



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**  
**ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ**  
**ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ**

**Επανασχεδιασμός & Επεκτασιμότητα Λογισμικού Ανοιχτού  
Κώδικα για την Χωροθέτηση Ολοκληρωμένων Κυκλωμάτων**

**Κρανός Γεώργιος**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**  
**Επιβλέπων**  
**Δαδαλιάρης Αντώνιος**

**Λαμία, 2022**



**UNIVERSITY OF THESSALY**

**SCHOOL OF SCIENCE**

**INFORMATICS AND COMPUTATIONAL BIOMEDICINE**

**Redesign & Extensibility of Open Source EDA Tools for  
Integrated Circuits Placement**

**Kranas Georgios**

**Master thesis  
Supervisor  
Dadaliaris Antonios**

**Lamia, 2022**





**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ  
ΚΑΤΕΥΘΥΝΣΗ**

**«ΠΛΗΡΟΦΟΡΙΚΗ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΣΦΑΛΕΙΑ, ΔΙΑΧΕΙΡΙΣΗ  
ΜΕΓΑΛΟΥ ΟΓΚΟΥ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ»**

**Επανασχεδιασμός & Επεκτασιμότητα Λογισμικού Ανοιχτού  
Κώδικα για την Χωροθέτηση Ολοκληρωμένων Κυκλωμάτων**

**Κρανάς Γεώργιος**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Επιβλέπων  
Δαδαλιάρης Αντώνιος**

**Λαμία, 2022**

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο [«τίτλος εργασίας»] αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο ΔΗΛΩΝ

Κρανάς Γεώργιος

Ημερομηνία 14/10/22

Υπογραφή

**Επανασχεδιασμός & Επεκτασιμότητα Λογισμικού Ανοιχτού  
Κώδικα για την Χωροθέτηση Ολοκληρωμένων Κυκλωμάτων  
Κρανάς Γεώργιος**

**Τριμελής Επιτροπή:**

Δαδαλιάρης Αντώνιος

Λουκόπουλος Αθανάσιος

Τζιρίτας Νικόλαος

**Επιστημονικός Σύμβουλος:**

Δαδαλιάρης Αντώνιος

## ΠΕΡΙΛΗΨΗ

---

Η διαδικασία του Placement, αποτελεί ένα βασικό βήμα της ροής Φυσικής Σχεδίασης Ολοκληρωμένων Κυκλωμάτων, κατά την οποία τα δομικά στοιχεία ενός κυκλώματος εισάγονται στην περιοχή που έχει οριστεί ενώ παράλληλα θα πρέπει να αποφεύγονται σχεδιαστικά λάθη και να βελτιστοποιείται η δεδομένη σχεδίαση βάση συγκεκριμένων μετρικών. Για την εφαρμογή κάθε βήματος του Placement γίνεται χρήση διαφόρων αλγορίθμων κι ευριστικών διαδικασιών. Στο πρώτο κεφάλαιο της εργασίας παρουσιάζεται ένα βοηθητικό εργαλείο ανάπτυξης και σχεδιασμού τέτοιων αλγορίθμων, το Bookshelf Format Parser. Έπειτα παρατίθενται οι παραλλαγές ενός αλγορίθμου που επικεντρώνεται στο δεύτερο βήμα του Placement, την φάση του Legalization, ενώ παράλληλα μπορούν να χρησιμοποιηθούν και ως ξεχωριστές διαδικασίες του τρίτου βήματος που είναι το Detailed Placement.

## ABSTRACT

---

Placement is a key procedure of the flow Integrated Circuits Physical Design in which, the structural elements of a design, are placed into a defined area. At the same time, any design errors that occur need to be avoided and the final outcome of the design should be optimized based on a set of metrics. There is a number of algorithms and heuristics that perform upon any step of the Placement procedure. In the first chapter of this thesis, Bookshelf Format Parser, an auxiliary tool designed to help with the development of such algorithms is presented. The second and third chapters, are emphasizing upon the development of a number of procedures targeting the Legalization phase of Placement and are based upon an original algorithm.



## Πίνακας περιεχομένων

---

ΠΕΡΙΛΗΨΗ.....	7
ABSTRACT.....	8
ΕΙΣΑΓΩΓΗ.....	10
PLACEMENT, ΧΩΡΟΘΕΤΗΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ .....	11
ΚΕΦΑΛΑΙΟ 1: REDESIGN, EXTENSIBILITY & EVALUATION OF A PLACEMENT UTILITIES TOOLSET .....	13
Εισαγωγή.....	13
Θεωρητικό Υπόβαθρο.....	13
Σχεδιασμός & Υλοποίηση .....	15
Αξιολόγηση & Συμπεράσματα .....	19
ΚΕΦΑΛΑΙΟ 2: VARIATIONS ON A CONNECTIVITY-BASED LEGALIZER FOR STANDARD CELL DESIGN .....	22
Εισαγωγή.....	22
Σχεδιασμός των Παραλλαγών.....	23
Αξιολόγηση & Συμπεράσματα .....	26
ΚΕΦΑΛΑΙΟ 3: EXPLOITING NET CONNECTIVITY IN LEGALIZATION AND DETAILED PLACEMENT SCENARIOS.....	30
Εισαγωγή.....	30
Νέες Προσεγγίσεις.....	30
Αποτελέσματα & Σχολιασμός .....	31
ΣΥΝΟΨΗ.....	38
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	39

