decrementare (sărim de la elementul curent în la precedentul) adică trecem de la cei 3 biți $q_2^{(t)} q_1^{(t)} q_0^{(t)}$ la $q_2^{(t+1)} q_1^{(t+1)} q_0^{(t+1)}$. Scrieți expresia logică (simplificată maxim) pentru $q_0^{(t+1)}$ în funcție de $q_2^{(t)}$, $q_1^{(t)}$ și $q_0^{(t)}$.

Întrebarea 9 (calcul eficient, minim de înmulțiri)

1. Pentru un număr natural dat a , explicați cum calculați eficient (număr minim înmulțiri) a^{15}
2. Pentru un număr natural dat a , explicați cum calculați eficient (număr minim înmulțiri) a^{13}
3. Pentru un număr natural dat a , explicați cum calculați eficient (număr minim înmulțiri) a^9
4. Pentru un număr natural dat a , explicați cum calculați eficient (număr minim înmulțiri) a^{11}
5. Pentru un număr natural dat a , explicați cum calculați eficient (număr minim înmulțiri) a^{14}

Întrebarea 10 (reprezentarea în format IEEE FP)

1. Reprezentați numărul 0.33 (sau cel mai apropiat posibil) în formatul IEEE 754 Floating Point (răspuns hexa)
2. Reprezentați numărul 3.14 (sau cel mai apropiat posibil) în formatul IEEE 754 Floating Point (răspuns hexa)
3. Reprezentați numărul 0.67 (sau cel mai apropiat posibil) în formatul IEEE 754 Floating Point (răspuns hexa)
4. Reprezentați numărul 2.71 (sau cel mai apropiat posibil) în formatul IEEE 754 Floating Point (răspuns hexa)
5. Reprezentați numărul 1.62 (sau cel mai apropiat posibil) în formatul IEEE 754 Floating Point (răspuns hexa)

Întrebarea 11 (împărțirea numerelor întregi)

1. Se dau următoarele numere întregi (scrise în complement față de doi): 10010001 și 1010. Împărțiți cele două numere (calculați rezultatul și restul în binar și zecimal).
2. Se dau următoarele numere întregi (scrise în complement față de doi): 10010011 și 0011. Împărțiți cele două numere (calculați rezultatul și restul în binar și zecimal).
3. Se dau următoarele numere întregi (scrise în complement față de doi): 11101010 și 0110. Împărțiți cele două numere (calculați rezultatul și restul în binar și zecimal).
4. Se dau următoarele numere întregi (scrise în complement față de doi): 11110000 și 011. Împărțiți cele două numere (calculați rezultatul și restul în binar și zecimal).
5. Se dau următoarele numere întregi (scrise în complement față de doi): 111010101 și 0101. Împărțiți cele două numere (calculați rezultatul și restul în binar și zecimal).

Întrebarea 12 (performanța sistemelor multi-procesor)

1. Aveți un sistem de calcul cu 6 procesoare și un program care se poate paraleliza în proporție de 80%. Care este îmbunătățirea în timpul de rulare serial vs. paralel?
2. Aveți un sistem de calcul cu s procesoare și un program care se poate paraleliza în proporție de 90%. De câte procesoare este nevoie ca îmbunătățirea în timpul de rulare serial vs. paralel să fie de 5 ori mai rapid?
3. Aveți un sistem de calcul cu 3 procesoare și un program care se poate paraleliza în proporție de $p\%$. Care este p astfel încât îmbunătățirea în timpul de rulare serial vs. paralel să fie de 2 ori mai rapid?
4. Aveți un program care se poate paraleliza în proporție de 75%. Dacă avem disponibil un număr infinit de procesoare pentru a rula programul, care este îmbunătățirea în viteza de execuție?
5. Aveți sistem de calcul cu s procesoare și un program care se poate paraleliza în proporție de 30%. De câte procesoare este nevoie ca îmbunătățirea în timpul de rulare serial vs. paralel să fie de 5 ori mai rapid?

