

# ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

## 3<sup>η</sup> ΑΣΚΗΣΗ

Ακαδημαϊκό Έτος 2019-2020

Εξάμηνο 8<sup>ο</sup>

Χρόνης Σάκος

03116168

### Θεωρητική ανάλυση

Γνωρίζουμε ότι τα βασικά χαρακτηριστικά ενός superscalar επεξεργαστή είναι το dispatch width και το window size.

- Το dispatch width δηλώνει τον αριθμό των εντολών μηχανής που μπορούν να αποσταλούν για εκτέλεση σε κάθε κύκλο μηχανής. Έτσι, όσο μεγαλύτερο είναι το dispatch width, τόσο μεγαλύτερη αναμένεται να είναι η απόδοση. Ωστόσο, είναι λογικό πως από έναν αριθμό και πάνω, η αύξηση του dispatch width, αναμένεται να μην επιδρά στην βελτίωση της απόδοσης, αφού δεν θα υπάρχουν άλλες εντολές να αποσταλούν για εκτέλεση.
- Το windows size δηλώνει τον αριθμό των εντολών μηχανής που βρίσκονται σε αναμονή, από τις οποίες αυτές που είναι έτοιμες επιλέγονται για έκδοση. Επομένως, όσες περισσότερες εντολές έχουμε σε αναμονή, τόσο πιο αποδοτικός ο επεξεργαστής. Ωστόσο, γίνεται αντιληπτό, πως όταν το window size γίνει αρκετά μεγάλο, καθυστερεί η εύρεση των εντολών με αποτέλεσμα να μειώνεται η αποδοτικότητα.

Γενικά, λοιπόν, ένας superscalar, out-of-order επεξεργαστής χρειάζεται αφενός μεγάλο dispatch width και αφετέρου μεγάλο windows size. Ωστόσο, όπως επισημάνθηκε παραπάνω, πολύ μεγάλο windows size μπορεί να καθυστερεί τις διεργασίες, μειώνοντας την απόδοση, ενώ επίσης και μεγαλύτερο dispatch width μπορεί να μην είναι αναγκαίο.

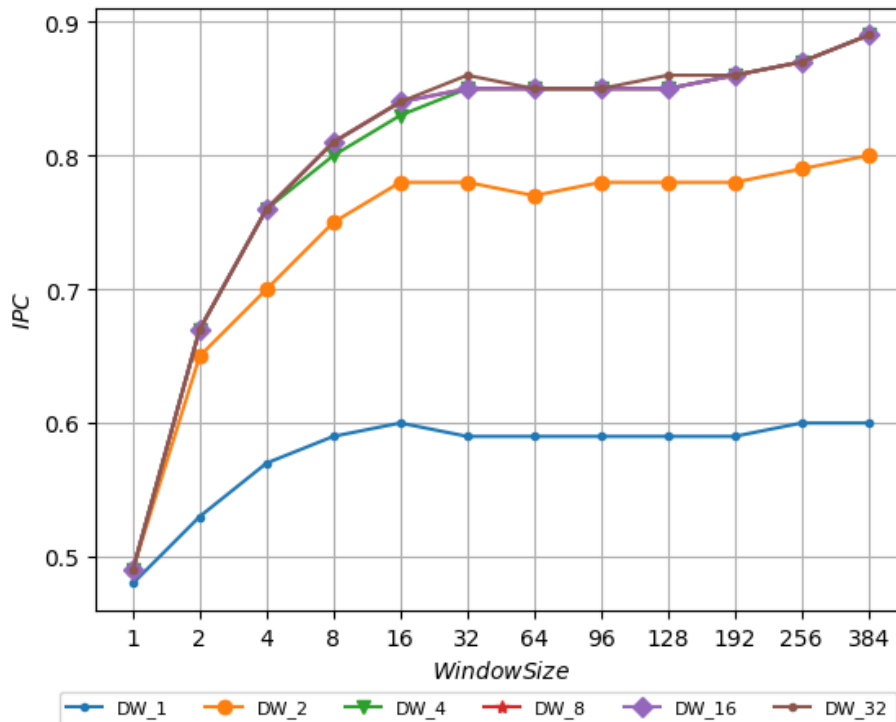
### Πειραματική αξιολόγηση

Ακολούθως, θα παρουσιάσουμε τα πειραματικά αποτελέσματα, που προέκυψαν από την εκτέλεση όλων των benchmarks για κάθε διαφορετικό επεξεργαστή που προκύπτει από το συνδυασμό των παρακάτω τιμών για τις παραμέτρους dispatch\_width και window\_size:

dispatch_width	1	2	4	8	16	32						
window_size	1	2	4	8	16	32	64	96	128	192	256	384

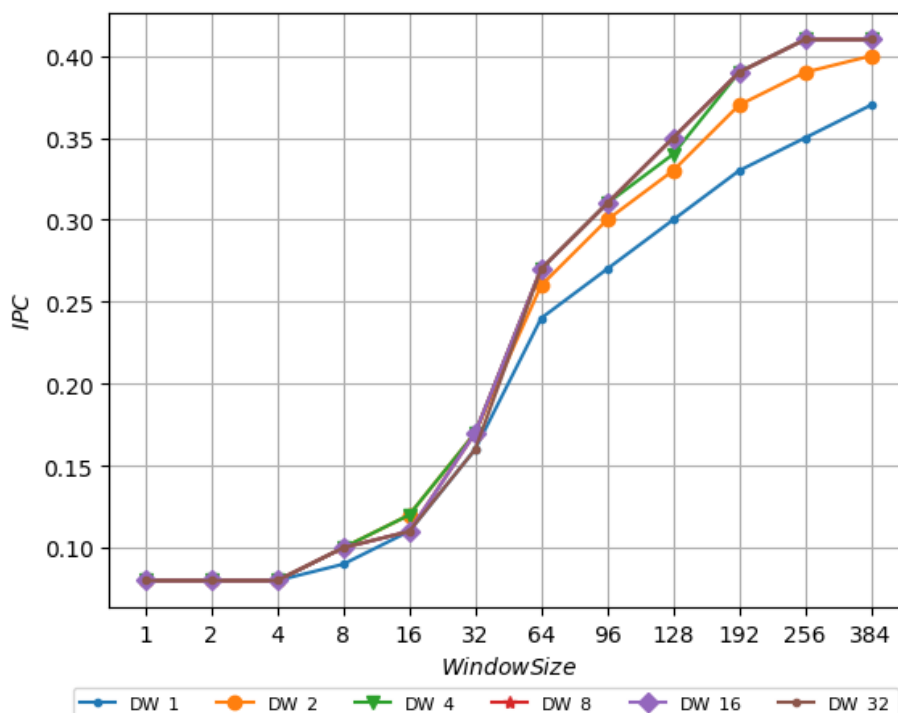
## gcc

Στο benchmark gcc βέλτιστη απόδοση έχουμε για window size = 384 και για τα διάφορα dispatch width από 4 έως 32. Συγκεκριμένα, παρατηρούμε ότι για μικρές τιμές του window size, η απόδοση αυξάνεται ραγδαία, ενώ από την τιμή 32 έως την τιμή 384 οι διαφορές είναι πολύ μικρές. Όσον αφορά το dispatch width, η αύξηση του συμβάλλει στη βελτίωση της απόδοσης, μέχρι να λάβει την τιμή 4.



