

A félév során változhat, kísérjék figyelemmel!

Milyen tárgyalási szintjei lehetnek egy számítógép felépítésének, melyek az architektúra szintjei?

Rajzolja fel a digitális számítógép Neumann-féle modelljének **blokkvázlatát**, sorolja fel a modell működését meghatározó **alapelveket!**

Mi **különbözteti** meg egymástól a memóriában tárolt **utasításokat és adatokat** egymástól?

Mit mond ki Moore törvénye?

Sorolja fel a számítógép generációkat, és adja meg technológiai jellemzőjüket!

Sorolja fel, milyen tényezőktől függ egy számítógép teljesítménye!

Írja fel a számítógép teljesítményét (P) meghatározó kifejezést, adja meg az egyes jelölések jelentését!

Sorolja fel mi jellemzi az utasításkészleteket!

Sorolja fel, és néhány szóval jellemezze az utasításrendszer tervezési szempontjait!

**Írja** be, hogy **négy-címes számítógép** utasításai esetén mit tartalmaznak az egyes **mezők!**

OP. Kód	1. Adat címe	2. Adat címe	Eredmény címe	Köv. utasítás címe
---------	--------------	--------------	---------------	--------------------

Adja meg milyen új **utasítástípust**, illetve milyen speciális célú **regisztert** alkalmaztak, hogy ebből **2 címes** megoldást hozzanak létre

**Utasítástípus:** *Veretlen átadó utasítás*

**Regiszter:** *PC regiszter* ..... **Feladata:** *Szon közvetlen utasítás címét tartalmazza*

Magyarázza el, hogyan lehet a négycímes utasításkészletű számítógépeknél alkalmazott megoldásból **3,- 2,- 1, -1.5 -0** címes megoldást kialakítani!

Magyarázza el mi az előnye, illetve a hátránya az általános, illetve a speciális célú regiszter-használatnak!

Ismertesse a leggyakrabban használt egykomponensű címzési módokat!

Ismertesse az automatikus címmódosító címzés megoldásait!

Ismertesse mit jelent a többkomponensű címzés, adjon egy lehetséges példát!

Milyen többkomponensű címzési mód használható előnyösen egy tömb elemeinek az elérésére, és ez hogyan állítja elő az effektív címet?

**Címzési mód:** *indexelt* ..... **Effektív cím =** *bazis cím + (index regiszter \* méret)*

Mi a stack frame (verem keret) alkalmazásának előnye?

Miben és miért különbözik egy Pascal, illetve egy C programnyelv stack frame implementációja?

Milyen többkomponensű címzési módot alkalmaznak a stack frame esetén, mi ennek az előnye?

Ismertesse a regiszter ablaktechnika megoldást?

Milyen ablaktechnika kialakítással érik el a gyors kontextus váltást?

Ismertesse a CISC, illetve a RISC utasításkészlet jellemzőit!

Jellemezze néhány szóval az alábbi elven kialakított utasításrendszereket	
CISC	RISC
<i>Sok, bonyolult utasítás</i>	<i>Kévs, egyszerű</i>
<i>Sok, bonyolult kévs regiszter</i>	<i>kévs, egyszerű sok regiszter</i>
<i>Regiszter és memória</i>	<i>Regiszterben, memória elérésre LOAD, STORE</i>
<i>Változó, hosszú, több drágul idejű</i>	<i>1 drágul idejű, rövid, állandó</i>
<b>utasítás</b>	
<b>Címzési mód</b>	
<b>Regiszter</b>	
<b>Operandus helye</b>	
<b>Műveleti idő</b>	



Magasszintű programozási nyelveknél függvényhívás (szubrutinhívás) implementálásakor a bemenő paraméterek átadására, illetve a lokális változók tárolására a verem keretet (stack frame) alkalmazzák. Milyen címezési módot használnak a bemenő paraméterek és a lokális változók elérésére, mi a módszer előnye?

Címezési mód: Bázisrelatív  
 Előnye: Relokálható

- SP hívás előtt
- változók megintezése
- SP (új)

Egy Pascal programban adott a következő függvény:

```
function f(i,j,k:integer):integer;
var m,n:integer;
begin .....
end;
```

8086-os processzornál az f(3,1,5) függvényhívás után rajzolja fel (írja be a memóriarekeszek tartalmát) a verem keretet (stack frame), feltételezve, hogy hívás előtt az SP a jelölt helyre mutat (az integer 16 bites)!

**Jelölje** be a felépített keret bázisát (BP). **Jelölje** be a **stack pointer** helyét a függvény végrehajtása alatt, valamint a visszatérés után!

Milyen címezési módot használnak a **bemenő paraméterek** és a **lokális változók elérésére**, mi a módszer előnye?

Címezési mód: Bázisrelatív

Előnye: Relokálható

Mem. cím(hexa)	
EFA02	
<u>SS:SP</u> → EFA00	(SP hívás előtt)
EF9FE	3
EF9FC	1
EF9FA	5
EF9F8	PC
EF9F6	BP
EF9F4	m
EF9F2	n
Ep. v. h. alatt → EF9F0	(SP végrehajtás alatt)
EF9EE	

→ ahova mentek az az az új keret aktuális címe

Egy Pascal programban adott a következő függvény:

```
function f(i,j:integer):integer;
var m:integer;
begin .....
end;
```

8086-os processzornál a fenti függvényt meghívtuk az aktuális paraméterekkel. A mellékelt ábrán a memóriának az a része látható amelybe a verem keret használatakor szokásos szerkezet (stack frame) is felépült (az integer 16 bites). Az SP a keret felépítése után a végrehajtás alatt a bejelölt helyre mutat. Adja meg milyen aktuális paraméter értékekkel hívták meg a függvényt.

$$i = 0013H \dots H \quad j = 001H \dots H$$

Adja meg a keret aktuális (függvény végrehajtása alatti) hexa értékét.

Keret: EF9F8.....H

C nyelvű függvény esetén a bemenő paraméterek helyének felszabadítása a hívó program feladata.

Miért? ő tudja mennyit kell felszabadítani

C nyelvű függvény esetén a bemenő paramétereket fordított sorrendben kell a frame-be írni.

Miért? hogy lehesse tudni mikor jön IP és BP

Mem. cím(hexa)	a)
EFA02	1221H
EFA00	1300H
EF9FE	0013H
EF9FC	0001H
EF9FA	IP
EF9F8	BP
m EF9F6	0A00H
<u>SS:SP</u> → EF9F4	0A01H
végrehajtás alatt	
EF9F2	1221H
EF9F0	0012H
EF9EE	0BC2H

Pascalban → a hívott program feladata



Egy Pascal programban adott a következő függvény:

```
function f(k;l:integer):integer;
var m:integer;
begin .....
end;
```

8086-os processzornál a fenti függvényt meghívtuk az aktuális paraméterekkel. A mellékelt ábrán a memóriának az a része látható amelybe a verem keret használatakor szokásos szerkezet (stack frame) is felépült (az integer 16 bites). Az SP a keret felépítése után a végrehajtás alatt a bejelölt helyre mutat. Adja meg, hogy a  $k$  paraméter milyen aktuális értékével hívták meg a függvényt.

$k = 1221H$

Mi az  $m$  lokális változó pillanatnyi értéke  $m = 2A00H$

Mi a stackpointer értéke közvetlenül a visszatérés után?

SS:SP =  $EFA04$ .....H

Milyen címzési módot használnak a bemenő paraméterek és a lokális változók elérésére, mi a módszer előnye?

Címzési mód: Relatív.....

Előnye: Relokálható.....

Mem.cím(hexa)

EFA02	1221H
EFA00	1500H
EF9FE	CS
EF9FC	IP
EF9FA	BX
EF9F8	BP
EF9F6	2A00H
SS:SP → EF9F4	0A01H
EF9F2	1221H
EF9F0	0012H
EF9EE	0BC2H

végrehajtás alatt

SP usna után

e

e

m

Rajzolja fel egy 1 címes utasításkészletet realizáló CPU ALU szervezésének lehetséges vázlatát!

Mi az előnye a 1,5 címes ALU szervezésnek a következő utasítás címe, illetve az operandus címszámításnak a szempontjából?

Rajzolja fel egy háromsínés regiszterközpontú CPU tipikus felépítését!

Milyen utasításkészlet esetén lehet a fenti elrendezést előnyösen kihasználni?

Mik a CPU vezérlő egységének a fő feladatai?

Rajzolja fel az un. huzalozott vezérlőegység blokkvázlatát, adja meg néhány jellemző tulajdonságát!

Rajzolja fel a mikroprogramozott vezérlőegység Wilkes féle modelljét!

Sorolja fel a mikroprogramozott vezérlőegység előnyös,- illetve hátrányos tulajdonságait!

Rajzolja fel a horizontális szervezésű mikroutasítás formátumot!

Mi az előnye a mezőnként kódolt horizontális mikroutasítás formátumnak?

Rajzolja fel a horizontális kétszintű szervezésű mikroutasítás formátumot, adja meg mi az előnye és a hátránya!

Rajzolja fel a vertikális szervezésű mikroutasítás formátumot!

Mi a fenti megoldás előnye, illetve hátránya a horizontális szervezéssel szemben?

Ismertesse milyen módszereket alkalmaznak a számítógép teljesítményének növelésére!

Sorolja fel az utasítás végrehajtás gyorsításának módszereit!

Mit mond ki Amdahl törvénye a párhuzamos műveletvégzők alkalmazásakor elérhető sebesség növekedésről?

Adja meg a párhuzamos műveletvégzők alkalmazásának hatásfokát /hatékonyságát/ definiáló összefüggést!

Adja meg és magyarázza el, hogyan mérhetjük egy RISC utasításkészlet hatékonyságát az Amdahl törvény módosított formájának felhasználásával!

Ismertesse a RISC processzoroknál alkalmazott elveket!

Ismertesse a processzoroknál alkalmazott PIPE LINE elvét!



Rajzoljon fel egy feldolgozószalagot, rajzolja fel a műveletvégzők ütemezési diagramját!

Mi a különbség a lappangási idő, az újraindítási idő, illetve az utasítás-áteresztő képesség /utasítás kibocsátás/ között?

Rajzolja fel egy ASAP ütemezésű szalag elvi blokkvázlatát!

Milyen utasításrendszerrel lehet célszerű az ASAP ütemezés?

Mit okoz a nagymértékben eltérő műveleti idő a feldolgozó szalag működésében?

Rajzolja fel az elemi műveletvégzők átmeneti túlterhelését kiegyenlítő ASAP ütemezésű szalag blokkvázlatát.

Rajzolja fel egy szinkron ütemezésű szalag elvi blokkvázlatát!

Hogyan lehet a feldolgozó szalag kiegyensúlyozotlanságát megszüntetni?

Milyen utasítás egymásra hatási problémák léphetnek fel a PIPE LINE alkalmazásakor?

EMV = 3

Egy szinkron ütemezésű feldolgozó szalagot alkalmazó processzor három elemi műveletvégzőt tartalmaz. Az első elemi műveletvégző végrehajtási ideje 50ns, a második 30ns, a harmadik 50ns, a részeredmény áttöltéshez szükséges időt elhanyagoljuk.

Hány ns alatt hajtódna végre egy utasítás pipe line nélkül?  $t_1 + t_2 + t_3$

$$T_{ut} = 50 + 30 + 50 = 130 \text{ ns}$$

Hány ns alatt hajtódik végre három utasítás a pipe-line működésekor? *legnagyobb*

$$T_{PL} = 50 \cdot 3 + (3-1) \cdot 50 = 250 \text{ ns}$$

Mekkora az újraindítási idő a fenti esetben?

$$T_{ui} = 50 \text{ ns}$$

Mekkora a szalag sebességnövelő hatása a fenti esetben?

$$S_u = \frac{130}{250} = \frac{13}{25}$$

Mekkora az utasításkibocsátás a fenti esetben /mértékegység is/?

$$TP = \frac{1}{50 \cdot 10^3} = \frac{1}{50 \cdot 10^6} = \frac{1}{50} \cdot 10^{-6} = \frac{1000 \text{ MIPS}}{50}$$

Mekkora a szalag hatékonysága?

$$H < 1$$

$$H = \frac{S_u}{EMV} = \frac{13}{25 \cdot 3} = \frac{13}{75}$$

Melyik ütemezésű feldolgozószalagnál valószínűbb a lappangási idő nagyobb növekedése?

**Milyen módszerrel** csökkenthető az elemi műveletvégző egységek időben egyenlőtlen terheléséből adódó probléma? *Egyforma sebességű egységek alkalmazásával*

Milyen jellegű túlterhelést képes kezelni a műveletvégzők közé iktatott váró sor jellegű puffertároló?

a) **Rajzolja** be a mellékelt ábrába öt utasítás pipe-line elven történő megvalósítását, ha feltételezzük, hogy minden utasítást három elemi műveletvégző dolgoz fel és ezek elemi művelet-végrehajtási ideje egyenlő.

b) **Adjon** egy lehetséges (gyakori) részművelet feladatot a három elemi műveletvégző egységnek (EMV)

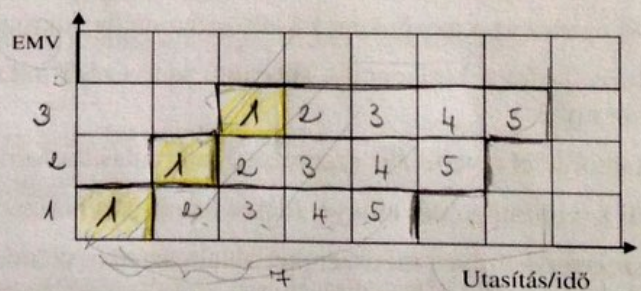
1EMV: ..... *beolvasás* .....

