

2012. őszi ZH-k és Vizsgák

1. ZH

1.1. ZH1

1.1.1. A csoport

1. Mikor hajtható végre egy művelet a tanult 3 információfeldolgozási modellben?

- Vezérlésáramlásos: amikor a vezérlő token odaér
- Adatáramlásos: amikor minden operandus rendelkezésre áll
- Igényvezérelt: amikor szükség van az eredményre.

2. Hogyan kell jelezni a program leírásában a párhuzamosítható részeket

- a vezérlésáramlásos modellben: Fork/Join primitívekkel
- az adatáramlásos modellben: ahogy
- az igényvezérelt modellben: ahogy

3. A precedenciagráf irányított élei megadják,

- hogy egy operandus kiszámolásához mely műveletet kell elvégezni
- hogy mi a soron következő végrehajtható utasítás
- **hogy egy művelet eredménye mely más műveletek operandusa.**

4. Melyek a Neumann architektúra főbb komponensei?

- **Processzor**
- Cache
- **Perifériák**
- **Memória**
- Rendszerbusz
- Háttértár

5. Honnan tudja egy Neumann elvű számítógép processzora, hogy egy memóriaeléréskor a meghivatkozott memóriabeli objektumot milyen típusúnak tekintse?

- Onnan, hogy a memóriában az objektumok típusa is le van tárolva.
- Az objektumok típusa nincs ugyan letárolva, de tudja, milyen típusú hiszen ő tette bele.
- **Onnan, hogy az utasítás, amivel a program a memóriához nyúl, milyen típusú adatra vonatkozik.**

6. Fix hosszúságú utasításkódolás: a processzor gyorsabban tudja dekódolni.

Változó hosszúságú utasításkódolás: helytakarékosabb, rövidebb programhoz vezet.

7. A hexadecimális 0x0A0B0C0D számot adj meg

- little endian bájtsorrenddel tárolva: 0D0C0B0A
- big endian bájtsorrenddel tárolva: 0A0B0C0D

8. Melyik forgalomszabályozás helyett milyen hibák fordulhatnak elő az adatátvitel során?

- **Nincs forgalomszabályozás:** adathiány, adat egymásra-futás.
- **Egyoldali feltételes forgalomszabályozás:** adathiány, adat egymásra-futás.
- **Kétoldali feltételes forgalomszabályozás:** semmilyen hiba nem fordulhat elő.

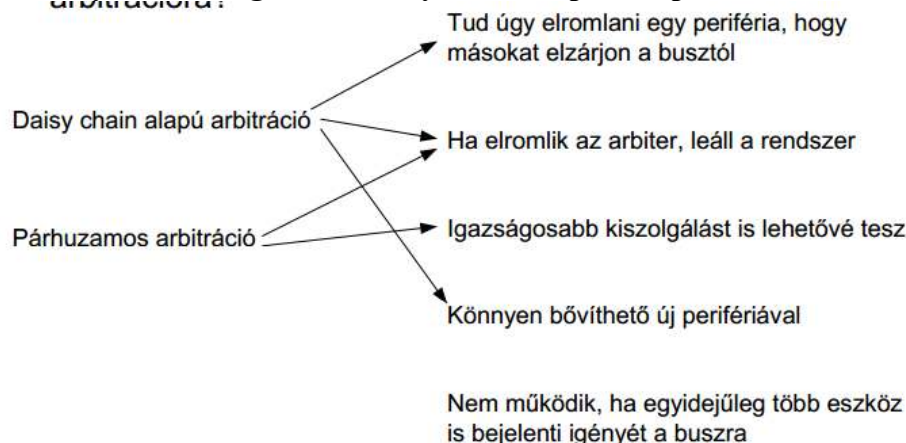
9. Tegyük fel, hogy a CPU támogatja az interruptok használatát és az engedélyezve is van. Mely állítások igazak?

- A lassú perifériákat célszerű pollinggal kezelni, mert úgy kisebb a processzor terhelése.
- A lassú perifériákat célszerű pollinggal kezelni, mert úgy gyorsabban értesülhetünk a periféria megváltozott állapotáról.
- A gyors perifériát célszerű interrupttal kezelni, mert úgy kisebb a processzor terhelése.

10. Van egy bemeneti perifériánk, és egy processzorunk. Milyen forgalomszabályozást célszerű használni az adatátviteli hibák elkerülésére, ha

- a periféria sebessége befolyásolható: kétoldali feltételes
- a periféria sebessége nem befolyásolható: egyoldali feltételes.

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



12. Milyen eszköz található az Arduino analóg bemenetein?

- Analóg-digitális átalakító
- Digitális-analóg átalakító
- PWM generátor

13. Milyen forgalomszabályozást használ a PCI?

- Semmilyen
- Az egyik fél jelezni tudja, ha kész az átvitelre
- Mindkét fél jelezni tudja, ha kész az átvitelre.

14. Hogyan tudja használni egy PCI eszköz a rendszerszintű DMA vezérlőt?

- A csatolófelületen erre szolgáló lábak segítségével
- Speciális tranzakciók segítségével
- **Sehogy**

15. Mely adatátviteli lehetőségeket támogatja a PCI és a PCIe? (Kezdeményező -> megszólított)

- CPU -> PCI periféria
- PCI periféria -> CPU
- **PCI periféria -> memória**
- memória -> PCI periféria
- **PCI periféria -> PCI periféria**

16. Mely 3 koordinátával lehet azonosítani egy pontot egy több adathordozó réteggel ellátott forgólemezes adattárolón?

- Radiális távolság
- Egy fix referenciaponthoz képesti szög

- Az adathordozó rétegek száma.

17. Ha egy merevlemez minden sávjában ugyanannyi egyofrma méretű szektor található, akkor

- a külső sáv szektorait gyorsabban kell leolvasni
- a belső sáv szektorait gyorsabban kell leolvasni
- **minden szektort ugyanannyi ideig tart leolvasni.**

18. Rajzolja fel egy merevlemez szektorának felépítését!



19. A mágneses adathordozó egy blokkjának állapota a következő: →←←←→→←→←←

A váltáshoz 1-es bitet, a váltás hiányához 0-as bitet rendelünk.

- Milyen bitsorozat tartozik a megadott mágneses mintázathoz? 10010111
- Változtassa meg a harmadik bitet (1-től számolva)! Milyen mintázatot kapunk?

→←←→←←→←→

1. Nagypélda:

Processzorunk órajel frekvenciája 100MHz. A számítógéphez egy billentyűzetet kötünk, melyen átlagosan 10 karaktert ütnek le másodpercenként, de két leütés akár 50 ms-onként is követheti egymást. A billentyűzet állapotának lekérdezése (mely tartalmazza a lenyomott gomb kódját is) 500 órajelet igényel. Az interrupt feldolgozási idő ezen felül meg 100 órajel.

- Hányszor kell másodpercenként lekérdezni a billentyűzetet, hogy ne maradjunk le semmiről?
- Mekkora terhelést jelent a processzor számára, ha a billentyűzet kezelésére polling-ot használ?
- Mekkora terhelést jelent a processzor számára, ha a billentyűzet kezelésére interrupt-ot használ?

2. Nagypélda:

Egy merevlemez 2 db kétoldalas lemezt tartalmaz, melyek mindegyiken 30000 sáv található, minden sávban 1000 szektorral. A szektorok mérete 500 bajt. ZBR nincs, a lemezek állandó 10000 fordulat/perc-el forognak.

- Adjuk meg CHS koordinata rendszerben a lemez kapacitását!
- A lemez kapacitása bajtokban merve?
- Mennyi ideig tart, míg a lemez megtesz egy fordulatot?
- Meddig tart egy szektor leolvasása az adathordozóról? (Ha mar ott van a fej)
- Ha a parancsfeldolgozási idő 0.1 ms, az adatátviteli interfész sebessége pedig $50 \cdot 10^6$ bajt/s, akkor átlagosan mennyi ideig tart egy 2000 bájtos keres teljes kiszolgálása?

1.1.2. B csoport

1. A mágneses adathordozó egy blokkjának állapota a következő: ←←←→→→←→←←

A váltáshoz 1-es bitet, a váltás hiányához 0-as bitet rendelünk.