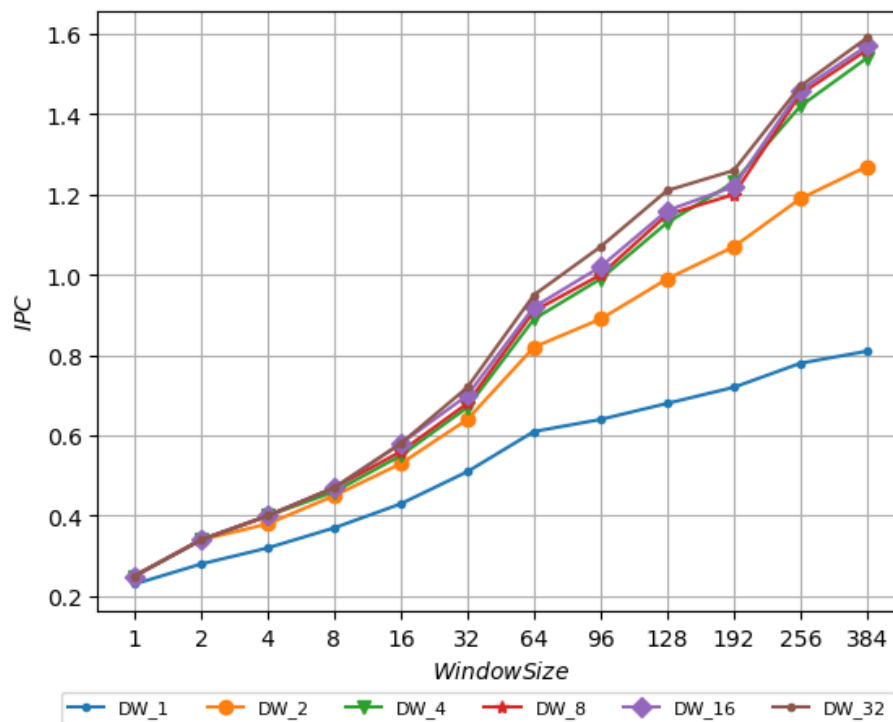
## mcf

Στο mcf βέλτιστη απόδοση έχουμε για τους συνδυασμούς window size = (256,384) και dispatch width = (4,8,16,32). Σε αυτό το benchmark εντοπίζουμε καλύτερα τη συμβολή του windows size, καθώς η απόδοση αυξάνεται ραγδαία όσο μεγαλώνει αυτό. Ενδιαφέρον στοιχείο είναι ότι κατά τη μετάβαση του windows size από 1 σε 2 και σε 4 δεν επηρεάζεται καθόλου η απόδοση.



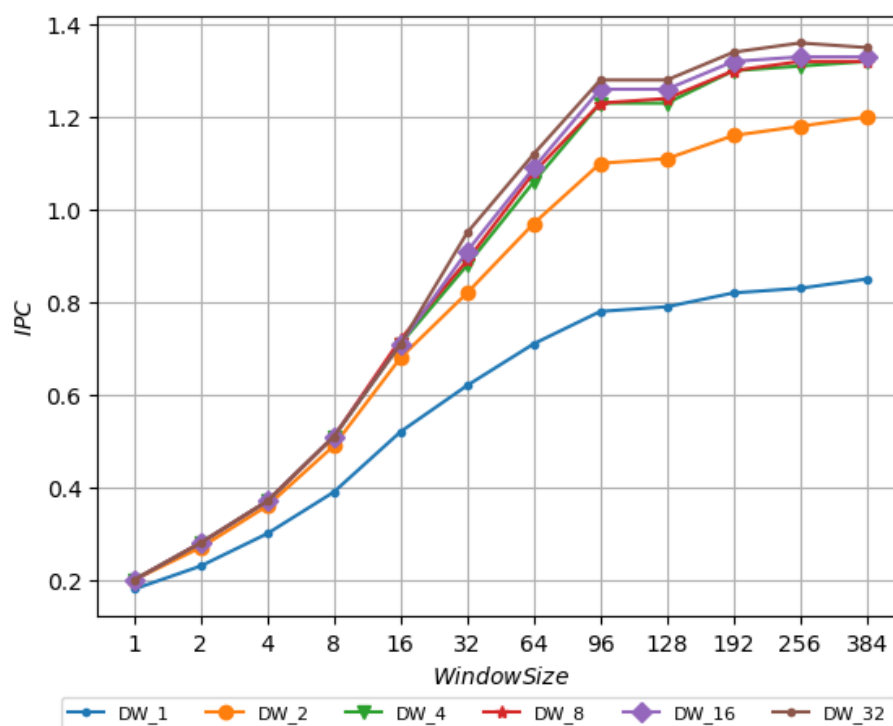
## zeusmp

Στο συγκεκριμένο benchmark έχουμε ως βέλτιστο συνδυασμό τον (32,384). Αφενός, παρατηρούμε ότι η αύξηση του windows size παρουσιάζει συνολικά 7 φορές μεγαλύτερη απόδοση. Αφετέρου, για dispatch width μεγαλύτερο του 4 παρατηρούμε ελάχιστη αύξηση της απόδοσης.



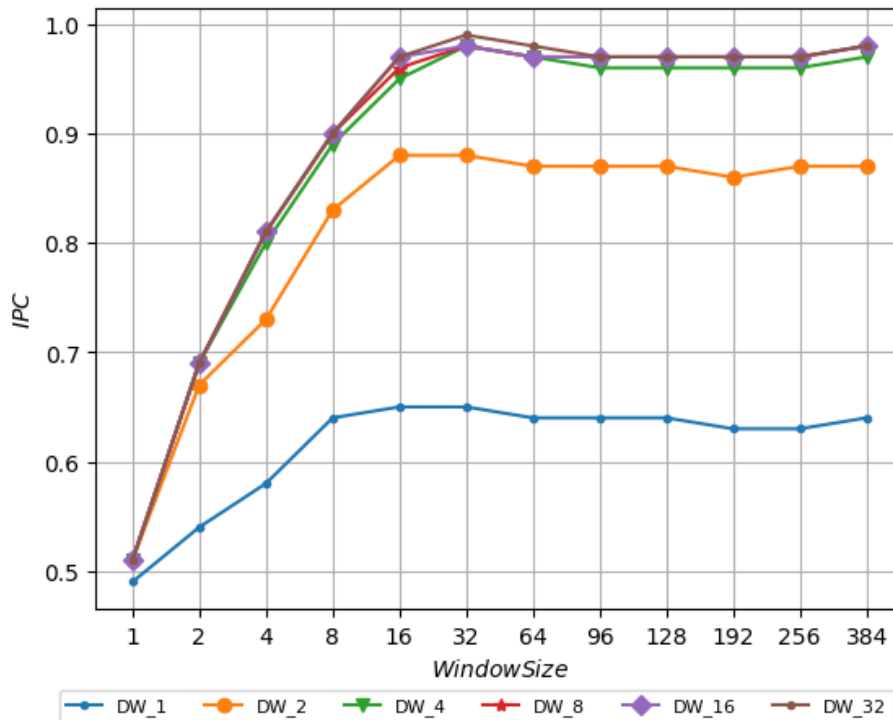
## cactusADM

Και σε αυτό το benchmark καταλήγουμε στα ίδια συμπεράσματα όπως στο zeusmp. Η μοναδική διαφορά είναι πως για windows size = 96 και πάνω, η απόδοση φαίνεται να σταθεροποιείται.



