

## Προηγμένες Αρχιτεκτονικές Υπολογιστικών Συστημάτων 2015

### Προγραμματιστική Εργασία #2

#### Θέμα

Στην επεξεργασία εικόνας συχνά εμφανίζεται ο μετασχηματισμός, κατά τον οποίο κάθε pixel αντικαθίσταται από μια νέα τιμή υπολογισμένη από τα 8 γειτονικά pixels (και το ίδιο). Κατά τον μετασχηματισμό, κάθε αρχικό pixel πολλαπλασιάζεται με μια σταθερά float.

Έτσι, αν έχουμε τα γειτονικά pixels:

P0 P1 P2

P3 P4 P5

P6 P7 P8

και τις σταθερές μετασχηματισμού:

K0 K1 K2

K3 K4 K5

K6 K7 K8

τότε η νέα τιμή για το pixel P4 ισούται με:

$$\text{newP4} = P0 \cdot K0 + P1 \cdot K1 + P2 \cdot K2 + P3 \cdot K3 + P4 \cdot K4 + \\ + P5 \cdot K5 + P6 \cdot K6 + P7 \cdot K7 + P8 \cdot K8$$

Η διαδικασία επαναλαμβάνεται για όλα τα pixel που είναι στην ίδια γραμμή με το P4, για όλες τις γραμμές του πίνακα.

Για τις ανάγκες της άσκησης θεωρήστε ότι:

- Η εικόνα είναι **μονόχρωμη** (1 float αριθμός ανά pixel).
- Η εικόνα έχει ήδη μετατραπεί και φορτωθεί σε έναν πίνακα float NxM (αρχικοποιήστε με **τυχαίες τιμές** τον πίνακα πριν αρχίσετε τη μέτρηση του χρόνου).
- Οι νέες τιμές που υπολογίζετε αποθηκεύονται σε δεύτερο πίνακα με τις ίδιες διαστάσεις (αρχικοποιήστε και αυτόν τον πίνακα για να μεταφερθεί στην κρυφή μνήμη πριν αρχίσετε να μετράτε χρόνο).
- Τα M και N είναι τέτοια ώστε να βολεύουν στις πράξεις σας, π.χ. μπορείτε να θεωρήσετε ότι το N είναι πολλαπλάσιο του 4 ή οτιδήποτε άλλο σας βολεύει.
- Μπορείτε, αν θέλετε, να αγνοήσετε τον υπολογισμό των pixels στο περίγραμμα της εικόνας.
- Για τις ανάγκες της άσκησης, τα K0, K1, K2, K3, K5, K6, K7, K8 ισούνται με 0.5, ενώ το K4 είναι ίσο με 5.0 (μπορείτε δηλαδή να χρησιμοποιήσετε 2 σταθερές μόνο!).

Ζητείται η επιλογή όσο το δυνατόν αποδοτικότερων SIMD πράξεων (από το σετ SSE/SSE2) για την υλοποίηση του μετασχηματισμού. Σε αντίθεση με παραδείγματα που θα βρείτε on-line, τα οποία χρησιμοποιούν 3 ανεξάρτητες συνιστώσες χρώματος (RGB) και εξάγουν την παραλληλία από αυτό ακριβώς το γεγονός, εδώ η εικόνα είναι μονόχρωμη και ο παραλληλισμός δυσκολότερος! **Βεβαιωθείτε ότι το τροποποιημένο πρόγραμμά σας δεν είναι αργότερο από το αρχικό!**

#### Ζητούμενο

Ο στόχος της άσκησης είναι:

α) Να κατασκευάσετε δοκιμαστικό πρόγραμμα σε C χωρίς εντολές SSE, έτσι ώστε να μελετήσετε τον

χρόνο εκτέλεσης χωρίς πρόσθετες βελτιώσεις. Χρησιμοποιήστε πίνακες που χωράνε στην κρυφή μνήμη (L2) του συστήματός σας, έως π.χ. 1000x1000.

β) Κατασκευάστε το αντίστοιχο πρόγραμμα με εντολές SSE. Χρησιμοποιήστε ως βάση τον κώδικα που δόθηκε στο εργαστήριο. Μετρήστε τους αντίστοιχους χρόνους εκτέλεσης και εισάγετε περαιτέρω βελτιώσεις, αν χρειάζεται. **Βεβαιωθείτε ότι οι αναγνώσεις και εγγραφές είναι ευθυγραμμισμένες στα 16 bytes!**

Μπορείτε να αναζητήσετε on-line πληροφορία για τα intrinsics που θα χρησιμοποιήσετε, π.χ. στο “Intel® 64 and IA-32 Architectures Software Developer’s Manual” της Intel, Vol.2 (Instruction Set Reference, A-Z):

<http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html>

### **Παραδοτέο**

Θα πρέπει παραδώσετε (μέσω e-mail) α) το .c πρόγραμμά σας για τις δύο περιπτώσεις (με ή χωρίς SSE) και β) αναφορά pdf με συνοπτική περιγραφή του κώδικά σας, της μεθόδου παραλληλισμού που ακολουθήσατε και των αποτελεσμάτων που πήρατε, καθώς και τις πηγές που πιθανόν χρησιμοποιήσατε.

**Προθεσμία παράδοσης: Πέμπτη 19/3/2015 10:00.**