**Πανεπιστήμιο Πειραιώς**

Τμήμα πληροφορικης

# Προηγμένη Αρχιτεκτονική Υπολογιστών

2η Εργασία 2021-2022

# Θέμα

Ανάλυση της απόδοσης της κρυφής μνήμης δεδομένων με χρήση του προσομοιωτή MARS

# Φοιτητές

**Όνομα:** Παπαδοπούλου Παρασκευή
**ΑΜ**: Π19132
**Email**: voulapapa6@gmail.com

**Όνομα**: Αργυρίου Κωνσταντίνος
**ΑΜ**: Π19017
**Email**: argyrioukost@gmail.com

# Μέρος 1ο

## Σκοπός

Ο σκοπός του 1ου μέρους της εργασίας είναι να εκτελέσουμε και να υπολογίσουμε στατιστικά του προγράμματος της 1ης εργασίας με χρήση της κρυφής μνήμης.

## Απαντήσεις

1. Με βάση τον προσομοιωτή **mars** το πρόγραμμα που εκτελέστηκε έχει ρυθμό αστοχίας ίσο με **92%** η αλλιώς **miss rate** = **92%**.

2. Έχουμε:

   **CPU Execution Cycles** = 5280 Cycles // Από την 1η εργασία.
   **Memory Accesses** = 1209 // Από τον **mars**.
   **Miss Rate** = 1% // Από την εκφώνηση.
   **Miss Penalty** = 100 Cycles // Από την εκφώνηση
   // Από την θεωρία.
   **Memory Stall Cycles** = **Memory Accesses x Miss Rate x Miss Penalty**
   = (1209 x 0.92 + 3636 x 0.01) x 100 = 114864 Cycles

   Επομένως ο συνολικός χρόνος εκτέλεσης του προγράμματος σε κύκλους ρολογιού είναι:

   // Από την θεωρία.
   **CPU Time** = **CPU Execution Cycles + Memory Stall Cycles**
   = 5280 + 114864 = 120144 Cycles

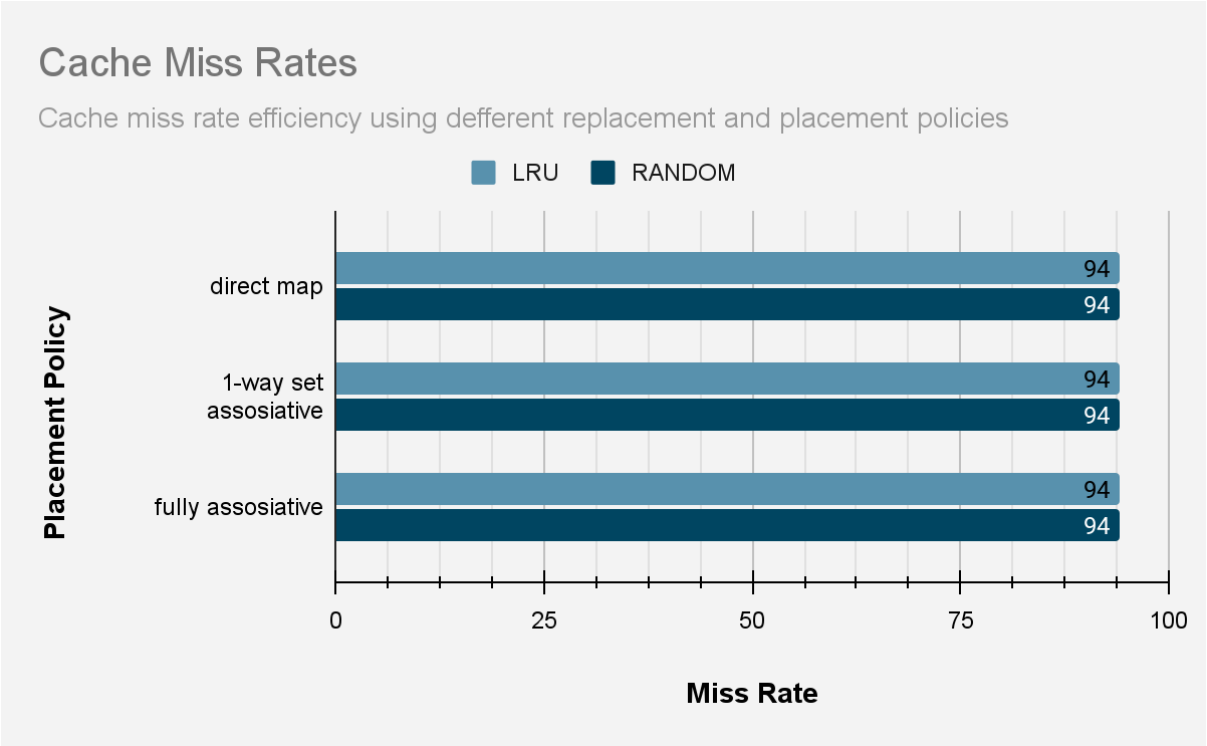
   Άρα καταλήγουμε ότι ο συνολικός χρόνος εκτέλεσης του προγράμματος σε κύκλους ρολογιού είναι 120144 κύκλοι.

