



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Ενσωματωμένα Συστήματα

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο 5

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα

1.Σκοπός της άσκησης.....	4
2.Παραδοτέα.....	4
3.Εφαρμογή Μετασχηματισμών Global Loop.....	4
4.Τροποποίηση κώδικα για μέτρηση SNR.....	4
5.Ο μετασχηματισμός Loop merging στους βρόχους με δείκτη i.....	5
6.Ο μετασχηματισμός Loop merging στους βρόχους με δείκτη k.....	5
7.Ο μετασχηματισμός loop merging στους βρόχους με δείκτη l.....	5
8.Αποκοπή των περιττών εκτελέσεων.....	5
9.Εφαρμογή των μετασχηματισμών επαναχρησιμοποίησης δεδομένων.....	7
10.Ο μετασχηματισμός επαναχρησιμοποίησης δεδομένων με την εισαγωγή του πίνακα current_line	9
11.Ο μετασχηματισμός επαναχρησιμοποίησης δεδομένων με την εισαγωγή του πίνακα block.....	9
12. Ο μετασχηματισμός επαναχρησιμοποίησης δεδομένων με την εισαγωγή του πίνακα current_line και block.....	10

1. Σκοπός της άσκησης

- Βελτιστοποίηση Εφαρμογής Ενσωματωμένων Συστημάτων με τη μεθοδολογία **DTSE** (*Data Transfer and Storage Exploration*) – **Μέρος Α**.

(A) 2 ερωτήσεις

(C) 7 ασκήσεις/προγράμματα

Διαβάστε όλο το φυλλάδιο με το θεωρητικό υπόβαθρο της άσκησης. Πριν συνεχίσετε παρακάτω να έχετε κατανοήσει τους αλγοριθμικούς μετασχηματισμούς και τον τρόπο υλοποίησης.

Σε αυτό το εργαστήριο θα βελτιστοποιήσουμε τον αλγόριθμο PHODS. Κατεβάστε το αρχείο **phods.zip**, από τη σελίδα του μαθήματος.

2. Παραδοτέαⁱ

- **Παραδοτέο C1:** Screenshot που φαίνεται το SNR
- **Παραδοτέο C2:** Κώδικας
- **Παραδοτέο C3:** Κώδικας
- **Παραδοτέο C4:** Κώδικας
- **Παραδοτέο C5:** Κώδικας
- **Παραδοτέο C6:** Κώδικας
- **Παραδοτέο C7:** Κώδικας

3. Εφαρμογή Μετασχηματισμών Global Loop

Η μεθοδολογία βελτιστοποίησης θα εφαρμοστεί σταδιακά στο κύριο τμήμα της εφαρμογής. Εδώ πρέπει να αναφερθεί ότι δεν θα γίνουν βελτιστοποιήσεις στις συναρτήσεις εισόδου. Το τμήμα του κώδικα που αποτελεί αντικείμενο μελέτης και εφαρμογής των μετασχηματισμών είναι η συνάρτηση «`phods_motion_estimation`».

Σε κάθε βήμα πρέπει:

(α) να βεβαιώνετε την ορθότητα των μετασχηματισμών που έχετε κάνει χωρίς warning/errors,

(β) να κάνετε προσομοίωση τον αλγόριθμο και να μετράτε τον αριθμό των εντολών που εκτελέστηκαν, τον αριθμό των προσβάσεων στους πίνακες (*οδηγίες στην τελευταία σελίδα*) και τους κύκλους που απαιτήθηκαν.

Θα πρέπει να μετράτε το SNR κάθε φορά για να επιβεβαιώνετε την ορθή λειτουργία. Όλοι οι κώδικες που θα παραδώσετε θα πρέπει να έχουν ονομασία της μορφής `phods.XX.c`, με `XX` τον αριθμό βήματος.