2EMV: ..... *dekódolás* .....

3EMV: ..... *végrehajtás* .....

Hány ns egy utasítás végrehajtási ideje?

$$3 \cdot 40 = 120 \text{ ns}$$



c) **Hány ns** alatt hajtódik végre az öt utasítás ha az elemi műveletvégzők végrehajtási ideje egyenként 40ns (az áttöltéshez szükséges időt elhanyagoljuk)?

$$7 \cdot 40 = 280 \text{ ns}$$

$$\text{Speedup} = \frac{T_{p \text{ nélkül}}}{T_{PL} + \text{átviteli idő}}$$

$$PL: T_L - \sum (t_{uk} + t_r) = \sum t_u = \text{legh.} \cdot EMV_{sz}$$

$$TP = \frac{1 \text{ ut}}{(\text{legnagyobb} + \text{átviteli})} \text{ MIPS} \rightarrow \text{MIPS} = 10^6$$

$$T_{su} = \frac{10^5 \cdot \sum EMV}{\text{legh.} \cdot 10^5 + (EMV_{sz} - 1) \cdot \text{legh.}}$$

$$H = \frac{S_u}{EMV_{sz}} \quad [H < 1]$$

$$T_{ui} = \text{leghosszabb} \quad \text{elhamarabb!}$$

$$T_{PL} = t_u \cdot N + (EMV_{sz} - 1) \cdot t_u \quad \rightarrow t_u \text{ a leghosszabb}$$



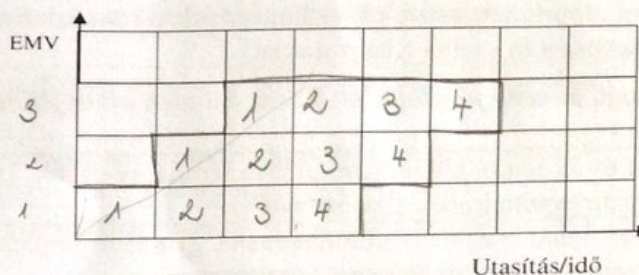
a) Rajzolja be a mellékelt ábrába négy utasítás szinkron ütemezésű pipe-line elven történő megvalósítását, ha feltételezzük, hogy minden utasítást három elemi műveletvégző dolgoz fel.

b) Adjon egy lehetséges (gyakori) részművelet feladatot a három elemi műveletvégző egységnek (EMV)

1EMV: beolvasás

2EMV: dekodolás

3EMV: végrehajtás



Hány ns alatt hajtódik végre egy, illetve négy utasítás, ha az első elemi műveletvégző végrehajtási ideje 4,5ns, a második 3ns a harmadik 4ns a részeredmény átöltéshez illetve kiadásához szükséges idő 0,5ns?

Egy utasítás:  $3 \cdot (4,5 + 0,5) = 15$  ns

Négy utasítás:  $6 \cdot 5 = 30$  ns

Rajzolja fel az utasítások végrehajtását /ütemdiagram/ xx ns-ig 1 ns-os felbontásban (pontossággal) **ASAP (aszinkron) ütemezésű** öt elemi műveletvégzőt tartalmazó pipe-line esetében (Jelölje, hogy a műveletvégző hányadik utasítás melyik elemi műveletét hajtja végre).

A műveletvégzők **üres idejét** (várakozás az előző eredményére) jelzéssel, a **foglaltságot** (következő műveletvégző felszabadulására várás) jelzéssel jelölje. A **műveletvégzők** működési ideje már tartalmazza az **átöltési időt** is. A műveletvégzők működési idejei a következők:  $UB=5ns$ ,  $UD=4ns$ ,  $EC=5ns$ ,  $MEM=10ns$ ,  $VI=4ns$ .

Rajzolja fel az utasítások végrehajtását yy ns-ig ns-os felbontásban (pontossággal) szinkron ütemezésű öt műveletvégzőt tartalmazó **pipe-line** esetében. A műveletvégzők üres idejét (várakozás) a jelöléssel jelölje!

A műveletvégzők működési idejei a következők:  $UB=5ns$ ,  $UD=4ns$ ,  $EC=5ns$ ,  $MEM=10ns$ ,  $VI=4ns$

EMV<sub>re</sub> = 5  
egyhosszabb: 100

\* Mekkora a lappangási idő?  $T_L = 10 \cdot 5 = 50$  ns. Mekkora a sebességnövekedés?  $Su_{PL} = 2,18$

Hogyan lehetne  $Su_{PL}$ -t növelni? *A leglassabbat ameddig tudjuk, felbontjuk (felezésűt, negyedévesűt...)*

Mi a hatása, ha egy feldolgozó szalag minden műveletvégzőjét további elemi lépésre bontunk?

Mi a hatása, ha egy feldolgozó szalag műveletvégzőinek a számát növeljük, illetve csökkentjük?

Mit jelent a feldolgozási egymásra hatás?

Soroljon fel néhány módszert a feldolgozási egymásra hatás kiküszöbölésére!

Mit jelent az adat egymásra hatás, és hogyan küszöbölhető ki?

Mi okozza a **procedúrális** utasítás egymásra hatást?

Hogyan oldja meg az **i386-os** mikroprocesszor a **procedúrális** utasítás egymásra hatást?

Hogyan oldja meg az **i486-os** mikroprocesszor a **procedúrális** utasítás egymásra hatást?

Miben különbözik a **Pentium megoldása az i486-nál** alkalmazott módszertől?

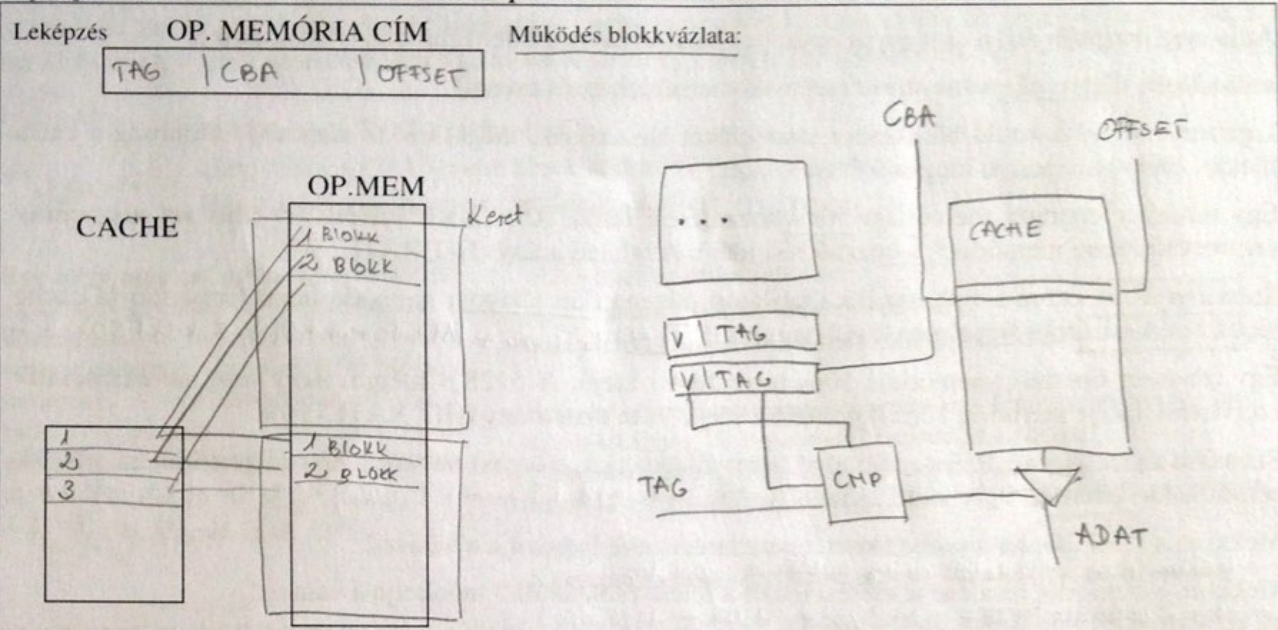
Mit jelent az  $n$  utas feldolgozó szalag kifejezés?

Mit jelent a Pentium 4 feldolgozó szalagnál a hiper feldolgozó szalag kifejezés?

*számánövekedés*

$$* Su_{PL} = \frac{10^5 \cdot \sum EMV}{\text{egyhosszabb} \cdot 10^5} = \frac{10^5 \cdot (5 + 4 + 5 + 10 + 4)}{10 \cdot 10^5 + (4 \cdot 10)} = \frac{10^5 \cdot 28}{10^5 \cdot 10} = 2,18$$





Röviden ismertesse a **leképzés elvét** és a **működést**  
 OM-ot Cache méretű kerekeltre osztjuk. Minden kerekben ugyanannyi blokk van mint a cacheben adott blokk csak a neki megfelelő számú cache belső blokkra léphet.  
 Mi a **leképzés előnye** és a **hátránya, hogyan csökkentik** az utóbbit?  
 Egyszerű, gyors, csend, blokkcsera algoritmus nélküli. HIT rate alacsony, mert növelésével HR javul. Legjobb kompromisszum n utas direkt leképzés. Asszociatív leképzéssel kombinált direkt leképzés.

Használhatnak LRU blokk-csere (block replacement policy) stratégiát?  
 Válasz: **LRU → NEM**  
 Indoklás: Minden blokk 1 helyre kerülhet.  
 Hogyan **változna** a komparátorok száma kétutas direkt leképzés esetén?  
 Változás: 1 helyett 2 komparátor kellene.  
 Indoklás: Mindkét részben külön külön meg kéne nézni, hogy min-c ott, az adott blokk mindegyik részbe kerülhet.

Hasonlítsa össze a különböző leképzési stratégiákat a találati arány, a keresés sebessége és a megvalósítás bonyolultsága szempontjából!

A cache kezelés egyik kérdése, hogy mikor hozzunk be egy blokkot a cache-be. Sorolja fel és egy mondatban ismertesse az itt használatos stratégiákat (fetch policy)!

Mit jelent az előrelátó behozatali stratégia? Ha behozunk egy blokkot, akkor hozzuk az utána következőt is, ha az szükség van rá.  
 Mit jelent az okos előrelátó behozatali stratégia? Megpróbálja kikövetkeztetni, hogy mely blokkokra lesz szükség.

**Sorolja fel** és ismertesse a cache blokkcsera stratégiákat! *utó helyre, LRU, LFU, FIFO, random*

A cache szervezésnél **igény szerinti, előrelátó és szelektív blokk-behozatali** (fetch) stratégiákat alkalmazhatnak. Adjon **legalább két példát** a **szelektív** stratégiára! *Cím szerint szelektív / típus szerint*

A **cache kezelés** egyik kérdése a **blokkbehozatal** (fetch policy).

Alkalmazható e az **igény szerinti** stratégia **direkt**, illetve **n utas** direkt leképzésnél?  
 Válasz: *igen*..... (direkt)

Indoklás: *Semmi nem gátolja. Ha kell, hozzuk, a megfelelő helyre, vagy az utána következőt asszociatív.*

Mit jelent az előrelátó stratégia? *Ha behozunk egy i-edik blokkot, hozzuk (i+1)-et is!*

Milyen **szelektív** behozatali stratégiát valósít meg a **486-os**, illetve a **Pentium** processzor?

**486**: *Egyszerű előrelátó, mely behozza a két blokkot is.*

**Pentium**: *Okos előrelátó, mely megpróbálja kikövetkeztetni mire lesz szükség.*

↳ *(superbarán architektúra, 2 fázisú, általában több mint 1 utasítás végrehajtás)*



Az operatív memória 32MB, a cache 64KB, a blokkméret 128 byte

Adja meg (rajzolja fel) a cím egyes részeit és határozza meg bitenkénti elhelyezkedésüket asszociatív, illetve négyutas direkt (set asszociatív) leképzés esetén!

Lapszervezésnél hasonló blokkcsere stratégiákat használnak, mégis mi az alapvető különbség a cache-blokk-, illetve a lapcsere megvalósítása között?

Egy rendszer operatív memóriája 50ns hozzáférési idejű. A 1024KB méretű két utas set asszociatív szervezésű cache memória 5ns hozzáférési idejű. A találati arány (HIT RATE) 90%.

Számítsa ki mekkora a felhasználó által látott átlagos (látszólagos) memória hozzáférési idő (a cache blokk betöltési idejét figyelmen kívül hagyjuk)!  $T_L = HR \cdot T_{cache} + MR \cdot T_{pmem} = 0,9 \cdot 5 + 0,1 \cdot 50 = 4,5 + 5 = 9,5 ns$

Egy rendszer operatív memóriája 50ns hozzáférési idejű. A 512KB méretű négy utas set asszociatív szervezésű cache memória 10ns hozzáférési idejű. A találati arány (HIT RATE) 90%.

