

Név:

Neptun:

Aláírás:

Rendelkezésre álló idő: **120 perc**. Használható segédeszközök: számológép
Összpontszám: **60 pont**. Szükséges kiskérdésekből: **9 pont**, összesen: **24 pont**

Feladattípusok:

T *Teszt*: a helyes állítások bejelölendők (akár több is). Tökéletes válasz: 1 pont, van bejelölt rossz válasz: -1 pont, nincs bejelölt rossz válasz, de nincs minden jó bejelölve: 0 pont

Ö *Összekötős*: A bal oldali felsorolás elemeinek összekötése a jobb oldali felsorolás elemeivel
Tökéletes válasz: 1 pont, nem tökéletes válasz, vagy a válasz hiánya: 0 pont

K *Kiegészítő*: A kérdésre adott válasz beírása a keretbe, illetve a pontozott vonalra, tömören
Tökéletes válasz: 2 pont, nem tökéletes válasz, vagy a válasz hiánya: 0 pont

Ha a "T" típusú kérdésekből negatív pontszám jönne ki, a többi pontszámot **nem** csökkentjük.

T:	Ö:
K:	Sz1:
Sz2:	Sz3:
Sz4:	Sz5:
Sz6:	

1. Mely kivételek fordulhatnak elő az EX fázisban?

T

- Laphiba
- Érvénytelen utasítás
- Aritmetikai hiba
- Egyik sem

2. Mely állítások igazak az ugrási cím buffer-re (BTB-re)?

T

- Cache-ként működik, az utasításszámláló a cache tag
- A rendszermemóriában tároljuk
- A processzorban tároljuk
- Egyik állítás sem igaz

3. Milyen arbitrációt használ a PCI Express?

T

- Nem használ arbitrációt
- Párhuzamos arbitrációt
- Önkiválasztó arbitrációt
- Rejtett arbitrációt

4. Tegyük fel, hogy a CPU támogatja az interruptok használatát, és ez engedélyezve is van. Mely állítások igazak? (Igaznak tekintünk egy összetett állítást, ha minden részállítása igaz, és a köztük lévő reláció is helytálló)

T

- A ritkán tevékeny perifériákat célszerű polling-gal kezelni, mert úgy kisebb a processzor terhelése
- A ritkán tevékeny perifériákat célszerű polling-gal kezelni, mert úgy gyorsabban értesülünk a periféria megváltozott állapotáról
- Egyik állítás sem igaz

5. Mi a memória blokkok idő előtti cache-be töltésének célja?

T

- Hogy időben kiderüljön, hogy egy blokk cache szemét-e
- Hogy a blokk még azelőtt a cache memóriába kerüljön, mielőtt először meghivatkoznánk
- Hogy egy memória blokkot a rendszermemória megkerülésével tudjunk a cache-be betölteni

6. Mely állítások igazak a fizikai regiszterekre?

T

- A programozó elől el vannak takarva
- Több van belőle, mint a logikai/architektúrais regiszterből
- Az utasításkészlet architektúra definiálja a darabszámukat és elnevezésüket
- A programozó ezeket, és nem a logikai/architektúrais regisztereket használja

7. Mit jelent a "laphiba" fogalma a virtuális memóriakezeléssel kapcsolatban?

T

- A futó program olyan lapra hivatkozott, ami nincs bent a virtuális memóriában
- A futó program olyan lapra hivatkozott, amire vonatkozó információ nincs a TLB-ben
- A futó program olyan lapra hivatkozott, ami nincs bent a fizikai memóriában
- A futó program olyan keretre hivatkozott, ami nincs bent a fizikai memóriában

8. Mely állítások igazak az SRAM memóriára?

T

- Egy bit tárolásához 1 tranzisztor és 1 kondenzátor szükséges
- A DRAM-hoz képest gyorsabb az írás és olvasás művelete
- A tápellátás megszűnésével a tárolt adatok elvesznek
- A DRAM-hoz képest sűrűbben helyezhetők el a cellák a tárolómátrixban

9. Jelölje be, mely fázisok végeznek tényleges munkát egy store (memóriába írás) utasítás feldolgozása közben a tanult 5 fokozatú pipeline-ban!