4. Τροποποίηση κώδικα για μέτρηση SNR

Χρησιμοποιήστε το αρχείο **aux-functions.c**, το οποίο έχει δυο συναρτήσεις. Μια για την ανακατασκευή της εικόνας από την προηγούμενη με τη χρήση των motion vectors (**motion compensation()**) και μια για τον υπολογισμό της διαφοράς, ανάμεσα σε δυο εικόνες (**snr()**). Θα πρέπει μετά τον υπολογισμό των motion vectors να καλείτε το motion_compensation για τη δημιουργία ενός νέου frame και στη συνέχεια να συγκρίνετε αυτό το νέο frame που κατασκευάστηκε με το frame που έπρεπε να είχε κατασκευαστεί (δηλαδή το *current frame*). Τοποθετήστε τον κώδικα, εκτελέστε το και δείτε το SNR. Θα πρέπει να είναι πάνω από 25.

Παραδοτέο C1: Screenshot που φαίνεται το SNR

5. Ο μετασχηματισμός Loop merging στους βρόχους με δείκτη i

Στο πρώτο στάδιο θα εφαρμοστεί ο μετασχηματισμός loop merging για τους δύο βρόχους `for(i=-S;i<S+1;i+=S)`. Επίσης στο διπλό βρόχο αρχικοποίησης των μεταβλητών `vectors_x` και `vectors_y` αλλάζουμε τις μεταβλητές των βρόχων από `i, j` σε `x, y`, ώστε να μπορεί να γίνει μετασχηματισμός συγχώνευσης με τους βρόχους που ακολουθούν. Ο μετασχηματισμός αυτός για να εφαρμοστεί θα πρέπει να μην παραβιάζονται οι εξαρτήσεις δεδομένων. Εξάρτηση δεδομένων έχουμε όταν κάποιες τιμές ενός πίνακα που δημιουργούνται και χρησιμοποιούνται στη συνέχεια του αλγορίθμου. Παραβίαση θα έχουμε όταν χρησιμοποιήσουμε μια τιμή ενός πίνακα πριν αυτή δημιουργηθεί. Στο συγκεκριμένο μετασχηματισμό δεν υπάρχει κάποια εξάρτηση, οπότε μπορούμε να εφαρμόσουμε το μετασχηματισμό. Ο αλγόριθμος μετά την

Οδηγίες:

- Μετονομασία των μεταβλητών βρόχου στην αρχικοποίηση των `vectors_x, vectors_y`
- Εφαρμογή loop merging στους 2 βρόχους `for(i=-S;i<S+1;i++)`

Παραδοτέο C2: Ο κώδικας ως αυτό το σημείο

6. Ο μετασχηματισμός Loop merging στους βρόχους με δείκτη k

Στο επόμενο βήμα, με ένα δεύτερο μετασχηματισμό loop merging συγχωνεύουμε τις εντολές που περιέχουν οι βρόχοι με δείκτες k . Έτσι οι εντολές των δύο ακολουθιακών βρόχων με δείκτη k ενσωματώνονται σε ένα βρόχο που εμπεριέχει τις εντολές των δύο αρχικών.

Μετά το προηγούμενο βήμα βλέπουμε ότι μπορούμε να συγχωνεύσουμε και τους βρόχους με δείκτη k .

Οδηγίες:

- Εφαρμογή loop merging στους 2 βρόχους με τη μεταβλητή k

Παραδοτέο C3: Ο κώδικας ως αυτό το σημείο

7. Ο μετασχηματισμός loop merging στους βρόχους με δείκτη l

Παρατηρούμε ότι τώρα μπορεί να γίνει συγχώνευση των δυο βρόχων με τη μεταβλητή l . Έτσι, μεταφέρονται οι εντολές του δεύτερου βρόχου στο εσωτερικό του πρώτου, με συνέπεια ο αλγόριθμος να παίρνει μια πιο συμπαγή μορφή. Επίσης, βλέπουμε ότι η εντολή ανάγνωσης $p1=current[B*x+k][B*y+1]$; εμφανίζεται δυο φορές μέσα στις εντολές του νέου βρόχου (μια φορά σε κάθε ένα από τους αρχικούς), οπότε αυτόματα διαγράφεται η επανάληψη της εντολής.

8. Αποκοπή των περιττών εκτελέσεων

Στον παραπάνω αλγόριθμο παρατηρούμε ότι για $i=0$ to $p2$ είναι ίσο με το $q2$ οπότε δεν είναι απαραίτητο να το υπολογίσουμε ξανά. Για να μειώσουμε τις περιττές προσπελάσεις στη μνήμη αλλά και τις περιττές εκτελέσεις εντολών εισάγουμε μια εντολή ελέγχου $if(i==0) disty=distx$; Οπότε αυτόματα παρακάμπτει τον υπολογισμό του $disty$ όταν το $i==0$. Έτσι ο αλγόριθμος μετασχηματίζεται ως εξής:

Οδηγίες:

- Να τοποθετήσετε την οριακή περίπτωση `if (i==0)` για να μειωθούν οι περιττές εκτελέσεις, μαζί με τις ανάλογες τροποποιήσεις.

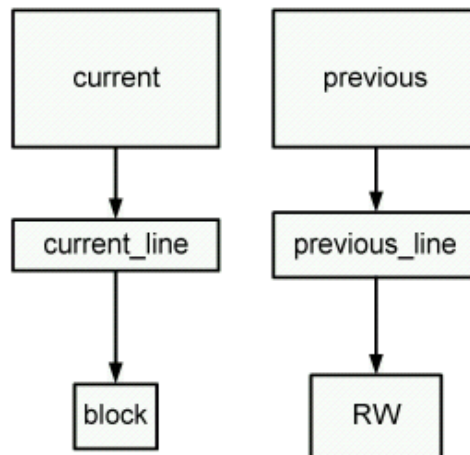
Παραδοτέο C4: Ο κώδικας ως αυτό το σημείο

9. Εφαρμογή των μετασχηματισμών επαναχρησιμοποίησης δεδομένων

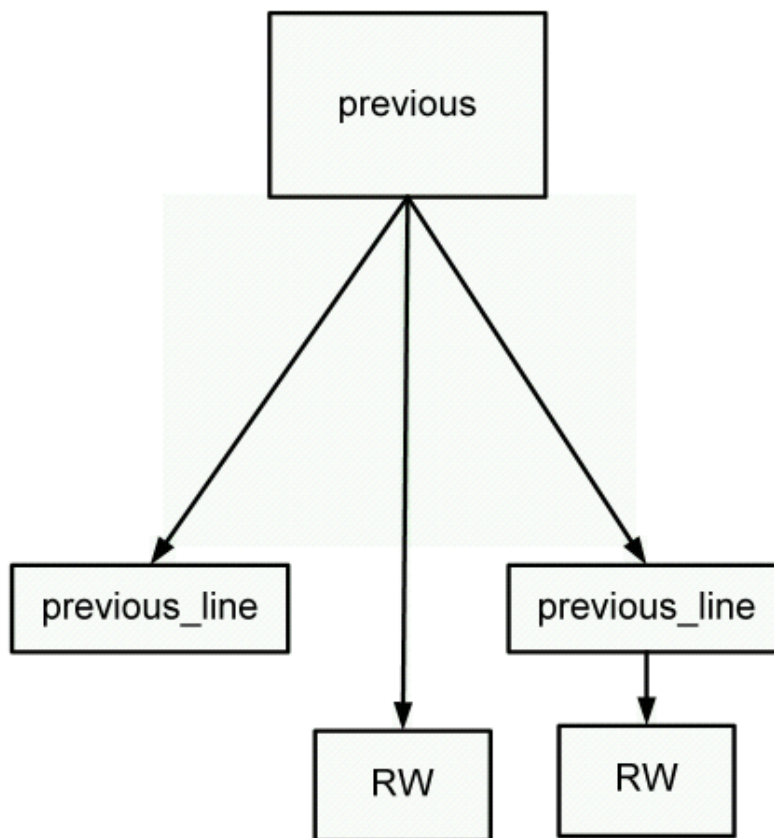
Μετά την εφαρμογή των μετασχηματισμών βρόχου ο αλγόριθμος έχει πάρει την κανονικοποιημένη μορφή των συγχωνευμένων βρόχων. Το επόμενο στάδιο της μεθοδολογίας είναι η εφαρμογή των μετασχηματισμών επαναχρησιμοποίησης δεδομένων. Αυτό που θέλουμε να πετύχουμε είναι να μειωθούν οι προσπελάσεις στους πίνακες δεδομένων `current` και `previous` οι οποίοι έχουν μεγάλο μέγεθος. Η μείωση θα γίνει με την εισαγωγή μικρότερου μεγέθους πινάκων δεδομένων για την προσωρινή αποθήκευση τμημάτων των αρχικών πινάκων. Οι νέοι πίνακες θα εισαχθούν στα σημεία του αλγορίθμου μετά από κάθε βρόχο και θα αποθηκεύουν τμήμα των δεδομένων ενός από τους αρχικούς πίνακες δεδομένων, τα οποία θα χρησιμοποιηθούν στο αντίστοιχο βρόχο.

Σε κάθε εκτέλεση του βρόχου με δείκτη `x`, τα δεδομένα που χρησιμοποιούνται από τους δύο πίνακες δεδομένων (`current` και `previous`) μπορούν να αποθηκευθούν σε δύο buffers με διαστάσεις $M \times B$ για τον `current` πίνακα και τον ονομάζουμε `current_line`. Ενώ για τον `previous` πίνακα εισάγουμε έναν buffer με διαστάσεις $M \times (B+2*p)$ (όπου $p=S+S/2+S/4$) και τον ονομάζουμε `previous_line`.

Εσωτερικά του βρόχου με δείκτη `y`, τα δεδομένα που χρησιμοποιούνται από τους αρχικούς πίνακες δεδομένων μπορούν να αποθηκευτούν προσωρινά σε δυο buffers διαστάσεων $B \times B$ και $(B+2*p) \times (B+2*p)$. Αντίστοιχα, μελετώντας τους εσωτερικότερους βρόχους δημιουργούμε μικρότερου μεγέθους buffers για την προσωρινή αποθήκευση τμημάτων δεδομένων από τους αρχικούς πίνακες. Τέλος, το μικρότερο μέγεθος buffer που μπορεί να χρησιμοποιηθεί για την προσωρινή αποθήκευση του πίνακα `current` είναι το `block` με διαστάσεις $B \times B$. Με την ολοκλήρωση της διερεύνησης πιθανών πινάκων για την προσωρινή αποθήκευση δεδομένων σχηματίζουμε τις αλυσίδες περιγραφής των πιθανών δομών μνήμης (Σχήμα 5). Με την διερεύνηση όλων των πιθανών τρόπων χρησιμοποίησης των προσωρινών πινάκων αποθήκευσης δημιουργούνται τα δέντρα αντιγραφής τα οποία παρουσιάζονται στα Σχήματα 6 και 7 για τον κλάδο του `current` και `previous` πίνακα αντίστοιχα.



Σχήμα 5. Οι αλυσίδες αντιγραφής δεδομένων από του αρχικούς πίνακες (current και previous) δεδομένων σε μικρότερου μεγέθους πίνακες δεδομένων τους current_line, block και τους previous_line, RW αντίστοιχα.

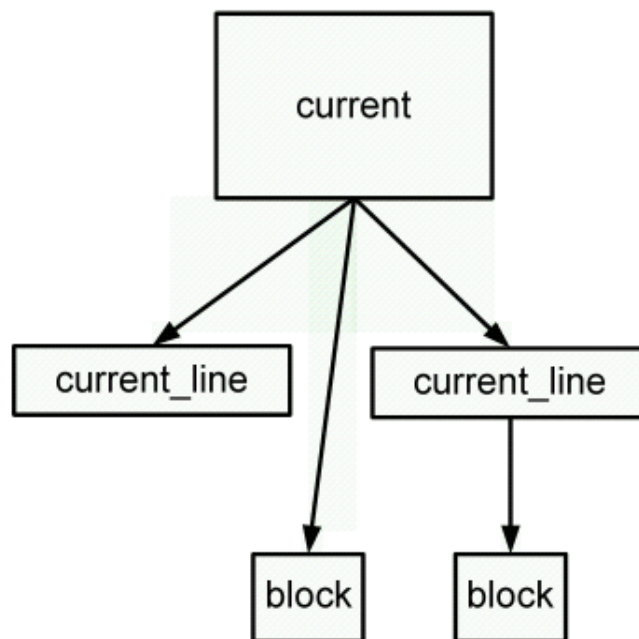


Σχήμα 7. Το δέντρο με όλους τους δυνατούς συνδυασμούς αντιγραφής δεδομένων του πίνακα previous

Η εφαρμογή των μετασχηματισμών επαναχρησιμοποίησης δεδομένων θα γίνει για κάθε ένα κλάδο των δένδρων αντιγραφής δεδομένων που παρουσιάζονται στα σχήματα 6 και 7. Έτσι κάθε κλάδος αποτελεί ένα ξεχωριστό μετασχηματισμό επαναχρησιμοποίησης δεδομένων.

10. Ο μετασχηματισμός επαναχρησιμοποίησης δεδομένων με την εισαγωγή του πίνακα `current_line`

Αρχίζοντας τους μετασχηματισμούς από το πίνακα δεδομένων `current` ο πρώτος μετασχηματισμός θα εισάγει τον πίνακα (buffer) `current_line` (Σχήμα 6). Η διαδικασία είναι η ακόλουθη, τμήμα των δεδομένων από το πίνακα `current` θα αντιγραφούν στο πίνακα `current_line` και στην εφαρμογή θα χρησιμοποιηθούν μέσω του πίνακα `current_line` που έχει μικρότερο μέγεθος. Ο αλγόριθμος του πρώτου μετασχηματισμού επαναχρησιμοποίησης δεδομένων θα είναι ο εξής:



Σχήμα 6. Το δέντρο με όλους τους δυνατούς συνδυασμούς αντιγραφής δεδομένων του πίνακα `current`

Οδηγίες:

- Δηλώστε τον πίνακα `current_line` με διαστάσεις `[M][B]`
- Ακριβώς μόλις εισέλθουμε στο βρόχο `x`, δημιουργήστε ένα βρόχο από `j=0` έως `M` και μέσα σε αυτόν από `i=0` έως `B` για να αντιγράψετε τα στοιχεία από τον πίνακα `current` στον πίνακα `current_line` ως εξής:

```
current_line[i][j]=current[B*x+i][j];
```

- Λίγο παρακάτω, στο σημείο που γίνεται η ανάγνωση του πίνακα `current` και η τοποθέτηση του στοιχείου στο `p1`, να αντικατασταθεί με ανάγνωση στον πίνακα `current_line` και κατάλληλη τροποποίηση στους δείκτες.

11. Ο μετασχηματισμός επαναχρησιμοποίησης δεδομένων με την εισαγωγή του πίνακα `block`

Ο επόμενος μετασχηματισμός επαναχρησιμοποίησης δεδομένων εισάγει τον πίνακα πρόχειρης αποθήκευσης `block`. Ο πίνακας `block` θα αποθηκεύει προσωρινά τμήμα δεδομένων από τον `current` και θα το χρησιμοποιεί ο επεξεργαστής μέσω του πίνακα `buffer`. Οπότε ο αλγόριθμος μετασχηματίζεται ως ακολούθως:

Οδηγίες:

- Να δηλωθεί ο πίνακας `block` διαστάσεων $B \times B$
- Μέσα στο `while (S>0)` να τοποθετήσετε ένα διπλό βρόχο από $k=0$ έως B και $l=0$ έως B , ώστε στο `block[k][l]` να τοποθετείται η τιμή `current[B*x+k][B*y+l]`
- Να τροποποιηθεί η γραμμή που γράφουμε στο `p1`, ώστε να γράφουμε σε αυτό τιμή που έρχεται από το `block`.

12. Ο μετασχηματισμός επαναχρησιμοποίησης δεδομένων με την εισαγωγή του πίνακα `current_line` και `block`.

Ο τρίτος μετασχηματισμός επαναχρησιμοποίησης δεδομένων εισάγει και τους δυο προηγούμενους πίνακες δεδομένων τους `current_line` και `block`. Η διαδικασία αντιγραφής δεδομένων θα ακολουθεί την ιεραρχία μνήμης, έτσι τμήμα δεδομένων από τον πίνακα `current` θα αντιγράφεται στο `current_line` και στη συνέχεια, τμήμα από το `current_line` θα αντιγράφεται στο `block`. Μέσω του πίνακα `block` θα γίνεται η ανάγνωση των δεδομένων από το κριτήριο του αλγορίθμου. Ο αλγόριθμος του τρίτου μετασχηματισμού θα είναι ο παρακάτω.

Οδηγίες:

- Να δηλωθούν οι πίνακες `current_line[B][M]` και `block`
- Να τοποθετηθεί, όπως προηγουμένως, στην αρχή του `x` βρόχου, η μεταφορά από τον `current` στο `current_line`.
- Να τοποθετηθεί μέσα στο `y` βρόχο, η μεταφορά από τον `current[k][B*y+1]` στο `block`.
- Να τροποποιηθεί κατάλληλα η γραμμή που γράφουμε στο στοιχείο `p1`.

Παραδοτέο C7: Ο κώδικας ως αυτό το σημείο

Ερώτηση A1: Απαντήστε με δικιά σας λόγια: Γιατί οι παραπάνω μετασχηματισμοί βελτιστοποιούν το συγκεκριμένο πρόγραμμα;

Ερώτηση A2: Υπάρχει περίπτωση με την εφαρμογή των ανωτέρω μετασχηματισμών σε μια διαφορετική εφαρμογή, αντί για βελτιστοποίηση να επιτευχθεί αντίθετο αποτέλεσμα, και γιατί;

→ Τέλος A μέρους εργαστηρίου. Στο επόμενο εργαστήριο θα συνεχίσετε τη βελτιστοποίηση της εφαρμογής `rhods`. Θα χρησιμοποιήσετε τους κώδικες που δημιουργήθηκαν σε αυτό το εργαστήριο.

i Παραδοτέα αυτού του εργαστηρίου:

- Κώδικες για κάθε βήμα βελτιστοποίησης
- Προσομοίωση κάθε βήματος στο ARMulator (*ARM7TDMI*) και εξαγωγή μετρήσεων ως προς εντολές που εκτελέστηκαν, κύκλοι εκτέλεσης, αριθμό προσβάσεων σε πίνακες και τιμή SNR. Αν μπορείτε, να μετρήσετε μόνο τις εντολές και τους κύκλους από τη συνάρτηση `rhods_motion_estimation()`, αφού αυτή είναι η βασική συνάρτηση της εφαρμογής μας.
- Για να μετρήσουμε το συνολικό αριθμό των προσβάσεων σε κάθε πίνακα τοποθετήστε μετρητές προσβάσεων σε κάθε πίνακα. Για παράδειγμα, θα χρησιμοποιήσουμε τη μεταβλητή `counter_current_read` `counter_current_write` για να μετρήσουμε τις προσβάσεις στον πίνακα `counter_current`. Κάθε φορά που γίνεται εγγραφή στον πίνακα `counter_current` (δηλαδή βρίσκεται στην αριστερή πλευρά της εντολής C), τότε θα πρέπει εμείς από κάτω να γράψουμε **`counter_current_write++`**; Κάθε φορά που γίνεται ανάγνωση στον πίνακα `counter` (δηλαδή, βρίσκεται στη δεξιά πλευρά της εντολής C), τότε θα πρέπει εμείς από κάτω να γράφουμε **`counter_current_read++`**; Στο τέλος, έξω από τη συνάρτηση `rhods_motion_estimation()`, θα πρέπει να εκτυπώνονται όλες αυτές οι τιμές, όπως παρακάτω

“Number of read accesses on current: XXXXX” “Number of write accesses on current: XXXXX”

για κάθε πίνακα.

➔ Υπάρχουν και άλλοι τρόποι καταμέτρησης των προσβάσεων. Αν γνωρίζετε κάποιον άλλο τρόπο, μπορείτε να τον χρησιμοποιήσετε (π.χ. `gprof-gcov`).