Tema 2 P.S.C

**Baicu Ioan Grupa 442c**

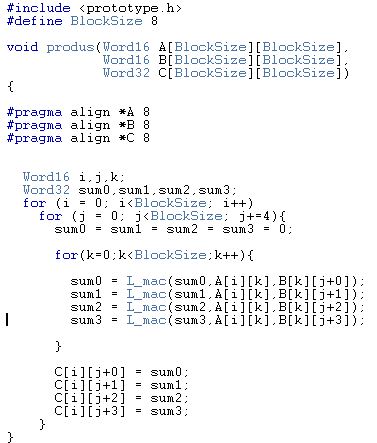
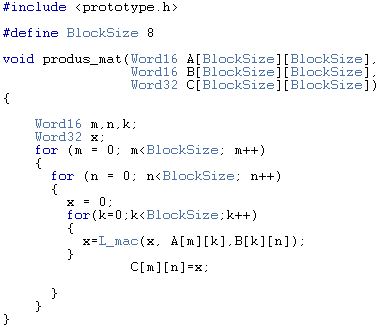
**Enunț tema 10**

Realizați un proiect C pentru StarCode140 care să realizeze produsul a doua matrice pătratice (de dimensiune multiplu de 4). Programul și funcția C vor folosi tipurile de date asociate numerelor fracţionare şi funcţiile intrinseci specifice DSP StarCore140.

De asemenea se va realiza un program de test Matlab care va genera un număr suficient de valori pentru a se testa programul C pentru SC140. Valorile de test generate în Matlab vor fi scrise în fişier şi vor fi citite de funcţia main din proiectul C. Apoi se va apela funcţia implementată şi rezultatele vor fi scrise (tot din main) în fişiere. În Matlab se compara apoi rezultatele obţinute, evaluându-se inclusiv erorile datorate formatelor de reprezentare.

**Codul C din CodeWarrior**

**Varinata neoptimizată Varianta Optimizată**



Prin implemantarea metodei optimizate am încercat să realizez înmulțirea matricelor folosind înmulțiri independente ce se pot face in același ciclu de tact. Am abordat metoda ”**loop-unroll**” pentru a realiza optimizarea codului și după câte se poate vedea se folosesc toate cele 4 operații mac disponibile la un ciclu de tact și totodată și parcurgerea for-ului din 4 în 4, drept uramare am ales ca dimeniunile matricelor să fie multiplu de 4.

**Algoritm**

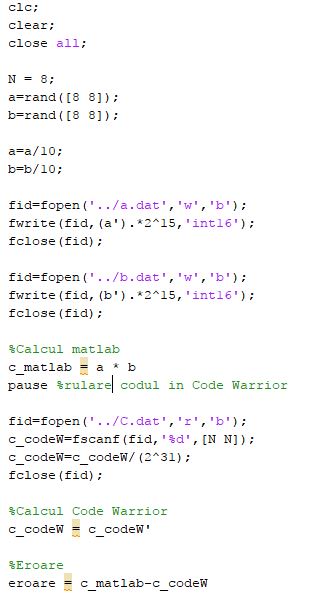
Avem **A** și **B** două matrice patratice de 4x4. Elementul 1 de pe linia 1 a matricei A va fi înmultit cu fiecare din elementele de pe linia 1 a matricei **B** și astfel vor rezulta 4 valori ce vor fi de fapt rezultate parțiale pentru primele 4 elemente de pe prima linie din matricea rezultat (**C**). În momentul în care s-a terminat de parcurs prima linie din matricea **A**, prima linie a matricei **C** va fi populate cu rezultatele finale și astfel se va trece la a doua linie din matrice **A**.

**Funcția principală**

Proiectul realizat pentru DSP-ul StarCore140 conține de asemenea și funcția **starcore\_main.c**, de mai jos, în cadrul căreia se citesc cele doua matrice generate din matlab. Dupa citirea matricelor se apelează funcția **produs\_mat** ce realizază înmultirea matricelor generate din matlab (A.dat, B.dat) oferind ca rezultat fișierul C.dat ce reprezintă matricea rezultat.



