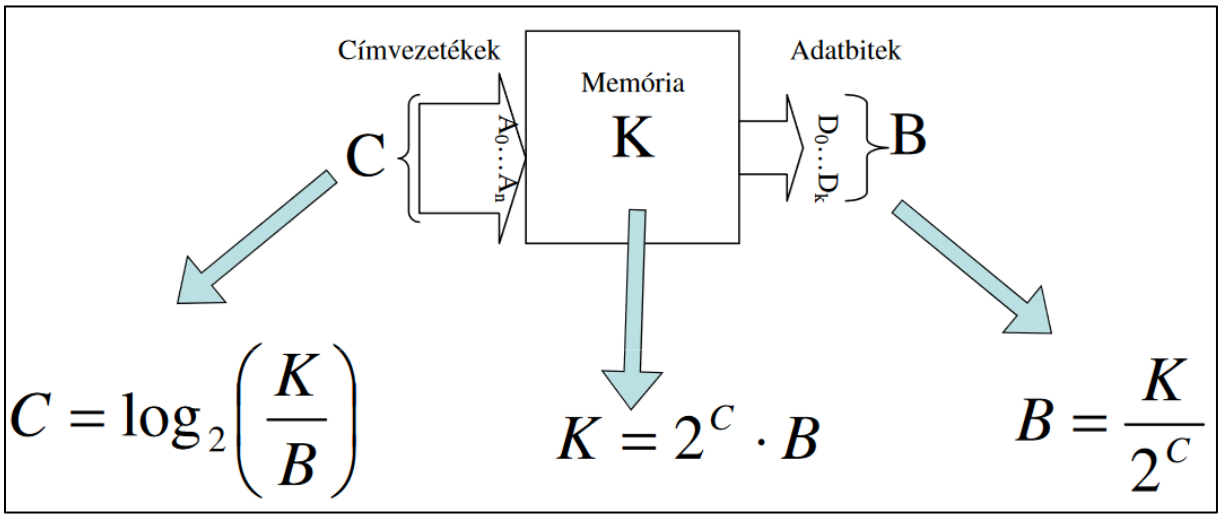
HWA- Útmutató SZGA-s feladatok megoldásához

1. típusú feladat: töltse ki a táblázat hiányzó adatait...

Ehhez egy képletet kell megtanulni, mely az előadásdián is megtalálható: $K=2^C*B$. a másik kettő képlet ennek a képletnek az átrendezéséből következik. Mivel a vizsgán nem lehet majd számológépet használni, ezért célszerű egy kicsit az ilyen számításokkal megbarátkozni.



A két megadott adatból a harmadik mindig kiszámítható. Érdemes 2-hatványos alakban dolgozni. Fontos, hogy ha meg van adva a K, és annak mértékegysége nem Byte, hanem KiB, vagy MiB, … akkor át kell váltani először a számot byte-ba, és aztán bit-be.

16. Töltse ki a táblázat hiányzó adatait

($A_0…A_n$) Címvezetékek	(C) Címbitek száma	(B) Adatbitek száma	($D_0…D_m$) Adatbitek	(K) Memória kapacitása [kbyte]
$A_0…A_{14}$	15	32	$D_0…D_{31}$	
		16	$D_0…D_{15}$	1024 kbyte
$A_0…A_{14}$	15			32 kbyte
		4	$D_0…D_3$	8 kbyte

A feladat első sorát így számolhatjuk ki: $K=2^{15}*32=2^{12}*2^5=2^{17}$ bit = $2^{17}/2^3=2^{14}$ byte = $2^{14}/2^{10} = 2^4$ KiB = **16 KiB**. **És látjuk, hogy rossz a mintafeladat megoldása.** (*Pilászy elfelejtette a bitet bájtba átváltani.*)

A második sorban a K = 1024 KiB = $2^{10}*2^{10}$ Byte=2^{20} Byte = 2^{23} bit. Innentől a C könnyen kiszámítható. Mivel B = 16, ami 2^4, így felírható, hogy $2^C=2^{19}$. Ebből következik, hogy **C értéke** a második sorban **csakis 19 lehet**. Mivel 19 címvezetékünk van, így A0...A18 kerül a második sor első oszlopába, mert az első címvezeték száma mindig 0 (Python lista feeling 😊). Itt már jó a hivatalos megoldás.