Întrebarea 13 (performanța, număr mediu de instrucțiuni)

1. Considerăm un sistem de calcul de 32 de biți. Sistemul poate realiza operațiile următoare: operații aritmetice/logice (1 ciclu), operații de citire/sciere date în memorie (2 cicli) și operații de branch/salt (3 cicli). Pentru ca operațiile aritmetice/logice să fie executate programul realizează înainte două instrucțiuni de citire (citirea operandilor) și apoi o operație de sciire (sciirea rezultatului). Avem un program care are în componență 10% operații aritmetice/logice, 60% operații de citire/sciere (40% operații citire și 20% operații scriere) și 30% operații de branch/salt.
Presupunem că adăugăm o nouă instrucțiune pentru operațiile aritmetice/logice care include cele două citiri și scrierea rezultatului. Dar, noua instrucțiune are nevoie de 3 cicli.
Cât de mult (procentual) este îmbunătățit sistemul de calcul?
2. Considerăm un sistem de calcul de 32 de biți. Sistemul poate realiza operațiile următoare: operații aritmetice/logice (1 ciclu), operații de citire/sciere date în memorie (2 cicli) și operații de branch/salt (3 cicli). Pentru ca operațiile de salt să fie executate programul realizează înainte două instrucțiuni de citire (citirea operandilor care să fie comparați). Avem un program care are în componență 10% operații aritmetice/logice, 70% operații de citire/sciere (45% operații citire și 25% operații scriere) și 20% operații de branch/salt.
Presupunem că adăugăm o nouă instrucțiune pentru operația de salt care include citirile. Dar, noua instrucțiune are nevoie de 4 cicli. Cât de mult (procentual) este îmbunătățit sistemul?
3. Considerăm un sistem de calcul de 32 de biți. Sistemul poate realiza operațiile următoare: operații aritmetice/logice (1 ciclu), operații de citire/sciere date în memorie (2 cicli) și operații de branch/salt (3 cicli). Pentru ca operațiile de salt să fie executate programul realizează înainte o instrucțiune de citire (citirea operandului care să fie comparat cu o constantă). Avem un program care are în componență 10% operații aritmetice/logice, 60% operații de citire/sciere (40% operații citire și 20% operații scriere) și 30% operații de branch/salt.

Presupunem că adăugăm o nouă instrucțiune pentru operația de salt care include citirea. Dar, noua instrucțiune are nevoie de 4 cicli. Cât de mult (procentual) este îmbunătățit sistemul?

Întrebarea 14 (data hazards)

1. Rx sunt regiștrii, ce fel de hazard este present în următoarea secvență de cod: $R6 = R1 \times R2$, $R7 = R5 \times R6$. Explicați de ce.
2. Rx sunt regiștrii, ce fel de hazard este present în următoarea secvență de cod: $R3 = R1 / R2$, $R5 = R4 \times R3$, $R4 = R0 + R6$. Explicați de ce.
3. Rx sunt regiștrii, ce fel de hazard este present în următoarea secvență de cod: $R6 = R1 \times R2$, $R6 = R4 + R5$. Explicați de ce.

Întrebarea 15 (great ideas in computer architecture)

1. Conform Patterson & Hennessy, care sunt cele 8 idei importante din arhitectura sistemelor de calcul moderne? Enumerați-le pe toate dar alegeți două dintre ele (cele care vi se par cele mai importante vouă) și explicați-le detaliat (1-2 propoziții cu esența ideilor).

Întrebarea 16 (mixt)

