

Simularea si Optimizare Arhitecturilor de Calcul Proiect 2022-2023

Autori: Alexandru Dan Bucur & Timar Cosmin
Grupa: 243/1

1. Tema proiectului

Se dorește implementarea unui simulator pentru o arhitectura superscalara parametrizabila. Scopul principal este acela de a determina diferiți parametri de performanță, configurații optime, pentru o arhitectură Harvard de memorie (cache-uri de instrucțiuni și date separate).

Cerință:

Determinați influența numărului maxim de instrucțiuni ce pot fi trimise simultan în execuție asupra ratei de procesare $IR(IR_{max})$. La acest punct nu se va mai considera număr nelimitat de seturi de regiștrii generali. Se va determina numărul optim de seturi de regiștrii (2, 3, 4, ... IR_{max}) în variantele cu cache de date uniport (o singură instrucțiune cu referire la memorie se poate executa) sau biport (două instrucțiuni cu referire la memorie se pot executa: L+L sau L+S). Pentru valoarea optimă determinată la punctul 2.2. a numărului de seturi de regiștrii, studiați comparativ performanța (rata de procesare) pe două tipuri de cache de date (uniport sau biport).

2. Detaliile arhitecturii hardware

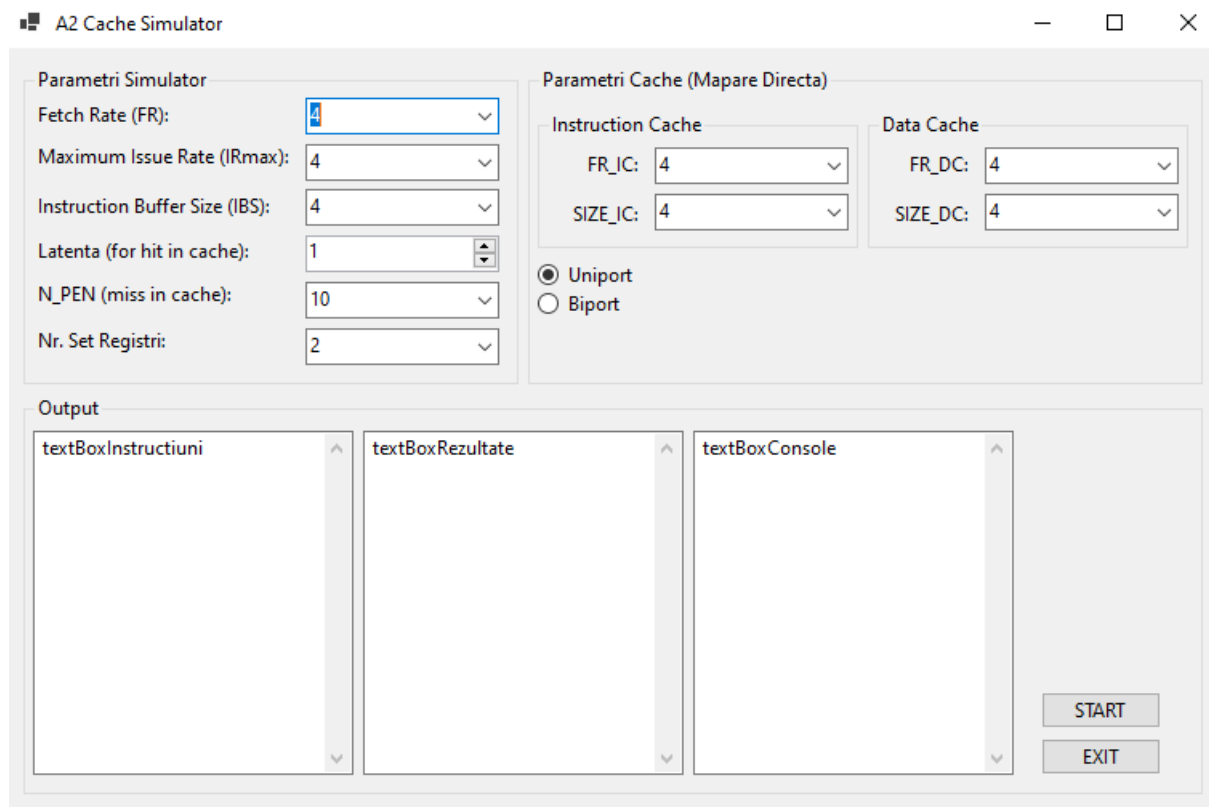
Principalii parametri ai arhitecturii sunt: FR (rata de fetch), specifică numărul de instrucțiuni citite simultan din cache sau memorie într-un ciclu de tact, poate lua valori de 4, 8 sau 16 instrucțiuni. IR_{max} (issue rate maxim), numărul maxim de instrucțiuni lansate în execuție simultan într-un ciclu de execuție din bufferul de prefetch. Poate lua valorile: 2, 4,

