



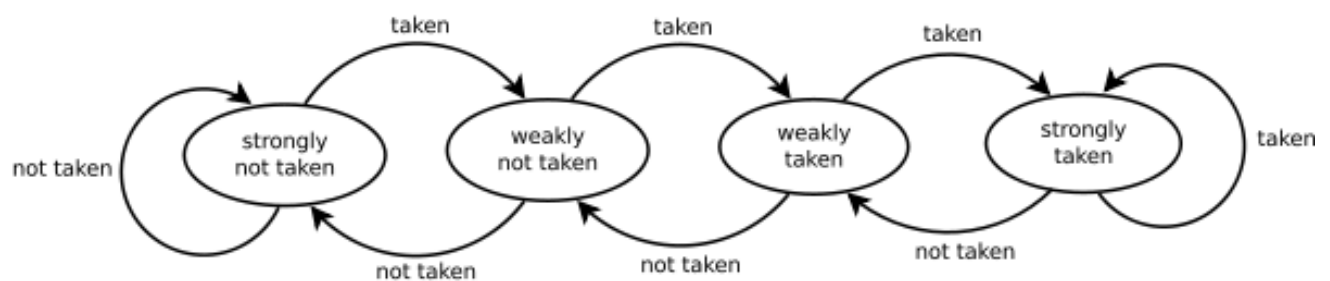
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών

2^η Άσκηση

Μελέτη της επίδρασης συστημάτων πρόβλεψης
εντολών άλματος



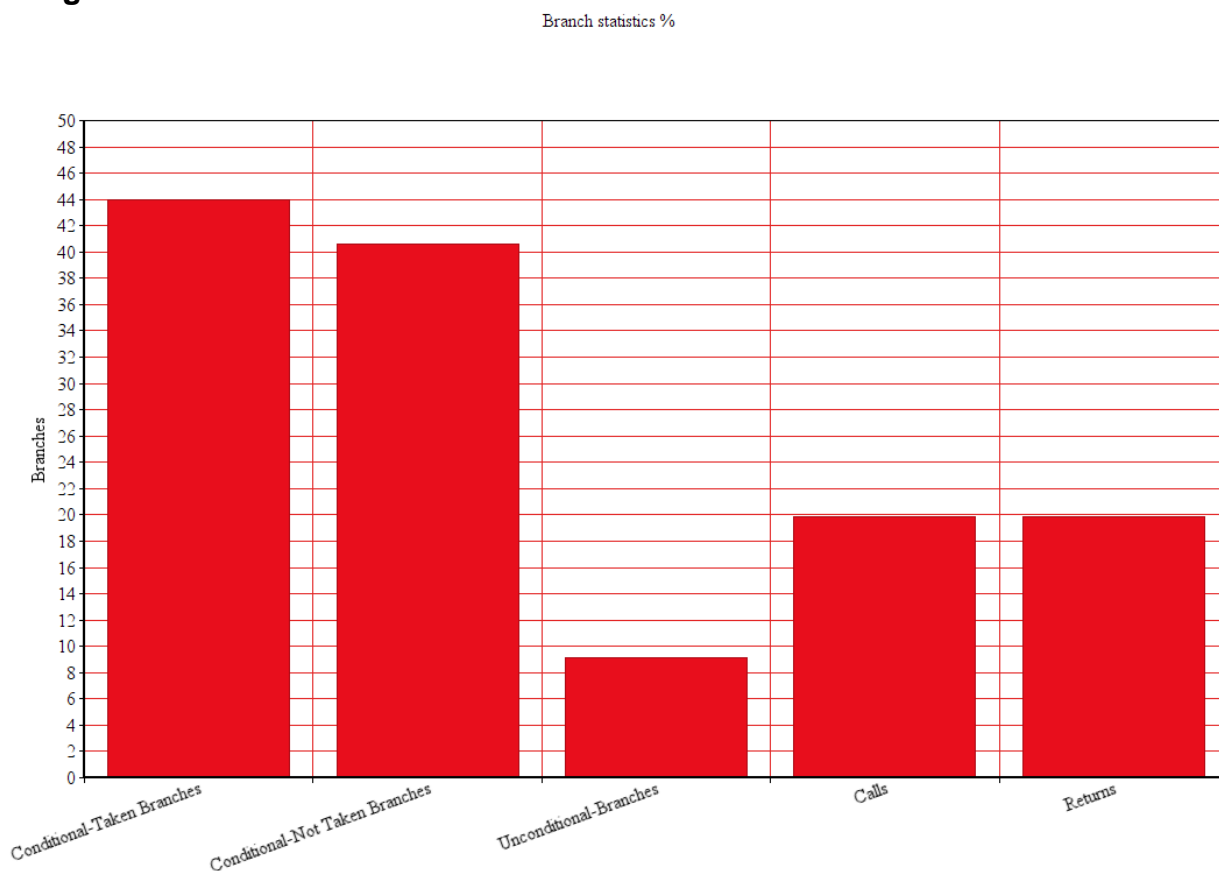
Σκουρτσίδης Γιώργος (03114307)

2019-2020

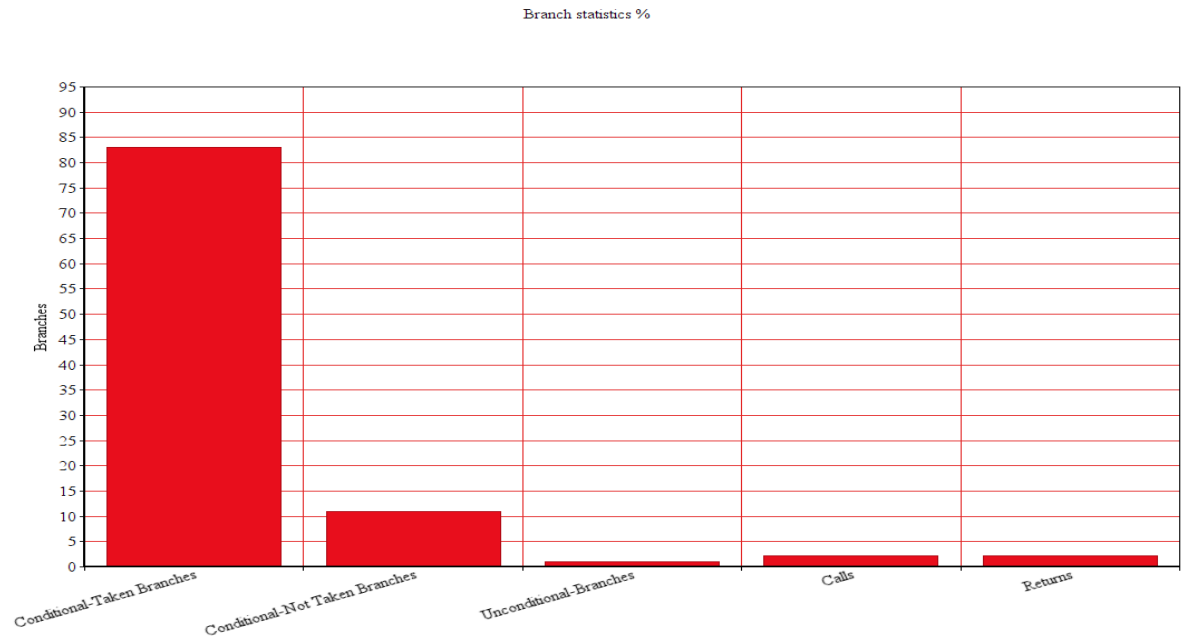
4.1 Μελέτη εντολών άλματος

Παρατήρησα πως σε κάθε μετροπρόγραμμα μπορεί να είναι περισσότερα τα conditional taken branches ή τα conditional not-taken branches. Τα Unconditional-Branches και τα Calls και Returns είναι πολύ λιγότερα σε συνολικό αριθμό. Προφανώς έχουμε ίδιο αριθμό Calls και Returns σε κάθε πρόγραμμα. Παρακάτω ακολουθούν τα διαγράμματα.

