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	Következő utasítás címe

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

Utasítástípus: Vezérlésátadó utasítás

Regiszter PC - programszámláló regiszter ***Feladata:*** Soron következő 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!

Ismertese 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=*** báziscím + indexregiszter * 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***
sok, bonyolult	***utasítás***	kevés, egyszerű
sok, összetett	***Címzési mód***	kevés, egyszerű
kevés	***Regiszter***	sok
regiszter / memória	***Operandus helye***	regiszter
változó, hosszabb	***Műveleti idő***	rövid, egységes

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

Címzési mód: bázisrelatív

Előnye: Bázisregiszter beállítása után relatívan lehet címezni a paramétereket és lokális változókat, nem kell abszolút címeket ismerni.

Egy ***Pascal*** programban adott a következő ***függvény***:

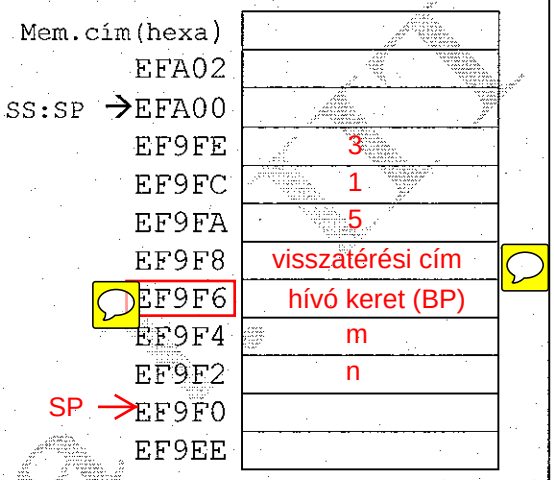
```
function f(i,j,k:integer):integer;
  var m,n,:integer;
begin …….
end;
```

8086-os processzornál az ***f(3,1,5)*** függvényhívás után ***rajzolja*** fel (írja be a memória rekeszek 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 címzési módot használnak a ***bemenő paraméterek*** és a ***lokális változók elérésére***, mi a módszer előnye?

Címzési mód: bázisrelatív

Előnye: Beállítás után relatívan címezhetőek a változók és paraméterek.



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 ***j***=0001H

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

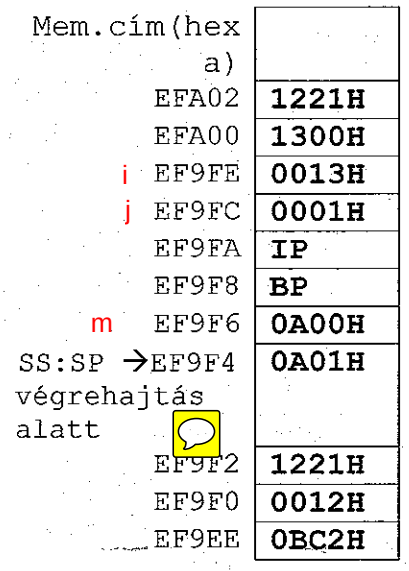
Keret: EF9F8H

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

Miért? Nem kell minden paraméter meghívásnál, hívó tudja, mennyit kell felszabadítani.

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

Miért? Alapértelmezett paraméterek miatt, hogy lehessen tudni, mikor jön az IP és BP.



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=1221……H

Mi az ***m*** lokális változó pillanatnyi értéke ***m***=2A00……H

Mi a stackpointer értéke közvetlenül a visszatérés uttán?

SS:SP=:EFA04……H

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

Címzési mód: bázisrelatív……

Előnye:Beállítás után relatívan címezhetőek a változók és paraméterek.……

Mem.cím(hexa)		
	EFA04	
k	EFA02	1221H
l	EFA00	1500H
	EF9FE	CS
	EF9FC	IP
	EF9FA	BX
	EF9F8	BP
m	EF9F6	2A00H
SS:SP végrehajtás alatt	→EF9F4	0A01H
	EF9F2	1221H
	EF9F0	0012H
	EF9EE	0BC2H

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 cím-számításnak a szempontjából?

Rajzolja fel egy háromsínes 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 kibocsájtás/ között?

Rajzolja fel egy ASAP ütemezésű szalag elvi blokkvázlatát!

Milyen utasításrendszernél 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 kiegyensúlyozottlanságát megszüntetni?

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

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. Tehát mindegyik 50ns sebességű, mert a leglassabbhoz igazodunk.

Hány ns alatt hajtódna végre egy utasítás pipe line nélkül? Tut=50+30+50=130 ns

Hány ns alatt hajtódik végre három utasítás a pipe-line működésekor? T=3*50+(3-1)50=250 ns

Mekkora az újraindítási idő a fenti esetben ? Tui=50 ns

Mekkora a szalag sebességnövelő hatása a fenti esetben ? Su=130/50 = 2.6

Mekkora az utasításkibocsájtás a fenti esetben /mértékegység is/? TP=1 utasítás / 50ns

Mekkora a szalag hatékonysága? H=2.6/3 = 87%

Melyik ütemezésű feldolgozószalagnál valószínűbb a lappangási idő nagyobb növekedése?

Milyen módszerrel csökkenthető az elemi műveletvégző egységek időben egyenlőtlen terheléséből adódó probléma?

Milyen jellegű túlterhelést képes kezelni a műveletvégzők közé iktatott várósor 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)

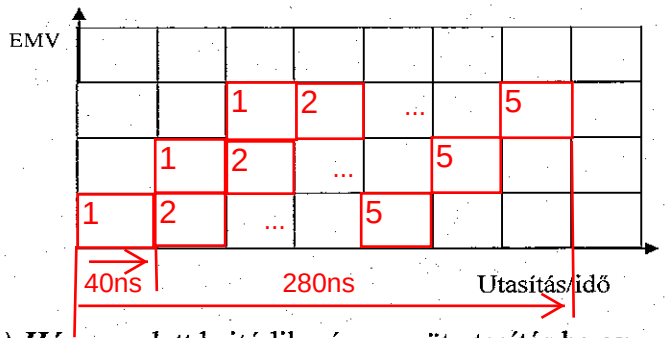
1EMV: beolvasás

2EMV: dekódolás

3EMV: végrehajtás

Hány ns egy utasítás végrehajtási ideje?

120 ns



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

280 ns

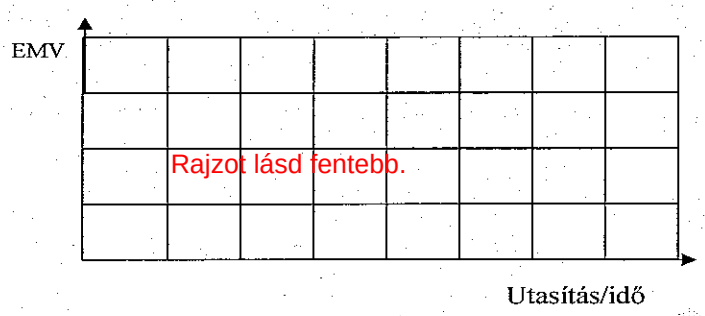
a) Rajzolja be a mellékelt ábrába ***négy utasítás*** szinkron ütemezésű ***pipe-line*** elven történő megvalósítását, ha feltételezzük, hogy minden utasítást ***három elemi műveletvégző*** dolgoz fel.

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

1EMV: beolvasás

2EMV: dekódolás

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 ***áttöltéshez illetve kiadásához szükséges idő 0,5ns***?

Egy utasítás: 3*5=15 ns

Négy utasítás: 6*5=30 ns

Szinkron, tehát mindegyik a leglasabb sebességével jár, mely jelen esetben 4.5 + 0.5 = 5ns

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 ***áttöltési időt is***. A műveletvégzők működési idői 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 idői a következők: ***UB=5ns, UD=4ns, EC=5ns MEM=10ns VI=4ns***

A műveletvégzők működési ideje már tartalmazza az ***áttöltési időt is***.

Mekkora a lappangási idő? T_L=…állandó: 50… ns. ***Mekkora*** a sebességnövekedés? Su_{PL} =…2.8…..

Hogyan lehetne Su_{PL}***-t növelni?***..

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

Mi a hatása, ha egy feldolgozó szalag műveltvégzőinek 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 i3***86***-os mikroprocesszor a ***procedúrális*** utasítás egymásra hatást?

Hogyan oldja meg az i***486***-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?

Neumann-alapelvnek megfelelő számítógépekre vonatkozó alábbi kijelentések közül jelölje **x**-szel az igaz állítás(oka)t és - jellel a hamis(ak)at!

*Pontozásnál **minden jó jelölés +0,5** pont, **minden hibás jelölés -0,5 pont**, eredő >=0*

A utasítást és az adatot külön memóriában tárolja, így azok, külön sínen gyorsabb elérésűek, és a hely alapján egyértelműen azonosíthatók.	--
A CPU egy –már meglévő- utasításkészlet gyorsabb implementálása (emulálása), érdekében mindig huzalozott vezérlő egységet tartalmaz.	--
Az indirekt és az indexelt címzés alkalmazása előnyösen alkalmazható összetett adatszerkezetek kezelésénél.	x
Négycímes utasításkészletnél nincs szükség vezérlésátadó utasításra (pl.: feltétel nélküli ugró utasításra). Az utasítás tartalmazza a következő utasítás címét.	x
A RISC elvű processzoroknál az összetett utasítások megvalósítására gyakran mikroprogramozott vezérlőegységet alkalmaznak.	--
Pipe-line alkalmazásakor az egymás után következő fokozatok (elemi műveletvégzők) között négy átmeneti tárolót kell alkalmazni.	--

A be és kimenő adatokat a gyorsabb elérés érdekében az aritmetikai-logikai (ALU) egységben tárolja	--
DMA vezérlő alkalmazása esetén a ki/bemenő adatok a ALU-n keresztül olvashatók be/írhatók ki a memóriába.	--
Multitask-os rendszereknél a fizikai és a virtuális processzor összerendelést az összerendelő társprocesszor végzi.	--
Indirekt memóriacímzésnél az utasítás címrésze a következő utasítást tartalmazó memóriahelyre mutat.	--
A stack frame (verem keret) alkalmazásakor a bemenő paraméterek helyének felszabadítása (keret lebontása) mindig a függvényt hívó program feladata.	--
A stack frame a szubrutinokat (függvényeket) megvalósító algoritmusok elejét és végét jelöli ki a memóriában.	--

Az utasításokat memóriában tárolja, az adatokat perifériából kapja.	--
Az utasításokat és az adatokat bináris formában tárolja.	x
Az utasításokat és az adatokat a memóriában csak a program algoritmusa különbözteti meg.	x
Az utasításhoz és az adathoz külön-külön cím- és adatbusz tartozik.	--

Az eredeti Neumann modellnél a BE -és KI meneti egység különálló volt és a memóriával nem, csak az ALU-val tudott közvetlenül információt cserélni.	x
A utasítást és az adatot külön memóriában tárolja, az utasítást és az adatot a memóriában a tárolás formátuma különbözteti meg.	--
A CISC elvű számítógépekben az utasítások nem azonos méretűek és rendszerint több eltérő számú óraciklus alatt hajthatók végre, s ez előnyös a pipe line alkalmazásánál.	--
Az ENTER és a LEAVE utasítás az x86-os processzornál a verem keret (stack frame) alkalmazását támogatja.	x
A RISC processzoroknál az aritmetikai utasítások operandusai vagy regiszterben, vagy a memóriában találhatók.	
A térbeli és az időbeli lokalitási elvek miatt gyorsító tárakat (cache) csak az utasítások tárolására használhatnak.	--

A utasítást és az adatot külön memóriában tárolja, az utasítást és az adatot a memóriában a tárolás formátuma különbözteti meg.	--
A RISC elvű processzoroknál a gyorsabb működés elérésére huzalozott vezérlő egységet alkalmaznak.	x
RISC elvű processzoroknál a többkomponensű összetett címzés (pl.: indirekt bázisregiszteres, eltolással és indexelt,) előnyösen alkalmazható összetett adatszerkezetek kezelésénél.	--
Kétcímes utasításkészletnél nincs szükség vezérlésátadó utasításra (pl.: feltétel nélküli ugró utasításra). Az utasítás NEM tartalmazza a következő utasítás címét.	--
A CPU egy –már meglévő - utasításkészlet gyorsabb implementálása (emulálása), érdekében mikroprogramozott vezérlő egységet tartalmazhat.	x
ASAP ütemezésű Pipe-line alkalmazásakor a műveletvégzők közé, a változó méretű utasítások miatti változó működési idő különbség miatt, változó méretű puffer tárolót lehet alkalmazni.	x