Η Χωροθέτηση Ολοκληρωμένων Κυκλωμάτων ή Placement είναι ένα από τα βήματα της ροής Φυσικής Σχεδίασης κι επηρεάζει σε αρκετά μεγάλο βαθμό το τελικό αποτέλεσμα μιας σχεδίασης. Κατά τη διαδικασία του Placement, όλες οι δομικές υπομονάδες της σχεδίασης, τοποθετούνται σταδιακά εντός μιας καθορισμένης ορθογώνιας περιοχής. Εφόσον η επιρροή της είναι αρκετά σημαντική για το αποτέλεσμα, κατά τη διαδικασία αυτή, εφαρμόζονται παράλληλα και διάφορες μέθοδοι βελτιστοποίησης με στόχο διάφορες μετρικές που σχετίζονται με το Placement όπως το συνολικό μήκος καλωδίου διασύνδεσης (hrwl).

Για την πρακτική εφαρμογή της διαδικασίας του Placement έχουν δημιουργηθεί αρκετοί αλγόριθμοι οι οποίοι λειτουργούν πάνω στις τρεις διακριτές της φάσεις, το Global Placement, το Legalization και το Detailed Placement. Στην πρώτη φάση γίνεται μαζική τοποθέτηση των δομικών μονάδων στην περιοχή του κυκλώματος σε περιοχές που θεωρούνται ιδανικές αλλά ενδεχομένως να παρουσιάζουν σχεδιαστικά σφάλματα. Τα λάθη αυτά διορθώνονται στην επόμενη φάση του Legalization και τέλος κατά την τρίτη φάση, πραγματοποιούνται διάφορες αλλαγές με σκοπό την περεταίρω βελτιστοποίηση.

Η δημιουργία τέτοιων αλγορίθμων κι ευριστικών διαδικασιών αποτελεί το κύριο κομμάτι έρευνας στο τομέα του Placement. Πριν τη δημιουργία, την εκτέλεση και την αξιολόγησή τους, υπάρχει ένα στάδιο προεργασίας κατά το οποίο είναι απαραίτητο να γίνει απεικόνιση μιας σχεδίασης στη μνήμη του υπολογιστή με βάση κάποιο συγκεκριμένο πρότυπο σχεδιάσεων όπως το Bookshelf Format. Η μεθοδολογία αυτής της προεργασίας αφήνεται συνήθως στην κρίση του εκάστοτε προγραμματιστή/σχεδιαστή αφού δεν υπάρχει μεγάλη γκάμα εργαλείων που να εξυπηρετεί τον συγκεκριμένο σκοπό καθώς και όποιες άλλες ανάγκες και λειτουργίες χρειάζεται ο αλγόριθμος προς δημιουργία. Στα παραπάνω πλαίσια, αναπτύχθηκε ένα βοηθητικό εργαλείο το Bookshelf Format Parser, το οποίο αποτελεί προϊόν επανασχεδιασμού ενός προηγούμενου εργαλείου δικής μου δημιουργίας το PyPUT. Στο κεφάλαιο 1, αναλύεται ο τρόπος σχεδιασμού, ανάπτυξης καθώς και οι επιμέρους συγκρίσεις μεταξύ των δύο αυτών εργαλείων.

Το βήμα του Legalization, και οι αλγόριθμοι που χρησιμοποιούνται για την εφαρμογή του, έχουν ως στόχο την εξάλειψη οποιωνδήποτε αστοχιών εμφανίστηκαν κατά το Global Placement. Για την διόρθωση όμως, πρέπει να μετακινηθούν οι δομικές μονάδες της σχεδίασης από την αρχική θέση που τους δόθηκε, μια θέση που θεωρείται ιδανική καθώς πληρεί τα κριτήρια βελτιστοποίησης που χρησιμοποιούνται στο Placement. Στα κεφάλαια 3 και 4 αναλύεται η ανάπτυξη και η αξιολόγηση κάποιων παραλλαγών ενός συγκεκριμένου αλγορίθμου οι οποίοι πέραν από Legalizers, λειτουργούν και ως εργαλεία βελτιστοποίησης, προσπαθώντας να διατηρήσουν ισορροπία μεταξύ της διόρθωσης και της συνολικής απόδοσης της σχεδίασης.

Τέλος, να σημειωθεί πως όλα τα παραπάνω κεφάλαια, αποτελούν επίσης την δουλειά δημοσιευμένων ερευνών στις οποίες είχα ενεργή συμμετοχή κατά τη διάρκεια των μεταπτυχιακών μου σπουδών για την αποφυγή οποιασδήποτε παρερμηνείας.

