Ενσωματωμένα Συστήματα Μικροεπεξεργαστών – ΗΡΥ411

Αναφορά Εργαστηρίου 10 – Κάνοντας υπολογισμούς με «μικρούς» μικροελεγκτές – Πολλαπλασιασμός 3x3 πινάκων float και int

Ομάδα: LAB41145877
Κιούλος Ευάγγελος 2016030056

Εισαγωγή:

Σκοπός της δέκατης εργαστηριακής άσκησης ήταν ο πολλαπλασιασμός πινάκων 3x3, αρχικά χρησιμοποιώντας int και στην συνέχεια χρησιμοποιώντας float. Με αυτό το απλό πρόγραμμα, μπορούμε εύκολα να παρατηρήσουμε ότι οι πράξεις με float χρειάζονται αρκετά παραπάνω πόρους από τον επεξεργαστή σε σχέση με τους int.

Χάρτης Μνήμης:

Πολλαπλασιασμός με int πίνακες:

0x0060	…	…	…	…	…	…	0x0071
0x0095	…	…	…	…	…	…	0x00A6
0x00CA	…	…	…	…	…	…	0x00DB
…	…	…	…	…	…	…	0x045F

- **Διεύθυνση "0x0060 – 0x0071":** (πορτοκαλί χρώμα) Στοιχεία του πίνακα A.
- **Διεύθυνση "0x0095 – 0x00A6":** (μπλε χρώμα) Στοιχεία του πίνακα B.
- **Διεύθυνση "0x00CA – 0x00DB":** (Κίτρινο χρώμα) Στοιχεία του πίνακα C.
- **Διεύθυνση "0x045F":** (Γκρι χρώμα) top of stack.

Πολλαπλασιασμός με float πίνακες:

0x0060	…	…	…	…	…	…	0x0083
0x0095	…	…	…	…	…	…	0x00B8
0x00CA	…	…	…	…	…	…	0x00ED
…	…	…	…	…	…	…	0x045F

- **Διεύθυνση "0x0060 – 0x0083":** (πορτοκαλί χρώμα) Στοιχεία του πίνακα A.
- **Διεύθυνση "0x0095 – 0x00B8":** (μπλε χρώμα) Στοιχεία του πίνακα B.
- **Διεύθυνση "0x00CA – 0x00ED":** (Κίτρινο χρώμα) Στοιχεία του πίνακα C.
- **Διεύθυνση "0x045F":** (Γκρι χρώμα) top of stack.

Υλοποίηση Εργαστηρίου:

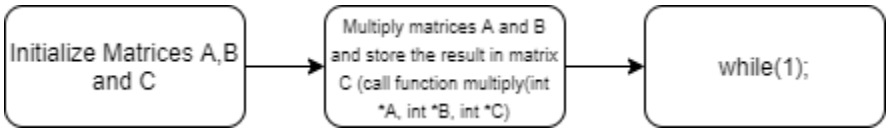
Πολλαπλασιασμός με int πίνακες:

Αρχικά, δημιουργήθηκαν οι συναρτήσεις "void init_A(void)", "void init_B(void)" και "void init_C(void)" που αρχικοποιούν τις τιμές των πινάκων στην μνήμη. Οι τρείς πίνακες είναι αρχικοποιημένοι στην μνήμη με την μορφή ενός μονοδιάστατου πίνακα. Για τον κάθε πίνακα χρησιμοποιείται ένας int pointer που δείχνει στην πρώτη διεύθυνση του κάθε πίνακα στην μνήμη. Για τον πίνακα A χρησιμοποιείται ο pointer "int *A" ο οποίος δείχνει στην διεύθυνση

“0x0060”, για τον B ο “int *B” που δείχνει στην διεύθυνση “0x0095” και για τον C ο “int *C” ο οποίος δείχνει στην διεύθυνση “0x00CA”. Κάθε πίνακας έχει 9 στοιχεία επομένως ο κάθε πίνακας χρειάζεται 18 bytes της μνήμης RAM αφού το κάθε στοιχείο χρειάζεται 2 bytes. Ο πίνακας C είναι ο πίνακας στον οποίο αποθηκεύεται το αποτέλεσμα του πολλαπλασιασμού και στην αρχή όλα τα στοιχεία του είναι 0.

