

Kezdés ideje	2022. május 27., péntek, 11:06
Állapot	Befejezte
Befejezés dátuma	2022. május 27., péntek, 11:54
Felhasznált idő	48 perc 5 mp
Pont	27,00 a(z) 30,00 maximumból (90%)

1 kérdés

Helyes

2,00/2,00 pont

Mely szereplő kezdeményezi a virtuális címek leképzését fizikai címekre?

- ✓ A felhasználói program, ha a hivatkozott adat a háttértáron van
- ✓ Az operációs rendszer, ha a hivatkozott adat a háttértáron van
- ✓ A processzor / az MMU, minden memóriahivatkozás alkalmával
- ✓ A processzor / az MMU, minden laphiba alkalmával

2 kérdés

Helyes

2,00/2,00 pont

Jelölje be, mely fázisok végeznek tényleges munkát egy load (memóriából olvasás) utasítás feldolgozása közben a tanult 5 fokozatú pipeline-ban!

- ✓ ID (Instruction Decode)
- ✓ EX (Execute)
- ✓ MEM (Memory)
- ✓ WB (Write Back)

3 kérdés

Részben helyes

0,00/2,00 pont

Hogyan hasznosítja az agree prediktor a globális és lokális információkat az elágazásbecslés során?

- ✓ A globális előzmény regisztert és az utasításszámlálót XOR-olja
- ✗ Egy tisztán lokális és egy globális, korrelált becslő eredményét NXOR-olja
- ✗ A globális és a lokális becslő közül annak hisz, amelyik az utóbbi időben pontosabban becsül
- ✓ Az agree prediktor nem vegyít globális és lokális információkat

4 kérdés

Helyes

2,00/2,00 pont

Egy sorrendvisszaállító bufferrel rendelkező out-of-order processzorban az utasításvégrehajtás alábbi fázisai közül melyek történnek a program szerint megadott sorrendben?

- IGAZ Utasítások bejegyzése a sorrendvisszaállító buffer-be
- HAMIS Utasítások végrehajtása
- IGAZ Utasítások lehívása
- IGAZ Utasítások elhelyezése az utasítástárolóban

5 kérdés

Helyes

2,00/2,00 pont

X

Mely állítások igazak a RISC, és melyek a CISC utasításkészletekre?

	CISC	RISC
Tömörebb programkód	X <input checked="" type="checkbox"/>	<input type="checkbox"/>
Tipikusan több regiszter	<input type="checkbox"/>	X <input checked="" type="checkbox"/>
Változó hosszúságú utasításkódolás	X <input checked="" type="checkbox"/>	<input type="checkbox"/>
Külön I/O és memóriakezelő utasítások	X <input checked="" type="checkbox"/>	<input type="checkbox"/>

6 kérdés

Helyes

2,00/2,00 pont

X

Az alábbi helyzetek milyen adat-egymásrahatásnak felelnek meg?

	RAW	WAW	WAR
Két utasítás ugyanabba a regiszterbe írja az eredményét	<input type="checkbox"/>	X <input checked="" type="checkbox"/>	<input type="checkbox"/>
Két utasítás ugyanazt a regisztert olvassa	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Egy utasítás felhasználja egy korábbi utasítás eredményét	X <input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A program egy elágazáshoz érkezik, és nem ismert a folytatás iránya	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

7 kérdés

Kész

2,00/2,00 pont

Soroljon fel 3 cache blokk csere stratégiát!

LRU: least recently used, legtöbb ideje nem használtat dobja ki.

Random: egy véletlen kiválasztott blokkot dob ki.

Forgó: először az az első blokkot dobja ki, utána a másodikat, harmadikat, stb... Ha a végére ért, kezdi az elejétől.

Megjegyzés:

8 kérdés

Részben helyes

1,00/2,00 pont

Adja meg, hogy egy 2 utas in-order szuperskalár processzorban melyik egységből hány található!

Vezérlő (dekóder) egységek száma: ✘ [2]

Aritmetikai/logikai egységek száma: ✔

Adat cache-ek száma: ✔

9 kérdés

Helyes

2,00/2,00 pont

Amdahl törvénye szerint elméletileg mennyivel gyorsabban futhat egy program egy végtelen sok processzorból álló multiprocesszoros rendszerben, mint egy 1 processzoros rendszerben, ha a programnak csak a 3/4-ed része párhuzamosítható tökéletesen, a többi csak szekvenciálisan futtatható?

A gyorsulás mértéke: ✔

10 kérdés

Helyes

6,00/6,00 pont

Van egy 512 bájt összkapacitású cache memóriánk, 64 bájos blokk mérettel és 4-utas asszociatív szervezéssel. Az épp futó program a rendszermemória alábbi **blokkjait** hivatkozta:

- 21, 48, 17, 196

Az alábbi táblázatban adja meg a fenti blokkszámokhoz tartozó tag és index mezőket!