Számítsa ki mekkora a felhasználó által látott átlagos (látszólagos) memória hozzáférési idő, ha a blokk betöltési idő 100ns?  $T_L = 0,9 \cdot 10 + 0,1 \cdot (50 + 100) = 9 + 15 = 24 ns$

Mekkora a fenti idő, ha a cache méretét megduplázzuk? Indokolja a választ!

Mekkora a fenti idő, ha a cache elérési idejét a felére csökkentik? Indokolja a választ!

Mekkorára kell csökkenteni a cache elérési idejét a fenti esetben, hogy a  $T_L$  a felére csökkenjen?

Indokolja a választ! A cache-t nem lehet elég gyorsítani, mert a blokkbetöltési idő és az operatív memória hozzáférési idő túl nagy ezek a kis adatok beírásánál. MR-t lehet drasztikusan csökkenteni vagy a memória hozzáférési időt.

Hogyan lehetne a fenti megoldásnál jó eséllyel a találati arányt növelni? Indokolja a választ!

Rajzolja fel egy hierarchikus felépítésű cache /pl.: Pentium/ vázlatát!

Sorolja fel a cache írási stratégiákat! write back, write through  
Vázolja fel, és magyarázza el az információ áramlását a keresztülíró stratégiánál találat, illetve hiány esetén, áthelyezéssel, illetve áthelyezés nélkül!

Vázolja fel, és magyarázza el az információ áramlását Write back írási stratégia esetén!

Soroljon fel néhány módszert az adat konzisztencia biztosítására!

Az egyik legegyszerűbb cache-írási stratégia a write through stratégia. Mi történik ennél a stratégiánál egy byte írásakor, ha a hivatkozott adat blokkja bent van a cache-ben?

Atérünk cache-ben, majd onnan kiírjuk DM-ba.  
Használható-e ez a stratégia lapszervezésű virtuális tárkezelés esetén? Indokolja a választ!

Nem.  
Indoklás: Nem biztos, hogy a hivatkozott lap az DM-ben található.

Rajzolja fel cache blokk behozatalnál, a gyorsítás érdekében, alkalmazott memória átlapolás (memória interleaving) blokkvázlatát! Magyarázza el a működését!

Adja meg kb. mekkora gyorsítást érnek el egy nyolcszoros átlapolásnál... 8x.....

Használható-e a módszer direkt leképzés esetén? Igen

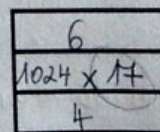
Indoklás: Ha az DM-et jól osztjuk fel, akkor lehet. Nem szabad egymás után ugyanolyan sorozámú blokkot venni

Egy négy utas set asszociatív vezérlést alkalmazó gyorsító tár(cache) adatai a következők: a blokkméret 64 byte, a teljes gyorsító tár összesen 4096 blokkot tartalmaz. A vezérlőtárakban (TAG) 1 bittel jelezzük a bejegyzés érvényességét. Az operatív memória címe 32 bites.

Hány bites az offset (eltolás)?

Milyen szervezésű (szószám x bitszám) egy vezérlő (TAG) tár?

Hány TAG komparátort tartalmaz a cache?



$$\frac{4096}{4} = 1024$$

$$1-1$$

OPMEM = 32 bit. CA = 10  
Blokkméret = 64 byte  
 $64 = 2^6$   
TAG = 32 - 10 - 6 = 16  
BITZÁM = TAG + V = 17

$2^{CA} = \text{blokkszám}$   
 $TAG = \text{OPMEM} - CA - \text{OFFSZET}$   
Blokkméret = 2 offset



Az operatív tárhoz egy **4 utas direkt leképzésű cache** kapcsolódik. Az operatív tár byte-os szervezésű, a cím 32 bites. A teljes cache összesen 512 blokkot tárol, egy blokk 256 byteból áll.

$\text{Blokkszám} = 512$   
 $\text{Blokkméret} = 256$   
 $n = 4$

$\text{TAG} = \text{OPMEM} - \text{CBA} - \text{OFFSET} = 32 - 7 - 8 = 17$

- Hány bites a TAG a cache-ben?
- Hány bites a CBA (cache block address)?
- Hány TAG-komparátort tartalmaz a cache?

17
7
4

$\frac{512}{4} = 128 \xrightarrow{\text{CBA}} \text{CBA} = 7$

Egy **négy utas set asszociatív** vezérlést alkalmazó gyorsító tár (cache) adatai a következők: a blokkméret 64 byte, a teljes gyorsító tár összesen 4096 blokkot tartalmaz. A vezérlőtárban (TAG) 1 bittel jelezzük a bejegyzés érvényességét. Az operatív memória címe 32 bites.

- Hány bites az offset (eltolás)?
- Hány bites a CBA(blokkazonosító) mező?
- Hány TAG komparátort tartalmaz a teljes cache?

6
10
4

Hogyan **változhatna a találati arány** ha a fenti cache-ben két utas direkt leképzést alkalmaznánk? **Indokolja** a választ!

**Változás:** ...  
**Indoklás:** ...

**Komparátorok száma:** ...-ről(ról) ...-ra(re)  
**Komparátor(ok) bitszélessége:** ...bitről...bitre

Egy **teljesen asszociatív** vezérlést alkalmazó gyorsító tár adatai a következők: a blokkméret 128 byte, a gyorsító tár mérete 256 Kbyte, a hozzáférési idő 10 ns. A vezérlőtárban 1 bittel jelezzük a bejegyzés érvényességét. A számítógép címsínje 32 bites, az operatív memória hozzáférési ideje 75 ns, a találati hibák aránya 8%.

$256 \text{ Kbyte} = 256 \cdot 10^3 \text{ byte}$   
 $256 = 2^8 \cdot 2^8$

- Hány bites az offset (eltolás)?
- Milyen szervezésű (szószám x bitszám) a vezérlőtár?

7
256 x 26
15,2

Mennyi a memória átlagos elérési ideje?

A **fenti** cache szervezésnél **LRU, LFU, FIFO, RANDOM blokkcsere stratégiákat** alkalmazhatnak.

Mit jelent az **LFU** stratégia? ...

**Használható e LRU** blokkcsere stratégia **direkt** leképzésű cache-nél? **Indokolja** a választ!

Az operatív tárhoz egy **direkt leképzésű cache** kapcsolódik. Az operatív tár byte-os szervezésű, a cím 32 bites. A teljes cache összesen 512 blokkot tárol, egy blokk 256 byte-ból áll.

- Hány bites a TAG a cache-ben?
- Hány bites a CBA (cache block address)?
- Hány TAG-komparátort tartalmaz a cache?

15
9
1

Az operatív tárhoz egy **direkt leképzésű cache** kapcsolódik. A behozatali stratégia igény szerinti, az írási stratégia write-through. Az operatív tár byte-os szervezésű, a cím 32 bites. A CBA bitek száma 9, egy blokk 256 byteból áll.

- Hány TAG-komparátort tartalmaz a cache?
- Hány bites a TAG komparátor?
- Számítsa ki a cache méretét (Kbyte-ban)?

1
15
128

Egy **négy utas set asszociatív** vezérlést alkalmazó gyorsító tár (cache) adatai a következők: a blokkméret 256 byte, a teljes gyorsító tár mérete 128Kbyte, a hozzáférési idő 10 ns. A vezérlőtárban (TAG) 1 bittel jelezzük a bejegyzés érvényességét. A cache vezérlő write through with write allocate írási stratégiát alkalmaz. A számítógép címsínje 32 bites, az operatív memória hozzáférési ideje 75 ns, a találati arány HR=92%

- Hány bites az offset (eltolás)?
- Milyen szervezésű (szószám x bitszám) egy vezérlő (TAG) tár?
- Mennyi a memória látszólagos átlagos elérési ideje?

8
128 x 18
15,2

$\text{Teljes cache} = 128 \text{ Kbyte} \rightarrow 10 \text{ ns}$   
 $n = 1 \rightarrow \text{CBA} = 7$

$\text{OPMEM} = 32 \rightarrow 75 \text{ ns}$   
 $\text{HR} = 0,92$   
 $\text{MR} = 0,08$

$T_E = 0,92 \cdot 10 + 0,08 \cdot 32$   
 $9,2$

$\text{TAG} = 32 - 8 - 4$



Blokkméret:  $128 \rightarrow \text{Offset} = 7$  blokkszám =  $2048 \rightarrow \text{CBA} = 10$  TAG =  $32 - 10 - 7 = 15$

Egy két utas set asszociatív vezérlést alkalmazó gyorsító tár(cache) adatai a következők: a blokkméret 128 byte, a teljes gyorsító tár összesen 2048 blokkot tartalmaz. A vezérlőtárakban (TAG) 15 bittel jelezzük a bejegyzés érvényességét. Az operatív memória címe 32 bites.

Hány bites az offset (eltolás)?

Milyen szervezésű (szószám x bitszám) egy vezérlő (TAG) tár?

Hány TAG komparátort tartalmaz a cache?

7
1024 x 16
20

Rajzolja fel a tömbkapcsolásos elven működő memóriabővítés blokkvázlatát! Röviden magyarázza el a működést, sorolja fel előnyeit és hátrányait!

Milyen tárkapacitás problémát old meg a **Tömbkapcsolásos** memóriakezelés?

Milyen követelményt támaszt ez a megoldás a tömbkapcsolót vezérlő, illetve a megszakítás kezelő programokkal szemben?

Rajzolja fel az **Indexelt leképzés** elvén működő memóriakezelés blokkvázlatát! Röviden magyarázza el a működést, sorolja fel előnyeit és hátrányait!

**Indexelt leképzés** esetén a logikai cím 16 bites, amelyből a legmagasabb helyértékű 4 bit az index. A fizikai memória mérete 1Mbyte. Számítsa ki mekkora az **indexregiszter-tömb** mérete, ha vezérlésre 2 bitet alkalmaz! Adott egy 4KB (1000H) kezdőcímmel címfolytonosan lefordított 12KB méretű program. A fizikai memóriában a 512KB..520KB (80000H-81FFFH) és az utolsó 8KB tartományban (FE000H-tól) van szabad memóriahely. Írja fel az indexregiszter-tömb **programhoz tartozó** regisztereinek sorszámaát és a címrészének az értékeit!

$2^{20} \rightarrow 20$  regiszter-tömb.  $20 - 4 = 16$   $16 - 4 \rightarrow \text{offset}$   $\text{offset} - \text{vez} = 12$

<p><b>Indexelt leképzés</b> esetén a logikai cím 16 bites, amelyből a legmagasabb helyértékű 4 bit az index. A fizikai memória mérete 1Mbyte. Számítsa ki mekkora az indexregiszter-tömb mérete, ha vezérlésre 2 bitet alkalmaz! Adott egy 1000H kezdőcímmel (4KB) címfolytonosan lefordított 12KB méretű program. A fizikai memóriában az 512KB..520KB (80000H-81FFFH) és az utolsó 4KB tartományban (FE000H-tól) van szabad memóriahely. Írja fel az indexregiszter-tömb a program futásakor felhasznált regisztereinek sorszámaát hexadecimális tartalmát, ha a legfelső helyértéken alkalmazott vezérlő bitek értéke 01!</p>	Indexregiszter-tömb mérete (regiszter x bit)	16 ..... x 10 20-index
	Regisztersorszám	Hexadecimális érték
	1	0180 = 180H
	2	0181 = 181H
3	01FF = 1FFH	

Egy 16 bites logikai és egyben fizikai címmel rendelkező rendszerben (pl.: 8085) indexelt leképezésű (index regiszter-tömböt alkalmazó) memóriaszervezéssel 1Mbyte-ra kell bővíteni a fizikai memória méretét. Az index regiszter-tömb 32 regisztert tartalmaz. Vezérlésre (adminisztrációra) 3 bitet használunk)

Hány bites az offset (eltolás)?

Hány bites a regiszter tömb egy regisztere?

Hány blokkot (lapot) tartalmaz a teljes memória?

Maximálisan hány blokk (lap) lehet egyszerre aktív?

11
12
512
32

Blokkok =  $(20 - 06) / 2$

Ha 32 regiszter  $\rightarrow 2^5 \rightarrow 5$  bittel ábrázál

16 bites rendszer  $\rightarrow 16 - 5 = 11 \rightarrow$  marad offsetnek.

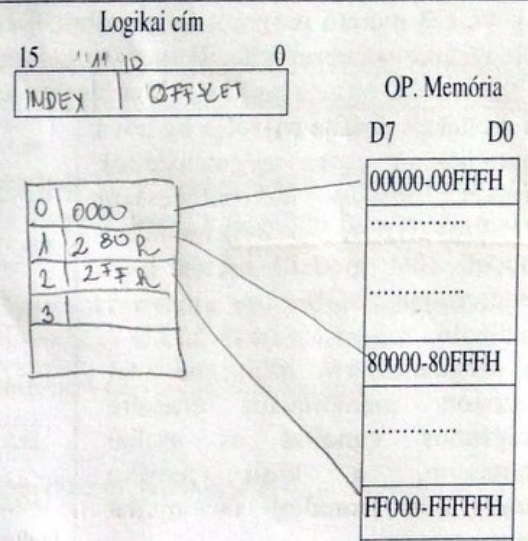
1MB  $\rightarrow 2^{20} \rightarrow 20 \rightarrow 20 - 11 = 9$

Regiszter tömb regiszter =  $20 - \text{offset} + 1 = 20 - 11 + 1$

9 bites nagy blokk:  $2^9 = 512$ , de csak 32 regiszter van, így egyszerre ennnyi lehet aktív.