1. Se dă un șir de N biți $a_{N-1}, a_{N-2}, \dots, a_1, a_0$. Care este rezultatul operației $1a_{N-1}a_{N-2}\dots a_1a_0 + 0a_{N-1}a_{N-2}\dots a_1a_0$? (binar și zecimal)
2. Se dă un șir de N biți $a_{N-1}, a_{N-2}, \dots, a_1, a_0$. Care este rezultatul operației $1!a_{N-1}!a_{N-2}\dots!a_1!a_0 + 0a_{N-1}a_{N-2}\dots a_1a_0$? (binar și zecimal)
3. Se dă un șir de N biți $a_{N-1}, a_{N-2}, \dots, a_1, a_0$. Care este rezultatul operației $0!a_{N-1}!a_{N-2}\dots!a_1!a_0 + 1a_{N-1}a_{N-2}\dots a_1a_0$? (binar și zecimal)
4. Se dă un șir de N biți $a_{N-1}, a_{N-2}, \dots, a_1, a_0$. Care este rezultatul operației $0!a_{N-1}!a_{N-2}\dots!a_1!a_0 + 1!a_{N-1}!a_{N-2}\dots!a_1!a_0$? (binar și zecimal)
5. Se dă un set cu un număr infinit de simboluri $\{a_0, a_1, a_2, \dots\}$ fiecare cu probabilitate de apariție $\{1/2, 1/4, 1/8, 1/16, 1/32, 1/64, \dots\}$. Care este entropia acestui set?
6. Avem 3 variabile separate: S pe 1 bit (semnul), M pe 23 de biți (mantisa) și E pe 8 biți (exponentul). Scrieți secvența de operații aritmetice (eficient numeric) care crează numărul F, un număr în reprezentarea IEEE FP care are semnul S, mantisa M și exponentul E.
7. Fie F un număr scris în reprezentarea IEEE FP. Schimbați (eficient) valoarea lui F astfel încât toți biți cu index par din exponent să fie puși la zero (reprezențați constantele folosite în hex).
8. Fie F un număr scris în reprezentarea IEEE FP. Schimbați (eficient) valoarea lui F astfel încât toți biți cu index par din exponent să fie puși la unu (reprezențați constantele folosite în hex).
9. Fie F un număr scris în reprezentarea IEEE FP. Schimbați valoarea lui F astfel încât primii 3 biți și ultimii 3 biți din mantisă să fie puși la zero (reprezențați constantele folosite în hex).
10. Fie F un număr scris în reprezentarea IEEE FP (semnul este S, exponentul este E și mantisa este M). Calculați eficient o aproximare pentru $\log_2(F)$. Scrieți pas cu pas calculul. (veți folosi faptul că $\log_2(1+x) \approx x$)

Întrebarea 17 (tabel de adevăr)

1. Fie următorul tabel de adevăr. Calculați expresia pentru X și apoi simplificați X.

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

2. Fie următorul tabel de adevăr. Calculați expresia pentru X și apoi simplificați X.

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

3. Fie următorul tabel de adevăr. Calculați expresia pentru X și apoi simplificați X.

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

4. Fie următorul tabel de adevăr. Calculați expresia pentru X și apoi simplificați X.

A	B	C	X
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	1
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	1

Întrebarea 18 (operații ciudate cu variabile digitale)

1. Am discutat la curs despre faptul ca operația “+” (adunare) reprezintă operația “OR”. Care ar fi rezultatul operațiilor $X = 1 - A$, $Y = 0 - A$, $Z = B - A$? (cum am defini “scăderea”?)
2. Am discutat la curs despre faptul ca operația “x” (înmulțire) reprezintă operația “AND”. Care ar fi rezultatul operațiilor $X = 1/A$, $Y = 0/A$, $Z = B/A$? (cum am defini “împărțirea”?)
3. Am discutat la curs despre faptul ca operația “x” (înmulțire) reprezintă operația “AND”. Care ar fi rezultatul operației $X = \text{sqrt}(A)$, $Y = \text{sqrt_ordin_n}(A)$? (cum am defini “radicalul”?)

Întrebarea 19 (GPU)