# Μέρος 2ο

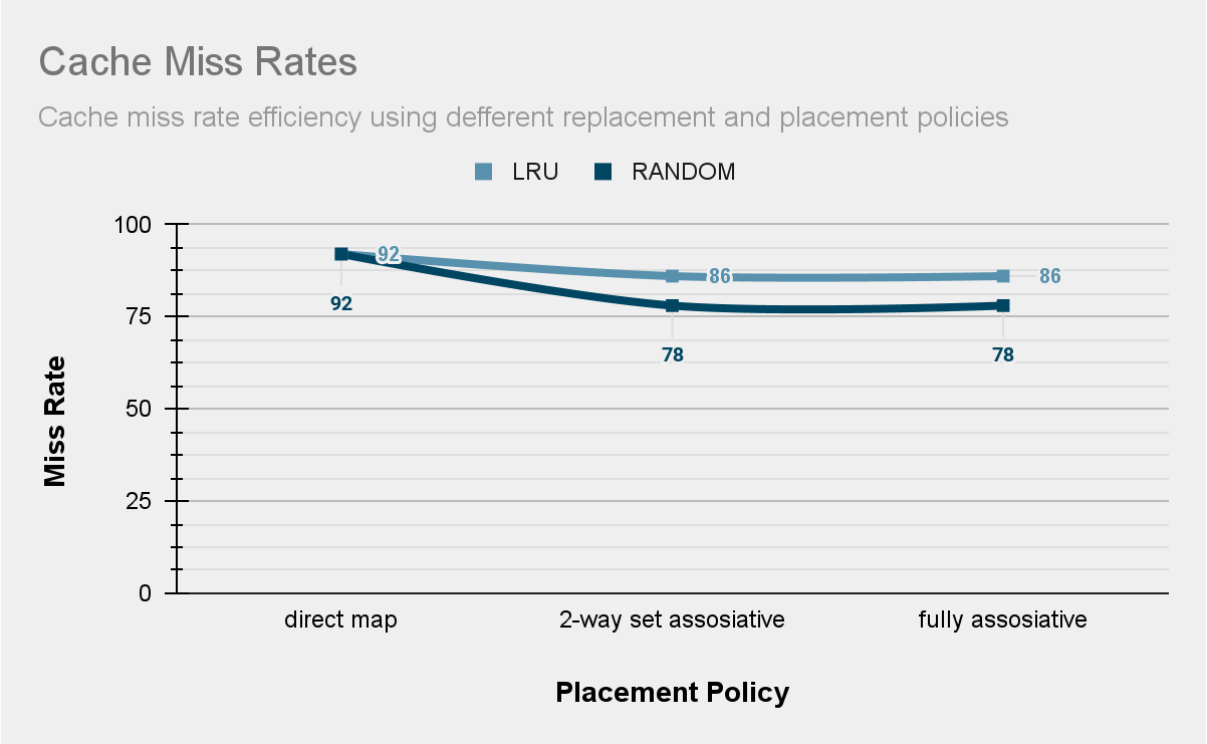
## Σκοπός

Ο σκοπός του 2ου μέρους της εργασίας είναι να εξεταστεί η απόδοση της κρυφής μνήμης με την τροποποίηση των παραμέτρων που μας δίνονται απο το mars.