## PLACEMENT, ΧΩΡΟΘΕΤΗΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ

Η Χωροθέτηση Ολοκληρωμένων Κυκλωμάτων ή Placement αποτελεί ένα βασικό βήμα μιας τυπικής ροής Φυσικής Σχεδίασης. Κατά τη διαδικασία του Placement, ένας αλγόριθμος (Placer) αναλαμβάνει να τοποθετήσει τα δομικά στοιχεία ενός κυκλώματος εντός μίας καθορισμένης περιοχής του, η οποία ονομάζεται die area. Η συγκεκριμένη διαδικασία, αν δεν γίνει αποδοτικά, είναι ικανή να επηρεάσει την ποιότητα του τελικού αποτελέσματος της σχεδίασης σε σημείο που ενδεχομένως να την καθιστά μη-κατασκευάσιμη. Έτσι ένας Placer, θα πρέπει να συμπεριλαμβάνει μεθόδους βελτιστοποίησης με βάση τις οποίες θα πρέπει να βελτιώνει συγκεκριμένες μετρικές ώστε το κύκλωμα να τηρεί τις απαιτήσεις απόδοσης που έχουν οριστεί στα πρώτα βήματα της Φυσικής Σχεδίασης.

Η περιοχή διάταξης των Ολοκληρωμένων Κυκλωμάτων αποτελείται συνήθως από ένα ορισμένο πλήθος γραμμών (rows) ίδιου ύψους πάνω στις οποίες θα τοποθετηθούν τα δομικά στοιχεία. Οι τοποθεσίες που ορίζονται σε κάθε γραμμή είναι συγκεκριμένες σε πλήθος και θεωρούνται ελεύθερες όταν κανένα δομικό στοιχείο δεν έχει καταλάβει τον χώρο. Πρακτικά το σύνολο των γραμμών συνήθως καθορίζει και το die area του κυκλώματος, με δεδομένο ότι αυτές οι γραμμές δεν έχουν κενό μεταξύ τους. Τα δομικά στοιχεία της σχεδίασης χωρίζονται σε δύο διακριτές κατηγορίες, σε μετακινήσιμα (non-terminal) και μη-μετακινήσιμα (terminal). Τα μετακινήσιμα συνήθως ονομάζονται cells, ενώ τα μη-μετακινήσιμα χωρίζονται επίσης σε τερματικά κελιά (macros) και σε απολήξεις διασυνδέσεων εισόδου/εξόδου (I/O Pins). Τα macros πρακτικά είναι στοιχεία αρκετά μεγαλύτερα από τα cells που συνήθως καταλαμβάνουν χώρο σε παραπάνω από μια γραμμές καθώς κι επίπεδα της σχεδίασης ενώ τα I/O Pins τοποθετούνται στην περιφέρεια της σχεδίασης. Τέλος, τα cells έχουν σταθερό και ίσο πάντα ύψος με αυτό των γραμμών ώστε να μπορούν να τοποθετηθούν εντός τους, ενώ έχουν μεταβλητό πλάτος. Προφανώς, επειδή η λειτουργία ενός Placer περιλαμβάνει όλα τα δομικά στοιχεία της σχεδίασης, μπορούμε να διαπιστώσουμε πως δημιουργείται δυσκολία, κυρίως λόγω των macros τα οποία συνήθως λαμβάνουν συγκεκριμένες τοποθεσίες από το βήμα του Floorplanning.

Η διαδικασία του Placement αποτυπώνεται ως πρόβλημα βελτιστοποίησης υπό περιορισμούς οι οποίες επηρεάζουν την απόδοση του κυκλώματος. Οι περιορισμοί αυτοί ή μετρικές που δύναται να βελτιστοποιηθούν είναι οι παρακάτω. Wirelength ή Συνολικό μήκος καλωδίου διασύνδεσης. Συνήθως αναγράφεται και ως Half Perimeter Wirelength – hwpl. Πρακτικά αποτελεί την τιμή του συνολικού μήκους του καλωδίου που χρειάζεται η σχεδίαση ώστε να παρέχει μονοπάτια συνδέσεων μεταξύ των δομικών στοιχείων και των σημείων I/O της σχεδίασης. Η συγκεκριμένη τιμή πρέπει πάντα να διατηρείται σε όσο το δυνατόν χαμηλότερες τιμές καθώς επηρεάζει άμεσα το κόστος κατασκευής, την κατανάλωση ισχύος και τις καθυστερήσεις που μπορεί να εμφανίσει το κύκλωμα. Χρονισμός ή Timing είναι πρακτικά ο κύκλος του ρολογιού ενός chip, ο οποίος υπολογίζεται από την καθυστέρηση που εμφανίζει η κρίσιμη διαδρομή του, που δεν πρέπει ποτέ να ξεπερνά τη μέγιστη τιμή που έχει οριστεί στις προδιαγραφές. Συμφόρηση ή Congestion. Αφορά την συνολική πυκνότητα πληθυσμού δομικών στοιχείων σε συγκεκριμένες περιοχές της σχεδίασης. Υψηλά σημεία συμφόρησης μπορεί να δημιουργήσουν προβλήματα στη σχεδίαση, δημιουργώντας μεγάλες παρακάμψεις δρομολόγησης έως και αδυναμία ολοκλήρωσης όλων των διαδρομών. Εξαιτίας αυτού, πάντα πρέπει να υπάρχει μια ισορροπία στη συγκέντρωση δομικών στοιχείων σε όλες τις περιοχές του chip, σε βάρος του hwpl. Ισχύς ή Power είναι η τιμή που αφορά την κατανάλωση ενέργειας του κυκλώματος. Η συγκεκριμένη τιμή έχει άμεση σχέση με την συμφόρηση, καθώς διάφορα σημεία υψηλής συγκέντρωσης καταναλώνουν μεγαλύτερα ποσά ενέργειας αυξάνοντας παράλληλα την θερμοκρασία. Προφανώς κατάλληλη διανομή θέσεων στα cells, μειώνει παράλληλα και την καταναλισκόμενη ισχύ. Τέλος, ο συνολικός χρόνος εκτέλεσης ενός Placer