Az utasításokat és az adatokat az op. memóriában csak a program algoritmusa különbözteti meg.	x
Az utasítást és az adatot külön memóriában tárolja, így az külön sínen gyorsabb elérésű, és a hely alapján egyértelműen azonosítható.	--
Indexelt címzésnél az effektív címet az utasításban lévő címrészből és egy regiszter tartalmából állítja elő.	x
Indirekt memóriacímzésnél az utasítás címrésze a következő utasítást tartalmazó memóriahelyre mutat.	--
A stack frame (verem keret) alkalmazásakor a bemenő paraméterek és a lokális változók címzésére a keretben bázisrelatív címzést alkalmaznak.	x
A stack frame a szubrutinokat (függvényeket) megvalósító algoritmusok elejét és végét jelöli ki a memóriában.	--

A utasítást és az adatot az operatív memóriában a tárolás formátuma és a helye különbözteti meg.	--
A RISC elvű processzoroknál a gyorsabb működés elérésére huzalozott vezérlő egységet alkalmaznak.	x
Az indirekt és az indexelt címzés alkalmazása előnyösen alkalmazható összetett adatszerkezetek kezelésénél.	x
Kétcímes utasításkészletnél nincs szükség feltétel nélküli ugró utasításra, illetve vezérlésátadó utasításra.	--
A CPU egy –már meglévő- utasításkészlet gyorsabb implementálása (emulálása), érdekében mikroprogramozott vezérlő egységet tartalmazhat.	x
Utasítás feldolgozó szalag alkalmazásakor, az egymás után következő műveletvégzők közé, átmeneti tárolót kell alkalmazni.	x

Az eredeti Neumann modellnél a kombinált BE/KI meneti egység csak a memóriával tud közvetlenül információt cserélni az ALU csak a memórián keresztül érhető el.	--
A stack frame (verem keret) alkalmazásakor a bemenő paraméterek és a lokális változók címzésére a keretben bázisrelatív címzést alkalmaznak.	x
A RISC elvű processzoroknál a gyorsabb működés elérésére mindig mikroprogramozott vezérlő egységet alkalmaznak.	--
Egycímes utasításkészletnél az utasítás címrésze a következő utasítást tartalmazó memória helyét adja meg.	--
A többkomponensű címzési módok előnyösen alkalmazhatók összetett adatszerkezetek kezelésére.	x
Ha az utasításkészlet tartalmaz I/O utasítást akkor annak végrehajtására mindig önálló I/O processzort kell alkalmazni.	--

Az egycímes utasításkészlet csak egy operandust használhat.	x
A háromcímes utasításkészlet egyik címe az eredmény helyét jelöli ki.	x
Kétkomponensű címzésnél az effektív címet az utasításban lévő címrészből és egy regiszter tartalmából állítja elő.	--
Indirekt memóriacímzésnél az utasítás címrésze a következő utasítást tartalmazó memóriahelyre mutat.	--
A stack frame a szubrutinokat (függvényeket) megvalósító algoritmusok elejét és végét jelöli ki a memóriában.	--
A stack frame (verem keret) alkalmazásakor a bemenő paraméterek és a lokális változók címzésére a keretben indexregiszteres többkomponensű címzést alkalmaznak.	--

Egyenlő időt igénylő elemi műveletvégzők esetén a feldolgozó szalagot csak ASAP ütemezéssel lehet működtetni.	--
A gyorsító tár (cache) a virtuális tár és az operatív memória közötti átvitel sebességét növeli meg.	--
Memória átlapolás (memory interleave,) esetén /32bites eléréskor/ egy LWORD és a közvetlenül utána következő LWORD ugyanabban a memóriatömbben (bank) található.	--
Egycímes utasításkészletnél az egyik operandus mindig a veremmemóriában található.	--
A CISC elvű processzoroknál az összetett utasítások megvalósítására gyakran mikroprogramozott vezérlőegységet alkalmaznak.	x
A tárkezelő egység (memory management unit, MMU) laphibát (page fault) jelez, ha egy felhasználói módú program az operációs rendszer adataihoz próbál hozzáférni. Akkor, ha egy lap nincs benn az operatív memóriában...	--

A CPU a -már meglévő- utasításkészlet gyorsabb implementálása(emulálása), érdekében gyakran mikroprogramozott vezérlő egységet tartalmaz. könnyebb (TICC OP.)	--
Az utasításokat és adatokat a memóriában csak a program algoritmusa különbözteti meg.	x
Az adatok könnyű, flexibilis kezelése érdekében sokféle, bonyolult, többkomponensű címzési módokat valósíthat meg. Mi valósíthat meg? Tömbök, stb. támogatására van a sok címzési mód.	x
Az utasításokat és az adatokat bináris formában tárolja.	x
A CISC elvű CPU legalább háromcímes utasításkészletet alkalmaz.	--
RISC elvű CPU operandusok címzéséhez kevés egyszerű címzési módot,és a memóriában található operandusokhoz csak LOAD(olvasás) és STORE(írás) típusú műveletet alkalmaz.	x

Az eredeti Neumann modellnél a kombinált BE/KI meneti egység az ALU-val nem, csak a memóriával tudott közvetlenül információt cserélni.	--
A RISC elvű processzoroknál a gyorsabb működés elérésére mikroprogramozott vezérlő egységet alkalmaznak.	--
Pipe-line alkalmazásakor, ha az egymás után következő fokozatok (elemi feldolgozó egységek) száma 2n, a teljesítmény és az utasításáteresztő képesség maximum 4n-szeresére nőhet. Csak 2n.	--
Egycímes utasításkészletnél az utasítás címrésze a következő utasítás helyét adja meg.	--
A többkomponensű címzési módok előnyösen alkalmazhatók összetett adatszerkezetek kezelésére.	x
Az I/O processzor, az átadott kezdőcímtől, a memóriában tárolt utasításokból álló perifériakezelő algoritmust önállóan, a CPU-val párhuzamosan hajtja végre.	x

Négycímes utasításkészlet esetén, a program következő utasítását az éppen végrehajtott jelöli ki.	x
Kétcímes utasításkészlet esetén az eredmény mindig az akkumulátorban keletkezik.	--
Kétcímes utasításkészlet esetén mindig kell vezérlésátadó (ugró) utasítás.	x
Bázisregiszteres memóriacímzésnél az utasítás címrésze a következő utasítást tartalmazó memóriahelyre mutat.	--
A RISC processzoroknál egy ciklus alatt végrehajtható utasításokat használnak, mert ez elősegíti a pipe-line szervezést.	x
A stack frame alkalmazása esetén a szubrutinok (függvények) lokális változóinak mindig a szubrutint hívó program foglal helyet. C vs Pascal	--

A háromcímes utasításkészlet egyik címe az eredmény helyét jelöli ki.	x
Az egycímes utasításkészlet csak egy operandust használhat.	x
Kétkomponensű címzésnél az effektív címet az utasításban lévő címrészből és egy regiszter tartalmából állítja elő.	--
Indirekt memóriacímzésnél az utasítás címrésze a következő utasítást tartalmazó memóriahelyre mutat.	--
A stack frame (verem keret) alkalmazásakor a bemenő paraméterek és a lokális változók címzésére a keretben bázisrelatív címzést alkalmaznak.	x
A stack frame a szubrutinokat (függvényeket) megvalósító algoritmusok elejét és végét jelöli ki a memóriában.	--

Az eredeti Neumann modellnél a BE -és KI meneti egység különálló volt és csak az ALU-val tudott közvetlenül információt cserélni, a memóriával nem.	x
A RISC elvű processzoroknál a gyorsabb működés elérésére decimális aritmetikát alkalmaznak.	--
Pipe-line esetén az utasítás egymásra hatás egyik fajtája a procedúrális egymásra hatás. Ennek elkerülésére a Pipe-line-t mindig teljesen kiürítik, és újra töltik.	--
Egycímes utasításkészletnél nincs szükség vezérlésátadó utasításra (pl.:feltétel nélküli ugró utasításra).	--
A DMA egység az utasításkészletben szereplő, de a CPU-ban nem megvalósított utasításokat hajtja végre a memóriában tárolt szubrutinok felhasználásával.	--
A többkomponensű címzési módok hátránya, hogy nem alkalmazhatók összetett adatszerkezetek kezelésére.	--

Az eredeti Neumann modellnél a BE - és KI meneti egység különálló volt, csak az ALU-val tudott közvetlenül információt cserélni, a memóriával nem.	x
A RISC elvű processzoroknál a gyorsabb működés elérésére huzalozott vezérlő egységet alkalmaznak.	x
Pipe-line alkalmazásakor, ha az egymás után következő fokozatok (elemi feldolgozó egységek) száma n, a teljesítmény maximum n-szeresére nőhet.	x
Egycímes utasításkészletnél a cím az egyik operandus helyét adja meg.	x
A többkomponensű címzési módok hátránya, hogy nem alkalmazhatók összetett adatszerkezetek kezelésére.	--
A társprocesszor az utasításkészletben szereplő, de a CPU-ban nem megvalósított utasításokat (pl.: aritmetikai) önállóan, a CPU működésével párhuzamosan, hajthatja végre.	x

Pipe-line alkalmazásakor az egymás után következő két utasítás azonos típusú rész-műveleteit (két fetch, két dekódoló, stb.) azonos időszeletben egyszerre dolgozzák fel.	--
Pipe-line esetén az utasítás egymásra hatás egyik fajtája a procedúrális egymásra hatás. Ez kiküszöbölhető, ha négy utasításon belül nincs két vezérlésátadó utasítás.	--
RISC elvű processzoroknál a gyorsabb működés érdekében nem használnak mikroprogramozott vezérlő egységet.	x
A CISC elvű processzoroknál csak LOAD és STORE típusú adatmozgató utasításokat valósítanak meg az operandusok elérésére.	--
A kétcímes utasításkészletnél az egyik cím az operandusok címe a másik az eredmény címe.	--
A stack frame (verem keret) a stack-ként (verem) felhasználható memóriaterület elejét és a végét jelöli ki a memóriában.	--

Négycímes utasításkészlet esetén mindig szükség van akkumulátor regiszterre.	--
Közvetlen operandusú (immediate) címzésnél az operandust az utasítást követő memóriahely tartalmazza.	x
A veremmutató (SP) tartalma x86 mikroprocesszornál aritmetikai utasításokkal módosítható, s ez felhasználható pl.: stack frame-né a lokális változó helyének lefoglalására.	x
Az ENTER és a LEAVE utasítás a verem keret (stack frame) alkalmazását támogatja.	x
A RISC processzoroknál az aritmetikai utasítások operandusai vagy regiszterben, vagy a memóriában találhatók.	--
A gyorsító tár (cache) tartalma write back írási stratégia esetén hosszabb ideig eltérhet az operatív memória megfelelő rekeszeinek tartalmától.	x

Négycímes utasításkészlet esetén nincs szükség vezérlésátadó (pl.: ugró) utasításra.	x
Négycímes utasításkészlet esetén, a program következő utasítását mindig az éppen végrehajtás alatt lévő jelöli ki.	x
A RISC processzoroknál egyforma számú órajelciklus alatt végrehajtható utasításokat használnak, mert ez elősegíti a pipe-line szervezést.	x
A CISC elvű processzoroknál csak LOAD és STORE típusú adatmozgató utasításokat valósítanak meg, mert ezek egyszerű címképzésűek és gyorsak.	--
Pipe-line alkalmazásakor, ha az egymás után következő fokozatok (elemi feldolgozó egységek) száma n az utasítás kibocsájtás és a teljesítmény minden esetben n- szeresére nő.	--
Pipe-line esetén a feldolgozási egymásra hatás kiküszöbölhető, ha megtöbbszörözik a szükséges elemi feldolgozóegységek számát.	--