**Codul matlab**



Variabilele a și b reprezintă cele doua matrice de dimensune 8x8 generate random. În cadrul acestui program matricele sunt scrise în fișiere ca și transpuse ,deoarece CodeWarrior le va prelucra ca și transpuse.

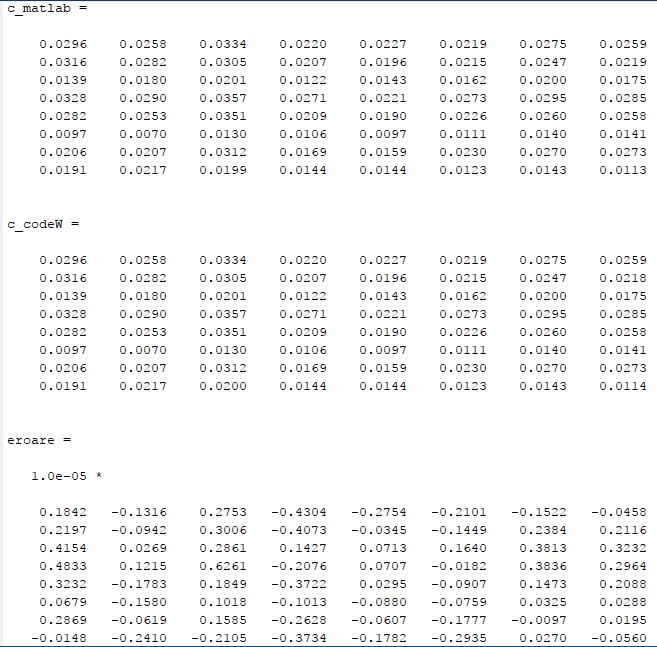
Variabila c reprezintă rezultatul înmulțirii lui a cu b ,iar ceastă matrice va fi comparata cu transpusa celei generate de CodeWarrior (d1=d’, d fiind matricea generată de CodeWarrior).

În urma rulării programului în matlab se vor genera cele 2 fisiere a.dat și b.dat, fisiere ce vor fi citite de Code Warrior.

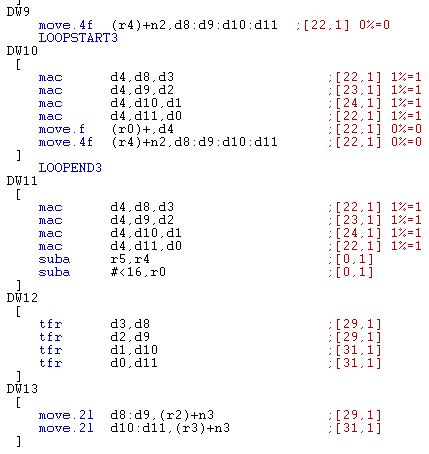
**Eroarea relativă**

Eroarea a fost calculată prin diferența celor două matrice calculate în matlab și în CodeWorrior .

Analizând rezultatele se observă că valorile erorilor sunt foarte mici, dar nu sunt nule datorită reprezentării pe biți a numerelor în matlab cât și a rotunjirilor făcute.



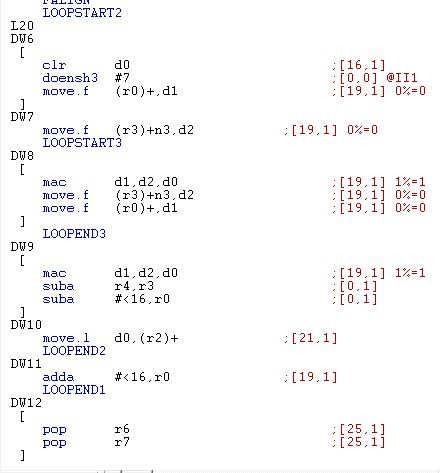
Varianta optimizatăîn limbaj de asmblare

****

Analiza codului în assembler folosind fisierele .sl pentru gerenearea carora am folosit nivel 2 de optimizare si smart unrolling : 0.

În cadrul acestei metode optimizate se poate observa folosirea tuturor celor 4 operatii de tip **mac** pe ciclul de tact.

Varianta neoptimizată în limbaj de asamblare



În varianta neoptimizată se poate observa folosirea unei singure operații **mac** pe un ciclu de tact.