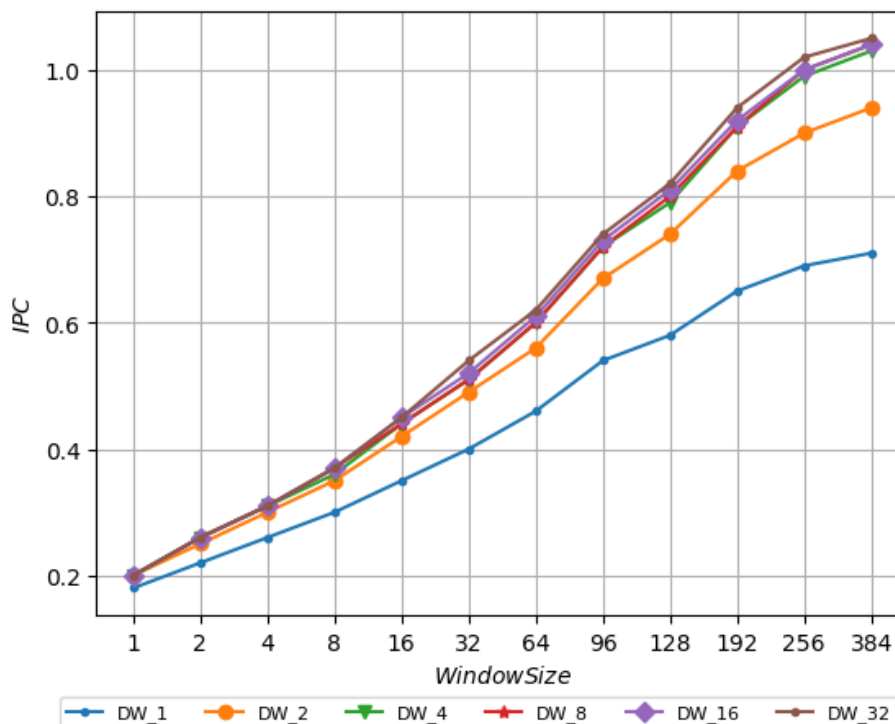
## gobmk

Στο gobmk βέλτιστη απόδοση έχουμε για τον συνδυασμό (32,32). Όπως και στα προηγούμενα benchmarks, για τιμή του dispatch width μεγαλύτερη του 8, εντοπίζουμε μηδαμινή αύξηση της απόδοσης. Επίσης, παρατηρούμε πως όταν το window size λάβει την τιμή 16 και έπειτα, τότε η απόδοση παρουσιάζει μικρές αυξομειώσεις γύρω από τη βέλτιστη τιμή.



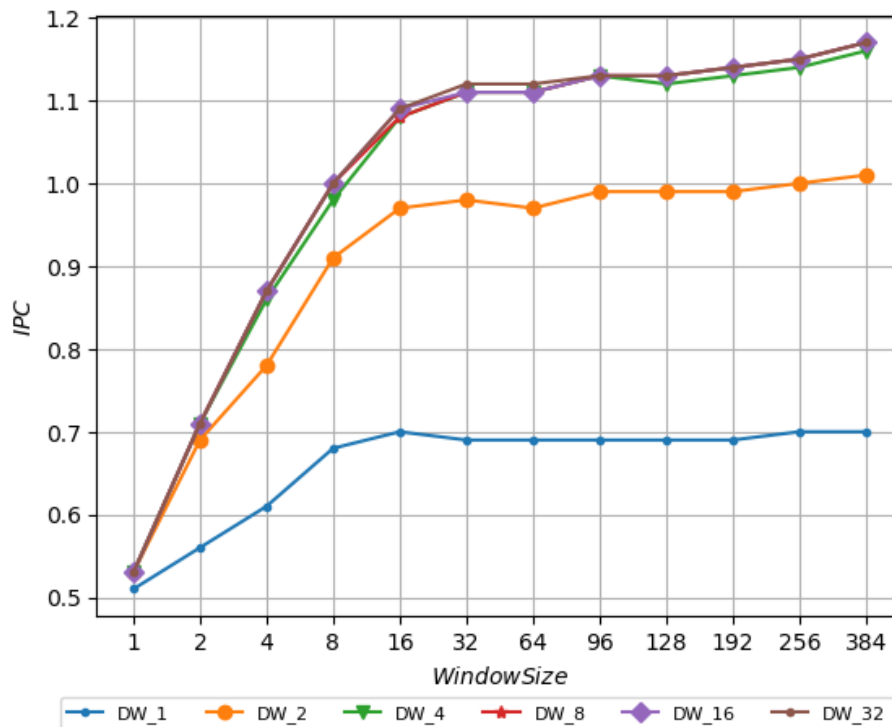
## GemsFDTD

Στο συγκεκριμένο benchmark έχουμε ως βέλτιστο συνδυασμό τον (32,384). Τα συμπεράσματα στα οποία καταλήγουμε είναι τα ίδια με εκείνα του zeusmp.



## sjeng

Και σε αυτό το benchmark βέλτιστη απόδοση έχουμε για (32,384), ενώ και το αποτέλεσμα της προσομοίωσης είναι παρόμοιο με προγενέστερα benchmark.



Σε αυτό το σημείο να τονίσουμε πως για την μελέτη της απόδοσης των σύγχρονων superscalar επεξεργαστών, αναλύσαμε και παρουσιάσαμε τα αποτελέσματα μόνο από 7 εκ των 12 benchmark που εκτελέσαμε, καθώς μόνο για εκείνα καταφέραμε να προσομοιώσουμε ικανό αριθμό από εντολές (>10% 1billion instructions).

### Συμπεράσματα

- Σε όλες τις περιπτώσεις που αναλύσαμε παραπάνω, παρατηρήσαμε πως αύξηση του window size προκαλεί βελτίωση της απόδοσης του συστήματος. Στα περισσότερα benchmark, η βελτίωση ήταν πιο γρήγορη για μικρές τιμές του window size, ενώ στις μεγαλύτερες, δηλαδή για τιμή από 32 και έπειτα, η απόδοση σταθεροποιούνταν. Ωστόσο, σχεδόν σε όλες τις περιπτώσεις, βέλτιστη απόδοση λαμβάναμε για window size = 384.
- Βάσει των αποτελεσμάτων των προσομοιώσεων, καταλήξαμε στο συμπέρασμα πως η αύξηση του dispatch width προκαλεί σημαντική βελτίωση της απόδοσης. Ωστόσο, παρατηρήσαμε ότι σε όλα τα benchmark, περαιτέρω αύξηση του dispatch width προκαλούσε είτε μηδαμινή είτε καμία άλλη βελτίωση της απόδοσης του συστήματος. Είδαμε, δηλαδή, ότι ήδη και για dispatch width = 4, σε πολλές περιπτώσεις, είχαμε βέλτιστη απόδοση.