T

- WB (Write Back)
- ID (Instruction Decode)
- EX (Execute)
- IF (Instruction Fetch)

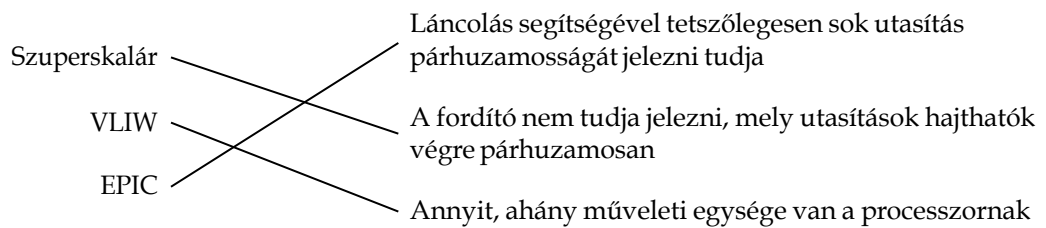
10. Az alábbiak közül melyek a RISC utasításkészletek jellemzői?

T

- Nagy számú utasítás az utasításkészletben
- Regiszter-memória műveletek
- Redundancia-mentesség
- Kevés címzési mód

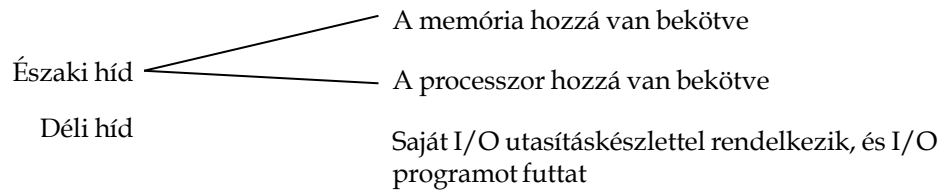
11. Legfeljebb hány utasítás párhuzamos végrehajthatóságát tudja jelezni a fordító a processzornak?

Ö



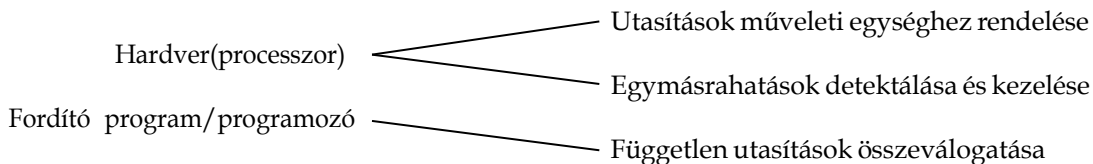
12. Mely állítások igazak az északi- és déli hídra híd alapú rendszerekben?

Ö



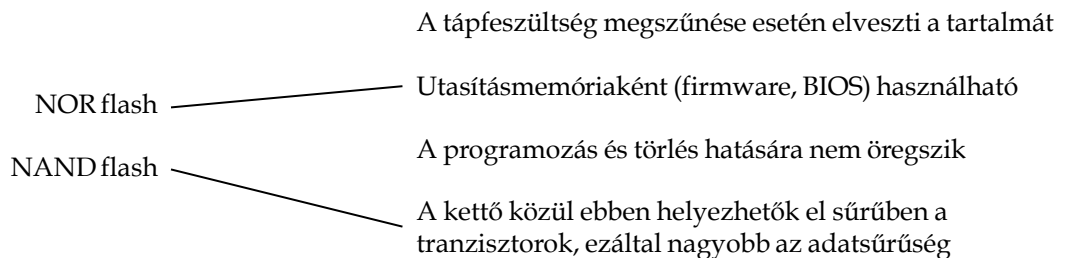
13. Mely tevékenységeket végzi a fordítóprogram/programozó, és melyeket a hardver EPIC architektúrában?