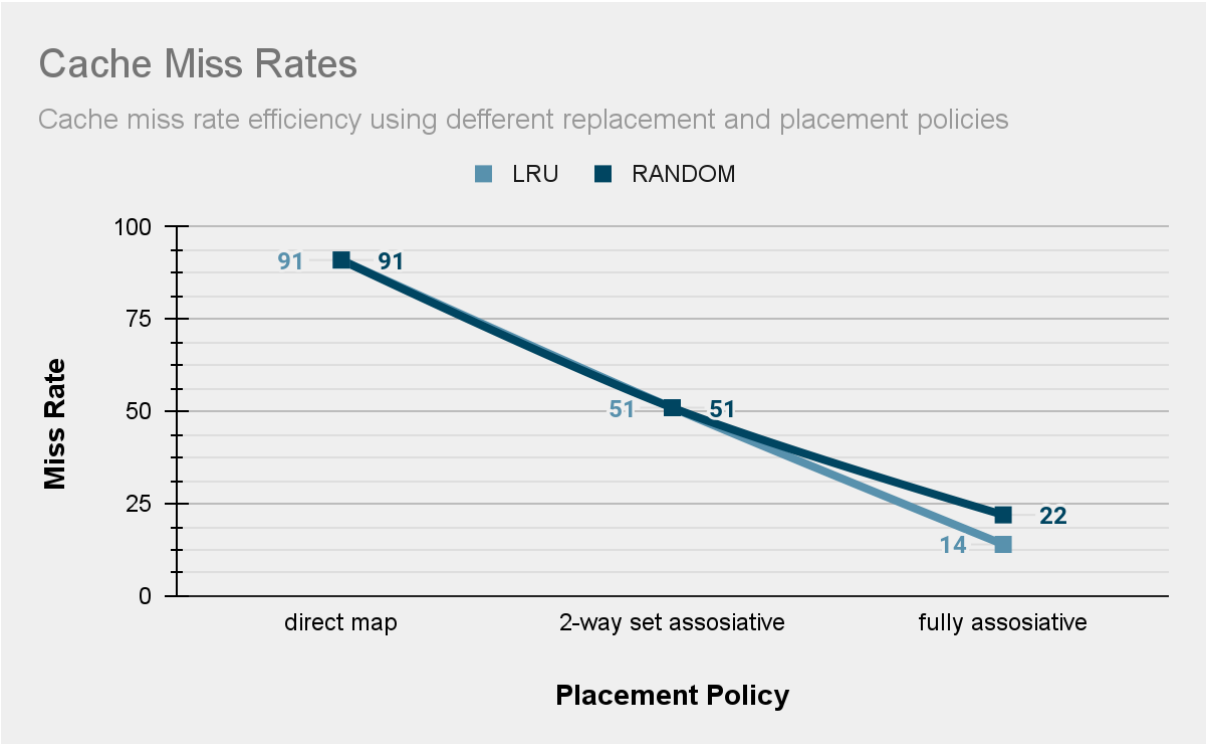
## Διαγράμματα




(Διάγραμμα 1.1) (αριθμός block 1, μέγεθος κρυφής μνήμης 64 byte και 16 λέξεις block)




(Διάγραμμα 1.2) (αριθμός block 2 και μέγεθος κρυφής μνήμης 64 byte και 8 λέξεις block)




(Διάγραμμα 1.3) (αριθμός block 4 και μέγεθος κρυφής μνήμης 64 byte και 4ης λέξεις block)

# Σχόλια

Με βάση το διάγραμμα 1.1 παρατηρούμε ότι το **miss rate** παραμένει ίδιο και στις τρεις διαφορετικές τεχνικές τοποθέτησης μπλοκ. Αυτό γίνεται διότι η κρυφή μνήμη αποτελείται μόνο από ένα μόνο μπλοκ που χωράει 16 λέξεις ( 64 byte ), με αποτέλεσμα το μπλοκ αυτό να αντικαθίσταται πολύ συχνά και έτσι ακόμα και αν στο παρελθόν υπήρξαν τα ζητούμενα δεδομένα έχουν αντικατασταθεί, προκαλώντας έτσι πολλές αστοχίες.