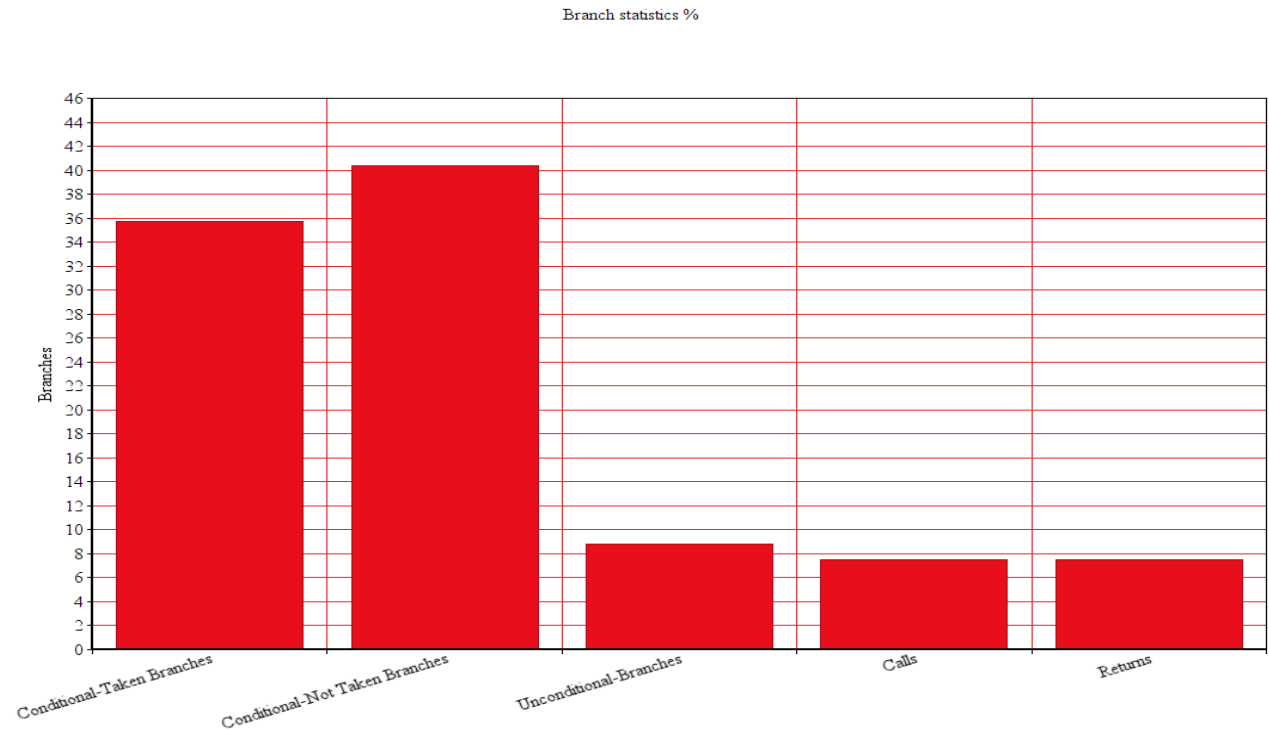
1)403.gcc



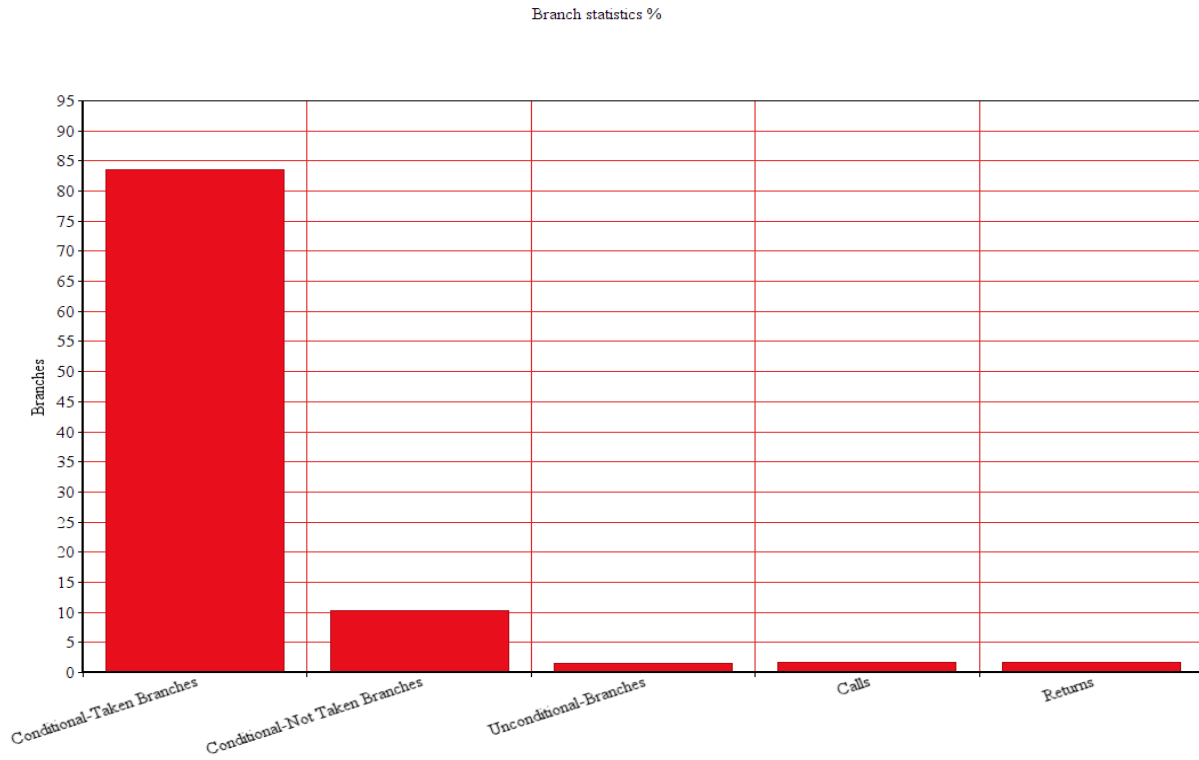
2)429.mcf



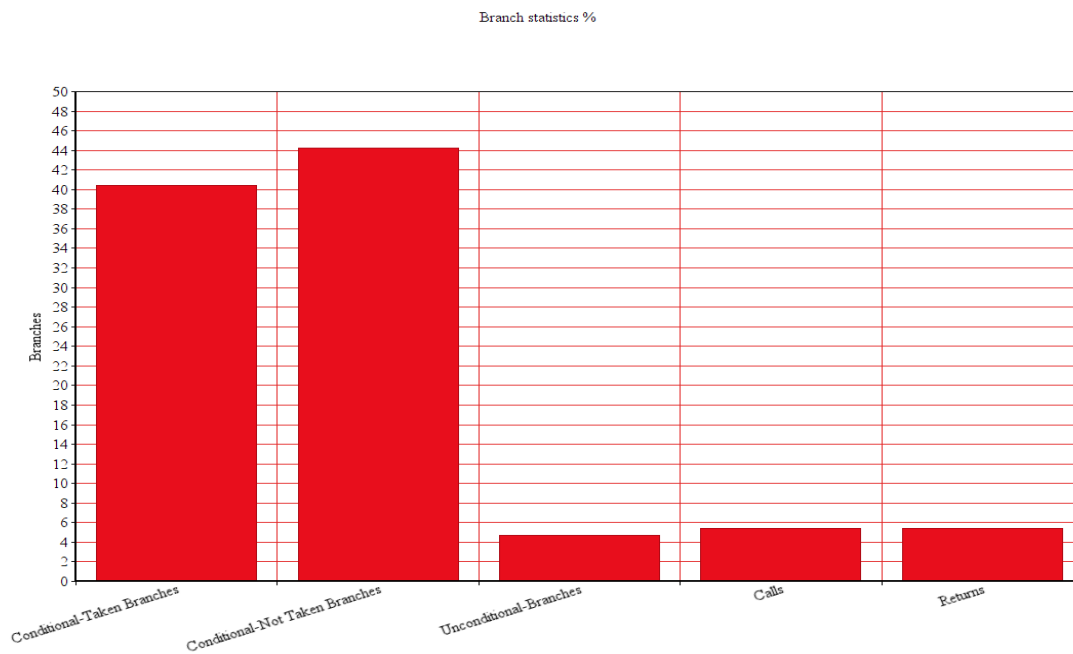
3)434.zeusmp



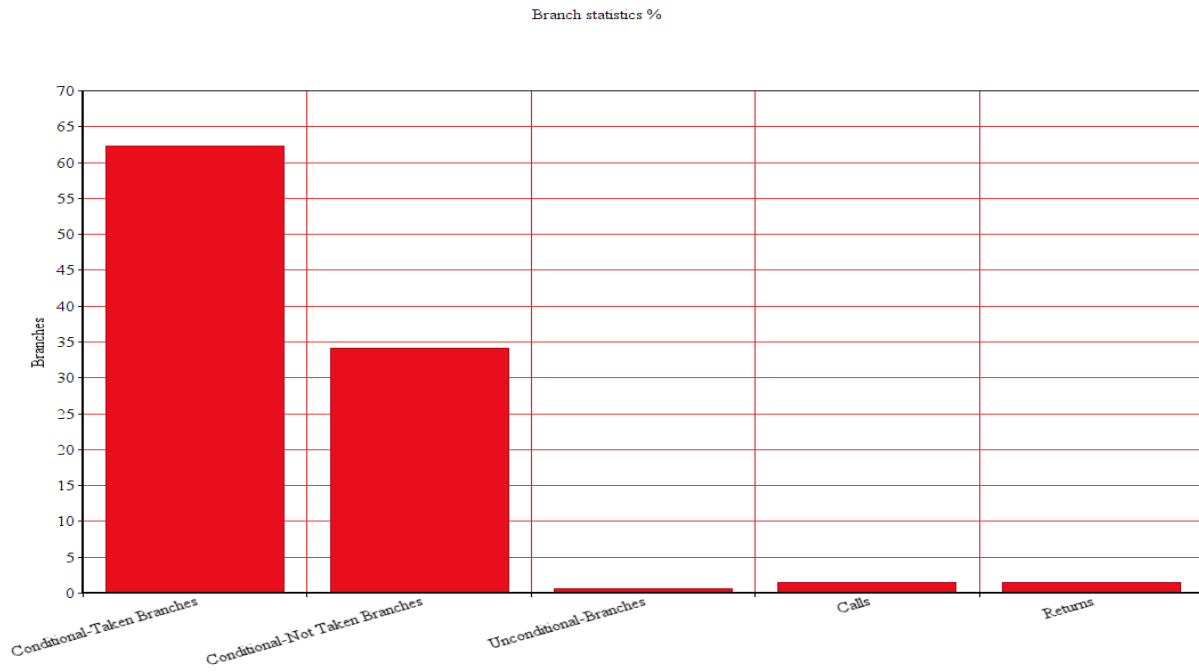
4) 436.cactusADM



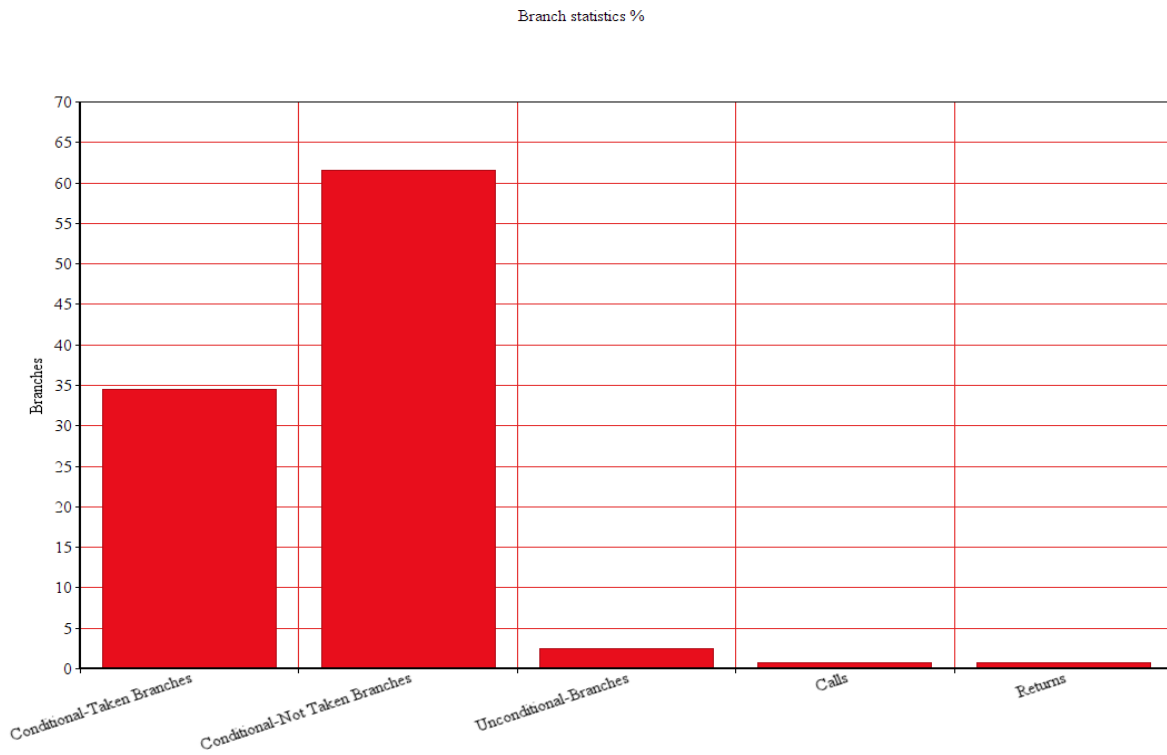
5) 445.gobmk



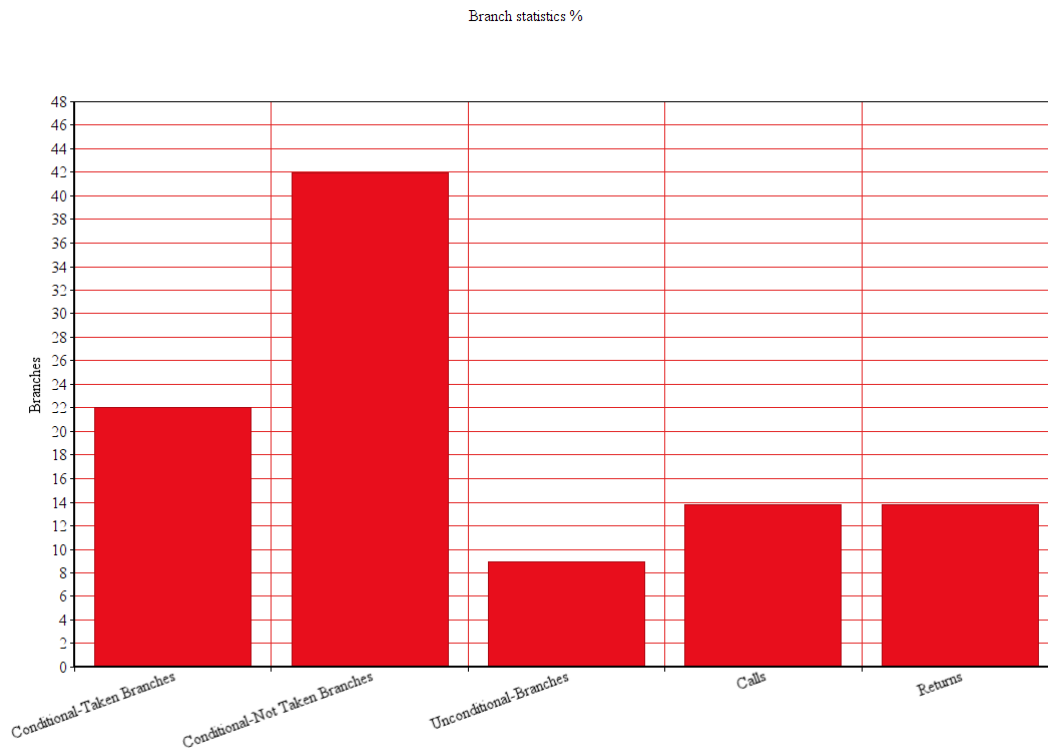
6) 450.soplex



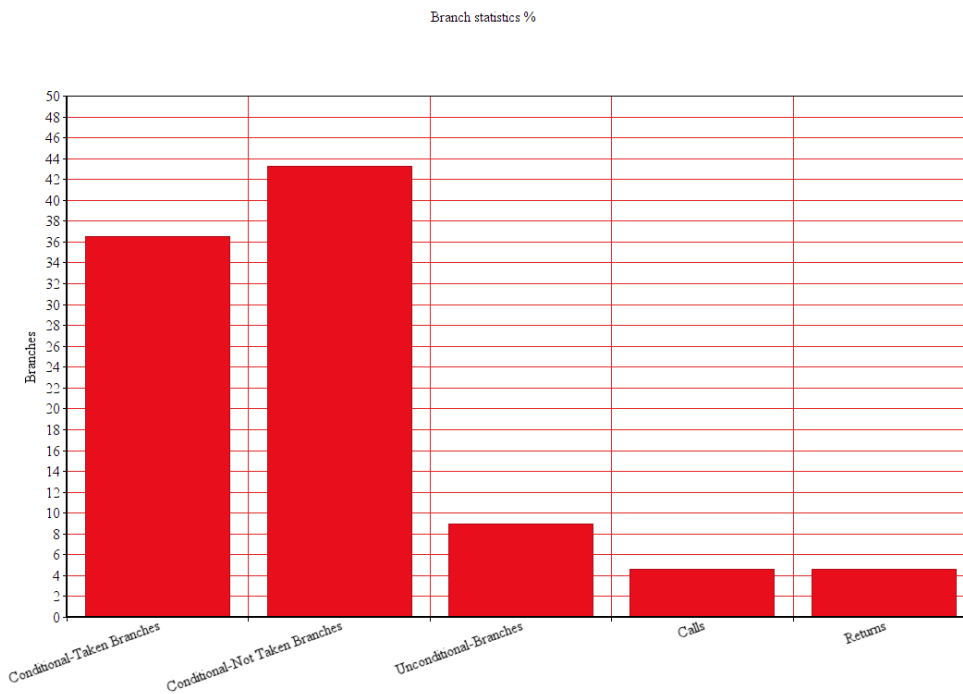
7) 456.hmmmer



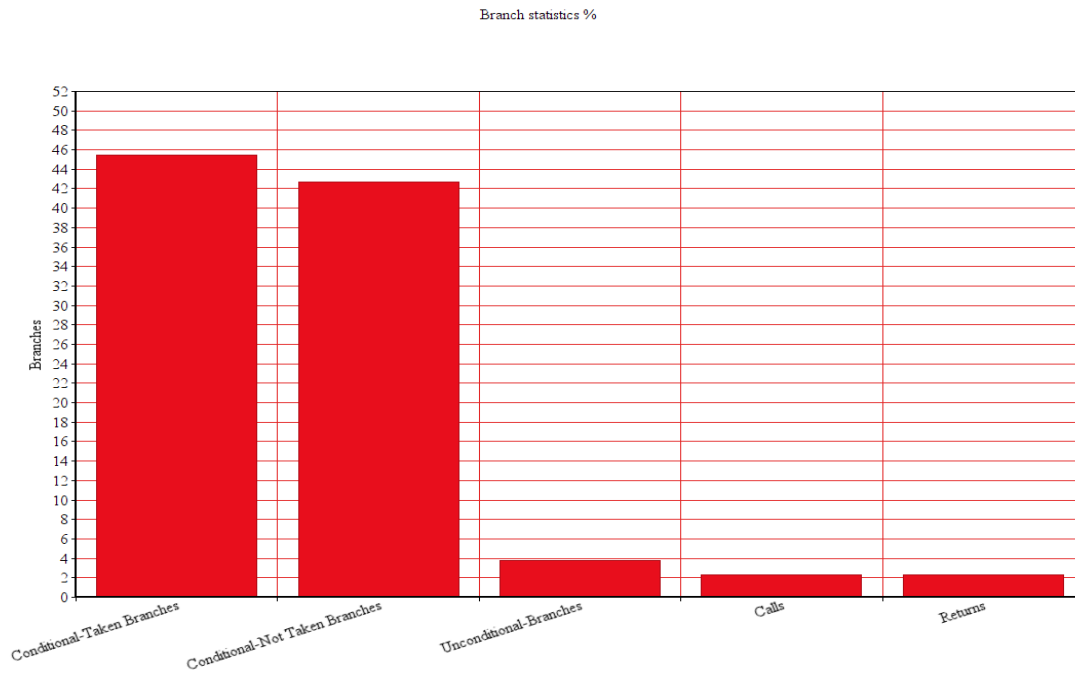
8) 458.sjeng



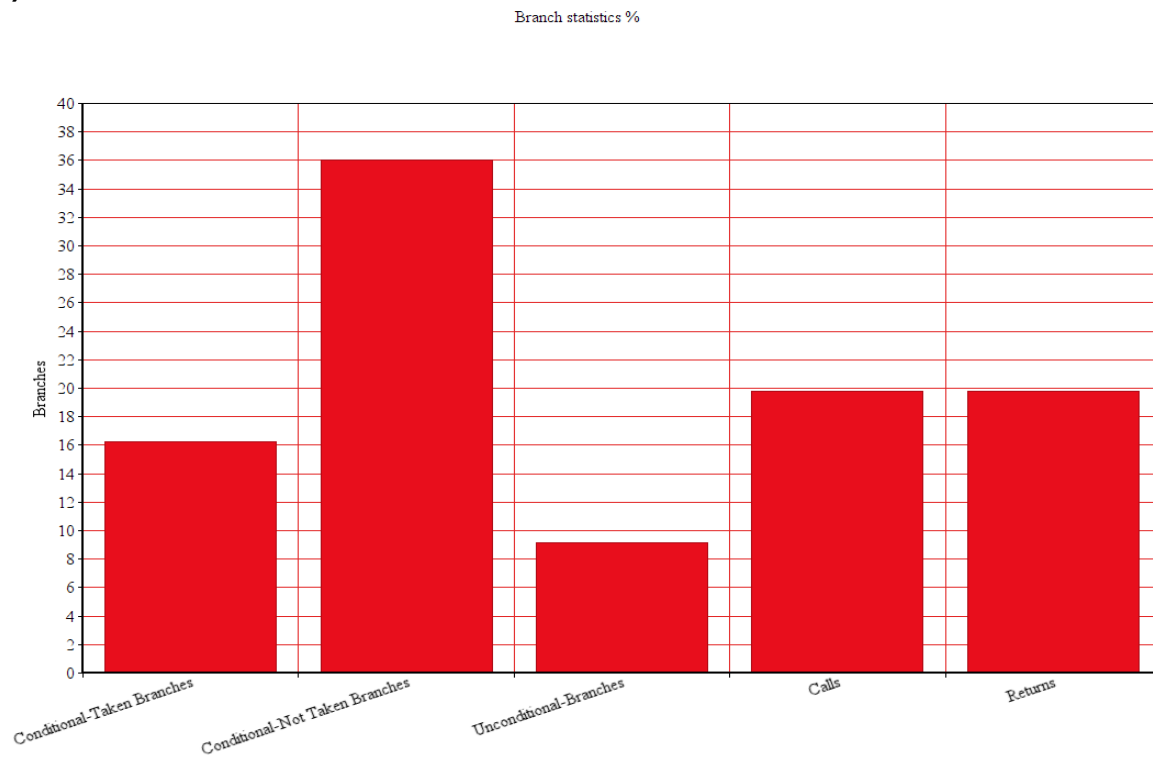
9) 459.GemsFDTD



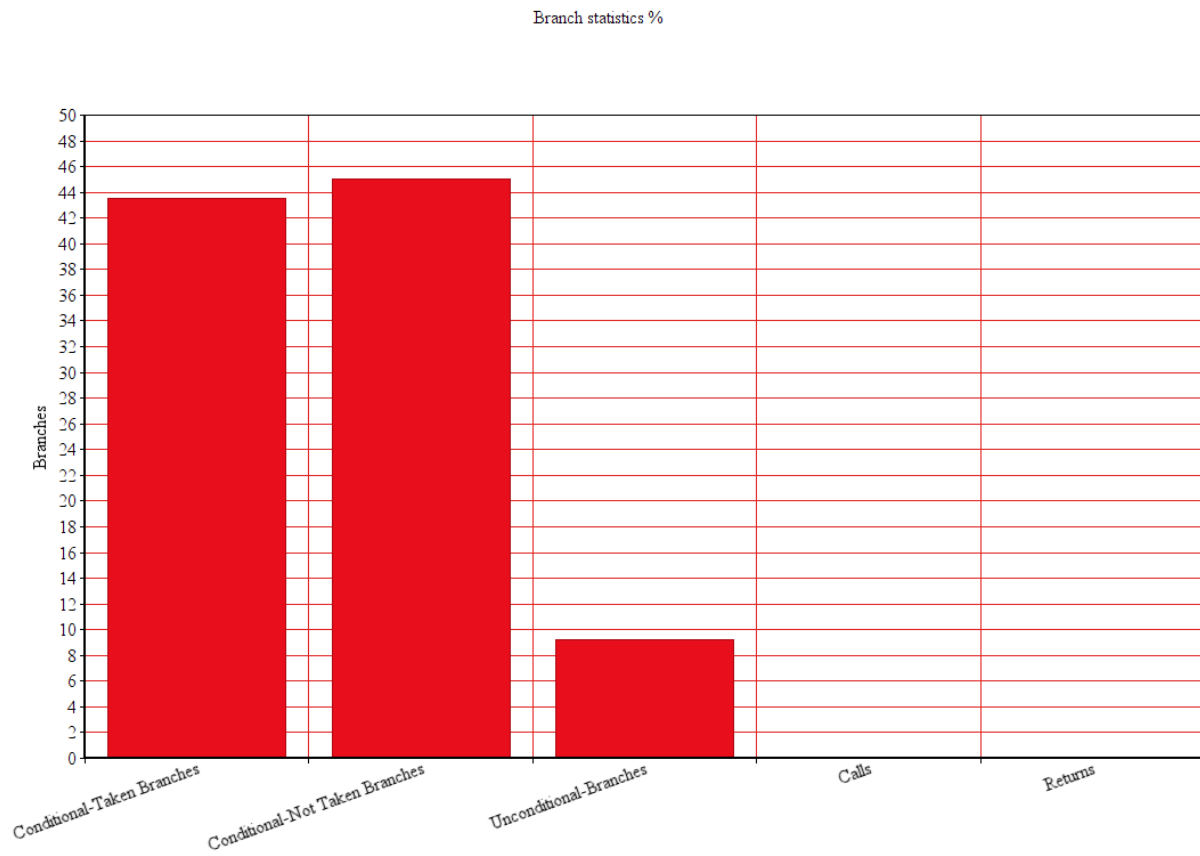
10) 471.omnetpp



11) 473.astar



12) 483.xalancbmk



4.2 Μελέτη των N-bit predictors

Θα παρουσιαστεί το ερώτημα 4.2.α και το ερώτημα 4.2.β σε κοινό γράφημα για κάθε μετροπρόγραμμα. Δηλαδή θα παραστήσουμε σε κοινό γράφημα τους n-bit predictors για hardware 16K και 32K.

i)

Παρατηρούμε πως η αύξηση του αριθμού των bit συνεπάγεται αύξηση της απόδοσης των predictors, καθώς το ποσοστό αστοχίας μειώνεται αποτελεσματικά. Γενικά σε όλα τα μετροπρόγραμμα η αύξηση από 1-bit predictor σε 2-bit predictor έχει εντυπωσιακά θετικά αποτελέσματα, αφού συνεπάγεται μείωση της αστοχίας έως και 7% σε ορισμένα μετροπρόγραμμα.

Στη συνέχεια η αύξηση από τα 2bit σε 3bit βοηθά επιπλέον στην μείωση της αστοχίας με ορισμένες εξαιρέσεις (zeusmp, hmmer) όπου η αστοχία αυξάνεται ή παραμένει σταθερή.

Η αύξηση από τα 3-bit στα 4-bit στα περισσότερα benchmarks δεν έχει ιδιαίτερα σημαντικά οφέλη καθώς ο ρυθμός αστοχίας παραμένει σταθερός ή ακόμα και σε ορισμένες περιπτώσεις αυξάνεται. Μόνο σε ένα μετροπρόγραμμα παρατηρήθηκε ικανοποιητική μείωση του ρυθμού αστοχίας με την αύξηση των bit από τα 3 στα 4.

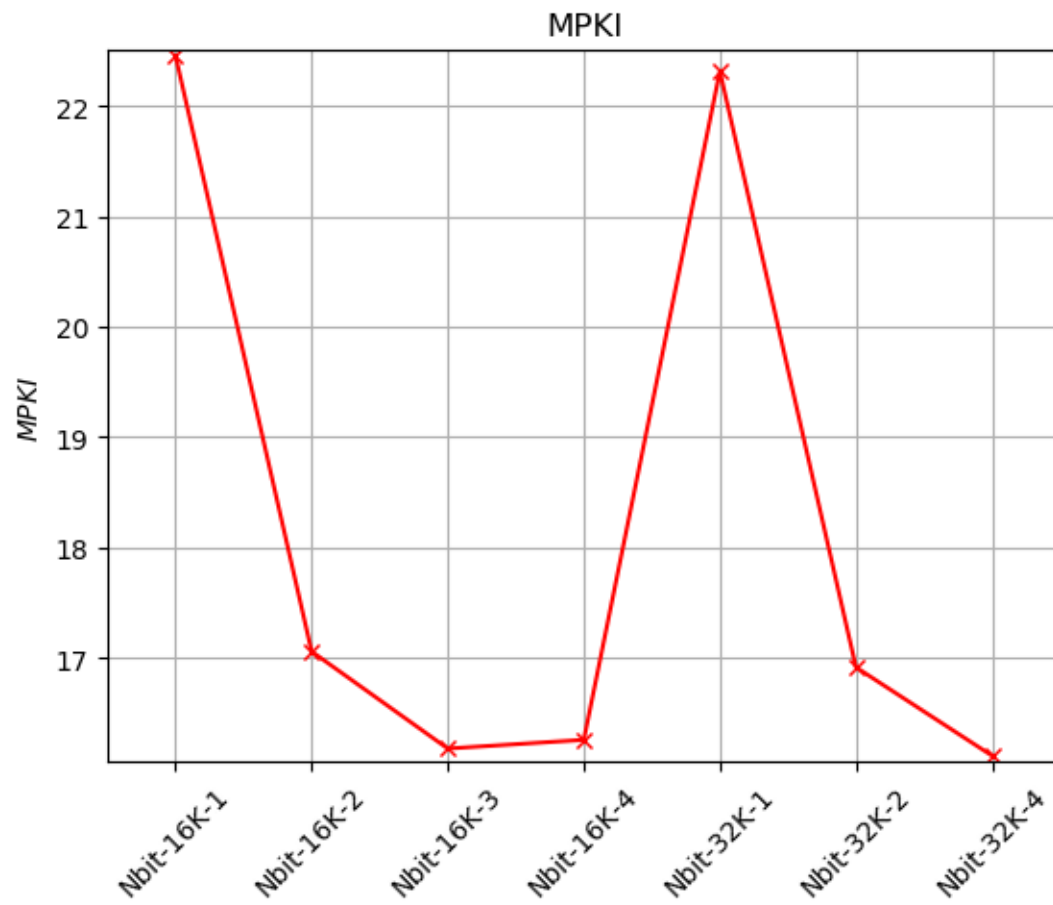
ii)

Η αύξηση του hardware στα 32K δεν μας δίνει ικανοποιητικά αποτελέσματα ώστε να δικαιολογείται το αυξημένο κόστος κατασκευής, καθώς ο ρυθμός αστοχίας κάθε n-bit predictor για κάθε μετροπρόγραμμα, είναι συγκρίσιμος ή πανομοιότυπος με τον αντίστοιχο ρυθμό αστοχίας που μας δίνει ο n-bit predictor με το hardware στα 16K. Παρατηρείται δηλαδή μια μικρή μείωση του ρυθμού αστοχίας, κρίνεται όμως ήσσονος σημασίας.