Ö



14. Mely állítások igazak a NOR, és melyek a NAND flash memóriára?

Ö



15. Soroljon fel 1-1 konkrét USB perifériát, mely az alábbi adatátviteli módokat használja!

K

Izokron: Hangszóró

Bulk: Nyomtató.....

Interrupt: Billentyűzet.....

16. Mire szolgál a "strip mining" a vektorprocesszorokban?

K

Ha a vektor mérete nagyobb, mint amennyit támogat a vektorprocesszor (VML), feldaraboljuk a vektortunkat megfelelő méretekre, így ezeken a szeleteken már tud dolgozni a vektorprocesszor

17. Adja meg, hogy egy 6 bites globális előzményregisztert és globális PHT-t használó elágazásbecslő eljárás,

K

hány darab PHT-t használ: Egyet.....
hány bejegyzéses PHT-t használ: 64.....

18. Állítsa sorba a teljesen összekötött (TÖ), a gyűrű (GY), valamint a kétdimenziós tórusz (2T) összeköttetés-hálózat topológiákat költség szerinti skálázhatóság alapján! Kezdje a felsorolást azzal, amelyik költség szerint a legrosszabbul skálázható!

K

Sorrend: TÖ > 2T > GY

19. Amdahl törvénye szerint elméletileg mennyivel gyorsabban futhat egy program egy végtelen sok processzorból álló multiprocesszoros rendszerben, mint egy 1 processzoros rendszerben, ha a program 3/4-e csak szekvenciálisan futtatható?

K

A gyorsulás mértéke: Az alap futási idő egyharmadszorosával lehet gyorsabb a program

20. Egy merevlemez 2 db lemezt tartalmaz, melyek mindkét oldalán van adathordozó réteg. Minden adathordozó rétegen 15000 sáv található, minden sávban 1500 szektorral. A szektorok mérete 500 bájt. Adja meg a merevlemez kapacitását bájtban:

K

Méret = 45×10^9 bájt.....

1. számpélda A processzor a program végrehajtása közben az alábbi sorrendben, az alábbi címeken lévő feltételes ugró utasításokat értékeli ki (a zárójelben "T" jelzi, ha az ugrás ténylegesen bekövetkezett, és "N", ha nem):

- 500 (N), 500 (N), 236 (T), 236 (T), 940 (N), 500 (N), 940 (T), 940 (N)

A processzor egyszerű, 2 bites állapotgépre alapozott lokális dinamikus elágazásbecslést használ, és 8 ugró utasítást tud követni. A PHT becsléshez használt bejegyzését az utasításszámláló utolsó 3 bitje jelöli ki. A PHT-ben tárolt állapotváltozók 2 bitesek (00,01: ne ugorj, 10,11:ugorj). (Az állapotátmenetek megegyeznek az órán tanultakkal)

Kérdések:

(a) Mekkora a PHT teljes mérete (bitben megadva)? (1p)

(a) 16

(b) A becslő állapotváltozóinak kiindulási értékei az alábbi táblázat 0.-7. soraiban láthatók. Hogyan alakulnak az állapotváltozók a fenti feltételes ugró utasítás-sorozat végrehajtása során? (3p)

(c) A becslő melyik utasítás esetén javasol ugrást (1) és melyeknél nem (0)? (1p)

(d) Mely becslések voltak jók (J) és melyek hibásak (H)? (1p)

	Kezdetben	500 (N)	500 (N)	236 (T)	236 (T)	940 (N)	500 (N)	940 (T)	940 (N)
0.	01								
1.	00								
2.	10								
3.	01								
4.	01	00	00	01	10	01	00	01	00
5.	00								
6.	11								
7.	11								
(c)	-	0	0	0	0	1	0	0	0
(d)	-	J	J	H	H	H	J	H	J

(A táblázatban elég csak a megváltozott mezőket kitölteni!)

