

Припрема за трећу лабораторијску вежбу

- Архитектура и организација рачунара 2 –

Главни програм симулатора се налази у TestCache.java у коме се креира кеш меморија по жељи и алгоритам замене. У тако направљену кеш меморију се шаљу захтеви (читање/упис и адреса). На крају се исписује резултат – време које је било потребно за извршавање операција.

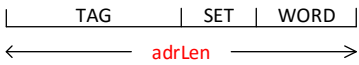
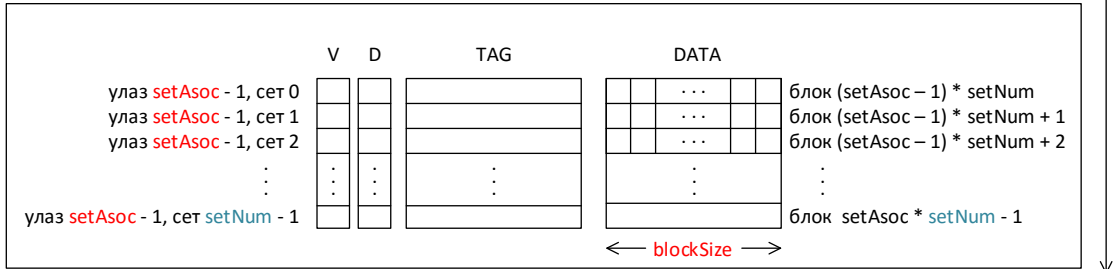
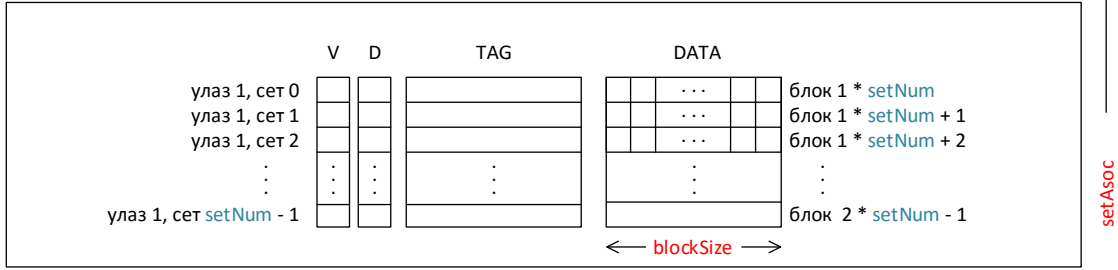
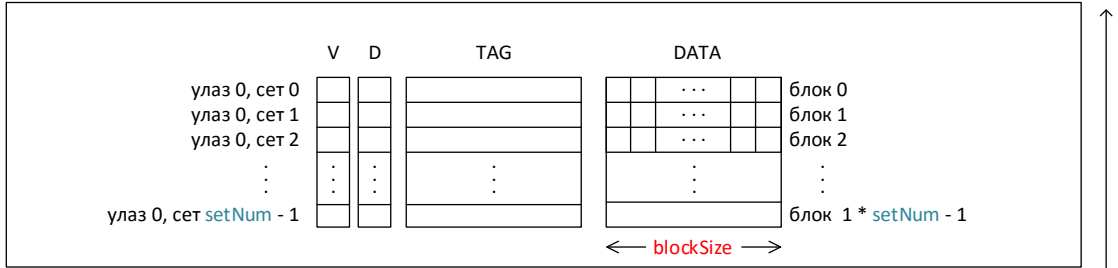
На основу начина рада кеш меморије (write back/write through i write allocate/no wite allocate) реализована су четири кеш меморије (није потребно знати реализацију за ову лабораторијску вежбу) :

- `CacheMemoryWriteBackNoWriteAlocated.java`
- `CacheMemoryWriteBackWriteAlocated.java`
- `CacheMemoryWriteThrouNoWriteAlocated.java`
- `CacheMemoryWriteThrouWriteAlocated.java`

Конструктори у свим случајевима реализације имају параметре `int adrLen`, `int setAsoc`, `int blockNum`, `int blockSize`.

<code>adrLen</code>	- дужина (у битовима) адресе којима се приступа у меморију
<code>setAsoc</code>	- број сетова
<code>blockNum</code>	- број блокова у целој кеш меморији
<code>blockSize</code>	- број речи у блоку

У наставку је шема која детаљније објашњава ове променљиве.



$$setNum = blockNum / setAsoc$$

setAsoc

1. Симулатор је потребно изменити тако што ће се додати могућност читања текстуалне датотеке у којем ће се налазити захтеви за меморију (операција и адреса којој се приступа). Сваки ред датотеке ће представљати нов захтев. Примери таквих датотека стоје уз симулатор (astar.trace, bzip2.trace, gcc.trace...).
Прво слово у реду представља операцију читања (load) "l" и операцију писања (store) "s", након тога постоји размак, а затим и адреса у хексадецималном ситему.
2. Од студента се очекује да зна да имплементира FIFO или LRU algoritam zamene (FIFOReplacementPolicy.java i LRUReplacementPolicy.java).
3. Потребно је да се табеларно прикаже време за извршавање захтева за меморију. Захтеви се налазе у датотекама из првог задатка.

Променљиве:

- назив датотеке
- број блокова кеш меморије
- сет-асоцијативност
- начин понашања кеш меморије у колико је било уписа, па се десио hit/miss (write through/write back, no write allocate/write allocate)

На основу таблице треба назначити која конфигурација кеш меморије је најбоља за дату датотеку.