

Question 1
Correct
Mark 2.00 out of 2.00
Flag question

Milyen állandó (!) összetevőkből állnak a keresési műveletek fizikailag indexelt, fizikai tag-eket használó direkt leképzésre alapozott cache esetén?

- IGAZ ✓ Indexelésből
- IGAZ ✓ A tag(ek) összehasonlításából
- HAMIS ✓ Személygyűjtésből
- HAMIS ✓ Lapcsereből

Question 2
Partially correct
Mark 0.50 out of 2.00
Flag question

Mely állítások igazak a fizikai regiszterekre?

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

Question 3
Not answered
Marked out of 2.00
Flag question

Egy utasításpipeline-ban az utasításvégrehajtás mely fázisai történnek a pipeline frontend-ben?

- ✗ Utasítások lehívása
- ✗ Utasítások elhelyezése az utasítástárolóban
- ✗ Utasítások ütemezése a megfelelő végrehajtó egység felé
- ✗ Utasítások tényleges végrehajtása

Question 4
Partially correct
Mark 0.50 out of 2.00
Flag question

A tanult 5 fokozatú, 1 szélességű pipeline-ban, azonos késleltetésű műveleti egységeket, és sorrendi végrehajtást feltételezve, mely adatfüggőségek igényelnek kezelést?

- IGAZ ✓ Read After Write
- HAMIS ✓ Read After Read
- IGAZ ✗ Write After Read
- IGAZ ✗ Write After Write

Question 5
Correct
Mark 2.00 out of 2.00
Flag question

Mely kivételek fordulhatnak elő a WB fázisban?

- HAMIS ✓ Aritmetikai hiba
- HAMIS ✓ Laphiba
- HAMIS ✓ Érvénytelen utasítás
- HAMIS ✓ Védelmi hiba

Question 6
Correct
Mark 2.00 out of 2.00
Flag question

Az alábbiak közül mely arbitrációs eljárások elosztottak?

- HAMIS ✓ Daisy chain-re alapozott arbitráció
- HAMIS ✓ Párhuzamos arbitráció
- IGAZ ✓ Önkiválasztó arbitráció
- IGAZ ✓ Ütközés-detekálásra alapozott arbitráció

Question 7
Partially correct
Mark 0.50 out of 2.00
Flag question

Mely tevékenységeket végzi a fordítóprogram/programozó az EPIC architektúrában?

- ✗ Független utasítások összeválogatása
- ✗ Egymáshatások detektálása és kezelése
- ✗ Utasítások műveleti egységhez rendelése
- HAMIS ✓ Az utasítások végrehajtása

Question 8
Correct
Mark 2.00 out of 2.00
Flag question

X

Mely állítások igazak a PCI-ra, és melyek a PCI Express-re?

	PCI Express	PCI
Pont-pont kapcsolatokon alapuló adatátvitel	X ✓	
Párhuzamos adatátvitel		X ✓
Osztott közeg alapú adatátvitel		X ✓
Soros adatátvitel	X ✓	

Question 9
Correct
Mark 2.00 out of 2.00
Flag question

X

Adja meg a három tanult cache szervezés tulajdonságait!

	Teljesen asszociatív szervezés	N-utas asszociatív szervezés	Direkt leképzés
--	--------------------------------	------------------------------	-----------------

Question 9
Correct
Mark 2.00 out of 2.00
Flag question

X

Adja meg a három tanult cache szervezés tulajdonságait!

	Teljesen asszociatív szervezés	N-utas asszociatív szervezés	Direkt leképzés
Megkötések nélküli blokk elhelyezés	X ✓		
A három közül ebben a legszélesebb a cache "tag"	X ✓		
A három közül ebben a legkevesebb az egyidejűleg működő komparátorok száma			X ✓

Question 10
Correct
Mark 2.00 out of 2.00
Flag question

X

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

	CISC	RISC
Tomorebb programkód	X ✓	
Tipikusan több regiszter		X ✓
Változó hosszúságú utasításkódolás	X ✓	
Külön I/O és memóriakezelő utasítások	X ✓	



Adja meg, mely tulajdonságok jellemzik az SRAM, és melyek a DRAM memóriákat!

	DRAM	SRAM
A tápellátás megszűnésével tartalma elvész	X ✓	X ✓
Tartalmát periodikusan frissíteni kell	X ✓	
Egységnyi felületen a kettő közül ezzel érhető el nagyobb tárolási kapacitás	X ✓	
1 bit tárolásához 6 kondenzátor szükséges		

Your answer is correct.