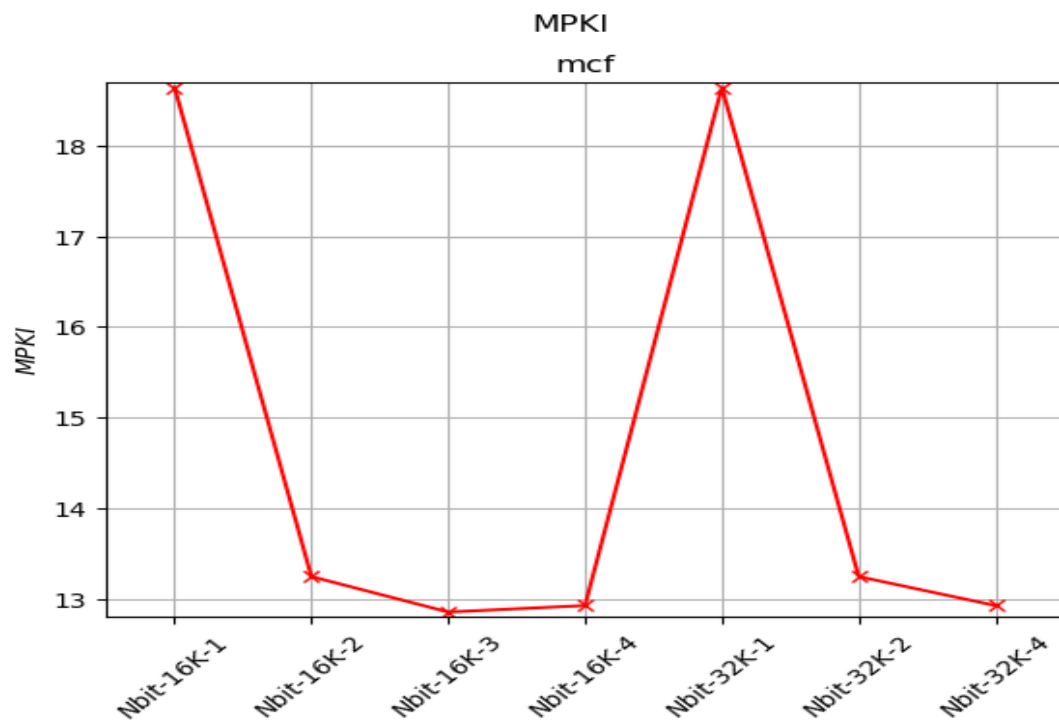
Καλύτερος n-bit predictor επιλέγεται ο 3-bit με το hardware ίσο με 16K.

Στη συνέχεια ακολουθούν τα διαγράμματα για κάθε benchmark.

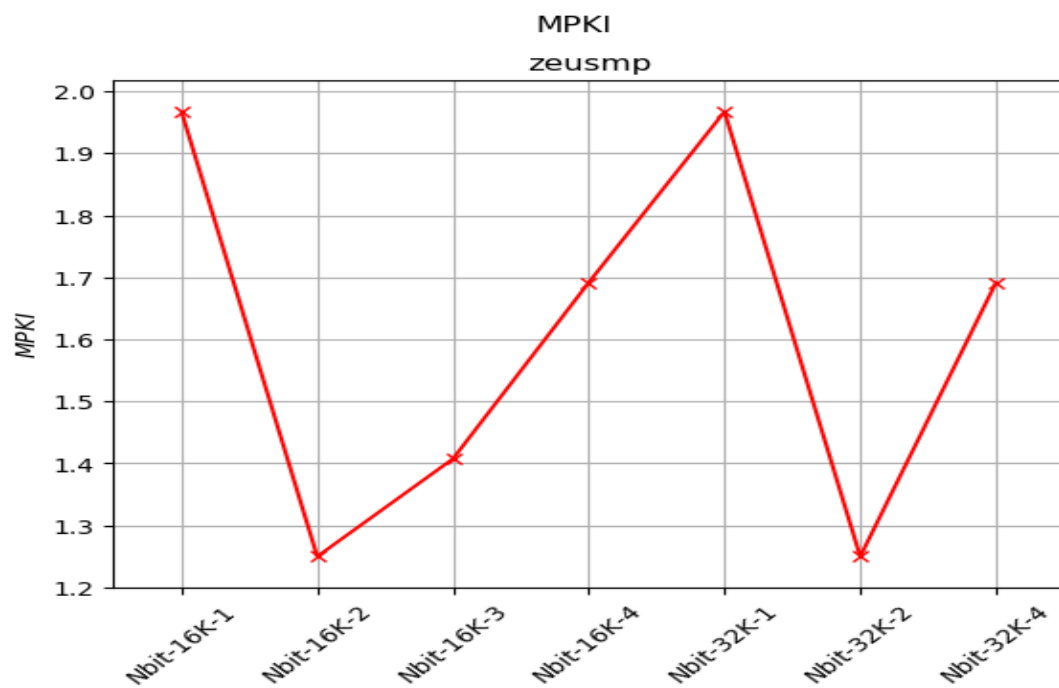
1) 403.gcc

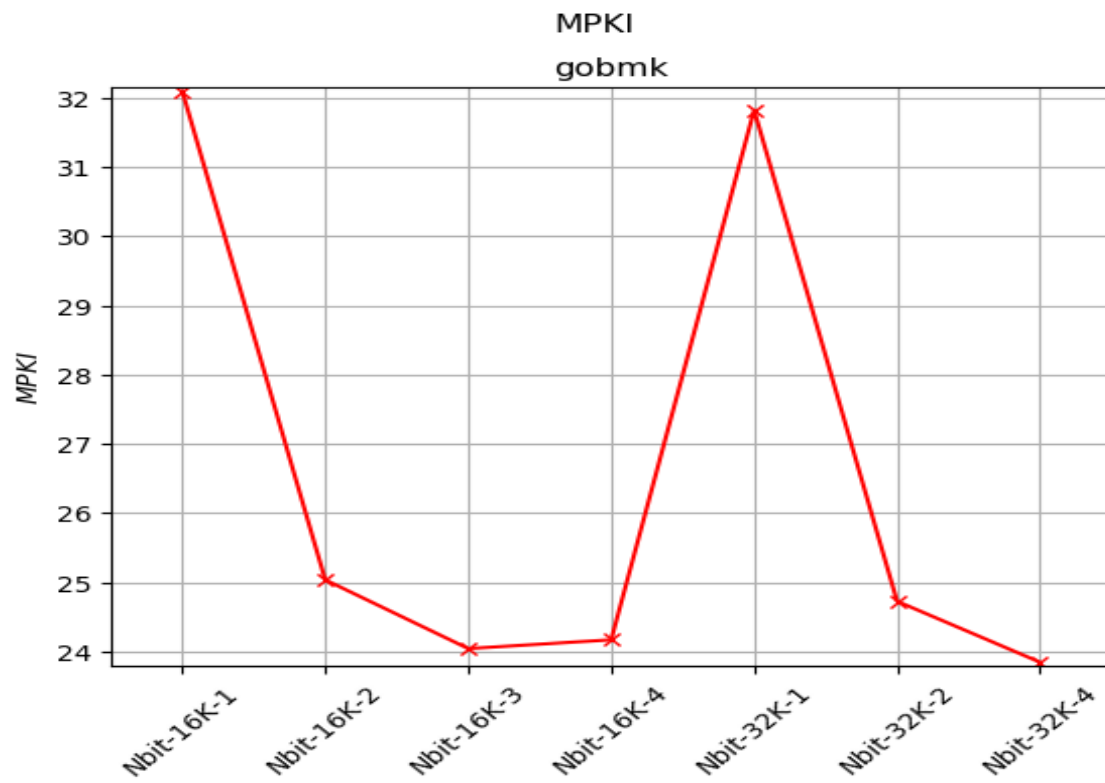
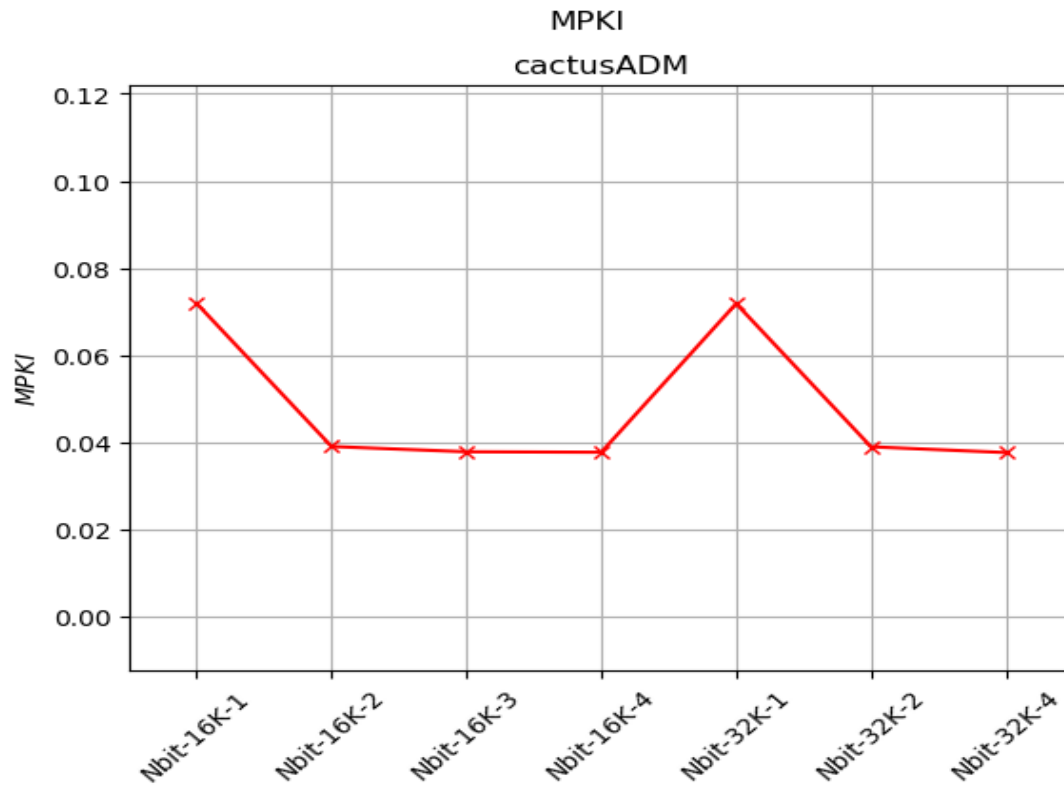


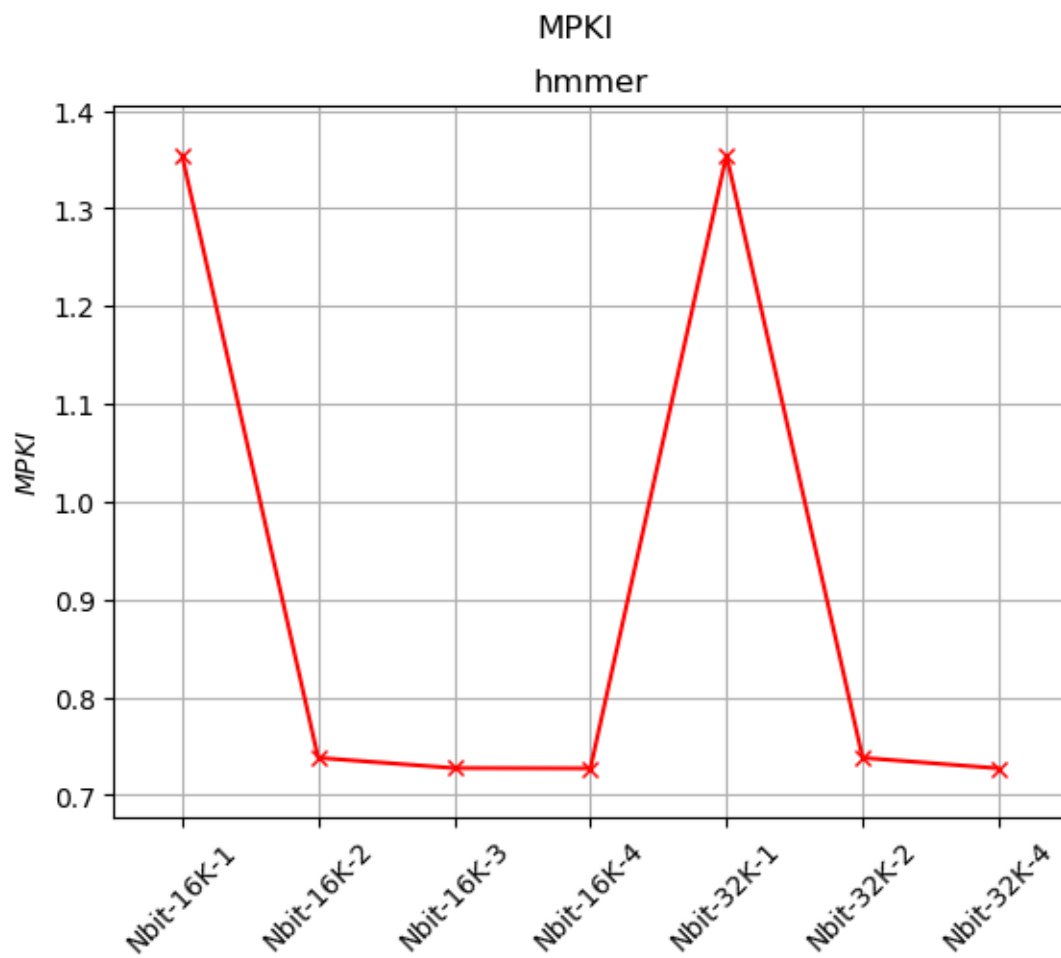
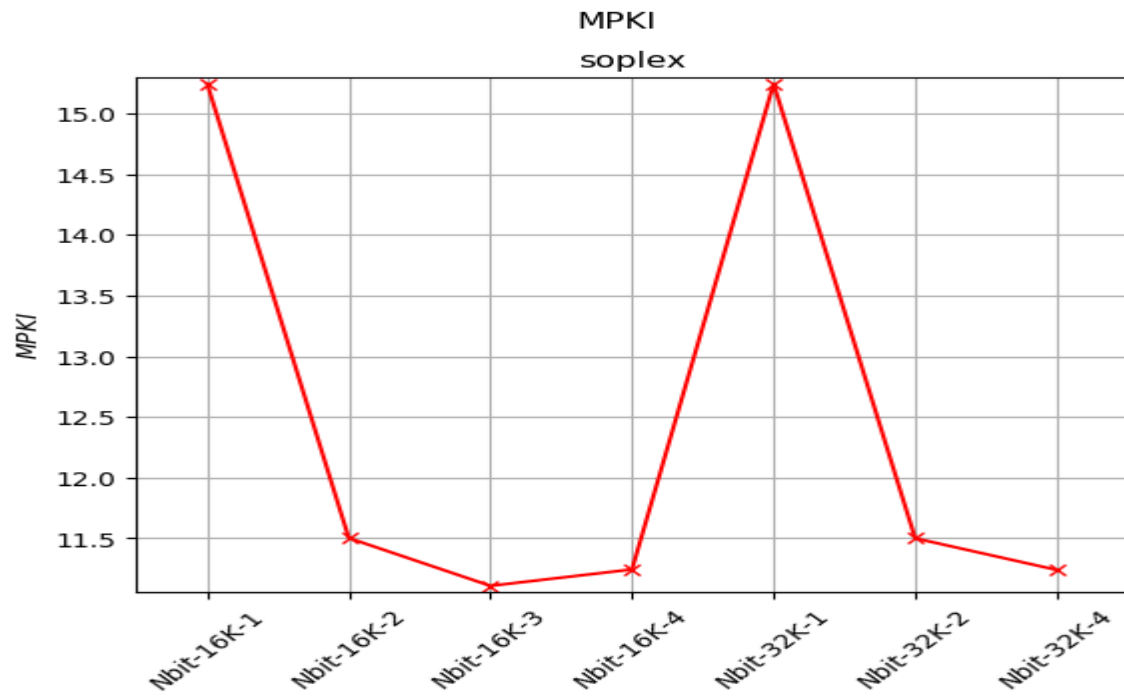
2) 429.mcf

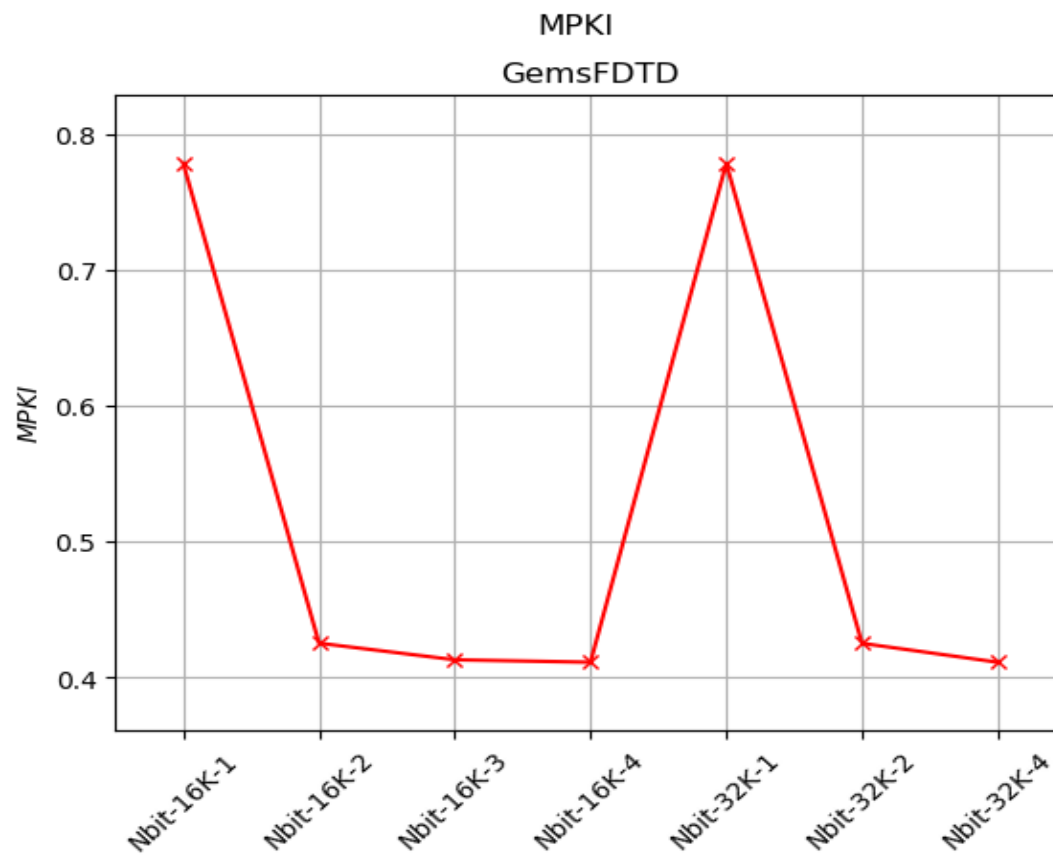
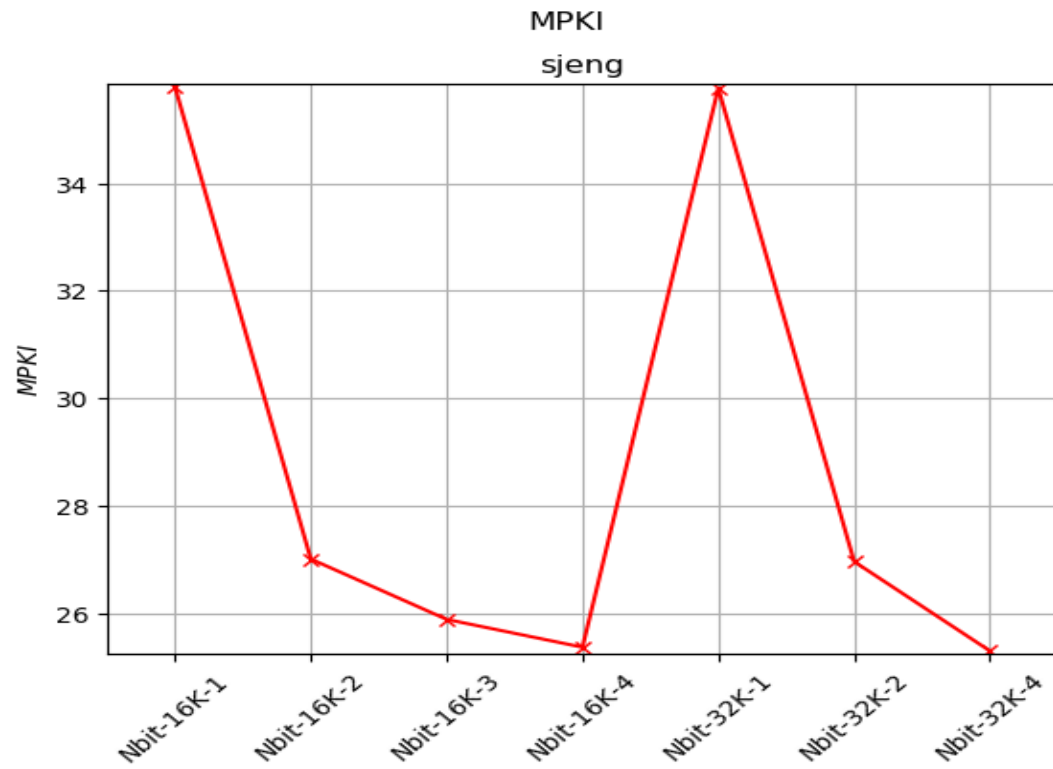