8, 16 instrucțiuni. IBS (instruction buffer size), dimensiunea bufferului de prefetch, măsurată în număr de instrucțiuni, plaja de valori: 4, 8, 16, 32. Latența, numărul de cicli necesare execuției instrucțiunilor aritmetice, de salt și cele cu referire la memorie (în cazul în care accesele pentru obținerea datei sunt cu hit în cache). Inițial are valoarea 1. Cache-ul de instrucțiuni (IC) și Cache-ul de date (DC) - sunt cache-uri mapate direct, organizate în blocuri de capacități parametrizabile [4, 8, 16 (maxim IBS) locații]. Memoria principală (se accesează numai la miss în cache) va avea o latență parametrizabilă de N_PEN (10, 15, 20) tact/procesor. Presupunem existența unui număr suficient de mare (maxim IRmax) de seturi de registre generali: un set de registre generali este necesar pentru execuția unei instrucțiuni de tip aritmetico-logic sau cu referire la memorie.

Programul va simula fișiere trace (*.trc), rulate pe arhitectura HSA (Hartfield Superscalar Architecture). Este vorba despre 8 benchmark-uri Stanford care cuprind probleme clasice de sortare, problema turnurilor din Hanoi, problema damelor, generare de permutări și înmulțiri de matrici.

3. Ghid de utilizare

Aplicația este foarte prietenoasă cu utilizatorul, iar pașii în utilizarea aplicației sunt minimaliști: Trace-urile sunt încărcate direct din program, și nu mai e nevoie să le alegem, din moment ce la o rulare vom avea toate rezultatele. În următorul pas, utilizatorul poate configura în funcție de preferințe parametrii pe care algoritmul nostru îi va utiliza. Parametrii trebuie să respecte anumite reguli. Chiar dacă nu va selecta nimic, și se va încerca rulare direct, acest factor nu este o problemă, pentru că aplicația salvează configurația fostei simulări. Algoritmul va rula în momentul apăsării butonului "Start". În acest punct, utilizatorul va fi înștiințat în cazul în care regulile menționate anterior nu sunt respectate. Pentru ieșirea din program, utilizatorul poate apăsa în orice moment butonul "Exit".