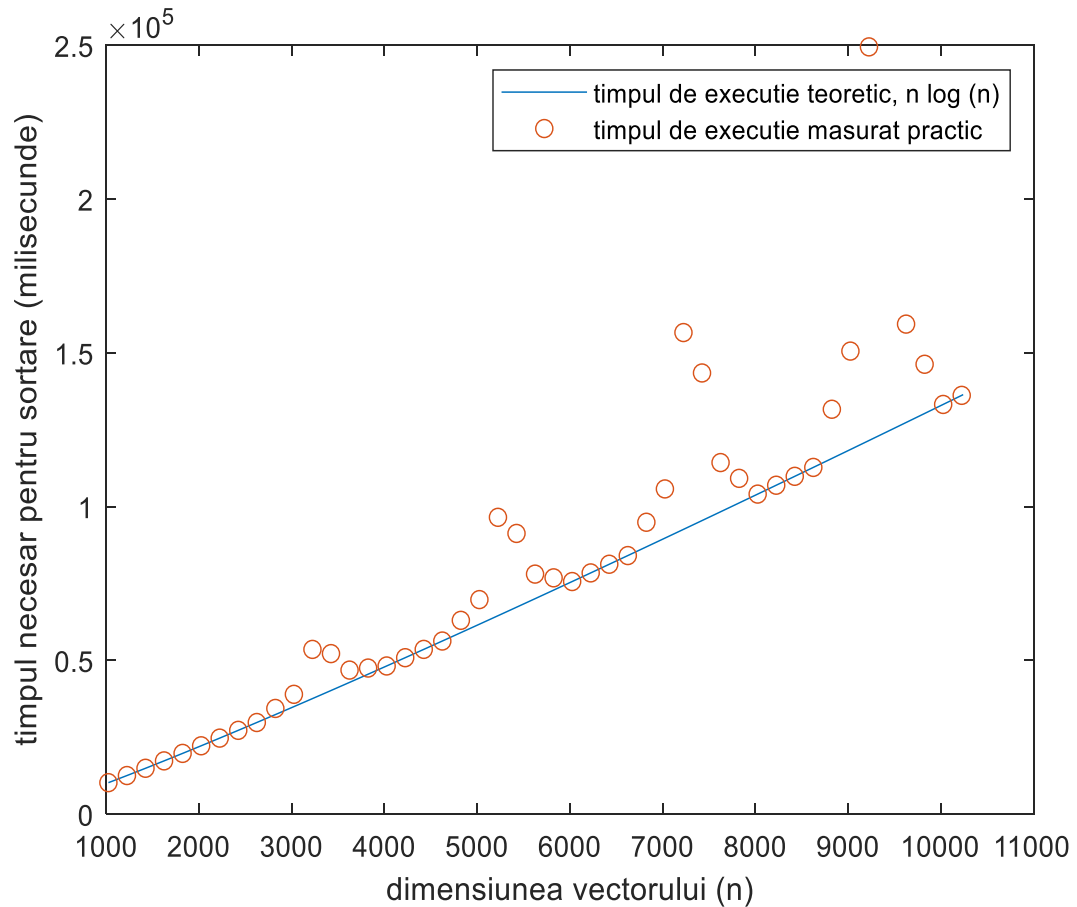
1. În septembrie 2018 a fost lansată placa grafică GeForce RTX 2080. Întrebări:
 - a. ce microarhitectura folosește această placă grafică?
 - b. câți tranzistori are această placă grafică?
 - c. câtă memorie are această placă grafică?
 - d. care este performanța de calcul? (FLOPS)
 - e. câte unități identice de calcul conține placa grafică?
 - f. cine (ce companie) dezvoltă această placă?
2. În august 2018 a fost lansată placa grafică Quadro RTX 8000. Întrebări:
 - a. ce microarhitectura folosește această placă grafică?
 - b. frecvența de funcționare?
 - c. câtă memorie are această placă grafică?
 - d. care este performanța de calcul? (FLOPS pentru double precision arithmetic)
 - e. câte unități identice de calcul (CUDA cores) conține placa grafică?
 - f. cine (ce companie) dezvoltă această placă?
3. În august 2018 a fost lansată placa grafică Radeon Pro 5300M (Navi 14). Întrebări:
 - a. ce microarhitectura folosește această placă grafică?
 - b. frecvența de funcționare?
 - c. câtă memorie are această placă grafică?
 - d. care este performanța de calcul? (FLOPS pentru double precision arithmetic)
 - e. câte unități identice de calcul (CUDA cores) conține placa grafică?
 - f. cine (ce companie) dezvoltă această placă?

Întrebarea 20 (din cursul de sisteme integrate)

1. Enumerați 3 avantaje ale unui sistem “system on a chip”. Dați un exemplu de sistem de calcul care folosește system on a chip.
2. Enumerați avantajele unui sistem RISC față de unul CISC.
3. Enumerați avantajele unui sistem CISC față de unul RISC.
4. Care este diferența între concepul de “hard FP” și “soft FP”?
5. Care este diferența majoră între un sistem de operare “normal” și un sistem de operare “real-time”?
6. De ce nu este Raspberry Pi-ul afectat de atacurile “Spectre” și “Meltdown”?