- Milyen bitsorozat tartozik a megadott mágneses mintázathoz? 00100111

- • **Valtoztassa meg a harmadik bitet (1-től számolva)! Milyen mintazatot kapunk?**

←←←←←→←→

2. Sorolja fel, milyen összetevőkből áll egy szektor olvasási kérés teljes kiszolgálási ideje! Melyeket lehet ezek közül átlapolni?

- Parancsfeldolgozási késleltetés
- Seek idő
- Forgási idő
- **Adatleolvasási idő**
- **Adatok küldése az interfészen**

3. Ha egy merevlemez minden sávjában ugyanannyi, egyforma méretű szektor található, akkor

- a külső sávokban nagyobb az adatsűrűség
- **a belső sávokban nagyobb az adatsűrűség**
- minden sávban egyforma az adatsűrűség

4. Mely 3 koordinátával azonosít egy szektort fizikailag a merevlemez?

- Cilinder
- Fej
- Szektorszám

5. Mi a PCI és a PCIE viszonya kompatibilitás szempontjából?

- **Szoftver szinten kompatibilisek**
- Hardver szinten
- Egyáltalán nem
- Minden szempontból

6. A PCI ablakok milyen címtartományra vonatkoznak?

- Csak memória címtartományára
- Csak a periféria címtartományára
- Mindkettőre vonatkozhatnak, az op. rendszer határozza meg a rendszer indításakor
- **Mindkettőre vonatkozhatnak, a periféria határozza meg, hogy melyik ablak melyikre vonatkozzon.**

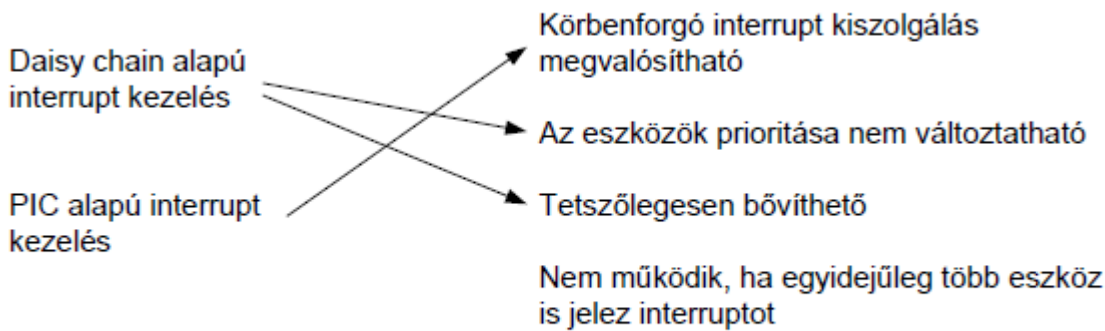
7. Milyen arbitrációt használ a PCI?

- Semmiylet, mert csak egy master lehet a buszon
- Daisy chain alapú arbitrációt
- **Párhuzamos arbitrációt**

8. Mit jelent a PWM kimenet az Arduino-n?

- **Impulzusélesség-moduláció**
- Impluzusfrekvencia-moduláció
- Impluzusszám-moduláció

9. Interrupt



10. Miért jelent gondot, ha a polling intervallum

- **túl kicsi:** nagy terhelés a processzornak
- **túl nagy:** lemaradunk egy eseményről (adatvesztés).

11. Jelölje meg az igaz állításokat!

- A DMA vezérlőt a periféria programozza fel
- **A DMA vezérlő alkalmazása csökkenti a CPU interrupt terhelését**
- **Több DMA vezérlő is lehet egy buszon**
- A DMA vezérlő minden adategység sikeres átvitelét interrupt-tal jelzi a processzornak.

12. Forgalom szabályozás hiányában mikor és milyen hibák fordulhatnak elő?

- **Adat egymásra futás:** küldő gyorsabb, mint a fogadó
- **Adathiány:** fogadó gyorsabb, mint a küldő.

13. Tegye ki a relációs jeleket a RISC és CISC viszonyával kapcsolatban!

Utasítások száma:

- $RISC < CISC$

Regiszterek száma:

- $RISC > CISC$

Címzési módok száma:

- $RISC < CISC$

Program mérete:

- $RISC > CISC$

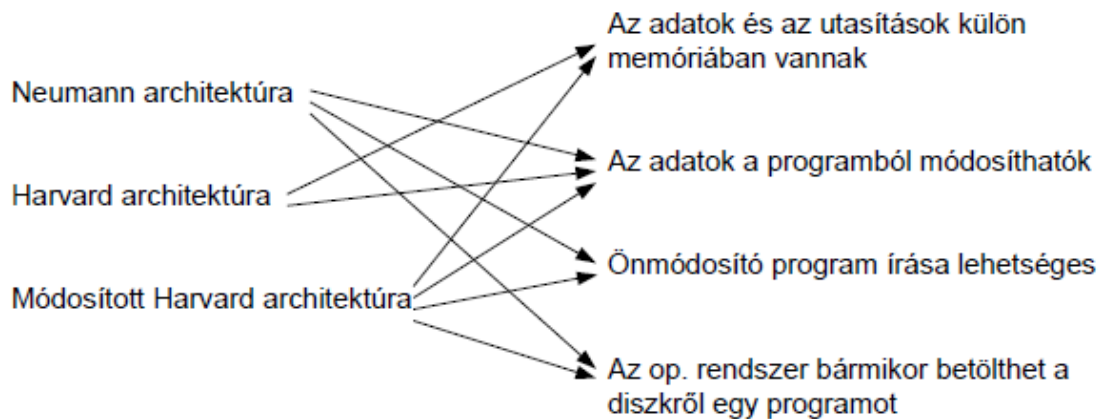
14. Egy utasításkészletet ortogonálisnak nevezünk, ha

- egy adott célra csak egy utasítást tartalmaz
- egy adott célra sokféle, kényelmesen használható utasítás is rendelkezésre áll
- **minden címzést használó utasítás minden címzési módot használhat**
- minden címzést használó utasítás csak egyetlen címzési módot használhat

15. Adat és utasítás egyidejűleg olvasható ki

- a neumann architektúrában
- **a harvard architektúrában**
- **a módosított harvard architektúrában**

16. Mely tulajdonságok mely architektúra(k)ra jellemző?



17. Az igényvezérelt architektúrában az igénygráf irányított élei megadják, hogy

- egy operandus kiszámolásához mely műveleteket kell elvégezni
- mi a soron következő végrehajtható utasítás
- egy művelet eredménye mely más művelet operandusa.

18. Milyen információfeldolgozási modell szerint működnek az alábbiak?



19. Milyen formális eszközzel írunk le egy programot

- a vezérlésáramlásos modellben: folyamatábra
- az adatáramlásos modellben: precedenciagráf / függőségi gráf

1. Nagypélda:

Processzorunk órajele frekvenciája 1GHz. A számítógéphez egy 100 Mbit/s sebességű hálózati interfészt kötünk, melyen 1500 bájtos (=12000 bit) csomagok közlekednek. A periféria állapotának lekérdezése (mely nem tartalmazza a hálózati csomag tartalmát) 600 órajelet igényel. Az interrupt feldolgozási idő 400 órajele. Ha jött csomag, annak átvitele 5000 órajelet vesz igénybe. Jelenleg epp egy 24 Mbit/s sebességű letöltés van folyamatban a hálózaton.

- a) Hány ms-onként kell lekérdezni a perifériát, hogy biztosan ne maradjunk le egy csomag érkezéséről?
- b) Mekkora terhelést jelent a periféria kezelése a processzor számára, ha a csomagok érkezését polling-al figyeljük?
- c) Mekkora terhelést jelent a processzor számára, ha interrupt-ot használunk a csomag érkezések jelzésére?

2. Nagypélda:

Egy merevlemez 3 db kétoldalas lemezt tartalmaz, melyek mindegyiken 20000 sav található, minden savban 1000 szektorral. A szektorok mérete 500 bájt. ZBR nincs. Az átlagos seek idő 4 ms. Mérésekkel megállapítottuk, hogy az egy szektorra vonatkozó olvasási keresek kiszolgálási ideje 10 ms.