A harmadik sorban K= 32 KiB = 2^{15} Byte = $2^{15} * 2^3$ bit = 2^{18} bit. Így felírható a $2^{18}=2^{15}*B$.
Látjuk, hogy **$B = 2^3 = 8$ bit lesz**. Az adatbitek értelemszerűen D0…D7 lesznek.

A negyedik sorban K= 8 KiB = 2^{13} Byte = $2^{13} * 2^3$ bit = 2^{16} bit. Így felírható a $2^{16} = 2^C*4 =2^C*2^2$.
Látjuk, hogy $2^C = 2^{14}$ 8 bit lesz. Innen **C=14**. A címbitek értelemszerűen A0…A13 lesznek.

HWA- Útmutató SZGA-s feladatok megoldásához

2. típusú feladat: Cache és memória hozzáférési idők számolása

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

a) Számítsa ki mekkora a felhasználó által látott átlagos (látszólagos) memória hozzáférési idő (T_L), ha a blokk betöltés idejét elhanyagolhatjuk?

- T_L=0,9*10+0,1*50=9+5=14ns
- Tehát 90%, hogy 10 ns alatt, és 10 %, hogy 50 ns alatt érjük el a blokkot. A súlyozott átlagszámítással kapjuk meg a 14ns-t.

b) Számítsa ki mekkora a felhasználó által látott átlagos (látszólagos) memória hozzáférési idő (T_L), ha a blokk betöltési idő 100ns?

- T_L=0,9*10+0,1*100=9+10=19ns
- Tehát 90%, hogy 10 ns alatt, és 10 %, hogy 100 ns alatt érjük el a blokkot. A súlyozott átlagszámítással kapjuk meg a 19ns-t. A blokkbetöltés magával vonja a hozzáférést is, ezért azzal már nem kell itt számolnunk.

c) Mekkora a b) feladatban kiszámolt idő, ha a cache méretét megduplázzuk?

- Ha nem változik a blokkméret, akkor marad T_L = 19ns.
- Ha a blokkméretet is a 2-szeresére növeljük, akkor T_L=0,9*10+0,1*2*100=9+20=29ns

d) Mekkora a b) feladatban kiszámolt idő, ha a cache elérési idejét a felére csökkentik?

- T_L=0,9*5+0,1*100=10+4,5=14,5ns
- Csak a cache elérési ideje csökkent a felére (5ns), a memória elérési ideje maradt 100ns.

e) Mekkorára kell csökkenteni a cache elérési idejét a fenti (b) esetben, hogy a T_L a felére csökkenjen?

- A b feladat alapján most T_L=19/2=9,5 ns kellene.
- Ez nem valósítható meg, még akkor sem, ha a cache elérési idejét elhanyagoljuk, mert akkor is T_L=0,9*0+0,1*100=0+10=10ns lenne, ami nagyobb, mint a 9,5.

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

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

- T_L = 0.9 * 5ns + 0.1*50ns = 9.5ns
- És minden mást is úgy kell számolni, mint az előbb, az, hogy hány utas a szervezés, és mekkora a memória mérete, ezekben a feladatokban SOSEM számít!

HWA- Útmutató SZGA-s feladatok megoldásához

3. típusú feladat: direkt leképzés

Az operatív tárhoz egy direkt leképzésű cache kapcsolódik. A behozatali stratégia igény szerinti, az írási stratégia write-through **EZEK FELESLEGES INFÓK**. Az operatív tár byte-os szervezésű, a cím 32 bites. A teljes cache összesen 512 blokkot tárol, egy blokk 256 byteból áll.

- Az Offsetet az egy blokk méretéből számolhatjuk ki. Mivel ez most $256=2^8$, így az offset 8 bit lesz.
- A CBA-t, amennyiben a blokkszám van megadva, abból számolhatjuk ki. Mivel ez most $512=2^9$, így az CBA 9 bit lesz.
- A TAG a maradék, vagyis összesen 32 bites a cím, ebből le kell vonni 9+8=17-et. Így a TAG 15 bit lesz.