Στην συνέχεια, δημιουργήθηκε η συνάρτηση “void multiply(int *A, int *B, int *C)” στην οποία γίνεται ο πολλαπλασιασμός των πινάκων. Τα πρώτα δύο ορίσματα δέχονται ένα pointer ο οποίος δείχνει στην πρώτη διεύθυνση των πινάκων που θα πολλαπλασιαστούν ενώ το τρίτο όρισμα δείχνει την πρώτη διεύθυνση του πίνακα στον οποίο θα αποθηκευτεί το αποτέλεσμα της πράξης.

Το ροή του κυρίως προγράμματος(main()) φαίνεται στο παρακάτω διάγραμμα:



Πολλαπλασιασμός με float πίνακες:

Για τον πολλαπλασιασμό float πινάκων 3x3 έγινε ξανά η παραπάνω διαδικασία με την διαφορά ότι οι pointers που δείχνουν στους πίνακες στην μνήμη αλλά και τα ορίσματα της συνάρτησης “multiply” είναι float. Επιπλέον, ο κάθε πίνακας χρειάζεται 36 bytes στην μνήμη καθώς το κάθε στοιχείο του χρειάζεται από 4 bytes. Η ροή του κύριου προγράμματος είναι ακριβώς η ίδια.

Αποτελέσματα Προσομοίωσης:

Πολλαπλασιασμός με int πίνακες:

Για τον πολλαπλασιασμό int πινάκων, θα πραγματοποιηθεί ο εξής πολλαπλασιασμός:

$$\begin{bmatrix} 150 & 12 & -45 \\ 6 & 21 & 52 \\ 104 & 89 & -32 \end{bmatrix} \begin{bmatrix} -3 & 27 & 62 \\ 10 & -128 & 46 \\ 72 & 1 & 69 \end{bmatrix} = \begin{bmatrix} -3570 & 2469 & 6747 \\ 3936 & -2474 & 4926 \\ -1726 & -8616 & 8334 \end{bmatrix}$$

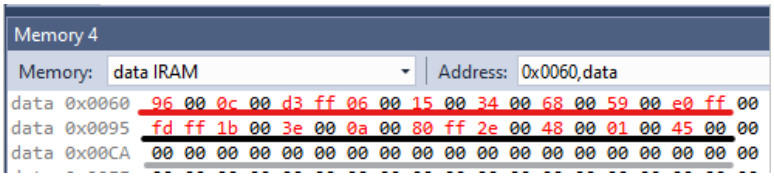
Όπου οι πίνακες A και B είναι:

$$A = \begin{bmatrix} 150 & 12 & -45 \\ 6 & 21 & 52 \\ 104 & 89 & -32 \end{bmatrix}, B = \begin{bmatrix} -3 & 27 & 62 \\ 10 & -128 & 46 \\ 72 & 1 & 69 \end{bmatrix}$$

Ο πίνακας C αρχικά έχει την τιμή:

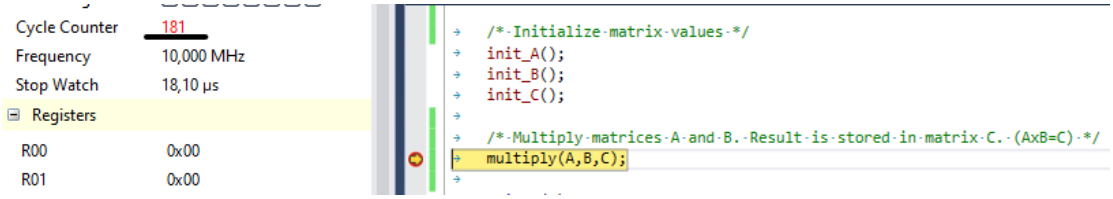
$$C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Ξεκινώντας την εκτέλεση του προγράμματος, παρατηρούμε ότι οι πίνακες A,B και C έχουν αρχικοποιηθεί στην μνήμη.