- a) Ha a parancsfeldolgozási késleltetéstől es az interfészen való átviteli időtől eltekintünk, milyen gyorsan forog a lemez? (fordulat/perc-ben megadva)
- b) Adjuk meg CHS koordinata rendszerben a lemez kapacitását!
- c) A lemez kapacitása bajtokban merve?
- d) Meddig tart egy szektor leolvasása az adathordozóról? (Ha mar ott van a fej)
- e) Ha a parancsfeldolgozási idő 0.1 ms, az adatátviteli interfesz sebessége pedig 50*106 bajt/s, akkor átlagosan mennyi ideig tart egy 2000 bájtos keres teljes kiszolgálása?

1.2.ZH2

1.2.1. A csoport

1.Miért célszerű az SSD-n az adatokat tömörítve tárolni?

- Mert így a felhasználó több adatot tud tárolni rajta.
- **Mert így kevesebbet kell írni, ami előnyös a kopás szempontjából.**
- Mert így egyszerűbb a kopásból eredő hibák detektálására.

2. A NAND flash memóriában mik az alábbi műveletek adategységei?

	a) bájt
1 Olvasás	b) 1 lebegő gate-es tranzisztor
2 Írás	c) Lap
3 Törlés	d) Blokk

1 – c

2 – c

3 – d

3. Mely szereplők kezdeményezhetnek kommunikációt az USB 2.0-ban?

- Csak a hub-ok
- **Csak a root hub**
- Csak a perifériák
- Minden hub és minden periféria

4.Tegy növekvő sorrendbe az USB sebességeket!

LS < FS < HS < SS

5.Mit tesz egy USB 2.0-ás hub, ha egy FS tranzakciót észlel a buszon, és csupa LS eszköz csatlakozik rá?

- Továbbítja az LS eszközöknek, azok észreveszik, hogy túl gyors nekik, és figyelmen kívül hagyják.
- **Nem továbbítja az LS eszközöknek.**
- SPLIT tranzakció segítségével lassabb sebességre konvertálja az FS tranzakciót, hogy az LS eszközök is megértsék.

6.Mely esetekben kell újraírni/újrafordítani egy programot indexelt leképezéssel megoldott tárbővítés esetén?

- **Ha a hardver a meglévők helyett fele annyi, egyenként kétszer akkora ablakok használatára tér át.**

- Ha a hardver a meglévők helyett kétszer annyi, egyenként fele akkora ablakok használatára tér át.
- Egyik esetben sem.

7. Hányszor kell a memóriához fordulni a címfordítás (és csak annak) érdekében 5 szintű laptábla használata esetén?

- TLB hiba esetén: 5
- TLB találat esetén: 0

8. A megfelelő összekötésekkel jelezze a felsorolt laptábla adatszerkezetek tulajdonságait!

	a) Mérete a fizikai memória méretével arányos
1 Egyszintű laptábla	b) Mérete a virtuális memória méretével arányos
2 Többszintű laptábla	c) Működéséhez a teljes adatszerkezetnek a fizikai memóriában a helye
3 Virtualizált laptábla	d) TLB hiba esetén, optimális esetben csak 1-szer kell a memóriához fordulni
4 Inverz laptábla (ahogy tanultuk)	e) Egyes részei a merevlemezen is tárolhatók

1 – b, c, d

2 – b, e

3 – b, d, e

4 – a, c

9. Mit tárol a TLB?

- A leggyakrabban használt lapokat
- A leggyakrabban használt kereteket
- **A leggyakrabban használt laptábla bejegyzéseket**
- A leggyakrabban használt laptábla bejegyzések sorszámait

10. Hány tranzisztor kell 2 bit tárolásához

- SRAM esetén: 12
- DRAM esetén: 2
- SLC flash memória esetén: 2
- 2 bites MLC flash memória esetén: 1

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

	a) Gyorsabban írható és olvasható
1 DRAM	b) A CPU-val egy lapkára integrálható
2 SRAM	c) Azonos kapacitás mellett több energiát fogyaszt
	d) A cache memória alapja

1 -

2 – a, b, c, d

12. Mely állítások igazak az n-utas asszociatív cache szervezésre?

- **n db azonos indexű blokkot tud tárolni**
- n féle különböző indexet különböztet meg
- **kereséskor n komparátor működik egyszerre**
- a cache tag-eket a rendszermemóriában tárolja

13. Write through írási politika mellett mikor vezeti át a processzor a megváltozott cache tartalmat a rendszermemóriába?

- **Amint a cache-ben megváltozik, rögtön átvezeti**
- Amikor a blokk kikerül a cache-ből
- Egyik sem, a write through írási politika íráshoz nem használja a cache-t

14. Soroljon fel 3 cache blokk csere stratégiát!

- véletlenszerű
- körbenforgó
- legrégebben használt (LRU)
- nem a legutóbb használt
- legritkábban használt

15. Milyen összetevőkből áll egy keresési művelet a direkt leképezésen alapuló cache szervezés esetén?

- Csak indexelésből
- Csak a tag(ek) összehasonlításából
- **Indexelésből, majd a tag összehasonlításból**

16. Pipeline egymásrahatások

	a) Az egyik művelet operandusa egy korábbi művelet eredménye
1 Feldolgozási egymásrahatás	b) Ugró utasítás esetén az ugrási feltétel kiértékeléséig nem tudjuk, hogy honnan kell betölteni a további utasításokat
2 Adategymásrahatás	c) A pipeline több fázisa verseng ugyanazon erőforrásért
3 Procedurális egymásrahatás	d) Utasításvégrehajtás közben egy kivétel keletkezett, ami azonnali kezelést igényel

1 – c

2 – a

3 – b

17. Mit jelent a pipeline átviteli sebessége?

Végrehajtott utasítások száma időegységenként.

18. Mely kivételek fordulhatnak elő az IF fázisban?

- **Laphiba**
- Érvénytelen utasítás
- Aritmetikai hiba
- Egyik sem

19. Az alábbi 3 utasításból álló kódrészletben az i3 feldolgozása közben mely pipeline regiszterből veszi az R8 értékét az ALU? (A tanult 5 fokozatú pipeline-t alapul véve)

$R7 \leftarrow R1 + R5$

$R8 \leftarrow R7 + R2$

$R5 \leftarrow R8 + R7$

A pipeline regiszter: „EX/MEM” regiszter

1. Számpélda Vegyünk egy 12 bites címtartománnyal rendelkező processzort, mely 4kB memóriát képes címezni. Nekünk 8kB memóriára van szükségünk, ezért tömbkapcsolást használunk, 2kB-os tömbmérettel. A memória-tartomány alsó 2kB-os fele mindig elérhető (ez a 0. tömb, ebben van a program, valamint az adatok egy része), a felső 6kB pedig kapcsolt. A program a 8kB-os fizikai memóriában, sorrendben az alábbi tartalmakat hivatkozza:

3002, 7101, 512, 7310, 2805

A kérdések a következők: (6 pont)

- a) Minden felsorolt memóriacímre adja meg, hogy melyik tömbre vonatkozik, illetve, hogy a programozónak a szükséges tömb bekapcsolása után a programjában milyen memóriacímre kell hivatkoznia, hogy a kívánt tartalmat elérje!

	3002	7101	512	7310	2805
Hányas tömb:	1	3	0	3	1
A programból kiadott memóriacím:	3002	3005	512	3214	2805

3 pont

- b) Hány tömbkapcsolásra volt szükség, ha kezdetben az állandó 0. mellett az 1. tömb volt bekapcsolva? (954) (957) (1166) (757)

2

2 pont

- c) A fenti memóriahivatkozások hány tömböt használtak ténylegesen?

3

1 pont

(Segítség: $2 \cdot 2048 = 4096$, $3 \cdot 2048 = 6144$, $4 \cdot 2048 = 8192$.)