είναι επίσης ένας άτυπος περιορισμός ο οποίος προφανώς επηρεάζει την διαδικασία παραγωγής και πρέπει πάντα να μειώνεται όσο είναι δυνατόν.

Το Placement, όπως αναφέρθηκε, αποτελεί ένα βήμα της ροής Φυσικής Σχεδίασης, το οποίο επίσης αποτελείται από τρία επιμέρους βήματα, την Καθολική Χωροθέτηση - Global Placement, την Λεπτομερή Χωροθέτηση - Detailed Placement καθώς και τον ενδιάμεσο βήμα της νομιμοποίησης – Legalization. Κατά το Global Placement, όλες οι δομικές μονάδες τοποθετούνται εντός του die area της σχεδίασης σε θέσεις οι οποίες θεωρούνται ιδανικές ως προς τις μετρικές που αναλύθηκαν στην προηγούμενη παράγραφο. Αν και οι θέσεις αυτές θεωρούνται ιδανικές, τις περισσότερες φορές δημιουργούνται επικαλύψεις μεταξύ cells οι οποίες θεωρούνται «παράνομες» για την σχεδίαση. Τις αστοχίες αυτές έρχεται να διορθώσει το βήμα του Legalization, κατά το οποίο εκτελούνται διάφοροι αλγόριθμοι κι ευριστικές διαδικασίες με σκοπό την επίλυση τους, παράλληλα όμως αποκλίνοντας από την αρχική και ιδανική λύση. Τέλος, κατά το Detail Placement, πραγματοποιούνται διάφορες αλλαγές πάνω στο κύκλωμα με σκοπό την βελτιστοποίηση όσων τιμών πλέον αποκλίνουν από τις αρχικές απαιτήσεις απόδοσης, υπακούοντας πάντα στους κανόνες της σχεδίασης. Να τονιστεί πως η συνολική ποιότητα της παραγόμενης σχεδίασης εξαρτάται κυρίως από την απόδοση του Global Placement.

Η διαδικασία του Placement, αρχικά γινόταν με την εκτέλεση διαφόρων συνδυαστικών μεθόδων. Όταν τα ολοκληρωμένα κυκλώματα βρισκόταν στην κλίμακα των χιλιάδων πυλών, χρησιμοποιούνταν μεθοδολογίες simulated annealing όπως ο αλγόριθμος TimberWolf [30], ενώ φτάνοντας στον επίπεδο των εκατομμυρίων πυλών άρχισε να γίνεται χρήση αλγορίθμων hypergraph partitioning όπως ο Capo [31].

Την ποιότητα αποτελέσματος που παρείχαν οι συνδυαστικές λύσεις ήρθε να ξεπεράσει το quadratic placement, το οποίο επιπρόσθετα ήταν πιο σταθερό στην αντιμετώπιση διαφορετικών περιπτώσεων. Μερικά παραδείγματα τέτοιων αλγορίθμων είναι ο Gordian [32] ο οποίος χρησιμοποιεί quadratic equations για να διαμορφώσει το hwp1 και ο [33] ο οποίος μοντελοποιεί την πυκνότητα της σχεδίασης χρησιμοποιώντας quadratic programming. Επιπρόσθετα έχουν δημιουργηθεί μοντέρνοι Placers όπως ο KraftWerk [34], ο FastPlace [35] και ο SimPL [36] οι οποίοι λειτουργούν σε παρόμοιο πλαίσιο με κάποιες διαφορές.

Η μη γραμμική Χωροθέτηση – non linear Placement, είναι η μεθοδολογία που πλέον παρουσιάζει καλύτερη συνολική απόδοση από τις προηγούμενες. Στον αλγόριθμο [37], αρχικά το hwp1 προσομοιώνεται με εκθετικές συναρτήσεις ενώ η πυκνότητα της σχεδίασης με quadratic equations πάνω στα δομικά στοιχεία που περιέχει. Έτσι βελτιώνεται η ακρίβεια και κατ' επέκταση η ποιότητα του αποτελέσματος. Επιπλέον ακαδημαϊκές μελέτες αποτελούν οι αλγόριθμοι APlace [38] και NTUPlace [39], ενώ ο ePlace [29] είναι ένας από τους καλύτερους αλγορίθμους Global Placement ο οποίος δημιουργεί με την μεθοδολογία του την έννοια της ελάχιστης ποιότητας επιβάρυνσης, επιτυγχάνοντας έτσι πολύ καλές αποδόσεις και τελικό αποτέλεσμα.