2. számpélda Legyen adott az alábbi utasítás sorozat:

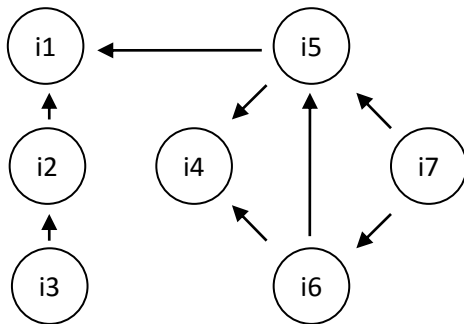
```

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)

(a) Rajzolja fel az utasítások végrehajtásához tartozó precedencia gráfot! (Gondoljon a WAW és WAR függőségekre is!) (3p)



Minden felrajzolt nyíl RAW függőséget jelöl. WAW és WAR függőség nincs.

(b) Ütemezze a megadott utasítássorozatot a megadott VLIW processzoron! Határozza meg az utasításcsoportok tartalmát! (3p)

	Mem1:	Mem2:	Int1:	Int2:
1	i1	i4	NOP	NOP
2	NOP	NOP	NOP	NOP
3	NOP	NOP	NOP	NOP
4	NOP	NOP	i2	i5
5	NOP	NOP	i3	i6
6	NOP	NOP	i7	NOP

(Nem feltétlenül kell minden utasításcsoportot kitölteni!)

3. **számpélda** Egy virtuális tárkezelésre képes processzor 16 bites virtuális és 15 bites fizikai címetek támogat. A lapméret 4096 bájt ($=2^{12}$). A címfordításhoz kétszintű laptáblát használ, valamint egy 4 bejegyzéses, LRU algorit-mussal menedzselte teljesen asszociatív TLB-t.

(a) Hány lapból áll a virtuális memória?

(a) 16

(b) Hány keretből áll a fizikai memória?

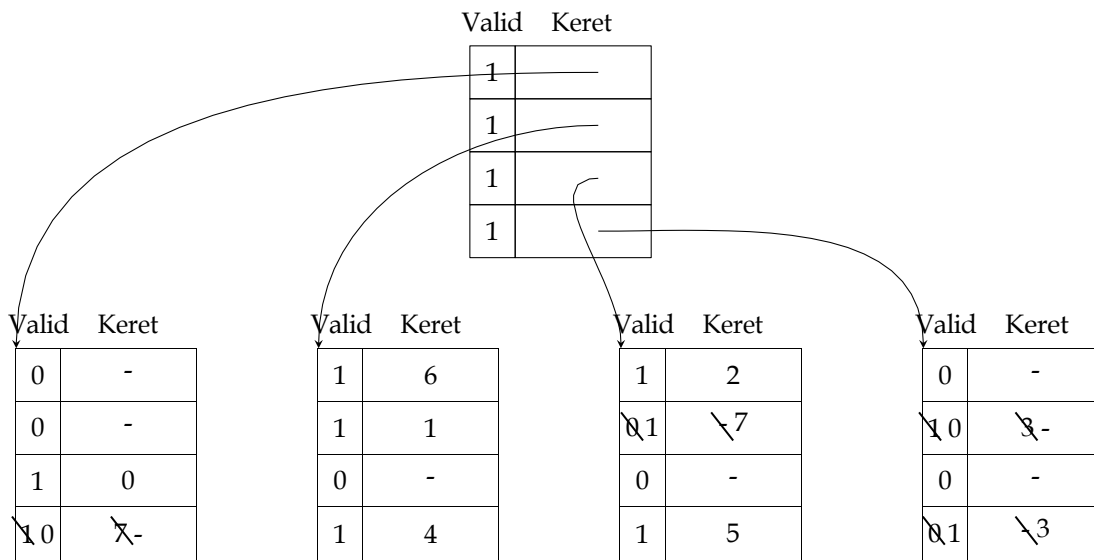
(b) 8

(c) A futó program sorban egymás után az alábbi lapokra hivatkozik:

- 11, 9, 15, 11, 5

Adja meg a laptábla és a TLB tartalmának alakulását! Ha a laptábla változik, húzza át a régi tartalmát és írja mellé az újat! A TLB tartalmát minden lap-hivatkozás után külön táblázatba írja (elegendő csak a megváltozott mezőket beírni)! Ha az operációs rendszer egy újabb lapot akar a fizikai memóriában elhelyezni, akkor egy bent lévő lapot ki kell vennie onnan. A soron következő kiszemelt áldozatok a 3, 13, 2, 8 lapok (ebben a sorrendben). Lapcsere esetén, ha a memóriából kikerült laphoz tartozik TLB bejegyzés, akkor az operációs rendszer azt az egy bejegyzést invalidálja. (6 pont)

Laptábla:



TLB:

Kezdőállapot				1. lépés (11)				2. lépés (9)			
Valid	Lap	Keret	Kor	Valid	Lap	Keret	Kor	Valid	Lap	Keret	Kor
1	5	1	1				2				3
1	7	4	2				3				4
1	4	6	3				4		9	7	1
1	8	2	4		11	5	1				2
3. lépés (15)				4. lépés (11)				5. lépés (5)			
Valid	Lap	Keret	Kor	Valid	Lap	Keret	Kor	Valid	Lap	Keret	Kor
			4				5				1
	15	3	1				2				3
			2				3				4
			3				1				2

(Puska: 0 = 0000, 1 = 0001, 2 = 0010, 3 = 0011, 4 = 0100, 5 = 0101, 6 = 0110, 7 = 0111, 8 = 1000, 9 = 1001, 10 = 1010, 11 = 1011, 12 = 1100, 13 = 1101, 14 = 1110, 15 = 1111)