Blokkszám:	21	48	17	196
Index:	1 ✓	0 ✓	1 ✓	0 ✓
Tag:	10 ✓	24 ✓	8 ✓	98 ✓

Hogy változik a cache memória tartalma, ha az alábbi kiinduló állapotokban a megadott blokkra hivatkozik a program? Használjon LRU cserealgoritmust, és a mezők melletti szürke, szaggatott vonallal leválasztott területen tartsa nyilván a blokkok egymáshoz képesti korát (1 és 4 között)! (Minél nagyobb a kor, annál régebben került a cache-be az adat.) Ha több lehetőség is van egy blokk elhelyezésére, válassza mindig a legbaloldalibbat. Elegendő csak a megváltozott mezőket beírni!

a)

Kiindulóállapot:

48	3	132	2	96	1	8	4
17	2	19	4	197	3	3	1

A 21 beszúrása után:

	3 ✓	21 ✓	1 ✓		4 ✓		2 ✓

b)

Kiindulóállapot:

48	3	132	2	96	1	8	4
17	2	19	4	197	3	3	1

A 48 beszúrása után:

48 ✓	1 ✓		3 ✓		2 ✓		

c)

Kiindulóállapot:

48	3	132	2	96	1	8	4
17	2	19	4	197	3	3	1

A 17 beszúrása után:

17 ✓	1 ✓						2 ✓

d)

Kiindulóállapot:

48	3	132	2	96	1	8	4
17	2	19	4	197	3	3	1

Az 196 beszúrása után:

	4 ✓		3 ✓		2 ✓	196 ✓	1 ✓

Your answer is correct.

11 kérdés

Helyes

3,00/3,00 pont

Legyen adott az alábbi utasítás sorozat:

```
i1: R2 ← MEM [R0+0]
i2: R3 ← R0 * R2
i3: R8 ← R4 / R3
i4: R5 ← MEM [R1+8]
i5: R6 ← R2 / R5
i6: R9 ← R5 / R6
i7: R10 ← R6 - R9
```

Az utasítássorozatot lefordítjuk egy olyan VLIW processzorra, melyben az alábbi utasítások helyezhetők el egy utasításcsoportban:

- 2 db memóriaművelet (végrehajtási idő: 3 órajel, iterációs idő: 3 órajel)
- 2 db egész aritmetikai művelet, vagy ugró utasítás (végrehajtási idő: 1 órajel)

Ütemezze a megadott utasítássorozatot a megadott VLIW processzoron! Határozza meg az utasításcsoportok tartalmát úgy, hogy a lehető legkevesebb utasításra legyen szükség! Ha egyes utasításokat több helyre is be lehet írni, mindig a kisebb sorszámú utasítást rendelje a kisebb sorszámú műveleti egységhez! (3p)

Utasításcsoport:	MEM 1:	MEM 2:	INT 1:	INT 2:
1:	i1 ✓	i4 ✓		
2:				
3:				
4:			i2 ✓	i5 ✓
5:			i3 ✓	i6 ✓
6:			i7 ✓	

(Nem feltétlenül kell minden utasításcsoportot kitölteni!)

Your answer is correct.

12 kérdés

Helyes

3,00/3,00 pont

Legyen adott az alábbi utasítás sorozat:

```
i1: R1 ← MEM[R0+0]
i2: MEM[R1+0] ← R3
i3: R3 ← R0 + 4
i4: R1 ← MEM[R3+0]
```

Szüntesse meg a WAW és WAR adat-egymáshatásokat regiszter átnevezés segítségével! A regiszter átnevezést végezze el minden utasításra szisztematikusan, ott is, ahol nem lenne rá szükség! A szükséges fizikai regisztereket jelölje U0, U1, U2, ... stb. Vegye figyelembe, hogy az alábbi regiszter leképző tábla a kezdeti állapotot is tartalmazza! Ha új fizikai regiszterre van szükség, válassza mindig a táblázatban szereplő legnagyobb után közvetlen következőt! Minden egyes programsor feldolgozása után frissítse a regiszter leképző táblát (csak a megváltozott bejegyzést kell beírni)! A mezőkbe sehova ne írjon szóköz karaktert! (3 pont)

Az utasítássorozat átnevezés után:

i1': U13 ✓ ← MEM[U4+0] ✓
 i2': MEM[U13+0] ✓ ← U12 ✓
 i3': U14 ✓ ← U4+4 ✓
 i4': U15 ✓ ← MEM[U14+0] ✓

Regiszter-leképző tábla:

	Kezdetben	i1	i2	i3	i4
R0:	U4				
R1:	U7	U13 ✓			U15 ✓
R2:	U3				
R3:	U12			U14 ✓	

Your answer is correct.