2. Számpélda Legyenek a virtuális címek 18 bitesek, a fizikai címek 14 bitesek, a lapméret legyen $2^{12} = 4\text{kB}$ méretű. Használjunk 3 szintű hierarchikus laptáblát, melyben a laptáblák mérete a hierarchia minden szintjén egyforma. A laptábla bejegyzések 16 bitesek. A feladatok a következők: (9 pont)

- a) Adja meg a virtuális címek tagozódását! Hány bites

az elsőszintű laptábla index:

2

a másodsztintű laptábla index:

2

a harmadsztintű laptábla index:

2

az eltolás mező:

12

1 pont

- b) Hány lapból áll a virtuális memória?

64

Hány keretből áll a fizikai memória?

4

2 pont

- c) Mekkora az összes laptábla összegzett mérete?

168

2 pont

- d) Optimális körülmények között mennyi a laptáblák által elfoglalt memória minimális mérete?

24

2 pont

- e) Mekkora lenne a laptábla mérete, ha 3 szintű helyett egyszintű laptáblát használnánk?

128

2 pont

1.2.2. B csoport

1. Jelölje be, mely utasítás típusok használják az EX fázisban az ALU-t a tanult 5 fokozatú pipeline-ban!

- Aritmetikai utasítások

- „Store” utasítások
- „Load” utasítások

2. Hogyan oldottuk fel a memóriát érintő feldolgozási egymáshatásokat a tanult 5 fokozatú pipeline-ban?

Külön utasítás és adat cache bevezetésével.

3. Elágazásbecslő eljárások

a) Mindig az ugrás megghiúsulására voksol	
b) Mindig az ugrás bekövetkezésére voksol	1 Dinamikus
c) Vissza ugrás esetén bekövetkezésre, egyébként megghiúsulásra voksol	2 Statikus
d) Az ugró utasítás múltbéli viselkedését tárolja, és abból következtet	

1 – d

2 – a, b, c

4. Az alábbi 2 utasításból álló kódrészletben az i2 feldolgozása közben mely pipeline regiszterből veszi az R6 értékét az ALU? (A tanult 5 fokozatú pipeline-t alapul véve)

$R6 \leftarrow \text{MEM}[R2]$

$R7 \leftarrow R6 + R4$

A pipeline regiszter: „MEM/WB” regiszter

5. Write back írási politika mellett mikor vezeti át a processzor a megváltozott cache tartalmat a rendszermemóriába?

- Amint a cache-ben megváltozik, rögtön átvezeti
- **Amikor a blokk kikerül a cache-ből**
- Egyik sem, a write back írási politika íráshoz nem használja a cache-t

6. Miért jelent gondot, ha a cache menedzsment során az idő előtti betöltés (prefetch):

- **túl óvatos:** a program nem éri el a potenciális sebességet (az első hivatkozáskor még nem lesz a cache-ben adat)
- **túl agresszív:** haszontalan adatokkal szemeteli tele a cache-t, kiszorítva hasznos blokkokat

7. Mik a virtuálisan indexelt, virtuális tag-eket használó cache előnyei a fizikailag indexelt, fizikai tag-eket használó cache-hez képest?

- Keskenyebb komparátorokra van szükség
- **A címfordítás bizonyos esetekben elhagyható**
- Nagobb méretű fizikai memória használatát támogatja

8. Mely állítások igazak a direkt leképezésen alapuló cache szervezésre?

- A cache mindig a legutóbb használt blokkot tartalmazza
- Adott méret mellett az összes tanult szervezés közül ebben a legszélesebb a cache tag
- **Adott méret mellett az összes tanult szervezés közül ebben a legszélesebb az „index” mező**
- **Nem helyezhető el benne 1-nél több azonos indexű blokk**

9. Egy „DDR2-800”-nak jelölt SDRAM-nak mi a

- **belső órajel frekvenciája: 200**
- **külső órajel frekvenciája: 400**

10. Adattárolásra szolgáló technológiák

	a) Tartalma tápfeszültség nélkül is megmarad
1 SLC flash memória	b) Állandó frisstítést igényel
2 DRAM	c) Az információt egy kondenzátor tárolja
3 SRAM	d) 1 bit tárolásához 1 tranzisztor szükséges
	e) A cache memória alapja

1 – a, d

2 – b, c, d

3 – e

11. Jelölje be a TLB-re vonatkozó igaz állításokat!

- A TLB mérete a fizikai memória méretével arányos
- A TLB mérete a virtuális memória méretével arányos
- A TLB-t a fizikai memóriában tároljuk
- A TLB-t a processzorban/MMU-ban tároljuk

12. Mely laptábla bejegyzések szükségesek feltétlenül az alábbi laptábla implementációk működéséhez? Csak azokat a mezőket jelölje be, melyek nélkül nem képzelhető el a laptábla működése!

	a) Valid bit
1 Többszintű laptábla	b) Keretszám
2 Virtualizált laptábla	c) Lapsorszám
3 Inverz laptábla (ahogy tanultuk)	d) Dirty bit
	e) Egy mutató a következő azonos hash értékű bejegyzésre

1 – a, b

2 – a, b

3 – a, c, e

13. Hányszor kell a memóriához fordulni a címfordítás (és csak annak) érdekében inverz laptábla használata esetén?

- TLB hiba esetén: legalább 2 (vagy több)
- TLB találat esetén: 0

14. Mely szereplő kezdeményezi az ablakok mozgatását idexelt leképezéssel megvalósított tárbővítés esetén?

- A cím leképző periféria, amikor észleli, hogy a program által kiadott cím nincs ablakkal lefedve
- A processzor, amikor észleli, hogy a program által kiadott cím nincs ablakkal lefedve
- A futó program, amikor a programozó egy addig ablakkal le nem fedett memóriaterületet kíván elérni

15. Mi történik az USB 2.0-ban, ha egy újonnan érkező periféria adatátviteli/tápellátási igényei nem elégíthetők ki?

- Nem léphet az USB rendszerbe
- Az USB driver közli vele, hogy kevesebbet kap, érje be ennyivel
- Az USB driver a többi periféria adatátviteli igényéből/tápellátásából lecsippentve igyekszik kiszolgálni az újonnan érkező kívánságát
- Egy kevésbé fontos eszközt kikapcsol, és az újnak adja az erőforrásait

16. Soroljon fel 2-2 konkrét USB perifériát, melyek az alábbi adatátviteli módokat használják!

Izokron: mikrofon, hangszóró, webcamera

Bulk: pendrive, külső merevlemez, printer, szkennel, memóriakártya olvasó

17. Milyen arbitrációt használ az USB 1.1?

- Soros arbitrációt (daisy chain)
- Párhuzamos, rejtett arbitrációt
- **Semmilyen arbitrációt nem használ**

18. Tárolók

	a) gyorsabban öregszik
1 MLC	b) azonos tranzisztorszám mellett több adatot tárol
2 SLC	c) egy tranzistor többféle tltöttségi szintjét különbözteti meg
	d) az olvasás adategysége a lap, a törlése a blokk

1 – a, b, c, d

2 - d

19. Mely állítások igazak a NAND flash alapú SSD tárolókra?

Egy lapra csak akkor lehet írni, ha...

- ... közvetlenül előtte letöröljük azt az egy lapot, amire írni szeretnénk
- ... ha az a lap vagy üres, vagy már nem használt adatokat tartalmaz
- **... ha a blokk törlése óta még senki nem írt a lapra**
- Egyik állítás sem igaz

1. Számpélda Vegyünk egy 12 bites címtartománnyal rendelkező processzort, mely 4kB memóriát képes címezni. Nekünk 8kB memóriára van szükségünk, ezért tömbkapcsolást használunk, 2kB-os tömbmérettel. A címtartomány alsó 2kB-os fele mindig elérhető (ez a 0. tömb, ebben van a program, valamint az adatok egy része), a felső 2kB pedig kapcsolt. A program a 8kB-os fizikai memóriában, sorrendben az alábbi tartalmakat hivatkozza:

902, 3008, 7104, 417, 6254

A kérdések a következők: (6 pont)

a) Minden felsorolt memóriacímre adja meg, hogy melyik tömbre vonatkozik, illetve, hogy a programozónak a szükséges tömb bekapcsolása után a programjában milyen memóriacímre kell hivatkoznia, hogy a kívánt tartalmat elérje!