Kétcímes utasításkészlet esetén, a program következő utasításának címét, a feltétel teljesülésétől függően, mindig az éppen végrehajtott utasítás címrészei jelölik ki.	--
Kétcímes utasításkészlet esetén az egyik cím az eredmény helyét, míg a másik cím a következő utasítás helyét jelöli ki.	--
Pipe-line esetén a procedúrális egymásra hatás kiküszöbölhető, ha a további műveletek feldolgozását felfüggesztik a feltételes vezérlés átadás feltételének kidolgozásáig.	x
Pipe-line alkalmazásakor az egymás után következő fokozatok (elemi feldolgozó egységek) között mindig átmeneti tárolókat kell alkalmazni.	x
A RISC elvű processzoroknál csak LOAD és STORE típusú adatmozgató utasításokat valósítanak meg, mert ezek egyszerű címképzésűek és gyorsak.	x
A CISC elvű processzoroknál az összetett utasítások megvalósítására gyakran mikroprogramozott vezérlőegységet alkalmaznak.	x

Az eredeti Neumann modellnél az utasításokat és az adatokat az operatív memóriában a tárolás helye és a tárolás formátuma különbözteti meg.	--
A stack frame (verem keret) alkalmazásakor a keretet az algoritmus kódja és a hozzá tartozó adatok elválasztására használják.	--
Pipe-line alkalmazásakor az egymás után következő négy utasítás azonos típusú részműveleteit (pl.: négy fetch, négy dekód., stb.) azonos időpillanatokban dolgozzák fel.	--
Kétcímes utasításkészletnél a két cím a két operandus helyét adja meg.	x
RISC elvű processzoroknál a gyors működés és az egyszerű címzés miatt csak LOAD és STORE típusú memória-referens utasításokat valósítanak meg.	x
A CPU egy-már meglévő- utasításkészlet gyorsabb implementálása (emulálása) érdekében mikroprogramozott vezérlőegységet tartalmazhat.	--

Az eredeti Neumann modellnél a kombinált BE/KI meneti egység az ALU-val nem, csak a memóriával tudott közvetlenül információt cserélni.	--
Az utasításokat és az adatokat az operatív memóriában azonos formában tárolja, így azokat csak a program algoritmusa különbözteti meg.	x
A RISC elvű processzoroknál a gyorsabb működés elérésére mindig mikroprogramozott vezérlő egységet alkalmaznak.	--
A RISC elvű processzoroknál egyetlen gépi ciklus alatt végrehajtható, egyforma hosszúságú utasítások kialakításával a PIPE-LINE elv előnyösen alkalmazhatóa gyorsításra.	x
Egycímes utasításkészletnél a cím a következő utasítás helyét adja meg.	--
A többkomponensű címzési módok előnyösen alkalmazhatók összetett adatszerkezetek kezelésére.	x

Az eredeti Neumann modellnél a BE -és KI meneti egység különálló volt és a memóriával nem, csak az ALU-val tudott közvetlenül információt cserélni.	x
A utasítást és az adatot fizikailag mindig külön álló memóriában tárolja, így az külön sínen gyorsabb elérésű, és a hely alapján egyértelműen azonosítható.	--
A RISC elvű processzoroknál a gyorsabb működés érdekében nem alkalmaznak mikroprogramozott vezérlő egységet.	x
A RISC elvű processzoroknál az egyszerű címzés miatt csak LOAD és STORE típusú műveletekkel érik el a memóriában lévő adatokat.	x
Háromcímes utasításkészletnél az egyik cím az eredmény helyét, a másik kettő az operandusok helyét adja meg.	x
Indirekt memóriacímzésnél az utasítás címrésze a következő utasítást tartalmazó memóriahelyre mutat.	--

A ***RISC***-alapelvnek megfelelő számítógépekre vonatkozó alábbi kijelentések közül jelölje x-szel az igaz(ak)at és - jellel a hamis(ak)at!

A CPU-nál a gyorsabb működéshez szükséges egyenlő számú órajelciklust igénylő utasítások alkalmazhatósága érdekében nem alkalmazzák a pipe-line elvet.	--
Az utasítást memóriában tárolja, az adatokat mikroprogram vezérlésű gyors elérésű perifériák tárolják.	--
Az adatok könnyű, flexibilis kezelése érdekében bonyolult, többkomponensű címzési módokat valósít meg.	--
Az utasításokat és az adatokat bináris formában a memóriában tárolja, ezért ezeket csak a program algoritmusa különbözteti meg.	x

Mikroprogramozott vezérlő egységet tartalmaz.	--
Az utasítást memóriában tárolja, az adatokat perifériából kapja.	--
Az utasításhoz és az adathoz külön-külön cím- és adatbusz tartozik.	--
Az utasításokat és az adatokat bináris formában tárolja.	x

A CPU a -már meglévő- utasításkészlet gyorsabb implementálása (emulálása), valamint a gyorsabb működés érdekében mikroprogramozott vezérlő egységet tartalmaz.	--
Négycímes utasításkészletet alkalmaz.	--
Az utasításokat és az adatokat bináris formában tárolja.	x
Az általában egyetlen ciklus alatt végrehajtható, egyforma hosszúságú utasítások kialakítása miatt, előnyösen alkalmazható a PIPE-LINE elv a gyorsításra.	x
Az utasításokat és az adatokat a memóriában csak a program algoritmusa különbözteti meg.	x
Az operandusok címzéséhez kevés egyszerű címzési módot alkalmaz, a memóriában található operandusokhoz csak LOAD(olvasás) és STORE(írás) típusú műveletet(címzést) használ.	x

Milyen összefüggés figyelhető meg a CPU-k tranzisztorainak száma és az idő között?

Milyen összefüggés figyelhető meg a CPU-k működési sebessége és az idő között?

Milyen összefüggés figyelhető meg a memóriák tranzisztorainak száma /és a vele arányos kapacitása/ és az idő között?

Milyen összefüggés figyelhető meg a memóriák működési sebessége és az idő között?

Ismertesse a memóriák hierarchikus felépítését!

Ismertesse a lokalitási elvet, mit jelent a ***találati arány*** (Hit rate), illetve a ***hiba arány*** (Miss rate)

Mit jelent a térbeli, illetve az időbeli lokalitási elv? Adjon egy-egy példát, ami alátámasztja, illetve ami sérti az előbbi elveket!

Mit jelent a látszólagos hozzáférési idő, hogyan számítható ki adott Hr, illetve Mr esetén?

Mi a cache szerepe?

Rajzolja fel egy direkt szervezésű cache leképezésének vázlatát!

Rajzolja fel egy direkt szervezésű cache ***működésének blokkvázlatát***!

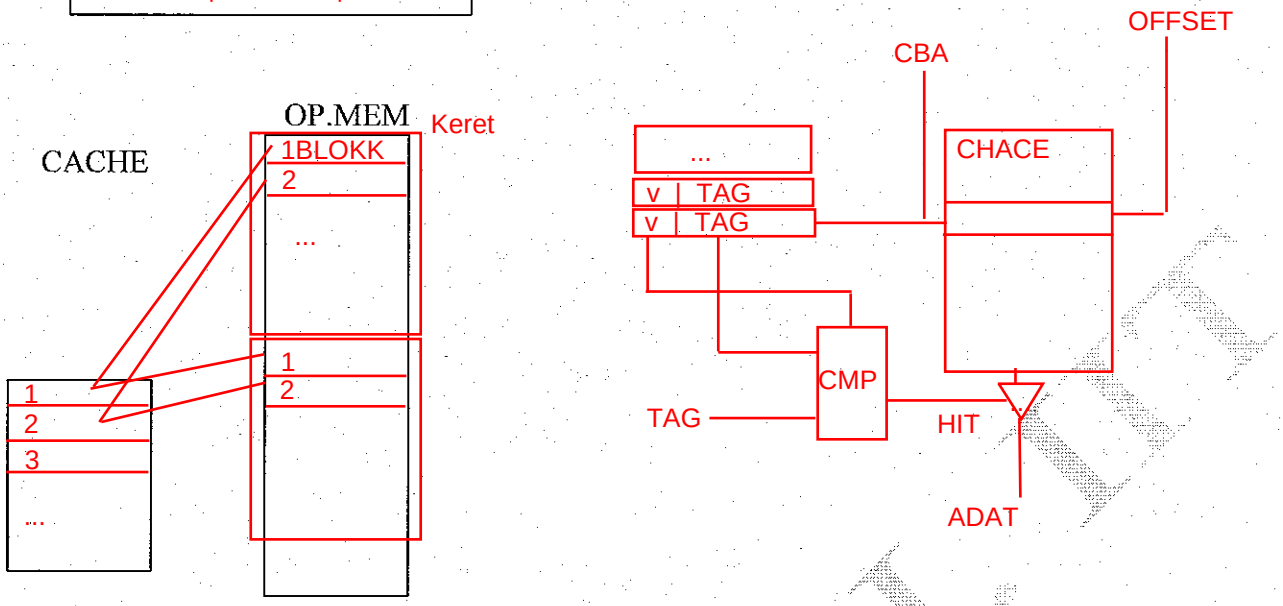
Rajzolja fel egy két-utas direkt leképzésű (set asszociativ) cache blokkvázlatát! Ismertesse a működést, valamint a megoldás előnyét és hátrányát!

Rajzolja fel a ***direkt*** szervezésű cache ***leképzésének*** és ***működésének blokkvázlatát!***

Leképzés OP. MEMÓRIA CÍM

TAG	CBA	OFFSET

Működés blokkvázlata:



Röviden ismertesse a ***leképzés elvét*** és a ***működést***

OM-et chace méretű keretekre osztjuk.
Minden keretben ugyan annyi blokk van, mint a chaceben, adott blokk csak a neki megfelelő számú chace beli blokkba kerülhet.

Mi a ***leképzés előnye*** és a ***hátránya, hogyan csökkentik*** az utóbbit?

Egyszerű, olcsó, gyors, blokkcsere algoritmus nélküli.
HIT rate alacsony. Méret növelésével hitrate javul.
Legjobb kompromisszum n-utas direkt leképzés.
Asszociatív leképzéssel kombinált direkt leképzés.

Használhatnak LRU blokk-csere (block replacement policy) stratégiát?

Válasz: Nem

Indoklás: Minden blokk 1 helyre kerülhet.

Hogyan ***változna*** a komparátorok száma kétutas direkt leképzés esetén?

Változás: 1 helyett 2 kell.

Indoklás: Mindkét részen külön-külön meg kéne nézni, hogy nincs-e ott az adott blokk, mert minden blokk mindkét 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?

Mit jelent az okos előrelátó behozatali stratégia?

Sorolja fel és ismertesse a cache blokkcsere stratégiákat!

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

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

Alkalmazható e az ***igény szerinti*** stratégia direrkt, illetve n utas direkt leképzésnél?

Válasz: Igen

Indoklás: Semmi nem gátolja. Ha kell, behozzuk a megfelelő helyre (direkt) vagy megfelelő helyek egyikére (asszociatív).

Mit jelent az előrelátó stratégia?: Megpróbáljuk kitalálni, hogy mely blokkra lesz szükségünk, és azt előre betöltjük.

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

486: Egyszerű előrelátó, mely behozza a következő blokkot is.

Pentium: Spekulatív előrelátó, mely megpróbálja kitalálni, mely blokkra lesz szükség.

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%.

senkit nem érdekel!

Számítsa ki mekkora a felhasználó által látott átlagos (látszólagos) memória hozzáférési idő (a cache blokk betöltési idejét figyelmen kívül hagyjuk)!

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=……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!

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!

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?

Atírjuk cacheben, majd onna kiírjuk OM-be.

Használható-e ez a stratégia lapszervezésű ***virtuális tárkezelés*** esetén? ***Indokolja*** a választ!

Szerintem nem.

Indoklás: Nem biztos, hogy a hivatkozott lap az OM-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: kb. 8x

Használható- e a módszer direkt leképzés esetén? Igen

Indoklás: Ha az OM-et jól osztjuk fel, akkor lehet. Nem szabad egymás után ugyan olyan sorszámú blokkot címezni.

Egy **négy utas set asszocitatív** 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)?	6
Milyen szervezésű (szószám x bitszám) egy vezérlő (TAG) tár?	1024*17
Hány TAG komparátort tartalmaz a cache?	4

CBA 10 bit (4096 = 4 * 1024)

TAG (16 + V) = 17 bit

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.