Mit jelent a TLB lefedettség a virtuális memóriával kapcsolatban?

a virtuális memória mekkora része van lefedve a tlb-vel (szóval az x GB virtuális memóriának mekkora százalékát/részét tudjuk elérni TLB segítségével azaz tlb hiba nélkül)

Comment:

Sorolja fel a Neumann architektúra három komponensét!

Mekkora a PHT mérete, ha az utasításszámláló utolsó 6 bite jelöli ki a használt bejegyzést, és az ugrási hajlandóság nyomon követésére 4 állapotot különböztetünk meg?

A PHT mérete (bitben): 128 ✓

Amdahl törvénye szerint mennyivel gyorsabban fut egy program egy 10 processzorból álló multiprocesszoros rendszerben, mint egy 1 processzoros rendszerben, ha a program harmada csak szekvenciálisan futtatható?

A gyorsulás mértéke: 2 ✗ [2.5 or 2,5]

Legyen adott az alábbi utasítás sorozat:

i1: R1 ← MEM[R0+0]
i2: R2 ← R1 * R0
i3: MEM[R2+0] ← R1
i4: R2 ← R0 * R3



Az alábbi táblázatokban jelölje meg, hogy mely utasítások között áll fenn RAW, WAR, illetve WAW függőség!

RAW függőség:					WAR függőség:					WAW függőség:				
i1	i2	i3	i4		i1	i2	i3	i4		i1	i2	i3	i4	
i1					i1					i1				
i2	X ✓				i2					i2				
i3	X ✓		X ✓		i3					i3				
i4					i4			X ✓		i4	X ✓			

Szűntesse meg a WAW és WAR adat-egymásrahatásokat 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 mezőbe sehoval ne írjon szököző karaktert! (3 pont)

Az utasítássorozat átnevezés után:

i1*: U9 ✓ ← MEM[U0+0] ✓
i2*: U10 ✓ ← U9*U0 ✓
i3*: MEM[U10+0] ✓ ← U9 ✓
i4*: U11 ✓ ← U0*U3 ✓

Regiszter-leképző tábla:

	Kezdetben	i1	i2	i3	i4
R0:	U0				
R1:	U8	U9 ✓			
R2:	U5		U10 ✓		U11 ✓
R3:	U3				

Your answer is correct.

Ha van egy ideális processzorunk, melyben a pipeline szélessége végtelen, és minden utasítást egyetlen órajel alatt feldolgoz, mennyi ideig tart a fenti utasítássorozat végrehajtása

Regiszterátnevezés nélkül: 4 ✓ órajel

Regiszterátnevezéssel: 3 ✓ órajel

(A processzor minden egyes órajelben minden olyan utasítást végre tud hajtani egyszerre, mely a függőségek figyelembe vételével lehetséges)

i1: R1 ← MEM[R0+0]
i2: MEM[R1+0] ← R3
i3: R0 ← R0 + 4
i4: R3 ← R3 * R2
i5: R4 ← R0 + R3

Az utasítássorozatot egy 6 fokozatú utasítás pipeline-t használó processzor hajtja végre. A pipeline minden utasítás végrehajtását 5 részmuveletre bontja: betölti (IF), dekódolja (ID), végrehajtja a vonatkozó aritmetikai műveletet (EX), majd a vonatkozó memóriaműveletet (MEM), végül a regisztertárolóba írja az eredményregiszter értékét (WB). Az IF, EX, MEM és WB fázisok kíséltetése 1 órajel, az ID fázis kíséltetése 2 órajel, de iterációs ideje 1 órajel (vagyis 2 pipeline fokozatként jelenik meg: ID0 és ID1). Az utasítás dekódolására az első órajelben kerül sor, míg az operandusok betöltése a regisztertárolóból a második órajel második felében történik, illetve, ha egy utasítás egymásrahatás miatt több időt tölt az ID fázisban, akkor az ott töltött utolsó órajel második felében. Minden utasítás végrehajtása mind az 5 fázison átesik, függetlenül attól, hogy szüksége van-e rá. 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.