Întrebarea 21 (timpii pentru sortarea pe un laptop)

1. Avem un laptop performant pe care rulăm un algoritm de sortare care are complexitatea numerică $O(n \log n)$. Rulăm pentru multe valori continue ale lui n (de la 1024 la 10240, cu o pauză de 5 secunde între rulări) și obținem graficul de mai jos pentru timpii de execuție. Ce se întâmplă? De ce avem aceste vârfuri față de timpul de execuție teoretic?



Întrebarea 22 (cache/pipelining/out of order execution)

1. Fie expresia $1/\sqrt{x^2 + y^2 + z^2}$. Folosind timpii de execuție pentru operațiile de bază din cursul B (performanța sistemelor de calcul) raspundeți la următoarele întrebări:
 - a. care este timpul necesar pentru calcularea expresiei (toate operațiile se fac secvențial)
 - b. vi se spune că aveți o unitate FPU care permite execuția a doua instrucțiuni simultan. care este timpul de execuție necesar acum?
 - c. expresia de mai sus este calculată folosind numere zecimale x , y și z în precizie finită reprezentate cu standardul IEEE FP. Avem nevoie să știm dacă expresia de mai sus este egală cu $1/\sqrt{3}$. Cum facem această comparație? (de exemplu, în limbajul C)

Întrebarea 23 (calcul matriceal eficient)

1. Explicați cum calculați eficient $X = A * A$ unde A este o matrice generală de dimensiune $n \times n$.
2. Explicați cum calculați eficient $X = A^T * A$ unde A este o matrice generală de dimensiune $n \times n$.
3. Explicați cum calculați eficient $X = A * A^T$ unde A este o matrice generală de dimensiune $n \times n$.
4. Explicați cum calculați eficient $X = A^T * A^T$ unde A este o matrice generală de dimensiune $n \times n$.

Întrebarea 24 (CPUs)

1. În noiembrie 2020, AMD a lansat procesoare cu microarhitectura Zen 3. Răspundeți la următoarele întrebări:
 - a. ce tip de tehnologie semiconductoare este Zen 3? (nm)
 - b. câte core-uri au sistemele AMD Ryzen™ 5000 Series Desktop Processors? (interval, valori tipice)
 - c. cât cache L1/L2/L3 au aceste sisteme?
 - d. pentru aceeași frecvență de operare, cu cât (%) este arhitectura Zen 3 mai rapidă decât tehnologia precedentă Zen 2. De ce (care este motivul principal)?
 - e. care este frecvența tipică de operare pentru AMD Ryzen™ 5000 Series Desktop Processors?
 - f. valori recomandate pentru overcloacking?
2. În septembrie 2020, Intel a lansat procesoare cu microarhitectura Tiger Lake. Răspundeți la următoarele întrebări:
 - a. ce tip de tehnologie semiconductoare este Tiger Lake? (nm)
 - b. câte core-uri au sistemele i7 din această generație? (valoare tipică)
 - c. cât cache L1/L2/L3 au aceste sisteme?
 - d. a câta generație de procesoare Intel reprezintă Tiger Lake?
 - e. care este frecvența tipică de operare pentru sistemele i7 din această generație?
 - f. frecvența maximă (overcloacking)?

Întrebarea 25 (împărțiri eficiente)

1. Ce realizează următoarea secvență de cod (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?

```
movsx rax, edi
imul rax, rax, -2004318071
shr rax, 32
add eax, edi
sar edi, 31
sar eax, 3
sub eax, edi
```
2. Ce realizează următoarea secvență de cod (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?

```
movsx rax, edi
sar edi, 31
```

- ```

 imul rax, rax, 818089009
 sar rax, 34
 sub eax, edi

```
3. Ce realizează următoarea secvență de cod (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?
 

```

 movsx rax, edi
 sar edi, 31
 imul rax, rax, 2021161081
 sar rax, 35
 sub eax, edi