4. számpélda Egy számítógép rendszermemóriája szinkron DRAM-ra épül. A memóriavezérlőhöz beérkező olvasási kérések az alábbi sor, oszlop koordinátákra vonatkoznak:

- (8. sor, 3. oszlop), (9. sor, 4. oszlop), (8. sor, 4. oszlop)

Adja meg a fenti kérésekhez tartozó, memóriavezérlő által kiadott DRAM parancsokat (sorrendhelyesen), feltéve, hogy a kérések egyazon bankra és egyazon rank-re vonatkoznak, és feltéve, hogy bankonként csak egyetlen sor lehet nyitva. Kezdetben a 9. sor nyitott állapotban van. Az utolsó parancs után a memóriavezérlő ne zárja le a nyitott sort! (4 pont)

FCFS ütemezés szerint:

1.: PRECHANGE
 2.: ACTIVATE 8.....
 3.: READ 3.....
 4.: PRECHANGE.....
 5.: ACTIVATE 9.....
 6.: READ 4.....
 7.: PRECHANGE.....
 8.: ACTIVATE 8.....
 9.: READ 4.....

FR-FCFS ütemezés szerint:

1.: READ 4.....
 2.: PRECHANGE.....
 3.: ACTIVATE 8.....
 4.: READ 3.....
 5.: READ 4.....
 6.:
 7.:
 8.:
 9.:

(Nem feltétlenül szükséges minden sort kitölteni)

5. számpélda Egy speciális számítógép egy melegezésre hajlamos, de ideiglenesen kikapcsolható perifériát, valamint egy hőmérséklet érzékelőt tartalmaz. A hőmérséklet-érzékelő a hőmérsékleti küszöb túllépésekor megszakítást kér, a megszakítás kiszolgálása 300 órajelet vesz igénybe. Túlmelegedés átlagosan percenként kétszer következik be, amikor is a processzor kikapcsolja a túlhevült perifériát, majd nem sokkal ezután újra bekapcsolja azt. A ki- és bekapcsolásra fordított idő olyan kicsi, hogy elhanyagoljuk.

(a) Másodpercenként átlagosan hány órajel terhelést jelent a hőmérséklet-érzékelő kezelése?

(a) 10

(b) Legalább mekkora órajelfrekvencián kell hajtani a processzort, hogy az a hőmérséklet-érzékelő kezelését maradéktalanul el tudja látni? Feltesszük, hogy a processzornak egyéb célokra másodpercenként átlagosan $5 \cdot 10^6$ órajelet igénylő terhelése is van.

(b) 5000600

(4 pont)

6. **számpélda** Legyen adott az alábbi utasítás sorozat, ahol a lebegőpontos regiszterek D-vel, a fix pontosak R-el kezdődnek:

i1: D3 ← D0 + D4
 i2: D1 ← D3 * D2
 i3: D5 ← MEM[R0+4]
 i4: D3 ← D3 + D5