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 oszlopába írja be, ha az utasítás feldolgozása során forwarding-ra volt szükség, hogy melyik pipeline regiszterekből kell kiolvasni a kívánt értéket (ha több ilyen is van, vesszővel elválasztva, szököző nélkül)! (3p)

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	Forwarding
I1	IF	ID0	ID1	EX	MEM	WB							
I2		IF	ID0	ID1	A*	EX	MEM	WB					MEMWB ✓
I3			IF	ID0	F*	ID1	EX	MEM	WB				
I4				IF	F*	ID0	ID1	EX	MEM	WB			
I5					IF	F*	ID0	ID1	A*	EX	MEM	WB	MEMWB ✗ (EX/MEM, MEMWB or MEMWB, EX/MEM)
					(blank)	(IF)		(EX)	(MEM)	(WB)	(blank)		

Your answer is partially correct.

Question 20

Correct

Mark 1.00 out of 1.00

Flag question

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

Az átrendezett utasítássorozat: I1 ✓ - I3 ✓ - I2 ✓ - I4 ✓ - I5 ✓

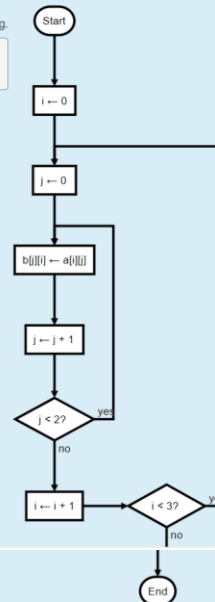
Information

Flag question

Egy programban egy algoritmus a jobb oldalt látható folyamatábra szerint működik.

Az algoritmus közelítőleg (bár nem pontosan) az alábbi C programkódnak felel meg.

```
for (int i=0; i<3; i++)
    for (int j=0; j<2; j++)
        b[j][i] = a[i][j];
```



Question 21

Correct

Mark 3.00 out of 3.00

Flag question

j<2 j<3 nem_ugrik_(no) ugrik_(yes)

Az alábbi táblázatban adja meg, hogy az algoritmus kezdetétől a végéig, sorrendhelyesen, mely feltételes ugró utasítások kerülnek sorra, és mi az ugrások kimenetelei!

	1. elágazás	2. elágazás	3. elágazás	4. elágazás	5. elágazás	6. elágazás	7. elágazás	8. elágazás	9. elágazás
Melyik feltételes ugró utasítás:	j<2	j<2	i<3	j<2	j<2	i<3	j<2	j<2	i<3
Mi az ugrás kimenetele:	ugrik_(yes)	nem_ugrik_(no)	ugrik_(yes)	ugrik_(yes)	nem_ugrik_(no)	ugrik_(yes)	ugrik_(yes)	nem_ugrik_(no)	nem_ugrik_(no)

Question 22

Correct

Mark 4.00 out of 4.00

Flag question

Tételezzük fel (az előbbi algoritmus tényleges viselkedésétől függetlenül), hogy a program a futása során az alábbi utasításslámláló értékeken érint feltételes ugró utasításokat (ebben a sorrendben):

- 924(N), 924(N), 848(T), 924(N), 924(N), 848(T)

A zárójelben szereplő "T" jelzés az ugrás bekövetkezésére utal, az "N" pedig azt jelzi, hogy az ugrási feltétel nem teljesül, nem volt ugrás.

Processzorunk korrelált elágazás-becslő egysége egyetlen globális ugrási-előzmény regisztert (GBHR-I) használ, ami egy 3 bites shift-regiszter, melynek LSB bitjére lép be a ténylegesen megvalósult ugrás-esemény kódja (1: történt ugrás, 0: nem történt ugrás). A (globális) PHT állapotváltozói 2 bitesek (00,01: ne ugorj, 10,11:ugorj). (Az állapotátmenetek megegyeznek az órán tanultakkal)

Kérdések:

- Adja meg a globális előzmény regiszter (GBHR) tartalmát binárisan a feltételes ugró utasítások végrehajtása után, ha a kezdeti tartalma (101) volt! (Az LSB a jobb szélen!!!)
- A PHT állapotváltozóinak kiindulási értékei az alábbi táblázat 0.-7. soraiban láthatók. Hogyan alakulnak az állapotváltozók a fenti feltételes ugró utasítás végrehajtása során?
- A "becslés" sor kitöltésével jelezze, hogy a becsló melyik utasítás esetén javasol ugrást (1) és melyeknél nem (0)?
- A "sikereség" sor kitöltésével jelezze, mely becslések voltak jók (J) és melyek hibásak (H)?