	902	3008	7104	417	6254
Hányas tömb:	0	1	3	0	3
A programból kiadott memóriacím:	902	3008	3008	417	2158

3 pont

b) Hány tömbkapcsolásra volt szükség, ha kezdetben az állandó 0. mellett az 1. tömb volt bekapcsolva?

1

2 pont

c) A fenti memóriahivatkozások hány tömböt használtak ténylegesen?

3

1 pont

(Segítség: $2 \cdot 2048 = 4096$, $3 \cdot 2048 = 6144$, $4 \cdot 2048 = 8192$)

2. Számpélda Legyenek a virtuális címek 16 bitesek, a fizikai címek 14 bitesek, a lapméret legyen $2^{12}=4\text{KB}$ méretű. Használjunk 2 szintű hierarchikus laptáblát, melyben a laptáblák mérete a hierarchia minden szintjén egyforma. A laptábla bejegyzések 16 bitesek. A feladatok a következők: (9 pont)

a) Adja meg a virtuális címek tagozódását! Hány bites

az elsőszintű
laptábla index:

2

a másodszintű
laptábla index:

2

az eltolás
mező:

12

1 pont

b) Hány lapból áll a virtuális memória?

16

Hány keretből áll a fizikai memória?

4

2 pont

c) Rajzolja fel az összes érintett laptáblát, melyek az 1000000000000000 és az 1011111111111111 közé eső virtuális címek címfordításához szükségesek! A laptábla bejegyzésekben tüntesse fel a virtuális tárkezelés működéséhez elengedhetetlen mezőket!

Helyezze el a laptáblában az alábbi összerendeléseket:

Az 1001 lap a fizikai memóriában a 2-es keretben található

Az 1010 lap a háttértáron van

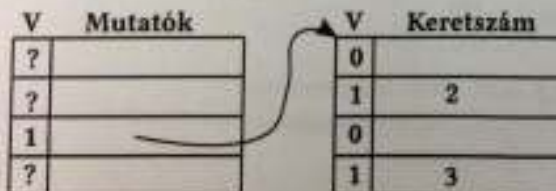
Az 1011 lap a fizikai memóriában a 3-mas keretben található

Az 1000 lap a háttértáron van

Látszik, hogy 2 laptábla van,
mindegyikben 4 bejegyzés: 1 pont

Van benne külön valid bit, és az
elsőszintű 3. bejegyzése mutat a
másodszintű elejére: 1 pont

Jók a beírt számok: 2 pont



d) Feltéve, hogy nincs laphiba, hány memóriaművelet szükséges a címfordításhoz

TLB találat esetén:

2

TLB hiba esetén:

0

2 pont

2. Vizsga

2.1. Vizsga1

1. Mik a szimultán többszálú feldolgozást támogató processzorok jellemzői? (12. előadás 14. dia)

- Az időosztásos alternatívája
- Csak szuperskalár architektúrával működik
- Szuperskalár esetben a sok műveleti egység gyakran kihasználatlan, mert nincs elég párhuzamosítható utasítás a programban
- A nem használt műveleti egységeken hajtjuk végre egy másik szál utasításait!

2. Hány élet kell elhagyni egy csomópont izolálásához egy 16 csomópontból álló (12. előadás 43-44 dia)

- Két D tórusz hálózatban: 4 (átmérő: $2\sqrt{N}$, ahol N =csomópontok száma)
- Hiperkocka topológiájú hálózatban: $\log_2 16=4$ (átmérő: $k = \log_2 N$ (dimenzió), N =csomópontok száma)

3. Amdahl törvénye. Mennyivel gyorsabban fut egy program egy 2 processzorból álló multiprocesszoros rendszerben, mint 1 processzoros rendszerben, ha a prog. 1/4-e csak szekvenciálisan futtatható? (12. előadás 23-25. dia)

- Legyen a programunk
 - P része tetszőlegesen párhuzamosítható
 - 1-P része szekvenciálisan végrehajtandó
- Legyen a futási idő 1 processzoros rendszerben: 1
- Kérdés: mennyi a futási idő N processzor esetén?
 - Ha az egész szekvenciális lenne: 1
 - Ha az egész párhuzamosítható lenne: 1/N
 - Ha P része párhuzamosítható: $(1-P)*1 + P/N$
- Amdahl törvénye: teljesítménynövekmény az 1 processzoros rendszerhez képest:**

$$Sp(N) = \frac{1}{(1-P) + P/N}$$

Ezek alapján: $1-P = 1/4 \rightarrow P = 3/4$, és tudjuk $N=2$ (2 processzoros rendszer). Tehát:

$$Sp(2) = \frac{1}{\frac{1}{4} + \frac{3/4}{2}} = \frac{8}{5} = 1,6$$

4. Összekötős (hálózat)

	a) indirekt összeköttetésű hálózat
1 Busz (12. előadás 35. dia)	b) Egyidejűleg több szereplő is kommunikálhat egymással
2 Crossbar (36. dia)	c) Költség szempontjából a 3 közül ez skálázható legjobban
3 Többfokozatú (37-40 dia)	d) átvitel szempontjából a 3 közül ez skálázható legjobban
	e) üzenetszórást támogat

1 - a,c,e

2 - a,b,d

3 - a,b

5. Jelöld be az igazat Szisztolikus tömbprocesszorra! (11. előadás 35-37 dia)

- Minden egység mindig ugyanazt az egy műveletet hajtja végre
- Van központi vezérlő egység
- Minden egység ugyanazon az egy adaton dolgozik
- Az egységek automatikusan dolgozni kezdenek, ha minden bemeneten megjött az adat

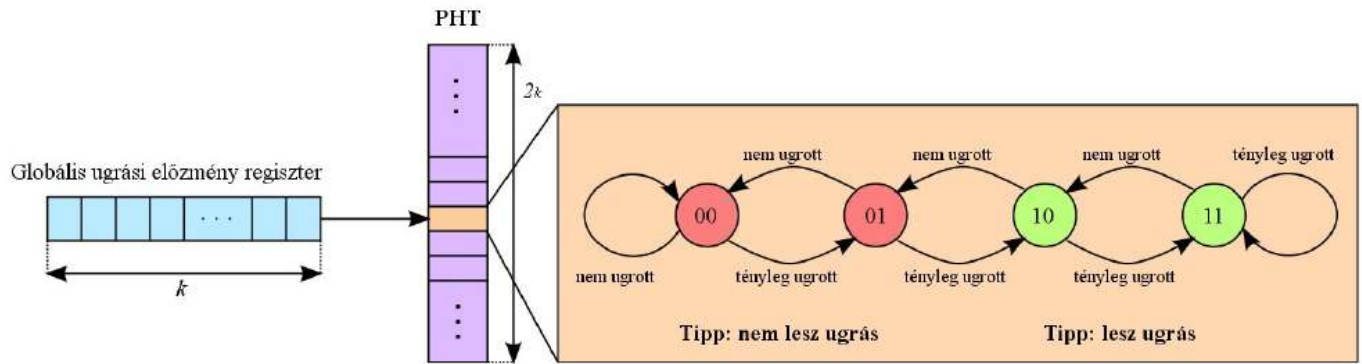
6. Adja meg egy 9 bites globális előzményregisztert és lokális PHT-t használó kétszintű elágazásbecslő eljárást, mely legfeljebb 512 utasítást viselkedését tudja nyomon követni. (10. előadás, 18. dia)

- Hány db PHT-t használ? **512** [lokális PHT esetén utasításonként van egy PHT, és mivel 512 utasítás van, ezért ennyi PHT van]
- Hány bejegyzéses PHT-t használ? **nemtudom :D (2a kilencediken a logikus), szerintem is 2^9 -en, mert a 9 bites branch history regiszter annyiféleképpen tudja indexelni a PHT-t. Lokális, tehát 512 PHT van, abban kettő a kilencediken sor, egy sorban pedig 1 bit, így van két szintje, 0 és 1. 4 szintűnél 2 bit kéne így van 00,01,10,11 szint. Én így értem.**

7. Globális előzményeket és glob. PHT (egyetlen rendszerszintű PHT (ugrási hajlandóság) minden utasításra) használó eljárásban (igaz állítások, nem írtam már le) (10. előadás 15. dia)

