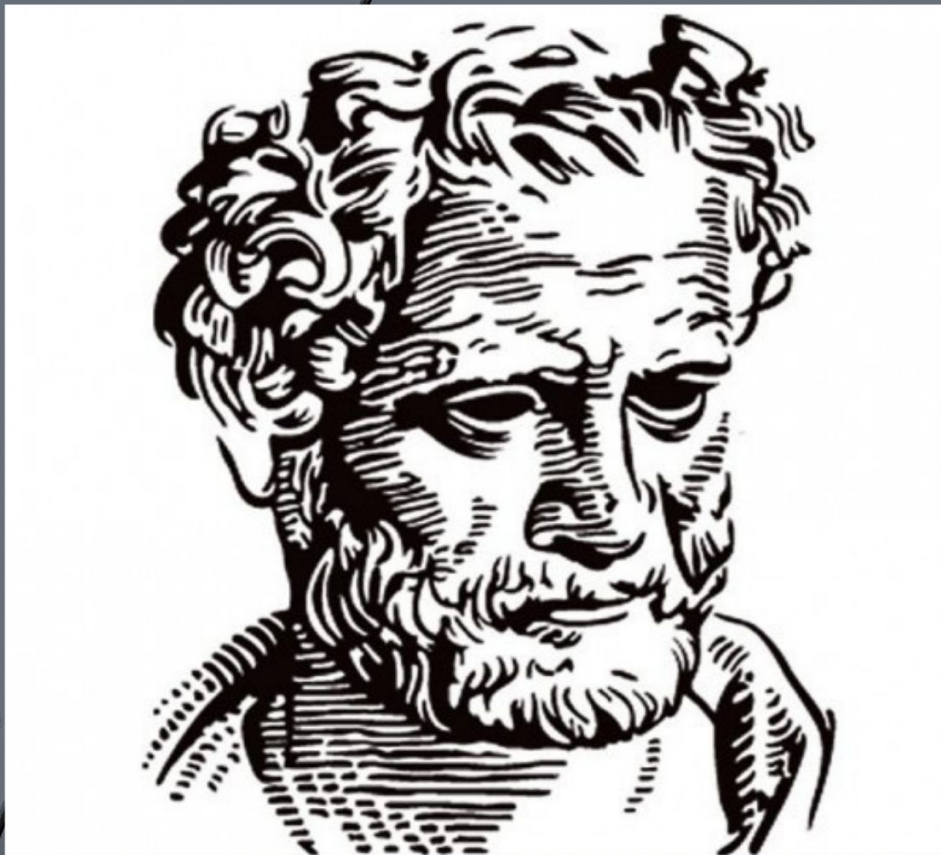


ΔΕΚΕΜΒΡΙΟΣ 2020

ΕΡΓΑΣΙΑ 1

Σχεδιασμός
Ενσωματωμένων
Συστημάτων

ΟΜΑΔΑ 39



Υπεύθυνος καθηγητής : Συρακούλης Γεώργιος

Φοιτητές : Κεφαλάς Γεώργιος (57406)

Τσίτσος Δημήτριος (57407)

1. Εισαγωγή

Στα πλαίσια του μαθήματος “Σχεδιασμός Ενσωματωμένων Κυκλωμάτων” υλοποιήθηκε η εργασία “Μετατροπή έγχρωμης εικόνας σε grayscale με τη μέθοδο της μέσης τιμής και με την τεχνική decomposition”. Οι αλγόριθμοι είναι βασισμένοι σε εικόνες μορφής raw με την κατάληξη “.yuv”. Γι’ αυτόν τον λόγο, ήταν απαραίτητη η μετατροπή των εικόνων σε “rgb” για την επεξεργασία των χρωμάτων. Τα λογισμικά τα οποία χρησιμοποιήθηκαν ήταν το “CodeWarrior” και το “AXD Debugger”, σύμφωνα με τα όσα διδάχθηκαν και στο πρώτο εργαστήριο του μαθήματος.