	Kezdetben	924 (N)	924 (N)	848 (T)	924 (N)	924 (N)	848 (T)
GBHR	101	010 ✓	100 ✓	001 ✓	010 ✓	100 ✓	001 ✓
PHT[0]	00						
PHT[1]	01				00 ✓		
PHT[2]	10		01 ✓			00 ✓	
PHT[3]	01						
PHT[4]	10			11 ✓			11 ✓
PHT[5]	10	01 ✓					
PHT[6]	10						
PHT[7]	11						
Becslés:	-	1 ✓	1 ✓	1 ✓	0 ✓	0 ✓	1 ✓
Sikereség:	-	H ✓	H ✓	J ✓	J ✓	J ✓	J ✓

A táblázatba elegendő csak a megváltozott bejegyzéseket beírni!

Question 23

Partially correct

Mark 2.00 out of 6.00

Flag question

Egy processzor 17 bites virtuális és 15 bites fizikai címet támogat. A lapméret 8192 bájt ($=2^{13}$). A processzor 2 szintű hierarchikus laptáblát használ, minden szintre egyforma széles indexekkel és 8 bites bejegyzésekkel.

Legyen a laptábla aktuális állapota a következő:

Elsőszintű laptábla:

V	Cím
00:	1
01:	1
10:	1
11:	1

Másodszintű laptáblák:

V	Cím	V	Cím	V	Cím	V	Cím
00:	?	00:	?	00:	1	00:	?
01:	?	01:	2	01:	3	01:	?
10:	0	10:	?	10:	?	10:	?
11:	?	11:	?	11:	0	11:	?

A kérdések a következők:

a) Mekkora a teljes laptábla elméletileg lehetséges legnagyobb összegzett mérete (bájtban)? ☒ [20]

Mekkora a teljes laptábla - megfelelő tartalom mellett elérhető - elméletileg lehetséges legkisebb összegzett mérete (bájtban)? ☒ [8]

b) Hány lapból áll a virtuális memória? ☒

Hány keretből áll a fizikai memória? ☒

c) Hol található

- A 8-as lap: ☒ [keret]
- A 10-es lap: ☒

(Lehetséges válaszok: "háttértár", vagy "keret", "keret1", stb. (szóköz nélküli), kérjünk ebben a formában beírni a választ a mezőbe)

d) Hányas lap található az alábbi kereteken:

- A 2-es kereten: ☒
- A 3-as kereten: ☒

(A válasz csak egy 0-tól kezdődő szám legyen)

e) Módosítsa a laptábla tartalmát a következők szerint:

- Az 8-as lap a háttértáron van
- A 3-as lap az 1-es kereten van

A módosításokat írja az alábbi laptáblába, elegendő csak a megváltozott mezőket beírni!

Elsőszintű laptábla:

V	Cím
00:	1
01:	1
10:	1
11:	1

Másodszintű laptáblák:

V	Cím	V	Cím	V	Cím	V	Cím
00:		00:		00:	0 ✓ 2/1 ✗ [? or 1 or blank]	00:	
01:		01:		01:		01:	
10:		10:		10:		10:	
11:	1 ✓ 1 ✓	11:		11:		11:	

Question 24

Partially correct

Mark 4.00 out of 6.00

Flag question

Egy processzor direkt szervezésű cache-t használ, melynek teljes mérete 2048 bájt ($=2^{11}$). A blokkméret 256 bájt ($=2^8$). A processzor 24 bites fizikai és 32 bites virtuális címekkel rendelkezik. A cache virtuális címekkel indexel, és fizikai tag-eket használ.

(a) Hány bitesek a cache tag-ek? ☒ [13]

(b) Hány bitesek a cache indexek? ☒

(c) Hány összehasonlítást kell elvégezni egy időben minden egyes kereséskor? ☒

(d) Hány bit szélességű összehasonlítást kell végezni? ☒ [13]

(e) Az épp futó program a rendszermemória alábbi blokkjait hivatkozza:

- 478, 51

Hogy változik a cache memória tartalma, ha az alábbi kiinduló állapotokban a megadott blokkra hivatkozik a program? Az üres mező használaton kívüli (érvénytelen) tartalomra utal. Ha több lehetőség is van egy blokk elhelyezésére, válassza mindig a legfelsőt. Elegendő csak a megváltozott mezőket beírni!

A 478 beszúrása után:

Kiindulóállapot:	
184	
249	
123	
76	
413	
334	
183	

478
✓

A 51 beszúrása után:

Kiindulóállapot:	
184	
249	
123	
76	
413	
334	
183	

51
✓

Your answer is partially correct.