Egy db GBHR (globális előzmény regiszter) van, minden ugró utasítás beleshifteli a kimenetelét

- ☐ Egy db PHT van, mely tarolja, hogy az adott GBHR érték (kimenetel kombinációk) mellett mennyire esélyes az ugrás
- ☐ Mar ismerjük (\rightarrow korreláció figyelembe vétele: A trükk: taroljuk az egymást követő utasítások kimeneteleit egy shift regiszterben \rightarrow **globális előzmény regiszter** (Global Branch History Register, GBHR); Ha feltételes ugrás történik, a *tényleges* kimenetele jobbról lép be (0 vagy 1); Egy k bites GBHR az utolsó k ugrás kimenetelét tárolja; A PHT-t ezzel indexeljük)

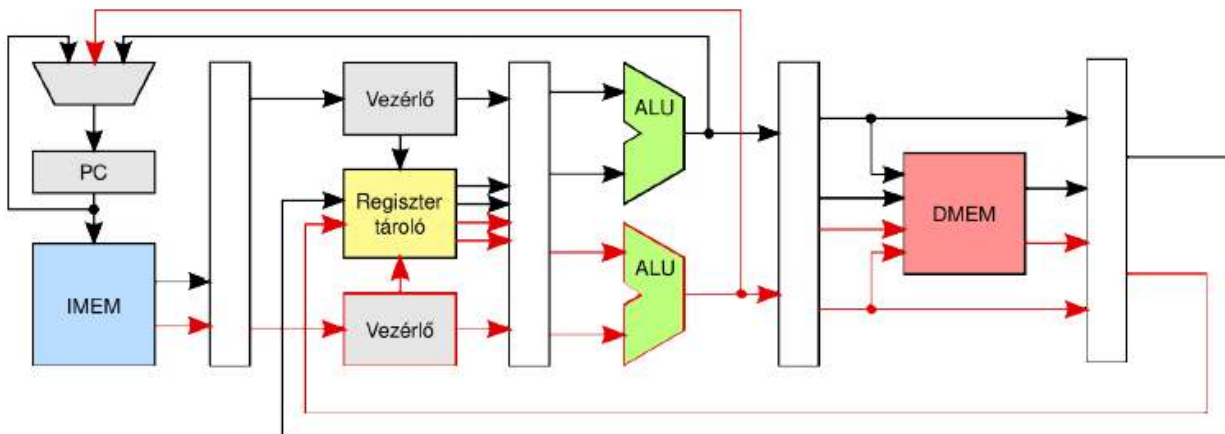


8. Mire jó a regiszter átnevezés? (8. előadás 48-52. dia)

- megszünteti a RAR egymásra hatásokat
- megszünteti a RAW egymásra hatásokat
- **megszünteti a WAR egymásra hatásokat**
- **megszünteti a WAW egymásra hatásokat**

9. Adja meg, hogy egy 2 utas in-order szuperskalár processzor melyikből mennyi van! (8. előadás 9. dia)

- vezérlő egységek száma: 2
- regiszter tárolók száma: 1
- utasítás számlálók száma: 1
- adat cache száma: 1



Két-utas in-order szuperskalár processzor

10. Architektúrák utasítás ütemezés

1 VLIW	a) fordító válogatja össze a független utasítások párhuzamos végrehajtása
2 Epic	b) cache használatát nem teszi lehetővé
3 Szuperskalár	c) A fordító detektálja és kezeli az egymásra hatást
	d) A fordító végzi a regiszter átnevezést

1 - a,b,c

2 - a

3

11. Jelölje be mely fázisok végeznek tényleges munkát egy Load utasítás feldolgozása közben a tanult 5 fokozatú pipeline-nál! (7. előadás 5. dia)

- IF, ID, EX, MEM, WB

Aritmetikai: IF, ID, EX, WB

Store: IF, ID, EX, MEM

Load: IF, ID, EX, MEM, WB

Ugrások: IF, ID, EX

12. Az alábbi 3 utasítás... i3 feldolgozása közben mely pipeline regiszterből veszi az R8 értékét az ALU?

i1: $R8 \leftarrow \text{MEM}[R1+5]$

i2: $R7 \leftarrow R7+R2$

i3: $R5 \leftarrow R8+R7$

Pipeline regiszter: **MEM/WB**

13. Mely összetevőkből áll a keresés egy direkt leképzés cacheben? (6. előadás 14. dia)

- csak a tag összehasonlításból
- **indexelésből, majd a tag összehasonlításából**
- a tag összehasonlításból, majd indexelésből
- egyik sem

14. cache szervezésekre:

	a) A cache mindig a legutóbb használt blokk tartozik
1 Direkt leképzés	b) A kettő közül ez az egyszerűbb és energia takarékos
2 Teljes asszociatív szervezés LRU	c) Transzparens címezés mellett használatos
	d) Elengedhetetlen kelléke a Valid bit

1 – b, a?

2 - d

15. DRAM memóriára igaz állítások. (7. gyak 20. dia – 28; összehasonlítás: 29-31)

- **Tartalmát rendszeresen frissíteni kell**
- **Ugyanannyi tranzisztorral több adatot tud tárolni, mint az SRAM**
- Gyorsabb, mint az SRAM
- A cache memória alapja
- A processzorral egy lapra integrálható

16. Rögzített lapméret mellett a virtuális címek 1 bittel való meghosszabbítása esetén hányszor több bejegyzés lesz ?

- egy egyszintű laptáblánál: **Kétszer több**
- egy inverz laptáblánál: **nem változik**
- virtualizáltnál: ?(szerintem itt is 2szeres lesz)

• Lapok mérete = 2^L

• Címekből alsó L bit: lapon belüli eltolás

• Felső bitek: virt. címeknél lapsorszám, fiz. címeknél keretsorszám

17. Indexelt leképzés megvalósított tárbővítés esetén 16 Bytes CPU címbuszt és 20 bites memória címbuszt feltételezve 16 egyforma méretű ablak használatával

- mekkora az ablakok mérete: $2^{12} = 4\text{Kb}$
- hány index regiszterünk van: 16

18. Split tranzakció szerepe USB 2.0-nál (4 opciót nem írtam le)

LS/FS tranzakciók osztottan.

SSPLIT tranzakció: root hub küld, CSPLIT tranzakció: root hub visszanéz

19. Merevlemez várakozási sorának Forgási késleltetés a seek időt (valami ilyesmi)

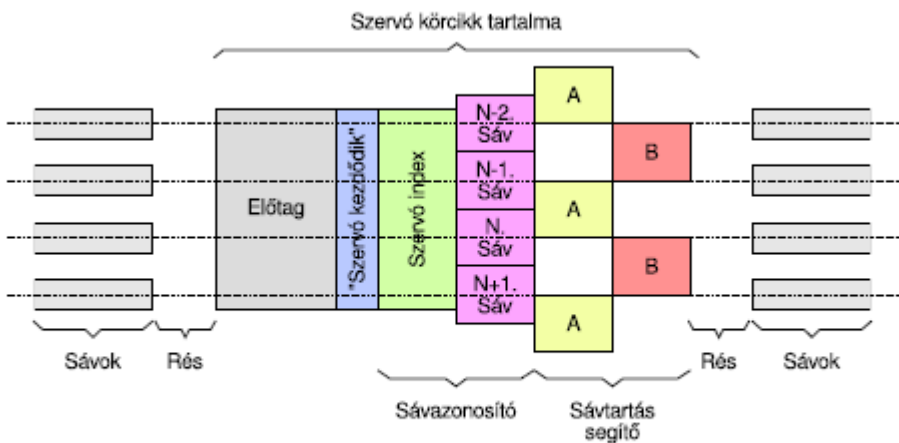
- SSTF -(szerintem) Adathordozóról való leolvasási idejét
- SATF -(szerintem) Adat interfészre való átváltás idejét

Stratégiák kiszolgálási sorrendre:

- 1) Csak seek időre optimalizálva
 - a) SSTF (Shortest Seek Time First)
 - i) Mindig azt választja, amelyik a fej aktuális cilinderéhez legközelebb eső cilinderen van
 - b) LOOK (Elevator Seek)
 - i) Elindul a fej az egyik irányba, az útbaeső igényeket kiszolgálja, majd visszafordul, ismét veszi az útba esőket, stb...
- 2) Teljes hozzáférési időre optimalizálva
 - a) SATF (Shortest Access Time First)
 - b) A forgási késleltetéssel is számol. Amelyik igény a legközelebb esik (seek+forgás), azt szolgálja ki
 - c) Sokkal hatékonyabb
 - d) Sokkal bonyolultabb
 - i) Pontos seek profile kell: meddig tart „a” cilinderből „b”-be tolni a fejet – minden cylinder parra!!!

