Proiect Simulare

Farcas Gabriel
Dragomir Andrei
Semigrupa 244/1
Proiect : A3

Cerinte :

3.1. Se va modela şi simula un cache de instrucţiuni şi date perfect din punct de vedere al ratei de hit (100% hit). Se va dovedi o metrică ideală cu care vom compara rezultatele obţinute folosind cacheuri de instrucţiuni şi date reale (rată de hit diferită de 100%). Grafic IR(Rata de hit).

3.2. Simulaţi execuţia realistă a branch-urilor penalizând 0 respectiv 1 sau 2 cicli, la fiecare branch trimis spre execuţie. La 0 cicli penalizare avem varianta optimă de predicţie perfectă. Reprezentaţi grafic comparativ rata de procesare în cele trei situaţii.

Clasa BufferPerfetch

Variabilele statice:

- trc: o listă de tip TrcFile, care va stoca instrucțiunile din fișierul trc pentru a fi procesate.
- oldAdr: un șir de caractere care va stoca adresa de memorie pentru instrucțiunea anterioară.
- nrCiclii: un întreg care va stoca numărul total de cicluri necesare pentru a procesa instrucțiunile din trc.
- varLoad: un întreg care va stoca adresa de memorie pentru care s-a făcut o instrucțiune de tip load.

Metodele:

- LoadStoreList: aceasta este o metodă publică care acceptă ca parametru o listă de tip TrcFile și returnează un întreg. În interiorul metodei, se resetează nrCiclii la 0 și se creează o listă numită contor. Apoi, din lista primită ca parametru, se șterg toate instrucțiunile de tip branch. Se salvează adresa de memorie a primei instrucțiuni în variabila oldAdr.
- În continuare, se parcurge lista de instrucțiuni și pentru fiecare instrucțiune se verifică tipul (load sau store) și adresa de memorie. În funcție de aceste condiții, se calculează numărul de cicluri necesare și se actualizează valoarea lui varLoad.
- La final, se returnează numărul total de cicluri necesare, stocat în variabila nrCiclii.

Clasa Cache

Acest cod implementează o clasă de simulare a unui cache de memorie.

- Clasa Block declară o listă de date de tip int, care reprezintă datele stocate în cache, precum și 3 proprietăți ale blocului: Dirty, Valid și tag.

- Clasa Cache declară proprietățile Misses, Accesses și MissRate, care sunt utilizate pentru a măsura performanța cache-ului, și proprietățile cacheSize și blockSize, care reprezintă dimensiunea cache-ului și dimensiunea fiecărui bloc din cache. De asemenea, declară o listă de obiecte Block.
- Metoda ResetStatistics resetează statisticile pentru cache (Misses, Accesses și MissRate).
- Metoda UpdateBlockSize initializează lista de obiecte Block cu un număr de elemente egal cu dimensiunea cache-ului și apoi inițializează fiecare bloc cu o dimensiune egală cu dimensiunea blocului.
- Metoda CalculeazaTag calculează tag-ul pentru o anumită adresă de memorie prin împărțirea adresei la dimensiunea cache-ului.
- Metoda CalculeazaOffset calculează offset-ul pentru o anumită adresă de memorie prin împărțirea restului împărțirii adresei la dimensiunea cache-ului la dimensiunea fiecărui bloc.
- Metoda Read caută adresa specificată în cache. Dacă adresa nu există în cache, se apelează metoda Write pentru a adăuga adresa în cache.
- Metoda Write adaugă adresa specificată în cache, actualizează numărul de accesări ale cache-ului și numărul de accesări care au dus la erori (Misses) și recalculează rata erorilor.

Clasa Simulare

- Este o clasa statica care contine mai multe proprietati si metode pentru a simula performanta unui procesor.
- La inceputul clasei, exista un constructor statica care initializeaza obiectele DataCache si InstructionCache cu clasa Cache .
- Latency, MemLatency si Penalty sunt proprietati publice care sunt utilizate pentru a seta latente sau penalitati.
- Metodele ConfigureIC() si ConfigureDC() sunt utilizate pentru a configura cache-ul de instructiuni si cache-ul de date, respectiv.
- Metoda Execute() este utilizata pentru a executa simularea propriu-zisa, primind o lista de obiecte TrcFile si o penalitate ca parametri. In aceasta metoda se face o parcurgere prin lista de TrcFile si se simuleaza daca citirea din cache este o acoperire sau o ratare. Daca instrucțiunea este un branch, se verifica in cache-ul de instrucțiuni si se adauga penalty la numarul total de ciclii de executie. Daca este o operatie de stocare sau de incarcare, se verifica in cache-ul de date. Numarul total de ciclii de executie este incrementat cu Latency.
- CicliiExecutie, CicliiExecutieML si nrInsAritm sunt proprietati publice care tine statistici despre ciclii de executie, ciclii de executie ai memoriei si numarul de instructiuni aritmetice efectuate.