Hány bites a TAG a cache-ben?	17
Hány bites a CBA (cache block address)?	7
Hány TAG-komparátort tartalmaz a cache?	4

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

TAG 16 bit

Hány bites az offset (eltolás)?	6
Hány bites a CBA(blokkazonosító) mező?	10
Hány TAG komparátort tartalmaz a teljes cache?	4

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

Változás: Csökkenne

Indoklás: Merevebb a betöltés, messzebb kerülünk a teljesen asszociatívtól.

Komparátorok száma: 4 **-ról(ről)** 2 **-ra(re)**

Komparátor(ok) bitszélessége 16 **bitről** 15 **bitre**

Egy teljesen asszocitatív 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%.

TAG: 25 + 1 bit

Hány bites az offset (eltolás)?	7
Milyen szervezésű (szószám x bitszám) a vezérlőtár?	2048*26
Mennyi a memória átlagos elérési ideje? 0.92*10+0.08*75	15.2

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

Mit jelent az ***LFU*** stratégia? Legritkábban használt blokk helyére hozzuk be az új blokkot.

Használható e LRU blokkcsere sratégia ***direkt*** leképzésű cache-nél ? ***Indokolja*** a választ! Nem, mert minden blokk csak a megadott helyre hozható be, nincs blokkcsere algoritmus.

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?	15
Hány bites a CBA (cache block address)?	9
Hány TAG-komparátort tartalmaz a cache?	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?	1
Hány bites a TAG komparátor?	15
Számítsa ki a cache méretét (Kbyte-ban)?	512

OFFSET: 8
CBA: 9
TAG: 15

Egy **négy utas set asszocitatív** 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)?	8
Milyen szervezésű (szószám x bitszám) egy vezérlő (TAG) tár?	128*18
Mennyi a memória látszólagos átlagos elérési ideje?	15.2

OFFSET: 8
CBA: 7
TAG: 17+1

Egy **két utas set asszocitatív** vezérlést alkalmazó gyorsító tár(cache) adatai a következők: a blokkméret 128 byte, a teljes gyorsító tár összesen 2048 blokkot tartalmaz. A vezérlőtárakban (TAG) 1 bittel jelezzük a bejegyzés érvényességét. Az operatív memória címe 32 bites.

Hány bites az offset (eltolás)?	7
Milyen szervezésű (szószám x bitszám) egy vezérlő (TAG) tár?	1024*16
Hány TAG komparátort tartalmaz a cache?	2

OFFSET: 7
CBA: 10
TAG: 15+1

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) kezdőcímre 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 sorszámát és a címrészének az értékeit!

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 ***1000H kezdőcímre*** (4KB) címfolytonosan lefordított ***12KB*** méretű program. A fizikai memóriában az ***512KB..520KB*** (80000H-81FFFH) és az utolsó ***4KB*** tartományban (FF000H-tól) van szabad memóriahely. Írja fel az indexregiszter-tömb –***a program futásakor felhasznált***– regisztereinek sorszámát hexadecimális tartalmát, ha a legfelső helyértéken alkalmazott vezérlő bitek értéke ***01***!

Indexregiszter-tömb mérete (regiszter x bit): 16 x 10

Regisztersorszám	**Hexadecimális érték**
1	0110000000b = 180h
2	0110000001b = 181h
3	0111111111b = 1FFh

1000h kezdőcím miatt kell az 1-es regisztertől számolni!

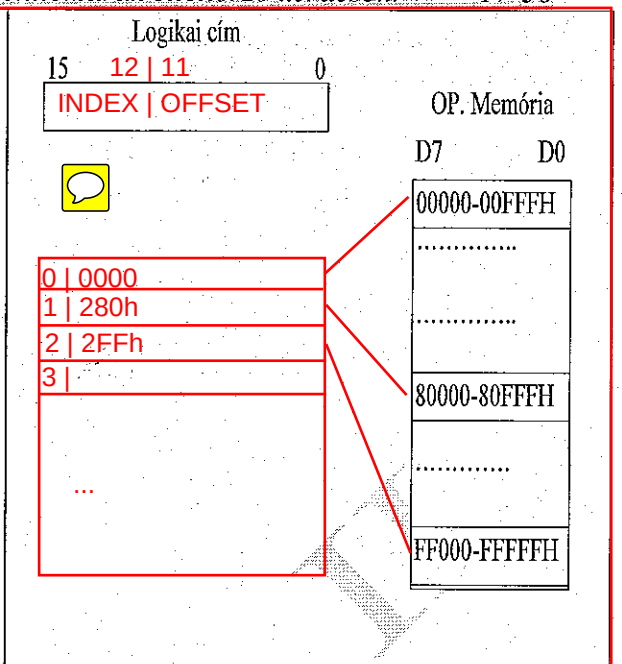
Egy 16 bites logikai és egyben fizikai címmel rendelkező rendszerben (pl.: 8085) indexelt leképezésű (index regiszter-tömböt alkalmazó) memóriaszervezéssel 1Mbyte-ra kell bővíteni a fizikai memória méretét. Az index regiszter-tömb 32 regisztert tartalmaz. Vezérlésre (adminisztrációra) 3 bitet használunk)

Hány bites az offset (eltolás)?	11
Hány bites a regiszter tömb egy regisztere?	9+3=12
Hány blokkot (lapot) tartalmaz a teljes memória?	512
Maximálisan hány blokk (lap) lehet egyszerre aktív?	32

Rajzolja fel az ***Indexelt leképzés*** 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*** **leképzésnél a**dott egy ***1000H*** kezdőcímre 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 konstans 0-t tartalmaz, nem programozható.



Egy indexelt leképezést alkalmazó számítógép címsínje 16 bites, ebből a 3 MSB az index. Az operatív memória 256 kByte méretű. 18 bit = 5index + 13offszet

Milyen az indexregiszter tábla szervezése (szó x bit) ha vezérlési célokra 2 bit szükséges? 8x7

Ha egy 1980h byte hosszú programrészt a hexa 8080h címre fordítunk le, hányas indexű indexregiszter(ek) tartamát kell beállítani? 4

Milyen (hexa) értéket kell ebbe az indexregiszterbe írni, ha a fizikai memóriában 2A000h-2BFFFh tartományban van üres hely a program számára? 15h

Egy indexelt leképezést alkalmazó számítógép logikai címe 16 bites, ebből a 4 MSB az index. Az operatív memória 512 kByte méretű. Offszet: 12 bit indexregiszter hossza: 7+2 bit

Milyen az indexregiszter tábla szervezése (szó x bit) ha vezérlési célokra 2 bit szükséges? 16*9

Ha egy 0E80h byte hosszú programrészt a hexa 8080h címre fordítunk le, hányas indexű indexregiszter(ek) tartamát kell beállítani a program futtatásához? 8

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? 1FAh

Adott egy címfolytonosan 0 kezdőcímre 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 lapleíró 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?

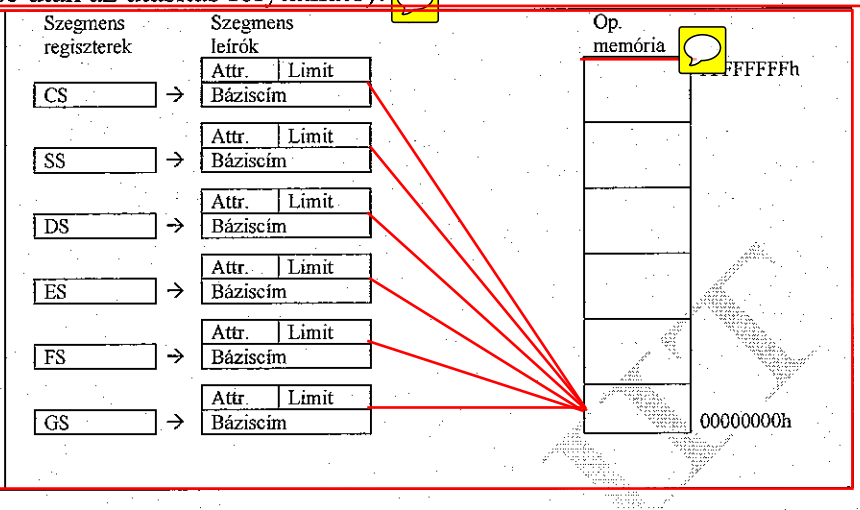
Ismertesse a szegmensszervezésű virtuális tárkezelés elvét, rajzolja fel a megoldás vázlatát!

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

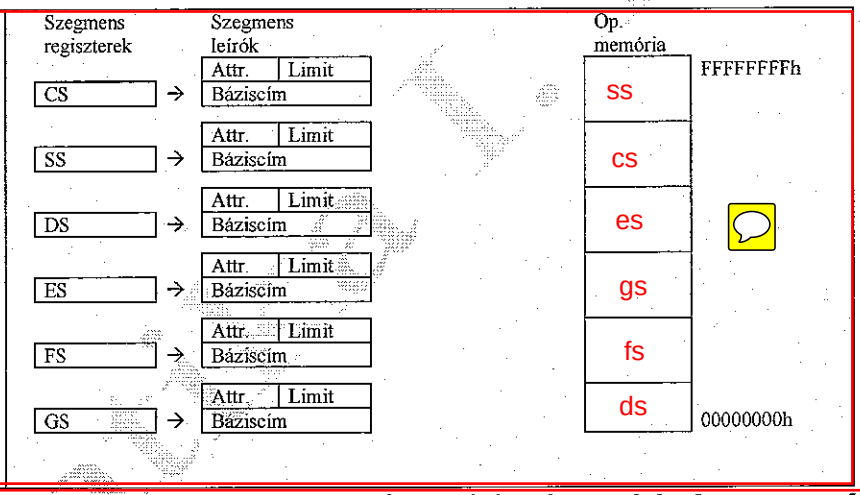
Milyen memóriaképeket /memória elérési modelleket/ biztosíthat a 386-os mikroprocesszor?

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) memória-modell (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.



A mellékelt ábrába **rajzolja be** hová mutatnak az egyes szegmensleírók értékei **386-os** mikroprocesszor **strukturált** memória-modell (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.



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 címtartománynak kell egybefüggőnek lennie adott szegmensnél.
Hátránya: Nagyobb memóriaterület szükséges a megvalósításához.

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

Milyen ***hátránnyal*** járna az ***eglépcsős*** laptábla alkalmazása?

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

A ***386***-os mikroprocesszor bekapcsolt lapszervezés esetén a 32bites lineáris címből kétlépcsős laptábla szervezéssel állítja elő a fizikai címet. A lapméret 4Kbyte. A laptábla könyvtár elejére a CR3regiszter tartalma mutat. Egy lapleíró bejegyzés 32bitet tartalmaz. Eltolás 12 bit
Marad 20 bit ->PDS = 10 és PTS = 10

Minimálisan mekkora op. memória kell a kétlépcsős laptábláknak?
Benn kell lennie a laptáblakönyvtárnak és egy lapleírónak legalább: 2^10 * 32 + 2^10 * 32 = 64kbit

Mekkora lenne a laptábla mérete eglépcsős szervezéssel?
20 bit lenne a laptáblakönyvtárt leíró PDS, és 32bit / bejegyzés: 32Mbit

Mi a kétlépcsős szervezés ***hátránya*** az eglépcsőssel szemben?
Lehet, hogy a hivatkozott lapleíró nincs benn az OM-ben, ezt be kell hozni így lassabb.

Hány értékes bitet kell a CR3-nak tartalmaznia? 20-t.

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!

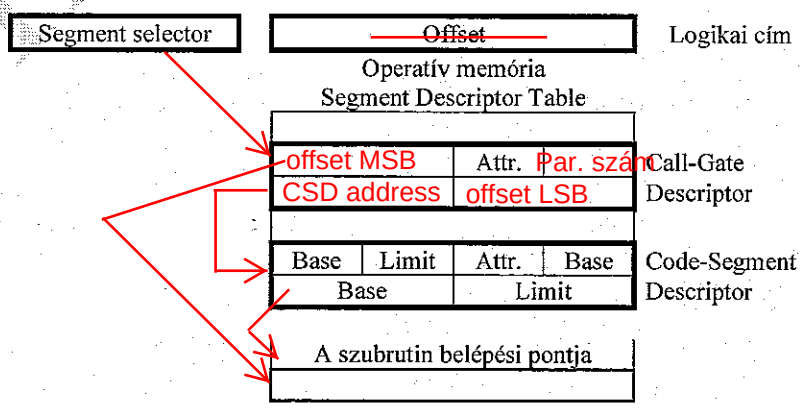
Sorolja fel az i386-os mikroprocesszor privilégium szintjeit, és a hozzájuk kapcsolódó elérési szabályokat!