20. Melyek képezik a merevlemezen tárolt szervó információk részeit?

- Szervó index
- Sávtartást segítő
- Hibajavító kód
- Hibadekódoló kód



4.15. ábra. A szervó információk felépítése

21. Sorolja fel milyen ?összetevőkből áll egy szektor olvasási kérés teljes kiszolgálási ideje! Melyeket lehet ezek közül átlapolni

- Parancsfeldolgozási késleltetés
- Seek idő
- Forgási idő
- Adatleolvasási idő
- Adatok átküldése az interfészen (utolsó 2-t lehet átlapolni)

22. Mikor dől el a PCI-ban hogy melyik eszköz nyeri el az i+1 tranzakció során a kezdeményező szerepét

- Aktuális (i) tranzakció után
- Aktuális (i) tranzakció előtt
- **Aktuális (i) tranzakció közben (rejtett arbitráció)**

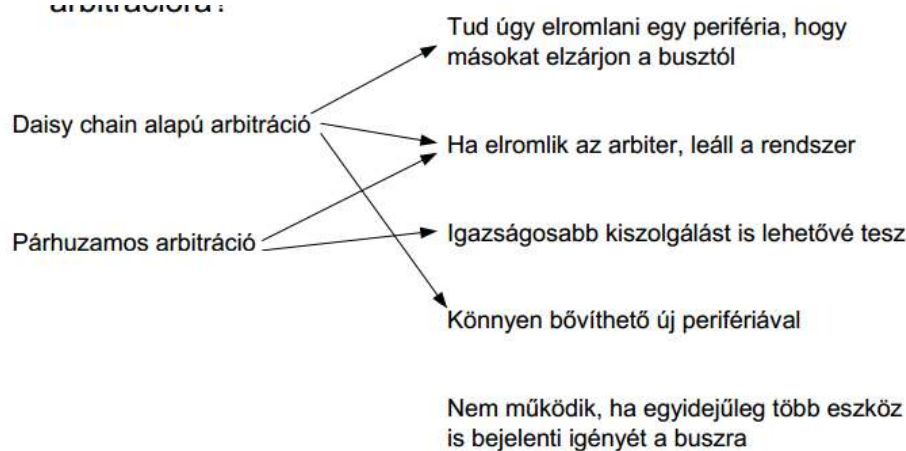
23. A rendszerindítás után lehet beállítani egy PCI/PCI express eszköz konfigurációs regisztereit

- A periféria egyik ablakát célzó tranzakcióval
- **Egy speciális konfigurációs regiszterbe írással**
- Sehogy

24. PIC alapú interruptra (igaz állítás)

- **Körben forgó interrupt kiszolgálás megvalósítható**
- Az eszköz prioritás nem változtatható
- Nem működik ha egyidejűleg több eszköz is jelez interruptot

25. Arbitráció



26. Cisc Risc

Utasítások száma:

- RISC < CISC

Regiszterek száma:

- RISC > CISC

Címzési módok száma:

- RISC < CISC

Program mérete:

- RISC > CISC

27. vezérlés áramlások összeköttetéses feladat (vezérlő token, eredményre szükség van...)

3. Vizsga2

1. Melyik ismeri fel magától a párhuzamosítást (3 féle modell közül)

- adatáramlásos
- igényvezérelt

2. Utasítás fajta:

- $R1 < R2$,

- $R1 \leftarrow R2 + R3$,
- JUMP-42,
- PUSH R1,
- $R2 \leftarrow \text{SQRT } 42$

3. Tranzakciós, adatkapcsolati, fizikai réteg, melyik mit csinál (összekötős)

Tranzakciós réteg	Adatkapcsolati réteg	Fizikai réteg

4. Többprocesszoros gép gyorsasága 1cpu-shoz képest

Amdahl törvény: $1 / ((1-P) + P/N)$ p:párh., 1-p: szekvenciális

5. Aszinkron dram típusok és módok.

- klasszikus aszinkron
- FPM DRAM (Fast Page Mode; gyors lapolvasás)
- EDO RAM (Extended Data Out; FPM DRAM kiegészítése egy kimeneti latch-csel)
- BEDO DRAM (Burst-Mode EDO)

6. NAND- NOR összekötéses

• **NOR flash:** Bitjei egyesével programozhatók, de csak egyben törölhetők. Nem csak háttértárként használják, hanem memóriaként, programok tárolására is. Mivel bajt szinten címezhető, a CPU közvetlenül onnan tudja lehívni az utasításokat. BIOS flash, PDA, telefon, multimédiás eszközök firmware-ének tárolására, stb.

• **NAND flash:** Cél: olcsóbb gyártás (kevesebb vezeték). Egy oszlop source és drain elektródái össze vannak kötve. Kevesebb vezeték → több hely marad a hasznos tranzisztoroknak. Elvileg lehet a biteket egyesével programozni, de csak egyben lehet törölni. Kezdetektől fogva háttértárnak szánták:

- Olvasás és írás egységei: **lapok**. Lap: a tárolómező egy sorának bitjei
- Az egész tároló mező neve: **blokk**
- Törölni csak *teljes blokkot* lehet, lapokat egyesével *nem*
- Tárolási hierarchia:
 - 1 lebegő gate-es tranzisztor: 1 - 3 bitet tarol
 - 1 lap: 512 byte - 8 kB
 - 1 blokk: 128 - 256 lap
 - 1 tárolósík: 1024 blokk
 - 1 szilíciumlapka: 1 - 4 tároló sík
 - 1 tok: 1 - 4 szilíciumlapka

NOR vs NAND:

NOR: nem csak háttértárként használják, hanem memóriaként, programok tárolására is. Mivel bajt szinten címezhető, a CPU közvetlenül onnan tudja lehívni az utasításokat.

• BIOS flash, PDA, telefon, multimédiás eszközök firmware-ének tárolására, stb.

NAND: a lap szerkezete a szekvenciális elérésnek kedvez

• Memóriakártyák (SD, CF, stb.), SSD-k, pendrive-ok, stb.

7. Adatátvitel-egyoldali/kétoldali/szabályozatlan milyen perifériához

Melyik forgalomszabályozás mellett milyen hibák fordulhatnak elő az adatátvitel során?

- **Nincs forgalomszabályozás:** adathiány, adat egymásra-futás.
- **Egyoldali feltételes forgalomszabályozás:** adathiány, adat egymásra-futás.
- **Kétoldali feltételes forgalomszabályozás: semmilyen hiba nem fordulhat elő**

8. Inverz laptáblás teszt (mivel arányos a mérete fiz/virt mem. hova kell tenni)

- Mérete a fizikai memória méretével arányos.
- Működéséhez a teljes adatszerkezetnek a fizikai memóriában a helye.

9. Teljesen asszoc/direkt/n utas asszozhoz összekötötös

Teljesen asszociatív	Direkt leképezés	n utas asszociatív
A blokkok a cache-ben bárhová elhelyezhetők (szabad blokkelhelyezés)	Minden memóriabeli blokk csak egyetlen helyre kerülhet a cache-ben (korlátozott blokkelhelyezés → versenyhelyzet)	Korlátozott blokk elhelyezés, n lehetőséggel → ritkábban van versenyhelyzet
Cache tag: ez a blokk az operatív memória hányas blokkja	A blokkszám alsó bitjei alapján dönti el egyértelműen, hogy hova	A blokkszám alsó bitjei alapján dönti el nem egyértelműen, hogy hova. halmazt jelölnek ki.
Sokat fogyaszt: keresés: cím blokk száma és az összes cache tag komparálása komparátorok szélessége: blokkszám szélessége	Keresés: indexelés+komparálás	Keresés: indexelés+komparálás moderált komplexitás, és fogyasztás
sok, széles komparátor működik benne	csak egy, keskenyebb komparátor dolgozik	csak n komparátor dolgozik

10. A lemezen mindenhol ugyanolyan az adatsűrűség, hogy kell forgatni hogy egyenletesen lehessen olvasni? (1.zh)

- belül -> gyorsan,
- kívül -> lassabban

11. Melyik skálázható jobban adatátvitel szempontjából:

TÖ, 2T ...