Az utasítássorozatot egy, az órán tanult pipeline-t használó processzor hajtja végre. A pipeline minden utasítás végrehajtását 5 részműveletre bontja: betölti (IF), dekódolja (ID), végrehajtja a vonatkozó aritmetikai műveleteket, majd a vonatkozó memóriaműveleteket (MEM), végül a regisztertárolóba írja az eredményregiszter értékét (WB). Minden utasítás mind az 5 fázison átesik, függetlenül attól, hogy szüksége van-e rá. Az IF, ID, MEM és WB fázisok késleltetése 1 órajel. Az aritmetikai művelet lehet egész művelet, mely késleltetése 1 órajel (EX, a címszámítást is ez az egység végzi), lebegőpontos összeadás, melynek késleltetése 3, iterációs ideje 1 (A0, A1, A2), valamint lebegőpontos szorzás, melynek késleltetése 5, iterációs ideje pedig 1 (M0, M1, M2, M3, M4). A három aritmetikai egység képes párhuzamos működésre, és az utasítások soron kívüli befejezésére, ha ennek semmilyen szemantikai vagy egymásrahatásbeli akadályja nincs. Minden forwarding út használata megengedett. Ha bármilyen egymásrahatás az utasítás megállítását igényli, az utasítás mindig a legutolsó olyan fázisban áll meg, ameddig egymásrahatás nélkül eljut.

(a) Azonosítsa az utasítások között fennálló adat-egymásrahatásokat, és töltsse ki az alábbi táblázatokat (X-el jelölje, ha két utasítás között egymásrahatást talál)! (2p)

RAW egymásrahatások:

	i1	i2	i3	i4
i1				
i2	x			
i3				
i4	x		x	

WAW egymásrahatások:

	i1	i2	i3	i4
i1				
i2				
i3				
i4	x			

WAR egymásrahatások:

	i1	i2	i3	i4
i1				
i2				
i3				
i4		x		

(b) Adja meg az utasítássorozat ütemezését (melyik utasítás mikor melyik fázisban van)! Ha szünetet kell beiktatni, jelezze, hogy mi az oka! Használja az alábbi jelöléseket:

- **A***: a szünet oka adategymásrahatás
- **F***: a szünet oka feldolgozási egymásrahatás
- **P***: a szünet oka procedurális egymásrahatás

A megoldását írja az alábbi táblázatba! Minden egyes sor utolsó bejegyzésében jelölje, hogy szükség van-e forwarding-ra, ha igen, melyik regiszter értékéről van szó! (3 pont)

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	Forwarding
i1	IF	ID	A0	A1	A2	MEM	WB								-
i2		IF	ID	A*	A*	M0	M1	M2	M3	M4	MEM	WB			A2/MEM
i3			IF	F*	F*	ID	EX	MEM	WB						-
i4						IF	ID	A*	A0	A1	A2	MEM	WB		MEM/WB

(c) Rendezze át az utasítássorozatot úgy, hogy az gyorsabban fusson le, mint eredetileg! (1 pont)

Az átrendezett utasítássorozat: i1, i3, i2, i4

(d) Adja meg az átrendezett utasítássorozathoz tartozó ütemezést is! (2p)

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	Forwarding
i1'	IF	ID	A0	A1	A2	MEM	WB								-
i2'		IF	ID	EX	MEM	WB									-
i3'			IF	ID	A*	M0	M1	M2	M3	M4	MEM	WB			A2/MEM
i4'				IF	F*	ID	A0	A1	A2	MEM	WB				A2/MEM MEM/WB