Ismertesse az i386 lapokra vonatkozó védelmét!

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

Rajzolja fel a CALL GATE működését jellemző szegmenselérés vázlatát!

A mellékelt ábrába **írja be**, hogy mit tartalmaz a Call-Gate Descriptor és **jelölje** (rajzolja fel), **hogyan történik** a Gate-n keresztüli belépési pont meghatározása (feltételezze, hogy az ábrán cél leíróját adtuk meg).

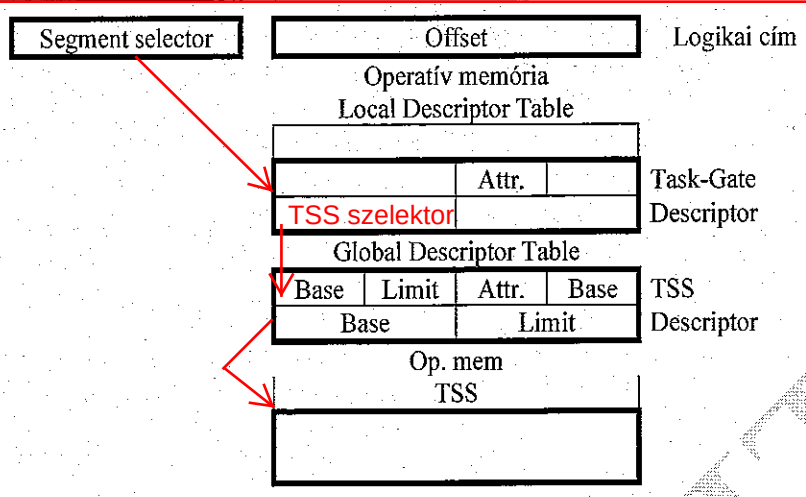


Mi a Task GATE szerepe, hogyan biztosítja ezt?

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).



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?

Multitaszkos rendszernél az alábbi kijelentések közül jelölje x-szel az igaz(ak)at és – jellel a hamis(ak)at!

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

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

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

Minden taszkhoz egy külön fizikai processzor tartozik, a taszk-fizikai processzor összerendelést hardver úton az arbiter egység végzi.	--
A processzor taszkváltáskor hardveresen menti a taszk teljes állapotát, amelyet a hardveres TSS egység végez.	--
A virtuális és fizikai processzorok időbeli összerendelését külön hardver egység végzi.	--
A i386/486 processzornál minden taszkhoz külön lokális szegmensleíró tábla és lapleíró könyvtár (directory) tartozhat.	--
A i386/486 processzornál a taszkok nem lehetnek újra belépők (re-entrant) mivel minden taszkhoz csak egy leíró (TSS) tartozhat.	x
A i386/486 processzornál minden új taszk TSS tartalmának betöltése után a CR3 tartalmának megváltozása miatt a lapszervezés TLB-jét realizáló cache tartalmát érvényteleníteni kell.	

Minden taszkhoz egy külön fizikai processzor tartozik.	--
A processzor taszk-váltáskor részben hardveresen részben szoftveresen menti a taszk teljes állapotát leíró információt.	x
A taszk-váltást az operációs rendszer (pl.: időosztásos elven működő) ütemezője is kezdeményezheti.	x
A i386/486 processzornál minden taszk csak a Globális szegmensleíró táblát (GT) használhatja.	--
A i386/486 processzornál minden taszkhoz annyi taszk állapot leíró szegmens(TSS) tartozhat, ahány helyről hívják ezért önmagát is többször meghívhatja.	--
Multiprocesszoros rendszereknél, statikus feladat hozzárendelés esetén, egy adott feladatot (az ezt megoldó taszkot) mindig ugyanaz a fizikai processzor végez el.	x
A i386/486 processzornál minden taszknak saját lap könyvtár (page directory) táblája lehet, ezért taszk-váltáskor ezt is cserélni kell.	x

Minden fizikai processzorhoz több virtuális processzor tartozhat.	x
A virtuális és fizikai processzorok időbeli összerendelését külön hardver egység végzi.	--
A taszkok állapotát leíró információt taszkváltáskor menteni, illetve cserélni kell.	x
A i386/486 processzornál minden taszkhoz csak egy taszk állapot leíró szegmens(TSS) tartozhat, ezért önmagát nem hívhatja meg.	x
A i386/486 processzornál minden taszkhoz külön lokális szegmensleíró tábla (LT) tartozhat.	x
Dinamikus feladat-hozzárendelésű multiprocesszoros rendszereknél, egy fizikai processzor minden funkciót (feladatot, illetve az ezt realizáló taszk futtatást) elláthat.	x

Minden fizikai processzorhoz több virtuális processzor tartozhat.	x
A virtuális és fizikai processzorok időbeli összerendelését külön hardver egység végzi.	--
A taszkok állapotát leíró információt taszkváltáskor menteni, illetve cserélni kell.	x
A i386/486 processzornál minden taszknak saját lap könyvtár (page directory) táblája lehet, ezért taszkváltáskor ezt is cserélni kell.	x
A i386/486 processzornál minden taszkhoz külön lokális szegmensleíró tábla (LDT) tartozhat, ezért taszkváltáskor az ennek a leírójára mutató regiszter (LDTR) tartalmát is cserélni kell.	x
Multiprocesszoros rendszereknél, dinamikus feladat hozzárendelésnél esetén egy fizikai processzor minden funkciót (feladatot, az ezt realizáló taszk futtatást) elláthat.	x

A 386/486 processzorban mely információkat tartalmazza a **szegmensleíró** bejegyzés az alábbiak közül? A **helyes** állítás(oka)t jelölje **x**-szel, a **hibás**(aka)t - jellel!

A szegmens hossza	x
A szegmens típusa	x
Az aktuális privilégiumszint(CPL)	--

A szegmens leírótábla kezdőcíme	--
A jelenlét (present) jelzőbit	x
A globális/lokális leírótáblát jelző (T) bit	--

A szegmens kezdőcíme	x
A szegmens leírótábla kezdőcíme	--
A szegmensleíró privilégiumszintje (DPL)	x

A szegmens típusa	x
A GDT meglétét (GDT present) jelző bit	
A globális/lokális leírótáblát jelző (T) bit	--

A szegmenshez tartozó aktív lapok leíró táblájának kezdőcíme és száma	x
A szegmens (lineáris-) kezdő címe	x
A szegmens be van töltve az operatív tárolóba	x

Az aktív szegmensleíró tábla kezdőcíme	--
A szegmensleíró privilégium szintje és a szegmens típusa	x
A szegmens hossza	x

A szegmensleíró hossza	x
A szegmens leírótábla kezdőcíme	--
Az aktuális privilégiumszint(CPL)	--

A szegmens típusa	x
Az LDT betöltve (present) jelzőbit a leírójában	x
A globális/lokális leírótáblát jelző (T) bit	--

A szegmens (lineáris-) kezdőcíme	x
A szegmens fut jelzés	
A szegmenst tartalmazó swap-file neve	--

A leíró privilégiumszintje	x
A szegmens hossza	x
Az aktív lapok száma	--

A szegmens (lineáris-)kezdőcíme	x
A szegmens be van töltve az op. tárba	x
A szegmensleíró tábla kezdőcíme	--

A TASK privilégiumszintje	--
A szegmens hossza	x
Az aktív lapok száma	--

A 386/486 processzorban mely információkat tartalmazza a ***lapleíró*** bejegyzés az alábbiak közül? A **helyes** állítás(oka)t jelölje **x**-szel, a **hibás**(aka)t - jellel!

A lap tartalma be van töltve az op. tárba (present)	x
A lapra betöltés után történő írást jelző (D) bit	x
A lap (lineáris-) kezdőcíme	x

A lapra hivatkozó szegmens leírótábla kezdőcíme	--
A lap hossza	--
A lapkönyvtár tábla kezdőcíme (CR3)	--

A lap (lineáris-)kezdőcíme	x
privilégiumszint	x
A lap tartalma be van töltve az operatív tárba (present)	x

A szegmens leírótábla kezdőcíme	--
A lap hossza	--
A laptábla-könyvtár tábla kezdőcíme	--

A lap tartalma be van töltve az operatív tárolóba	x
A lap (lineáris-) kezdő címe	x
A lap privilégium szintje és attribútumai	x

A lapkönyvtár tábla (directory) kezdőcíme	--
Az aktív lap hossza	--
A globális szegmensleíró tábla kezdő címe	--

A lap tartalma be van töltve az op. tárba (present)	x
A lapra betöltés után történő írást jelző (D) bit	x
A lap (lineáris-) kezdőcíme	x

A lapra hivatkozó szegmens leírótábla kezdőcíme	--
A lap hossza	--
Az egy–illetve kétlépcsős működést jelző bit	--

A **386/486** processzor védett üzemmódban, szegmentált lapszervezésű virtuális memóriakezelést alkalmaz. A **helyes** állítás(oka)t jelölje **x**-szel, a **hibás**(aka)t - jellel!

Minden taszkhoz külön laptábla könyvtár kezdőcím tartozhat.	x
Ha a kiépített fizikai memória méretét megduplázzuk, a laptábla könyvtár mérete is megduplázódik.	--
Védett üzemmódban a szegmentálás nem kapcsolható ki, mindig ez végzi a logikai címtérből a lineáris címtér előállítását.	
A kétlépcsős laptábla szervezés kétszer akkora memóriaterületet igényel, mint az egylépcsős laptábla.	--
A szegmensleíró báziscíme bitjeinek száma független a lapszervezés ki- vagy bekapcsolásától.	
A virtuális, a lineáris és a fizikai címtartomány egyaránt 4GByte.	--

A szegmentált lapszervezés virtuális címtartománya 64 Terabyte, a fizikai címtartomány 4GByte.	x
A laptábla könyvtár mérete függ a háttértárolón tárolt szegmensek számától és méretétől.	x
A kétlépcsős laptábla szervezés hátránya, hogy a laptábla nagyobb memóriaterületet igényel, mint az egylépcsős laptábla.	--
A virtuális és a fizikai processzor összerendelését az ütemezést megvalósító operációs rendszer a taszkhoz rendelt TSS tartalma alapján oldja meg.	x
A lapleíróban a lap báziscíme bitjeinek száma nem függ a fizikai memória éppen kiépített (pl.:2GB vagy 4GB) méretétől.	x
Minden új taszk TSS tartalmának betöltése után, a CR3 tartalmának megváltozása miatt, a TLB-t realizáló cache tartalmát érvényteleníteni kell.	

Egy szegmens tetszőleges méretű (max. 4 GB), tetszőleges címen kezdődhet, a lineáris címtartomány 4 Gbyte, a fizikai címtartomány 4 Gbyte, a virtuális címtartomány 64 Terabyte.	
A szegmensleíró tartalmazza a szegmens kezdőcímét, a szegmens hosszát és a leíró privilégium szintjét is.	
A kétlépcsős laptábla szervezés hátránya, hogy a laptábla nagyobb memóriaterületet igényel, mint az egylépcsős laptábla.	--
A lapleíróban a lap báziscíme bitjeinek száma függ a fizikai memória éppen kiépített (pl.:2 GB vagy 4 GB) méretétől.	--
Az I/O utasítások privilegizáltak, ezért nem minden privilégiumszinten futó program tudja végrehajtani azokat.	x
Minden taszkhoz külön lokális szegmensleíró tábla (LDT) és laptábla könyvtár (directory) kezdőcím tartozhat.	

