

<b>Started on</b>	Monday, 13 June 2022, 10:05 AM
<b>State</b>	Finished
<b>Completed on</b>	Monday, 13 June 2022, 11:09 AM
<b>Time taken</b>	1 hour 4 mins
<b>Grade</b>	47.75 out of 60.00 (80%)

Question **1**

Partially correct

Mark 1.25 out of 2.00

Ha a merevlemez nem alkalmaz ZBR adattárolást, azaz minden sáv azonos számú szektorból áll, akkor:

- HAMIS  A szektorok bájtban mért mérete a lemez széle felé csökken
- HAMIS  Az adatsűrűség mindenütt egyforma
- HAMIS  A merevlemez a lemez széle felé haladva megváltoztatja a diszk forgási sebességét
- HAMIS  A szektorok leolvasási sebessége mindenütt egyforma

Question **2**

Correct

Mark 2.00 out of 2.00

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

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

Question **3**

Correct

Mark 2.00 out of 2.00

Milyen arbitrációt használ a PCI?

- IGAZ  Centralizált arbitrációt
- IGAZ  Párhuzamos arbitrációt
- HAMIS  Önkiválasztó arbitrációt
- HAMIS  Ütközésetektáláson alapuló arbitrációt

Question **4**

Partially correct

Mark 0.50 out of 2.00

Mely állítások teljesülnek az USB "interrupt" adatátviteli módjára?

- IGAZ  Az USB rendszer garantálja a hibamentes átvitelt
- HAMIS  Az adatok minden keretben garantáltan helyet kapnak a keret méretének 90%-áig
- IGAZ  Az adatok minden keretben garantáltan helyet kapnak a keret méretének 10%-áig
- HAMIS  Csak kimeneti irányultsága lehet

Question **5**

Correct

Mark 2.00 out of 2.00

Adat és utasítás egyidejűleg (párhuzamosan) olvasható ki

- HAMIS  A Neumann architektúrában
- IGAZ  A Harvard architektúrában
- IGAZ  A módosított Harvard architektúrában
- HAMIS  Miden vezérlésáramlásos architektúrában

Question **6**

Correct

Mark 2.00 out of 2.00

Szoftver által menedzselte TLB esetén az alábbi tevékenységek közül melyek az operációs rendszer feladatai?

- IGAZ  A TLB hiba feloldása a laptábla bejárásával
- IGAZ  Áldozatválasztás a fizikai memóriában tárolt lapok közül, ha lapcsere szükséges
- IGAZ  Az érintett lap betöltése a háttértárról, ha lapcsere szükséges
- HAMIS  A virtuális címből a fizikai cím előállítás

Question **7**

Correct

Mark 2.00 out of 2.00

Az alábbiak közül mely állítások igazak a szuperskalár architektúrákra?

- HAMIS  A fordító válogatja össze a független utasításokat párhuzamos végrehajtásra
- HAMIS  Cache használatát nem teszi lehetővé
- IGAZ  A processzor detektálja és kezeli az egymáshatásokat
- HAMIS  A fordító végzi a regiszter átnevezést

## Question 8

Correct

Mark 2.00 out of 2.00

X

Állapítsa meg, hogy a bal oldalon található elágazásbecslési feladatokhoz a jobb oldali adatszerkezetek közül melyek használhatók!

	Szemetelési hajlam tábla	Ugrási előzmény regiszter (BHR)	Ugrási hajlandóság tábla (PHT)
Ugró utasítás kimenetelének becslése	<input type="text"/>	<input checked="" type="checkbox"/> X ✓	<input checked="" type="checkbox"/> X ✓
Szubrutinból való visszatérés címének becslése	<input type="text"/>	<input type="text"/>	<input type="text"/>
Ugró utasítás ugrási címének becslése	<input type="text"/>	<input type="text"/>	<input type="text"/>

## Question 9

Correct

Mark 2.00 out of 2.00

X

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

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

## Question 10

Correct

Mark 2.00 out of 2.00

X

Mely állítások igazak a Daisy chain alapú és a párhuzamos arbitrációra?

	Daisy chain alapú arbitráció	Párhuzamos arbitráció
Tud úgy elromlani egy periféria (nem az arbiter!), hogy másokat elzárjon a busztól	<input checked="" type="checkbox"/> X ✓	<input type="text"/>
Igazságosabb kiszolgálást is lehetővé tesz	<input type="text"/>	<input checked="" type="checkbox"/> X ✓
Elvben tetszőlegesen bővíthető új perifériával	<input checked="" type="checkbox"/> X ✓	<input type="text"/>
Nem működik, ha egyidejűleg több eszköz is bejelenti igényét a buszra	<input type="text"/>	<input type="text"/>

Your answer is correct.

## Question 11

Correct

Mark 2.00 out of 2.00

X

Mikor hajtható végre egy művelet a tanult 3 információfeldolgozási modellben?

	Vezérlésáramlásos modell	Adatáramlásos modell	Igényvezérelt modell
Amikor szükség van az eredményére	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> ✓
Amikor a vezérlő token odaér	<input checked="" type="checkbox"/> ✓	<input type="checkbox"/>	<input type="checkbox"/>
Amikor minden operandusa rendelkezésre áll	<input type="checkbox"/>	<input checked="" type="checkbox"/> ✓	<input type="checkbox"/>

Your answer is correct.

## Question 12

Incorrect

Mark 0.00 out of 2.00

Amdahl törvénye szerint mennyivel gyorsabban fut egy program egy 3 processzorból álló multiprocesszoros rendszerben, mint egy 1 processzoros rendszerben, ha a programnak csak az 1/4-a párhuzamosítható tökéletesen, a többi része pedig csak szekvenciálisan futtatható?

A gyorsulás mértéke:  ✗ [1.2 or 1,2]

## Question 13

Partially correct

Mark 1.00 out of 2.00

Hány tranzisztor szükséges 18 bit tárolásához egy SLC-t, MLC-t, illetve TLC-t használó NAND flash alapú SSD meghajtón?

- SLC esetben:  ✓
- MLC esetben:  ✗ [9]
- TLC esetben:  ✗ [6]

Your answer is partially correct.

Question **14**

Complete

Mark 2.00 out of 2.00

Mi a sorrendvisszaállító buffer szerepe a soron kívüli végrehajtást támogató processzorokban?

A sorrendvisszaállító buffer megjegyzi az utasítások eredeti sorrendjét, hogy a párhuzamosan elvégezhető műveletek befejezése után az eredeti sorrendbe íródjanak be a műveletek a memóriába.

Comment:

Question **15**

Complete

Mark 2.00 out of 2.00

Miért jelent gondot, ha a cache menedzsment során az idő előtti betöltés (prefetch) túl óvatos, illetve ha túl agresszív?

*Mindkét választ az alábbi szövegmezőbe írja, külön sorban. Jelezze egyértelműen, hogy melyik válasz vonatkozik a túl óvatos, és melyik a túl agresszív prefetch-re!*

Ha túl agresszív a betöltés akkor tele lehet a cache olyan adatokkal, amik lehet, hogy soha vagy csak ritkán kerülnek használatra ezért sok lesz a cache szemét

Ha túl óvatos a betöltés akkor nem kerülnek be olyan adatok a cachebe amire szükségünk lenne ezért sok lesz a cache hiba

Comment:

Question **16**

Partially correct

Mark 2.00 out of 3.00

Legyen adott az alábbi utasítás sorozat:

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

Az utasítássorozatot egy 5 fokozatú utasítás pipeline-nal rendelkező ARM926EJ-S processzor hajtja végre. A processzor minden utasítás végrehajtását 5 részműveletre bontja: IF-ID-EX-MEX-WB. Az első két fázis a betöltés (**IF**) és a dekódolás (**ID**). Ezt követően az **EX** fázis elvégzi az aritmetikai és logikai műveletek első lépését, a **MEX** pedig a másodikat. Azaz az aritmetikai műveletek két ciklusidőt igényelnek (iterációs idő: 1), a MEX fázis végére áll elő az eredmény. Memóriaműveletek esetén az EX fázisban történik a címszámítás, a MEX fázisban pedig maga a memóriaművelet. Az utolsó, ötödik fázis az eredmény visszaírása a regiszter tárolóba (**WB**).

Minden utasítás végrehajtása mind az 5 fázison átesik, függetlenül attól, hogy szüksége van-e rá. Minden forwarding út használata megengedett. Ha bármilyen egymáshatás az utasítás megállítását igényli, az utasítás mindig a legutolsó olyan fázisban áll meg, ameddig egymáshatás 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 adategymáshatás
- **F\***: a szünet oka feldolgozási egymáshatás
- **P\***: a szünet oka procedurális egymáshatás

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, szóköz nélkül)! (3p)

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	Forwarding
i1	IF ✓	ID ✓	EX ✓	MEX ✓	WB ✓								
i2		IF ✓	ID ✓	EX ✓	MEX ✓	WB ✓							
i3			IF ✓	ID ✓	A* ✓	EX ✓	MEX ✓	WB ✓					MEX/WB ✓
i4				IF ✓	F* ✓	ID ✓	A* ✓	EX ✓	MEX ✓	WB ✓			MEX/WB ✓
i5						IF ✓	F* ✓	ID ✓	A* ✗	EX ✗	MEX ✗	WB ✗	MEX/WB ✓
									{EX}	{MEX}	{WB}	{blank}	

(Nem feltétlenül van szükség a táblázat minden oszlopára)

Your answer is partially correct.

Question 17

Correct

Mark 3.00 out of 3.00

X

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

**RAW függőség:**

	i1	i2	i3	i4	i5
i1					
i2					
i3	X	X			
i4		X	X		
i5				X	

**WAR függőség:**

	i1	i2	i3	i4	i5
i1					
i2					
i3	X				
i4					
i5					

**WAW függőség:**

	i1	i2	i3	i4	i5
i1					
i2					
i3	X				
i4					
i5					

Your answer is correct.

Question **18**

Partially correct

Mark 6.00 out of 8.00

Egy virtuális tárkezelésre képes processzor 16 bites virtuális és 14 bites fizikai címeket támogat. A lapméret 8192 bájt ( $=2^{12}$ ). A címfordításhoz kétszintű laptáblát használ, valamint egy 2 bejegyzéses, LRU algoritmussal menedzselte teljesen asszociatív TLB-t.

- (a) Hány lapból áll a virtuális memória?  ✓
- (b) Hány keretből áll a fizikai memória?  ✓
- (c) A futó program az alábbi lapokra hivatkozik:
  - 1, 2, 10

Adja meg a laptábla és a TLB tartalmának alakulását a fenti laphivatkozások hatására! 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 áldozat a 8-as lap. Lapcsere esetén, ha a memóriából kikerült laphoz tartozik TLB bejegyzés, akkor az operációs rendszer azt az egy bejegyzést invalidálja.

Az elsőszintű laptábla kezdeti állapota:

V	Cím
00:1	
01:0	?
10:1	
11:0	?

A másodsztintű laptáblák kezdeti állapota:

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

A TLB kezdeti állapota pedig a következő:

Valid	Lap	Keret	Kor
1	8	1	1
1	10	3	2

**Figyelem:** az alábbi négy kérdésnél mindig újra a fenti kiinduló állapotból induljon ki, ne az előző választ folytassa!

1. Ha egy laptábla bejegyzés 2 byte-os, összesen mennyi memóriát foglal a laptábla a fizikai memóriában a fenti állapotában?

✗ [24] byte

2. Adja meg a TLB és a másodsztintű laptáblák állapotát, ha a feladatkiírásban megadott kezdeti állapotban az **1-es lap** meghívkozásra kerül (elegendő csak a megváltozott bejegyzéseket beírni):

A másodsztintű laptáblák új állapota:

Valid	Lap	Keret	Kor	V	Cím	V	Cím	V	Cím	V	Cím
			2 ✓	00:		00:		00:		00:	
1 ✓	1 ✓	2 ✓	1 ✓	01:		01:		01:		01:	
				10:		10:		10:		10:	
				11:		11:		11:		11:	

3. Adja meg a TLB és a másodsztintű laptáblák állapotát, ha a feladatkiírásban megadott kezdeti állapotban a **2-es lap** meghívkozásra kerül (elegendő csak a megváltozott bejegyzéseket beírni):

A másodsztintű laptáblák új állapota:

Valid	Lap	Keret	Kor	V	Cím	V	Cím	V	Cím	V	Cím
1 ✓	2 ✓	1 ✓		00:		00:	0	00:		00:	
				01:		01:		01:		01:	
				10:	1 ✓	10:		10:		10:	
				11:		11:		11:		11:	

4. Adja meg a TLB és a másodsztintű laptáblák állapotát, ha a feladatkiírásban megadott kezdeti állapotban a **10-es lap** meghivatkozásra kerül (elegendő csak a megváltozott bejegyzéseket beírni):

A másodsztintű laptáblák új állapota:

A TLB új állapota:

Valid	Lap	Keret	Kor
			2 ✓
			1 ✓

	V	Cím		V	Cím		V	Cím		V	Cím
00:			00:			00:			00:		
01:			01:			01:			01:		
10:			10:			10:			10:		
11:			11:			11:			11:		

Your answer is partially correct.

Question **19**

Correct

Mark 5.00 out of 5.00

Egy 1 GHz órajel-frekvencián üzemelő számítógéphez egy aktív toll (beviteli periféria) kapcsolódik. A tollhoz kapcsolódó rajztábla nagy felbontású, a toll pályájának minél pontosabb lekötése érdekében 0.04 ms-onként mintavételezi az érintés pozícióját. Az aktuális pozíció lekérdezése 2000 órajel ideig tart.

a) Másodpercenként hányszor kell lekérdezni a perifériát, ha egyetlen pozíciófrissítésről sem szeretnénk lemaradni?

Csak a számot írja a beviteli mezőbe! (1 pont)

Válasz:  ✓

b) Mekkora relatív terhelést jelentene a processzor számára, ha a periféria kezelését polling segítségével oldanánk meg?

Az eredményt %-ban kérjük megadni! Csak a számot írja a beviteli mezőbe! (2 pont)

Válasz:  ✓ %

c) Mekkora relatív terhelést jelent a periféria kezelése a processzor számára, ha a periféria képes a pozíció változását interrupt segítségével is jelezni? A megszakítás feldolgozási idő 400 órajel (a pozíció lekérdezését nem tartalmazza). Tegyük fel, hogy az átlagos használat során 1800 pozícióváltás következik be másodpercenként.

Az eredményt %-ban kérjük megadni! Csak a számot írja a beviteli mezőbe! (2 pont)

Válasz:  ✓ %

Your answer is correct.

