Κέρκυρα, 29/3/2017

ΕΡΓΑΣΙΑ ΣΤΟΝ ΠΑΡΑΛΛΗΛΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

του

ΓΙΑΝΝΙΟΥ ΑΝΤΩΝΙΟΥ
Α.Μ. **Π2013153**

Θέμα: «Προγραμματιστική Εργασία #1»

1^{ο} μέρος.

Το πρώτο πρόγραμμα no-sse.c υλοποιεί την επεξεργασία της εικόνας εισόδου (array a), παράγοντας την εικόνα εξόδου (array b), χωρίς την χρήση εντολών SSE/SSE2.

Η επεξεργασία της εικόνας, γίνεται ακολουθώντας τον τύπο που δίνεται στην εκφώνηση της άσκησης. Η εικόνα μας είναι αποθηκευμένη σε έναν μονοδιάστατο πίνακα a.

Ο πίνακας σαρώνεται από ένα διπλό βρόγχο με μεταβλητές i, j.

Οι συντεταγμένες του pixel που υπολογίζεται σε κάθε δεδομένη στιγμή, είναι i και j ενώ η θέση του, στον μονοδιάστατο πίνακα a δίνεται από τον τύπο i*M+j. (όπου M το πλήθος των στηλών).

Έτσι ο τύπος της εκφώνησης γίνεται με την χρήση των συντεταγμένων i kai j ως εξής:

```
b[i*M+j]=K0*a[(i-1)*M+(j-1)]+K1*a[(i-1)*M+j]+K2*a[(i-1)*M+(j+1)]+K3*a[i*M+(j-1)]
+K4*a[i*M+j]+K5*a[i*M+(j+1)]+K6*a[(i+1)*M+(j-1)]+K7*a[(i+1)*M+j]+K8*a[(i+1)*M+(j+1)];
```

Έτσι με τον διπλό βρόγχο, παράγεται ο πίνακας b που κάθε στοιχείο του είναι υπολογισμένο σύμφωνα με τον παραπάνω τύπο.

2^{ο} μέρος.

Το δεύτερο πρόγραμμα sse.c υλοποιεί την επεξεργασία της εικόνας ειδόδου με την χρήση εντολών SSE/SSE2.

Ο τρόπος σκέψης είναι ο ακόλουθος:

- Ο πίνακας εισόδου σαρώνεται πάλι με διπλό βρόγχο μεταβλητών i kai j (το i ξεκινά από την τιμή 1 μέχρι το N-1, ώστε να αγνοήσουμε τα περιθώρια, ενώ το j ξεκινά από την τιμή 0 έως M και έτσι δεν αγνοούμε τα πλαϊνά περιθώρια)

- Δημιουργούμε **3 δείκτες vfa1, vfa2, vfa3** τύπου __m128 που φορτώνουν, ο πρώτος την τετράδα pixels της γραμμής i-1, ο δεύτερος της γραμμής i, και ο τρίτος της γραμμής i+1.

- Έπειτα χρησιμοποιώ τις μάσκες vmask1 έως vmask5, που περιέχουν τους συντελεστές 5 και 0.5, για να κάνω τους μερικούς πολλαπλασιασμούς της κάθε τετράδας με την χρήση εντολών _mm_mul_ps. (αποθηκεύονται στην tempmul έως tempmul3) και έπειτα υπολογίζω την τιμή του πρώτου αριστερά pixel της μεσαίας τετράδας με το tempsum1 κάνοντας χρήση εντολών _mm_add_ps. Αθροίζω την τετράδα των επιμέρους όρων που περιέχει το tempsum1 με την χρήση του πίνακα e5.

- Ομοίως, κάνω το ίδιο και υπολογίζω και τα υπόλοιπα 3 pixel απλώς αλλάζοντας την μάσκα vmask1 με τις μάσκες vmask2, vmask3, vmask4 κτλ.

- Η μάσκα vmask5 περιέχει μόνο τον συντελεστή 0.5 και γι αυτό πολλαπλασιάζει μόνο την πάνω και κάτω τετράδα pixels.

- Οι μάσκες vmask1, vmask2, vmask3, vmask4 περιέχουν εκτός από το 0.5 και τον συντελεστή 5.0 ο οποίος αλλάζει θέση (επιλέγοντας την κατάλληλη μάσκα) ανάλογα με το pixel που υπολογίζω.

  Στο τέλος του βρόγχου αυξάνω τις vfa1, vfa2, vfa3 για να δείξουν στο επόμενο set τετράδων pixels.

Bug αλγορίθμου

Ο αλγόριθμος υλοποιεί το ζητούμενο της άσκησης, μόνο που δεν μπορεί να υπολογίσει τα γειτονικά pixels μεταξύ των γειτονικών τετράδων.

Δηλαδή αν έχω τα παρακάτω pixel σαν είσοδο

P1 P2 P3 P4 - P5 P6
Q1 Q2 Q3 Q4 - Q5 P5
R1 R2 R3 R4 - R5 P6

Και τα υπολογισμένα pixel εξόδου της μεσαίας τετράδας είναι

B1 B2 B3 B4 - B5 B6

Τότε μόνο το pixel B5 δεν περιλαμβάνει στον υπολογισμό του τα P4,Q4,R4.

Τα υπόλοιπα pixel υπολογίζονται κανονικά. Αυτό γίνεται, γιατί όταν αυξάνω τις τετράδες κατά μια θέση δυστυχώς αγνοούνται τα γειτονικά pixel της προηγούμενης τετράδας.