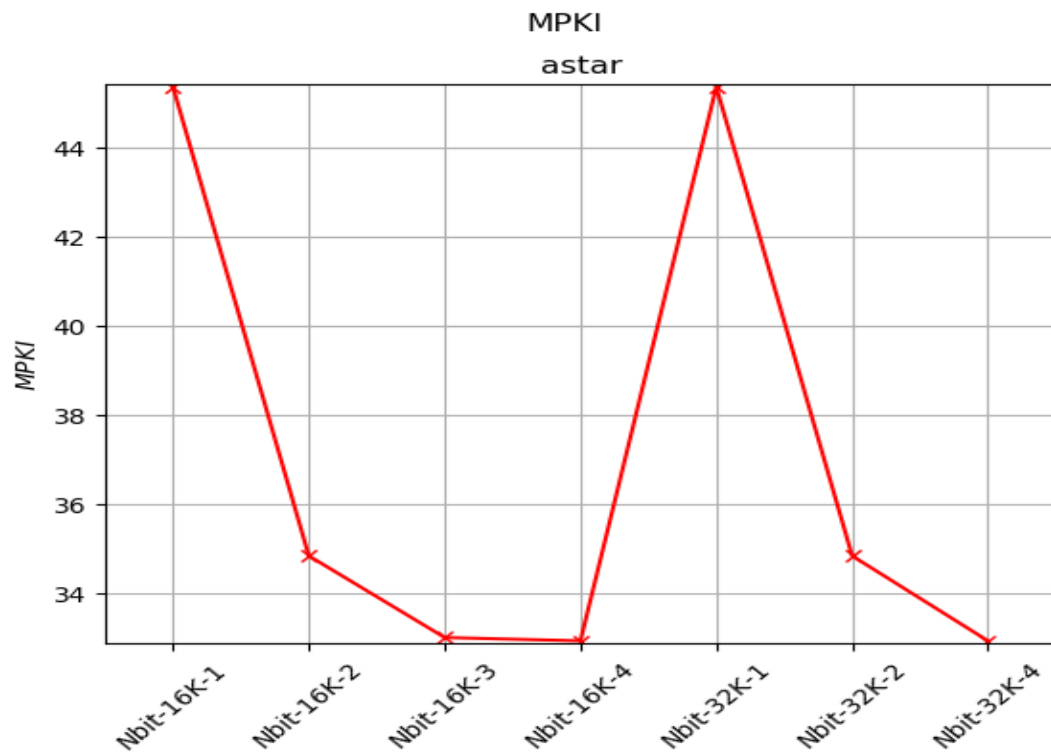
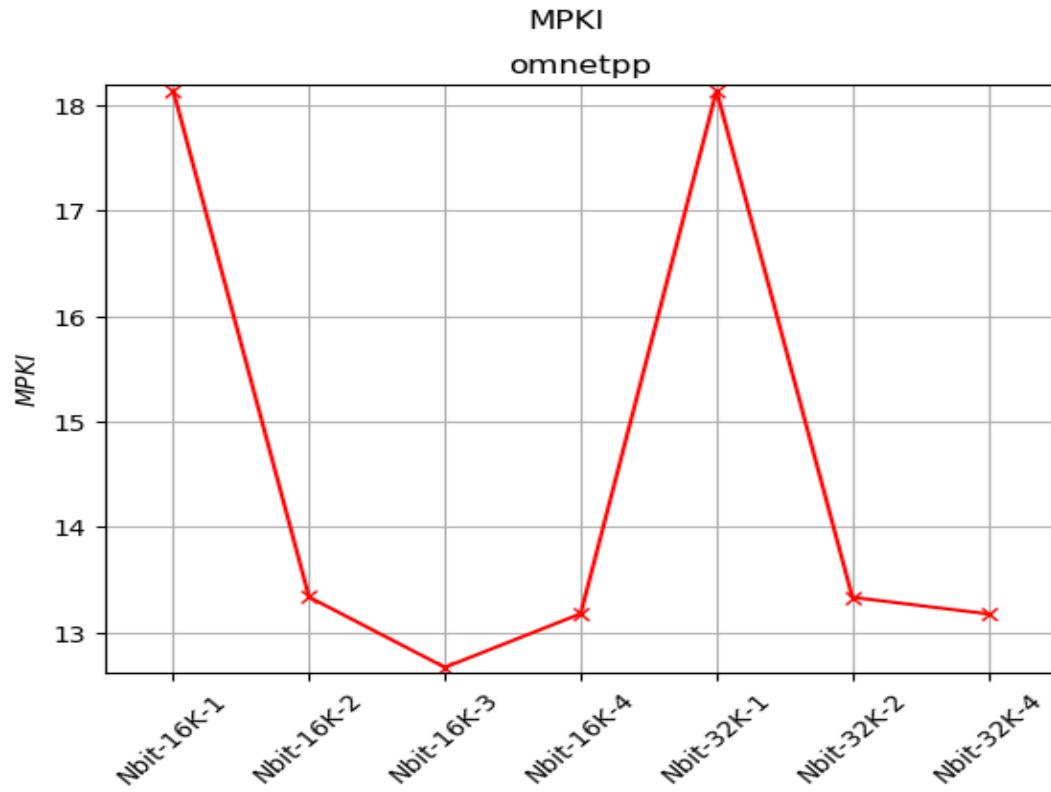
3) 434.zeusmp

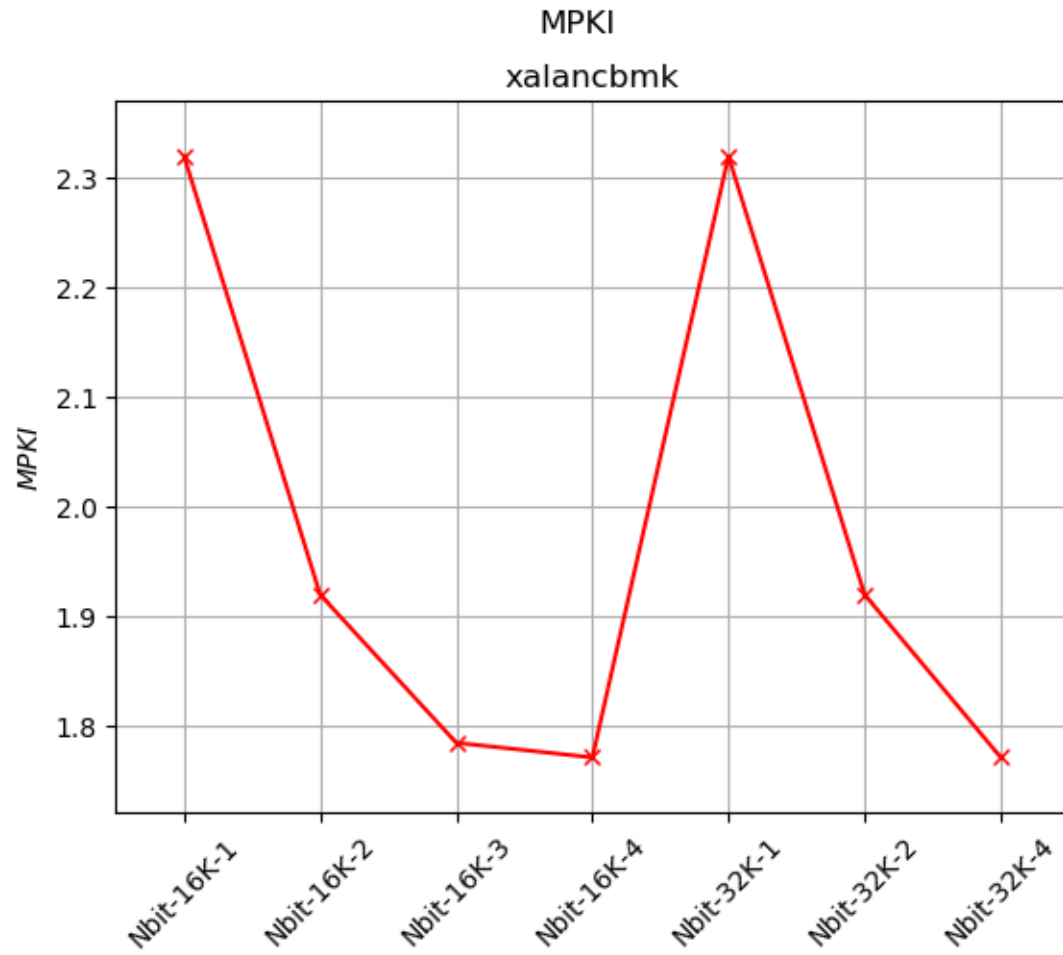












4.3 Μελέτη του BTB

Υλοποίησα έναν BTB και να μελέτησα την ακρίβεια πρόβλεψής του μέσω των τιμών του MPKI και του target MPKI για τις ακόλουθες περιπτώσεις:

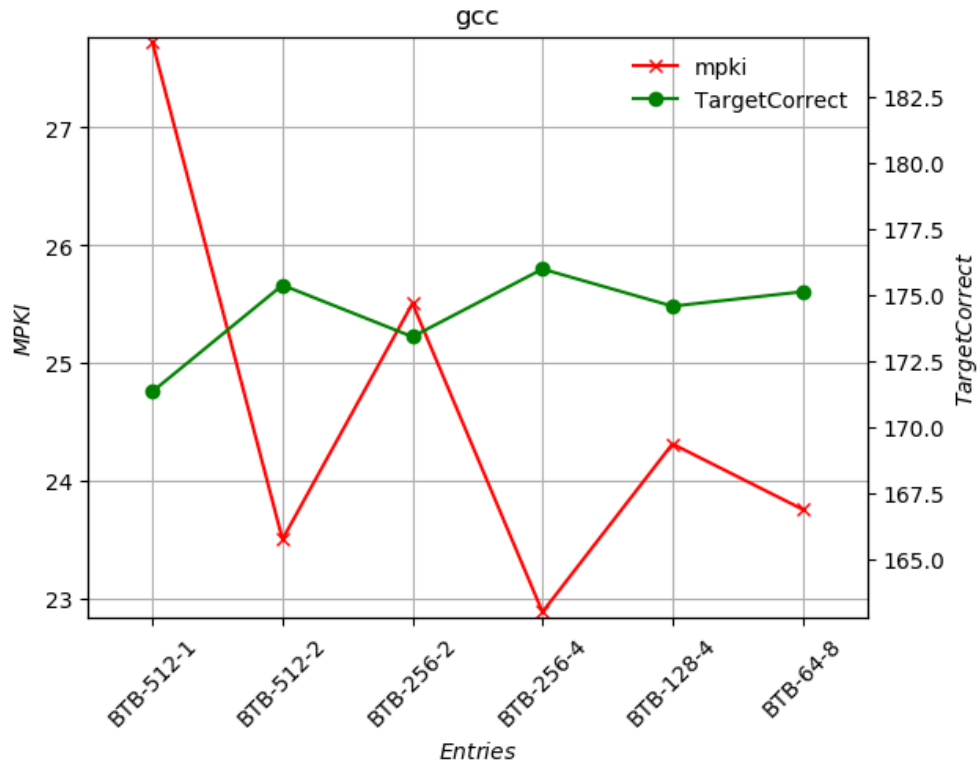
btb entries	btb associativity
512	1, 2
256	2, 4
128	4
64	8

Σε γενικές γραμμές, ο ρυθμός αστοχίας για κάθε benchmark φάνηκε να επηρεάζεται εντόνως από το associativity και τα entries του BTB. Υπήρχαν ελάχιστες εξαιρέσεις όπου οι αλλαγές στα χαρακτηριστικά του BTB δεν συνεπάγονταν αλλαγές στις μετρικές που ακολουθήσαμε.

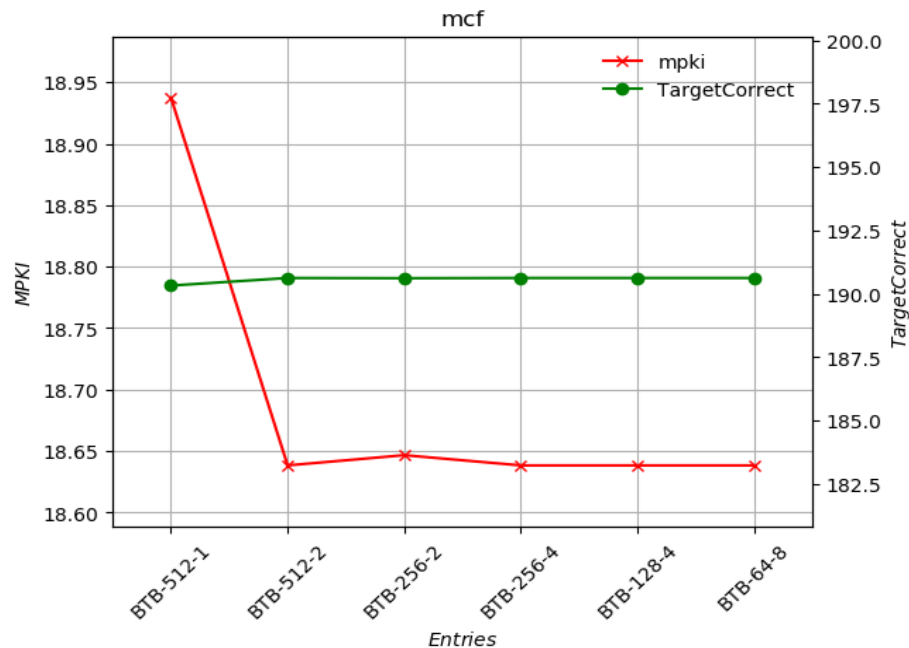
Βέλτιστος συνδυασμός θεωρήθηκε ο (btb entries, associativity) = (256, 4) ο οποίος στην πλειοψηφία των περιπτώσεων μας δίνει τα βέλτιστα αποτελέσματα.

Παρακάτω ακολουθούν τα διαγράμματα για κάθε μετροπρόγραμμα.

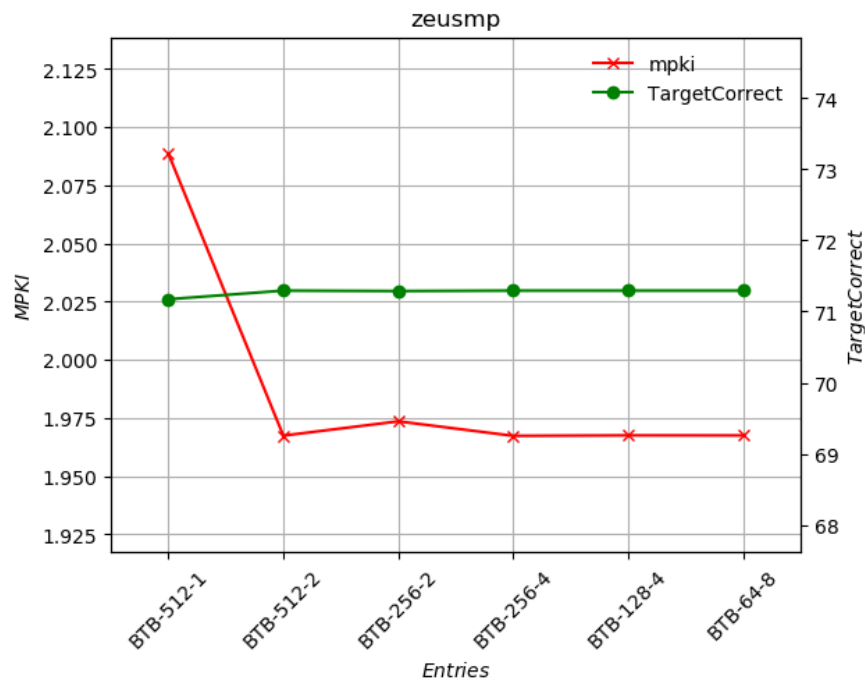
1) Gcc



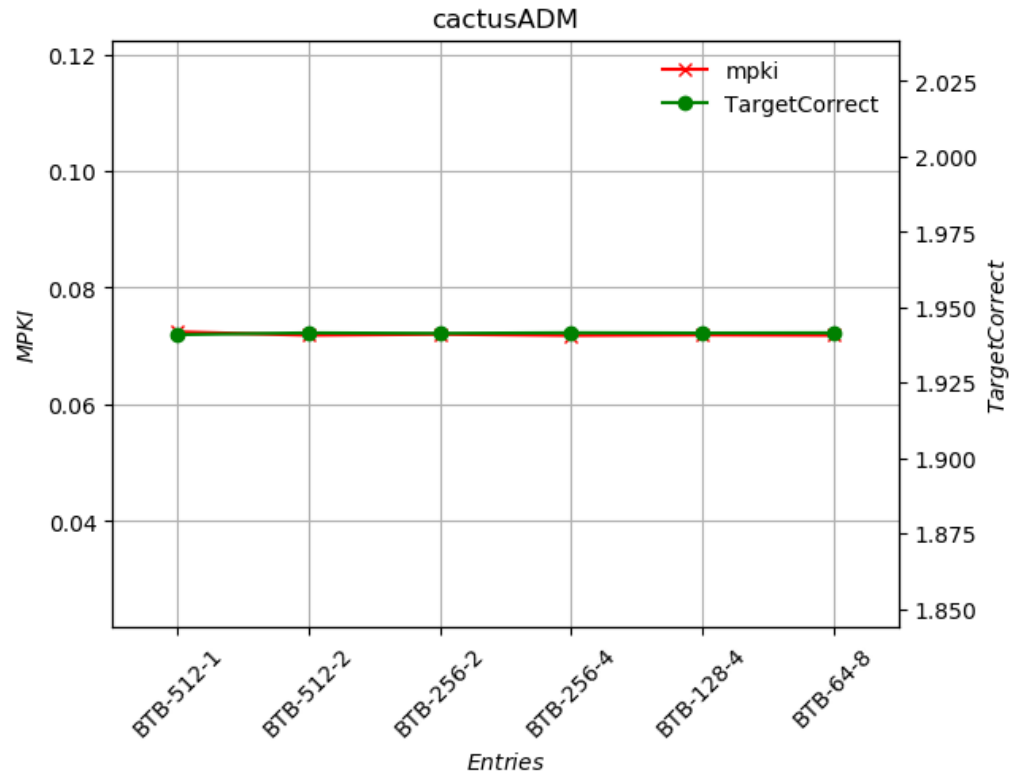
Παρατηρούμε πως καθώς αυξάνεται το associativity το mpki μειώνεται, ενώ καθώς αυξάνονται τα entries με σταθερό associativity το mpki αυξάνεται. Πιθανόν υπάρχουν collisions και το μεγαλύτερο associativity βοηθά στην μείωση αυτών. Βέλτιστος συνδυασμός ο (256,4).



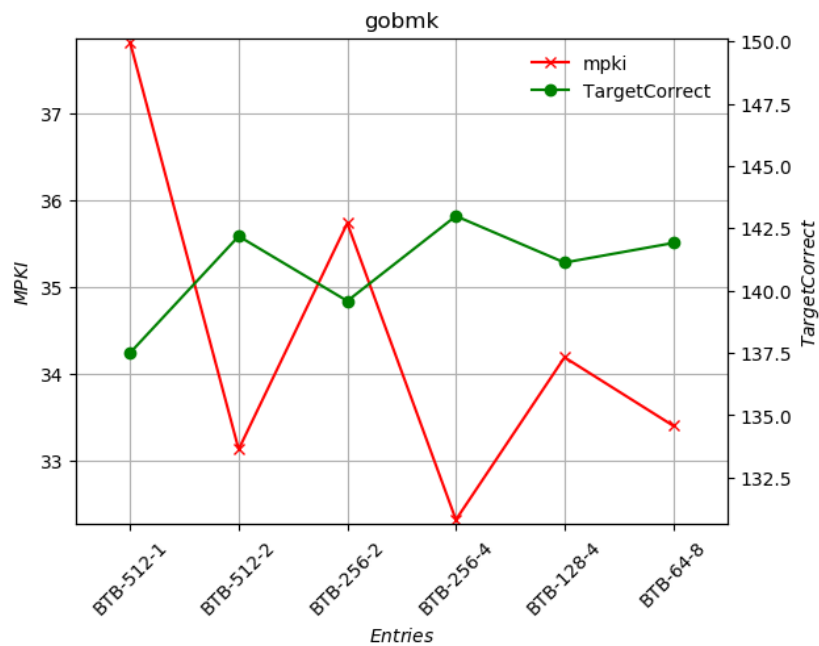
Το Target Correct παραμένει ανεπηρέαστο από τις αλλαγές στο BTB. Παρατηρούμε ότι με τουλάχιστον associativity ίσο με δύο επιτυγχάνονται σχεδόν βέλτιστα αποτελέσματα. Πιθανώς υπάρχουν collisions που για associativity >1 εξαλείφονται



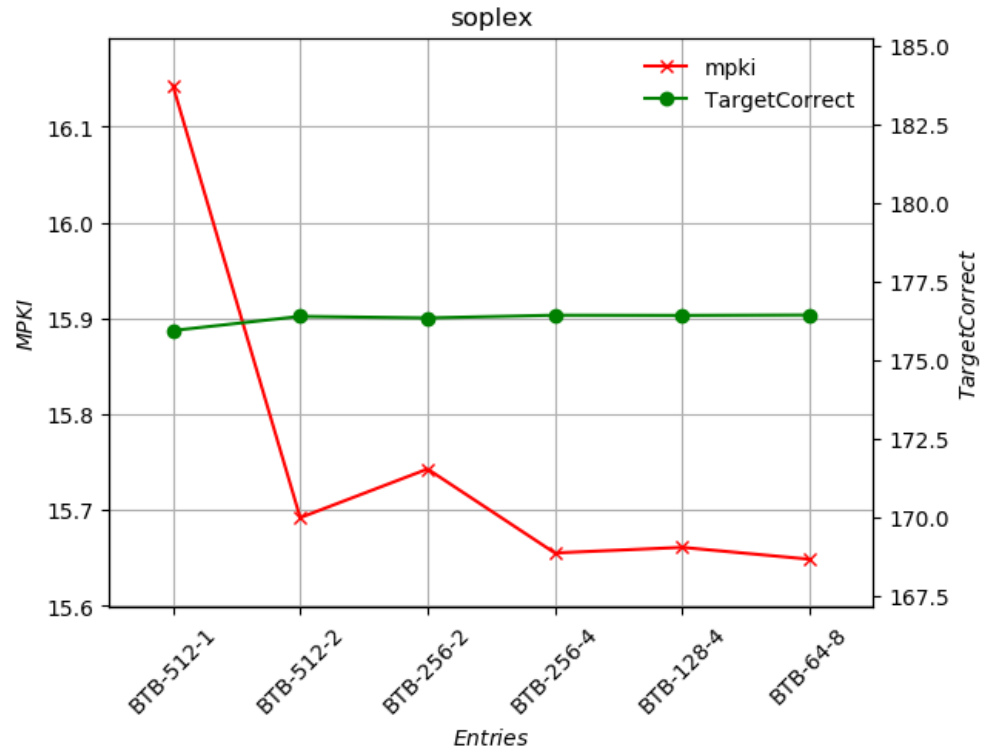
Ισχύει ότι και για το mcf.