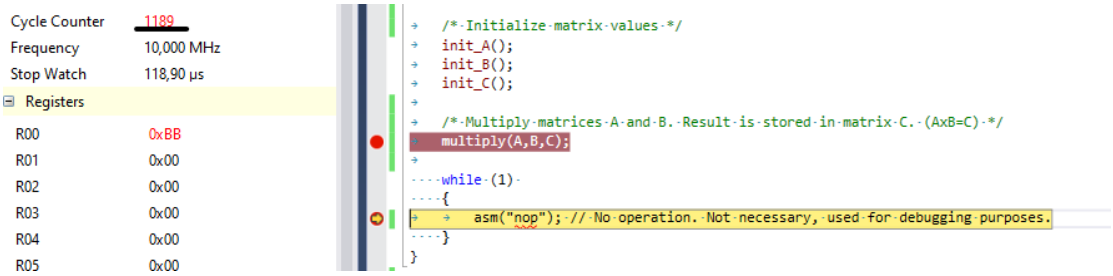


Στην παραπάνω εικόνα, με κόκκινη υπογράμμιση φαίνονται τα περιεχόμενα του πίνακα A, με μαύρη υπογράμμιση είναι τα περιεχόμενα του πίνακα B και με γκρι τα περιεχόμενα του πίνακα C. Τα περιεχόμενα στην μνήμη αποθηκεύονται με κωδικοποίηση 16-bit signed integer με σειρά little endian.

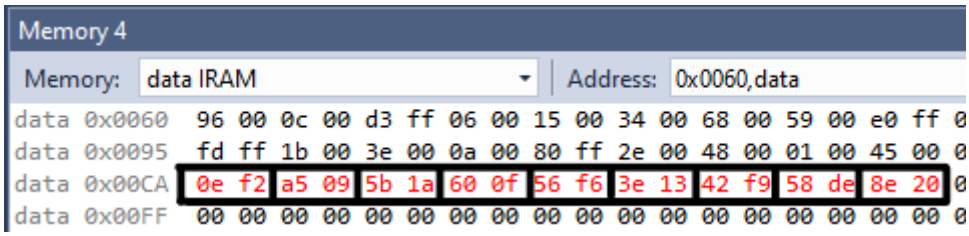
Επιπλέον, μετά την αρχικοποίηση και πριν την κλήση της συνάρτησης του πολλαπλασιασμού, το πρόγραμμα έχει ξοδέψει 181 κύκλους όπως βλέπουμε και στην παρακάτω εικόνα.



Μετά την κλήση της συνάρτησης του πολλαπλασιασμού, παρατηρούμε ότι το πρόγραμμα έχει φτάσει στους 1189 κύκλους. Από τους 1189 κύκλους που χρειάστηκαν για την εκτέλεση της συνάρτησης, χρειάστηκαν 614 κύκλοι μόνο για τον πολλαπλασιασμό των πινάκων.



Επιπλέον, παρατηρούμε ότι τα αποτελέσματα του πολλαπλασιασμού έχουν αποθηκευτεί στην μνήμη, στις διευθύνσεις του πίνακα C.



Παρατηρούμε ότι ο πίνακας που προκύπτει έχει τις παρακάτω τιμές.

$$C = \begin{bmatrix} 0x0EF2 & 0xA509 & 0x5B1A \\ 0x600F & 0x56F6 & 0x3E13 \\ 0x42F9 & 0x58DE & 0x8E20 \end{bmatrix}$$

Αν μετατρέψουμε τις δεκαεξαδικές τιμές σε δεκαδικές, ο πίνακας C θα είναι:

$$C = \begin{bmatrix} -3570 & 2469 & 6747 \\ 3936 & -2474 & 4926 \\ -1726 & -8616 & 8334 \end{bmatrix}$$

Επομένως, παρατηρούμε ότι στον πίνακα C προέκυψε το ζητούμενο αποτέλεσμα.