Rajzolja fel az **Indexelt leképezés** elvén működő memóriakezelés blokkvázlatát a következő adatok felhasználásával: **logikai cím 16** bites, amelyből a legmagasabb helyértékű **4** bit az **index**, **vezérlésre 2** bitet alkalmaz. A fizikai memória mérete **1Mbyte**! Az **előző leképezésnél** adott egy **1000H** kezdőcímmre címfolytonosan lefordított **7KB** méretű program. A fizikai memóriában az **512KB.-516KB** (80000H-80FFFH) és az utolsó **4KB** tartományban (FF000H-tól) van szabad memóriahely. Írja be az indexregiszter-tömb **programhoz tartozó** regisztere(i)nek sorszámát és a tartalmának **hexa** értéke(i)t, feltételezve, hogy a legmagasabb helyértéken lévő vezérlő bitek értéke **10**! **Mit kell biztosítani, és hogyan**, ha a regisztertömböt író és a megszakításkezelő programok logikailag és fizikailag az első 4kB-on helyezkednek el?



*Biztosítani kell hogy az első 4KB mindenképp aktív legyen. Első regiszter pontosan 0-t tartalmaz, NEH programéhoz*

Egy indexelt leképezést alkalmazó számítógép címsínya 16 bites, ebből a 3 MSB az index. Az operatív memória 256 kByte méretű.

Milyen az indexregiszter tábla szervezése (szó x bit) ha vezérlési célokra 2 bit szükséges?

Ha egy 1980h byte hosszú programrészt a hexa 8080h címre fordítunk le, hányas indexű indexregiszter(ek) tartamát kell beállítani?

Milyen (hexa) értéket kell ebbe az indexregiszterbe írni, ha a fizikai memóriában 2A000h-2BFFFh tartományban van üres hely a program számára?

Egy indexelt leképezést alkalmazó számítógép logikai címe 16 bites, ebből a 4 MSB az index. Az operatív memória 512 kByte méretű.

Milyen az indexregiszter tábla szervezése (szó x bit) ha vezérlési célokra 2 bit szükséges?

Ha egy 0E80h byte hosszú programrészt a hexa 8080h címre fordítunk le, hányas indexű indexregiszter(ek) tartamát kell beállítani a program futtatásához?

Milyen (hexa) értéket kell ebbe az indexregiszterbe írni, ha a fizikai memóriában 7A000h-7AFFFh tartományban van üres hely a program számára és az 11 értékű vezérlési célú bitek a legmagasabb helyértéken vannak?

Adott egy címfolytonosan 0 kezdőcímmre lefordított 48kB méretű program. A processzor 16 bites címet generál. Az indexelt leképezéssel kibővített fizikai memóriában a 0... 32 KB és a 64... 84 KB tartományban van szabad memóriahely. Az indexregiszter tábla 32db regisztert tartalmaz.

Kell-e változtatni a lefordított felhasználói programon? Indokolja a választ!

**Ismertesse** a lapszervezésű virtuális tárkezelés elvét, rajzolja fel a megoldás vázlatát!

Ismertesse mikor jelentkezik egy laphiba jelzés és ismertesse az utána szükséges lépéseket!

Hogyan csökkenthető a leíró táblához szükséges memória mérete?

Hogyan gyorsítható a leírók elérése, hol helyezkedhet el a TLB?

Hogyan befolyásolja a TLB elhelyezkedése a sebességet, mi a különböző megoldások előnye, hátránya?

**Ismertesse** a szegmensszervezésű virtuális tárkezelés elvét, rajzolja fel a megoldás vázlatát!

Hasonlítsa össze a lapszervezésű és a szegmensszervezésű virtuális tárkezelés előnyeit - hátrányait!

Milyen memóriaképeket /memória elérési modelleket/ biztosíthat a 386-os mikroprocesszor?



Egy 48 KB méretű program futtatásához mekkora méretű lineárisan összefüggő szabad memóriaterület kell szegmensszervezésű, illetve lapszervezésű virtuális tárkezelés esetén? Indokolja a válaszokat (feltételezzük, hogy a laphiba lekezelése után az utasítás folytatható)!

A mellékelt ábrába rajzolja be hová mutatnak az egyes szegmensleírók értékei 386-os mikroprocesszor **strukturálatlan** (lineáris) memóriamodel (flat modell) esetén ha a rendszerben 4GB op. memória található.

A szegmensleírók báziscíme által mutatott memóriacím értékére **folytonos vonallal** és nyíllal mutasson, a limit helyére **szaggatott vonallal** és nyíllal mutasson.

Minden szegmens 0 kezdőcímen 4 GB hosszú, tehát a 0 a kezdőcíme, tehát a limit.

A mellékelt ábrába rajzolja be hová mutatnak az egyes szegmensleírók értékei 386-os mikroprocesszor **strukturált** memóriamodel (protected multisegment modell) esetén ha a rendszerben 4GB op. memória található.

A szegmensleírók báziscíme által mutatott memóriacím értékére **folytonos vonallal** és nyíllal mutasson, a limit helyére **szaggatott vonallal** és nyíllal mutasson.

Minden szegmens 0 kezdőcímen 4 GB hosszú, tehát a 0 a kezdőcíme, tehát a limit.

Rajzolja fel a 386-os mikroprocesszor logikai-fizikai cím transzformációjának a vázlatát, protected módban bekapcsolt lapszervezés esetén!

Milyen megoldásokat használ a processzor a transzformáció végrehajtásának gyorsítására?

Mi a legfontosabb előnye a szegmentált lapszervezésnek az egyszerű szegmensszervezéssel szemben, és mi a hátránya?

Előnye: ... csak a logikai címtartományok kell, egyértelműsége, könnyű adat szegmenssel ...

Hátránya: ... Wajjal... memóriaterület szükség a megvalósításához ...

A 386-os mikroprocesszor példáján magyarázza el, hogy miért előnyös a kétlépcsős lapszervezésű memóriakezelés az egylépcsőshöz képest! Mi a hátránya és ezt hogyan csökkenti a 386-os?

Milyen hátránnyal járna az egylépcsős laptábla alkalmazása?

Hogyan csökkenti a 386-os mikroprocesszor a kétlépcsős laptábla alkalmazásának hátrányát (indokolja a válaszokat)?

A 386-os mikroprocesszor bekapcsolt lapszervezés esetén a 32 bites lineáris címből kétlépcsős laptábla szervezéssel állítja elő a fizikai címet. A lapméret 4Kbyte. A laptábla könyvtár elejére a CR3 regiszter tartalma mutat. Egy lapleíró bejegyzés 32bitet tartalmaz. Lapméret 4Kbyte  $\rightarrow 4 \cdot 2^{10} = 2^{12}$   $\rightarrow$  előadás (4096) bit

$2^2 - 1 = 20$  bit marad a lineáris címtől  $\rightarrow$  comp lehet CR3-ban, 2 lépéses kell  $\rightarrow$  lapleíró is laptábla:  $4 \cdot 4 \cdot 2^{10} \rightarrow$  kell 8 Kb kell

Minimálisan mekkora op. memória kell a kétlépcsős laptábláknak?

Mekkora lenne a laptábla mérete egylépcsős szervezéssel?

Mi a kétlépcsős szervezés hátránya az egylépcsőssel szemben?

Hány értékes bitet kell a CR3-nak tartalmaznia? 20 (felül 20 értékes bit  $\rightarrow$  1 laptábla van valahol)

laptábla mérete egylépcsős esetben:  $2^2 \cdot 32 \cdot 10^3$  bit

Optimalis érték 1-nél kell lennie (1 laptábla van az összes)



Ismertesse az i386 szegmensekre vonatkozó védelem lépéseit!

**Ismertesse** az i386 mikroprocesszor lapokra vonatkozó védelmét!

386-os mikroprocesszornál egy **248 KB** méretű program **futtatásához** mekkora méretű **lineárisan összefüggő** szabad memóriaterület kell **KI-**, illetve **BE**-kapcsolt lapszervezésű virtuális tárkezelés esetén? **Miért?**

386-os mikroprocesszornál, egy konkrét memóriahivatkozásnál a **szegmensleíró felhasználói** (3) privilégium szintet jelez. A **laptábla könyvtár** (directory) bejegyzés **user**, a laptábla bejegyzés pedig **system** beállítású. Mi történik? (**Indokolja** a választ!)

A 386-os processzornál egy memória típusú szegmens leírója tartalmazza a szegmens kezdőcímét, a hosszát és az attribútumait. **Miben különbözik** ettől (mit tartalmaz) egy **CALL GATE** leírója?

Rajzolja fel és magyarázza el egy CALL GATE-en keresztül történő vezérlés átadás vázlatát!

Mit tartalmaz egy TASK GATE?

Rajzolja fel egy TASK GATE működésének a vázlatát és magyarázza el a működést!

Miért kell TASK GATE-et használni?

Vázolja fel egy TASK több helyről történő elérésének vázlatát!

**Ismertesse**, hogy multitaszkos rendszerben mit értünk virtuális processzoron!

**Mi a különbség a virtuális processzor és a fizikai processzor között?**

**Hogyan** történik a virtuális és a fizikai processzor összerendelése?

A **rendszer** melyik komponense(i) végezheti(k) a fizikai-, virtuális processzor összerendelést!

Milyen támogatást nyújt az I386-os mikroprocesszor a taszkváltáshoz? Ismertesse az ehhez szükséges adatstruktúrát (TSS) és a taszkváltás lépéseit!

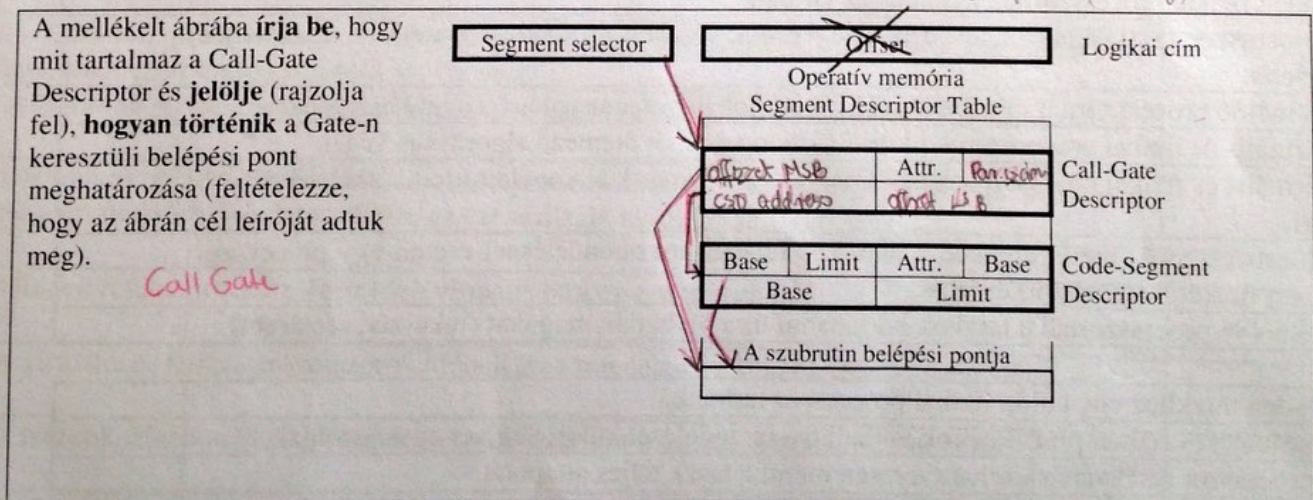
**Sorolja** fel az i386-os mikroprocesszor privilégium szintjeit, és a hozzájuk kapcsolódó elérési szabályokat!

**Ismertesse** az i386 lapokra vonatkozó védelmét!

Mi a GATE(kapu) szerepe, hogyan biztosítja ezt?

**Rajzolja** fel a CALL GATE működését jellemző szegmenselérés vázlatát!

(4 nyit) (4 válasz)



Mi a Task GATE szerepe, hogyan biztosítja ezt?