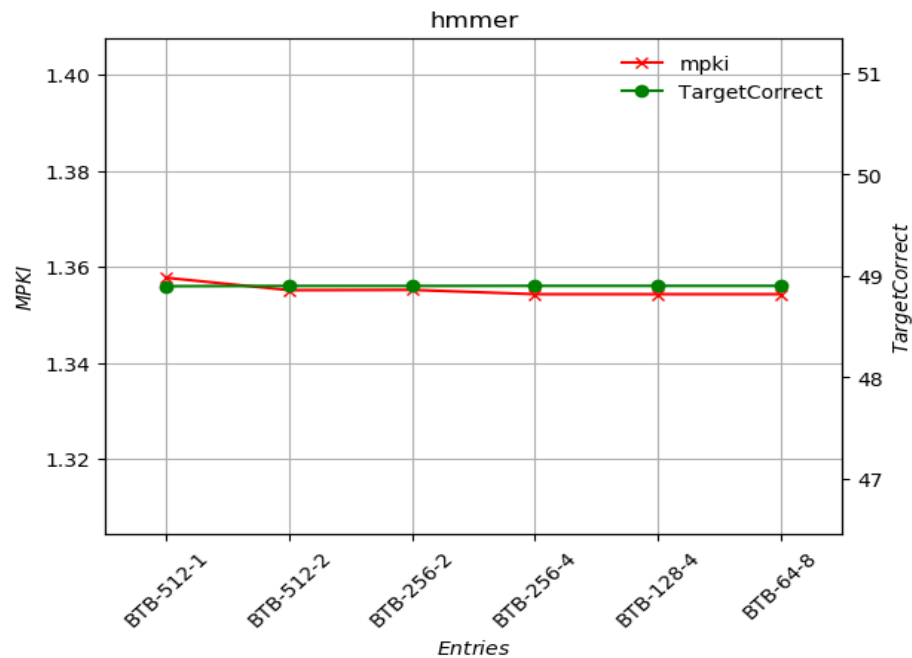
Δεν παρατηρείται κάποια αλλαγή στην απόδοση, ο BTB για το συγκεκριμένο μετροπρόγραμμα φαίνεται να μην μπορεί πλέον να επιτύχει καλύτερη απόδοση.



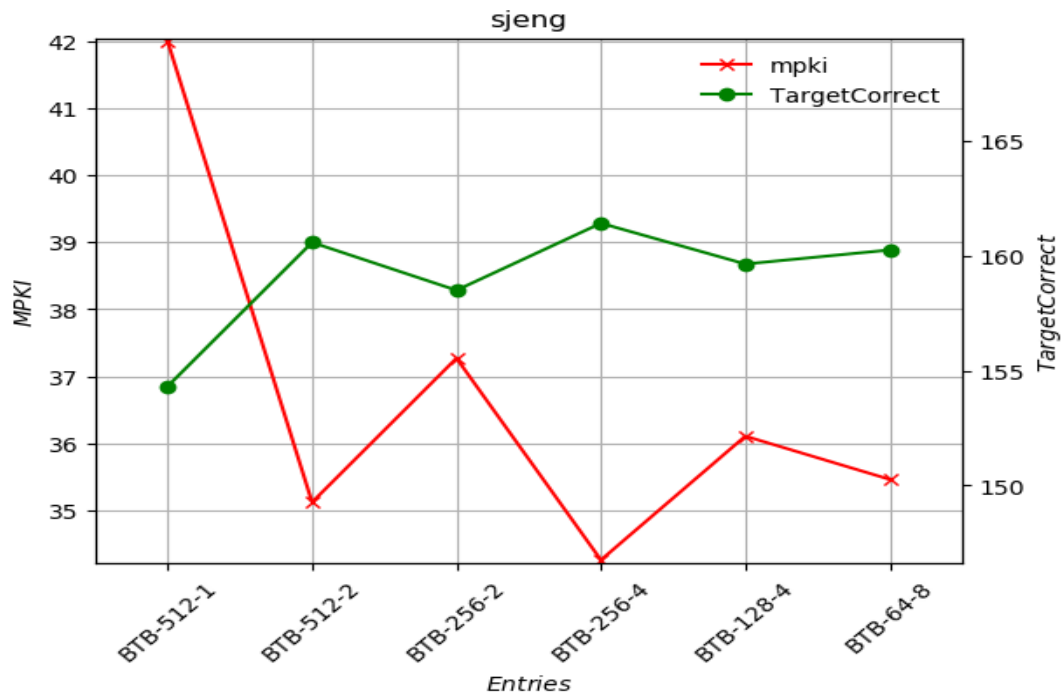
Ισχύει ότι και για το gcc



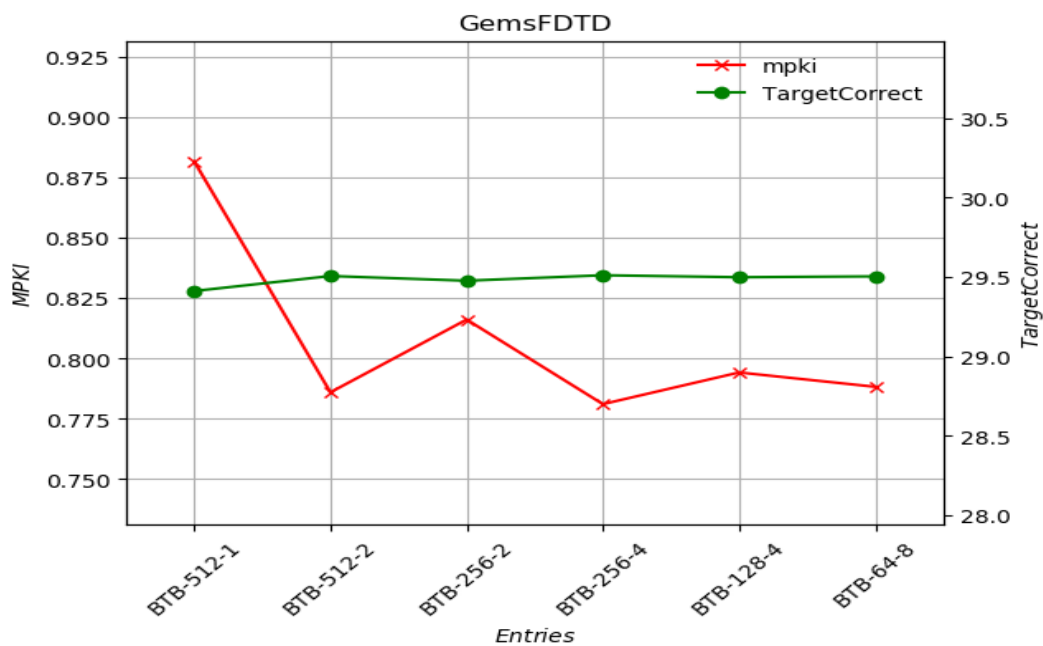
Όσο αυξάνει το associativity τόσο βελτιώνεται η απόδοση. Για τιμές όμως μεγαλύτερες του 4,ο ρυθμός αστοχίας δεν μειώνεται περαιτέρω.



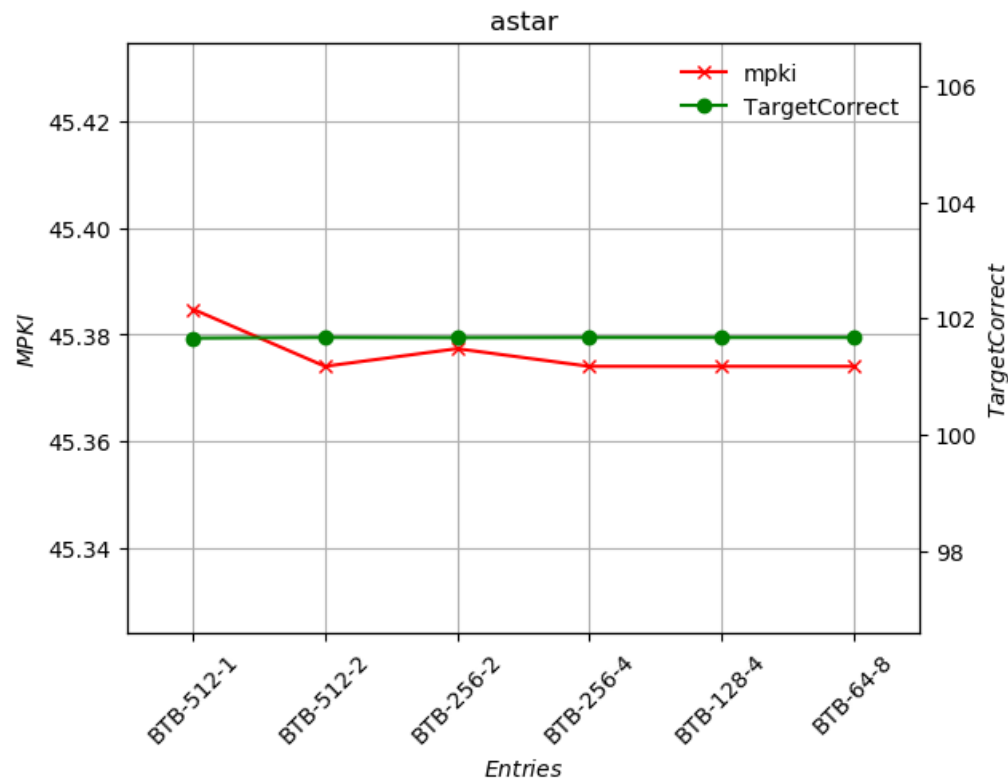
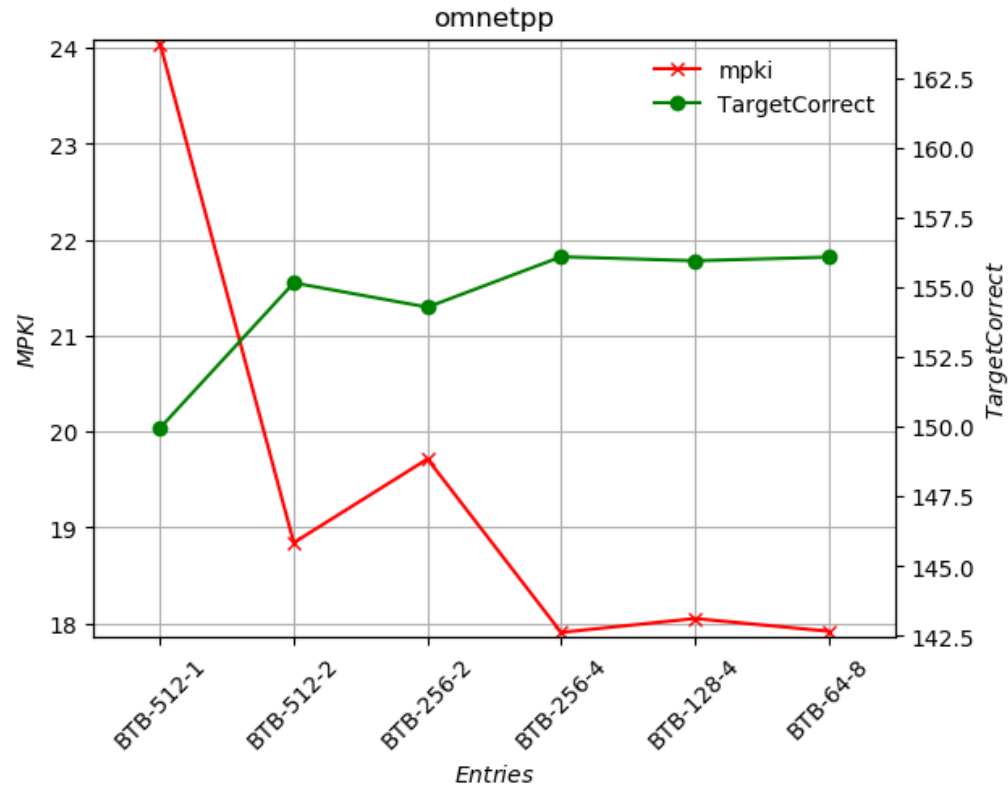
Ανεπηρέαστο από τις αλλαγές.

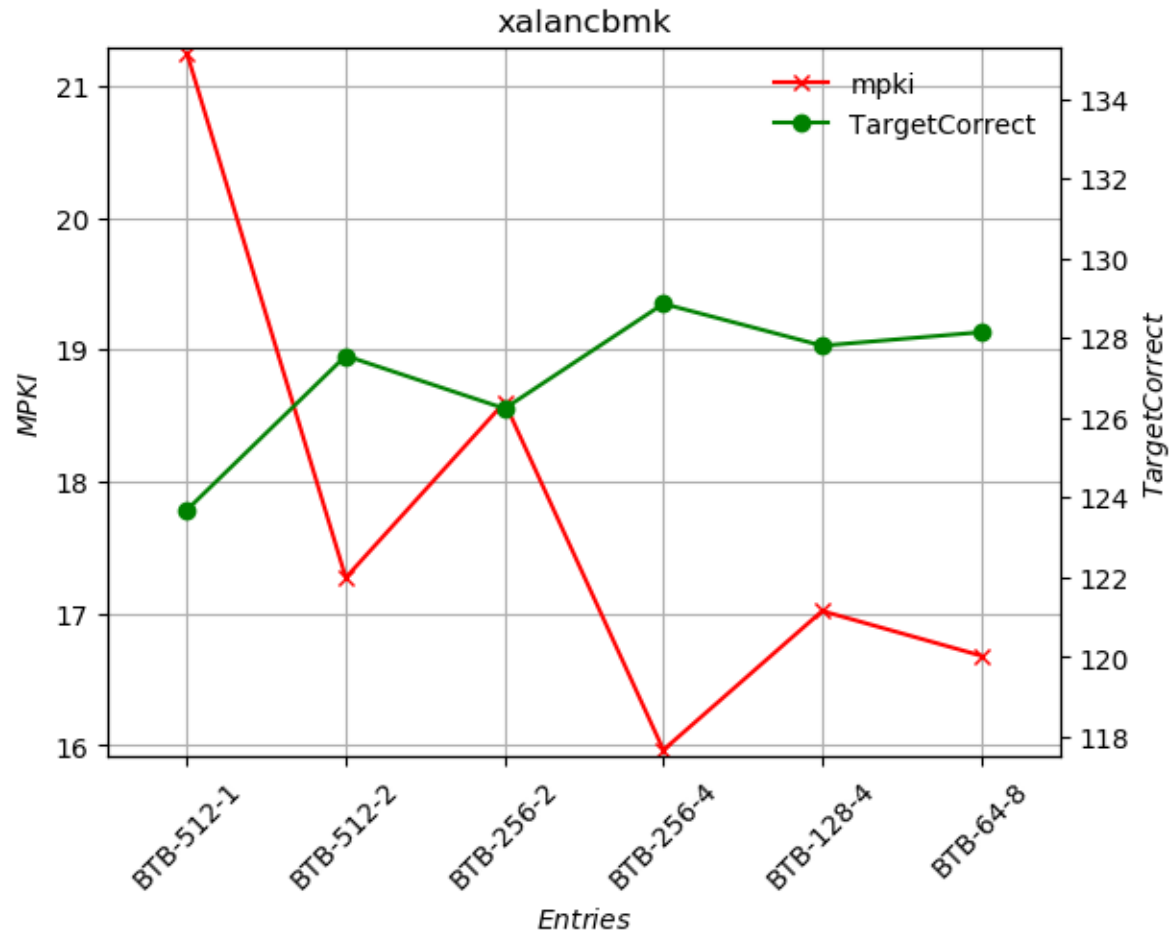


Ο βέλτιστος συνδυασμός είναι ο (256,4). Για περαιτέρω μείωση των entries ο ρυθμός αστοχίας αυξάνεται, ενώ περαιτέρω αύξηση του associativity πάνω από 4, μας δίνει μικρή μείωση του ρυθμού αστοχίας



Ισχύει ότι και για το sjeng. Το target correct μένει ανεπηρέαστο από τις αλλαγές.



