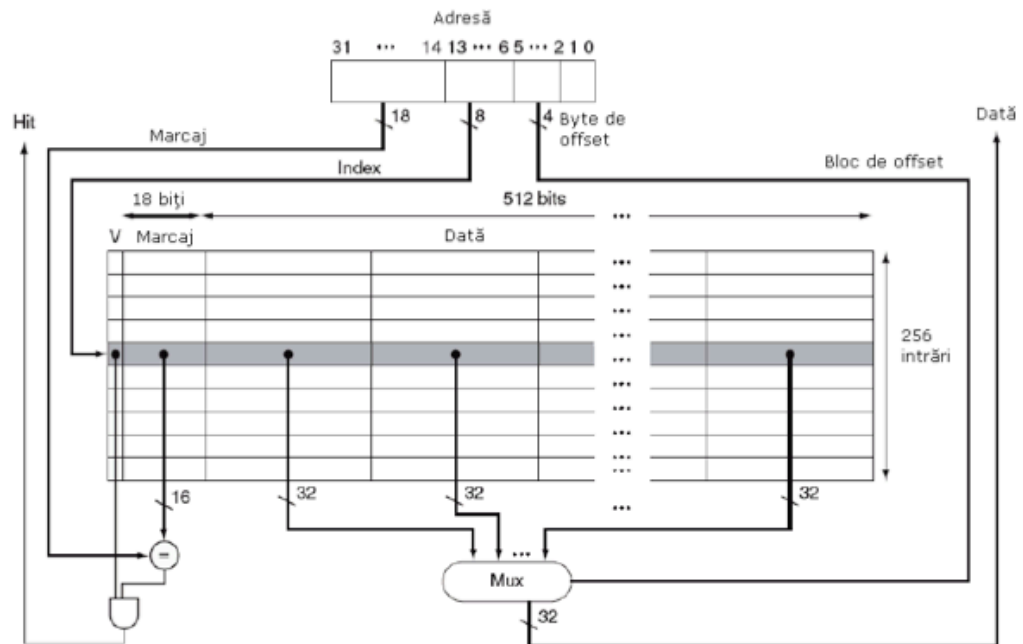


1. Descrieți caracteristicile generale ale unui program care folosește principiile localizării temporare și spațiale foarte puțin cu privire la accesul de date. Scrieți un exemplu la nivel de pseudocod.
2. Descrieți caracteristicile generale ale unui program care folosește principiul localizării temporare foarte mult și principiul localizării spațiale foarte puțin cu privire la accesul de date. Scrieți un exemplu la nivel de pseudocod.
3. Descrieți caracteristicile generale ale unui program care folosește principiul localizării temporare foarte puțin și principiul localizării spațiale foarte mult cu privire la accesul de date. Scrieți un exemplu la nivel de pseudocod.
4. Un processor nou poate utiliza tehnicile de write-through sau write-back folosite la memoria cache selectabile prin software.
 - a. Presupunem că procesorul va rula aplicații care folosesc intensiv date deci numărul operațiilor de LOAD și STORE este foarte mare. Explicați ce metodă de scriere în memoria cache trebuie utilizată
 - b. Considerați aceeași întrebare dar pentru un system în care integritatea datelor este critică; este mai importantă decât performanța memoriei.
5. Presupunem următoarea serie de referințe de adrese la memoria cache: 2, 3, 11, 16, 21, 13, 64, 48, 19, 11, 3, 22, 4, 27, 6 și 11. Presupunând că memoria cache este cu mapare direct, cu 16 blocuri de câte un cuvânt (memoria se consideră inițial goală), arăți conținutul final al memoriei cache și specificați numărul de HIT-uri sau MISS-uri.
6. Utilizând seria de referințe de la problema 5, arătați conținutul final al memoriei cache pentru o memorie cache cu mapare direct care are blocuri de 4 cuvinte și o dimensiune totală de 16 cuvinte. Totodată să se specifice numărul de HIT-uri sau MISS-uri.
7. Se dă următorul pseudocod:

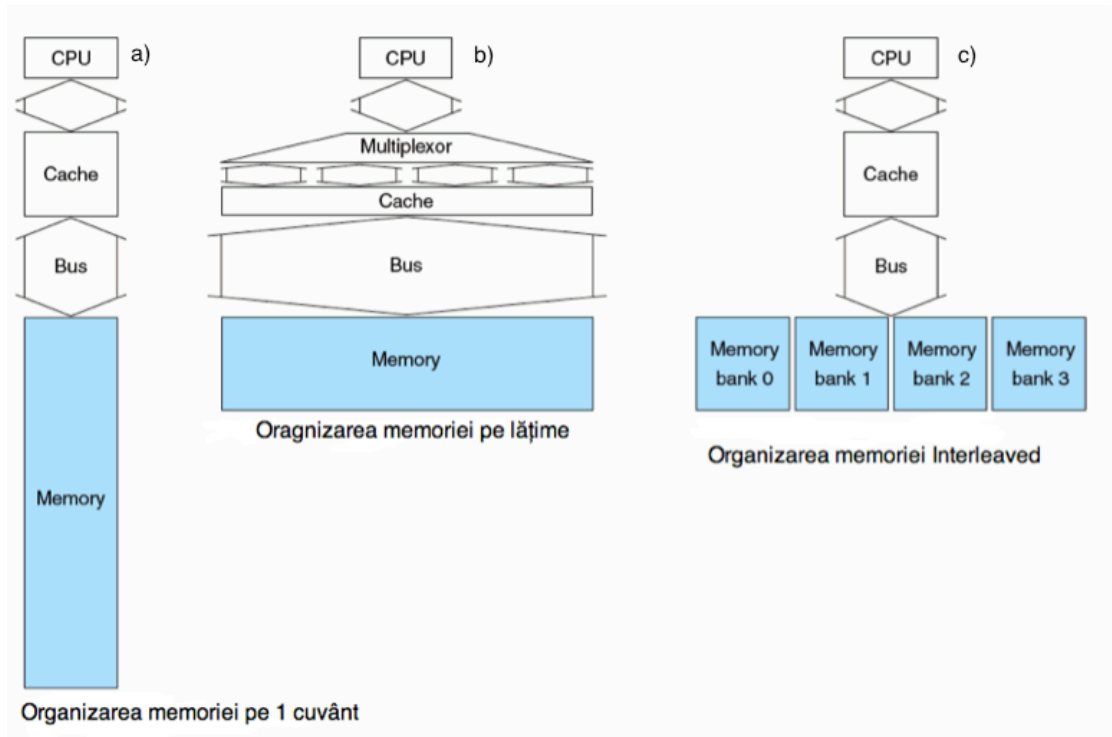
```
int array[10000, 100000];
pentru fiecare element array[i][j]{
    array[i][j] = array[i][j]*2;
}
```

Să se scrie două programe C care implementează acest algoritm: unul va trebui să acceseze elementele vectorului în ordinea de pe linie iar cel de al doilea program să acceseze elementele în ordinea de pe coloană. Comparați timpul de execuție al celor două programe. Ce va spune această comparație despre efectele asupra unei memorii cache implementată în fiecare din aceste situații.
8. Calculați numărul total de biți necesar pentru a implementa memoria cache din figura de mai jos. Acest număr este diferit de dimensiunea memoriei cache care usual este referită ca numărul de bytes de date memorat. Numărul de biți necesar pentru implementarea memoriei cache

reprezintă cantitatea totală de memorie necesară pentru memorarea datelor, tagurile și biții valizi.



- Considerăm o ierarhie de memorie care utilizează una din următoarele trei categorii de organizare a memoriei (pentru mai multe detalii consultați Anexa 1). Presupunem că mărimea blocului cache este de 16 cuvinte, dimensiunea organizării pentru cazul b) este de 4 cuvinte și că numărul de bank-uri din organizarea c) este 4. Dacă memoria principală are o latență pentru noile accese de 10 cicluri de ceas magistrală-memorie și timpul de transfer este de 1 ciclu de ceas magistrală-memorie, determinați MISS-urile pentru fiecare dintre cele 3 cazuri.



10. Presupunem un system de memorie care suportă interleaving (vezi Anexa 1) pentru 4 scrieri sau 4 citiri. Se dau următoarele adrese de memorie în ordinea în care ele apar pe magistrala de memorie: 3, 9, 17, 2, 51, 37, 13, 4, 8, 41, 67, 10. Care dintre ele rezultă într-un bank conflict ?
11. Buffer overflows este o problemă exploatată pentru preluarea controlului asupra unui system. Dacă buffer-ul este alocat pe o stivă, un hacker poate supraîncărca buffer-ul și insera o secvență de cod pentru a compromite sistemul. Oferiți un mechanism hardware care poate preveni această situație.
12. Un processor are un TLB cu 16 intrări și utilizează 4KB de pagini. Care este performanța pentru acest system dacă un program accesează cel puțin 2MB de memmorie la un moment de timp. Ce se poate face pentru îmbunătățirea performanței ?
13. Considerăm un system de memorie virtuală cu următoarele proprietăți:
 - a. 40-bit adrese virtuale
 - b. paginile au dimensiunea de 16KB
 - c. 36-bit adrese fizice

ANEXA 1. Proiectarea sistemului de memorie pentru a suporta memoria cache

MISS-urile la memoria cache sunt satisfăcute din memoria principală. Memoria principală este construită din DRAM. Acest tip de memorie este proiectat pentru densitate mare nu pentru viteză. Este dificil să se reducă latența citirii primului cuvânt din memorie, însă se poate reduce rata de MISS dacă creștem lățimea magistralei de la memorie la memoria cache.

Astfel vom putea utiliza blocuri de dimensiuni mari iar rata de MISS va fi la fel de mică ca în cazul blocurilor mari.