# ΚΕΦΑΛΑΙΟ 1: REDESIGN, EXTENSIBILITY & EVALUATION OF A PLACEMENT UTILITIES TOOLSET

---

## Εισαγωγή

---

Στο συγκεκριμένο κεφάλαιο της εργασίας αναλύεται ένα εργαλείο ανοιχτού κώδικα, το οποίο είναι αποτέλεσμα προσωπικής έρευνας κι ενασχόλησης με το πεδίο της Φυσικής Σχεδίασης Ολοκληρωμένων Κυκλωμάτων και πιο συγκεκριμένα το στάδιο του Placement. Το συγκεκριμένο εργαλείο αποτέλεσε κομμάτι της πτυχιακής μου εργασίας [1] και είχε ονομαστεί άτυπα ως Bookshelf Format Parser (BFP). Αν και τότε ήταν σε άρτια λειτουργική κατάσταση κι όπως είχε αναφερθεί βοήθησε στην δημιουργία και την εξέταση διαφόρων αλγορίθμων κι ευριστικών μεθόδων Placement [3][4][5] παραμένει ένα εργαλείο που στοχεύει στην βοήθεια όσων εντάσσονται στο ίδιο ερευνητικό πεδίο, κι όντας τέτοιο θα βρίσκεται συνεχώς σε κατάσταση συντήρησης κι επέκτασης ανεξαρτήτως των λειτουργιών του.

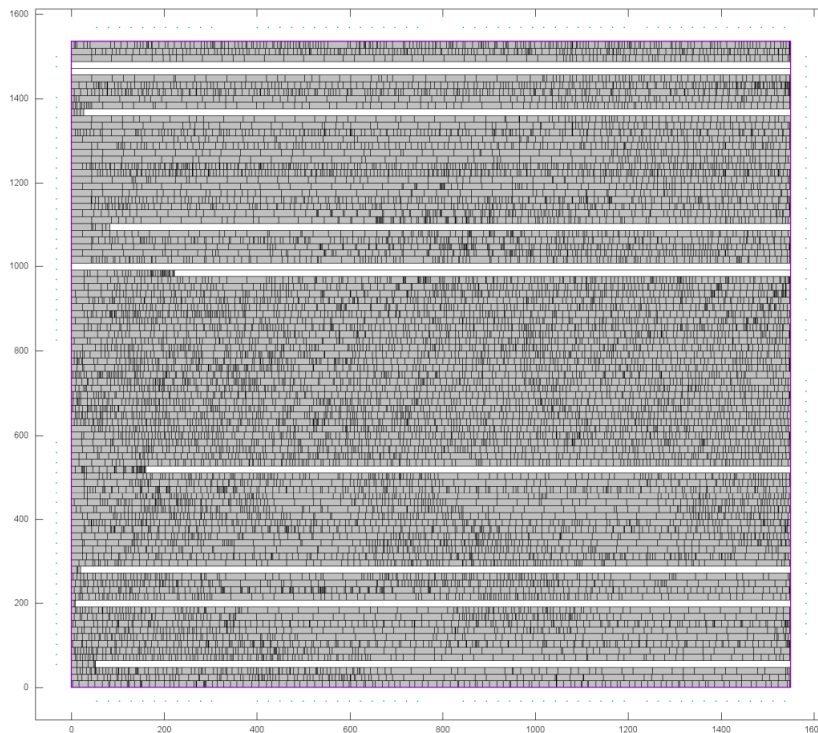
Το 2021, αποφασίστηκε πως το BFP, όντας ερευνητικό εργαλείο βρισκόταν σε ικανό στάδιο ώστε να θεωρηθεί από μόνο του προϊόν μελέτης κι έρευνας και με βάση αυτό συντάχθηκε paper [2] το οποίο δημοσιεύθηκε στο συνέδριο Seeda της ίδιας χρονιάς. Θα ήταν καλό, βεβαίως, να τονιστεί πως το συγκεκριμένο κεφάλαιο δεν αποτελεί αναπαραγωγή του όμοιου της πτυχιακής μου εργασίας, κι έχει στόχο να παρουσιάσει το σύνολο των αποτελεσμάτων που εμφάνισε το εργαλείο, τις διαφορές που έχει από την αρχική του έκδοση καθώς και τις μελλοντικές επεκτάσεις που ενδεχομένως θα προστεθούν. Βεβαίως για λόγους συνοχής θα πρέπει να επεξηγηθούν κάποια θεωρητικά σημεία στα οποία βασίζεται ο σχεδιασμός και η κατασκευή του εργαλείου.

## Θεωρητικό Υπόβαθρο

---

Κατά το βήμα της Φυσικής Σχεδίασης, ένα Ολοκληρωμένο Κύκλωμα πλέον μπορεί να απεικονιστεί με την μορφή διαφόρων γεωμετρικών σχημάτων ώστε η σχεδίαση να περάσει στο επόμενο στάδιο της πραγματικής κατασκευής, όπου με την χρήση πυλώνων πυριτίου θα τυπωθεί το εκάστοτε κύκλωμα. Βεβαίως η απεικόνιση αυτή δεν είναι πάντα συγκεκριμένη και συνήθως εξαρτάται από το πρότυπο στο οποίο επιλέγεται να γίνει η περιγραφή. Ένα από τα πιο συνηθισμένα πρότυπα, το οποίο χρησιμοποιείται κυρίως στην σχεδίαση, την ανάπτυξη καθώς και την αποτίμηση της λειτουργίας αλγορίθμων Placement, είναι το Bookshelf Format [6].

