

1. Feladat Egy számítógép rendszermemóriája egycsatornás, 64 bites adategységekkel rendelkező DDR3-2000 DRAM-ra épül, melyben a burst méret 8 adategységnyire van beállítva.

A több bank-os és több rank-os felépítéstől most tekintsünk el.

A memória késleltetések legyenek az alábbiak, **külső** órajelek számában megadva:

- $T_{RP} = 12$ (a PRECHARGE parancs végrehajtásához szükséges idő)
- $T_{RCD} = 12$ (ennyi ideig tart egy sor megnyitása)
- $T_{CAS} = 10$ (egy nyitott sor egy oszlopának a kiolvasásához szükséges idő. Az olvasás parancs után ennyi idő múlva jelenik meg az *első* adat a modul adatbuszán)

A memóriavezérlőhöz beérkező 64 bájtos (tehát 1 burst-nyi) olvasási kérések az alábbi sor, oszlop koordinátákra vonatkoznak:

- (3. sor, 8. oszlop), (3. sor, 2. oszlop), (7. sor, 9. oszlop)

Kezdetben a 7. sor van nyitott állapotban. Az utolsó parancs után a memóriavezérlő ne zárja le a nyitott sort.

- Adja meg a fenti kérésekhez tartozó, memóriavezérlő által kiadott DRAM parancsokat (sorrendhelyesen), FCFS, valamint FR-FCFS ütemezés mellett! (Feltesszük, hogy a 64 bájtos kérések nem lógnak túl az adott soron.)
- Az FCFS ütemezést alapul véve hányadik (külső) órajel múlva jelenik meg a (3. sor, 8. oszlop) kérésre érkező első adat a memóriamodul adatbuszán? Ez mennyi ns-ban mérve? És a (3. sor, 2. oszlop) kérésre érkező első adat?

2. Feladat Legyenek a virtuális címek 16 bitesek, a fizikai címek 15 bitesek, a lapméret legyen 2^{12} bájt = 4 kB méretű, a laptábla pedig legyen egyszerű egyszintű laptábla 8 bites bejegyzésekkel.

- A virtuális címekben hány bit tartozik a lapok azonosításához és hány a lapon belüli eltoláshoz? Rajzolja fel a virtuális címek tagozódását!

- Mekkora a teljes laptábla mérete?

- Hány lap fér a fizikai memóriába?

- A laptábla aktuális állapota legyen az ábra szerinti.

Hol található a 3-mas, az 6-os és a 11-es lap?

- Módosítsa a laptábla tartalmát a következők szerint:

- Hozzuk be a 11-es lapot a fizikai memóriába a 6-os lap helyére!
- Hozzuk be a 2-es lapot a fizikai memóriába a 8-as lap helyére!

- Feltéve, hogy nincs laphiba, hány memóriaművelet kell a címfordításhoz

- TLB találat esetén?
- TLB hiba esetén?

	Valid:	Keret:
Laptábla: 0:	1	5
1:	1	7
2:	0	?
3:	1	2
4:	0	?
5:	0	?
6:	1	1
7:	0	?
8:	1	6
9:	0	?
10:	0	?
11:	0	?
12:	1	0
13:	1	3
14:	0	?
15:	1	4

3. Feladat Egy virtuális tárkezelésre képes processzor 14 bites virtuális és 13 bites fizikai címeket támogat. A lapméret 1024 bájt ($=2^{10}$). A címfordításhoz kétszintű laptáblát használ, minden szintre egyforma széles indexekkel, 16 bites bejegyzésekkel. A címfordítás gyorsítására egy 4 bejegyzéses, LRU algoritmussal menedzselte teljesen asszociatív TLB-t tartalmaz.

- A virtuális címekben hány bit tartozik a lapon belüli eltoláshoz és hány a lapok azonosításához? Rajzolja fel a virtuális címek tagozódását!

- Hány lap fér a fizikai memóriába?

- Mekkora az összes laptábla összegzett mérete?

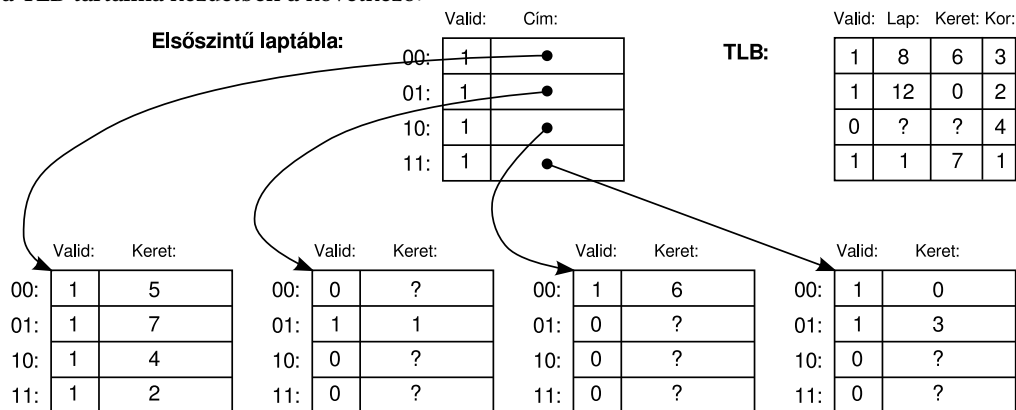
- Optimális körülmények között mennyi a laptáblák által elfoglalt memória minimális mérete? Mennyivel kisebb ez, mintha egyszintű laptáblát használnánk?