Clasa TrcFile

- Reprezinta un fisier .trc (fisier de tracer) care contine informatii despre instructiunile procesorului. Clasa contine proprietatile publice "branch", "store", "load", "pcCurent" si "adrDestinatie" care stocheaza informatiile din fisier.
-

- Metoda Parse() este utilizată pentru a parsa un rand din fisier si returneaza o lista de obiecte TrcFile care au valorile corespunzatoare celor din randul dat.
- Metoda GetInstructions() este utilizată pentru a citi un fisier selectat din ListView si returnează o listă de instrucțiuni din fisierul selectat.
- Metoda CountInstructions() este utilizată pentru a numara numarul de instructiuni de tip "B" (branch), "S" (store) si "L" (load) din fisier.
- Apoi returneaza o lista de intregi care reprezinta numarul de instructiuni pentru fiecare tip.

Simulare.ConfigureDC(64, 4): Aceasta functie configura dimensiunea cache-ului de date cu 64 si dimensiunea blocului cu 4.

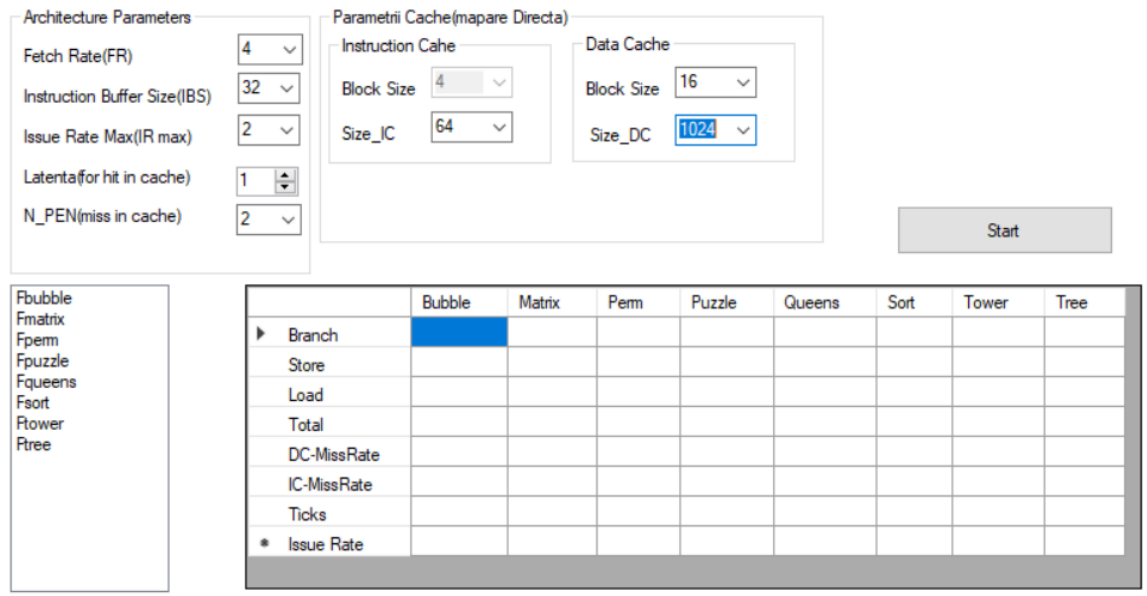
Simulare.ConfigureIC(64, 4): Aceasta functie configura dimensiunea cache-ului de instructiuni cu 64 si dimensiunea blocului cu 4.

ResetStatistics din Simuleaza
Aceasta functie pare sa reseteze statisticile pentru o anumita clasa. In cazul de fata statisticile sunt Misses, Accesses, si MissRate care sunt initializate la 0 sau 0.0f. Acestea ar putea fi folosite pentru a tine evidenta numarului de accesari la cache (Accesses) si numarul de accesari care au esuat (Misses). MissRate apoi ar fi calculat ca fiind raportul dintre numarul de accesari care au esuat si numarul total de accesari la cache. In plus, aceasta functie reseteaza aceste valori la 0, astfel incepand sa se acumuleze statistici noi din nou.