Στο συγκεκριμένο πρότυπο αναπαράστασης, η σχεδίαση απεικονίζεται ως μια βιβλιοθήκη, της οποίας τα βιβλία είναι τοποθετημένα σε συγκεκριμένα ράφια ιδίου μήκους. Στην αναλογία, τα βιβλία είναι τα δομικά μέρη ενός κυκλώματος (cells), ενώ τα ράφια είναι οι γραμμές στις οποίες θα τοποθετηθούν τα cells κατά τη διαδικασία του Placement, μέσω της χρήσης κάποιου αλγορίθμου ή ευριστικής μεθόδου αυτής της κατηγορίας. Επιπρόσθετα τα cells απεικονίζονται πλέον ως ορθογώνια τα οποία έχουν το ίδιο μήκος, αλλά μεταβλητό πλάτος και η ευρύτερη περιοχή στην οποία θα τοποθετηθούν, σε συγκεκριμένη διάταξη και πάνω στις γραμμές, ονομάζεται die area, το οποίο επίσης θα έχει ορθογώνιο σχήμα. Η συνολική αναπαράσταση μπορεί να φανεί στην παρακάτω εικόνα ή οποία είναι μια φωτογραφία που σχηματίστηκε με τη χρήση του εργαλείου μέσω προτύπου Bookshelf Format [Εικόνα 1].



*Εικόνα 1: Στιγμιότυπο Global Placement*

Περνώντας στην πρακτική υπόσταση του συγκεκριμένου προτύπου, πρόκειται για μια συλλογή αρχείων τα οποία περιέχουν τις κατάλληλες πληροφορίες με βάση τις οποίες σχηματίζεται η συνολική εικόνα ενός κυκλώματος με τα στοιχεία τα οποία αναφέρθηκαν στην προηγούμενη παράγραφο. Πιο αναλυτικά, τα αρχεία συνήθως διατηρούν την επωνυμία της σχεδίασης λαμβάνοντας μία συγκεκριμένη επέκταση η οποία σημειώνει τις πληροφορίες τις οποίες περιέχει το εκάστοτε αρχείο. Οι επεκτάσεις είναι οι εξής: `.aux`, `.nodes`, `.pl`, `.nets`, `.scl`, `.wts`. Η πρώτη μορφή αρχείου, περιέχει τις συνολικές πληροφορίες της σχεδίασης σε μια μορφή λίστας που δείχνει σε όλα τα υπόλοιπα αρχεία. Η δεύτερη μορφή αρχείου περιλαμβάνει μια λίστα με όλα τα cells της σχεδίασης τα οποία ονομάζονται εναλλακτικά και nodes. Για κάθε cell δίνονται, η ονομασία του, οι διαστάσεις του (μήκος και πλάτος) καθώς και η πληροφορία αν το εκάστοτε cell είναι μετακινήσιμο ή όχι. Το συγκεκριμένο δεδομένο είναι αρκετά σημαντικό καθώς cells τα οποία χαρακτηρίζονται ως αμετακίνητα είναι συνήθως απολήξεις συνδέσεων καλωδίου στην περιφέρεια (I/O pins), είτε πολυεπίπεδα cells τα οποία επεκτείνονται σε πολλά layers της σχεδίασης, τα οποία ονομάζονται και macros. Ο τρίτος κατά σειρά τύπος αρχείου (`.pl`), κρατάει επίσης πληροφορίες σχετικά με τα cells. Αναλυτικότερα, σε αυτό το αρχείο δίνονται οι συντεταγμένες αρχικής τοποθεσίας των cells σε ορθοκανονικό σύστημα (x,y) καθώς και ο προσανατολισμός του κάθε cell ξεχωριστά. Σε συνδυασμό με τα περιεχόμενα του αρχείου `.nodes`, μπορεί να σχηματιστεί πλέον η συνολική εικόνα κάθε cell, αφού είναι διδιάστατες οντότητες και μπορούν να απεικονιστούν μόνο αν κατέχουμε διαστάσεις και συντεταγμένες. Το αρχείο `.nets`, είναι το τρίτο κατά σειρά σημαντικότητας καθώς περιγράφει τις συνδέσεις τις οποίες δημιουργούν τα cells, δημιουργώντας πλέγματα (nets) μεταξύ τους. Για κάθε net περιέχονται οι πληροφορίες τους μεγέθους του (πόσα cells περιέχει) καθώς και ονομαστικά ποια cells το αποτελούν. Η πληροφορία η οποία μπορεί να εξαχθεί από το συγκεκριμένο αρχείο, σε συνδυασμό με τα προηγούμενα, είναι μια τιμή η οποία ονομάζεται ως