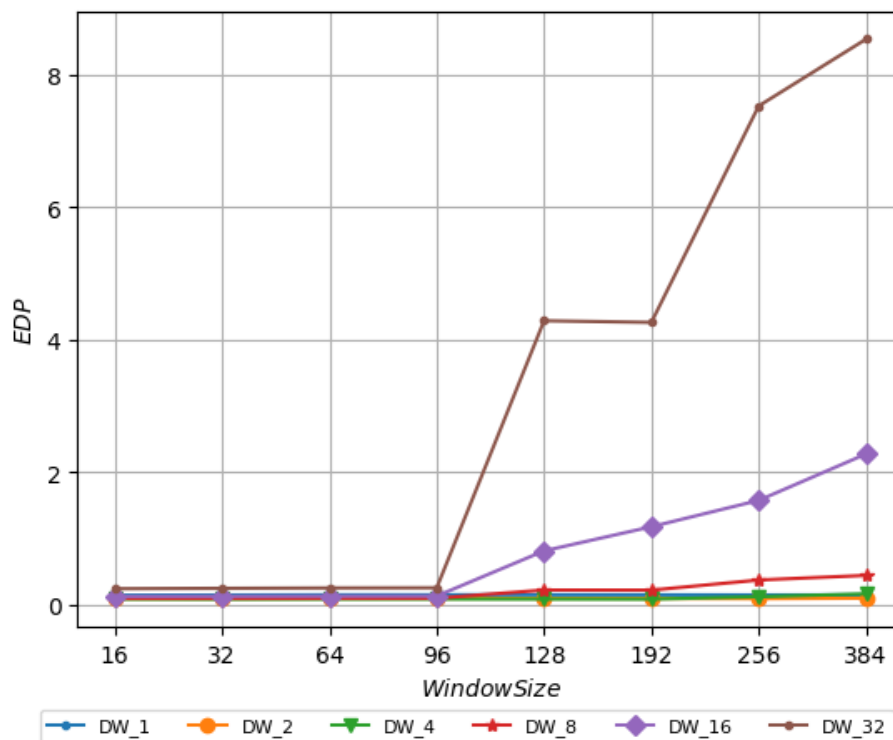
Κατά συνέπεια όσων παρατηρήσαμε από την προσομοίωση των benchmarks αλλά και όσων αναφέραμε στη θεωρητική ανάλυση, δεν έχει νόημα να εξετάσουμε όλες τις διαφορετικές τιμές του dispatch width. Στην πραγματικότητα, θα αρκούσε να περιοριστούμε στην ανάλυση έως τιμή width = 8, καθώς σε όλες τις περιπτώσεις η απόκλιση των αποτελεσμάτων για width = 8, 16, 32 ήταν μηδαμινή. Συνεπώς, θα μπορούσαμε να εξετάσουμε μόνο 48 από τους 72 συνδυασμούς και να λάβουμε τα ίδια αποτελέσματα.

## Κατανάλωση ενέργειας

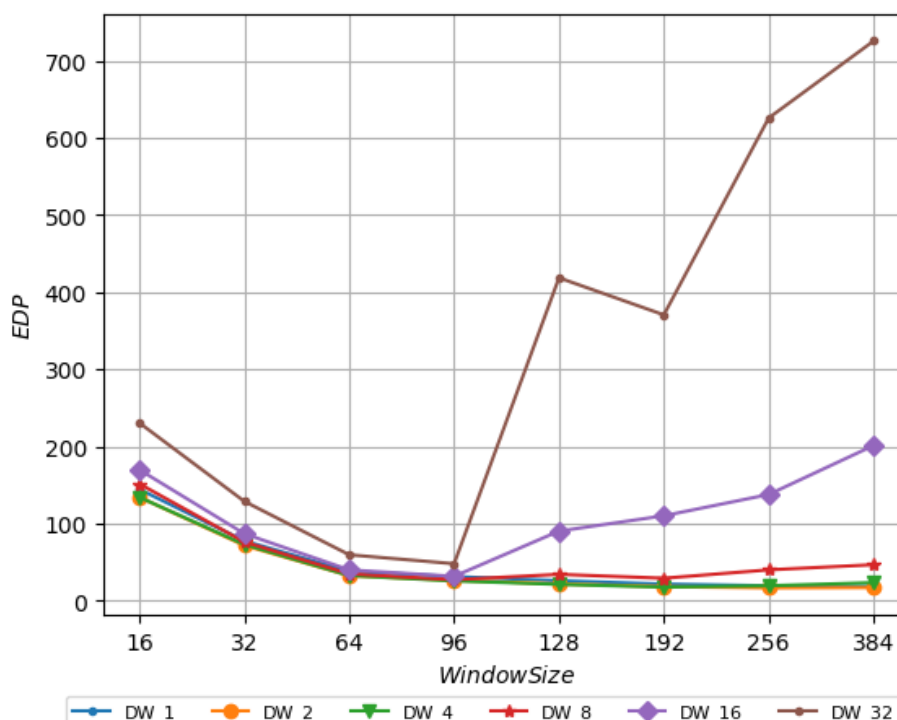
Παραδοσιακά, για την αξιολόγηση της κατανάλωσης ενός επεξεργαστή χρησιμοποιείται ως μετρική η συνολική κατανάλωση ενέργειας σε Joules. Ωστόσο για να μελετήσουμε την επίδραση διαφόρων χαρακτηριστικών του επεξεργαστή όχι μόνο στην κατανάλωση αλλά ταυτόχρονα και στην επίδοσή του, θα χρησιμοποιήσουμε ως μετρική το energy-delay product (EDP). Το EDP για την εκτέλεση ενός benchmark ορίζεται ως το γινόμενο της ενέργειας επί τον χρόνο εκτέλεσης του benchmark:

$$EDP = Energy(J) * runtime(sec)$$

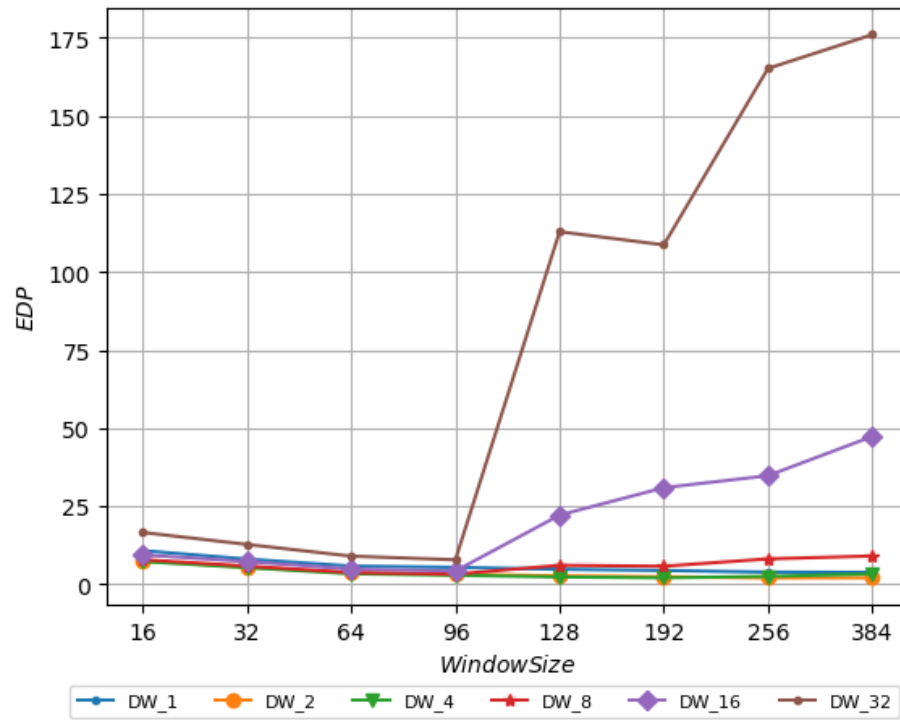
### gcc



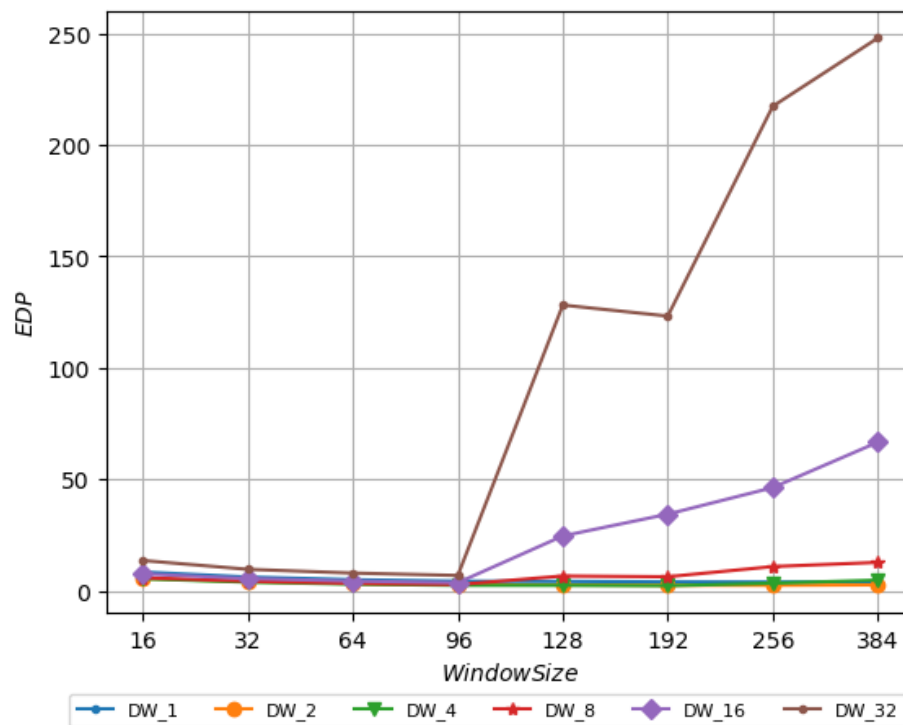
### mcf



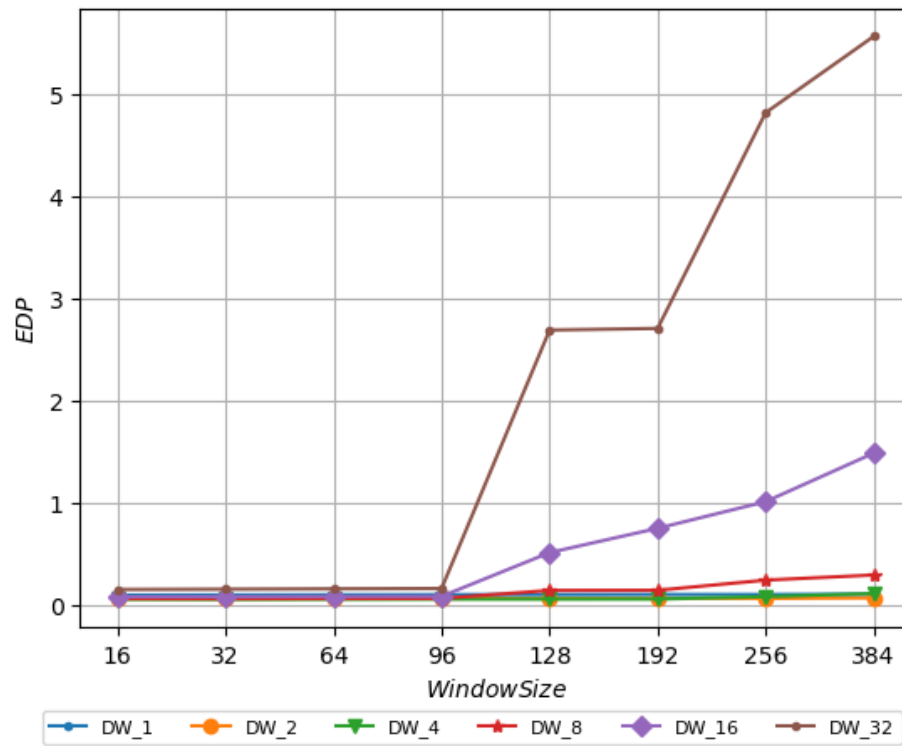
## zeusmp



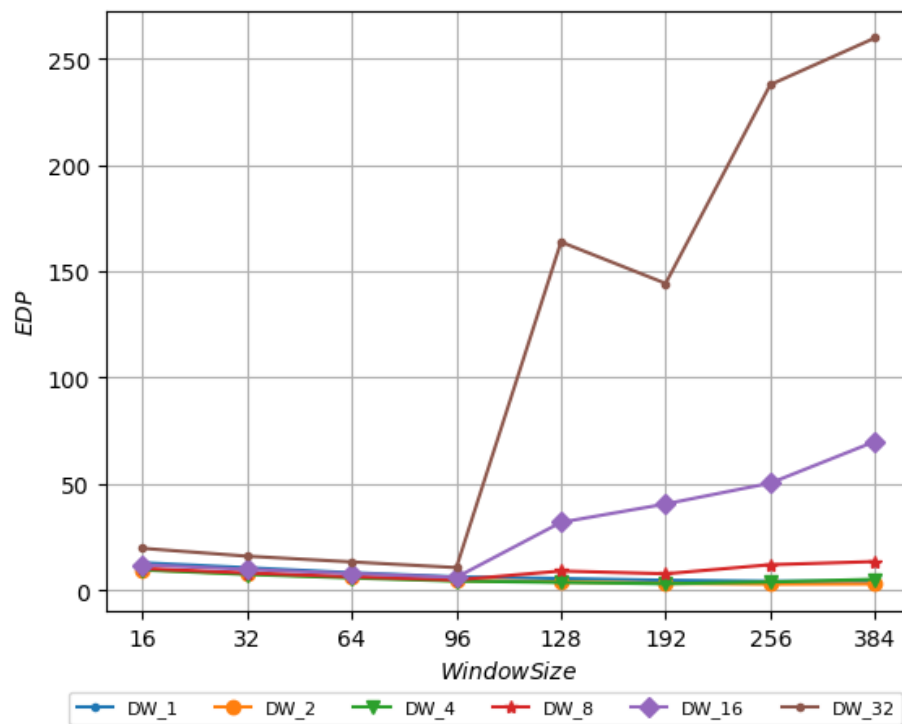
## cactusADM



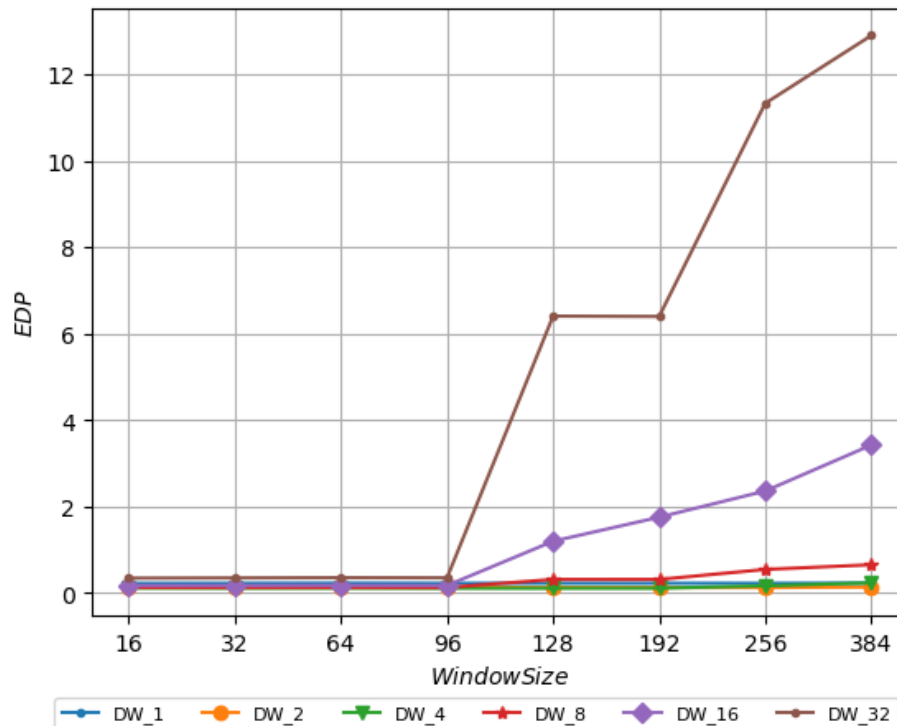
## gobmk



## GemsFDTD







Σε όλες τις παραπάνω γραφικές παραστάσεις, παρατηρούμε πως ο συνδυασμός του dispatch width και του windows size, επηρεάζουν σημαντικά το EDP, δηλαδή το γινόμενο της κατανάλωσης ενέργειας με τον χρόνο εκτέλεσης.

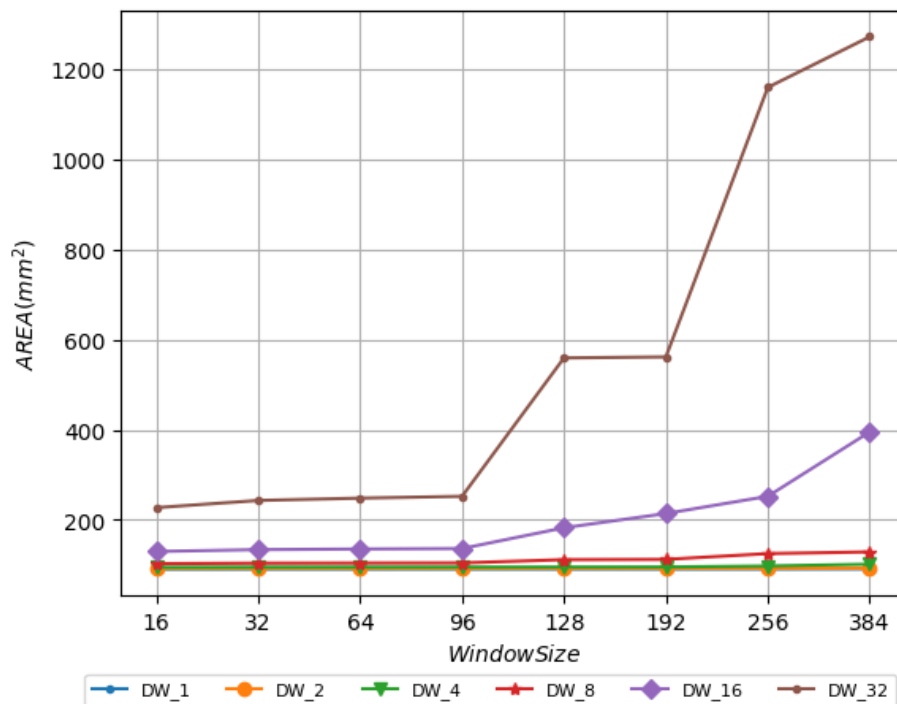
Συγκεκριμένα, παρατηρούμε πως μικρότερη τιμή EDP έχουμε για dispatch width = 2, ενώ μικρή απόκλιση έχουμε για width = 1 και 4. Επιπλέον, παρατηρούμε πως για window size μικρότερο ή ίσο του 96, κάθε dispatch width επηρεάζει σχεδόν το ίδιο την καταναλισκόμενη ενέργεια, με μικρότερες τιμές του width να είναι πιο οικονομικές.

Ωστόσο, από εκεί και έπειτα, όσο αυξάνεται το windows size, “εκτοξεύεται” η τιμή του EDP, για τις τιμές του dispatch width ίσο με 16 και 32. Αυτό είναι λογικό δεδομένου ότι όταν αποστέλλεται μεγάλος αριθμός εντολών για εκτέλεση σε κάθε κύκλο και βρίσκονται περισσότερες εντολές σε αναμονή, απαιτείται περισσότερη ενέργεια και χρόνος για να πραγματοποιηθεί η εκτέλεση αυτών.

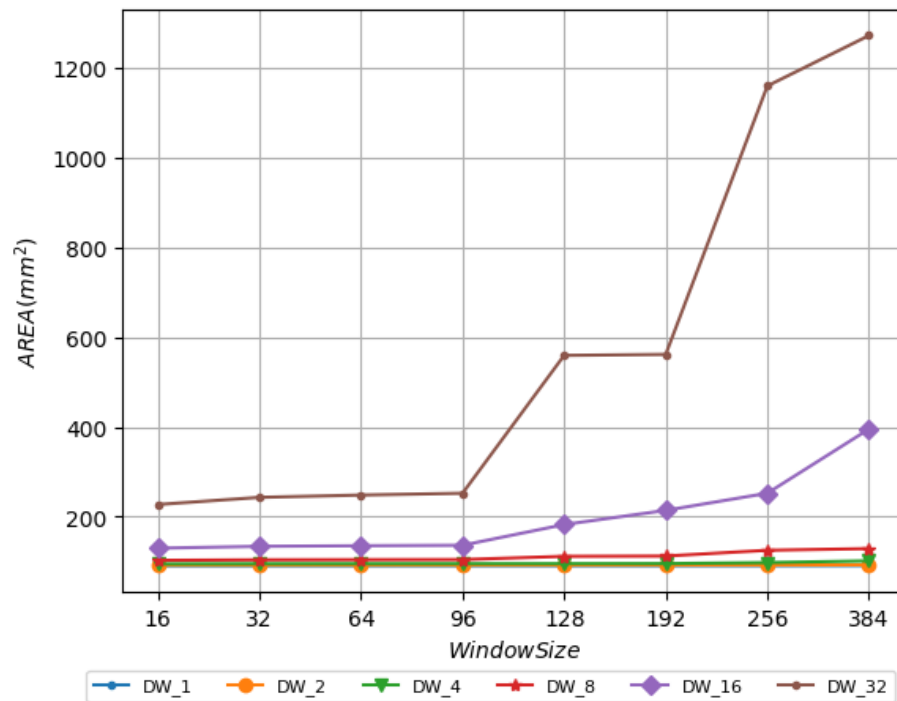
Συνολικά, λοιπόν, ο ιδανικός συνδυασμός των δύο μεταβλητών, προϋποθέτει dispatch width = 2 και windows size οποιαδήποτε από τις τιμές 192, 256, 384.

## Μέγεθος chip

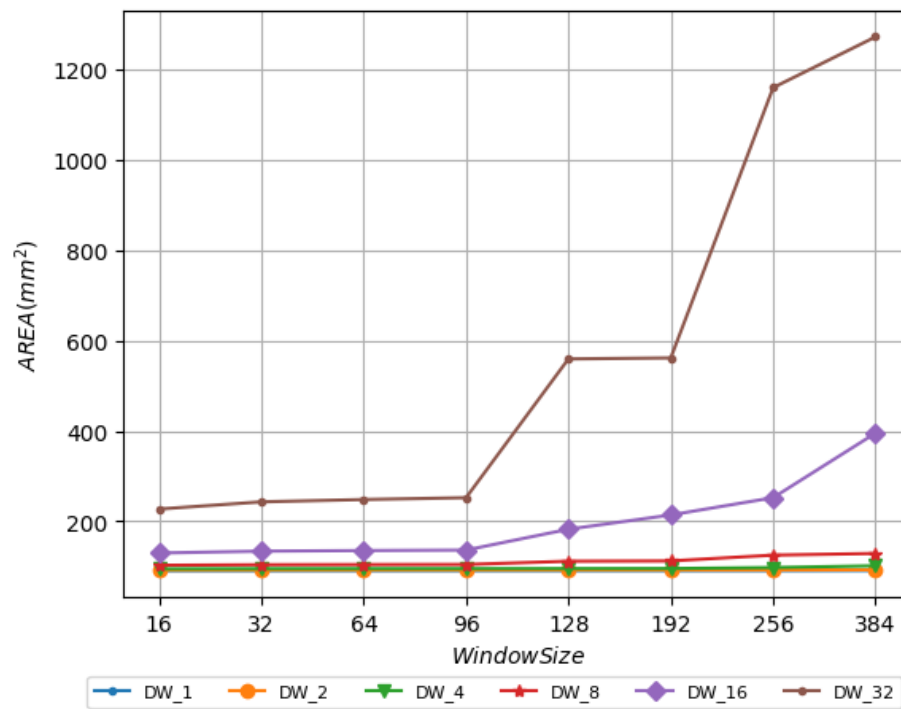
gcc



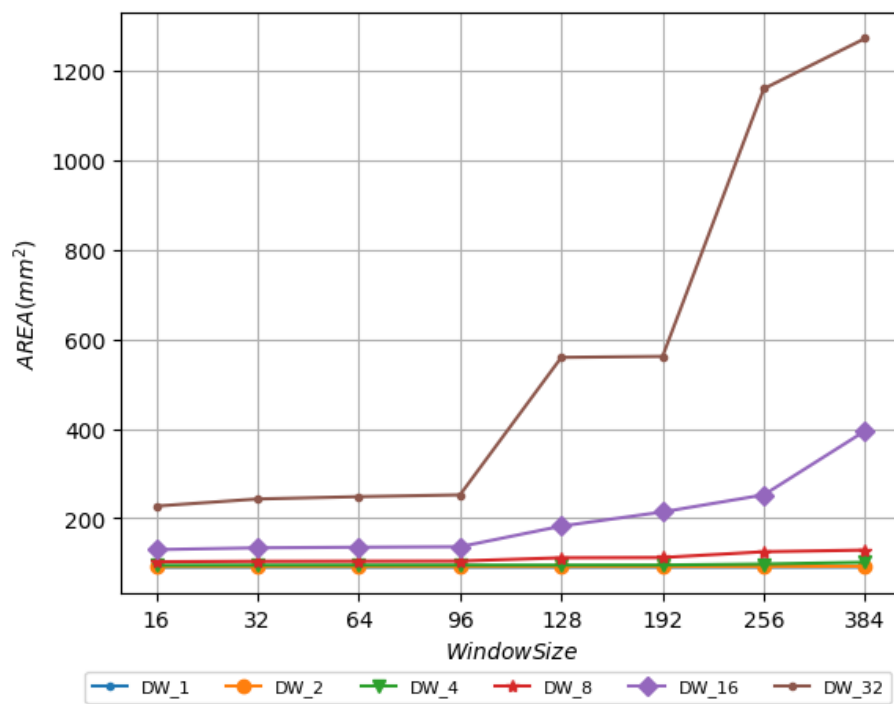
mcf



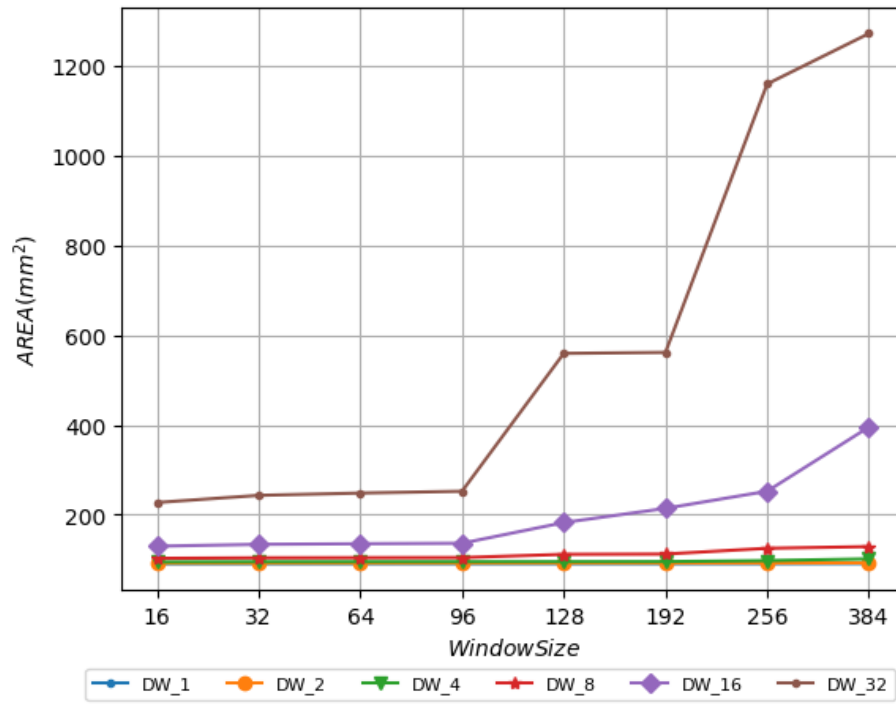
## zeusmp



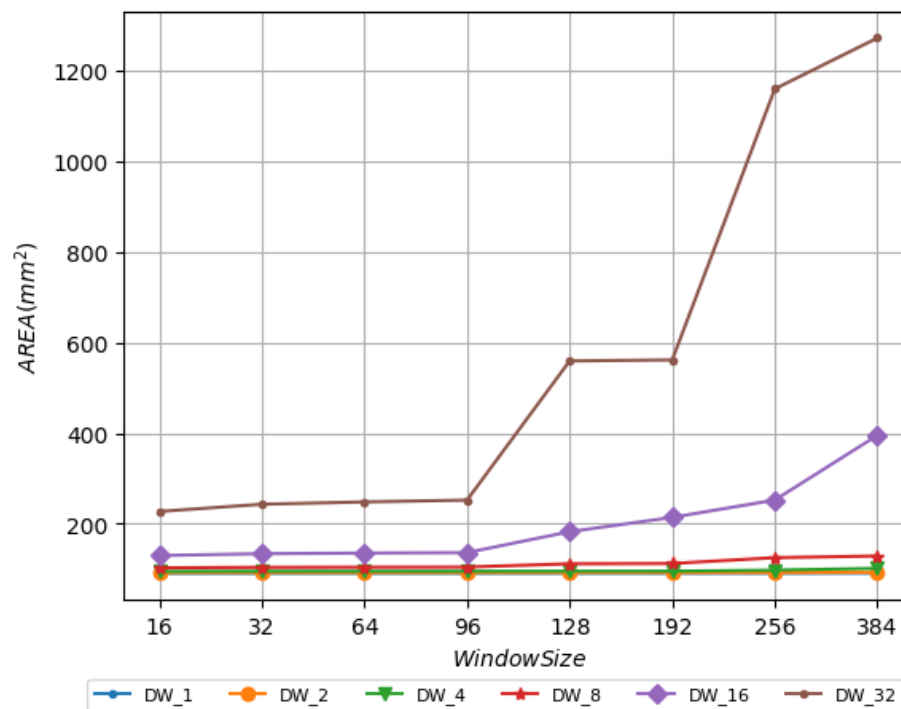
## cactusADM

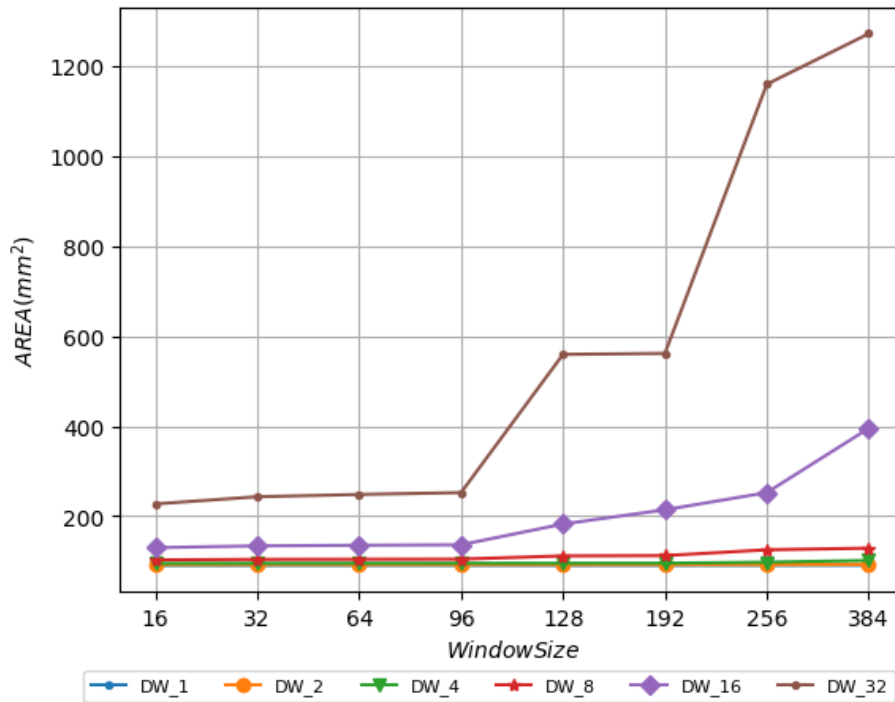


## gobmk



## GemsFDTD





Για όλα τα benchmarks παρατηρούμε πως ελάχιστο μέγεθος τσιπ έχουμε για dispatch width = 1 , 2 ανεξαρτήτως του window size. Επίσης, είναι προφανές πως όσο αυξάνεται το dispatch width και το window size τόσο μεγαλώνουν και οι απαιτήσεις για το μέγεθος του τσιπ.

Τέλος, όσον αφορά τους σύγχρονους επεξεργαστές που έχουμε στους υπολογιστές μας, οι αρχιτέκτονές τους έχουν επιλέξει αντίστοιχες τιμές dispatch width και window size με αυτές που προέκυψαν ως βέλτιστες από τις προσομοιώσεις που πραγματοποιήσαμε.

Συγκεκριμένα, ο επεξεργαστής Kaby Lake της intel έχει dispatch width = 8 και window size = 256. Αντίστοιχα, προγενέστερος επεξεργαστής όπως είναι ο Haswell έχει dispatch width = 8 και window size = 192.

Οι επιλογές τους, λοιπόν, βασίστηκαν σε έναν βέλτιστο συνδυασμό απόδοσης και εξοικονόμησης ενέργειας. Όπως προέκυψε από τις προσομοιώσεις μας, βέλτιστη απόδοση του συστήματος επιτυγχάνεται για μεγάλες τιμές window size (μεγαλύτερες του 192) και για dispatch width από 8 έως 32. Από την άλλη, είδαμε πως ελάχιστο ή αρκετά μικρό EDP λαμβάνουμε για ενδιάμεσες τιμές window size (έως και την τιμή 192) και επίσης για μικρές τιμές dispatch width (ιδανικά 1, 2 ή 4). Επομένως, δεδομένου πως για τιμή dispatch width = 8 έχουμε μικρή απόκλιση από το ελάχιστο EDP και ταυτόχρονα επιτυγχάνεται μέγιστη απόδοση, είναι πολύ λογική και αναμενόμενη η επιλογή της από τους αρχιτέκτονες των σύγχρονων επεξεργαστών.

Έτσι, λοιπόν, μέσω των πραγματικών περιπτώσεων του Kaby Lake και του Haswell επεξεργαστή, συμπεραίνουμε πως τα αποτελέσματα των προσομοιώσεών μας ανταποκρίνονται απόλυτα στην πραγματικότητα.

## Αρχιτεκτονική Kaby Lake processor – παρόμοια με του Skylake