Egy lap fix méretű (4KByte), tetszőleges címen kezdődhet, lineáris címtartománya 4GByte, a fizikai címtartomány 64 Terabyte.	--
A lapleíró tartalmazza a lap kezdőcímét, a lap hosszát a lapleíró privilégium szintjét, valamint a lap privilégium szintjét.	--
A kétlépcsős laptábla szervezés előnye, hogy a laptábla kisebb memóriaterületet igényel, mint az egylépcsős laptábla.	x
A szegmensleíróban a szegmens báziscíme bitjeinek a száma, valamint a szegmens hosszát megadó bitek száma függ a háttértárolón tárolt szegmensek számától és méretétől.	
Az I/O utasítások nem privilegizáltak, ezért minden privilégiumszinten futó program végre tudja hajtani azokat	--
Minden új taszk TSS tartalmának betöltése után a CR3 tartalmának megváltozása miatt az összes betöltött lap tartalmát érvényteleníteni kell.	

A szegmentálás és a lapszervezés virtuális címtartománya 16 GByte, a fizikai címtartomány 4GByte.	--
A laptábla könyvtár mérete felére csökkenhető, ha a kiépíthető 4 Gbyte fizikai memóriából csak 2 Gbyte-ot építünk be az adott számítógépbe.	--
A kétlépcsős laptábla szervezés kétszer akkora memóriaterületet igényel, mint az egylépcsős laptábla.	--
Védett üzemmódban a szegmentálás, a felhasználó igénye szerint, a lapleíróban adott információ alapján kapcsolható ki.	
A szegmensleíró báziscíme bitjeinek száma függ a fizikai memória éppen kiépített (2GB vagy 4GB) méretétől.	--
Minden taszkhoz külön lokális szegmensleíró tábla és laptábla könyvtár kezdőcím tartozhat.	

A lineáris és a fizikai címtartomány egyaránt 4GByte. Lineáris címtér MAX 4GB.	--
Ha a kiépített fizikai memória méretét megduplázzuk, a laptábla könyvtár mérete is megduplázódik.	--
A kétlépcsős laptábla szervezés kétszer akkora memóriaterületet igényel, mint az egylépcsős laptábla.	--
Védett üzemmódban a szegmentálás nem kapcsolható ki.	x talán
A szegmensleíró báziscíme bitjeinek száma független a lapszervezés ki- vagy bekapcsolásától.	
Minden taszkhoz külön laptábla könyvtár kezdőcím tartozhat.	

A 386/486 processzoros számítógépen többtaszkos (multitasking) operációs rendszer működik, a védelmi lehetőségek teljes kihasználásával. A **helyes** állítás(oka)t jelölje **x**-szel, a **hibás**(aka)t - jellel!

Multitaszkos rendszernél a fizikai és a virtuális processzor összerendelését megvalósító állapotleíró kezelését és a TASK váltás ütemezését az MMU végzi.	--
Minden TASK-nak külön saját szegmensleíró táblája lehet.	
A kétlépcsős laptábla szervezés előnye, hogy kevesebb memóriaterületet igényel, mint az egylépcsős.	x
A szegmensleíró báziscím bejegyzés bitjeinek a száma függ az aktuálisan kiépített fizikai memória méretétől.	--
A GATE a különböző privilégium szintek közötti kommunikációt megvalósító hardver egység.	--
A szegmensleírónak is van privilégium szintje, és ennek ellenőrzése is a védelmi mechanizmus része.	x

Minden taszkhoz 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.	--

Multitaszkos rendszernél egy taszknak minden védelmi szinten van kód adat és veremszegmense.	
Az I/O utasításokat nem minden privilégiumszinten lehet végrehajtani.	x
A tényleges privilégiumszint (EPL) értéke, az aktuális (CPL) és az operandus privilégiumszint (RPL) közül, a nagyobbik számértékűvel egyezik meg, azaz EPL=max(CPL,RPL).	x
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.	x
A megszakítási leíró tábla (IDT) is tartalmazhat taszk kapukat (TASK GATE).	

A 386-os mikroprocesszor bekapcsolt lapszervezés esetén a 32bites lineáris címből kétlépcsős laptábla szervezéssel állítja elő a fizikai címet. A lapméret 4Kbyte. A laptábla könyvtár elejére a CR3regiszter tartalma mutat. Egy lapleíró bejegyzés 32bitet tartalmaz. OFFSET : 12bit

Hány értékes bitet kell a CR3-nak tartalmaznia?	20
Minimálisan mekkora op. memória kell a kétlépcsős laptábláknak?	64kB
Egy TASK-hoz maximálisan hány lapleíró tábla tartozhat?	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 perifériakezelés blokkvázlatát!

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

Mit jelent a logikai szintű I/O kezelés?

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.	x
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.	x
Az eszközszintű 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.	? ha akarom igen, ha akarom nem, nem 1ért.

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.

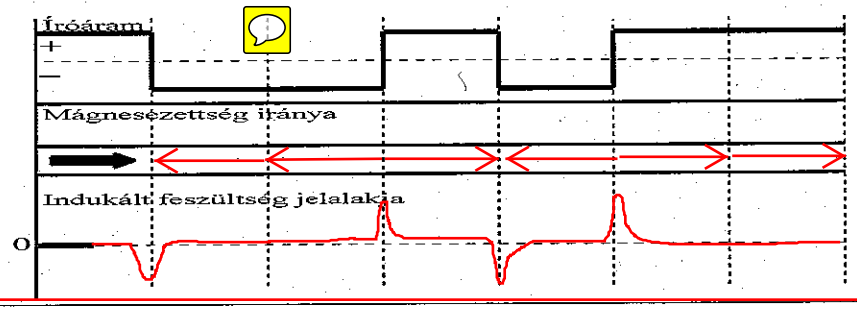
Röviden sorolja fel magneto-rezisztív olvasó fej előnyeit!

Miben különbözik a horizontális, illetve a vertikális (prependicular) 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 kel 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!

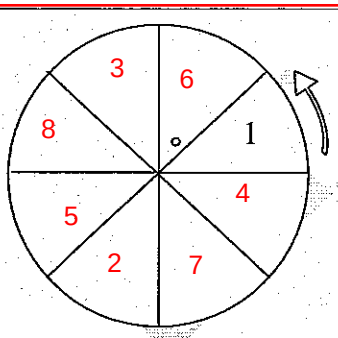
Röviden magyarázza el a szoft-szektor szervezés elvét és az ehhez szükséges információs mezők felépítését szerepét!

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

Mi a GAP szerepe?

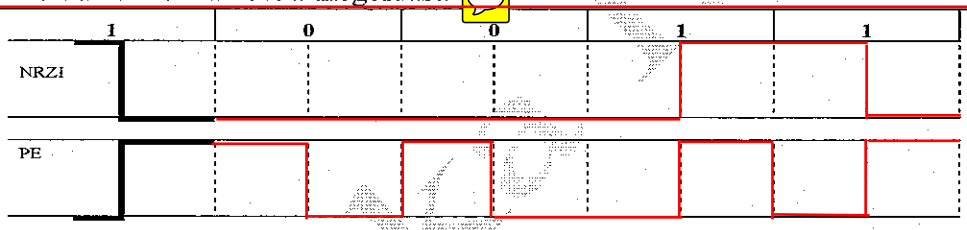
Magyarázza el, hogyan szinkronozódik fel íráskor, illetve olvasáskor a mágneslemezes iró olvasó elektronika a formázáskor felvitt információra

Egy floppy diszk egy ***sávja 8 szektort*** tartalmaz.
Rajzolja fel a szektorok ***fizikai*** elhelyezkedését a megadott forgásiránynál ***3:1*** szektor ***interleave*** esetén! A feldolgozás túl lassú.
Milyen probléma kiküszöbölésére használható a megoldás?
Optimális esetben hány körülfordulás kell egy sáv összes szektorának írásához vagy olvasásához ***3:1*** szektor-interleave esetén?
3
Feltételezve, hogy a ***fenti megoldás indokolt*** volt, ***mi történne 1:1 interleave létrehozása után***? 8 körbefordulásra volna szükség.



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

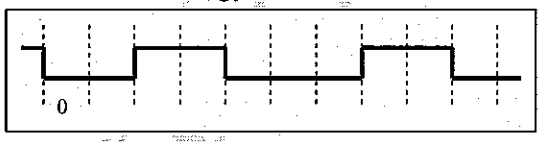
Rajzolja be a mellékelt ábrába egy mágneses tároló íróáram jelalakját a megadott bitsorozatra NRZI és PE kódolás esetén



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 adatmezejében van hibaellenőrző kód (CRC).	--
Adat írásakor a szektorazonosító mezejét 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.	--
A sáv és a szektor sorszámát.	x
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.



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

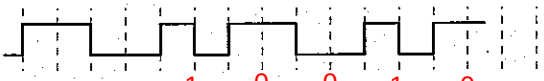
Írja be az üres kockákba a jelölt, első 0 értékű, bitet követő további 4 bit értékét!

0	0	0	1	1

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

Igen	NEM

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.



Bit **0 0**

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

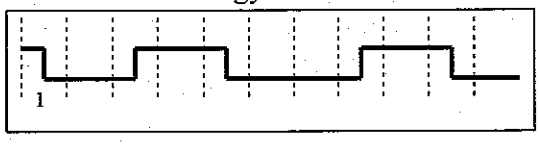
b/ Írja be az üres kockákba a jelölt első két 0 értékű bitet követő további 4 bit értékét!

0 0 [1][0][0][1]

c/

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

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]

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

1 [0][1][0][1]

A fenti három kódolási eljárás közül melyiknél a legnagyobb az egységnyi felületre eső fluxusváltozás? [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.

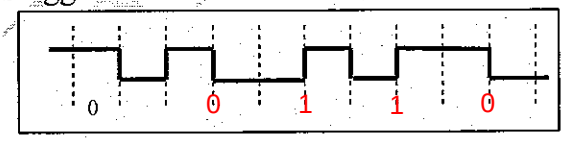
Bit **0 0**

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

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? [B]

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: [PE]

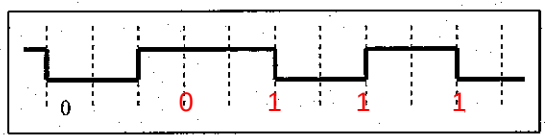
Írja be az üres kockákba a jelölt első 0 értékű bitet követő további 4 bit értékét!

0 [0][1][1][0]

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

[PE]

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.

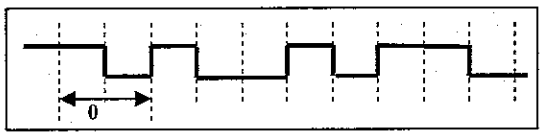


A felírásnál FM vagy MFM vagy PE kódolást alkalmazhattak. A jelalak alapján melyik kódolási eljárást használták a felírásnál? MFM

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

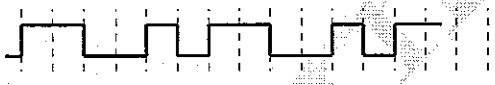
Írja be az üres kockákba a jelölt első 0 értékű bitet követő további 4 bit értékét!

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.

B



Bit **0 0**

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

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? B

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

Ismertesse mit jelent, hogy két esemény konkurens!

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!

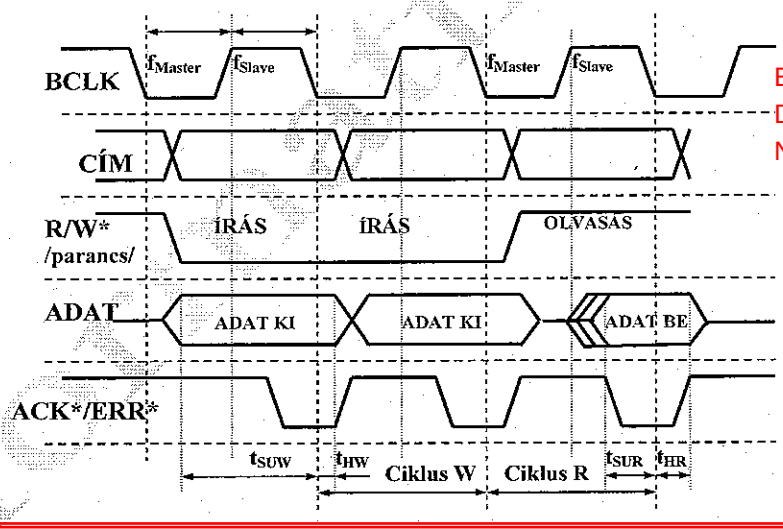
Sínrendszereknél mit jelent a geografikus címzés? Mondjon példát az alkalmazásra!

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

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

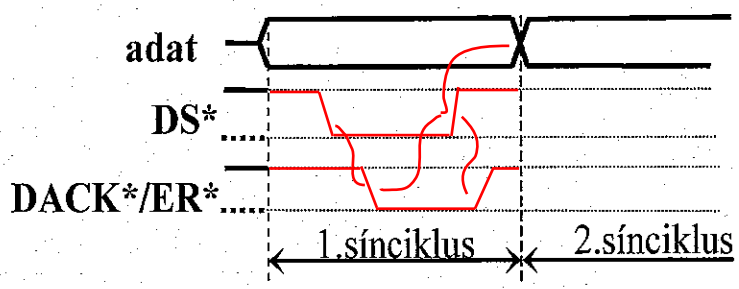
Milyen sínidőzítési protokoll látható az alábbi ábrán? Mik az ilyen protokoll előnyei és hátrányai?

Szinkron időzítésű adatátvitel

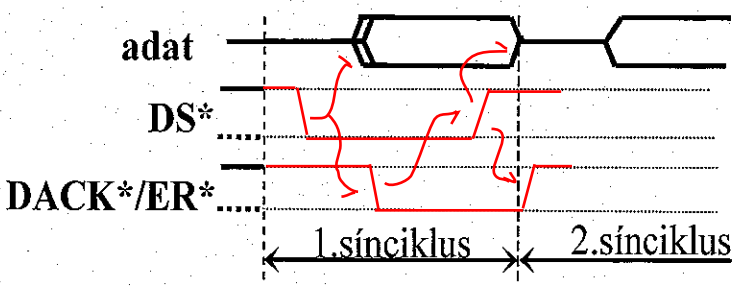


Elvileg leggyorsabb.
De leglassabb elem sebességével.
Nincs hibakezelés.

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

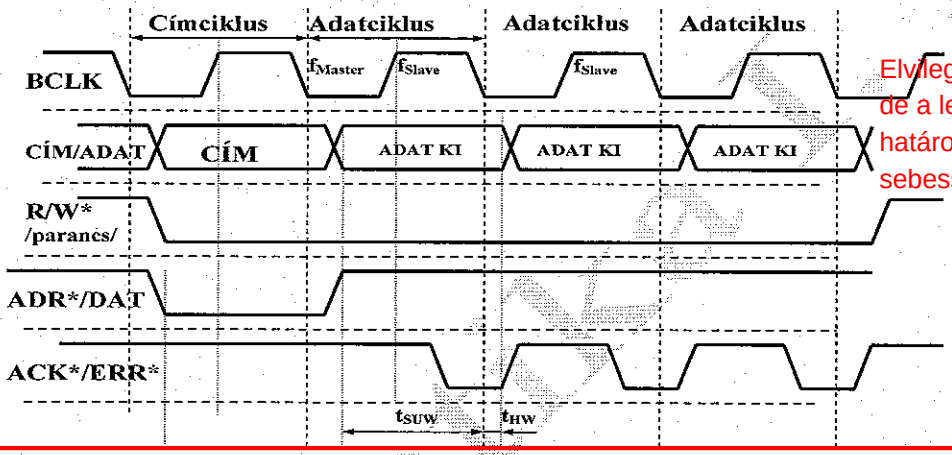


Rajzolja be az alábbi ábrába egy teljesen reteszelt protokollt alkalmazó sínrendszer adott jeleit és a jelátmenetek közötti, ok-okozati viszonyt mutató nyilakat olvasás esetén!



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

Szinkron időzítésű multiplexelt blokkos átvitel



Elvileg leggyorsabb, de a leglassabb elem határozza meg a sebességet.

Rajzoljon fel egy teljesen reteszelt (kapcsolt) szemiszinkron adatátviteli protokollt!
Milyen értékűek lehetnek egy ilyen protokoll időzítései? Órajel többszörösei.

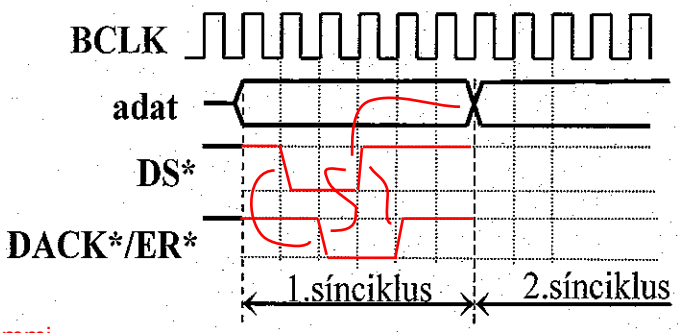
Teljesen kapcsolt protokollt alkalmazó szemiszinkron sínrendszer jelei láthatók az alábbi ábrán (az órajel 10MHz).

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

TWCmin=..... 300ns .....

Milyen hiba lép fel ilyen protokollnál nem létező címre íráskor...... A rendszer lefagy, mert várakozik és nem történik semmi. Időtúllépést ellenőrizni kell!

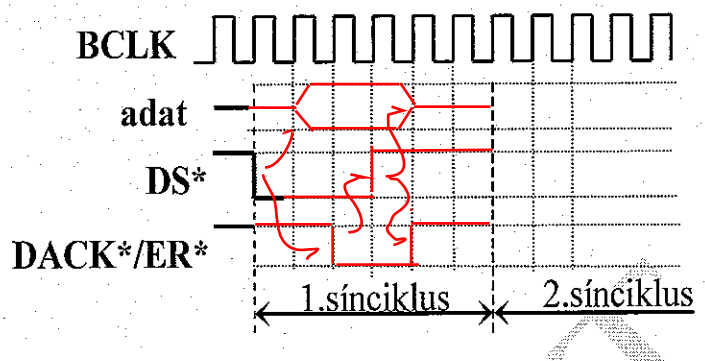
Rajzolja be a hiányzó jeleket és a jelátmenetek közötti, ok-okozati viszonyt mutató nyilakat írás esetén!



Teljesen ***reteszelt*** protokollt alkalmazó ***szemiszinkron*** sínrendszer jelei láthatók a mellékelt ábrán (az órajel **50MHz**).
Mekkora lehet egy olvasási sínciklusban a ***minimális elemi*** időzítés, egy ilyen rendszerben?
t_{min}=.....60.....ns

Milyen hiba lép fel ilyen protokollnál nem létező címről történő olvasás esetén?
Nem érkezik válasz, így a rendszer lefagy.

Rajzolja be a hiányzó ***jeleket*** és a jelátmenetek közötti, ***ok-okozati*** viszonyt mutató ***nyilakat olvasás*** esetén!



Adja meg, hogy az egyes jelek a forrástól (F) vagy a nyelőtől (N) származnak?

adat: F	DR: N	DA/DE: F

Az ilyen sín átviteli sebessége megegyezik a sínen lévő leglassúbb moduléval (1), a mester moduléval (2), az éppen aktív modulokéval (3), az éppen aktív modulok idejének egész számú órajelperiódusban kifejezett értékével (4)? 4

Milyen hiba léphet fel ***teljesen kapcsolt aszinkron*** protokollt használó rendszerben egy ***nemlétező*** címre történő íráskor illetve olvasáskor? ***Hogyan kezelik*** ezt a problémát?

Ismertesse a buszmegszerzési stratégiákat, mondjon példát, melyiket milyen esetben alkalmazzák!

Sorolja fel, és ***röviden ismertesse*** a buszelengedési stratégiákat!

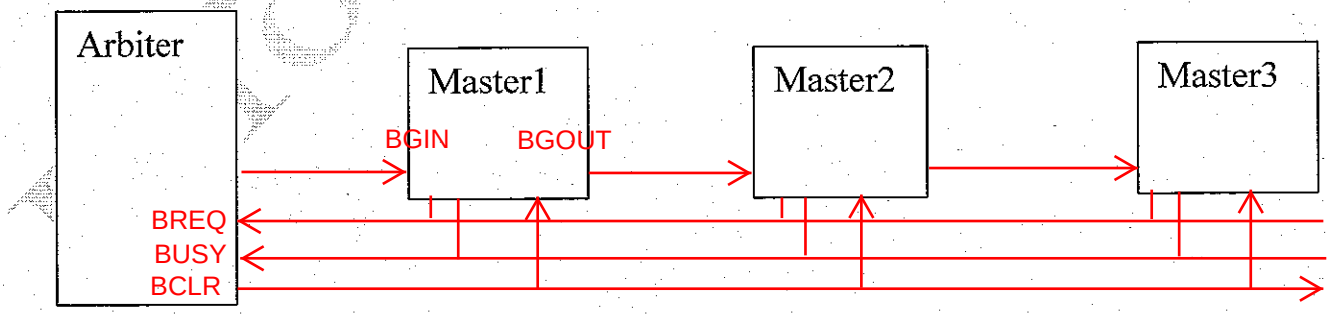
Multiprocesszoros rendszereknél a processzorokat ***statikusan***, vagy ***dinamikusan*** rendelhetik a feladatokhoz. Milyen buszmegszerzési stratégiát alkalmaznak az egyik, illetve a másik esetben? Röviden indokolja a választ!

Statikus hozzárendelésnél: Prioritásos ***Indoklás:*** A processzorok konkrét feladathoz vannak rendelve köztük gyakran állítható prioritási sorrend

Dinamikus hozzárendelésnél: Egyenlő esélyű ***Indoklás:*** Minden processzor egyenlő

Rajzolja fel a közös kérés –felfűzött válasz elvű arbitrációs mechanizmus vázlatát! Mi az előnye és mi a hátránya?

Rajzolja be egy közös kérés-felfűzött válasz (daisy chain) elvű sín-arbitrációs rendszerben használt jeleket és a jel funkciójára utaló elnevezésüket!



Milyen prioritási stratégia alakítható ki a fenti master-ek között Prioritásos

Rajzolja fel a VME sín kombinált arbitrációs rendszerének blokkvázlatát.

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

Arbiter: 1 darab ***Indoklás:*** az első kártyahelyen egyetlen arbitrációs áramkör dönti el, hogy ki kezelheti a sínt.

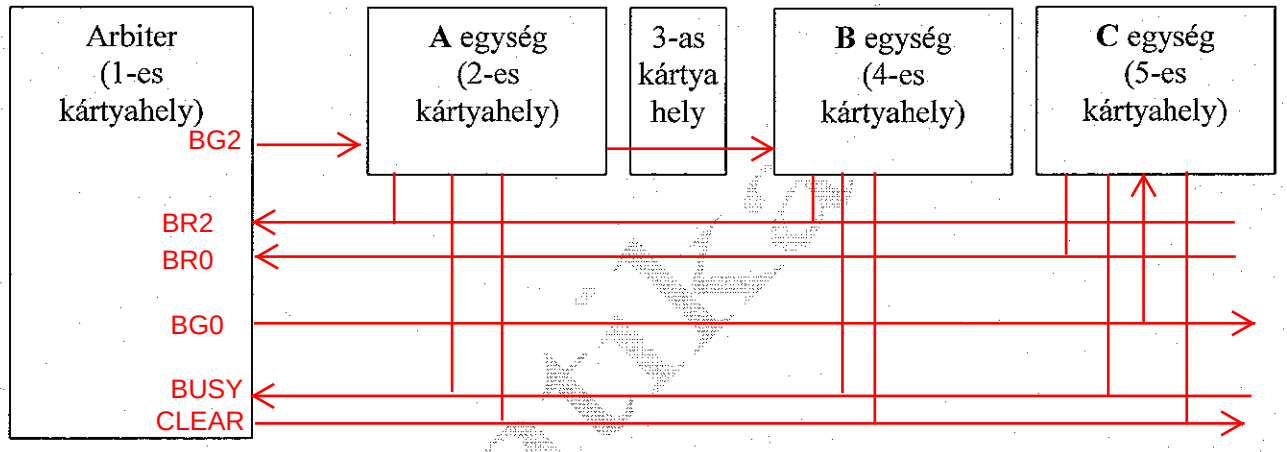
Mester: kártyahely - 1 ***Indoklás:*** Bármeddig lehet bővíteni minden stratégia esetén, csak bizonyos szám felett problémássá válik.