ημiperίμετρος καλωδίου ή Half Perimeter Wirelength (hpwl). Η συγκεκριμένη τιμή αναπαριστά το μήκος καλωδίου διασύνδεσης των πλεγμάτων και των περιεχομένων τους με το υπόλοιπο κύκλωμα και τις απολήξεις του. Για πρακτικούς λόγους το hpwl συνήθως είναι μια τιμή την οποία οι διάφοροι αλγόριθμοι κι ευριστικές διαδικασίες Placement προσπαθούν να ελαχιστοποιήσουν καθώς παράλληλα μειώνεται το κόστος κατασκευής της σχεδίασης και η συνολική θερμότητα ανά περιοχές του chip. Το επόμενο κατά σειρά είδος αρχείου έχει την κατάληξη .scl. Τα συγκεκριμένα αρχεία περιγράφουν τις διαστάσεις των γραμμών που θα καθορίσουν την περιοχή του κυκλώματος (πλάτος, μήκος, συντεταγμένες), πρακτικά σχηματίζοντας την συνολική εικόνα της περιοχής του πυρήνα (die area). Τέλος, υπάρχει το .wts αρχείο, το οποίο περιέχει ένα σύνολο από βάρη τα οποία αναλογούν στα cells που περιγράφονται στα προηγούμενα αρχεία.

Πρακτικά πριν τη δημιουργία οποιουδήποτε placer τα συγκεκριμένα αρχεία θα πρέπει να διαβαστούν, ώστε να εξαχθούν οι χρήσιμες πληροφορίες, αναλόγως με την λειτουργία που επιθυμεί ο σχεδιαστής/προγραμματιστής. Έπειτα θα πρέπει να σχηματιστούν κατάλληλες δομές ώστε τα δεδομένα αυτά να είναι ευκόλως διαχειρίσιμα. Η συγκεκριμένη διαδικασία είναι αρκετά τετριμμένη και φαντάζει εύκολη, αφού αποτελεί ένα προ-σχεδιαστικό βήμα πριν από την κανονική λειτουργία ενός placer. Παρόλα αυτά, μπορεί να αποδειχθεί αρκετά χρονοβόρα, ιδιαιτέρως αν σκεφτεί κανείς πως για κάθε πειραματικό αλγόριθμο θα πρέπει να υπάρχει μια διαδικασία συνεχούς ανανέωσης των περιεχομένων καθώς και των δεδομένων που υπολογίζονται με τους τρόπους που αναλύθηκαν στην προηγούμενη παράγραφο. Επιπλέον θα πρέπει να τονιστεί πως αν η μέθοδος ανάκτησης και υπολογισμού των δεδομένων γίνει με μη βέλτιστο τρόπο, ο placer και οι λειτουργίες του θα επηρεαστούν άμεσα, σε χρόνο εκτέλεσης και μέγεθος στην μνήμη του υπολογιστή.

Όπως αναγράφεται στην αντίστοιχη δημοσίευση και στην πτυχιακή μου εργασία, η πρώτη προσπάθεια δημιουργίας ενός εργαλείου που θα προσπαθεί να επιλύσει τα παραπάνω ζητήματα και να επιταχύνει τον χρόνο ανάπτυξης εργαλείων Placement, ήταν το PyPUT [7] το οποίο αποτέλεσε τον προκάτοχο καθώς και την βάση του BFP. Δυστυχώς όμως αποδείχθηκε μέσω της χρήσης του, πως αν και κατάφερε αποδεκτούς χρόνους εκτέλεσης των λειτουργιών του, ήταν αρκετά δύσκολο στην συντήρηση, την επέκταση και τη χρήση του. Έτσι το BFP δημιουργήθηκε αρχικά με σκοπό να επιλύσει αυτά τα προβλήματα καθώς και να επεκτείνει το εύρος των λειτουργιών του PyPUT.