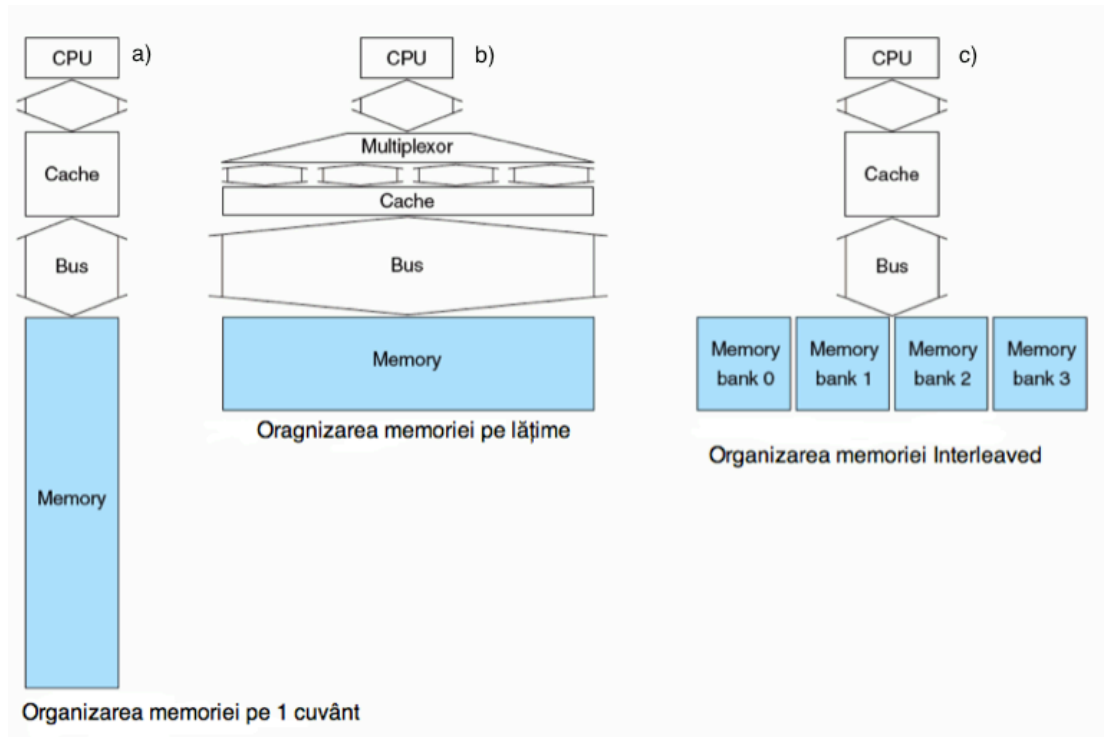
Procesorul este conectat la memorie prin intermediul unei magistrale. Frecvența ceasului pentru magistrală este mult mai mică decât frecvența procesorului, de regulă cu un factor de 10 ori.

Presupunem:

1. 1 ciclu de ceas de magistrală-memorie pentru trimiterea adresei
2. 15 cicluri de ceas de magistrală-memorie pentru fiecare acces DRAM inițiat
3. 1 ciclu de ceas de magistrală-memorie pentru a trimite un cuvânt de date

Dacă avem un bloc de memorie cache de 4 cuvinte și un bank de DRAM de dimensiune 1 cuvânt, rata de MISS va fi de $1 + 4 \times 15 + 4 \times 1 = 65$ cicluri de ceas de tipul magistrală-memorie. Deci numărul total de bytes transferat per ciclul de ceas pentru un singur MISS este de $4 \times 4 / 65 = 0.25$

În figura de mai jos se prezintă 3 posibilități de proiectare a sistemului de memorie.



Prima opțiune reflectă proiectarea cu o memorie de 1 cuvânt și toate accesese se fac secvențial. Cea de a doua proiectare mărește lățimea de bandă către memorie prin scalarea memoriei și a magistralelor între procesor și memorie; aceasta permite accesul paralel la toate cuvintele dintr-un bloc. Cea de a treia opțiune mărește lățimea benzii prin mărirea memoriei, dar nu și prin mărirea magistralei de interconectare.

Mărirea dimensiunii memoriei și a magistralei va crește în mod proporțional lățimea benzii memoriei; descrește astfel timpul de acces și timpul de transfer proporțional cu rata de MISS penalty. Cu o memorie a cărei lățime este de 2 cuvinte, MISS penalty va descrește de la 65 de cicluri de ceas la $1 + 2 \times 15 + 2 \times 1 = 33$ cicluri de ceas. Lățimea de bandă pentru un singur MISS este de 0.48 bytes per ciclu de ceas pentru o memorie de dimensiune 2 cuvinte, și 0.94 bytes per ciclu de ceas al magistralei dacă memoria este de dimensiune 4 cuvinte.

Cipurile de memorie pot fi organizate în bank-uri pentru citirea/scrierea de cuvinte multiple într-un singur acces. Fiecare bank poate fi de dimensiunea unui cuvânt a.î. lățimea magistralei și a memoriei cache să nu fie schimbate – dar trimiterea unei adrese la anumite bank-uri permite tuturor să realizeze o citire simultană.