

1. Feladat Vegyünk egy 256 byte méretű cache-t 64 byte-os blokkmérettel. A cache kezdetben csupa érvénytelen blokkot tartalmaz. Egy program az alábbi memóriablokkokról olvas (ebben a sorrendben):

- 1, 3, 8, 4, 3, 6, 8, 1

Adja meg a cache hibák számát és a cache végső tartalmát (blokkszámokkal) LRU algoritmus mellett

- direkt leképzés esetén,
- teljesen asszociatív szervezés esetén,
- két utas asszociatív szervezés esetén.

2. Feladat Vegyünk egy 32 kB méretű cache-t 64 byte-os blokkmérettel. A CPU rendelkezzen 32 bites fizikai, 48 bites virtuális címeikkel. 4-utas asszociatív szervezés mellett hány darab és hány bites összehasonlítást kell végezni egy keresés során, ha

- Fizikailag indexelt cache-t használunk, fizikai tag-ekkel
- Virtuálisan indexelt cache-t használunk, virtuális tag-ekkel

3. Feladat Vegyünk egy 512 byte méretű cache-t 64 byte-os blokkmérettel. A cache fizikai címeket használ mind az indexeléshez, mind a tag-ekhez. A fizikai címek 16 bit szélesek. Egy program az alábbi memóriacímekről olvas (ebben a sorrendben):

- 13, 136, 490, 541, 670, 74, 581, 980

(a) Adja meg a fenti címekhez tartozó "tag", "index" és "eltolás" mezőket

- teljesen asszociatív szervezés
- direkt leképzés
- két utas asszociatív szervezés

esetén.

(b) Adja meg a cache végső tartalmát (blokkszámokkal) mind a 3 esetben, LRU algoritmus mellett! (A cache kezdetben csupa érvénytelen blokkot tartalmaz)

4. Feladat Vegyünk egy 1 kB méretű cache-t 64 byte-os blokkmérettel. A cache kezdetben csupa érvénytelen blokkot tartalmaz. Lefuttatjuk az alábbi programot:

```
short int t[32][32];
register int sum = 0;

for (int i=0; i<32; i++)
    for (int j=0; j<32; j++)
        sum += t[i][j];
```

Feltételezések: a `short int` 2 byte-os, a `t` tömb blokkhatáron kezdődik, a két dimenziós tömb a memóriában sor-folytonosan helyezkedik el, a cache direkt leképzést használ. Az `i,j` változó regiszterekben van tárolva, a cache-t nem terhelik.

- Hány cache hibát vált ki a fenti algoritmus? Számolja ki a cache hiba-arányt!
- Hány cache hibát vált ki a fenti algoritmus, ha megcseréljük a két ciklust? Számolja ki a cache hiba-arányt!
- Mekkora cache-re lenne szükség, hogy a megcserélt for ciklusokkal is ugyanolyan hibaarányt kapjunk, mint az eredeti algoritmussal?