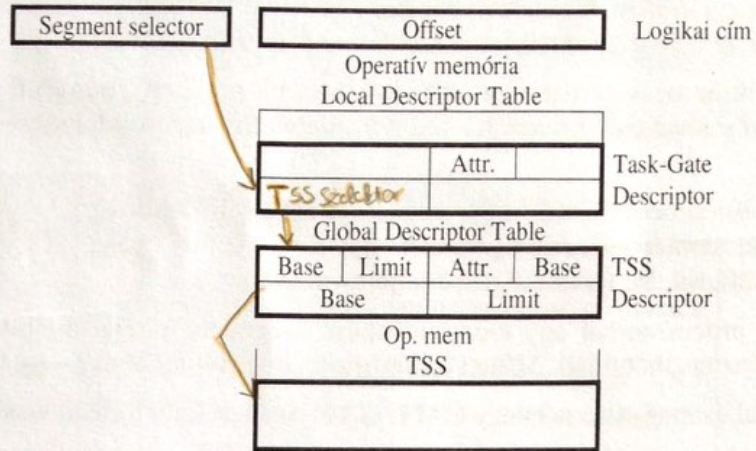
A 386-os processzornál egy memória típusú szegmens leírója tartalmazza a szegmens kezdőcímét, a hosszát és az atribútumait. **Miben különbözik** ettől (mit tartalmaz) egy **TASK GATE** leírója?

Rajzolja fel a TASK GATE működését bemutató vázlatot!



A mellékelt ábrába írja be, hogy mit tartalmaz a **TASK-Gate Descriptor** és jelölje (rajzolja fel), **hogyan történik** a Gate-n keresztül a task-váltás (TSS elérése).

*Task Gate*



**Rajzolja fel** és röviden ismertesse, a processzor tehermentesítésének blokkvázlatát társprocesszorral (8086-8087)!

**Magyarázza el**, hogyan működik időben átlapolva a processzor és a matematikai társprocesszor?

Hogyan tudja a 8086 processzor a társprocesszor műveletének az eredményét, annak felhasználása előtt, kivárni?

**Multitaskos rendszer**nél az alábbi kijelentések közül jelölje x-szel az igaz(ak)at és - jellel a hamis(ak)at!

Minden taskhoz egy külön fizikai processzor tartozik.	
A processzor taskkváltáskor hardveresen menti a task teljes állapotát.	
A i386/486 processzornál minden taskhoz külön lokális szegmensleíró tábla tartozhat.	
A virtuális - és fizikai processzorok időbeli összerendelését külön hardver egység végzi.	
Multitaskos rendszernél a taskok állapotát leíró információt taskkváltáskor menteni, illetve cserélni kell.	
Multiprocesszoros rendszereknél, statikus feladat hozzárendelés esetén, egy adott feladatot mindig ugyanaz a processzor lát el.	

Minden fizikai processzorhoz több virtuális processzor tartozhat.	
Minden fizikai processzorhoz külön task tartozik.	
A processzor taskkváltáskor részben hardveresen, részben szoftveresen menti a task teljes állapotát.	
A i386/486 processzornál minden task csak a Globális szegmensleíró táblát használhatja.	
A virtuális és fizikai processzorok időbeli összerendelését ütemező algoritmus végzi.	
A virtuális és fizikai processzorok összerendelését egy task állapot leíró felhasználásával végzik.	
Multiprocesszoros rendszereknél, dinamikus feladat hozzárendelésnél esetén egy processzor minden funkciót (feladatot) elláthat.	
A i386/486 processzornál a taskok korlátlanul újra hívhatják magukat (rekurzív, reentrant).	

Minden taskhoz egy külön fizikai processzor tartozik.	
A virtuális és fizikai processzorok időbeli összerendelését külön hardver egység végzi.	
A processzor taskkváltáskor hardveresen menti a task teljes állapotát.	
Az i386/486 processzornál a taskok nem működhetnek re-entrant módon.	
A i386/486 processzornál minden taskhoz külön lokális szegmensleíró tábla tartozhat.	
A i386/486 processzornál taskkváltáskor a cache tartalmát érvényteleníteni kell.	



Minden taszkhhoz külön laptábla könyvtár kezdőcím tartozhat.	
Ha a kiépített fizikai memória méretét megduplázzuk, a laptábla könyvtár mérete nem változik.	
Védett üzemmód bekapcsolása szegmentálás leírókon alapuló működését is aktivizálja ez külön nem kapcsolható ki.	
A kétlépcsős laptábla szervezés kétszer akkora memóriaterületet igényel, mint az egylépcsős laptábla.	
A szegmensleíró báziscíme bitjeinek száma független a lapszervezés ki- vagy bekapcsolásától.	
A lineáris és a fizikai címtartomány egyaránt 4GByte.	

Multitaskos rendszerben egy taszknak minden védelmi szinten van kód adat és veremszegmense.	
Az I/O utasításokat nem minden privilégiumszinten lehet végrehajtani.	
A tényleges privilégiumszint (EPL) értéke, az aktuális (CPL) és az operandus privilégiumszint (RPL) közül, a nagyobbik számértékével egyezik meg, azaz $EPL = \max(CPL, RPL)$ .	
A taszkok állapotának nyilvántartására szolgáló különleges szegmens (TSS) címe a CR3 vezérlőregiszterben található.	
A CALL kapu (CALL GATE) tartalmazza az elérni kívánt objektum új szelektorát, az új eltolást és a stack-en keresztül átadandó paraméterek számát.	
A megszakítási leíró tábla (IDT) is tartalmazhat taszok kapukat (TASK GATE).	

A 386-os mikroprocesszor bekapcsolt lapszervezés esetén a 32 bites lineáris címből kétlépcsős laptábla szervezéssel állítja elő a fizikai címet. A lapméret 4Kbyte. A laptábla könyvtár elejére a CR3 regiszter tartalma mutat. Egy lapleíró bejegyzés 32 bitet tartalmaz.

Hány értékes bitet kell a CR3-nak tartalmaznia? 32-12

Minimálisan mekkora op. memória kell a kétlépcsős laptábláknak? 8

Egy TASK-hoz maximálisan hány lapleíró tábla tartozhat?

1 TASK-hoz max 1024 lapleíró tábla

20
8Kb
1024

Milyen kivételes állapotot okoz az az eset, amikor a processzor "page not present" bejegyzést talál a lapleíróban?

- A abort    **B** fault    C reset    D trap    B

Mit jelent az eszköz-szintű I/O kezelés?

**Sorolja fel** és egy-egy mondatban ismertesse az eszközszintű I/O kezelés eseteit!

**Mondjon** példákat a direkt és a feltételes I/O kezelésre!

**Magyarázza el** mit jelent a perifériák szinkron működése, adjon példát ilyen működésre!

Röviden ismertesse a programozott ellenőrzésű feltételes I/O kezelés lépéseit, a módszer tulajdonságait!

Ismertesse a megszakításos I/O kezelés fázisait, a módszer tulajdonságait!

**Rajzolja fel** és röviden ismertesse, a processzort DMA vezérlővel tehermentesítő megoldás blokkvázlatát!

**Rajzolja fel** az I/O processzorra alapozott I/O, illetve periféria kezelés blokkvázlatát!

**Sorolja fel** milyen feladatokat lát el az I/O processzor!

Egy rendszerben a tároló elérési ideje 60 ns, az effektív elérési idő 140 ns. Vergődés.  
 állapotban van-e a rendszer?  $\rightarrow 2 \times 60 < 140$  igen  $2 \times \text{tároló} < \text{Teff, verődés}$



Rajzolja fel az SCSI interfész általános blokkvázlatát! Sorolja fel a jellemző tulajdonságokat és az előnyöket!

Rajzolja fel az SCSI interfészrendszer általános blokkvázlatát! Hány host és target egység alkalmazható?

Mi a különbség az SCSI interfész szinkron, illetve aszinkron működése között?

Mi a különbség a szimmetrikus, illetve az aszimmetrikus vonali adó/vevők között?

Mi a szimmetrikus vonali adó/vevők alkalmazásának előnye és hátránya az aszimmetrikussal szemben?

Az alábbi állítások közül jelölje x-szel az igazakat!

A nyolcbites SCSI interfészre max. 8db egység (host és target) kapcsolódhat, az információ cserében résztvevő egységet a host jelöli ki egy szelekciós fázisban.	
A logikai I/O kezelés esetén a közvetlen input-output műveleteket az operációs rendszer végzi. A felhasználó csak op.-rendszer függvény hívásokon keresztül érheti el ezeket.	
Az eszközsintű I/O kezelés esetén kell egy eszköz, ami a KI/Be vonalakat az operációs rendszerhez kapcsolja	
Aszinkron működésű periféria vezérlésére mindig UART-ot kell használnunk.	
Az aszinkron vezérlésű periféria működési idejét az I/O egység működési sebessége is befolyásolhatja	
Az I/O processzor az IN-és OUT utasításokat hajtja végre a CPU helyett.	

Rajzolja fel és magyarázza el a mágneses háttértároló Író/olvasó fejének működési elvét!

Mi korlátozza az adatrögzítés frekvenciáját?

Magyarázza el mit jelent a csúcseltolódás, és hogyan küszöbölik ki?

Mi a magneto-rezisztív olvasó fej működésének alapja?

Rajzolja fel és röviden magyarázza el az nagymértékű magneto-rezisztív (GMR) hatás elvét!

Magyarázza el a spinszelep elvű olvasófej működését.

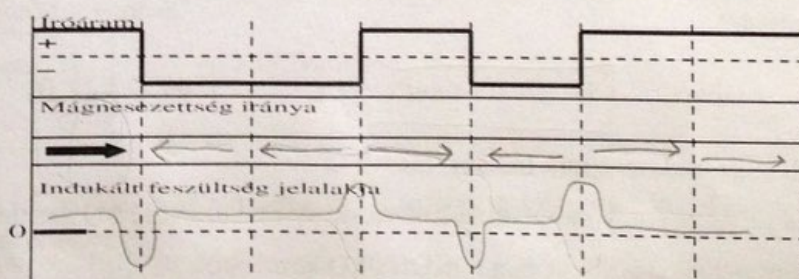
Röviden sorolja fel magneto-rezisztív olvasó fej előnyeit!

Miben különbözik a horizontális, illetve a vertikális (perpendicular) mágneses adatrögzítés?

Miben különbözik a fenti, kétféle, adatrögzítésnél a lemez felépítése?

**Mágneses** háttértároló íróáram jelalakja látható az alábbi ábrán.

Adja meg a mágneses réteg **mágnesezettségének irányát** és a kiolvasáskor létrejövő **indukált feszültség** jelalakját.



Miért kell kódolni a rögzítendő információt, mit jelent tulajdonképpen a kódolás?

Rajzolja fel NRZ, NRZI, PE, FM, MFM, RLL0,2, és RLL2,7 kódolás esetén a mágnesezettség vagy az íróáram jelalakját a következő bitsorozatra: 01000111011 (az RLL kódokhoz használja a megadott táblázatot)

Magyarázza el, miért lehet, azonos technológiát feltételezve, RLL2,7 kódolás alkalmazásakor 50%-al több információt rögzíteni, az MFM kódoláshoz képest!



Röviden magyarázza el a szoft-szektor szervezés elvét és az ehhez szükséges információs mezők felépítését szerepét!

Mi a mark (jelző) szerepe, hogyan valósítják meg?

Mi a GAP szerepe?

Magyarázza el, hogyan szinkronozódik fel íráskor, illetve olvasáskor a mágneslemezes író olvasó elektronika a formázáskor felvitt információra

Egy floppy diszk egy sávja 8 szektort tartalmaz.

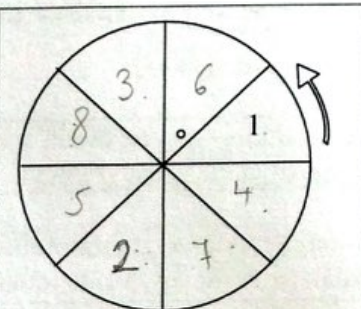
Rajzolja fel a szektorok fizikai elhelyezkedését a megadott forgásiránynál **3:1** szektor **interleave** esetén!

Milyen probléma kiküszöbölésére használható a megoldás? *Ha a feltételezésből lassú*

Optimális esetben hány körfordulás kell egy sáv összes szektorjának írásához vagy olvasásához **3:1** szektor-interleave esetén?

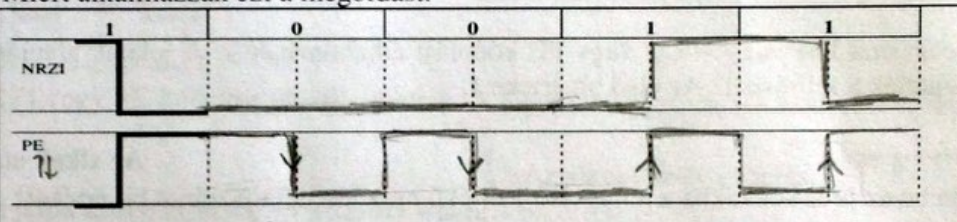
*3*

Feltételezve, hogy a fenti megoldás indokolt volt, mi történne **1:1** interleave létrehozása után? *8 körfordulás kellene.*



Egy floppy diszk egy sávja 9 szektort tartalmaz. Rajzolja fel a szektorok fizikai elhelyezkedését **2:1** szektor interleave esetén! Miért alkalmazzák ezt a megoldást.

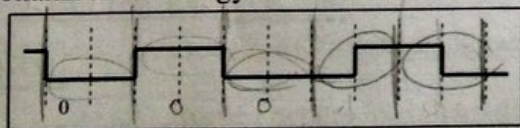
Rajzolja be a mellékelt ábrába egy mágneses tároló íróáram jelalakját a megadott bitsorozatra NRZI és PE kódolás esetén



A szokásos formátumú floppy-lemez egy szektora két fő részből áll, az azonosító mezőből (identification field) és az adatmezőből (data field). Az alábbi állítások közül jelölje x-szel az igazakat!

A szektornak csak az adatmezőjében van hibaellenőrző kód (CRC).	-
Adat íráskor a szektorazonosító mezőjét is újraírják.	
Nem csak a szektorok közt, hanem a szektoron belül is van egy "gap".	X
Az adatmező hosszára vonatkozó információ az adatmező elején található.	X
A sáv és a szektor sorszámát.	
Az azonosító mező tartalmazza a drive fizikai címét.	

Egy mágneses elvű adattárolóban a sáv mágnesezettsége (**B**) az ábrán látható módon változik. A szaggatott vonalak az ábrán egyforma időközönként vannak.

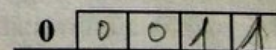


*0-nál szélen, 1-nél közepén.*

A felírásnál FM vagy MFM vagy PE kódolást alkalmazhattak. A jelalak alapján melyik kódolási eljárást használták a felírásnál, ha az első bit értéke 0?

Az alkalmazott kódolás: **MFM**

Írja be az üres kockákba a jelölt, első 0 értékű, bitet követő további 4 bit értékét!



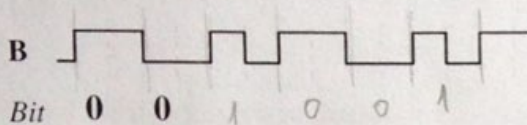
Azonos fluxusváltási sűrűséget feltételezve MFM kódolással kétszer annyi adat tárolható, mint FM kódolással

**Igen** **NEM**

*MFM = 2FM*



Egy mágneses elvű adattárolóban (pl. diszk) a sáv mágnesezettsége (B) az ábrán látható módon változik. A szaggatott vonalak az ábrán egyforma időközönként vannak. Az első 2 bit értéke 0.



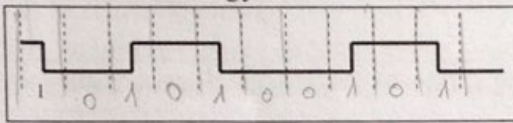
a/ Melyik kódolási eljárást használták a felírásnál  A FM  B MFM  C NRZI  FM

b/ Írja be az üres kockákba a jelölt első két 0 értékű bitet követő további 4 bit értékét!  
 0 0

c/

Szoft-szektor szervezésnél speciális jelzések előállítására előre definiált módon  Igen  NEM megsértik a kódolási szabályt.

Egy mágneses elvű adattárolóban a sáv mágnesezettsége (B) az ábrán látható módon változik. A szaggatott vonalak az ábrán egyforma időközönként vannak.



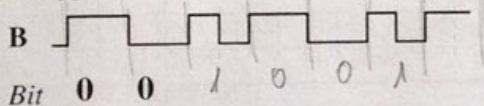
A felírásnál FM vagy NRZI vagy PE kódolást alkalmazhattak. A jelalak alapján melyik kódolási eljárást használták a felírásnál? Az első bit értéke 1.

Az alkalmazott kódolás:  NRZI  PE

Írja be az üres kockákba a jelölt első 1 értékű bitet követő további 4 bit értékét!

A fenti három kódolási eljárás közül melyiknél a legnagyobb az egységnyi felületre eső fluxusváltozás?  FM  PE  NRZI

Egy mágneses elvű adattárolóban (pl. diszk) a sáv mágnesezettsége (B) az ábrán látható módon változik. A szaggatott vonalak az ábrán egyforma időközönként vannak. Az első 2 bit értéke 0.

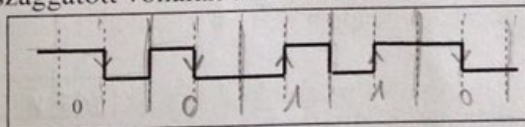


Melyik kódolási eljárást használták a felírásnál  A FM  B MFM  C PE  FM

Írja be az üres kockákba a jelölt első két 0 értékű bitet követő további 4 bit értékét!  
 0 0

A fenti három kódolási eljárás közül melyikkel érhető el a legnagyobb adatbit sűrűség?  FM  MFM  PE

Egy mágneses elvű adattárolóban a sáv mágnesezettsége (B) az ábrán látható módon változik. A szaggatott vonalak az ábrán fél bit időközönként vannak.



A felírásnál FM vagy NRZI vagy PE kódolást alkalmazhattak. A jelalak alapján melyik kódolási eljárást használták a felírásnál? Az első bit értéke 0.

Az alkalmazott kódolás:  FM  PE  NRZI

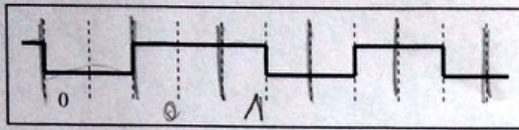
Írja be az üres kockákba a jelölt első 0 értékű bitet követő további 4 bit értékét!

c/ Adja meg, hogy a fent említett kódolási módok közül melyiknél legkisebb a bitsűrűség!

FM  PE  NRZI



Egy mágneses elvű adattárolóban a sáv mágnesezettsége (B) az ábrán látható módon változik. A szaggatott vonalak az ábrán fél bit időközönként vannak. Az első bit értéke 0.



A felírásnál FM vagy MFM vagy PE kódolást alkalmazhattak. A jelalak alapján melyik kódolási eljárást használták a felírásnál?

MFM

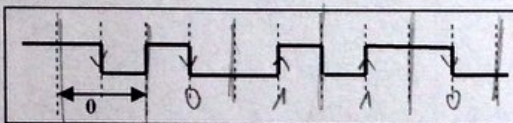
A felírásnál FM vagy MFM vagy PE kódolást alkalmazhattak. A jelalak alapján melyik kódolási eljárást használták a felírásnál?

MFM

Adja meg, hogy a fent említett kódolási módok közül melyikkel érhető el a legnagyobb adatbitsűrűség!

MFM

Egy mágneses elvű adattárolóban a sáv mágnesezettsége (B) az ábrán látható módon változik. A szaggatott vonalak az ábrán fél bit időközönként vannak. Az első bit értéke 0.



A felírásnál FM vagy NRZI vagy PE kódolást alkalmazhattak. A jelalak alapján melyik kódolási eljárást használták a felírásnál?

Az alkalmazott kódolás:

PE

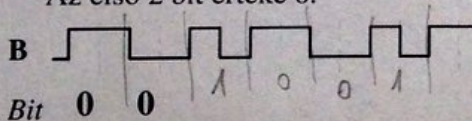
Írja be az üres kockákba a jelölt első 0 értékű bitet követő további 4 bit értékét!

0 0 1 1 0

Adja meg, hogy a fent említett kódolási módok közül melyiknél legkisebb a bitsűrűség!

PE

Egy mágneses elvű adattárolóban (pl. diszk) a sáv mágnesezettsége (B) az ábrán látható módon változik. A szaggatott vonalak az ábrán egyforma időközönként vannak. Az első 2 bit értéke 0.



Melyik kódolási eljárást használták a felírásnál A FM B MFM C PE?

FM

Írja be az üres kockákba a jelölt első két 0 értékű bitet követő további 4 bit értékét!

0 0 1 0 0 1

A fenti három kódolási eljárás közül melyikkel érhető el a legnagyobb adatbit sűrűség?

MFM

Mi az alapvető különbség egy drive szintű (pl.:ST410) illetve egy IDE, vagy SCSI vezérlő között?



**Ismertesse** mit jelent, hogy két esemény konkurens! *Ha egyik nem tudja időzíteni befolyásolni a másikat*

**Ismertesse** mit jelent többprocesszoros rendszereknél a statikus, illetve a dinamikus feladat-hozzárendelés, sorolja fel tulajdonságaikat, előnyüket és hátrányukat!

**Rajzolja** fel egy lazán csatolt rendszernél a pont-pont közötti adatátvitel hardver jelzőbitre alapozott megoldásának blokkvázlatát! Mi a megoldás előnye, hátránya? Milyen probléma kiküszöbölésére használnak a továbbfejlesztett változatban FIFO elven működő átmeneti tárolót?

**Rajzolja fel** egy szorosan csatolt rendszer vázlatát!

**Magyarázza el** a postafiók szemaforok használatának elvét a szorosan csatolt rendszereknél?

**Multiprocesszoros** szorosan csatolt rendszereknél **milyen hardver** támogatás szükséges a szemaforkezelésnél szükséges **kölcsönös kizárás** biztosításához? **Írjon** példákat a megoldásra!

**Sorolja** fel a rendszersín funkcióit!

**Ismertesse** a rendszervezérlő, a master (mester) és a slave (szolga) szerepét!

**Ismertesse** mit értünk erőforrás-particionált rendszeren? **Sorolja fel** az alapvető tulajdonságait!

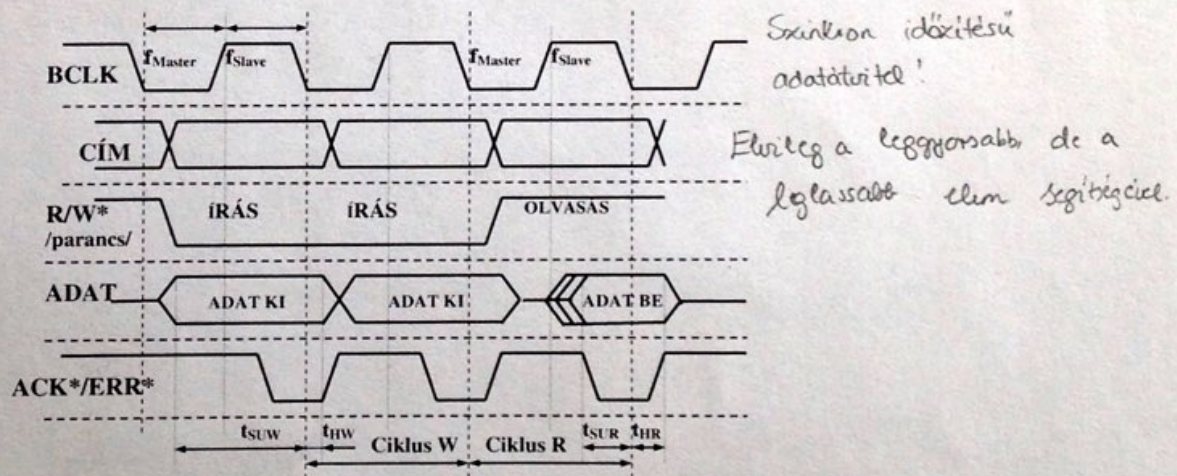
**Ismertesse** mit értünk feladat-particionált rendszeren? **Sorolja fel** az alapvető tulajdonságait!

Sínrendszereknél mit jelent a geografikus címzés? Mondjon példát az alkalmazásra!

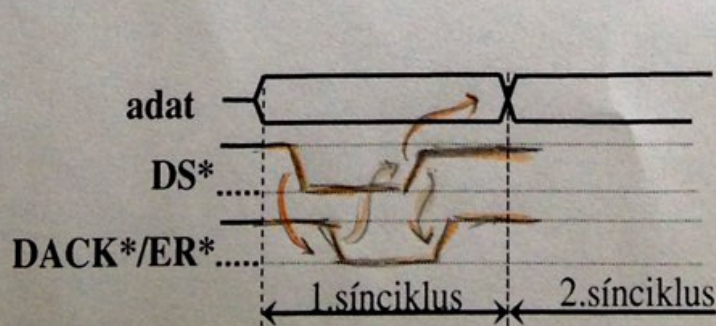
Ismertesse, hogyan osztja ki a geografikus címet a MULTIBUS II rendszervezérlő!

**Ismertesse** egy üzenetkapcsolt rendszer működésének elvét (fő lépéseit) pl.: háttértárolóról történő adatolvasás esetén!

**Milyen sínidőzítési protokoll** látható az alábbi ábrán? Mik az ilyen protokoll előnyei és hátrányai?

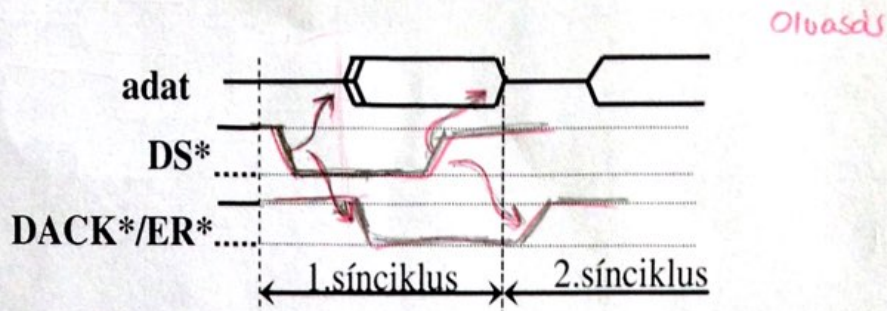


**Rajzolja** be az alábbi ábrába egy teljesen reteszelt protokollt alkalmazó sínrendszer adott jeleit és a jelátmenetek közötti, ok-okozati viszonyt mutató nyilakat **írás** esetén!