<u>Πολλαπλασιασμός με float πίνακες:</u>

Για τον πολλαπλασιασμό float πινάκων, θα πραγματοποιηθεί ο εξής πολλαπλασιασμός:

$$\begin{bmatrix} 150.32 & 12.253 & -45.05 \\ 6.371 & 21.0 & 52.12 \\ 104.73 & 89.99 & -32.007 \end{bmatrix} \begin{bmatrix} -3.2 & 27.52 & 62.9 \\ 10.0 & -128.4 & 46.3 \\ 72.72 & 1.0 & 69.3 \end{bmatrix} =$$

$$= \begin{bmatrix} -3634.5298 & 2518.4714 & 6900.4775 \\ 3979.7793 & -2468.95 & 4984.9512 \\ -1642.7852 & -8704.5518 & 8533.9697 \end{bmatrix}$$

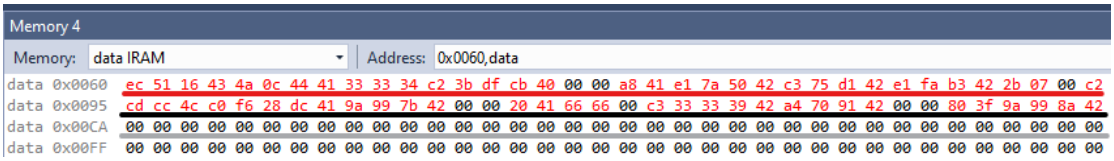
Όπου οι πίνακες A και B είναι:

$$A = \begin{bmatrix} 150.32 & 12.253 & -45.05 \\ 6.371 & 21.0 & 52.12 \\ 104.73 & 89.99 & -32.007 \end{bmatrix}, B = \begin{bmatrix} -3.2 & 27.52 & 62.9 \\ 10.0 & -128.4 & 46.3 \\ 72.72 & 1.0 & 69.3 \end{bmatrix}$$

Ο πίνακας C αρχικά έχει την τιμή:

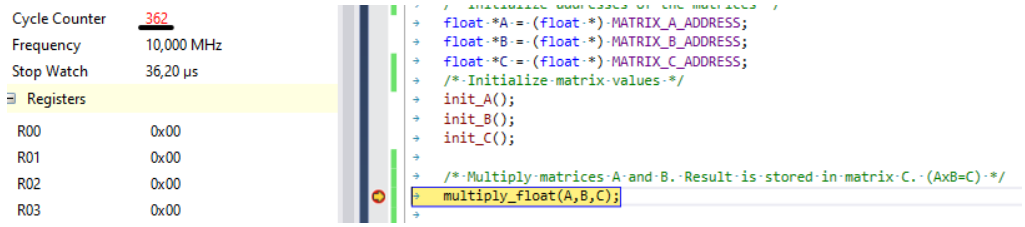
$$C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Ξεκινώντας την εκτέλεση του προγράμματος, παρατηρούμε ότι οι πίνακες A,B και C έχουν αρχικοποιηθεί στην μνήμη.

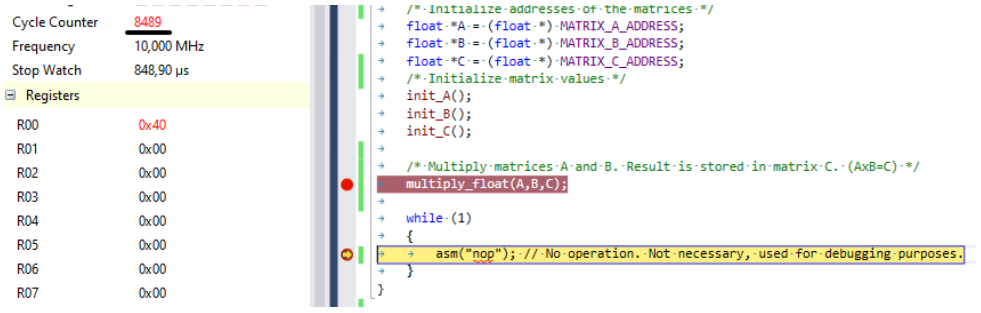


Στην παραπάνω εικόνα, με κόκκινη υπογράμμιση φαίνονται τα περιεχόμενα του πίνακα A, με μαύρη υπογράμμιση είναι τα περιεχόμενα του πίνακα B και με γκρι τα περιεχόμενα του πίνακα C. Τα περιεχόμενα στην μνήμη αποθηκεύονται με κωδικοποίηση IEEE 754 με σειρά little endian.