Με βάση το διάγραμμα 1.2 παρατηρούμε ότι το **miss rate** παραμένει ίδιο στις τεχνικές συσχετιζόμενου συνόλου και πλήρους συσχετισμένης τεχνικής και είναι κατά λιγο μεγαλύτερο με την χρήση της τεχνικής άμεσης απεικόνισης. Με την χρήση άμεσης απεικόνισης έχουμε πιο ψηλό **miss rate** διότι σε πολλές περιπτώσης αντικαθίστανται μπλοκς που έχουν πληροφορίες ( λόγο του ότι πολλές διευθύνσεις της μνήμης μπορεί να αντιστοιχούν στο ίδιο μπλοκ ) που χρειάστηκε ο επεξεργαστής,  με αποτέλεσμα να μην τις βρει και να προκαλέσει μια αστοχία (**miss**). Στην περίπτωση της πλήρως συσχετιστικής τεχνικής και της τεχνικής συσχετιζόμενου συνόλου το **miss rate** είναι το ίδιο, διότι όταν η τεχνική συσχετιζόμενου συνόλου χρησιμοποιεί ως σύνολο όλα τα μπλοκς της μνήμης, τότε είναι ισοδύναμο με το να χρησιμοποιούμε την πλήρως συσχετιστική τεχνική.

Με βάση το διάγραμμα 1.3 παρατηρούμε ότι την καλύτερη απόδοση απο τις τρεις τεχνικές την έχει η πλήρως συσχετιζόμενη τεχνική ενώ την χειρότερη απόδοση την έχει η τεχνική της άμεσης απεικόνισης. Οι διαφορές αυτές στην απόδοση προκαλούνται λόγο του ότι οι τεχνικές της πλήρως συσχετιστικής και της τεχνικής συσχετιζόμενου συνόλου διαχειρίζονται πιο αποτελεσματικά τον χώρο της κρυφής μνήμης και έτσι είναι πιο αποδοτικές ( σε περιπτώσεις  που η μνήμη δεν είναι πολύ μεγάλη ).

# Μέρος 3ο

## Σκοπός

Ο σκοπός είναι να τροποποιήσουμε είτε το τμήμα του κώδικα είτε το τμήμα δεδομένων, ωστε να επιτύχουμε καλύτερο **hit rate**.

## Απαντήσεις

Με σκοπό να πετύχουμε το καλύτερο δυνατό ποσοστό ευστοχίας (**hit rate**), αποφασίσαμε να τροποποιήσουμε το τμήμα δεδομένων του προγράμματος. Με βάση την λειτουργία της τεχνικής συσχετιζόμενου συνόλου, ξέρουμε ότι κάθε διεύθυνση της μνήμης αντιστοιχεί σε ένα σύνολο της κρυφής μνήμης. Παρατηρήσαμε ότι γίνονταν πολλές αστοχίες (**miss**) επειδή τα περισσότερα δεδομένα περιορίζονταν στα ιδια σύνολα, με αποτέλεσμα να χάνουμε απο την κρυφή μνήμη δεδομένα τα οποία μας ήταν χρήσιμα. Ο τρόπος που το λύσαμε είναι πειράζοντας το τμήμα των δεδομένων, ώστε να γίνει μια μικρή μετάθεση των διευθύνσεων στην μνήμη και έτσι να αναγκάσουμε την κρυφή μνήμη να βάλει τα δεδομένα σε διαφορετικά σύνολα και έτσι να μην χάνουμε τα χρήσιμα δεδομένα. Πριν από την αλλαγή αυτή το ποσοστό ευστοχίας ήταν ίσο με **59%** ενώ μετά την αλλαγή καταφέραμε να πετύχουμε ποσοστό ευστοχίας ίσο με **77%**. Διάγραμμα απόδοσης: