



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

"Προηγμένα Θέματα Αρχιτεκτονικής Υπολογιστών"

3^η Άσκηση

Ορφέας Μενής-Μαστρομιχαλάκης
ΑΜ: 03114052

1. Εισαγωγή

Στην άσκηση αυτή θα μελετήσουμε τα χαρακτηριστικά των σύγχρονων superscalar, out-of-order επεξεργαστών και ο τρόπος με τον οποίο επηρεάζουν την απόδοση του συστήματος, την κατανάλωση ενέργειας καθώς και το μέγεθος του chip του επεξεργαστή

8. Μετροπρογράμματα

Στα πλαίσια της παρούσας άσκησης χρησιμοποιήσαμε τα SPEC_CPU2006 benchmarks, όπως και στην προηγούμενη άσκηση. Συγκεκριμένα, χρησιμοποιήσαμε τα παρακάτω 12 benchmarks:

1. 403.gcc
2. 429.mcf
3. 434.zeusmp
4. 436.cactusADM
5. 445.gobmk
6. 450.soplex
7. 456.hmmer
8. 458.sjeng
9. 459.GemsFDTD
10. 471.omnetpp
11. 473.astar
12. 483.xalancbmk

9. Πειραματική Αξιολόγηση

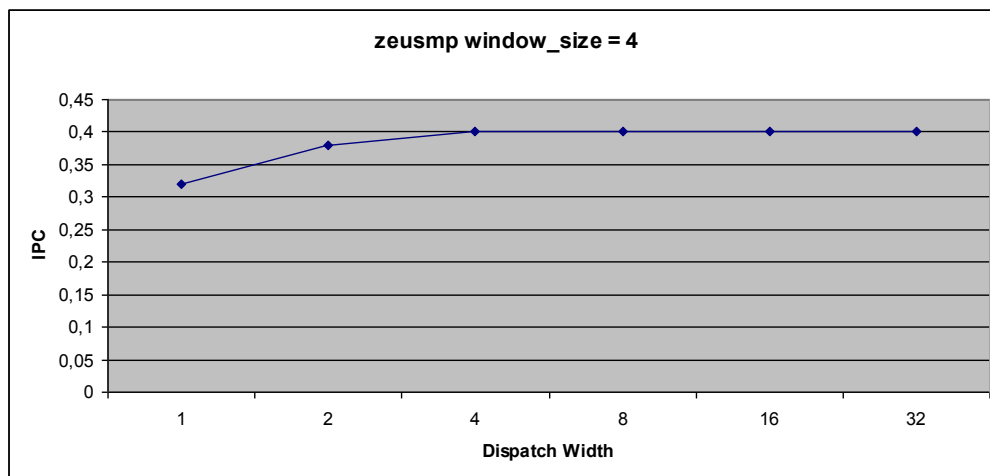
Εκτελέστε όλα τα benchmarks για κάθε διαφορετικό επεξεργαστή που προκύπτει από το συνδυασμό των παρακάτω τιμών για τις παραμέτρους `dispatch_width` και `window_size`:

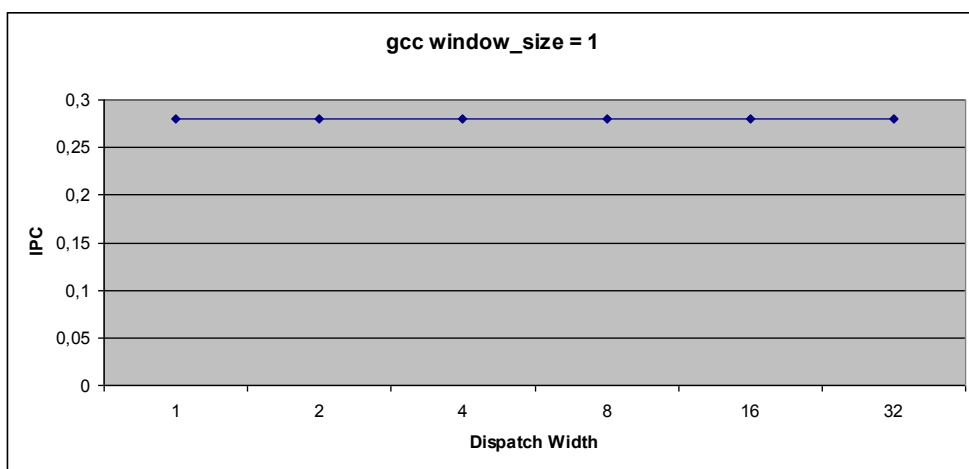
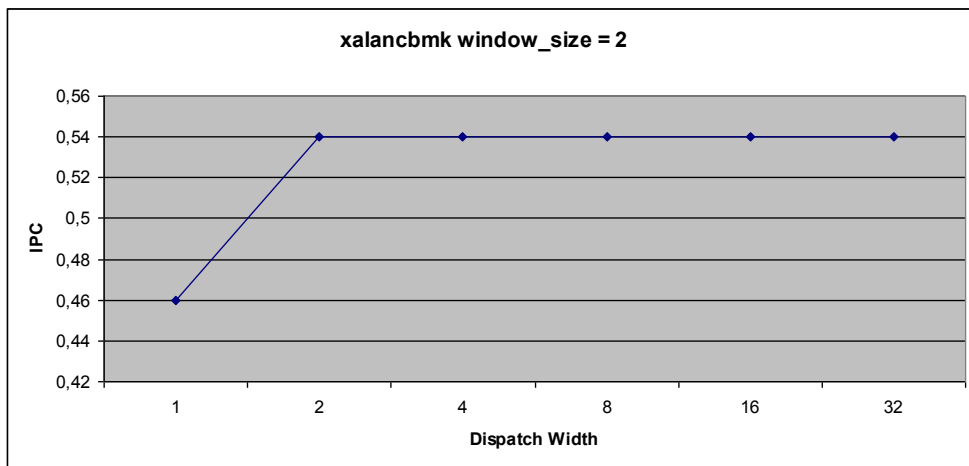
<code>dispatch_width</code>	1	2	4	8	16	32					
<code>window_size</code>	1	2	4	8	16	32	64	128	256	512	

i)

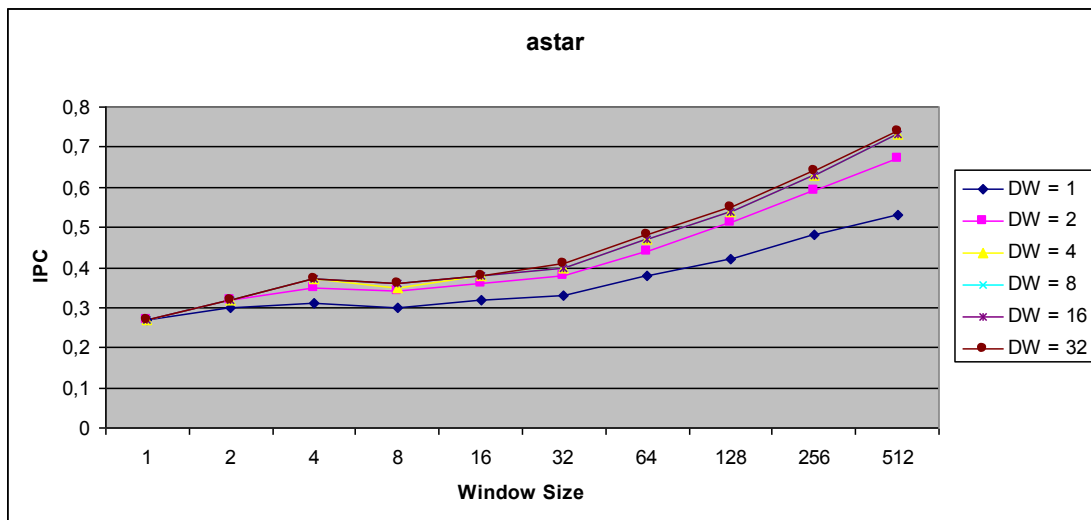
Δεν χρειάζεται να προσομοιώσουμε και τους 60 διαφορετικούς επεξεργαστές παρά μόνο αυτούς που έχουν `window size` τουλάχιστον όσο και το `dispatch width`. Ο επεξεργαστής αντλεί εντολές προς εκτέλεση απ' το ROB, οπότε δεν έχει νόημα να ξεκινήσει την εκτέλεση περισσότερων εντολών ανά κύκλο από όσες βρίσκονται στον ROB. Αυτό ισχύει γιατί δεν θα υπάρχουν διαθέσιμες εντολές να εκτελεστούν. Γι' αυτό το λόγο εκτελέσαμε μόνο τα conf που είχαν `window size` μεγαλύτερο/ίσο του `dispatch width`.

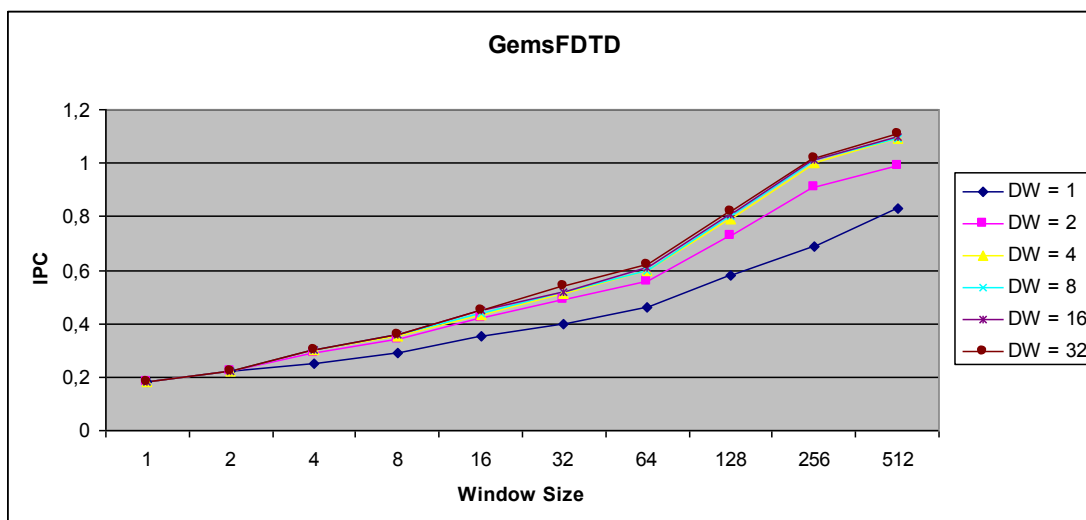
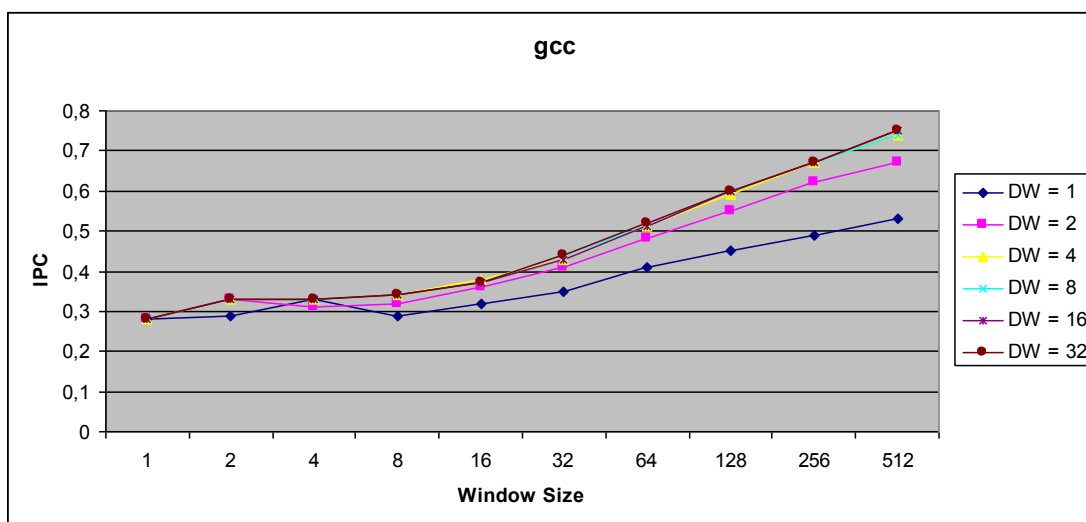
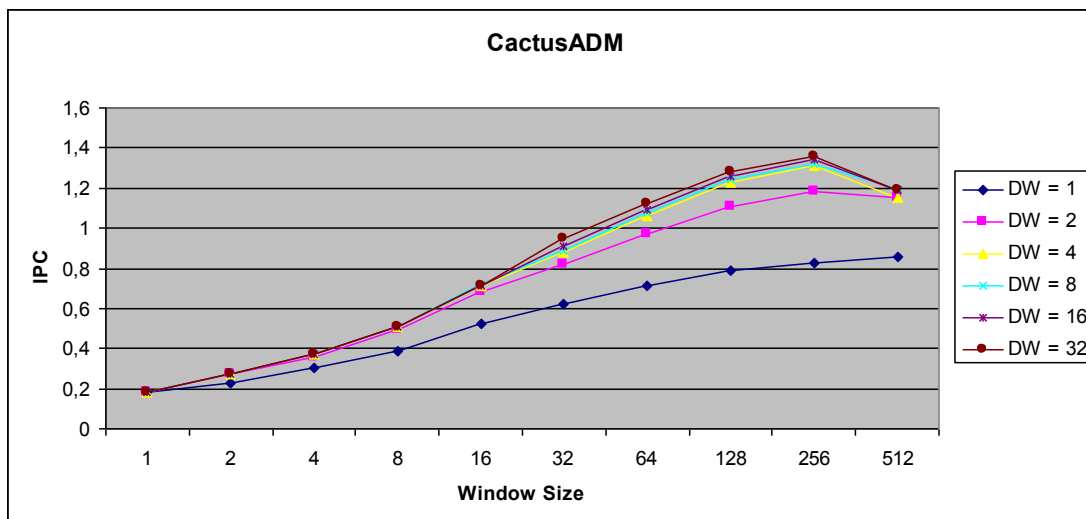
Ο παραπάνω ισχυρισμός επαληθεύτηκε και πειραματικά μιας και όπως βλέπουμε στα ακόλουθα 3 διαγράμματα, η αλλαγή του `dispatch width` σε τιμές μεγαλύτερες του `window size` μας έδωσε ακριβώς το ίδιο IPC με την περίπτωση `dispatch width = window size`.

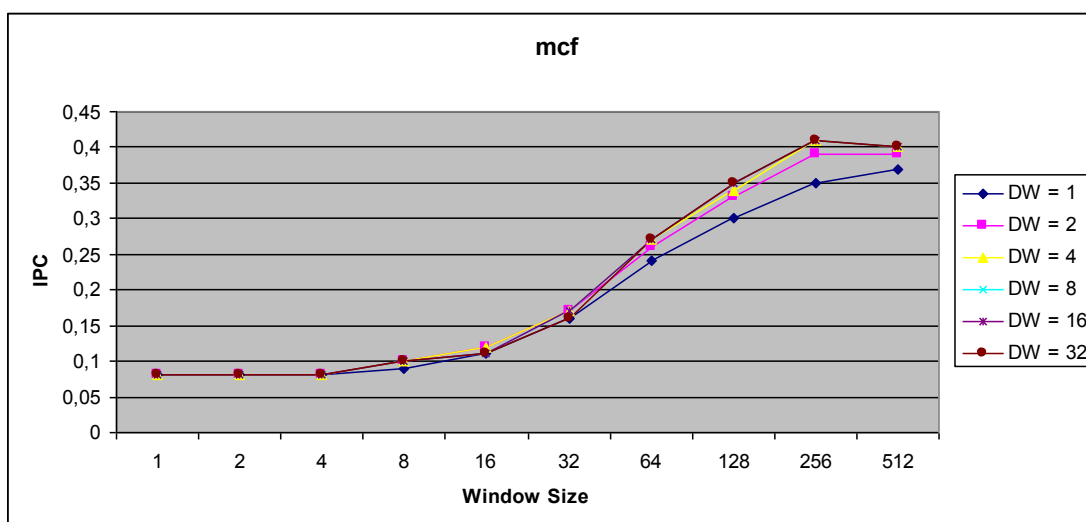
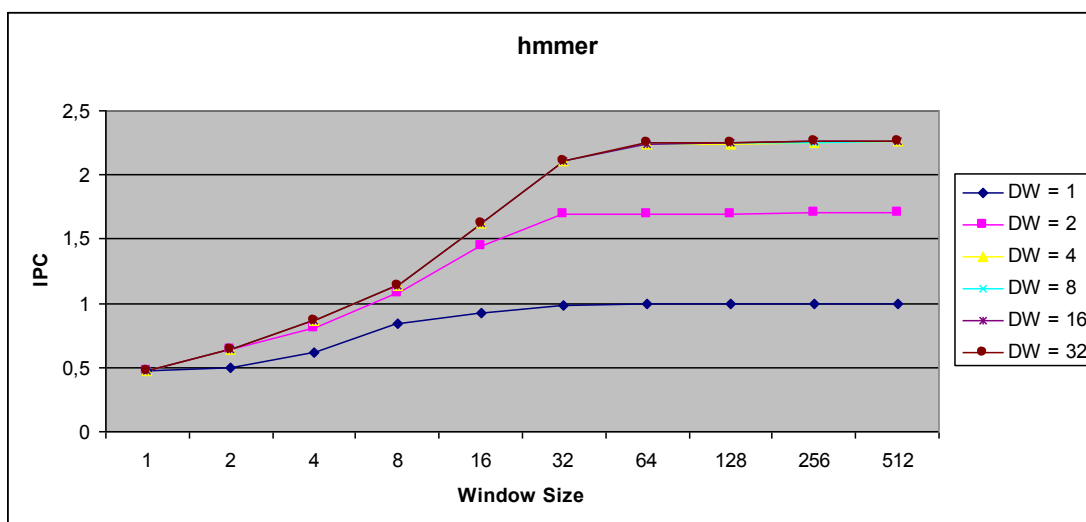
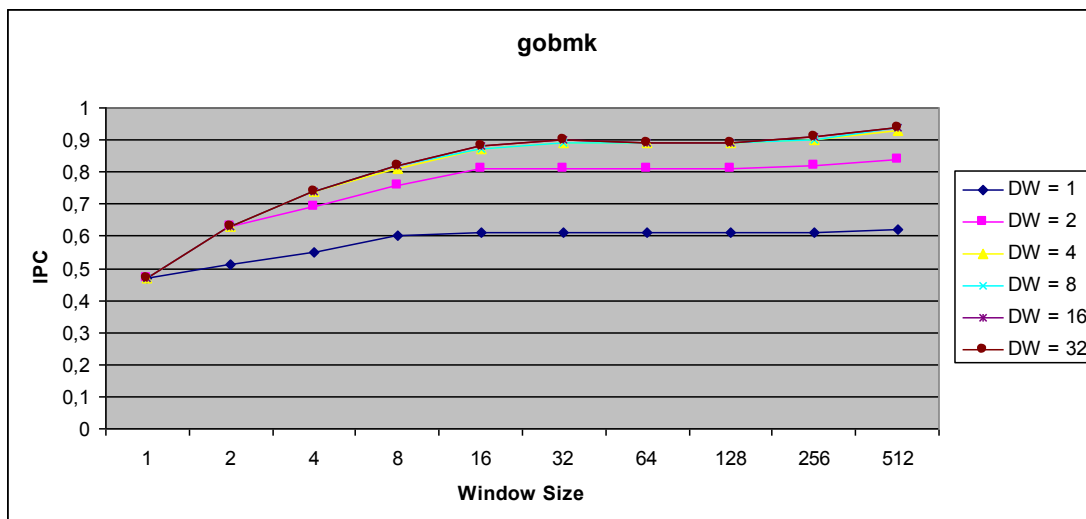


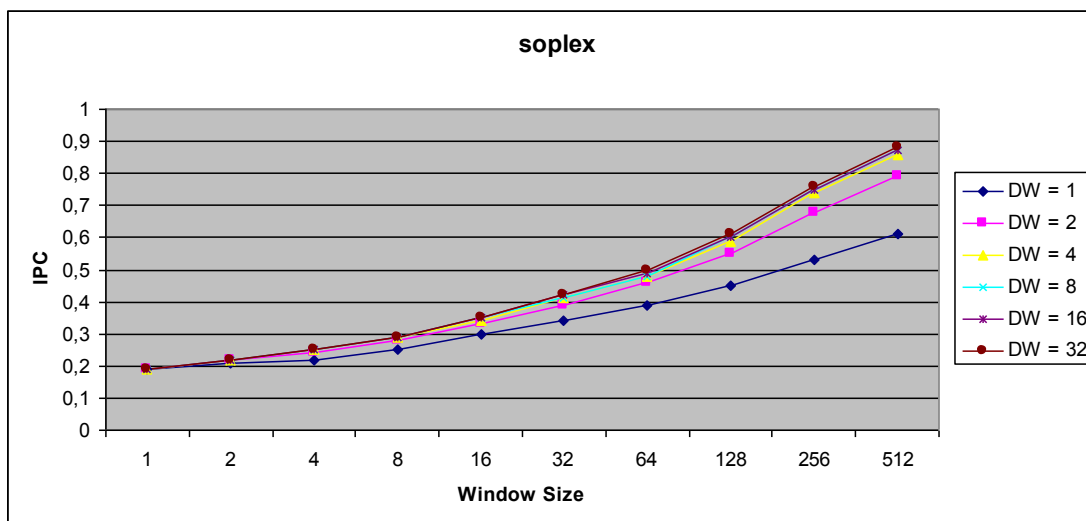
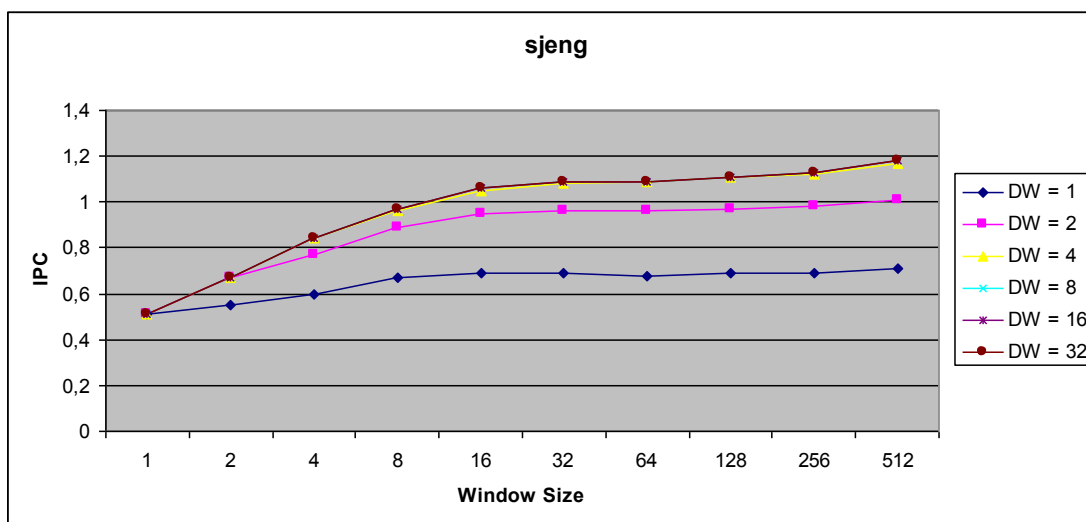
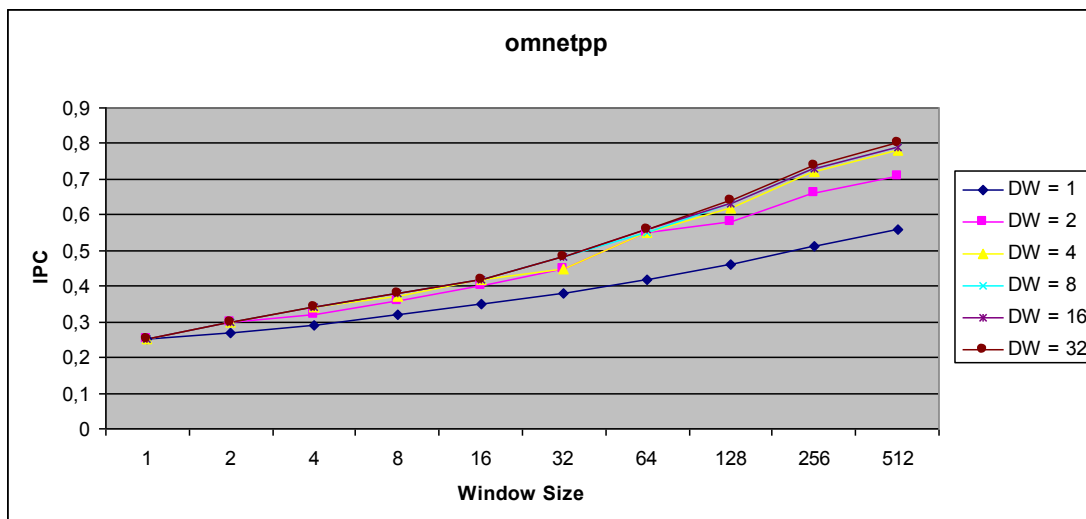


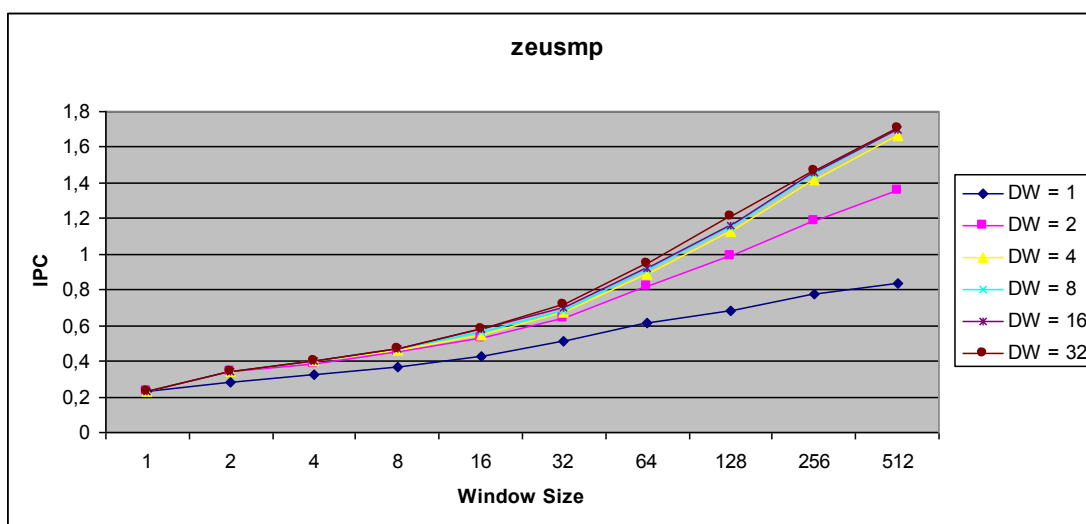
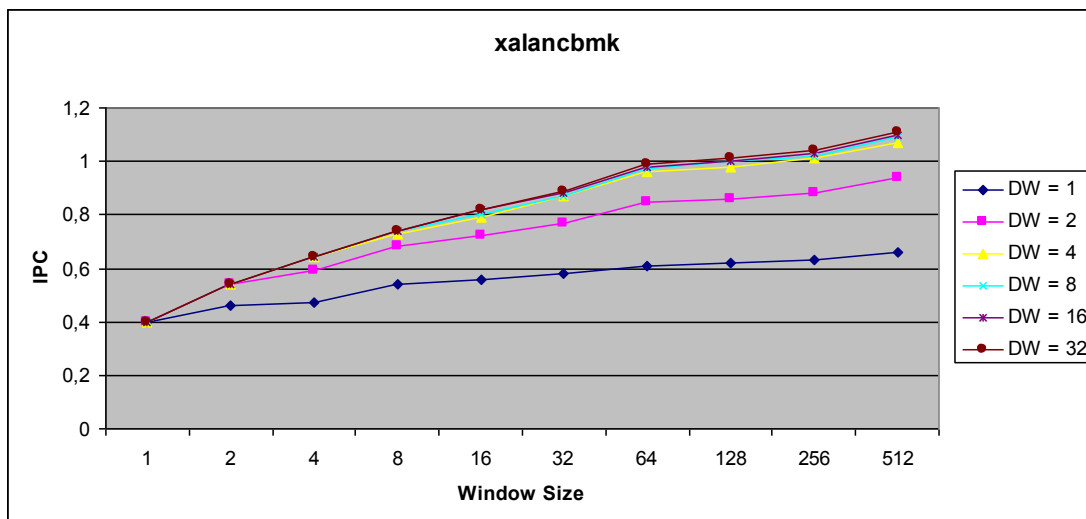
Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις που προέκυψαν από τις μετρήσεις για το IPC σε όλους τους συνδυασμούς window size και dispatch width που αναφέρονται στην εκφώνηση:











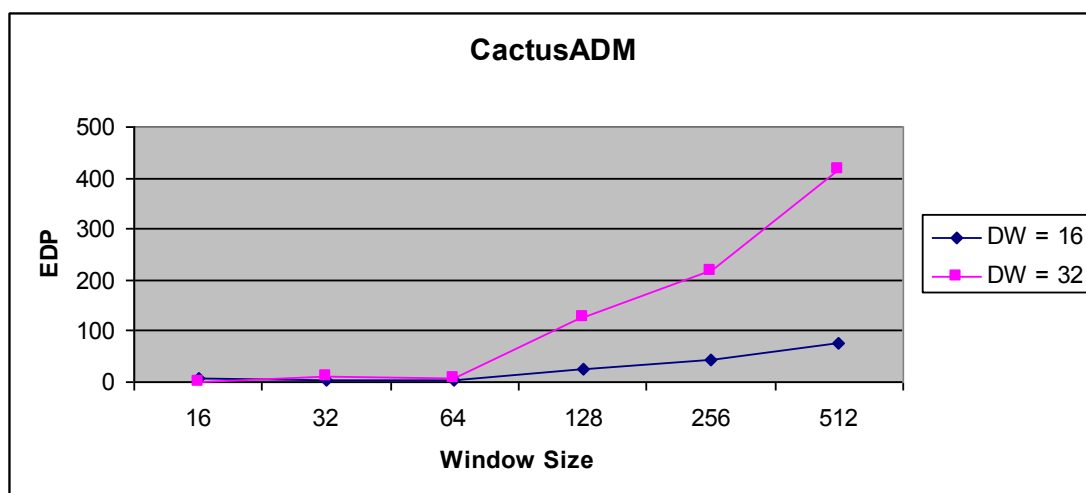
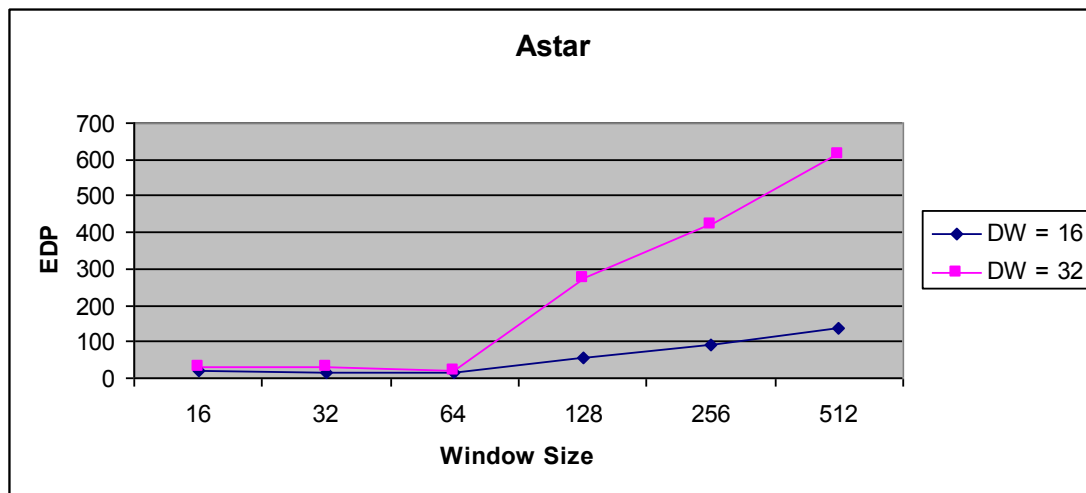
ii)

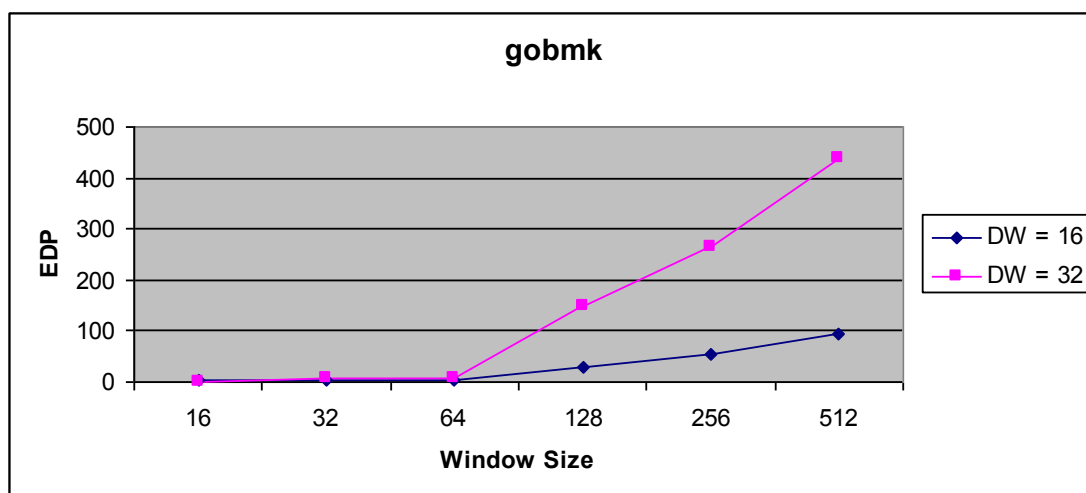
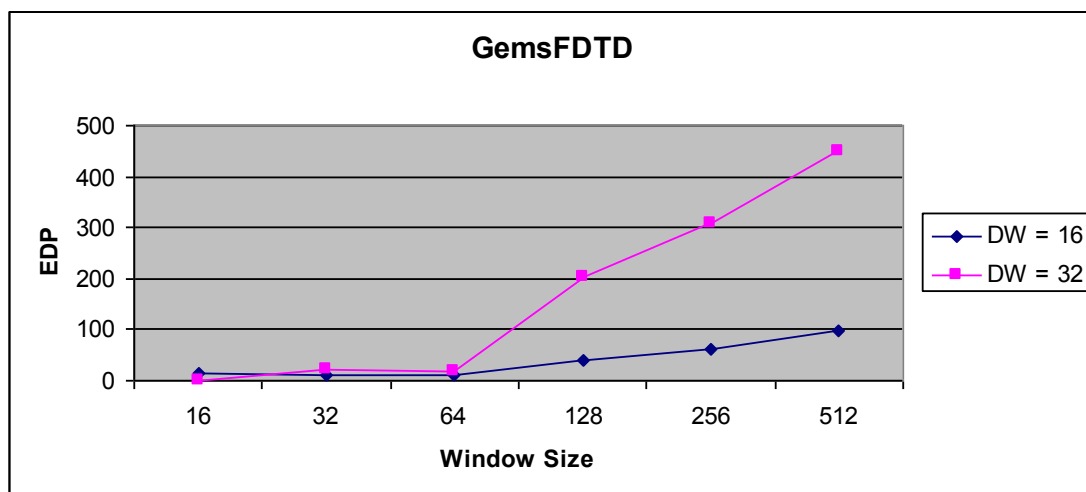
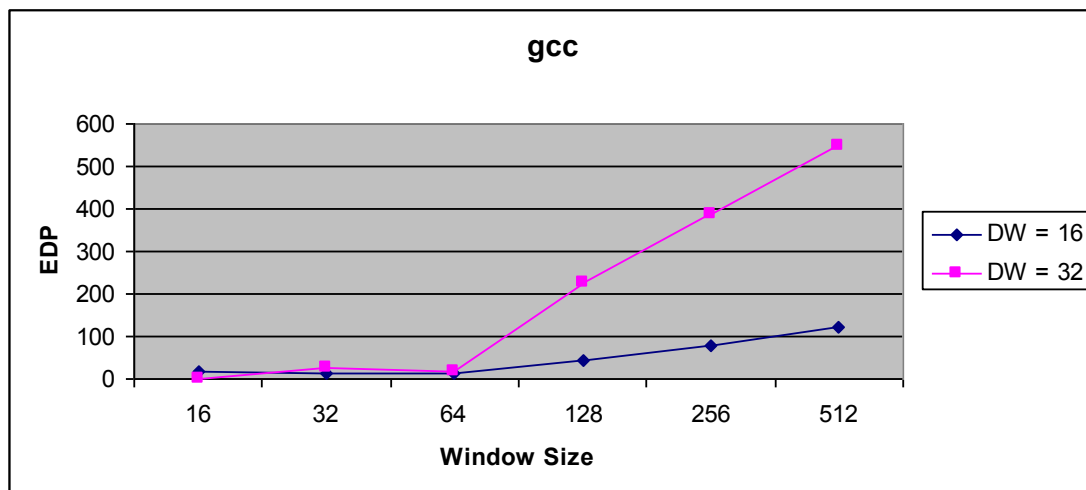
Παρατηρούμε πως όλες οι ανωτέρω συναρτήσεις $IPC(window_size)$ είναι αύξουσες ή γνησίως αύξουσες, με μοναδική εξαίρεση τις συναρτήσεις του CactusADM όπου παρατηρούμε μια φθίνουσα πορεία στο τελευταίο κομμάτι τους ($256 \rightarrow 512$).

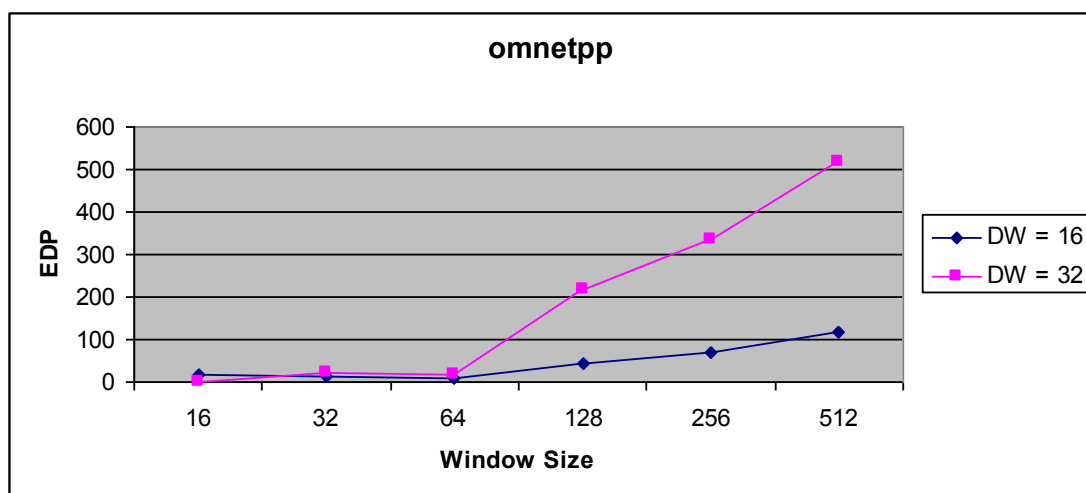
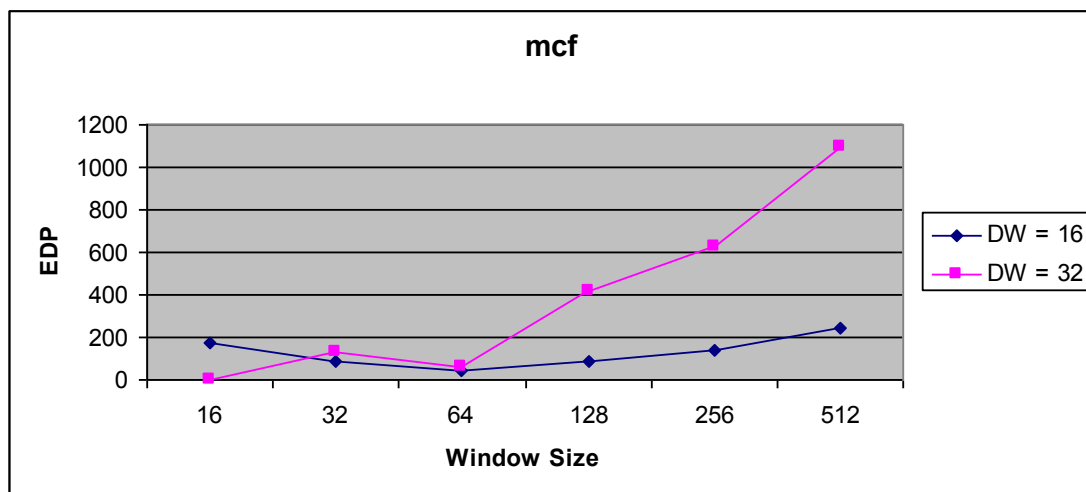
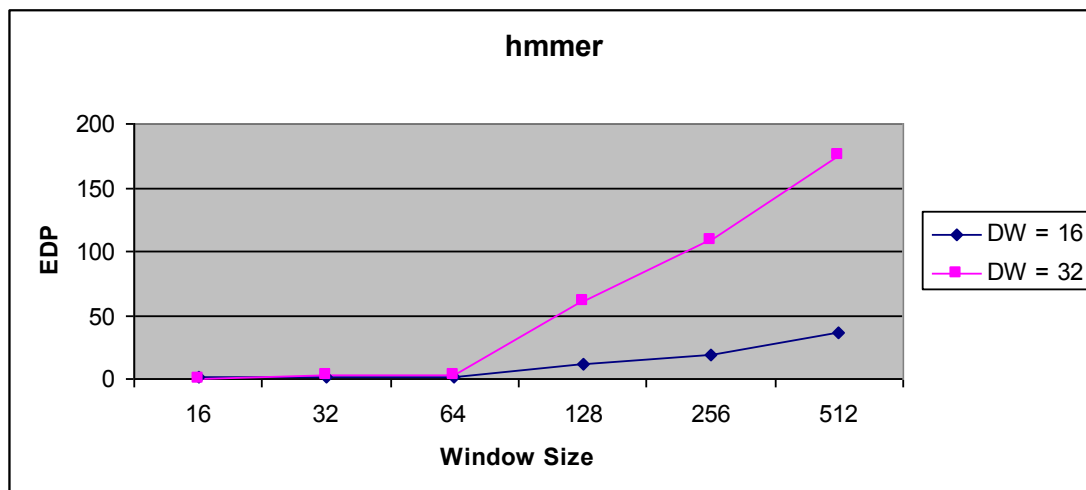
Μπορούμε να συμπεράνουμε λοιπόν πως η αύξηση του window size οδηγεί σε καλύτερη απόδοση (σε ότι αφορά το IPC τουλάχιστον). Επίσης παρατηρούμε ότι στις περισσότερες περιπτώσεις έχουμε μια σημαντική αύξηση της κλίσης της συνάρτησης στην περιοχή του window size από 16 μέχρι 64 ή και μεγαλύτερο. Για μικρές μεταβολές του window size, όταν κινούμαστε σε τιμές του κάτω του 16 δεν έχουμε σημαντική βελτίωση, όμως από αυτό το σημείο και μετά έχουμε εντυπωσιακή σε ορισμένες περιπτώσεις βελτίωση του IPC. Σε ορισμένα benchmarks φυσικά αυτή η αύξηση της κλίσης συμβαίνει λίγο νωρίτερα (π.χ. sjeng \rightarrow window size = 2) ή και λίγο αργότερα (π.χ. astar \rightarrow window size = 64). Επίσης παρατηρούμε ότι μέσα στο ίδιο διάγραμμα οι συναρτήσεις που αντιστοιχούν σε dispatch width μεγαλύτερο, βρίσκονται στις περισσότερες περιπτώσεις και υψηλότερα, άρα αντιστοιχούν σε μεγαλύτερο IPC, δηλαδή και καλύτερες επιδόσεις. Αυτό όμως παρατηρείται κυρίως για dispatch width από 1 έως 4, για τιμές του dispatch width μεγαλύτερες του 4 τα αποτελέσματα είναι σχεδόν ίδια, σε πολλές περιπτώσεις δε διακρίνεται καν η

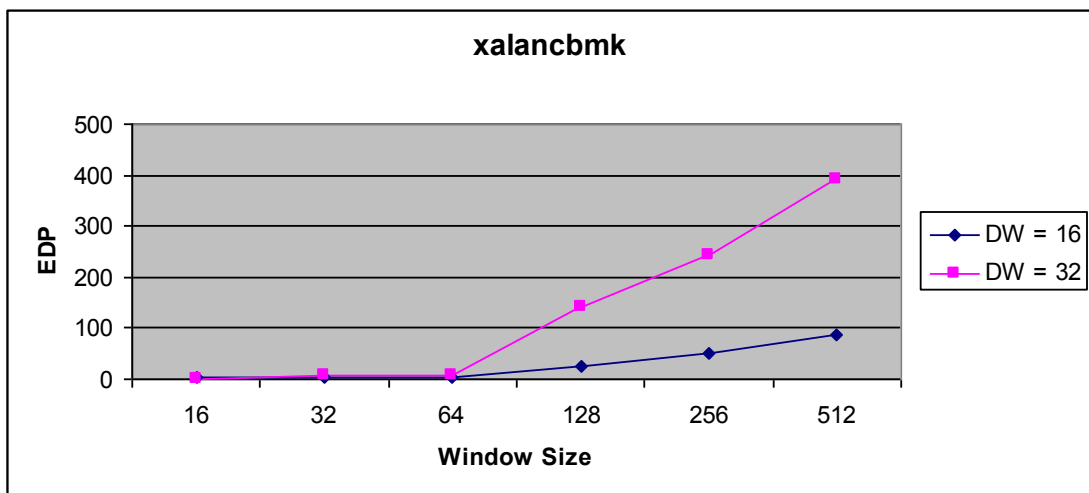
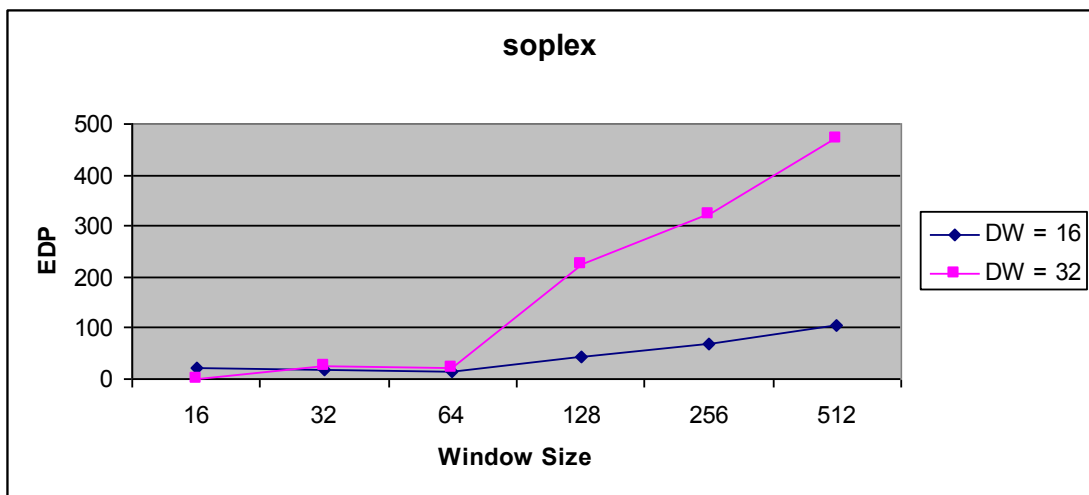
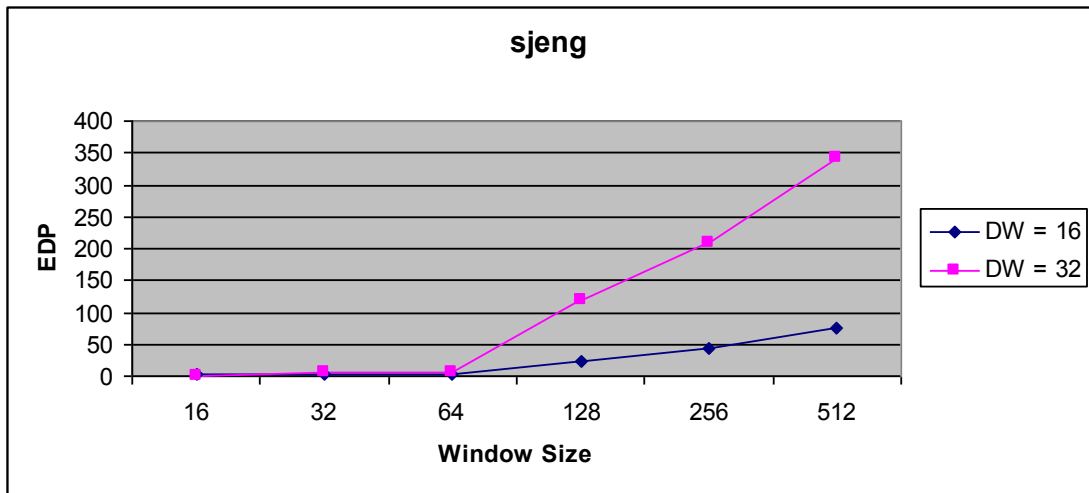
διαφοροποίησή τους στο διάγραμμα. Φτάνουμε λοιπόν στο συμπέρασμα ότι στο σχεδιασμό ενός superscalar, out-of-order επεξεργαστή, τιμή του dispatch width ίση με 4 είναι επαρκής μιας και καλύπτει τα περιθώρια του παραλληλισμού (των συγκεκριμένων τουλάχιστον benchmarks), και όσο καλύτερη επίδοση επιθυμώ, τόσο μεγαλύτερο θα βάλω το window size ίσως με ορισμένα όρια, π.χ. window size ίσο με 256 δίνει στις περισσότερες περιπτώσεις ένα πολύ καλό αποτέλεσμα στις τιμές του IPC σε σύγκριση με τις υπόλοιπες τιμές του window size.

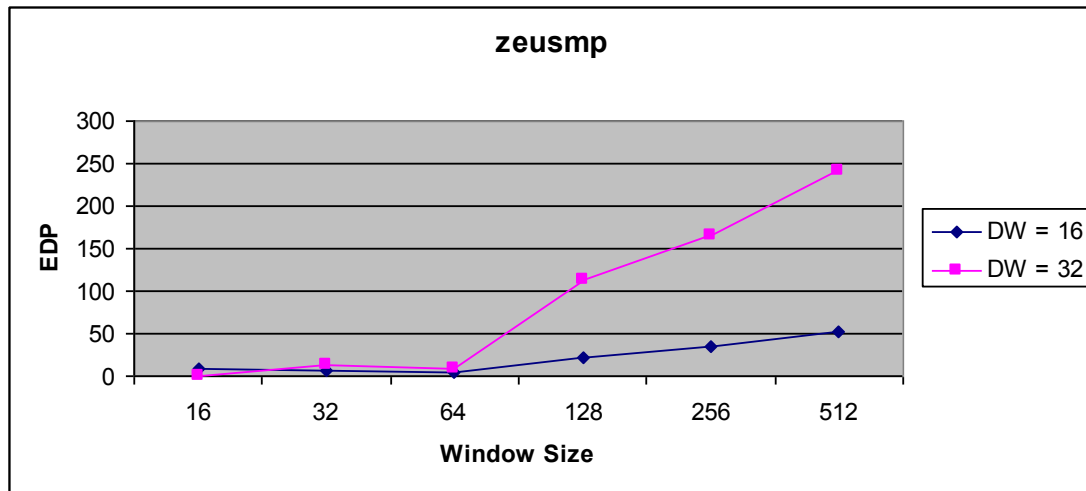
Στη συνέχεια ακολουθούν τα διαγράμματα που προέκυψαν από τις μετρήσεις σχετικά με την κατανάλωση ενέργειας των επεξεργαστών. Συγκεκριμένα έχουμε γραφικές παραστάσεις του EDP σε συνάρτηση με το window size, για τιμές του dispatch width 16 και 32, διότι για μικρότερες τιμές του, κατά την εκτέλεση του script παίρναμε το Error: "ValueError: No valid McPAT output found".







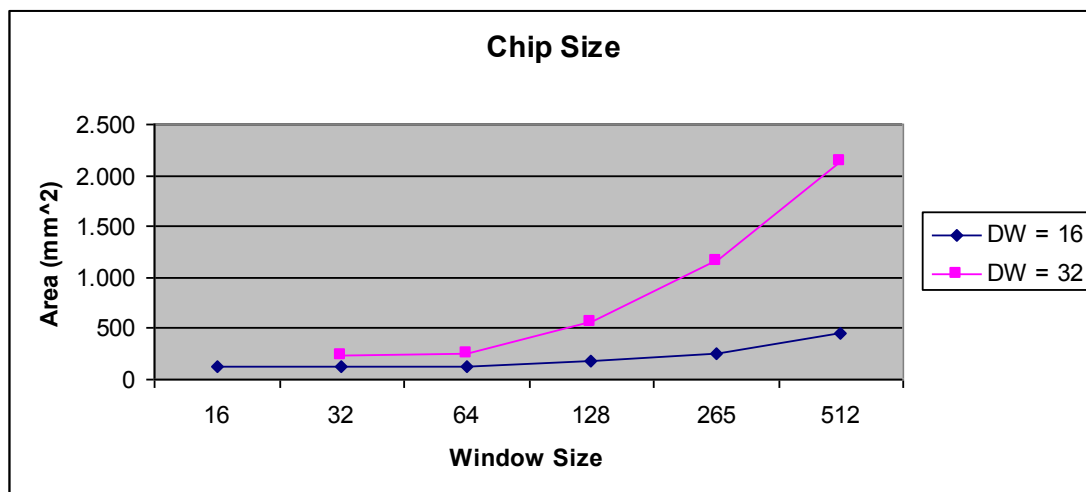




iii)

Στα παραπάνω διαγράμματα παρατηρούμε σχεδόν μηδενική κατανάλωση ενέργειας για τιμές του window size μικρότερες ή ίσες του 64, ανεξάρτητα από την τιμή του dispatch width. Για window size από 128 έως και 512 παρατηρούμε μια αύξηση στην κατανάλωση ενέργειας η οποία σε ορισμένες περιπτώσεις είναι σημαντική ειδικά για dispatch width ίσο με 32. Μεταξύ των 2 τιμών του dispatch width βλέπουμε πολύ υψηλότερη κατανάλωση για dispatch width ίσο με 32 που σε ορισμένες περιπτώσεις φτάνει και την πενταπλάσια κατανάλωση σε σύγκριση με την αντίστοιχη τιμή (ίδιο window size) για dispatch width ίσο με 16. Εάν λάβουμε υπόψη και τις μετρήσεις του IPC που παρουσιάστηκαν προηγουμένως, σχεδιαστικά πολύ καλύτερη επιλογή φαίνεται το dispatch width ίσο με 16 σε σύγκριση με το dispatch width = 32 μιας και μας δίνει σχεδόν την ίδια επίδοση (σε IPC) με πολύ μικρότερη κατανάλωση ενέργειας.

Στη συνέχεια παρατίθεται ένα διάγραμμα που δείχνει πως επηρεάζουν το dispatch width και το window size, το μέγεθος του chip του επεξεργαστή.
(Προφανώς οι τιμές της περιοχής του chip είναι ίδιες ανεξάρτητα από το benchmark μιας και πρόκειται για περιοχή υλικού που εξαρτάται από τα χαρακτηριστικά του επεξεργαστή κι όχι από το πρόγραμμα που τρέχει.)



Παρατηρούμε λοιπόν πως όσο μεγαλύτερο είναι το window size καθώς και όσο μεγαλύτερο είναι το dispatch width, τόσο μεγαλώνει και το μέγεθος του chip. Σημαντική διαφορά στο μέγεθος σημειώνεται στην αύξηση του window size από 128 και πάνω, για dispatch width ίσο με 32.

iv) Στους επεξεργαστές SkyLake οι μηχανικοί της Intel επέλεξαν window size ίσο με 224 και dispatch width ίσο με 6. Παρατηρούμε αύξηση και των 2 παραμέτρων σε σύγκριση με έναν από τους προκατόχους του SkyLake, τους επεξεργαστές Sandy Bridge που είχαν Window Size = 168 και dispatch width = 4. Στους SkyLake λοιπόν έχουμε ROB που μπορεί να φιλοξενεί μέχρι 224 μικροεντολές, και scheduler που διαθέτει 8 θύρες που μπορεί να προωθεί μέχρι 6 εντολές ανά κύκλο ρολογιού . Οι τιμές που επέλεξαν οι αρχιτέκτονες συμβαδίζουν με τα όσα είδαμε από τις προσομοιώσεις μας. Συγκεκριμένα επέλεξαν dispatch width μεγαλύτερο ή ίσο του 4 όπως αναφέραμε από τις προσομοιώσεις και συγκεκριμένα λίγο μεγαλύτερο του 4 το οποίο όπως είδαμε μας έδινε λίγο καλύτερη επίδοση χωρίς να είναι αρκετά μεγάλο ώστε να επηρεάζει σημαντικά την κατανάλωση ενέργειας. Επίσης επέλεξαν window size ίσο με 224 το οποίο δίνει πολύ καλές επιδόσεις (κοντά στο 256 που μελετήσαμε εμείς και είδαμε πως μας δίνει πολύ καλά αποτελέσματα σε σύγκριση με τις άλλες τιμές). Μεγαλύτερο window size ίσως έδινε λίγο καλύτερη επίδοση αλλά πιθανώς να δημιουργούσε σημαντικά προβλήματα με την κατανάλωση ενέργειας και την ανάγκη ψύξης του επεξεργαστή.