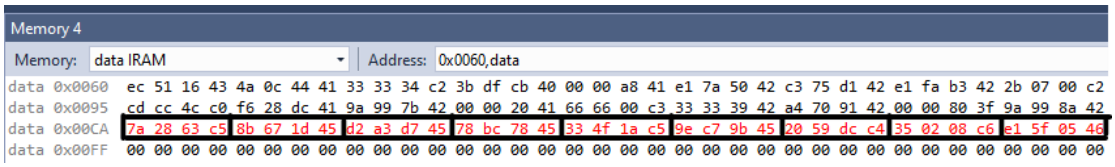
Επιπλέον, μετά την αρχικοποίηση και πριν κλήση της συνάρτησης του πολλαπλασιασμού, το πρόγραμμα έχει ξοδέψει 362 κύκλους όπως βλέπουμε και στην παρακάτω εικόνα.



Μετά την κλήση της συνάρτησης του πολλαπλασιασμού, παρατηρούμε ότι το πρόγραμμα έχει φτάσει στους 8489 κύκλους, από τους οποίους 5335 κύκλοι χρειάστηκαν για τον πολλαπλασιασμό των πινάκων.



Επιπλέον, παρατηρούμε ότι τα αποτελέσματα του πολλαπλασιασμού έχουν αποθηκευτεί στην μνήμη, στις διευθύνσεις του πίνακα C.



Παρατηρούμε ότι ο πίνακας που προκύπτει έχει τις παρακάτω τιμές.

$$C = \begin{bmatrix} 0x7A2863C5 & 0x8B671D45 & 0xD2A3D745 \\ 0x78BC7845 & 0x334F1AC5 & 0x9EC79B45 \\ 0x2059DCC4 & 0x350208C6 & 0xE15F0546 \end{bmatrix}$$

Αν μετατρέψουμε τις τιμές του πίνακα από IEEE 754 σε δεκαδικές, ο πίνακας C θα είναι:

$$C = \begin{bmatrix} -3570 & 2469 & 6747 \\ 3936 & -2474 & 4926 \\ -1726 & -8616 & 8334 \end{bmatrix}$$

Επομένως, παρατηρούμε ότι στον πίνακα C προέκυψε το ζητούμενο αποτέλεσμα.

Συμπεράσματα:

Μετά από την εκτέλεση και των δύο προγραμμάτων, παρατηρούμε ότι ο πολλαπλασιασμός float πινάκων 3x3 είναι πολύ πιο απαιτητικός σε σχέση με τον πολλαπλασιασμό int πινάκων 3x3. Πιο συγκεκριμένα, ο πολλαπλασιασμός με float χρειάστηκε περίπου 9 φορές περισσότερους κύκλους σε σχέση με τον πολλαπλασιασμό με int. Επιπλέον, για τον πολλαπλασιασμό int πινάκων, χρειαζόμαστε 18 bytes της μνήμης για κάθε πίνακα επομένως συνολικά 54 bytes και για τους τρείς πίνακες, ενώ για τον πολλαπλασιασμό float πινάκων, χρειαζόμαστε 36 bytes της μνήμης για κάθε πίνακα επομένως συνολικά 108 bytes και για τους τρείς πίνακες.

ΣΗΜΕΊΩΣΗ: Το πρόγραμμα με τον πολλαπλασιασμό int πινάκων 3x3 βρίσκεται στο project με όνομα “HPY411_Lab_10” ενώ πρόγραμμα με τον πολλαπλασιασμό float πινάκων 3x3 βρίσκεται στο project με όνομα “HPY411_Lab_10_float” στον παραδοτέο φάκελο.