Milyen buszmegszerzési stratégiák valósíthatók meg ***VME*** rendszerben? (Indokolja a választ)

Rajzolja fel, a decentralizál, párhuzamos versenyeztetés elvű hardver mechanizmust használó arbiter blokkvázlatát!

VME rendszernél az **A** egység a **BR2***-őn, a **B** a **BR2***-őn, 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 rendszernél az **A** egység a **BR0***-án, a **B** a **BR2***-őn, a **C** egység a **BR2***-őn 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.

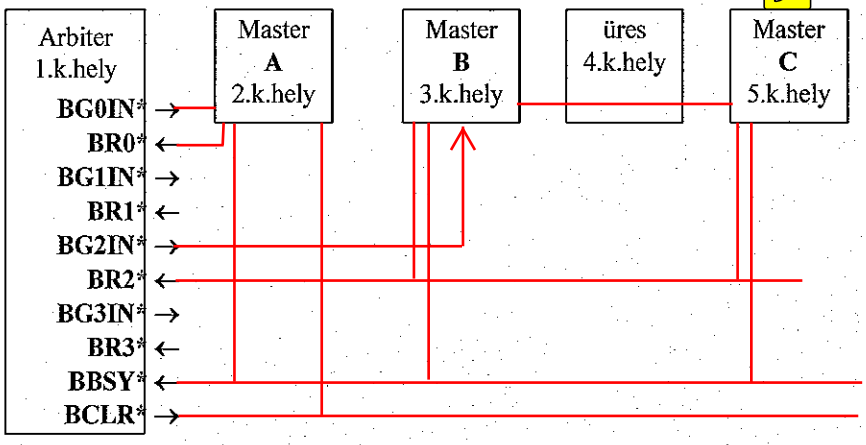
Milyen sorrendben kapják meg a buszvezérlési jogot ha a pillanatnyi prioritás, csökkenő prioritási sorrendben, BR2-BR1-BR0-BR3?

1	2	3
B	A	C

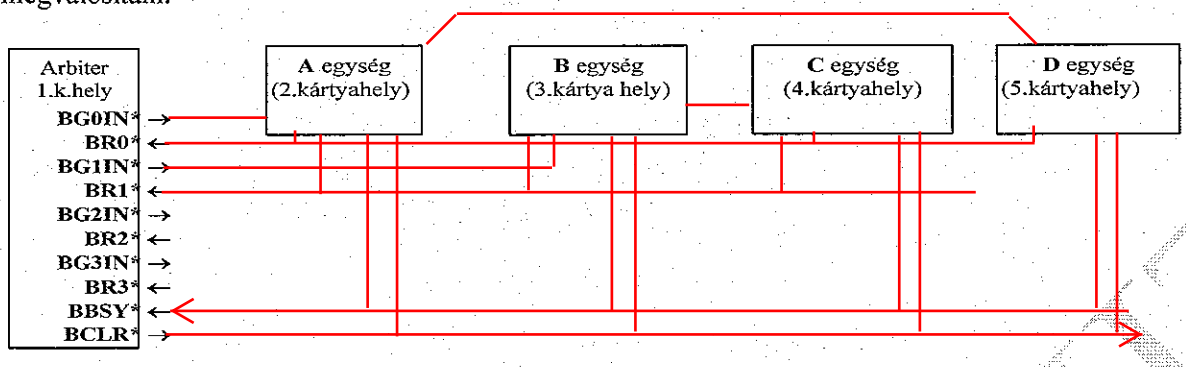
Mi a sorrend, ha a pillanatnyi prioritásnál BR0 a legmagasabb?

1	2	3
A	B	C

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



VME rendszernél 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 **Pre emption** buszelengedési stratégiát képes megvalósítani.



Milyen sorrendben kapják meg a buszvezérlési jogot?

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

Mi a sorrend ha az első egység adatátvitele alatta 6.kártya-helyen lévő **E** egység **BR3*-on** is kér buszvezérlési jogot?

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

Rajzolja fel a egy párhuzamos versenyeztetés elvű decentralizált arbitrációs rendszer blokkvázlatát, röviden ismertesse a működés elvét!

Multibus II rendszernél 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

AND

```
1 0 1 1 1 1        1 0 1 1 1 1
1 0 0 0 1 1        1 0 0 0 1 1
1 0 0 0 1 1        1 0 0 0 1 0
-----------        -----------
1 0 0 0 1 1        1 0 0 0 1 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?

1: C	2: B	3: A

Milyen buszmegszerzési stratégiát valósít meg a ***MBII*** a fenti megoldással?
Buszmegszerzési stratégia: Igazságos.
Hogyan biztosítja ezt a ***MBII*** ? Nem lehet bármikor belépni a döntési fázisba (kivéve 0-val kezdődő érték esetén).

Multibus II rendszernél 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

```
1 0 1 1 1 1        1 0 1 1 1 1
1 0 0 1 1 0        1 0 0 1 1 0
1 0 0 1 1 1        1 0 0 1 1 1
-----------        -----------
1 0 0 1 1 0        1 0 0 1 1 0
```

Első Mester: B

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

Buszmegszerzési stratégia: Igazságos (egyenlő esélyű).

Multibus II rendszernél az **A, B, C** mester az alábbi arbitrációs azonosító kódot adja a buszra:
Mutassa be, hogyan dől el, ki kapja meg elsőnek a buszvezérlési jogot

	ARB5*	ARB0*										
A	1	0	0	1	1	1		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		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		1	0	0	1	1	0		1	0	0	1	1	0

Az **ABC** bejelentkezése után három órajellel később a mellékelt arbitrációs kódú **D** és **E** egységnél is buszvezérlési igény keletkezik. Adja meg hogy az **ABCDE** egységek milyen sorrendben kapnak buszvezérlési jogot!

D 1 0 0 0 0 1
E 0 0 1 1 1 1

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

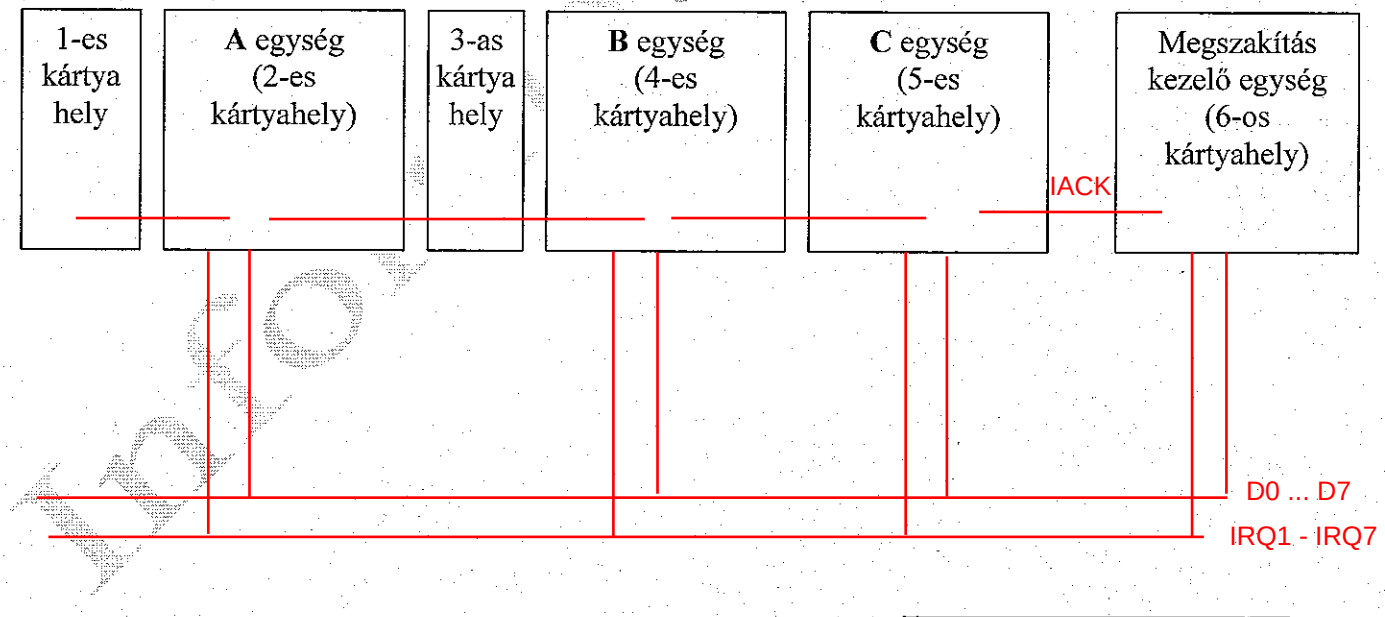
Rajzolja fel a láncolásos (dasy chain) bővítésű bus-vektoros megszakítási rendszer blokkvázlatát, magyarázza el a működését, sorolja fel az előnyeit és hátrányai!

Rajzolja fel a VME láncolásos (dasy chain) bővítésű bus-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 rendszernél 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

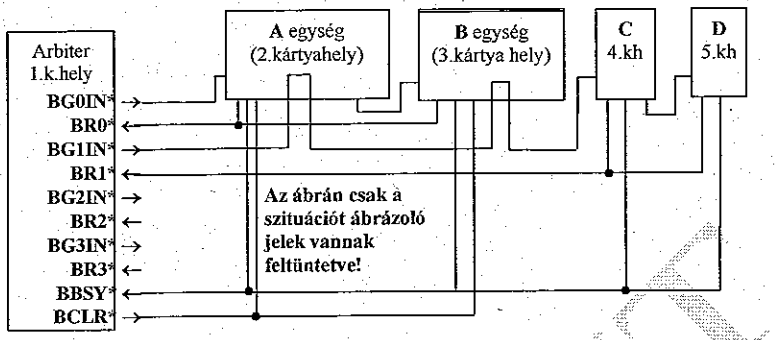
Mi a sorrend, ha az IRQ7*-es a legmagasabb prioritású szint és az elfogadott szint a 2-es?

1: A	2: C	3: B

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.



1: C	*2:* A	*3:* D	*4:* B

b) Mi a sorrend, ha a második egység kiszolgálása alatt az első mesternél újabb igény keletkezik?

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

c) Milyen buszmegszerzési stratégiát valósít meg a fenti rendszer?

Kombinált, a szintek közt round-robin, de szinten belül prioritásos. ***Indoklás:***

d) ***Hogyan*** oldható meg, hogy az ***A*** és ***B*** egység egyenlő esélyű buszmegszerzési stratégia szerint kerüljön kiszolgálásra?

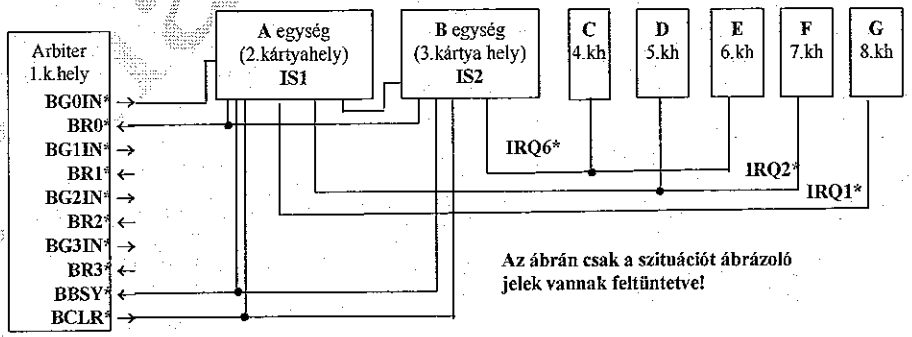
A B egység kérjen sínvezérlési jogot a BR2 vezetéken például.

e) Milyen ***buszelengedési stratégiát kell*** megvalósítania a ***C és a D jelű mesternek?***

RWD, mivel a többi kérő vonalat nem figyelik.

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. ***ISl*** 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.

1: D	*2:* F	*3:* G	*4* C	*5* E

Indoklás:

A prioritása nagyobb mint B-é, így előbb A megszakításai jutnak érvényre, majd a B megszakítása a felfűzésnek megfelelő prioritás szerint.

b) Mi a sorrend, ha csak egy ***IS*** kezelő van a rendszerben valamennyi IRQ vonal számára?

1: 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 bemeneteket és az ezekhez tartozó BG kimeneteket (ne pedig felfűzve legyenek).

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!