

ΗΡΥ411- Ενσωματωμένα Συστήματα Μικροεπεξεργαστών

Εργαστήριο 10

LAB41145851

28/12/2020

Εμμανουήλ Πετράκος AM 2014030009

Στο 10ο εργαστήριο μελετάται η απόδοση ενός μικροελεγκτή τύπου AVR χρησιμοποιώντας δεδομένα σε μορφή float και long. Για αυτό έχει κατασκευαστεί ένα μικρό πρόγραμμα πολλαπλασιασμού πινάκων 3x3. Έχουν αφαιρεθεί τα κομμάτια των προηγούμενων εργαστηρίων.

Επεξήγηση προσέγγισης

Υπάρχουν δύο υλοποιήσεις του προγράμματος όπου η μόνη διαφορά τους είναι στην δήλωση των δεδομένων, στην πρώτη float και στην δεύτερη long. Οι πίνακες δηλώνονται ως global μεταβλητές για να μην κάνει ανεπιθύμητες βελτιστοποιήσεις ο compiler. Στη περίπτωση που ήταν τοπικές μεταβλητές, ο compiler θεωρούσε όλο το πρόγραμμα ως dead code καθώς δεν έκανε καμία αλλαγή εκτός του heap. Επίσης, αρχικοποιούνται με τυχαίες τιμές κατά την δήλωση και ο compiler παράγει τον απαραίτητο κώδικα.

Παρότι οι πίνακες είναι global και μπορούν να προσπελασθούν από παντού, η συνάρτηση πολλαπλασιασμού πινάκων δέχεται ως είσοδο pointers προς αυτούς για να συνάδει το πρόγραμμα με τις προδιαγραφές του εργαστηρίου. Η συνάρτηση καλείται από την main και υλοποιεί τον παρακάτω αλγόριθμο.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \\ = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} & a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} & a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \\ a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} & a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33} \end{pmatrix}$$

Πολλαπλασιασμός πινάκων 3x3

Και στις δύο περιπτώσεις ο compiler φαίνεται να χαρτογραφεί την μνήμη με τον παρακάτω τρόπο.

0x0060 - 0x0083	Input Matrix1
0x0084 - 0x00A7	Input Matrix2
0x00A8 - 0x00CB	Output Matrix

Επιλέχθηκε τα δεδομένα να είναι ίδιου μεγέθους και στις δύο υλοποιήσεις, ώστε για την διαφορά στην απόδοση να ευθύνεται μόνο το υπολογιστικό μέρος και όχι οι προσβάσεις στην μνήμη.

Πειραματική Διαδικασία

Ως απόδοση κάθε υλοποίησης θεωρείται ο χρόνος εκτέλεσης της συνάρτησης πολλαπλασιασμού πινάκων. Για αυτό χρησιμοποιούνται δύο breakpoints, ένα πριν την κλήση της και ένα αμέσως μετά την επιστροφή της. Οι κύκλοι μεταξύ αυτών είναι η καθυστέρηση της συνάρτησης. Τότε, σύμφωνα με τον simulator, οι καθυστερήσεις κάθε υλοποίησης είναι:

Υλοποίηση	float	long
Κύκλοι	6249	2655

Ανάλυση & Παρατηρήσεις

Από τον παραπάνω πίνακα είναι προφανές ότι η υλοποίηση με long έχει καλύτερη απόδοση. Πιο συγκεκριμένα, χρειάζεται περίπου το 42% του χρόνου της υλοποίησης με float. Φαίνεται ότι η χρήση διακριτής ή fixed point μορφή δεδομένων είναι πιο ελκυστική σε σχέση με την χρήση float, εφόσον το πρόβλημα το επιτρέπει. Παρότι δεν έχουν χρησιμοποιηθεί σε αυτό το εργαστήριο, ο compiler υποστηρίζει Fixed-Point μεταβλητές (_Fract, _Accum) διαφόρων μεγεθών και εύρους τιμών κάνοντας ευκολότερη την αποφυγή float μορφή δεδομένων.

Πηγές

AVR-GCC Type Layout

https://gcc.gnu.org/wiki/avr-gcc#Type_Layout

3x3 Matrix Multiplication

<https://ncalculators.com/matrix/3x3-matrix-multiplication-calculator.htm>