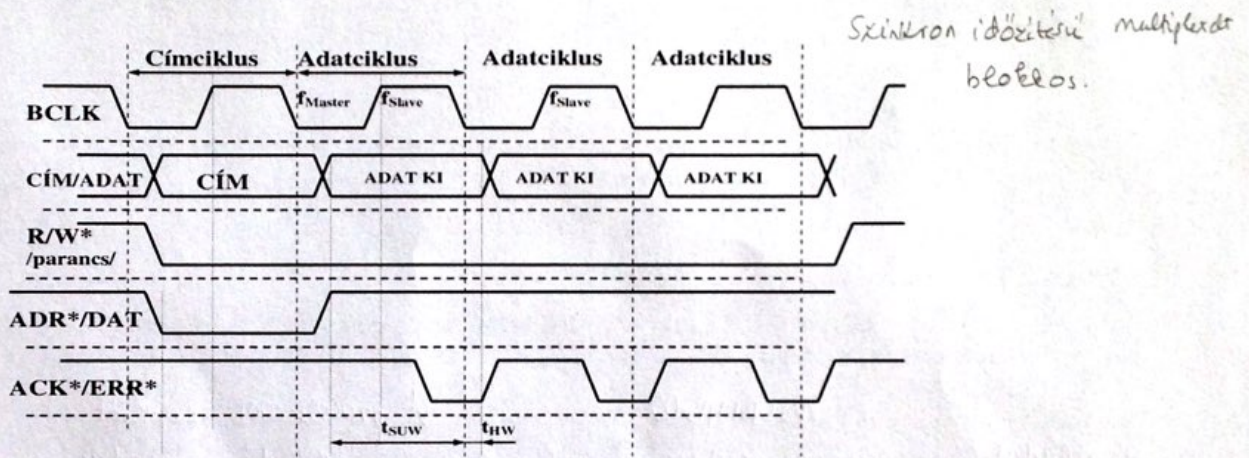




Rajzolja be az alábbi ábrába egy teljesen reteszelt protokollt alkalmazó sínrendszer adott jeleit és a jelátmenetek közötti, ok-okozati viszonyt mutató nyilakat olvasás esetén!



Milyen sínidőztítési protokoll látható az alábbi ábrán? Mik az ilyen protokoll előnyei és hátrányai?



Rajzoljon fel egy teljesen reteszelt (kapcsolt) szemiszinchron adatátviteli protokollt!

Milyen értékűek lehetnek egy ilyen protokoll időzítései? *Minimális ciklusidő 4 órajel, csak az órajel egész részére lehet az időzítés lehet az órajel többszöröse lehet az órajel*  
 Rajzolja be a hiányzó jeleket és a jelátmenetek közötti, ok-okozati viszonyt mutató nyilakat írás esetén!

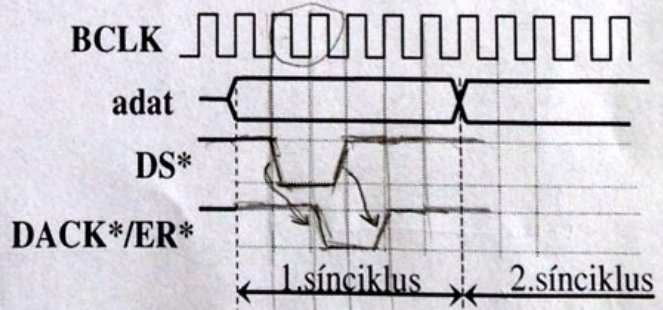
Teljesen kapcsolt protokollt alkalmazó szemiszinchron sínrendszer jelei láthatók az alábbi ábrán (az órajel 10MHz).

Mekkora lehetne egy írási sínciklus minimális ideje, egy ilyen rendszerben?

$TWC_{min} = 300 \text{ ns}$

*minimális idő ≠ min ciklus idő*

Milyen hiba lép fel ilyen protokollnál nem létező címre íráskor... *lelaga...*



*A rendszer lefagy, mert várakozik, és nem történik semmi.*

*Idő-tulajdonság ellenőrizni kell!*

*csúszási idő  
1 órajel  
perciklus*



Hány **arbitrációs modul** (arbiter) és hány **mester modul** lehet egy VME rendszerben? **Indokolja a választ!**

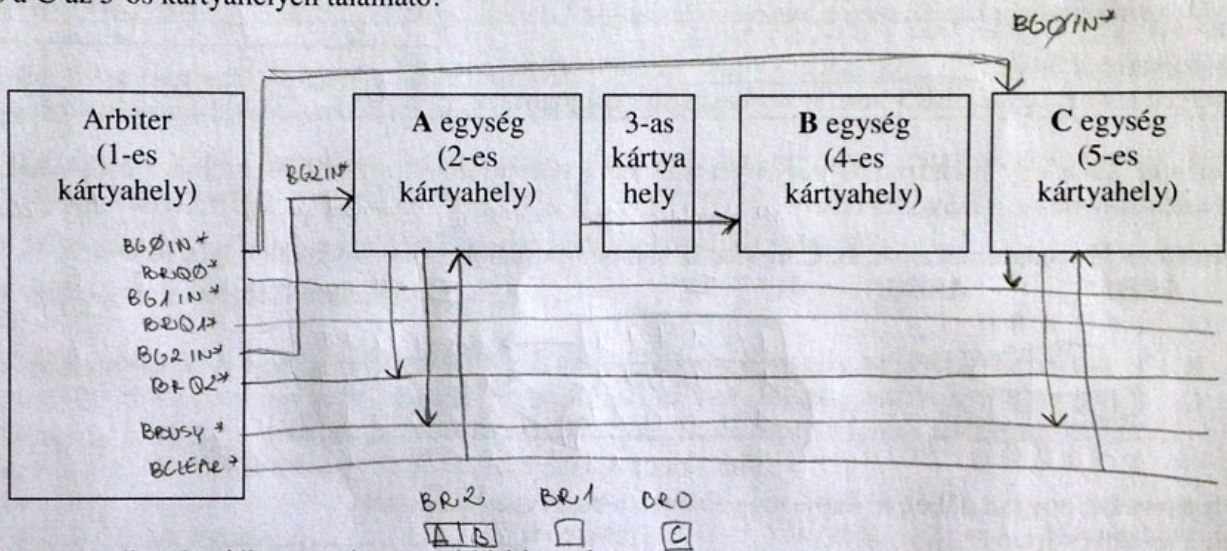
**Arbiter:** 1 arbiter ..... **Indoklás:** A 7. elő kártyahelyen egyetlen arbitrációs modulon dönti el hogy ki beszélhet a sinton.

**Mester:** kártyahely - 1 ..... **Indoklás:** Bámmeddig lehet követni, minden stratégia esetén, csak bizonyos sávon felül problémája lehet.

**Milyen busz megszerzési stratégiák** valósíthatók meg **VME** rendszerben? (Indokolja a választ) *Prioritásos, Egyszerűsített, Rajzolja fel, a decentralizál, párhuzamos versenyztetés elvű hardver mechanizmust használó arbiter blokkvázlatát!*

VME rendszerben az **A** egység a **BR2\***-ön, a **B** a **BR2\***-ön, a **C** egység a **BR0\***-án kér buszvezérlési jogot. Az arbiter a BR3-BR0 szintek között fix prioritásos elven működik. Prioritás, csökkenő prioritási sorrendben, BR3-BR2-BR1-BR0

Rajzolja be az arbitrációs rendszerhez tartozó fontos jeleket az alábbi ábrán, ha az **A** a 2-es, a **B** a 4-es, és a **C** az 5-ös kártyahelyen található!



Milyen sorrendben kapják meg a buszvezérlési jogot?

1: A	2: B	3: C
------	------	------

VME rendszerben az **A** egység a **BR0\***-án, a **B** a **BR2\***-ön, a **C** egység a **BR2\***-ön kér **egyszerre** buszvezérlési jogot. Az arbiter a BR3-BR0 szintek között **Round-Robin** elven működik. Az **A** master **ROR** (elengedés kérésre) a **B**- és **C** mester **RWD** (elengedés, ha kész) buszelengedési stratégiát használ.

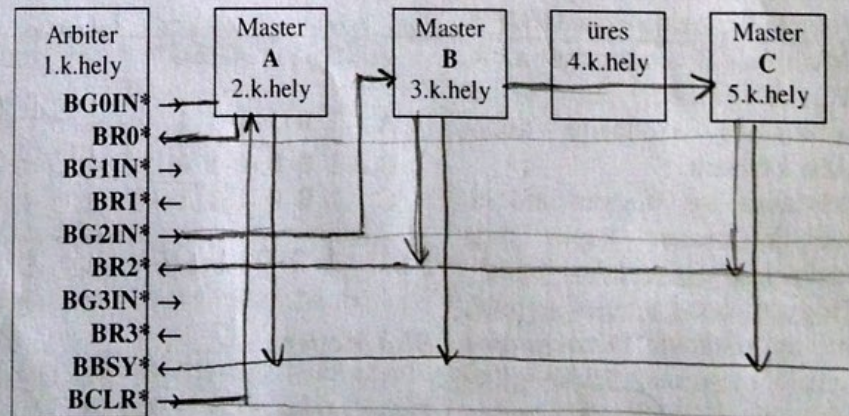
**Milyen sorrendben** kapják meg a buszvezérlési jogot ha a pillanatnyi prioritás, csökkenő prioritási sorrendben, BR2-BR1-BR0-BR3?

1	2	3
B	A	C

Mi a sorrend, ha a pillanatnyi prioritásnál BR0 a legmagasabb?

1	2	3
A	B	C

**Rajzolja be** az arbitrációs rendszerhez tartozó jeleket, a fentieknek megfelelően, az ábrába!



Ha átkérzi van mester mindig

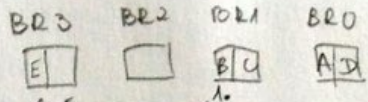
ROR: elenged, ha kap feltételt hogy elengedre (clearbe kell)

RWD: Elengedi ha kész -> NEM KÖRZÜK CLEARBE

BR2	BR1	BR0	BR3
B	C	A	E

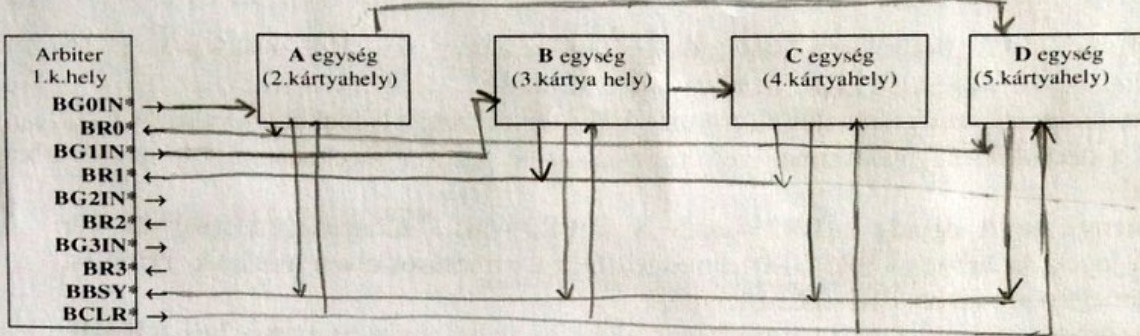
Round Robin





VME rendszerben az **A és D** egység a **BR0\***-án, az **B és C** egység a **BR1\***-en kér buszvezérlési jogot. Az arbiter a BR3-BR0 szintek között **Round-Robin** elven működik. Pillanatnyi prioritás, csökkenő prioritási sorrendben, BR3-BR2-BR1-BR0.

A és C egység valamennyi, míg a B és D csak a **RWD** és a **Preemption** buszelengedési stratégiát képes megvalósítani.



Milyen sorrendben kapják meg a buszvezérlési jogot?

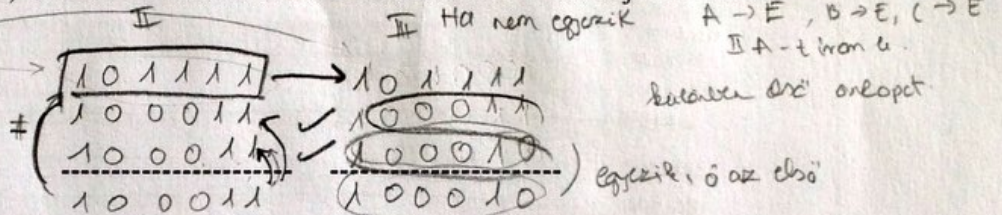
1: B	2: A	3: C	4: D	
1: B	2: E	3: A	4: C	5: D

Mi a sorrend ha az **első egység** adatátvitel alatt a 6. kártyahelyen lévő **E** egység **BR3\*-on** is kér buszvezérlési jogot?

**Rajzolja fel** a egy párhuzamos versenyztetés elvű decentralizált arbitrációs rendszer blokkvázlatát, röviden ismertesse a működés elvét! *(1) Amíg egyik, addig ugyanazt írja, utána 1-esel.*

**Multibus II** rendszerben az **A, B, C** mester az alábbi arbitrációs azonosító kódot adja a buszra:

	ARB5*	ARB0*
A	1 0 1 0 0 0	
B	1 0 0 0 1 1	
C	1 0 0 0 1 0	
	1 0 0 0 0 0	



**Mutassa be**, hogyan dől el, ki kapja meg elsőnek a buszvezérlési jogot!

Mi a további sorrend?