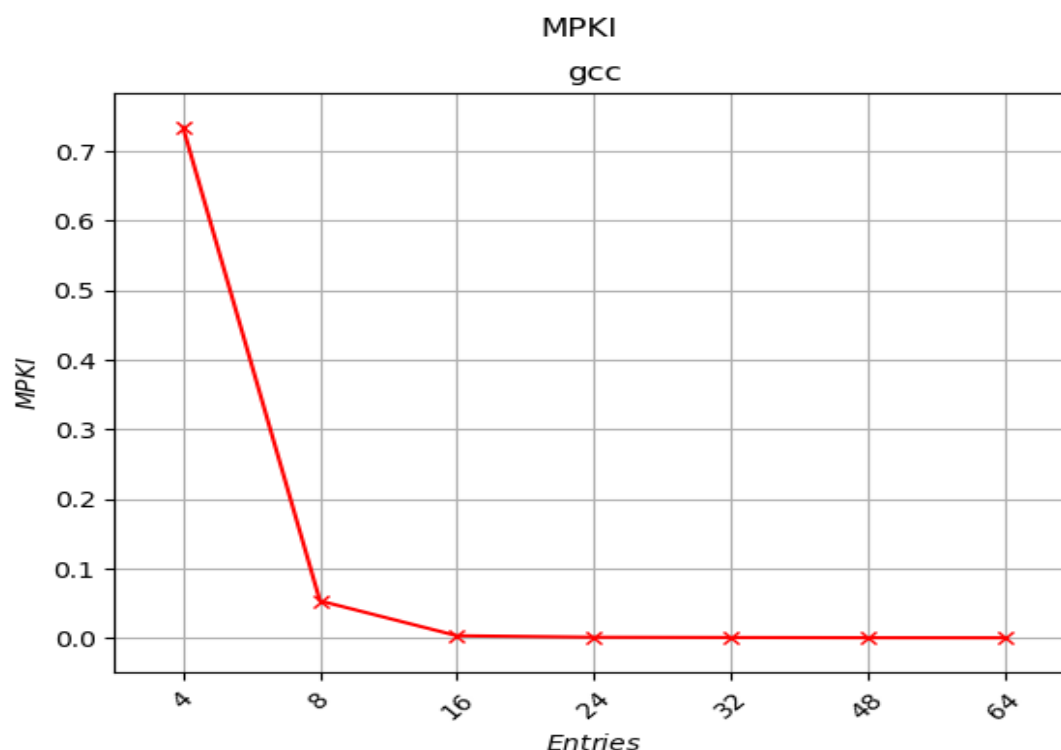


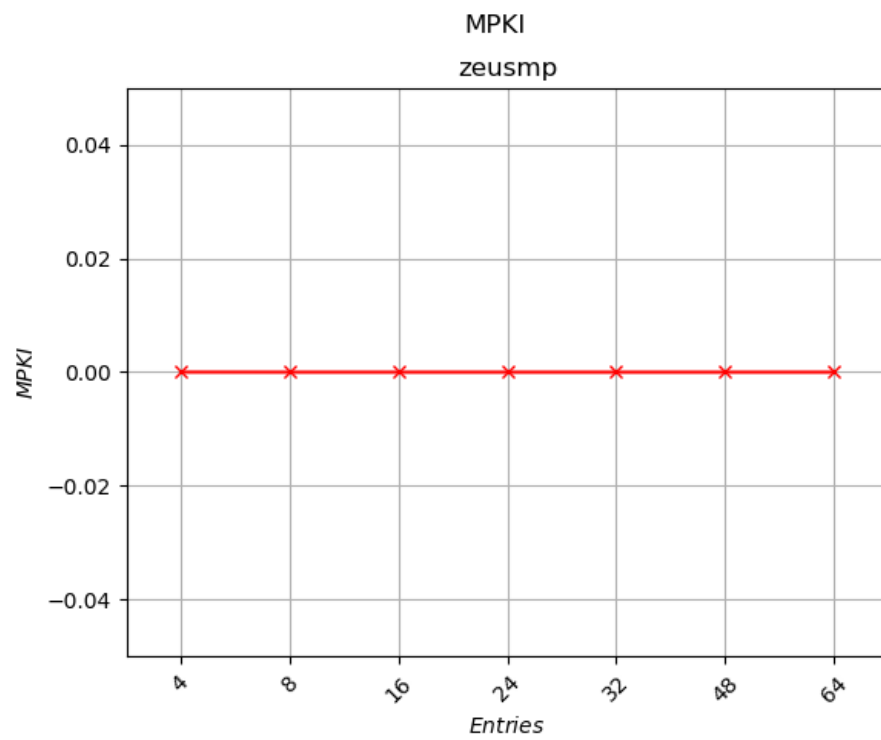
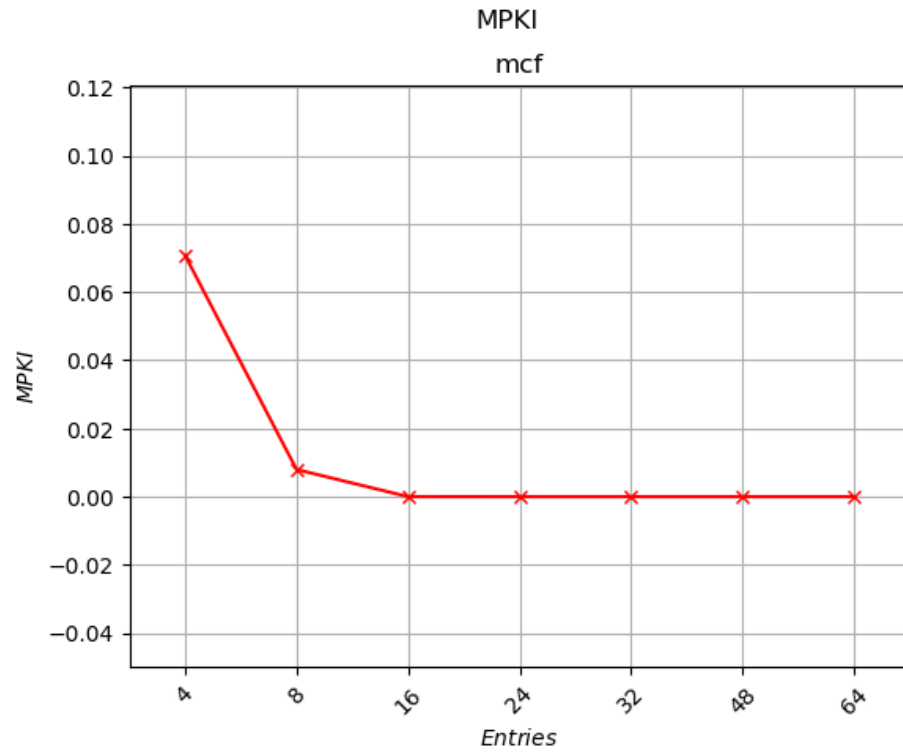
4.4 Μελέτη του RAS

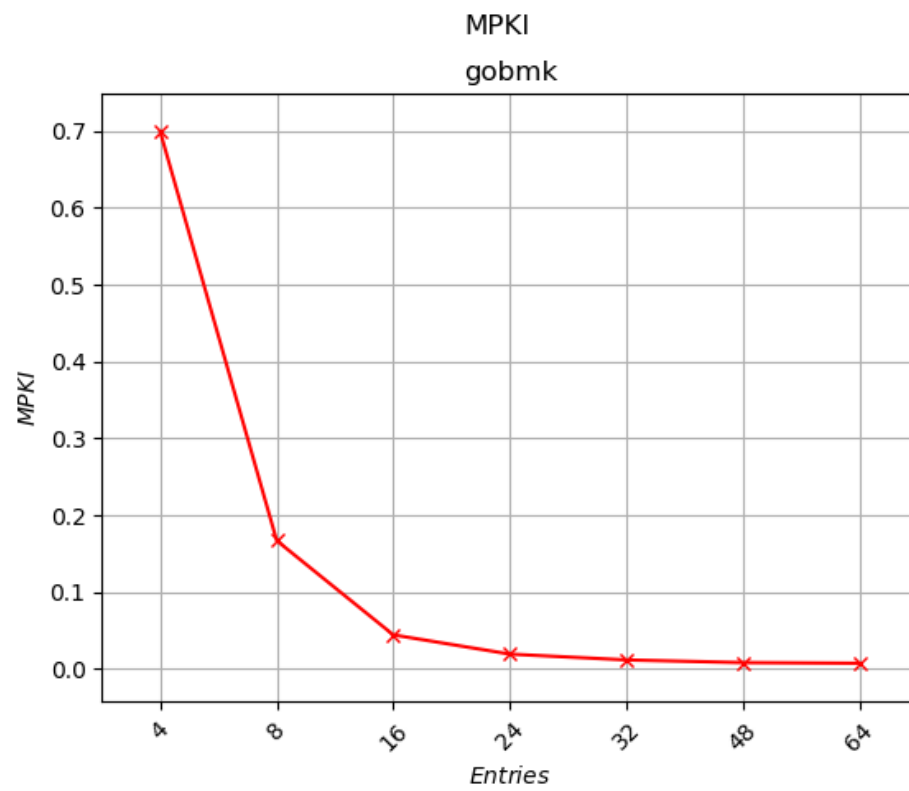
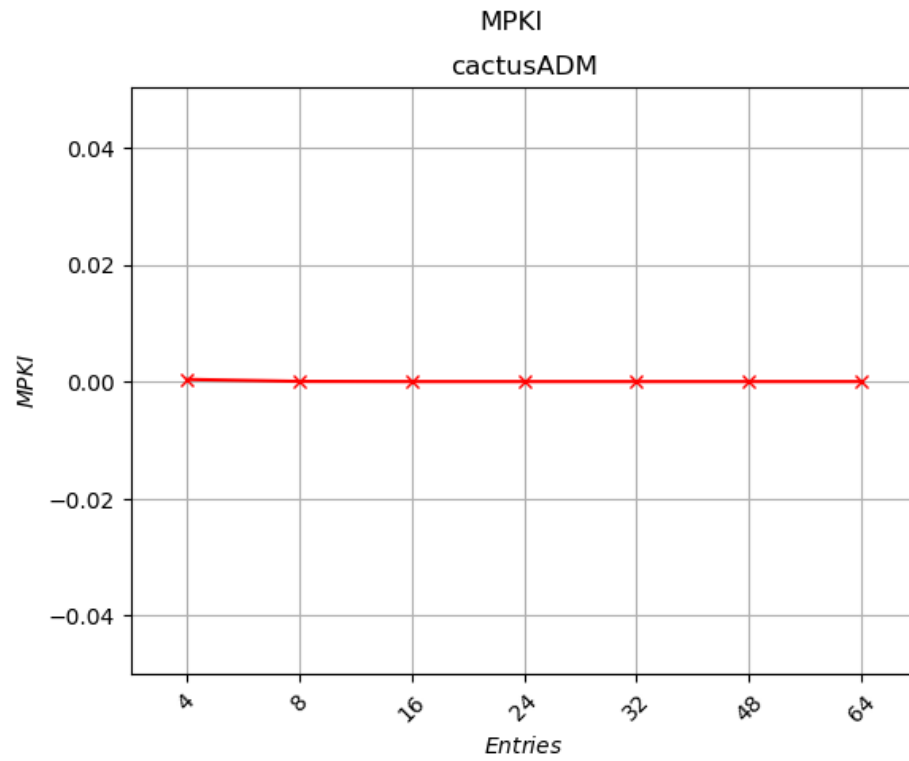
Καθώς οι εγγραφές στη RAS αυξάνονται, το MPKI μειώνεται. Αυτό συμβαίνει διότι όσο μεγαλύτερη RAS έχουμε, τόσο μεγαλύτερες αναδρομικές ή εμφωλευμένες συναρτήσεις μπορούμε να αποθηκεύσουμε μέσα στη RAS χωρίς να έχουμε λανθασμένη πρόβλεψη. Συνεπώς μια πολύ μεγάλη RAS θα έχει μηδενικές λάθος προβλέψεις, διότι μπορεί να αποθηκεύσει όλες τις εμφωλευμένες κλήσεις συναρτήσεων ενός προγράμματος.

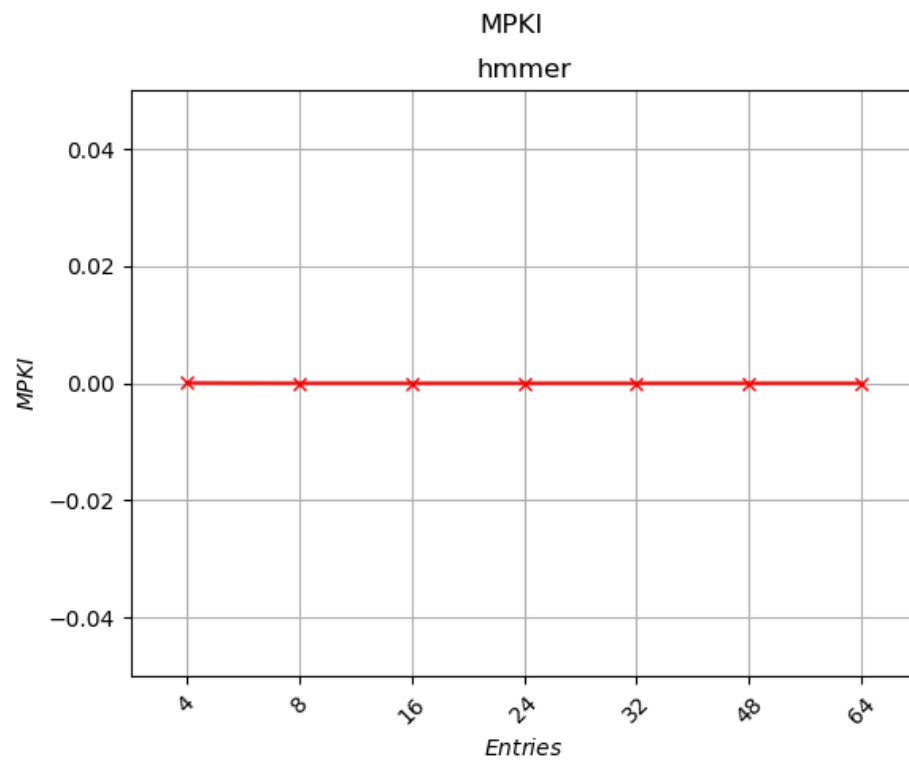
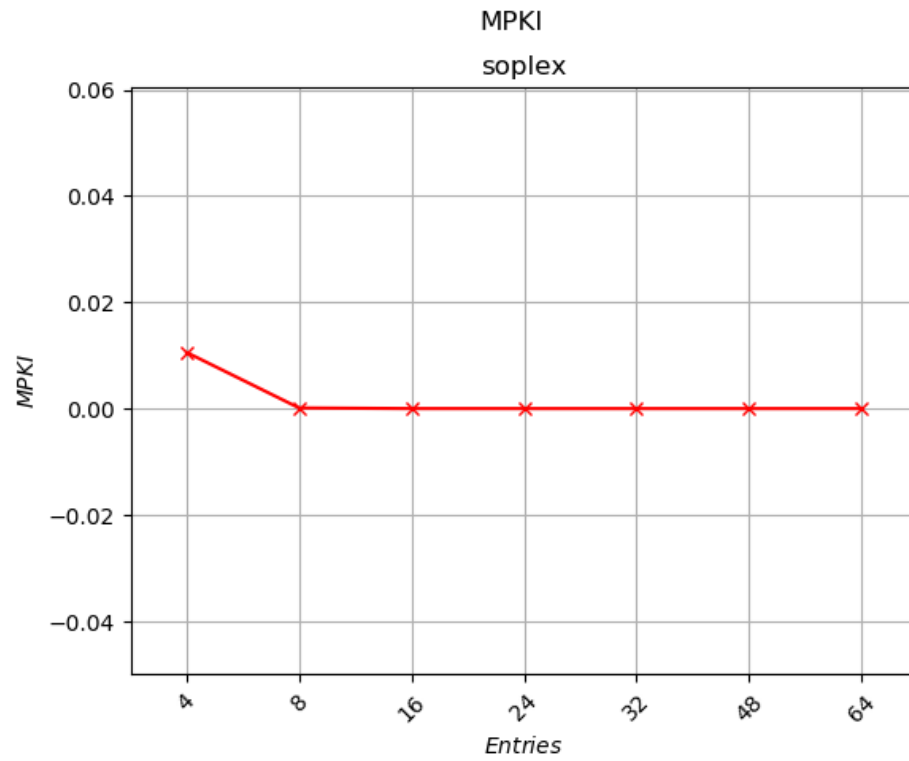
Βλέπουμε λοιπόν φθίνοντα διαγράμματα που συνήθως μετά τις 16 εγγραφές έχουν μηδενική αστοχία. Διαγράμματα με ευθεία γραμμή σημαίνει πως δε χρησιμοποιούν μεγάλο αριθμό εμφωλευμένων συναρτήσεων.

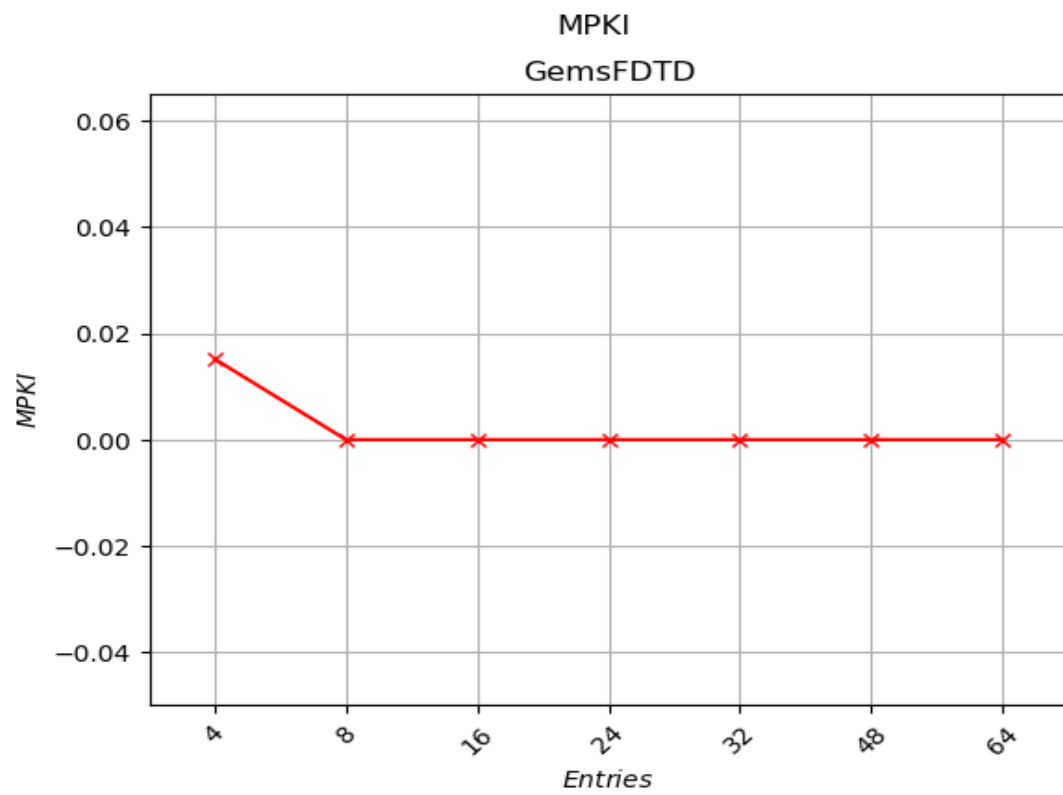
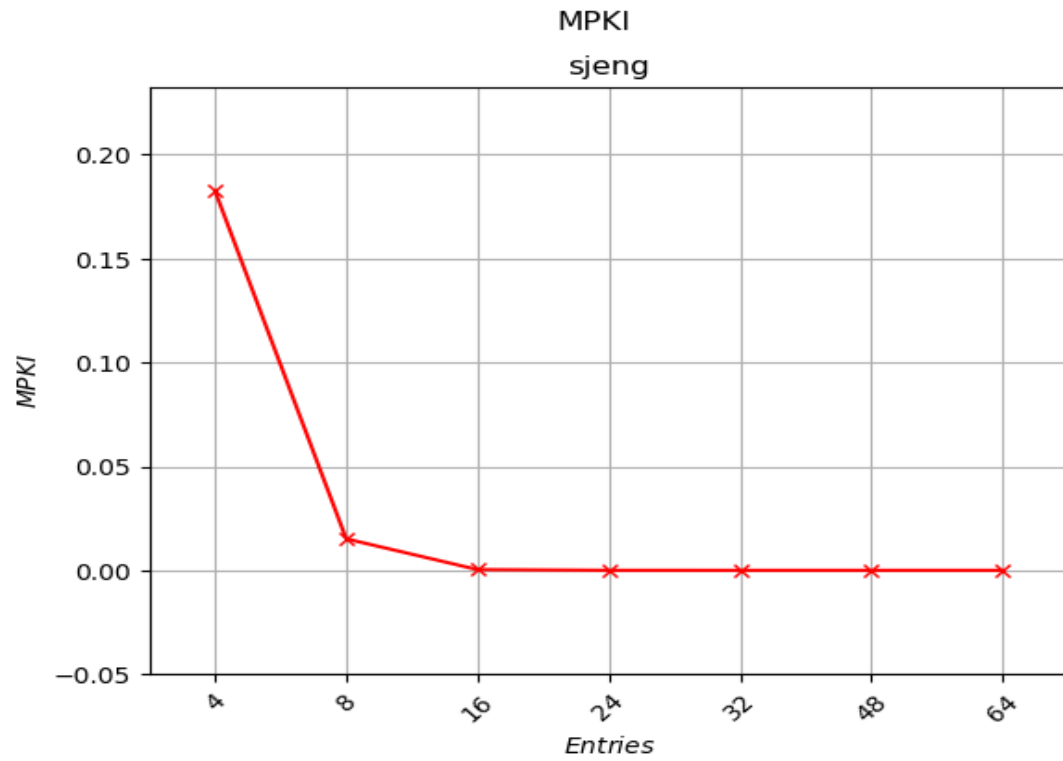
Βέλτιστο μέγεθος λοιπόν για τη RAS επιλέγονται τα 16 entries, με γνώμονα την απόδοση και την οικονομία.