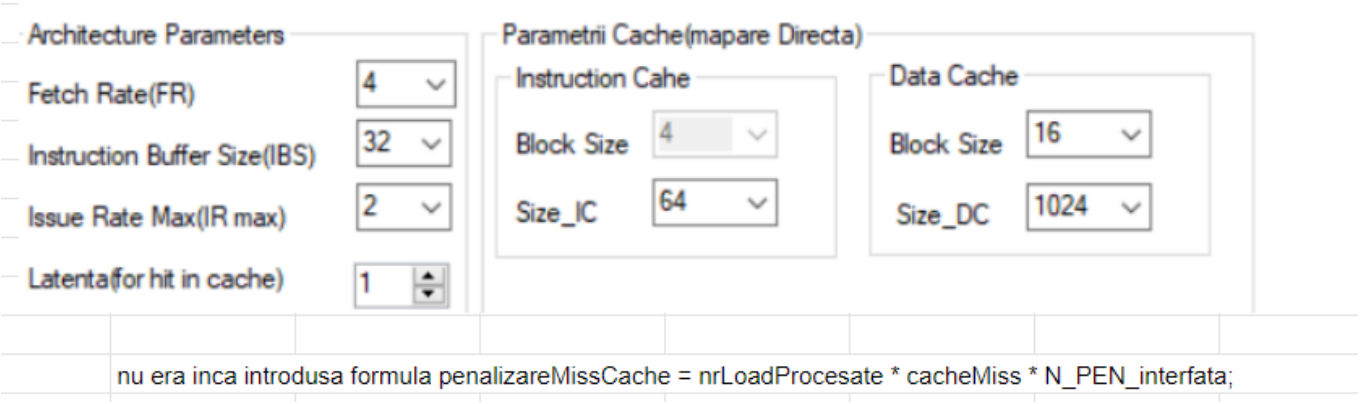
Aceasta metoda UpdateBlockSize din clasa Cache este folosita pentru a actualiza dimensiunea fiecarui bloc din cache. Aici se face alocarea dinamica a memoriei pentru cache.

In primul rand se initializeaza un obiect List denumit "block" care are capacitatea de a stoca Block-uri si este de dimensiunea cacheSize. Apoi prin intermediul unui for se parcurge aceasta lista si se adauga in fiecare pozitie un nou obiect Block.
In fiecare iteratie, se apeleaza metoda UpdateBlockSize din clasa Block, care primeste ca si parametru dimensiunea blockSize, care va fi folosita pentru a actualiza dimensiunea fiecarui bloc din cache
Aceasta metoda este folosita pentru a seta dimensiunea blocului pentru fiecare cache.

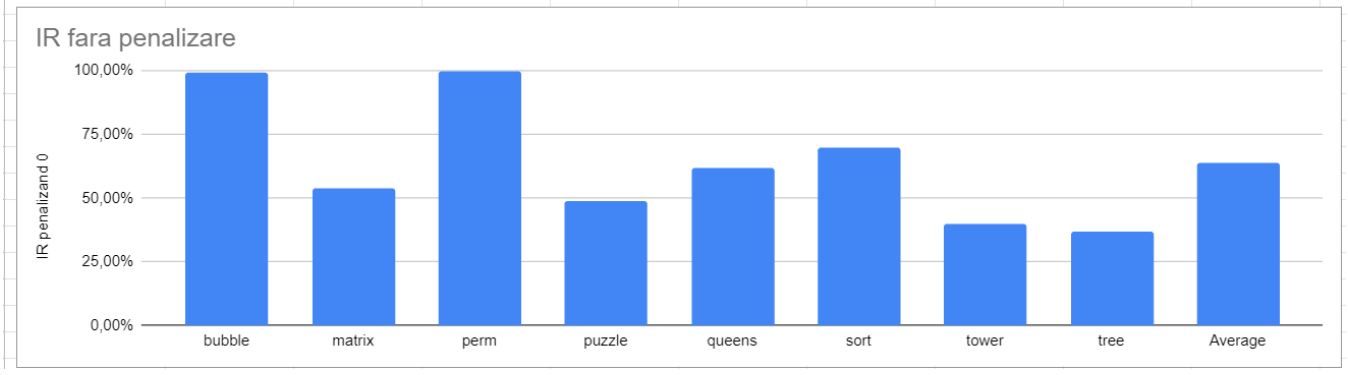
Simulare :