```
  4. Ce realizează următoarea secvență de cod (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?
 

```

 movsx rax, edi
 sar edi, 31
 imul rax, rax, 1321528399
 sar rax, 34
 sub eax, edi

```
  5. Ce realizează următoarea secvență de cod (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?
 

```

 movsx rax, edi
 sar edi, 31
 imul rax, rax, 780903145
 sar rax, 33
 sub eax, edi

```

### Întrebarea 26 (cod ASM simplu)

1. Ce face codul de mai jos (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?
 

```

 mov edx, 1
 mov eax, 1
 .L2:
 imul rax, rdx
 add rdx, 1
 cmp rdx, 11
 jne .L2
 ret

```
2. Ce face codul de mai jos (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?
 

```

 test edx, edx
 jle .L4
 lea ecx, [rdx+1]
 mov eax, 1
 xor r8d, r8d
 .L3:
 mov edx, DWORD PTR [rdi+rax*4]
 imul edx, DWORD PTR [rsi+rax*4]

```

```

add rax, 1
add r8d, edx
cmp rcx, rax
jne .L3
mov eax, r8d
ret

```

```

.L4:
xor r8d, r8d
mov eax, r8d
ret

```

3. Ce face codul de mai jos (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?

```

test esi, esi
jle .L4
lea eax, [rsi-1]
lea rdx, [rdi+4]
lea rcx, [rdi+8+rax*4]
xor eax, eax

```

```

.L3:
add eax, DWORD PTR [rdx]
add rdx, 4
cmp rdx, rcx
jne .L3
idiv esi
ret

```

```

.L4:
xor eax, eax
ret

```

4. Ce face codul de mai jos (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?

```

test edi, edi
js .L4
add edi, 1
xor edx, edx
mov eax, 1
mov ecx, 1

```

```

.L3:
mov esi, eax
add edx, 1
add eax, ecx
mov ecx, esi
cmp edi, edx
jne .L3
ret

```

```
.L4:
 mov eax, 1
 ret
```

### Întrebarea 27 (cod ASM complex)

1. Ce face codul de mai jos (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?

```
1. mov r9, rdi
2. lea r8d, [rsi-1]
3. lea r11d, [rsi-2]
4. xor edi, edi
5. lea r10, [r9+4]
6. .L5:
7. mov eax, r8d
8. sub eax, edi
9. test eax, eax
10. jle .L2
11. mov edx, r11d
12. mov rax, r9
13. sub edx, edi
14. lea rsi, [r10+rdx*4]
15. .L3:
16. mov edx, DWORD PTR [rax]
17. mov ecx, DWORD PTR [rax+4]
18. cmp edx, ecx
19. jle .L4
20. mov DWORD PTR [rax], ecx
21. mov DWORD PTR [rax+4], edx
22. .L4:
23. add rax, 4
24. cmp rsi, rax
25. jne .L3
26. .L2:
27. add edi, 1
28. jmp .L5
```

2. Ce face codul de mai jos (x86-64, generat de godbolt.org, x86-64 gcc 10.2)?

```
1. push rbp
2. mov rcx, rdi
3. lea r11, [rdi+4]
4. mov ebp, 1
5. push rbx
6. mov rbx, -1
7. .L4:
8. mov r8d, DWORD PTR [r11]
9. mov edx, DWORD PTR [r11-4]
10. mov rax, r11
11. cmp r8d, edx
12. jge .L2
13. lea r9d, [rbp-2]
14. mov rax, rbx
15. sub r9d, ebx
16. .L3:
17. lea esi, [r9+rax]
18. mov DWORD PTR [rcx+8+rax*4], edx
19. mov edx, DWORD PTR [rcx+rax*4]
20. mov r10, rax
21. cmp esi, -1
22. sete dil
23. cmp edx, r8d
24. setle sil
25. sub rax, 1
26. or dil, sil
27. je .L3
28. lea rax, [rcx+4+r10*4]
29. .L2:
30. mov DWORD PTR [rax], r8d
31. add ebp, 1
32. add r11, 4
33. add rbx, 1
34. jmp .L4
```