TAG (15 bit)	CBA (9 bit)	OFFSET (8 bit)
31 17	16 8	7 0

- Hány TAG-komparátort tartalmaz a cache?
  - Direkt leképzés esetén mindig 1 db TAG komparátor van.
- Hány bites a CBA (cache block address)?
  - 9 bites
- Hány bites a TAG komparátor?
  - 15 bites

Az operatív tárhoz egy direkt leképzésű cache kapcsolódik. A behozatali stratégia igény szerinti, az írási stratégia write- through **EZEK felesleges INFÓK**. 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?

- Direkt leképzés esetén ez az érték mindig egy.

Hány bites a TAG komparátor?

- Mivel egy blokk 256 byteból áll, és $256=2^8$, így 8 bites az Offset. A CBA adott, 9 bit. Mivel összesen 32 bites a cím, így 32-8-9=**15=TAG**

Számítsa ki a cache méretét (Kbyte-ban)?

- Mivel a CBA bitek száma 9, így 2^9=512 blokkot tartalmaz a cache. De nem ez volt a feladat.
- A cache méretét a CBA+OFFSET összegből lehet kiszámítani, ez 9+8=17 → A cache mérete így 2^{17} Byte=2^7 KiB = **128 KiB**

HWA- Útmutató SZGA-s feladatok megoldásához

Gyakorló feladatok – direkt leképzés

Direkt leképzés 1:

cím: 32 bites (ez a teljes hossz), CACHE: 128 kByte-os, 1 blokk mérete: 64 Byte

- 128 kByte: 17 bit → CBA+Offset (Byte vagy KiB esetén mindig CBA+Offset van)
  - ha Kbyte van megadva a cache mérete, akkor azt úgy kell számolni, hogy $2^{10}*2^{7}=2^{17}$
    - Hiszen a cache bájtos szervezésű és 1 KiB az 2^{10}-en Byte.
- 64 Byte: 6 bit → Offset (mert $2^{6}=64$)
- a TAG a maradék

TAG	CBA	OFFSET
31 17	16 6	5 0

1 db komparátor, mert direkt esetén mindig egy van csak

Direkt leképzés 2:

cím: 24 bites (ez a teljes hossz), 256 blokkot tartalmaz, 1 blokk mérete: 32 Byte

- 256: 8 bit → CBA (mert $2^{8}=256$) →Blokkszám megadása esetén csak a CBA bithosszát kell abból értelmezni.
- 32 Byte: 5 bit → Offset (mert $2^{5}=32$)
- a TAG a maradék

TAG	CBA	OFFSET
23 13	12 5	4 0

1 db komparátor, mert direkt esetén mindig egy van csak

HWA- Útmutató SZGA-s feladatok megoldásához

4. típusú feladat: Set-asszociatív cache

Sokféleképpen hivatkozhatunk rá *(ha n legalább 2)***:**
n utas asszociatív leképzés = n utas set-asszociatív leképzés = n utas direkt leképzés

---

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, bájtos szervezésű.

VALID BIT	TAG (16 bit)	CBA (10 bit)	OFFSET (6 bit)
	31 16	15 6	5 0

a) Hány bites az offset (eltolás)?

- Az eltolást a blokkméretből tudjuk kiszámolni, pont úgy, mint a direkt leképzés esetében: $64=2^6$, ezért az eltolás 6 bites lesz.

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

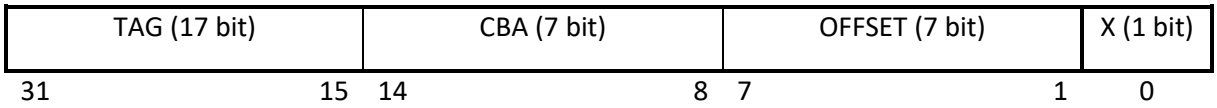
- A CBA 10 bit, mert a blokkszámot el kell osztani azzal, ahány utas a leképzés. Ebben a példában ez $4096/4=1024=2^{10}$. A CBA tehát esetünkben 10 bites.
- A TAG a maradék, ami 16 bit.
- Van még tovább 1 db jelzőbitünk (valid bit), ezt még balra beírjuk. Ez a memória címén kívül esik, abba nem számít bele.
- A szószám úgy számolható, hogy a blokkszámot el kell osztani azzal, ahány utas a leképzés. Ebben a példában ez 4096/4=1024.
- A bitszám egyenlő a TAG bitjei + a valid bittel, amit 16+1=17
- **A szószám*bitszám így 1024*17**

c) Hány TAG komparátort tartalmaz a cache?