Date folosite :

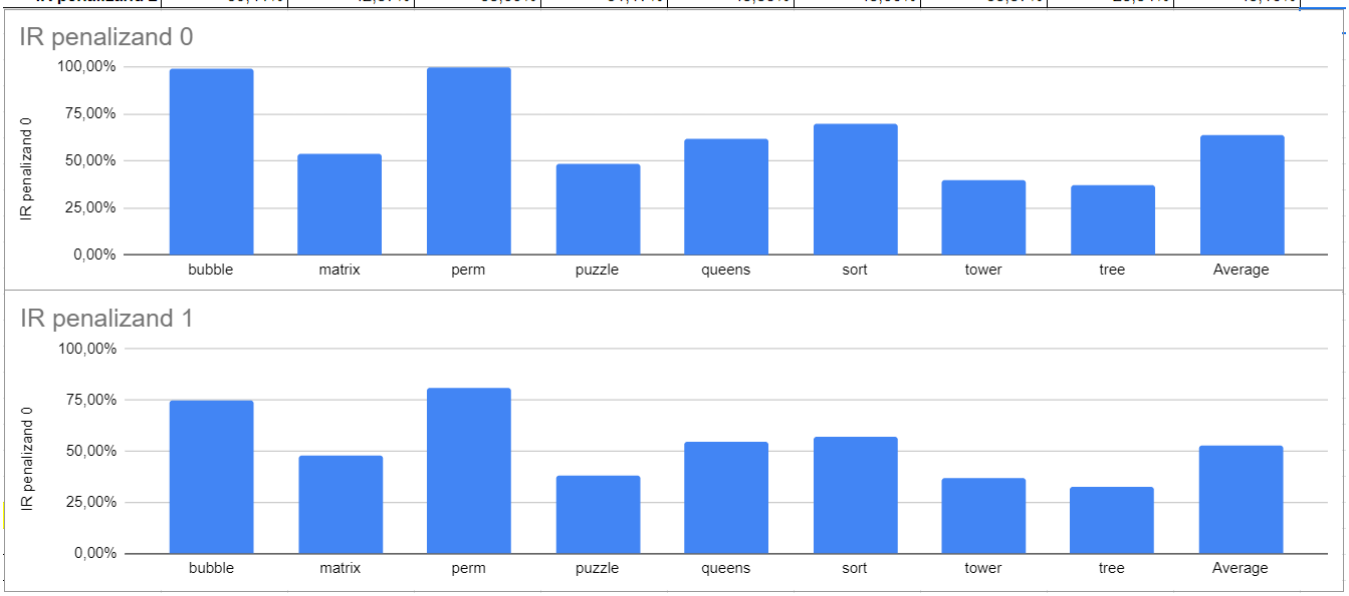


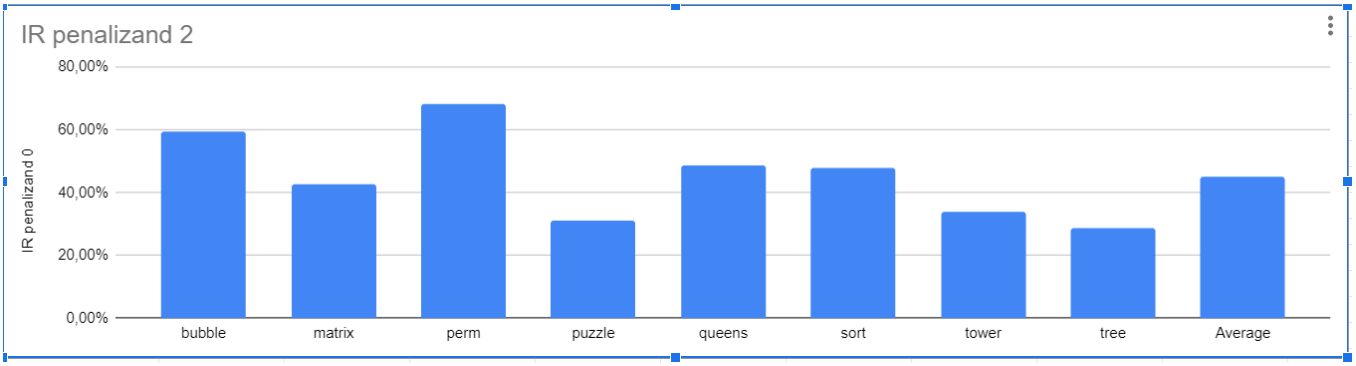
3.1									
	bubble	matrix	perm	puzzle	queens	sort	tower	tree	Average
IR fara penalizari	99,26%	53,73%	99,84%	48,62%	62,02%	69,71%	40,11%	36,99%	63,79%