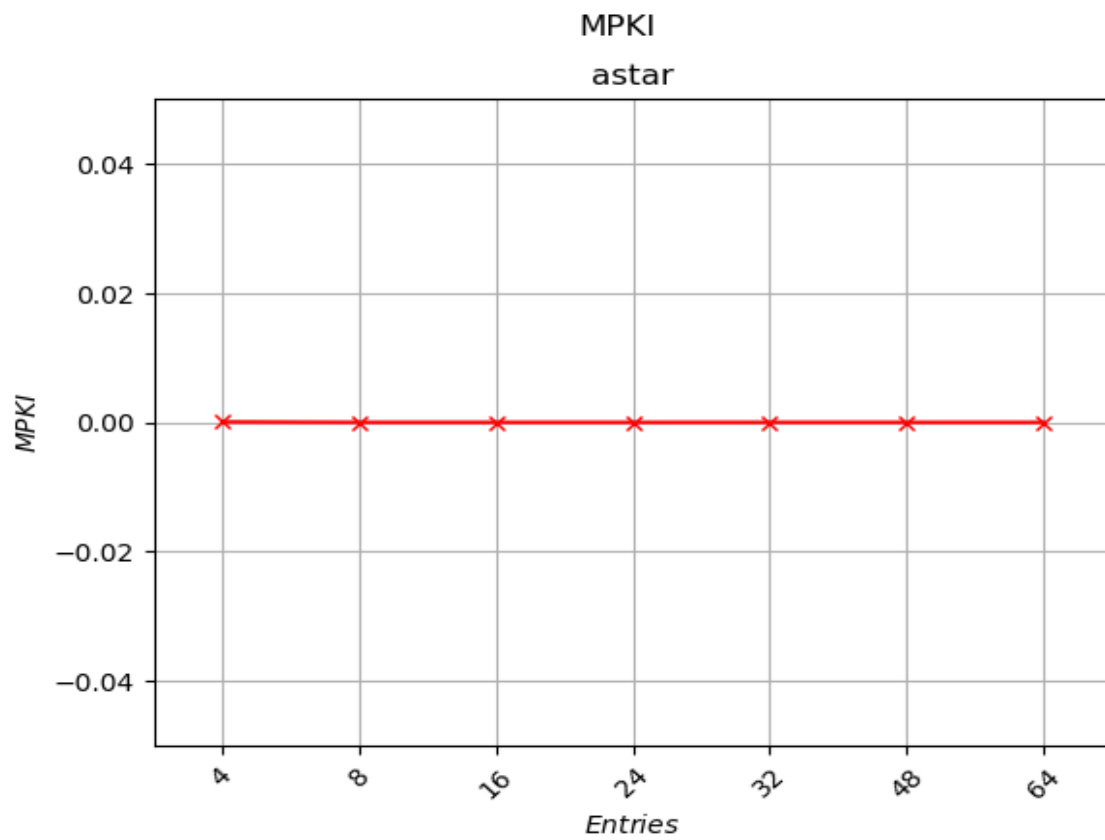
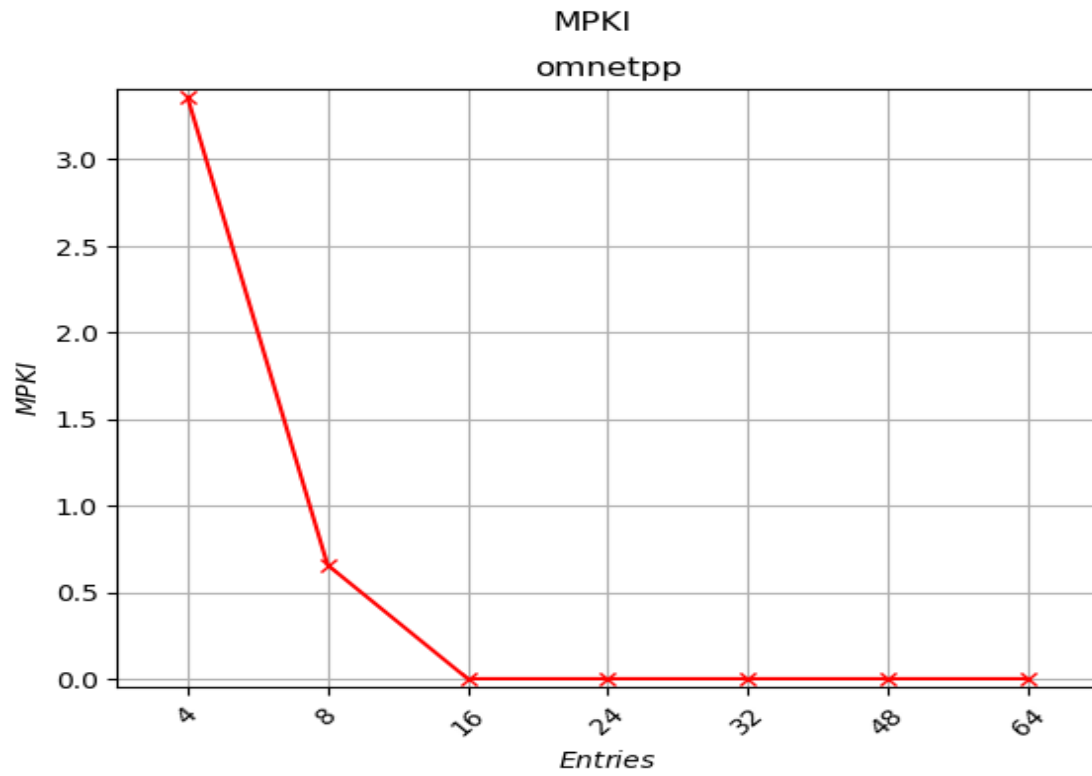


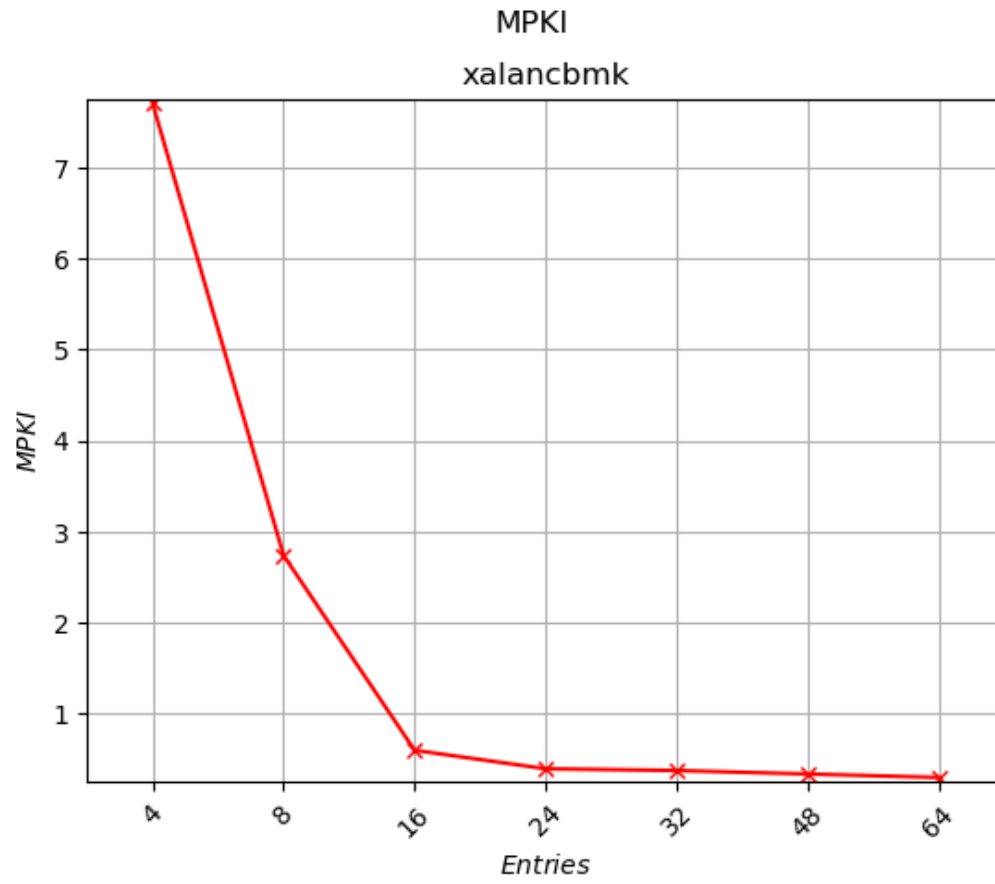






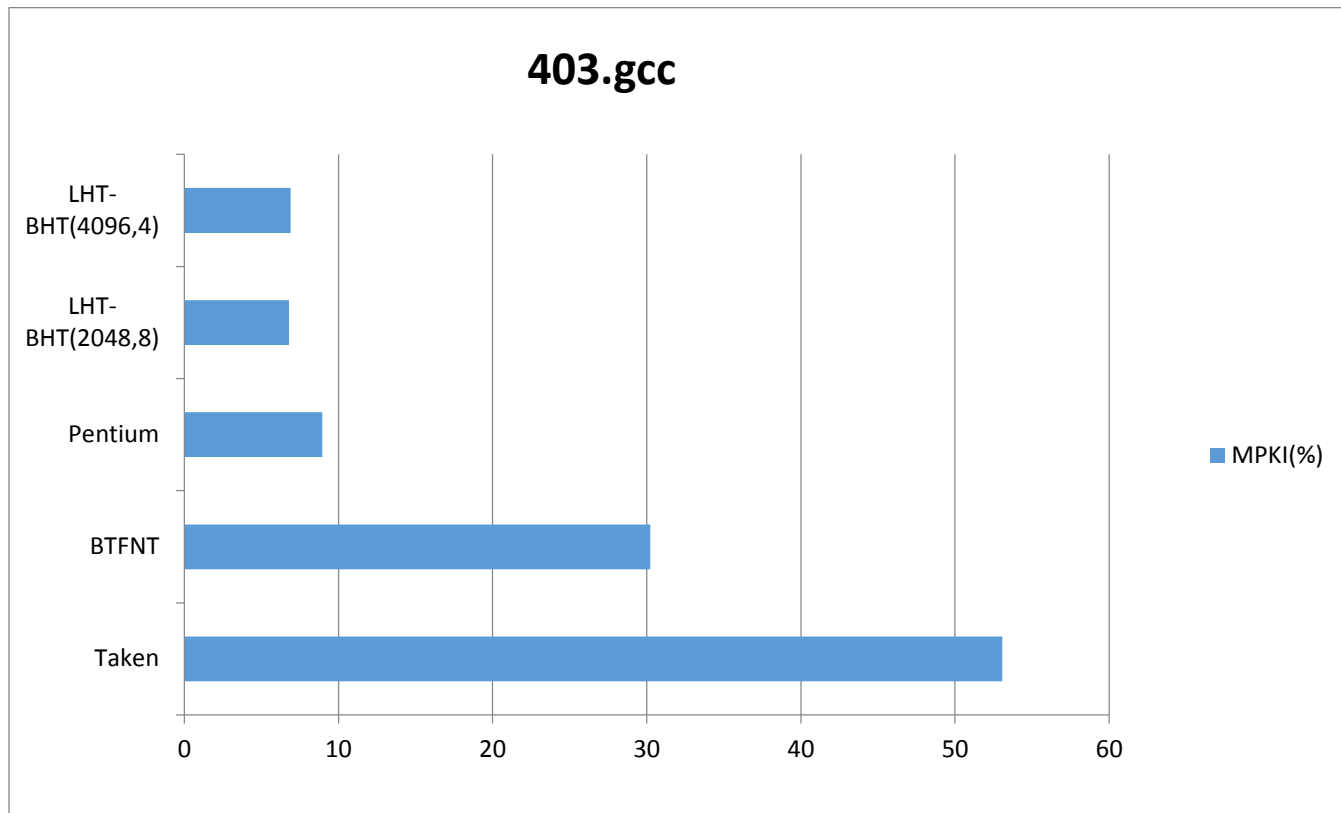


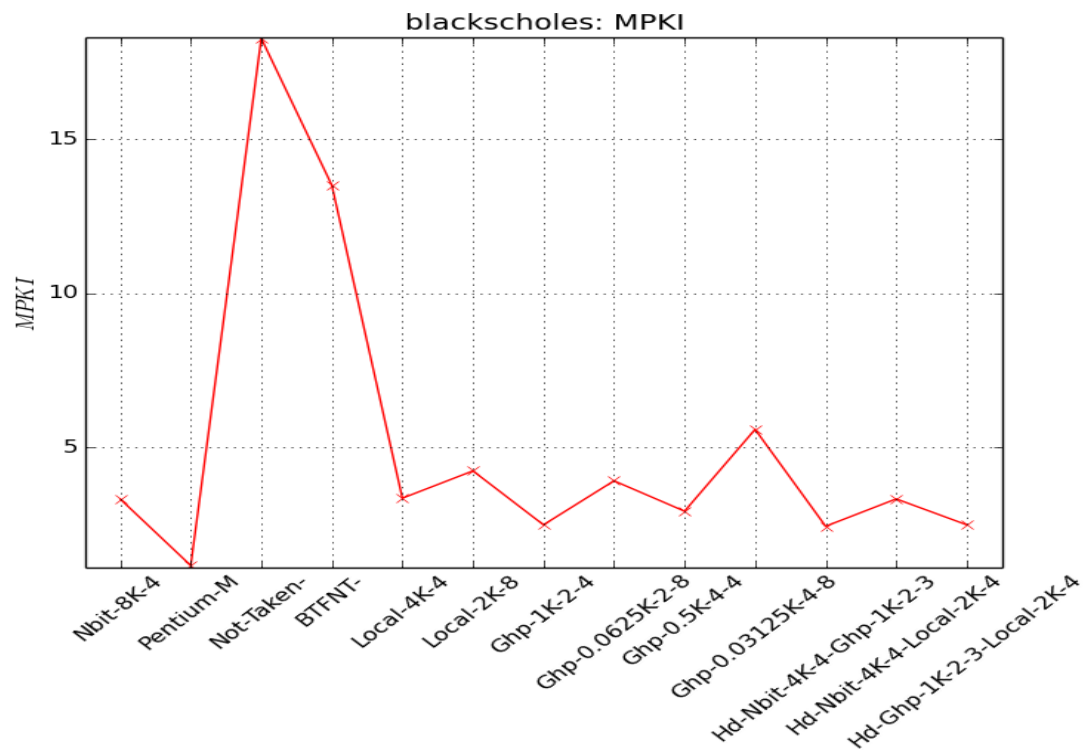
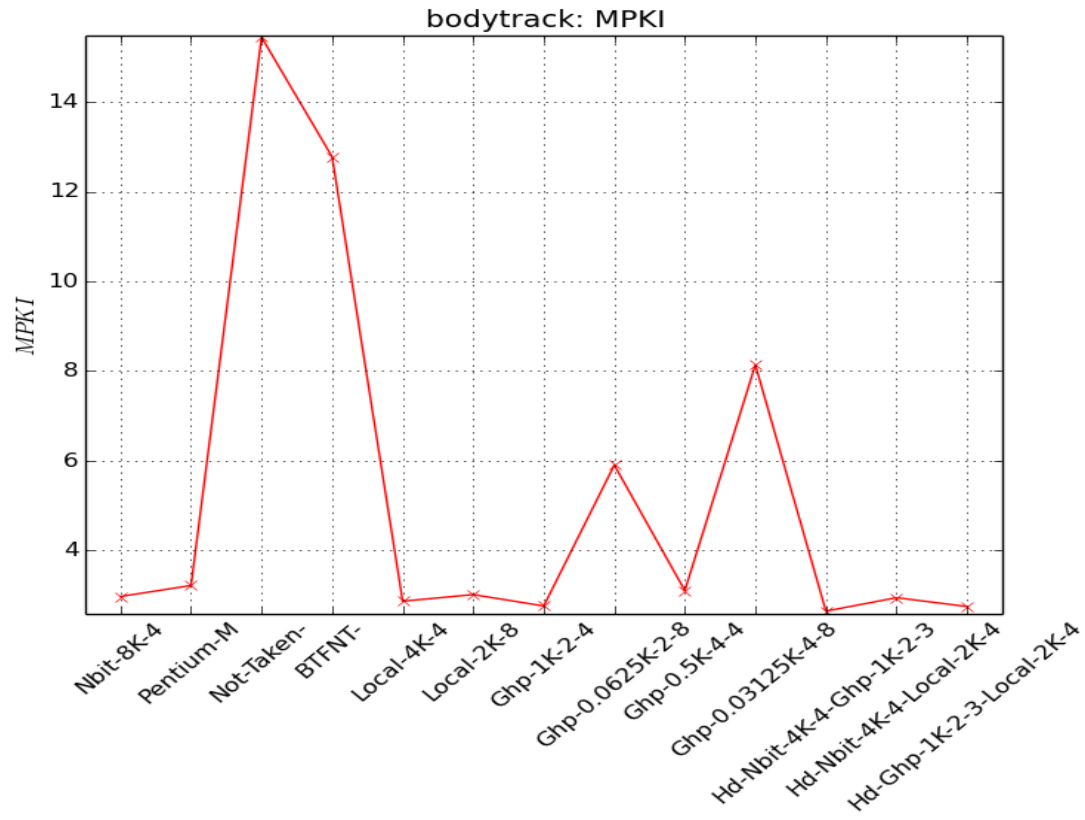


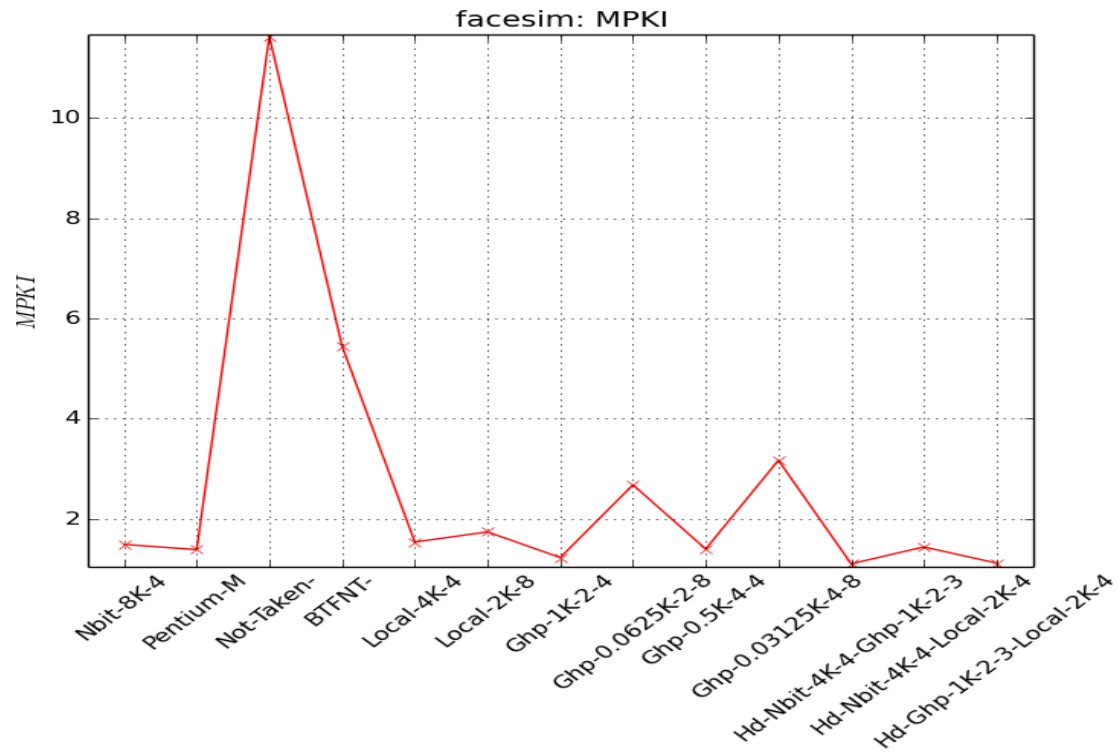
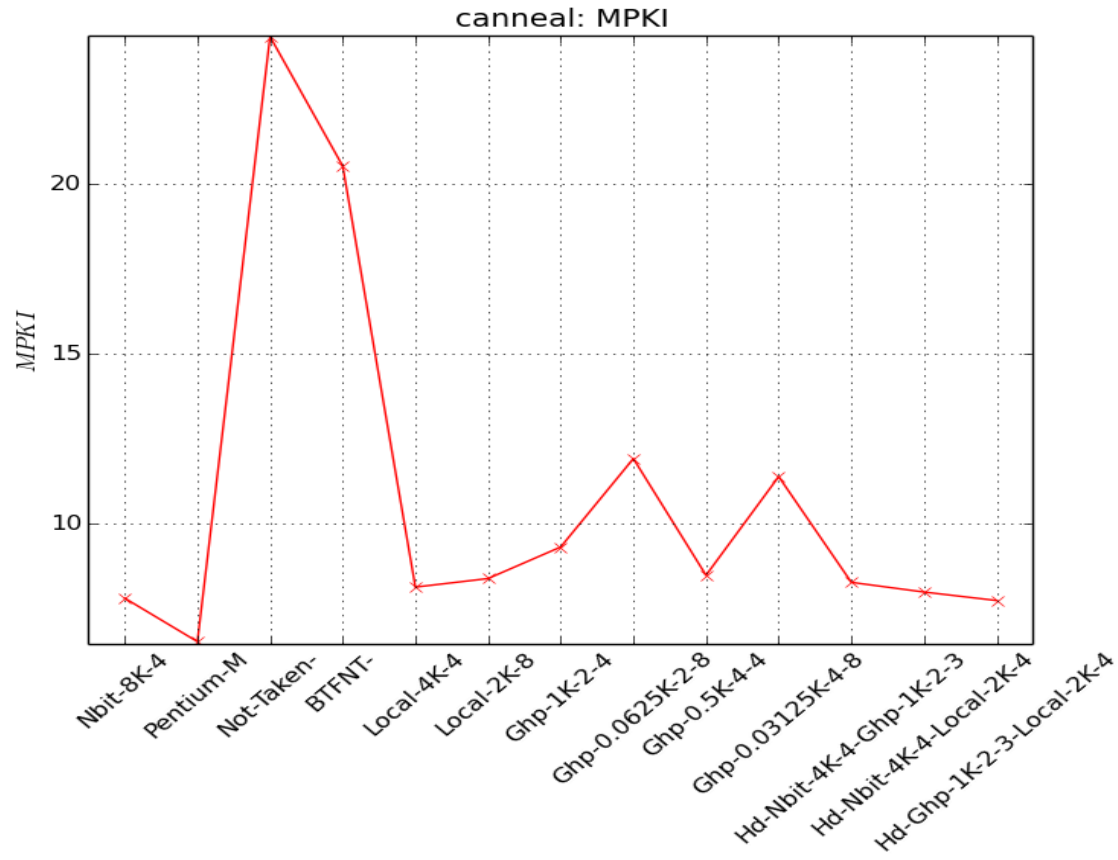