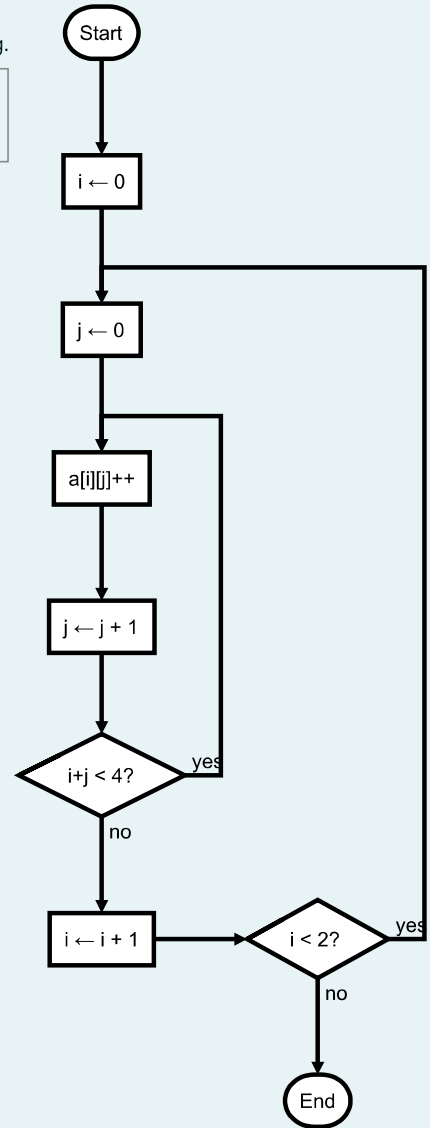
Question **20**

Correct

Mark 3.00 out of 3.00

Egy programban egy algoritmus a jobb oldalt látható folyamatábra szerint működik.  
Az algoritmus közelítőleg (bár nem pontosan) az alábbi C programrészletnek felel meg.

```
for (int i=0; i<2; i++)
    for (int j=0; i+j<4; j++)
        a[i][j]++;
```



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

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

i<2   nem\_ugrik\_(no)   ugrik\_(yes)   i+j<4

Your answer is correct.

Question **21**

Correct

Mark 4.00 out of 4.00

Tételezzük fel (az előbbi algoritmus tényleges viselkedésétől függetlenül), hogy a program a futása során az alábbi utasításszámláló 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-t) 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 (BHR) tartalmát binárisan a feltételes ugró utasítások végrehajtása után, ha a kezdeti tartalma (101) volt! (Az LSB a jobb szélén!!!)
- 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 becslő melyik utasítás esetén javasol ugrást (1) és melyeknél nem (0)?
- A "sikeresség" 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)
<b>GBHR</b>	101	010 ✓	100 ✓	001 ✓	010 ✓	100 ✓	001 ✓
<b>PHT[0]</b>	00						
<b>PHT[1]</b>	01				00 ✓		
<b>PHT[2]</b>	10		01 ✓			00 ✓	
<b>PHT[3]</b>	01						
<b>PHT[4]</b>	10			11 ✓			11 ✓
<b>PHT[5]</b>	10	01 ✓					
<b>PHT[6]</b>	10						
<b>PHT[7]</b>	11						
<b>Becslés:</b>	-	1 ✓	1 ✓	1 ✓	0 ✓	0 ✓	1 ✓
<b>Sikeresség:</b>	-	H ✓	H ✓	J ✓	J ✓	J ✓	J ✓

A táblázatba elgondó csak a megváltozott bejegyzéseket beírni!

Your answer is correct.

Question **22**

Incorrect

Mark 0.00 out of 4.00

Vegyünk egy 4 kB méretű cache-t 64 byte-os blokkmérettel. A cache kezdetben csupa érvénytelen blokkot tartalmaz. Lefuttatjuk az alábbi programot:

```
double a[1024];
double b[1024];
for (int i=0; i<1024; i++)
    a[i] = 0.0;
for (int i=0; i<1024; i++)
    a[i] += b[i] * 2.0 * M_PI * 1024 / i;
```

Feltételezések: a double 8 byte-os, az **a** és **b** tömb blokkhatáron kezdődik, a memóriában sorfolytonosan helyezkednek el, a cache *teljesen asszociatív* leképzést használ *LRU* menedzsmenttel, az írási műveletek kezelése *write-allocate*.

a) Összesen hány cache hibát vált ki a fenti algoritmus az **a** és **b** tömb elemeinek elérésekor? (2 pont)  ✘ @384@

b) Optimalizálja a fenti algoritmust ciklus egyesítéssel! Összesen hány cache hibát vált ki az optimalizált algoritmus az **a** és **b** tömb elemeinek elérésekor? (2 pont)  ✘ @256@

Your answer is incorrect.