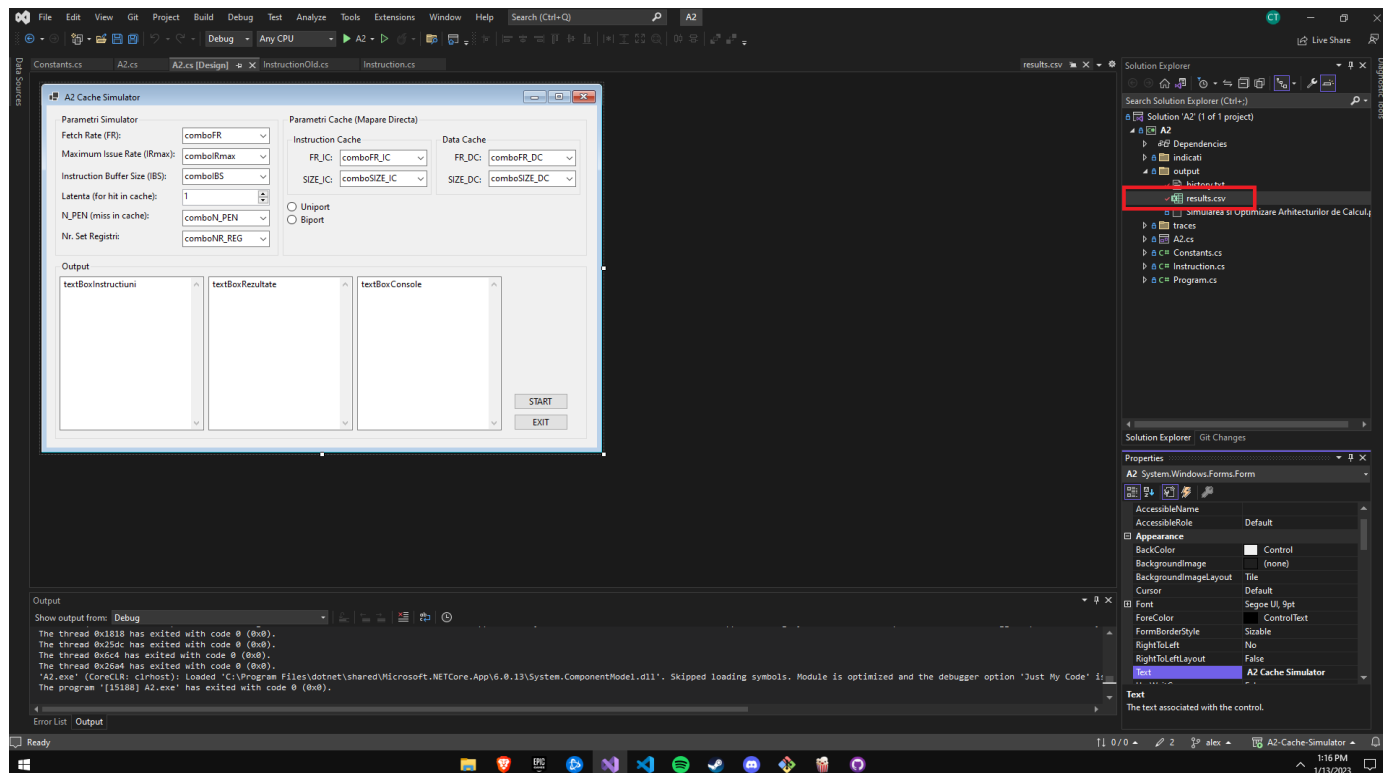
4. Resurse necesare

Resursele necesare sunt aceleași ca ale aplicației Visual Studio 2012. În crearea proiectului s-au folosit resurse minimaliste și nu s-au utilizat niciun pachet sau librărie specială.

5. Ghid de dezvoltare

Clasele corespunzătoare parametrilor simulatorului: Clasa "Constants.cs" o clasă în care avem hard codate toate constantele de pe parcursul aplicației. Clasa Instruction.cs, clasa care ține informații despre fiecare instrucțiune (instruction, currentPC, target), Clasa A2.cs, o clasă de tip Windows Forms, aici este implementată logica simulatorului. Funcția "Simulate" din aceasta clasă produce output-ul final.

În final, aplicația generează un fișier excell cu toate datele dintr-o simulare, ceea ce ajută la analiza acestora.



A	B	C	D	E	F	G	H	
FSORT.TRC	FTREE.TRC	FQUEENS.TRC	FMATRIX.TRC	FBUBBLE.TRC	FTOWER.TRC	FPERM.TRC	FPUZZLE.TRC	
11906	24278	37941	48145	40811	51225	72130	125411	
5152	11941	31547	18771	20084	45179	72132	11488	
8187	24166	19281	20638	31076	28777	43957	185951	
46855	75654	117650	144259	114063	125967	167423	481770	
72100	136039	206419	231813	206034	251148	355642	804620	
2.715	2.61	2.391	2.536	2.514	2.263	2.208	2.984	
26553	52118	86348	91411	81955	110988	161041	269603	

6. Concluzii

În ciuda faptului că am folosit un model maximal, nu am putut atinge mai mult de o instrucțiune, iar la mărirea fetch rate-ului se observă o mică îmbunătățire.