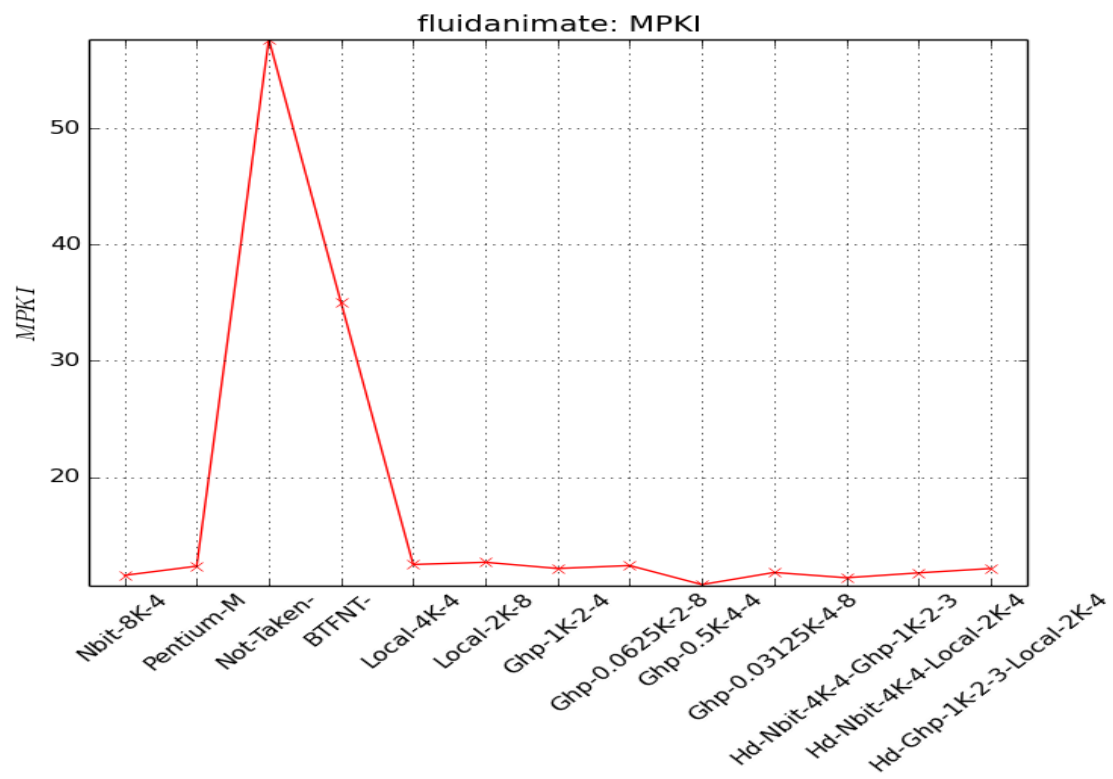
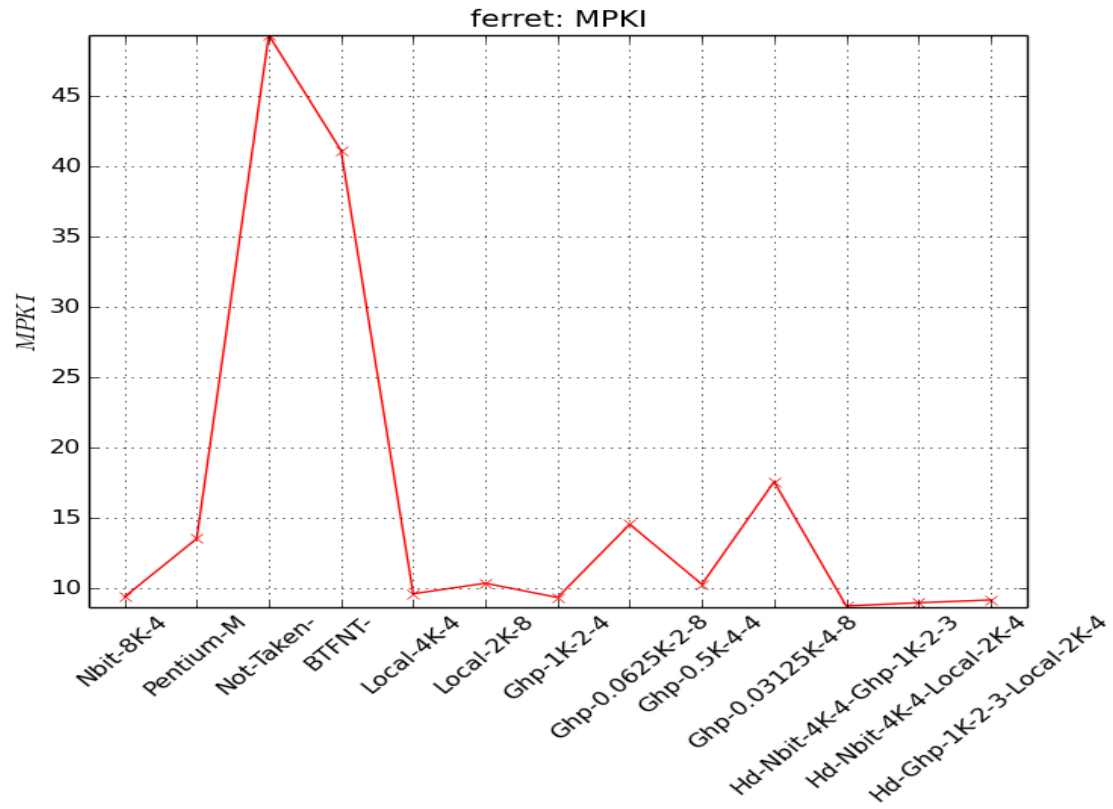
4.5 Σύγκριση διαφορετικών predictors

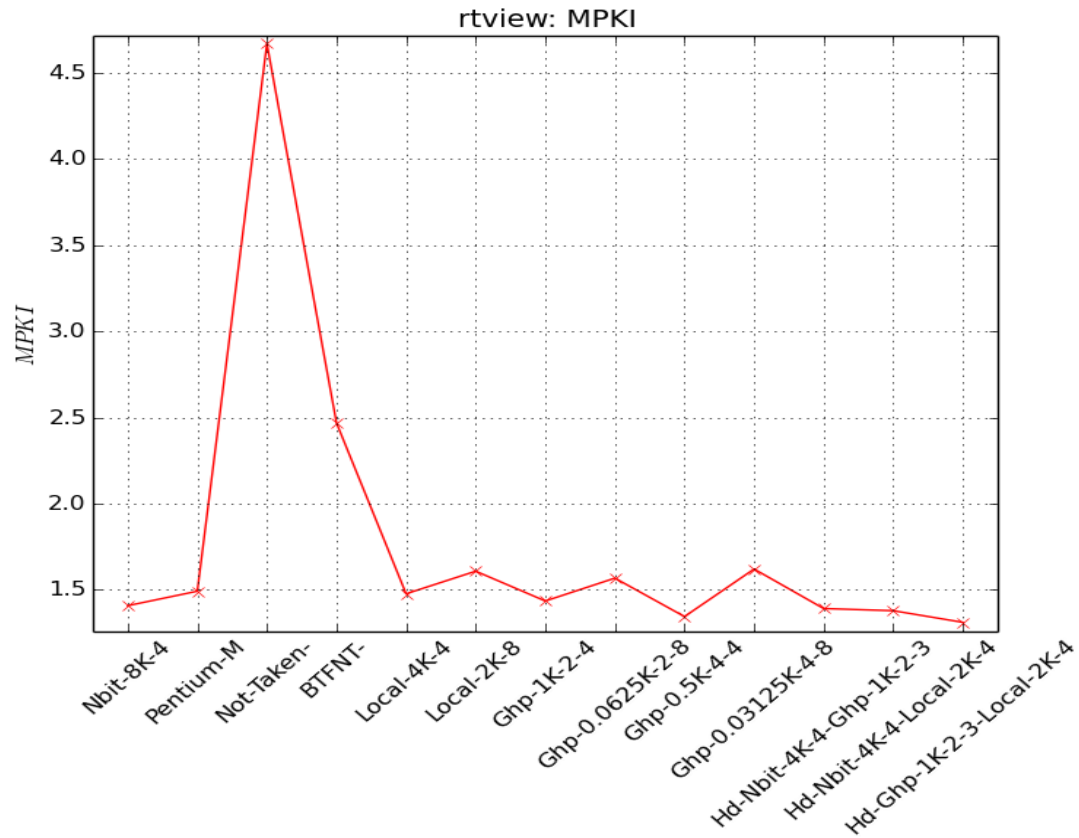
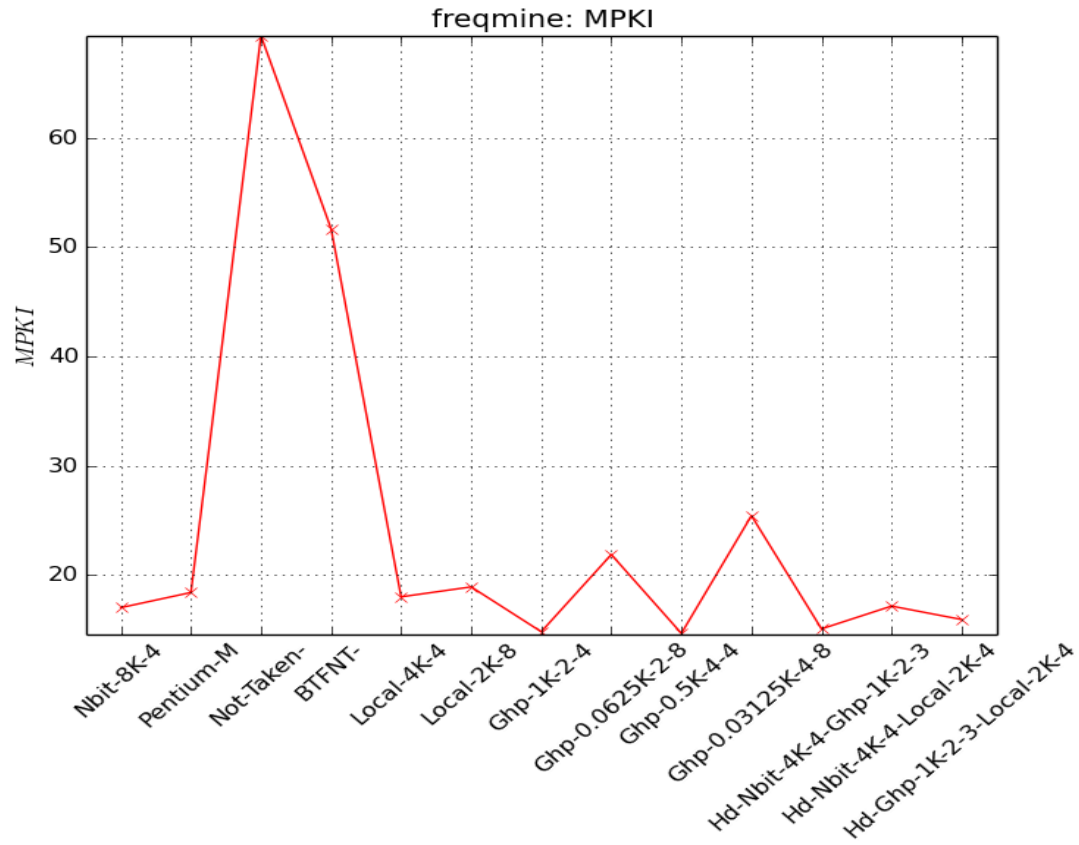
Αρχικά θα παρουσιαστούν σε γραφήματα όλοι οι branch predictors που υλοποιήθηκαν για το συγκεκριμένο ερώτημα και θα συγκριθούν με τους predictors από τα προηγούμενα ερωτήματα.

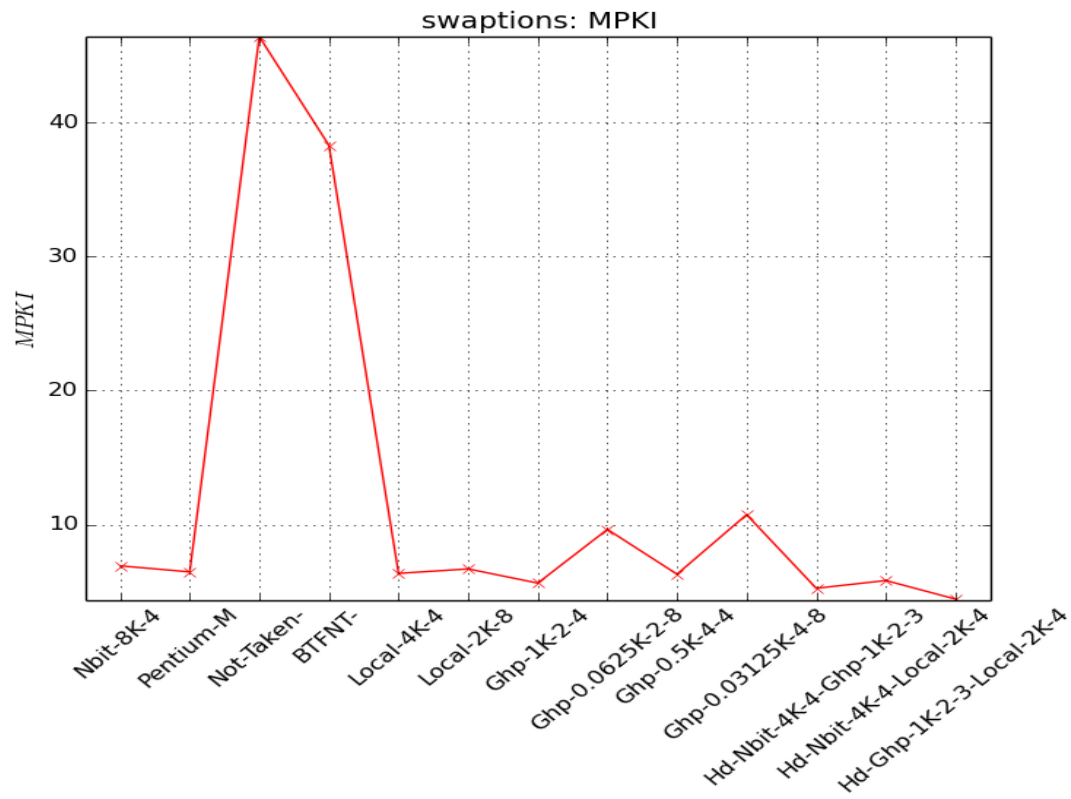
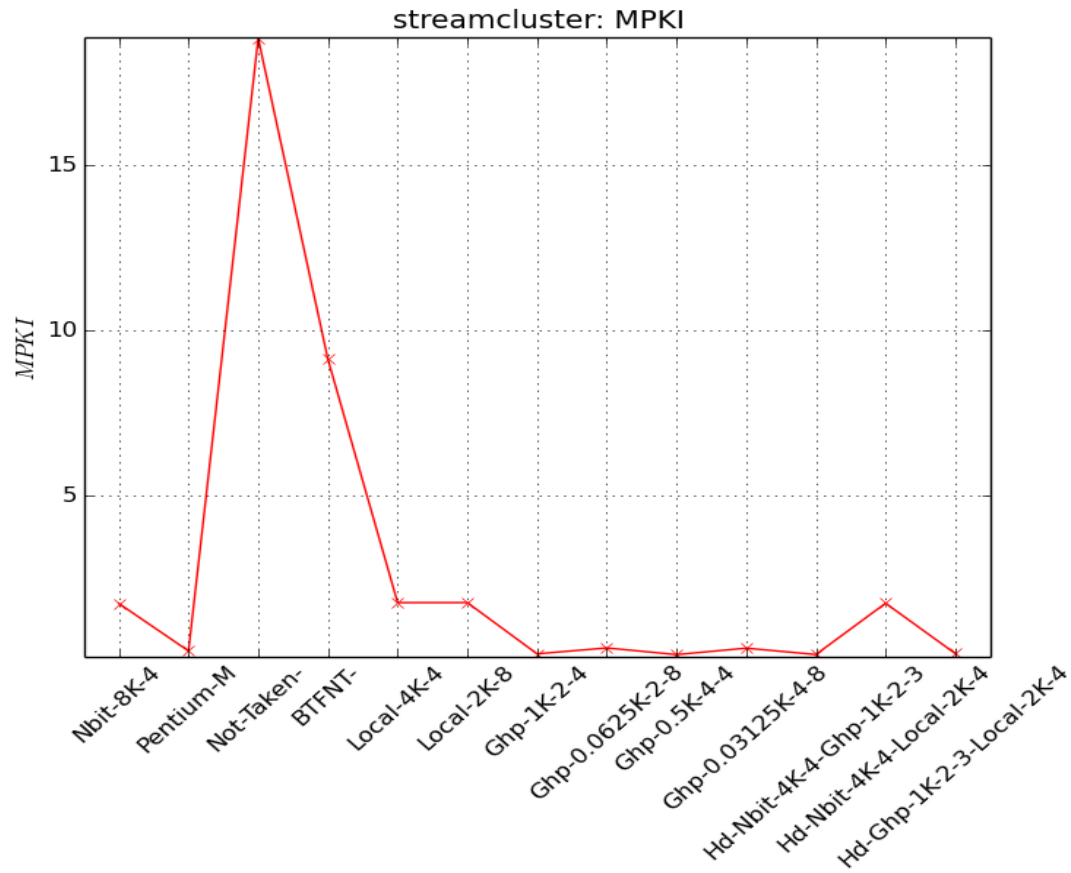












Παρατηρούμε πως

- Ο στατικός προβλέπτης διακλάδωσης έχει πολύ υψηλό ρυθμό αστοχίας και δεν είναι ικανοποιητικός.
- Ο BNTP προβλέπτης έχει πολύ καλύτερη αποδοτικότητα, αναλογικά με το κόστος κατασκευής τους.
- Με ελάχιστα υψηλότερο κόστος από τον BNTP, αλλά καλύτερες μετρικές, συναντούμε τον 3-bit predictor που επιλέξαμε από το ερώτημα 2.
- Ο Pentium-M έχει πολύ μειωμένο MPKI σε σχέση με τους προηγούμενους 3, όμως σε αυτό το επίπεδο πολυπλοκότητας/κόστους συναντούμε καλύτερους predictors.
- Οι local και global history predictors έχουν ελάχιστα αυξημένο κόστος από τον Pentium-M, όμως μας παρέχουν αυξημένη αποδοτικότητα.
- Οι tournament predictors περιμέναμε να δίνουν μειωμένο MPKI σε σχέση με τους local και history predictors, όμως αυτό δε συμβαίνει. Ίσως οι επιλογές για τους P0, P1 predictors να μην ήταν κατάλληλες και αυτό οδήγησε σε μειωμένη αποδοτικότητα. Χρειάζονται περισσότερες προσομοιώσεις και έρευνες προκειμένου να βελτιωθεί περαιτέρω η απόδοση των tournament predictors.

Τελικά καταλήγουμε πως με βάση τα δεδομένα που έχουμε συλλέξει (και κρίνοντας μόνο με αυτά) ο βέλτιστος προβλέπτης είναι ο Global History two-level predictor με χαρακτηριστικά:

ο PHT entries = 16384

ο PHT n-bit counter length = 2

ο BHR length = 10