- Mindig annyit, ahány utas a leképzés. Itt most 4 darabot.

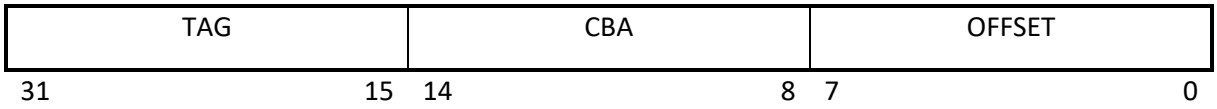
HWA- Útmutató SZGA-s feladatok megoldásához

Az operatív tárhoz egy 4 utas direkt leképezésű cache kapcsolódik. Az operatív tár 16bites szervezésű, a cím 32 bites. A teljes cache összesen 512 blokkot tárol, egy blokk 256 byteból áll.



- Hány bites az offset (eltolás)?
  - Mivel az operatív tár 16 bites, ami 2 bájtnak felel meg, így a 256-ot el kell osztani 2-vel. Ez $128=2^7$, így az **Offset 7 bit hosszú lesz**.
  - A fennmaradó **egy bit pedig üresen marad a cím legelején**. (Hiszen amúgy 8 bit hosszú lenne az Offset)
- Hány bites a CBA (cache block address)?
  - Itt már nem kell figyelembe venni az operatív tár szervezését, az előző feladatban látott módszer alapján a CBA kiszámolása a következőképpen történik: $512/4=128=2^7$. **Így a CBA 7 bites lesz.**
- Hány bites a TAG a cache-ben?
  - A TAG a maradék, 32-1-7-7=17 bites a TAG.

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?
  - A CBA az előbb látottak alapján 7 bites
  - Az Offset 8 bites, mert $256=2^8$.
  - **A TAG így 32-7-8=17 bites.**
- Hány bites a CBA (cache block address)?
  - Az előzőek szerint a blokkszám 512, amit el kell osztanunk az utak számával, ami 4. Így $512/4=128=2^7$ → vagyis **a CBA 7 bit hosszú**.
- Hány TAG-komparátort tartalmaz a cache?
  - 4-et, mert ennyi utas a leképzés

HWA- Útmutató SZGA-s feladatok megoldásához

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

VALID BIT	TAG (16 bit)	CBA (10 bit)	OFFSET (6 bit)
	31 16	15 6	5 0

- Hány bites az offset (eltolás)?
  - 6 bites
- Hány bites a CBA (blokkazonosító) mező?
  - 10 bites
- Hány TAG komparátort tartalmaz a teljes cache?
  - 4-et, mert ennyi utas a leképzés
- Hogyan változhatna a találati arány, ha a fenti cache-ben két utas direkt leképzést alkalmaznánk? Indokolja a választ!
  - Csökkenne, mert merevebb lenne a betöltés, messzebb kerülnénk a teljesen asszociatívtól.
  - A Komparátorok száma 4-ről 2- re csökkenne
  - Az egyes komparátorok bitszélessége 16 bitről 15 bitre bitre csökkenne

---

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 írási stratégiát alkalmaz. A számítógép címsínje 32 bites, az operatív memória hozzáférési ideje 75 ns, a találati arány HR=92%)

VALID BIT	TAG (17 bit)	CBA (7 bit)	OFFSET (8 bit)
	31 15	14 8	7 0

- Hány bites az offset (eltolás)?
  - 8 bites, mert a blokkméret alapján $256=2^8$.
- Milyen szervezésű (szószám x bitszám) egy vezérlő (TAG) tár?
  - A CBA kiszámítása itt egy kicsit nehezebb lesz. Ez meghatározható a gyorsítótár mérete / (utak száma * blokkméret) képletből. Meg kell jegyezni, hogy ez a képlet a direkt leképzés esetében is működne, ott az utak száma 1 lenne.
  - Tehát a CBA kiszámítása: 128Kbyte = 2^7 Kbyte =2^{17} Byte. Ezután 2^{17} / (4*256) = $2^{17}/2^{10}=2^7$ → **a CBA 7 bites lesz.**
- Mennyi a memória látszólagos átlagos elérési ideje?
  - A korábban látottakhoz képest ugyanúgy számoljuk
  - 0,92*10+0,08*75 = 15,2 ns

HWA- Útmutató SZGA-s feladatok megoldásához

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

VALID BIT	TAG (15 bit)	CBA (10 bit)	OFFSET (7 bit)
	31 17	16 7	6 0

- Hány bites az offset (eltolás)?
  - 7 bites, mert $128=2^7$.
- Milyen szervezésű (szószám x bitszám) egy vezérlő (TAG) tár?
  - A CBA meghatározása a szokásos módon történik: 2048/2=1024. 1024=210. Ezért a CBA bitszáma 10.
  - A TAG 15 bites, mert 32=7+10+15.
  - A szószám = 2048/2=1024 (blokkok száma / utak száma)
  - A bitszám= TAG+VALID=15+1=16
  - **Tehát a szószám * bitszám = 1024 * 16**
- Hány TAG komparátort tartalmaz a cache?
  - Kettőt, mert ennyi utas a leképzés

Gyakorló feladatok – set-asszociatív leképzés

Set- asszociatív leképzés 1.:

2 utas leképzés, cím: 24 bites, 256 blokkot tartalmaz, 1 blokk mérete: 32 Byte

- 256/2=128: 7 bit → CBA
- 32 Byte: 5 bit → Offset (egy)

TAG	CBA	OFFSET
23 12	11 5	4 0

2 db komparátor, mert ennyi utas a leképzés

Set- asszociatív leképzés 2.:

4 utas leképzés, cím: 32 bites, 1024 blokkot tartalmaz, 1 blokk mérete: 256 Byte

- 1024/4=256: 8 bit → CBA
- 256 Byte: 8 bit → Offset

TAG	CBA	OFFSET
31 16	15 8	7 0

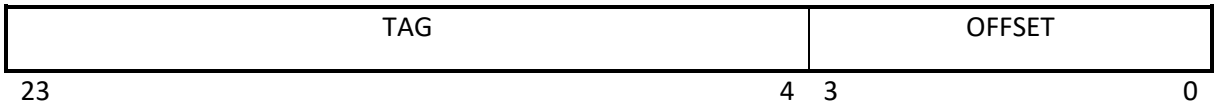
4 db komparátor, mert ennyi utas a leképzés

HWA- Útmutató SZGA-s feladatok megoldásához

5. Típusú feladat: teljesen asszociatív leképzés

Az operatív tárhoz egy teljesen asszociatív cache kapcsolódik. Az operatív tár byteos szervezésű, a cím 24 bites. A cache 128 blokkot tárol, egy blokk 16 byteból áll.

Teljesen asszociatív leképzés esetében nincsen CBA, csak Offset és TAG. Az offsetet ugyanúgy számoljuk, mint eddig, a TAG a maradék. $16=2^4$. Vagyis az Offset 4 bit.



a) Hány bites a TAG a cache-ben?

- 24-4=20 bit

b) Hány TAG-komparátort tartalmaz a cache?

- Teljesen asszociatív leképzés esetén mindig annyi, ahány blokkot tárolunk, jelen esetben ez 128.

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

Hány bites az offset (eltolás)?

- A szokásos módon számítjuk: $128=2^7$. Így az Offset 7 bit.

Milyen szervezésű (szószám x bitszám) a vezérlőtár?

- A TAG 32-7=25 bites.
- A szavak számát a szokott módon számíthatjuk ki, ne zavarjon meg minket, hogy most nincsen CBA. (256*1024) / 128=2048. (a cache méretét Byte-ban megadva kell elosztani a blokkmérettel, hiszen az utak száma itt megegyezik a blokkmérettel, lásd a TAG komparátorok számát).
- A bitek száma a már ismert módon 25+1=26 bit.
- **A szószám * bitszám érték így 2048*26 bit.**

Mennyi a memória átlagos elérési ideje?

- A már jól ismert módon számolható ki.
- 0.92*10+0.08*75=15.2 ns

Mivel ezek a számpéldák viszonylag egyszerűek, így itt most nem lesznek gyakorló feladatok.

HWA- Útmutató SZGA-s feladatok megoldásához

6. Típusú feladat: Lapcsere-stratégiák: Real-LRU, Pseudo-LRU és FIFO

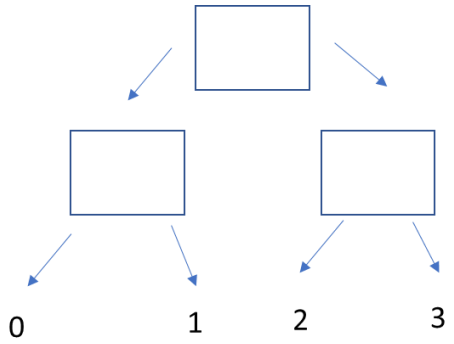
Real-LRU működése:

Rajzoljunk minden egyes hivatkozáshoz egy annyi sorból és oszlopból álló négyzetes táblázatot, ahány utas a cache. Ha négy utas, akkor minden hivatkozáshoz rajzoljuk egy 4*4-es táblázatot. Az egyes oszlopokat számozzuk be balról jobbra, 0…n-1, ahol n az utak száma. Az egyes sorokat pedig számozzuk be alulról felfelé, 0…n-1, ahol n az utak száma. Ha ez megvan kezdjük őket kitölteni. Ahol kezdetben nincsen semmi, oda képzeljük 0-s értékeket.

1. Írjunk a megfelelő sorszámú sorba (amelyikre a hivatkozás történt) mindenhova 1-et, felülírjuk, ha már van ott valami.
2. Írjunk a megfelelő sorszámú oszlopba (amelyikre a hivatkozás történt) mindenhova 0-át, felülírjuk, ha már van ott valami (akár egy 1-es az előbbi lépésből).
3. Az a sorszámú lap (ha nem egyértelmű, akkor lapok) lesz az áldozat (LRU), amelyiknek a sorában (alulról felfele) csupa nulla érték van.
4. Lépjünk a következő hivatkozásra, és vissza az 1. pontra, a kiindulási alap táblázat, az imént kitöltött táblázatunk lesz.

Pseudo-LRU működése

Rajzoljunk egy olyan kiegyensúlyozott bináris fát, melyen minden csomópontnak két-két gyereke van. Ezt addig csináljuk, amig annyi levél nem lesz, ahány utas a leképzés. Pl négy esetében elég három csomópont, ekkor pontosan négy levél lesz. Az ábra ezt mutatja. Az ilyen bináris fából annyit rajzoljunk, ahány hivatkozás van.



Töltsük ki a hiányzó cellákat a következőképp: Ha balra kell mennünk, írjunk az adott cellába 1-est. Ha jobbra kell menni, írjunk az adott cellába 0-ást. Pl: az első hivatkozás a 0. Ekkor a felső oszlopba egy kerül, mert balra szeretnénk menni, ezt követően a bal alsó oszlopba ismét egy kerül, mert megint balra kell lépnünk ahhoz, hogy 0-s hivatkozáshoz jussunk. A másik alsó oszlop értéke nem változik.

Amikor a lépéseknél azt szeretnénk meghatározni, hogy ki legyen az áldozat, azt úgy tehetjük meg, hogy a fa tetejéről indulunk, és ahol 0 van, ott balra lépünk, ahol egy ott pedig jobbra. Vigyázat, ez az előzőnek pont a fordítottja! Amelyik levélre érkezünk, az lesz az áldozat. Ez az első esetben a 2, mert az 1-esnél jobbra lépünk, majd egy üres cellával találjuk szembe magunkat, amiről az elején feltételeztük, hogy annak értéke 0, így ott balra megyünk tovább.

A következő lépésben a kiindulási alap az imént kitöltött cellák lesznek, tehát lesz kettő, ami felülíródik (de az értéke nem feltétlenül változik), és lesz egy, aminek semmiképpen sem változik majd az értéke.

HWA- Útmutató SZGA-s feladatok megoldásához

FIFO működése

Minden egyes hivatkozáshoz rajzoljunk egy oszlopot, melynek annyi cellája (sora) van, ahány utas a leképzés.

Írjuk be az egyes számokat alulról felfelé. Ha egy szám benne van már a FIFO-ban, ne csináljunk semmit. Ha a szabad helyek elfogytak, de új szám érkezik, akkor arra a helyre írjuk be a számot, ahová a legelső került (tehát a legalsó, majd minden egyes felülírásnál eggyel feljebbi cellát válasszunk, ha pedig a legfelső cellát is felülírtuk, kezdjük megint alulról).

A Hr-t (HIT RATE) úgy kell számolni, hogy azokat az eseteket, amikor a szám már a FIFO-ban volt, és nem csináltunk semmit megszámoljuk, és elosztjuk az összes hivatkozás számával. Itt a kezdetben üres cellákat üresnek vesszük, és nem 0-nak.

LRU tár működése

Minden egyes hivatkozáshoz rajzoljunk egy oszlopot, melynek annyi cellája (sora) van, ahány utas a leképzés.

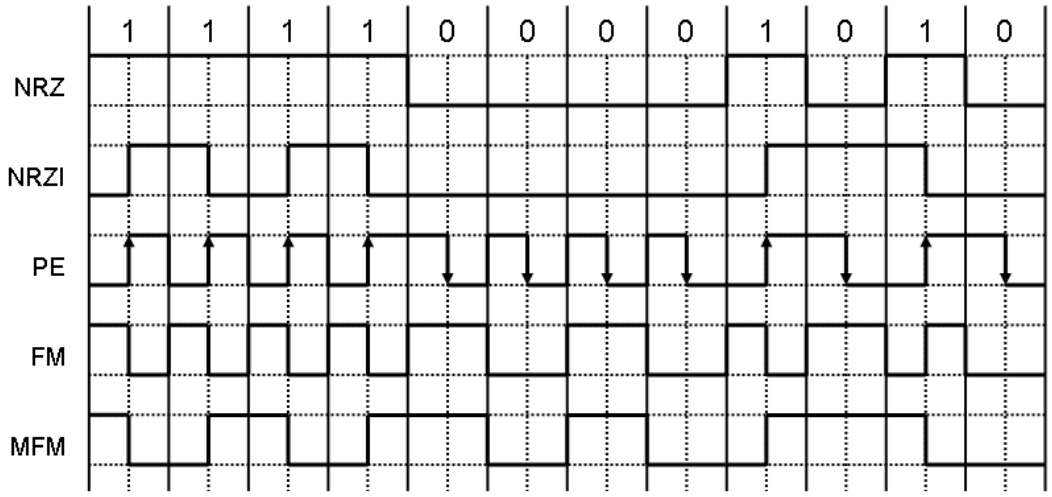
Írjuk be az egyes számokat alulról felfelé. Ha egy szám benne van már az LRU tárban, ne írjuk be máshová. Ha a szabad helyek elfogytak, de új szám érkezik, akkor arra a helyre írjuk be a számot, ahová a legrégebben írtunk utoljára (FOTOS: ha egy cellában van egy adat, és aztán később újra ugyanaz az adat jön, akkor a cella „kora” visszaáll 0-ra, vagyis úgy tűnik, hogyha felülírtuk volna az adatot, miközben az ilyen találat beszámít a Hit Rate-be is).

A Hr-t (HIT RATE) úgy kell számolni, hogy azokat az eseteket, amikor a szám már az LRU tárban volt, és nem cseréltük ki egy másikra, megszámoljuk, és elosztjuk az összes hivatkozás számával. Itt a kezdetben üres cellákat üresnek vesszük, és nem 0-nak.

7. Típusú feladat: Merevelemez-kódolások

- NRZ: követi a bejövő biteket, a legegyszerűbb
- NRZI
  - Ha 1-es jön, akkor a cella közepén invertálok
  - Ha 0-s jön, akkor nem csinálok semmit
- PE
  - Ha 1-es jön, a cella közepén felfutó élet rajzolok
  - Ha 0-s jön, a cella közepén lefutó élet rajzolok
  - Ha ugyanolyan bit után ugyanolyan jön, akkor a két cella közötti határon is invertálok
- FM
  - Ha 0 jön, csak a (kezdeti) cellahatáron invertálok
  - Ha 1 jön, a (kezdeti) cellahatáron és középen is invertálok
- MFM
  - Ha 0 jön, akkor a (kezdeti) cellahatáron van az invertálás
  - Ha 1 jön, akkor a cella közepén van az invertálás
  - DE: Ha 0 után 1 jön, vagy ha 1 után 0 jön, akkor a (kezdeti) cellahatáron nincsen váltás.
- RLL 2,7 kódolás esetén lesz egy megadott segédtáblázat. Itt meg kell keresni sorban, az adat éppen aktuális részére leghosszabban illeszkedő kódsort, és az abból kapott kódot úgy kell használni, hogy minden bitet beírunk sorban minden cellaközbe (egy bithez két érték fog kerülni, mert a cella középen is van egy szaggatott vonal). Ahol az érték 1, ott invertáljuk a jelet, ahol az érték 0, ott nem csinálunk semmit.