12. Állítsuk sorrendbe költséghatékonyság szempontjából:

teljesen összekötött, tórusz, 2d tórusz

13. 5 szintű hierarchikus TLB: hányszor kell a memóriához fordulni,

- ha TLB találat van 0
- TLB hiba van 5

14. n-utas szuperskalár prohoz teszt.

15. Mit tartalmaz a szervó

-> rajz Jegyzet : 73. oldal



4.15. ábra. A szervó információk felépítése

16. Hányszoros lesz a lapméret ha 1 bittel növeljük a fizikai címeket inverz: és virtualizált laptáblánál: ?

17. Miért kell a cache-be előbb beolvasni az adatot?

Idő előtti betöltés (prefetch)

- Kulcsfontosságú funkció
- Cél: a CPU működése ne akadjon meg cache hiba miatt
- Ezért már be kell hozni minden adatot, mielőtt meg először meghívhatkoznak

18. HDD 1 szektor felépítése.



19. Logikai, fizikai regiszter jellemzői (ez mit jelent?) (8. előadás 50. dia – Tomasulo algoritmus)

A regiszter átnevezés alkalmazásához meg kell különböztetni az utasításkészlet által definiált *logikai* vagy *architektúrális* regisztereket a processzoron implementált és használt *fizikai* regiszterektől.

20. N-utas asszociatív leképezésre mi igaz (=2.zh kérdés)?

- **n db azonos indexű blokkot tud tárolni**
- n féle különböző indexet különböztet meg
- **kereséskor n komparátor működik egyszerre**
- a cache tag-eket a rendszermemóriában tárolja

21. Pci, Pci Expressnél milyen irányultságú kommunikációk lehetnek?

- programozott I/O: cpu→periferia,
- a dma: periferia→memoria,
- és a peer-to-peer: periferia→periferia

22. Lehet-e több DMA egy buszon?

IGEN.

23. DMA miért jó?

A DMA vezérlő alkalmazása csökkenti a CPU interrupt terhelését

24. Mit jelent a pontos kivételkezelés?

Olyan “ideális” állapot, amikor a kivételt okozó utasítás előtti utasítások végrehajtása befejeződött, az utána következőké pedig egyáltalán nem.

25. Milyen típusú hibákat kell kezelni pipeline-ban?

- war,
- raw,
- rar,
- waw

26. hullámfront, szisztolikus , klasszikus tömbprocira mi igaz, összekötős (pl. vezérlőegysége van)

Klasszikus	Szisztolikus	Hullámfront
------------	--------------	-------------

vezérlőegység + feldolgozóegységek	feldolgozóegységek kizárólag a legközelebbi szomszédokkal vannak összekötve -> rövid jelutak - > kedvező fogyasztás és magasabb órajelfrekvencia	mint szisztolikus, csak nincs globális órajelhálózat
műveleti egységek mellett lokális memória is	a kommunikáció teljesen szinkronizált (közös órajel)	működése tulajdonképpen adatfolyam alapú
utasításokat üzenetszórással terjeszti	a tömb csak a széleken kapcsolódik a külvilághoz	olcsóbb előállítás
adatcsere sokáig tart, túlmelegszik	nincs központi vezérlőegység	kisebb fogyasztás
használható a vektorprocesszoroknál látott maszkolási technika	hatékony gyártástechnológia	még nagyobb órajelfrekvencia
	könnyen bővíthető	

27. Mikor kell újrarendezni egy programot pl. indexelt leképezés esetén?

A programot az ablakok számának és méretének konkrét ismeretében kell megírni, ha ezek változnak, a programot is meg kell változtatni

4. Vizsga3 - január 14.

1. C, H, S -t kérdezték 1.zh ból

Mely 3 koordinátával azonosít egy szektort fizikailag a merevlemez?

- Cylinder
- Fej
- Szektorszám

Mely 3 koordinátával lehet azonosítani egy pontot egy több adathordozó réteggel ellátott forgólemezes adattárolón?

- Radiális távolság
- Egy fix referenciaponthoz képesti szög
- Az adathordozó rétegek száma.

2. RISC CISC megint volt

Utasítások száma:

- RISC < CISC

Regiszterek száma:

- RISC > CISC

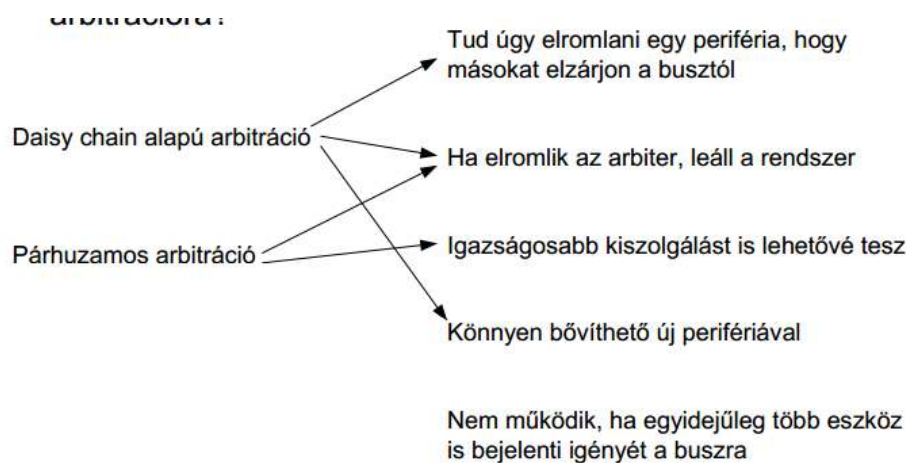
Címzési módok száma:

- RISC < CISC

Program mérete:

- RISC > CISC

3.Daisy chain is



4. Adja meg, hogy egy 3 utas in-order szuperskalár processzor melyikből mennyi van

- vezérlő egységek száma: **3**
- regiszter tárolók száma: **1**
- utasítás számlálók száma: **1**
- adat cache száma: **1**

5. SRAM teszt (cache alapja ...)

- Gyorsabban írható és olvasható, mint a DRAM
- A CPU-val egy lapkára integrálható
- Azonos kapacitás mellett több energiát fogyaszt, mint a DRAM
- A cache memória alapja
- 2 bit tárolásához 12 tranzisztor kell

6. 16 bites címbusz 20 bites memória címbusz 8kb méretű lap mérete(2^{13})

hány laptábla van: $16-13=3$ $2^3=8$ db

7. Neumann architektúra 3 része:

Melyek a Neumann architektúra főbb komponensei?

- **Processzor**
- Cache
- **Perifériák**
- **Memória**
- Rendszerbusz
- Háttértár

8. Rögzített lapméret mellett a FIZIKAI címek 1 bittel való meghosszabbítása esetén hányszor több bejegyzés lesz

- egy egyszintű laptáblánál: nem változik
- egy inverz laptáblánál: kétszer több

9. Mely összetevőkből áll a keresés egy teljesen asszociatív leképzés cacheben

- csak a tag összehasonlításból
- indexelésből, majd a tag összehasonlításából
- a tag összehasonlításból, majd indexelésből
- egyik sem

cím blokk száma és az összes cache tag komparálása

10. Volt még 1 Amdahl tövényes

11. Ár szerint melyik skálázható jobban?

TÖ, 2T, Gy,
sorrendbe kell rakni

12. Melyik fázisban mi történhet?

- **IF fázis:** Laphiba, védelmi hiba
- **ID fázis:** Érvénytelen utasítás
- **EX fázis:** Aritmetikai hiba (pl. integer túlcsordulás)
- **MEM fázis:** Laphiba, védelmi hiba
- **WB fázis:** Itt nem történhet kivétel