3.2

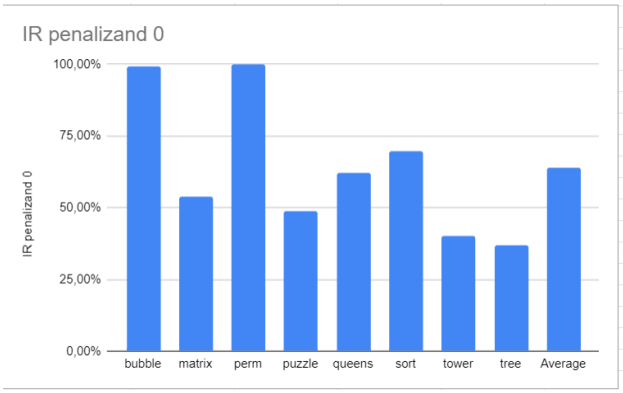
	bubble	matrix	perm	puzzle	queens	sort	tower	tree	Average
IR penalizand 0	99,26%	53,73%	99,84%	48,62%	62,02%	69,71%	40,11%	36,99%	63,79%
IR penalizand 1	74,59%	47,69%	80,96%	37,99%	54,66%	56,86%	36,73%	32,22%	52,71%
IR penalizand 2	59,41%	42,87%	68,09%	31,17%	48,86%	48,00%	33,87%	28,54%	45,10%





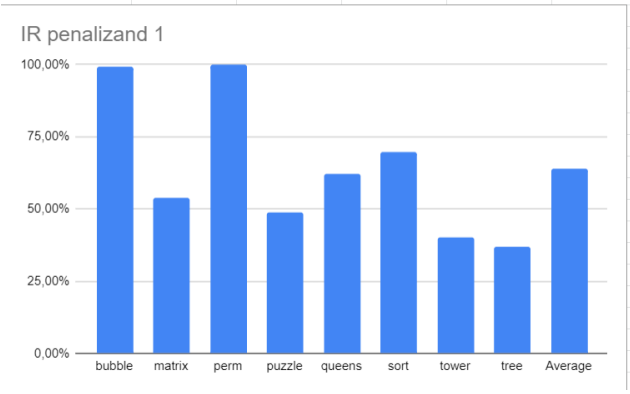
Rezultatele modificate :

	bubble	matrix	perm	puzzle	queens	sort	tower	tree	Average
IR penalizand 0	99,26%	53,73%	99,84%	48,62%	62,02%	69,71%	40,11%	36,99%	63,79%
IR penalizand 1	99,26%	53,73%	99,84%	48,62%	62,02%	69,71%	40,11%	36,99%	63,79%
IR penalizand 2	99,18%	53,74%	99,83%	48,62%	62,02%	69,71%	40,11%	36,99%	63,78%



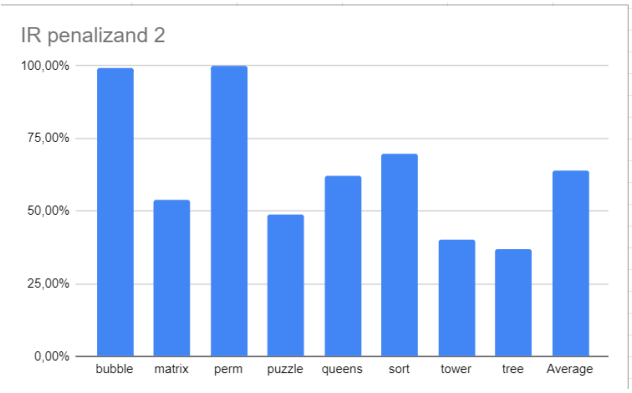
Architecture Parameters
Fetch Rate(FR) 4
Instruction Buffer Size(IBS) 32
Issue Rate Max(IR max) 2
Latenta(for hit in cache) 1
N_PEN(miss in cache) 0

Parametrii Cache(mapare Directa)
Instruction Cahe
Block Size 4
Size_IC 64
Data Cache
Block Size 16
Size_DC 1024



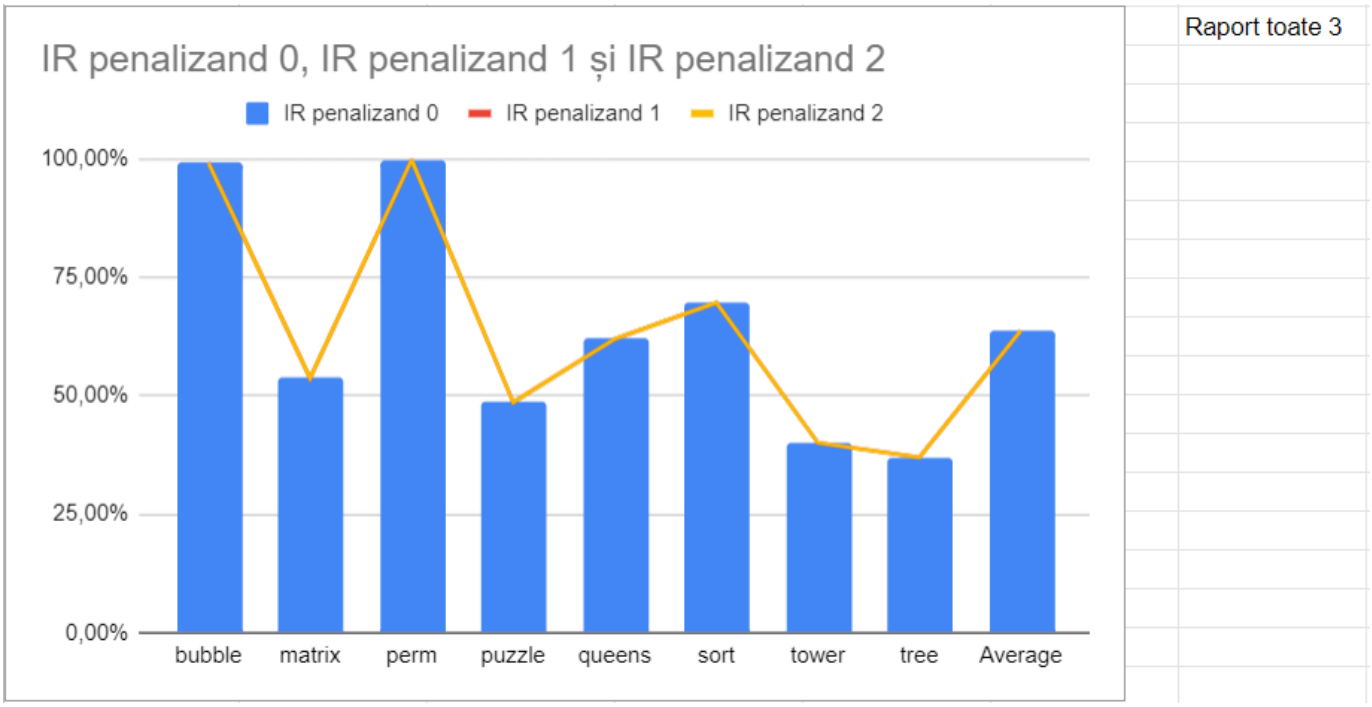
Architecture Parameters
Fetch Rate(FR) 4
Instruction Buffer Size(IBS) 32
Issue Rate Max(IR max) 2
Latenta(for hit in cache) 1
N_PEN(miss in cache) 1

Parametrii Cache(mapare Directa)
Instruction Cahe
Block Size 4
Size_IC 64
Data Cache
Block Size 16
Size_DC 1024



Architecture Parameters
Fetch Rate(FR) 4
Instruction Buffer Size(IBS) 32
Issue Rate Max(IR max) 2
Latenta(for hit in cache) 1
N_PEN(miss in cache) 2

Parametrii Cache(mapare Directa)
Instruction Cahe
Block Size 4
Size_IC 64
Data Cache
Block Size 16
Size_DC 1024



Link pentru a vedea graficele:

https://docs.google.com/spreadsheets/d/1a7g0CaIP8CR3wFK6etvzuhjjMVAHajOUQ0pjxoGXXb4/edit?usp=sharing