Gyakorló feladat:



8. Típusú feladat: kiskérdések

Cache - A set-asszociatív cache szervezésnél LRU, LFU, FIFO, RANDOM blokkcsere stratégiákat alkalmazhatnak. Mit jelent az LFU stratégia?

- A legritkábban használt blokk helyére hozzuk be az új blokkot.

Cache -Használható-e LRU (vagy bármilyen) blokkcsere stratégia direkt leképzésű cache-nél? Indokolja a választ!

- Nem, mert minden blokk csak a megadott helyre hozható be, nincs blokkcsere algoritmus.

HDD - Melyik kódolással és mennyivel több adatot rögzíthetnek azonos fluxusváltási sűrűséget eltételezve?

- MFM 2X hatékonyabb, mint az FM
- RLL2,7 1,5X hatékonyabb, mint az FM
- RLL2,7 3X hatékonyabb, mint az FM

Sorolja fel a Cache blokk behozatali stratégiákat

- Igény szerinti, előrelátó, szelektív

Sorolja fel a Cache blokk írási stratégiákat.

- Write back, write trough

Sorolja fel a Cache blokkcsere stratégiákat.

- LRU, LFU, FIFO, RANDOM

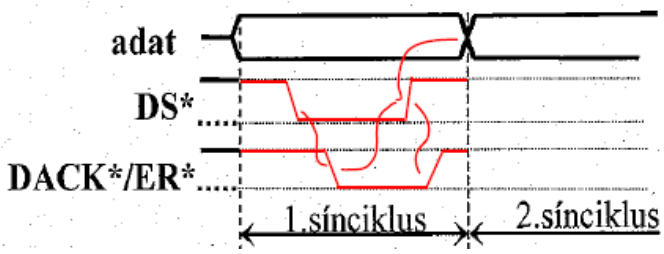
Milyen műszaki megoldást alkalmaznak a mai korszerű memória áramkörök a blokkbetöltés gyorsítására?

- Átlapolt memóriaműködési eljárásokat (FPM, EDO, BEDO, SDRAM)

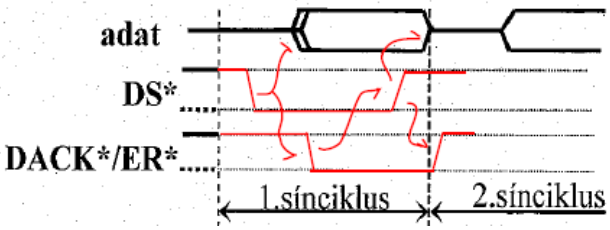
HWA- Útmutató SZGA-s feladatok megoldásához

9. Típusú feladat: Sínrendszerek szerepe

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!



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

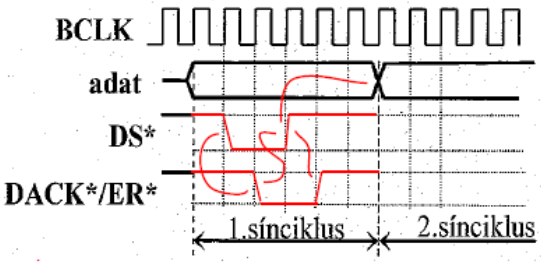
Teljesen kapcsolt protokollt alkalmazó szemiszinkron sínrendszer jelei láthatók az alábbi ábrán (az órajel 10MHz).

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!

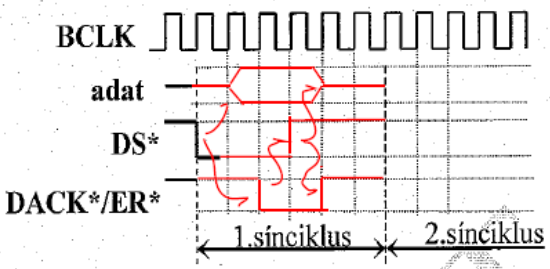
Rajzolja be a hiányzó jeleket és a jelátmenetek közötti, ok-okozati viszonyt mutató nyilakat írás esetén!



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