1: C	2: B	3: A
------	------	------

Hogyan alakul a fenti sorrend, ha az elsőként kiszolgálásra kerülő master-nél **újabb** buszvezérlési igény lép fel a **második** egység kiszolgálása alatt?

*schop*

1: C	2: B	3: A
------	------	------

Milyen buszmegszerzési stratégiát valósít meg a **MBII** a fenti megoldással?

**Buszmegszerzési** stratégia: *gazdaságos*

Hogyan biztosítja ezt a **MBII**? *Wem lehet párhuzamosan beleszól a döntés felvételébe (elvére ha 0-val kezdődik)*

**Multibus II** rendszerben az **A, B, C** mester az alábbi arbitrációs azonosító kódot adja a buszra:  
**Mutassa be**, hogyan dől el, melyik mester kapja meg elsőnek a buszvezérlési jogot.  
 Hogyan alakul a teljes sorrend, ha az **elsőként** kiszolgálásra kerülő mesternél **újabb** igény lép fel a **második** egység **kiszolgálása** alatt? Milyen buszmegszerzési stratégiát valósít meg a fenti mechanizmus?

	ARB5*	ARB0*
A	1 0 1 1 1 1	
B	1 0 0 1 1 0	
C	1 0 0 1 1 1	
	1 0 0 1 1 0	

**Első Mester:** B

**Teljes sorrend:** 1: B 2: C 3: A 4: B

**Buszmegszerzési stratégia:** *gazdaságos (egyenlő esélyű)*

A	1 0 1 1 1 1
C	1 0 0 1 1 1
	1 0 0 1 1 1



<p><b>Multibus II</b> rendszerénél az <b>A, B, C</b> mester az alábbi arbitrációs azonosító kódot adja a buszra:  <b>Mutassa be</b>, hogyan dől el, ki kapja meg elsőnek a buszvezérlési jogot</p>	<p><b>ARB5* ARB0*</b></p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; text-align: center;">A</td> <td style="width: 33%; text-align: center;">1 0 0 1 1 1</td> <td style="width: 33%; text-align: center;">1 0 0 1 1 1</td> </tr> <tr> <td style="text-align: center;">B</td> <td style="text-align: center;">1 0 1 1 1 1</td> <td style="text-align: center;">1 0 1 1 1 1</td> </tr> <tr> <td style="text-align: center;">C</td> <td style="text-align: center;">1 0 0 1 1 0</td> <td style="text-align: center;">1 0 0 1 1 0</td> </tr> <tr> <td colspan="2" style="border-top: 1px dashed black;"></td> <td style="border-top: 1px dashed black;"></td> </tr> <tr> <td></td> <td style="text-align: center;">1 0 0 1 1 0</td> <td style="text-align: center;">1 0 0 1 1 0</td> </tr> </table> <p style="text-align: right; margin-right: 20px;"><b>CAB</b></p>	A	1 0 0 1 1 1	1 0 0 1 1 1	B	1 0 1 1 1 1	1 0 1 1 1 1	C	1 0 0 1 1 0	1 0 0 1 1 0					1 0 0 1 1 0	1 0 0 1 1 0
A	1 0 0 1 1 1	1 0 0 1 1 1														
B	1 0 1 1 1 1	1 0 1 1 1 1														
C	1 0 0 1 1 0	1 0 0 1 1 0														
	1 0 0 1 1 0	1 0 0 1 1 0														
<p>Az <b>ABC</b> bejelentkezése után három órajellel később a mellékelt arbitrációs kódú <b>D</b> és <b>E</b> egységénél is buszvezérlési igény keletkezik. Adja meg hogy az <b>ABCDE</b> egységek milyen sorrendben kapnak buszvezérlési jogot!</p>	<p><b>D 1 0 0 0 0 1</b> (1-enel kezdődik, biztos utolsó)  <b>E 0 0 1 1 1 1</b></p> <p>1: ... 2: <b>E</b> 3: ... 4: ... 5: ...</p> <p>A 1 0 0 1 1 1          B 1 0 1 1 1 1          E 0 0 1 1 1 1          -----          1 0 0 1 1 1</p> <p style="font-size: small;">A 1 1 1 1 1 1          B 1 1 1 1 1 1          E 0 0 1 1 1 1          -----          0 0 1 1 1 1</p>															

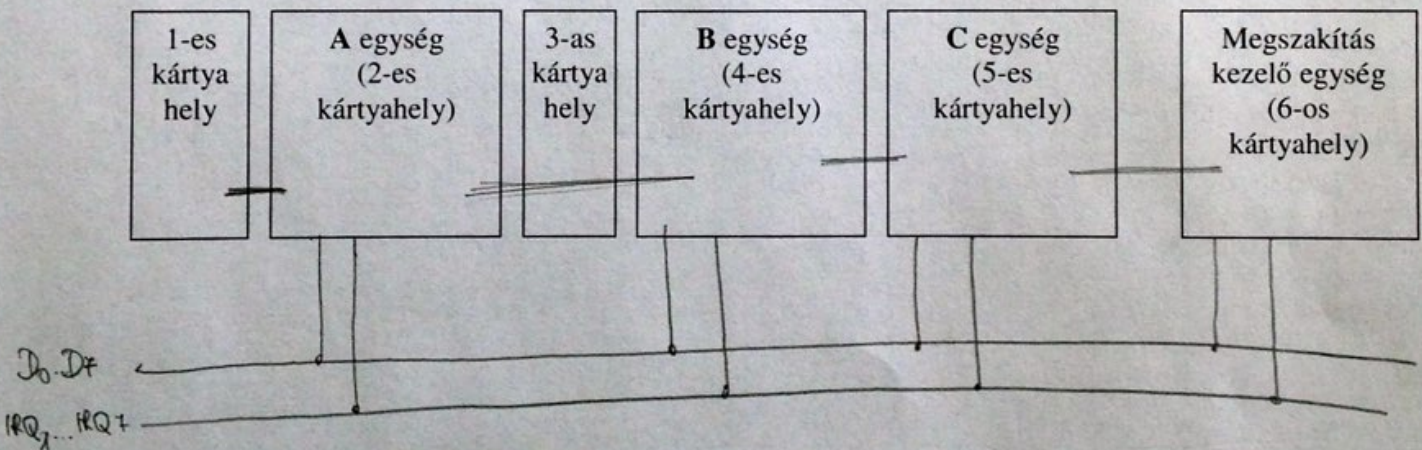
**Rajzolja fel** a láncolós (daisy chain) bővítésű busz-vektoros megszakítási rendszer blokkvázlatát, magyarázza el a működését, sorolja fel az előnyeit és hátrányait!

**Rajzolja fel** a VME láncolós (daisy chain) bővítésű busz-vektoros megszakítási rendszerének blokkvázlatát! Magyarázza el a működését, sorolja fel az előnyeit és hátrányait!

Hány **Megszakítás kezelő** lehet egy **VME** rendszerben (indokolja a választ)?

Mit értünk virtuális megszakítás-kezelő rendszeren?

**VME** rendszerénél az **A** egység az **IRQ7\***-en, a **B** az **IRQ2\***-n, a **C** egység az **IRQ2\***-en kér megszakítást. A megszakítás kezelő egység **fix prioritású**, az 1-es a legmagasabb, 7-es a legalacsonyabb szint. Rajzolja be a megszakításkérő egységekhez tartozó fontos jeleket az alábbi ábrán, ha az **A** a 2-es, a **B** a 4-es, a **C** a 5-ös kártyahelyen található!



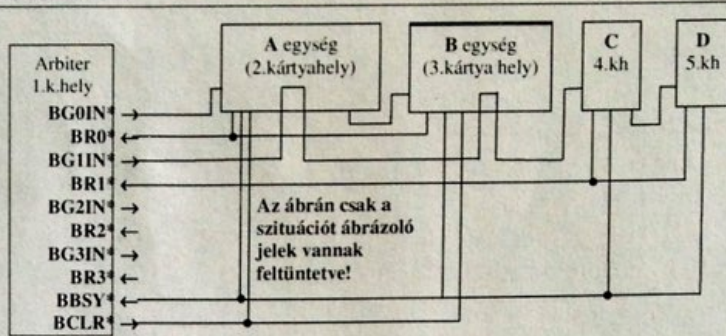
Milyen sorrendben adhatják fel a megszakítási vektoraikat, ha az éppen elfogadott szint a 2-es? 1: C    2: B    3: A

Mi a sorrend, ha az **IRQ7\***-es a legmagasabb prioritású szint és az elfogadott szint a 2-es? 1: A    2: B    3: C



VME rendszernél a **A** (2.kh), **B** (3kh) **C** (4kh) és **D** (5kh) egység mester. Az **A** és a **B** egység a **BR0\*** jelen, a **C, D** a **BR1\*** kapcsolódik az arbiterhez.

Valamennyi egységnél sínvezérlési igény keletkezik. Az arbiter ROUND-ROBIN elven működik. A pillanatnyi prioritás **csökkenő** sorrendben **BR3\*, BR2\*, BR1\*, BR0\***



a) Adja meg milyen sorrendben kerülnek kiszolgálásra.

I: C    2: A    3: D    4: B

b) Mi a sorrend, ha a második egység kiszolgálása alatt az első mesternél újabb igény keletkezik?  
 I: C    2: A    3: C    4: B    5: D

c) Milyen buszmegszerzési stratégiát valósít meg a fenti rendszer?  
 Kombinált. A szűkebb kábel Round-Robin. Indoklás: Hasonlóan belső prioritásos.

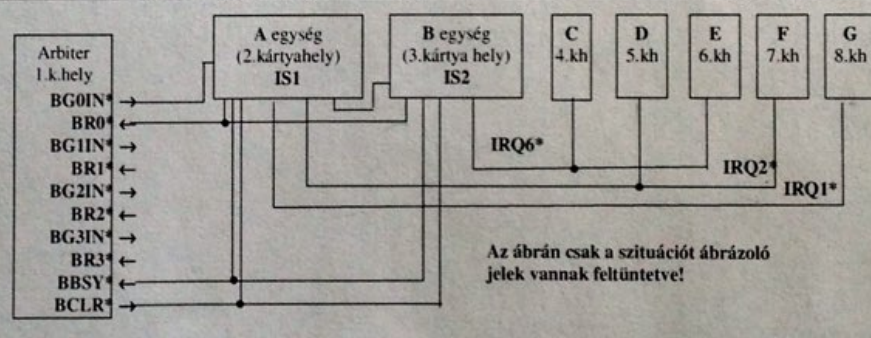
d) Hogyan oldható meg, hogy az A és B egység egyenlő esélyű buszmegszerzési stratégia szerint kerüljön kiszolgálásra?  
 A és B helyen buszvezérlési jogot. Pl. BR2-n.

e) Milyen buszelengedési stratégiát kell megvalósítania a C és a D jelű mesternek?  
 Round, ha végzetek töltés magukat, a többi kértő analóg nem figyelik.

152 13A

VME rendszernél a 2.kártyahelyen (2.kh) **A** egység egy **IS1** jelű a 3kh-en **B** egység egy **IS2** jelű megszakításkezelőt tartalmazó mester. Az **A** és a **B** egység a **BR0\*** jelen kapcsolódik az arbiterhez. **IS1** fogadja az **IRQ1\*, IRQ2\*, IRQ3\*** jeleket, míg **IS2** fogadja az összes többi megszakításkérő jelet. **IS1** és **IS2** fix prioritással működik a magasabb számú vonalon érkező kérés magasabb prioritást jelent a hozzá tartozó kiszolgálónál.

**C** egység (4.kh) és **E** egység (6.kh) az **IRQ6\*** vezetékre a **D** (5.kh) és **F** (7kh) egység az **IRQ2\*** vezetékre **G** egység (8.kh) **IRQ1\*** vezetékre kapcsolódik. **C, D, E, F, G** egységeknél egyszerre keletkezik megszakításkérési esemény.



a) Adja meg milyen sorrendben kerülnek kiszolgálásra.  
 I: D    2: F    3: G    4: C    5: E

Indoklás:  
 A prioritása nagyobb mint B-é... így előbb A megszakítását... juttat... elvégzem...

b) Mi a sorrend, ha csak egy IS kezelő van a rendszerben valamennyi IRQ vonal számára?  
 I: C    2: E    3: D    4: F    5: G

c) Hogyan oldható meg, hogy az A és B egység egyenlő esélyű buszmegszerzési stratégia szerint kerüljön kiszolgálásra?  
 Használjanak külön BR bemenetet, és az egyikhez tartozó BG bemeneteket

(ne Daisy Chain)

**Magyarázza** el mit értünk virtuális megszakítási rendszeren! Mondjon egy alkalmazási példát!  
 Hány darab megszakítás-kérő és hány megszakítás-kezelő lehet a **Multibus II** rendszerben? **Indokolja** a választ!

8 bites címzési csúsz, tehát 256-256 db kérő és vezérlő lehet!