2. Κώδικας Εργασίας

Αρχικά, με βάση τον κώδικα που μας παραδόθηκε διαβάζουμε τα δεδομένα τα οποία έχουμε από το αρχείο εικόνας. Επιπλέον, πραγματοποιήσαμε μια μικρή αλλαγή στον τρόπο με τον οποίο αποθηκεύουμε τους πίνακες `current_u` και `current_v` ώστε αυτό να ταιριάζουν στον αντίστοιχο πίνακα `y` καθώς ο τελευταίος έχει το τετραπλάσιο μέγεθος από τους δύο προηγούμενους. Έτσι, τους προσαρμόσαμε με βάση τις οδηγίες που δόθηκαν ώστε κάθε μια τετράδα των τελικών πινάκων να αντιστοιχεί σε κάθε ένα στοιχείο των αρχικών πινάκων `u` και `v`. (Συνάρτηση `read()`)

Έπειτα, όσον αφορά την υλοποίηση του grayscale αναπτύχθηκαν οι συναρτήσεις `grayscale_averaging` και `grayscale_decomposition`. Σε κάθε μια από αυτές τις υλοποιήσεις, παίρνουμε διαδοχικά κάθε ένα στοιχείο των πινάκων που διαβάστηκαν από την εικόνα. Στη συνέχεια, υπολογίζουμε τις τιμές των `r`, `g` και `b` του κάθε pixel βάση των αντίστοιχων τιμών `y`, `u` και `v`. Σε αυτό το σημείο, προσέχουμε οι τελικές τιμές των `r`, `g` και `b` να μην ξεπερνούν το εύρος τιμών από 0 έως 255 για να μην υπάρχει κάποιο πρόβλημα στο τελικό αποτέλεσμα. Έτσι, με μια συνθήκη `if` γίνεται ο παραπάνω έλεγχος, και σε περίπτωση που παρατηρηθεί αυτό, δίνονται οι τιμές 0 και 255 αντίστοιχα.

Στη συνέχεια, διαχωρίζεται σε κάθε μια συνάρτηση ο υπολογισμός της τιμής `gray`. Στην πρώτη εκ των 2 συναρτήσεων, αυτή υπολογίζεται με τον μέσο όρο των τιμών `r`, `g` και `b` ($gray = (r + g + b)/3$) και έπειτα αυτή αποθηκεύεται στις τελικές τιμές των πινάκων `r`, `g` και `b`. Από την άλλη, με την μέθοδο `decomposition`, δηλώνεται αρχικά η τιμή του `r` ως `min/max` και έπειτα ελέγχονται και οι υπόλοιπες τιμές των `g` και `b` για να βρεθεί η μικρότερη ή μεγαλύτερη τιμή αυτών. Αυτή η τιμή στη συνέχεια θα είναι και η τελική τιμή των πινάκων.

Τέλος, πραγματοποιείται ο μετασχηματισμός του κάθε pixel από τις τιμές των πινάκων `r`, `g` και `b` σε `y`, `u` και `v` οι οποίες γράφονται στο τελικό αρχείο της εικόνας που εξάγεται από τον κώδικα. (Συνάρτηση `write()`)

3. Βελτιστοποιήσεις

Σκοπός της εργασίας είναι να υλοποιηθεί σε γλώσσα C ο προτεινόμενος αλγόριθμος και στη συνέχεια να εφαρμοστεί η μεθοδολογία βελτιστοποίησης μεταφοράς και αποθήκευσης δεδομένων στον αλγόριθμο αυτό.

Για την πιθανή βελτιστοποίηση του αλγορίθμου μας δοκιμάστηκαν οι εξής μέθοδοι :

I. Averaging

- Loop Fusion (υπάρχουσα μορφή κώδικα)

(Αποτελέσματα make)

Image component sizes						
	Code	RO Data	RW Data	ZI Data	Debug	
	4496	60	0	3775212	6372	Object Totals
	13972	314	0	300	6108	Library Totals
=====						
	Code	RO Data	RW Data	ZI Data	Debug	
	18468	374	0	3775512	12480	Grand Totals
=====						
	Total RO	Size(Code + RO Data)			18842 (18.40kB)	
	Total RW	Size(RW Data + ZI Data)			3775512 (3687.02kB)	
	Total ROM	Size(Code + RO Data + RW Data)			18842 (18.40kB)	
=====						

(Αποτελέσματα statistics)

Debugger Internals							
Internal Variables		Statistics					
Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics	255106950	368427870	281940531	67673773	70160007	0	419774311

- Loop Unrolling

Image component sizes						
	Code	RO Data	RW Data	ZI Data	Debug	
	5576	60	0	3775212	6528	Object Totals
	13972	314	0	300	6108	Library Totals
=====						
	Code	RO Data	RW Data	ZI Data	Debug	
	19548	374	0	3775512	12636	Grand Totals
=====						
	Total RO	Size(Code + RO Data)			19922 (19.46kB)	
	Total RW	Size(RW Data + ZI Data)			3775512 (3687.02kB)	
	Total ROM	Size(Code + RO Data + RW Data)			19922 (19.46kB)	
=====						

(Αποτελέσματα statistics)

Debugger Internals							
Internal Variables		Statistics					
Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics	134232256	197357040	146855957	39743264	36680861	0	223280082

- Loop Interchange

(Αποτελέσματα make)

```

Image component sizes

Code      RO Data      RW Data      ZI Data      Debug
4480           60           0      3775212      6388  Object Totals
13972         314           0           300      6108  Library Totals
=====

Code      RO Data      RW Data      ZI Data      Debug
18452         374           0      3775512      12496  Grand Totals
=====

Total RO  Size(Code + RO Data)           18826 ( 18.38kB)
Total RW  Size(RW Data + ZI Data)       3775512 (3687.02kB)
Total ROM Size(Code + RO Data + RW Data) 18826 ( 18.38kB)
=====

```

(Αποτελέσματα statistics)

Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics	257570180	371757790	284113471	68542223	70625893	0	423281587

- Loop Tiling

(Αποτελέσματα make)

```

Image component sizes

Code      RO Data      RW Data      ZI Data      Debug
4612      60      0      3775212      6592      Object Totals
13972     314      0      300      6108      Library Totals
=====

Code      RO Data      RW Data      ZI Data      Debug
18584     374      0      3775512      12700     Grand Totals
=====

Total RO   Size(Code + RO Data)      18958 ( 18.51kB)
Total RW   Size(RW Data + ZI Data) 3775512 (3687.02kB)
Total ROM  Size(Code + RO Data + RW Data) 18958 ( 18.51kB)
=====

```

(Αποτελέσματα statistics)

Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics	255780250	370016094	282597297	68218142	70564468	0	421379907

- Loop Inversion

(Αποτελέσματα make)

```

Image component sizes

Code      RO Data      RW Data      ZI Data      Debug
4496      60                0      3775212      6404      Object Totals
13972     314                0        300      6108      Library Totals
=====

Code      RO Data      RW Data      ZI Data      Debug
18468     374                0      3775512      12512     Grand Totals
=====

Total RO   Size(Code + RO Data)                18842 ( 18.40kB)
Total RW   Size(RW Data + ZI Data)            3775512 (3687.02kB)
Total ROM  Size(Code + RO Data + RW Data)  18842 ( 18.40kB)
=====

```

(Αποτελέσματα statistics)

Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics	255106950	368427870	281940531	67673773	70160007	0	419774311

- Συνδυαστική μέθοδος (Loop Unrolling & Loop Interchange)

Χρησιμοποιήσαμε μία συνδυαστική μέθοδος, με τη χρήση των loop unrolling & loop interchange :

(Αποτελέσματα make)

```

Image component sizes

Code      RO Data      RW Data      ZI Data      Debug
5556      60              0      3775212      6532      Object Totals
13972     314            0      300          6108      Library Totals
=====

Code      RO Data      RW Data      ZI Data      Debug
19528     374            0      3775512      12640     Grand Totals
=====

Total RO   Size(Code + RO Data)          19902 ( 19.44kB)
Total RW   Size(RW Data + ZI Data)     3775512 (3687.02kB)
Total ROM  Size(Code + RO Data + RW Data)  19902 ( 19.44kB)
=====

```

(Αποτελέσματα statistics)

Debugger Internals							
Internal Variables				Statistics			
Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics	134665271	197533700	147289027	39632989	36578964	0	223500980

II. Decomposition

Χρησιμοποιήθηκε ο αλγόριθμος *decomposition max*.

- Loop Fusion (υπάρχουσα μορφή κώδικα)

(Αποτελέσματα make)

Image component sizes						
	Code	RO Data	RW Data	ZI Data	Debug	
	3328	60	0	3775212	6200	Object Totals
	13972	314	0	300	6108	Library Totals
=====						
	Code	RO Data	RW Data	ZI Data	Debug	
	17300	374	0	3775512	12308	Grand Totals
=====						
	Total RO	Size(Code + RO Data)			17674	(17.26kB)
	Total RW	Size(RW Data + ZI Data)			3775512	(3687.02kB)
	Total ROM	Size(Code + RO Data + RW Data)			17674	(17.26kB)
=====						

(Αποτελέσματα statistics)

Debugger Internals							
Internal Variables				Statistics			
Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics	249738174	364122242	276172294	68382209	70929305	0	415483808

- Loop Unrolling

(Αποτελέσματα make)

Image component sizes						
Code	RO Data	RW Data	ZI Data	Debug		
4496	60	0	3775212	6380	Object Totals	
13972	314	0	300	6108	Library Totals	
Code	RO Data	RW Data	ZI Data	Debug		
18468	374	0	3775512	12488	Grand Totals	
Total RO Size(Code + RO Data)				18842	(18.40kB)	
Total RW Size(RW Data + ZI Data)				3775512	(3687.02kB)	
Total ROM Size(Code + RO Data + RW Data)				18842	(18.40kB)	

(Αποτελέσματα statistics)

Debugger Internals							
Internal Variables				Statistics			
Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics131877776		196440921	144519058	40387608	37466282	0	222372948

- Loop Interchange

(Αποτελέσματα make)

Image component sizes						
Code	RO Data	RW Data	ZI Data	Debug		
3312	60	0	3775212	6200	Object Totals	
13972	314	0	300	6108	Library Totals	
Code	RO Data	RW Data	ZI Data	Debug		
17284	374	0	3775512	12308	Grand Totals	
Total RO Size(Code + RO Data)				17658	(17.24kB)	
Total RW Size(RW Data + ZI Data)				3775512	(3687.02kB)	
Total ROM Size(Code + RO Data + RW Data)				17658	(17.24kB)	

(Αποτελέσματα statistics)

Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics	252201404	366871362	278345234	68960259	71104791	0	418410284

- Loop Tiling

(Αποτελέσματα make)

```

Image component sizes

Code      RO Data      RW Data      ZI Data      Debug
3440      60                0      3775212      6376      Object Totals
13972     314                0        300      6108      Library Totals
=====

Code      RO Data      RW Data      ZI Data      Debug
17412     374                0      3775512      12484     Grand Totals
=====

Total RO   Size(Code + RO Data)                17786 ( 17.37kB)
Total RW   Size(RW Data + ZI Data)            3775512 (3687.02kB)
Total ROM  Size(Code + RO Data + RW Data)  17786 ( 17.37kB)
=====

```

(Αποτελέσματα statistics)

Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics	250409989	365708981	276827575	68926578	71333766	0	417087919

- Loop Inversion

(Αποτελέσματα make)

```

=====
Image component sizes
=====
Code      RO Data    RW Data    ZI Data    Debug
3328      60           0          3775212    6228      Object Totals
13972     314          0          300        6108      Library Totals
=====
Code      RO Data    RW Data    ZI Data    Debug
17300     374         0          3775512    12336     Grand Totals
=====
Total RO   Size(Code + RO Data)          17674 ( 17.26kB)
Total RW   Size(RW Data + ZI Data)      3775512 (3687.02kB)
Total ROM  Size(Code + RO Data + RW Data) 17674 ( 17.26kB)
=====

```


(Αποτελέσματα statistics)

Debugger Internals							
Internal Variables				Statistics			
Refere...	Instru...	Core_C...	S_Cycles	N_Cycles	I_Cycles	C_Cycles	Total
\$statistics	249738174	364122242	276172294	68382209	70929305	0	415483808

- Συνδυαστική μέθοδος (Loop Unrolling & Loop Interchange)

(Αποτελέσματα make)

Image component sizes						
	Code	RO Data	RW Data	ZI Data	Debug	
	4436	60	0	3775212	6384	Object Totals
	13972	314	0	300	6108	Library Totals
=====						
	Code	RO Data	RW Data	ZI Data	Debug	
	18408	374	0	3775512	12492	Grand Totals
=====						
	Total RO	Size(Code + RO Data)			18782	(18.34kB)
	Total RW	Size(RW Data + ZI Data)			3775512	(3687.02kB)
	Total ROM	Size(Code + RO Data + RW Data)			18782	(18.34kB)
=====						

(Αποτελέσματα statistics)

Debugger Internals							
Internal Variables				Statistics			
Ref...	Instruc...	Core_C...	S_Cycles	N_Cy...	I_Cy...	C_...	Total
\$statist	132020556	196000811	144734658	39986933	37255485	0	221977076

Συμπεράσματα :

- Ο κώδικας μας είναι γραμμένος με μορφή Loop Fusion, δηλαδή οι for είναι συνδυασμένες όλες μαζί.
- Και στις δύο περιπτώσεις πιο σημαντική αλλαγή βλέπουμε στη μορφή Unrolling, η οποία μειώνει σημαντικά τους κύκλους του ρολογιού.
- Στην περίπτωση του Loop Interchange αυξάνονται οι κύκλοι του ρολογιού και αυτό οφείλεται στην αρχιτεκτονική της γλώσσας C. στο πως γίνεται η προσπέλαση των στηλών και γραμμών ενός πίνακα.
- Στην περίπτωση του Loop Inversion βλέπουμε ότι οι κύκλοι ρολογιού παραμένουν ίδιοι με το Loop Fusion, πράγμα αναμενόμενο μιας και δεν υπάρχει ανάγκη για pipelining.
- Στην περίπτωση του Loop Tiling, προσθέτει κάποιου κύκλους εξαιτίας των for των οποίων προστίθενται.

4. Πίνακας Δεδομένων & Αριθμός Προσπελάσεων

Για τον πίνακα δεδομένων, έχουμε 330×440 στοιχεία του πίνακα τον οποίο διαβάζουμε οπότε συνολικά 145200. Επιπλέον, επειδή αυτά τα στοιχεία είναι αποθηκευμένα με την μορφή ακεραίων, αντιστοιχούν σε 32 bit το κάθε ένα οπότε έχουμε συνολικά 4646400 bits στον πίνακα δεδομένων. Τέλος, επειδή οι προσπελάσεις οι οποίες πραγματοποιούνται στον πίνακα είναι 3 (μία για την μετατροπή από YUV σε RGB, μία για τον grayscale και μία για την μετατροπή από RGB σε YUV). Άρα, με βάση και τα προηγούμενα δεδομένα έχουμε συνολικά $3 \times 145200 = 435600$ προσπελάσεις.

5. Αποτελέσματα εκτέλεσης

Αρχική Εικόνα



Averaging Εικόνα



Decomposition(max) Εικόνα



Decomposition(min) Εικόνα