- Feltéve, hogy nincs laphiba, hány memóriaművelet kell a címfordításhoz TLB hiba, ill. TLB találat esetén?

(f) A futó program sorban egymás után az alábbi lapokra hivatkozik:

- 13, 12, 7, 2, 13.

A laptábla és a TLB tartalma kezdetben a következő:



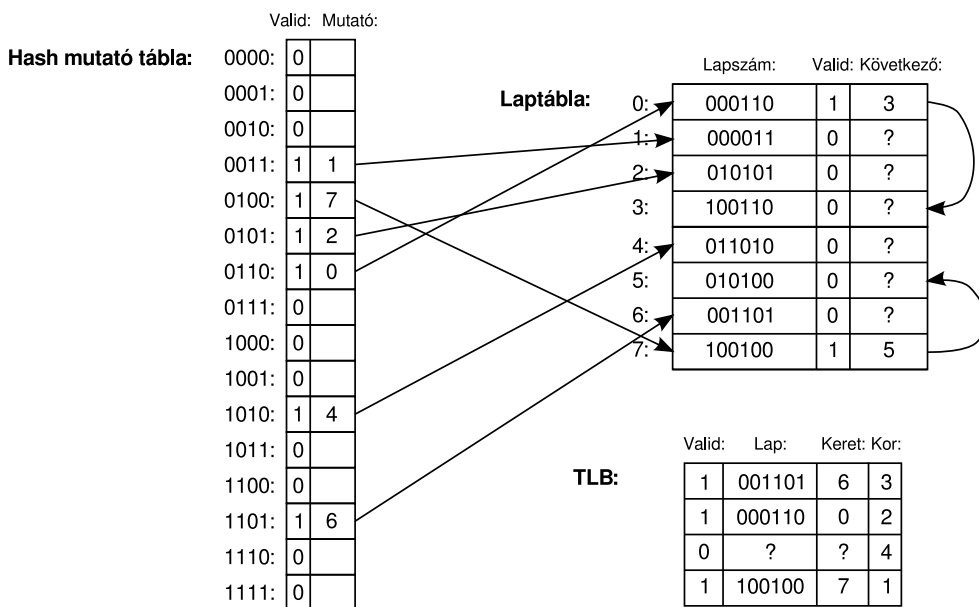
- Adja meg a laptábla és a TLB tartalmát a fenti laphivatkozások után! Ha az operációs rendszer egy újabb lapot akar a fizikai memóriában elhelyezni, akkor egy bent lévő lapot ki kell vennie onnan. A soron következő kiszemelt áldozatok legyenek az 1-es, majd az 5-ös lapok. Lapcsere esetén az operációs rendszer a TLB teljes tartalmát érvényteleníti.
- A megadott hivatkozásokból hány esetén volt TLB találat?
- A megadott hivatkozásokból hány esetén volt laphiba?

4. Feladat Egy virtuális tárkezelésre képes processzor 16 bites virtuális és 13 bites fizikai címeket támogat. A lapméret 1024 bájttal ($=2^{10}$). A címfordításhoz inverz laptáblát használ, melyben a hash függvény értéke a virtuális cím 10.-13. bitje által meghatározott szám. A laptáblabejegyzések mérete 16 bit. A TLB 4 bejegyzéses, teljesen asszociatív, LRU algoritmussal menedzsel.

- Rajzolja fel a virtuális címek tagozódását!
- Hány lap fér a fizikai memóriába?
- Mekkora a teljes laptábla, és mekkora a hash mutató tábla mérete? Mennyivel kisebb ez, mintha egyszintű laptáblát használnánk?
- A futó program sorban egymás után az alábbi lapokra hivatkozik:

- 010100, 000110, 110100, 000011, 010100.

A laptábla és a TLB tartalma kezdetben a következő:



- Adja meg a hash mutató tábla, a laptábla és a TLB tartalmát a fenti laphivatkozások után! Ha az operációs rendszer egy újabb lapot akar a fizikai memóriában elhelyezni, akkor egy bent lévő lapot ki kell vennie onnan. A soron következő kiszemelt áldozatok legyenek az 100110-ás, majd az 100100-ás lapok. Lapcsere esetén, ha a memóriából kikerült laphoz tartozik TLB bejegyzés, akkor az operációs rendszer azt az egy TLB bejegyzést érvényteleníti.
- A megadott hivatkozásokból hány esetén volt TLB találat?
- A megadott hivatkozásokból hány esetén volt laphiba?

(e) Feltéve, hogy nincs laphiba, hány memóriaművelet kell a legjobb és a legrosszabb esetben a címfordításhoz TLB találat, illetve TLB hiba esetén (a (d) feladattól függetlenül)? Mikor áll elő a legrosszabb eset?