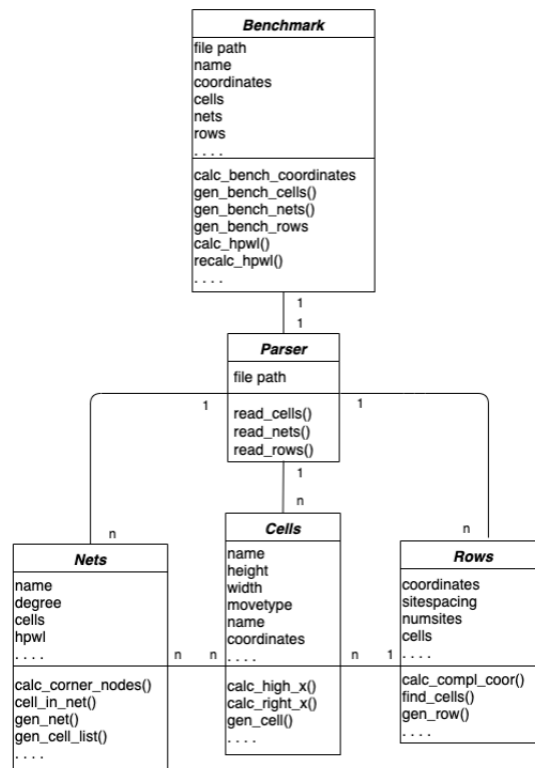
## Σχεδιασμός & Υλοποίηση

---

Ο σχεδιασμός του BFP, είχε ως γνώμονα την βελτιστοποίηση του PyPUT, το οποίο είναι μια συλλογή συναρτήσεων που έχουν τον ίδιο σκοπό, την βοήθεια στην ανάπτυξη αλγορίθμων Placement. Πιο συγκεκριμένα, οι συναρτήσεις του είναι σχεδιασμένες να παρέχουν τις πληροφορίες και τα δεδομένα που έχουν αναφερθεί, με μια πληθώρα τρόπων ώστε να είναι αρκετά ευλύγιστο για τις ανάγκες των χρηστών του. Αν και η κύρια λειτουργία του επετεύχθη, αποδείχθηκε αρκετά δύσκολο στην χρήση, την επέκταση και την συντήρηση, πράγματα τα οποία θα αποδειχθούν και στη συνέχεια κατά την σύγκριση των δύο εργαλείων. Ο κύριος λόγος αυτού του φαινομένου ήταν η επιλογή ανάπτυξης του PyPUT σε συναρτησιακό μοντέλο προγραμματισμού. Έτσι όλες οι λειτουργίες του είναι διαιρεμένες σε πολλαπλές συναρτήσεις με αποτέλεσμα να χρειάζεται αρκετή αναζήτηση και κλήση πολλαπλών συναρτήσεων για μια απλή εργασία πάνω σε οποιοδήποτε βήμα μια τυπικής ροής Placement.

Έτσι λοιπόν ο σχεδιασμός του BFP καλείται να αντιμετωπίσει αυτά τα προβλήματα ξεκινώντας από το μοντέλο υλοποίησης το οποίο πλέον είναι το αντικειμενοστραφές. Η αρχιτεκτονική που σχεδιάστηκε ακολουθεί πιστά το Bookshelf Format. Όπως φαίνεται και στο

σχήμα[Εικόνα 2] κάθε κλάση που θα υλοποιηθεί θα πρέπει να απεικονίζει τις οντότητες που περιγράφονται στα αρχεία ώστε τελικώς να σχηματιστεί μια πλήρης αναπαράσταση του κυκλώματος στη μνήμη του υπολογιστή.



Εικόνα 2: Διάγραμμα κλάσεων του BFP, Πηγή 2

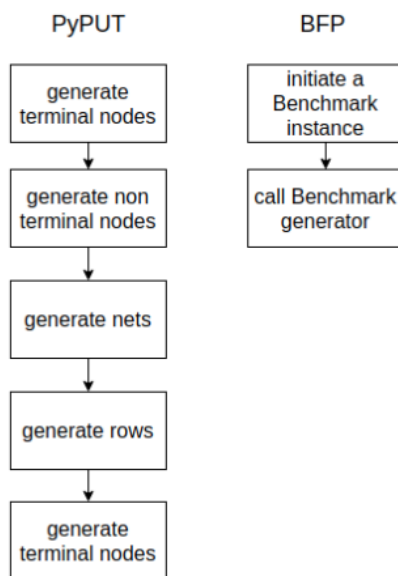
Περνώντας σε μια πιο λεπτομερή ανάλυση του σχεδιασμού, ξεκινάμε από τη βάση ενός κυκλώματος, που είναι τα cells, τα nets και τα rows. Αρχικά η κλάση cells περιγράφει τις αντίστοιχες οντότητες που περιέχονται στα αρχεία .nodes και .pl κάθε σχεδίασης σε Bookshelf Format. Κάθε αντικείμενο cell μετά την κατασκευή του θα περιέχει όλες τις απαραίτητες πληροφορίες για τη δομή του, οι οποίες αναφέρθηκαν και προηγουμένως. Πέραν όμως αυτών, η κλάση περιέχει και μεθόδους οι οποίες βοηθούν στον υπολογισμό των υπόλοιπων δεδομένων, ενώ πρακτικά χρησιμοποιώντας μια μόνο μέθοδο, η οποία θα καλείται επαναληπτικά για κάθε cell μιας σχεδίασης, αυτοματοποιείται η δημιουργία όλων των οντοτήτων χωρίς να χρειάζεται να κληθούν πολλές συναρτήσεις, όπως στο PyPUT. Σε αυτό το σημείο, θα ήταν καλό, να γίνει και η περιγραφή μια κλάσης-εργαλείου, που είναι η Parser. Πρακτικά, η συγκεκριμένη κλάση αποτελεί τον ενδιάμεσο μεταξύ κάθε αντικειμένου που σχηματίζει το εργαλείο και των αρχείων που παρέχει η σχεδίαση. Έτσι για τη δημιουργία κάθε cell καθώς και των αντικειμένων των κλάσεων που θα εξηγηθούν στη συνέχεια γίνεται χρήση αυτής. Επόμενη στη σειρά, είναι η κλάση των nets, που πρακτικά σημειώνει τις συνδέσεις που σχηματίζουν τα cells μεταξύ τους, ενώ για κάθε net που δημιουργείται, πέραν των βασικών πληροφοριών που παρέχει το αρχείο υπολογίζεται και το «τοπικό» hpwl που καταλαμβάνει το κάθε net. Πρακτικά τα περιεχόμενα cells σε κάθε net σχηματίζουν ένα νοητό ορθογώνιο. Με τη χρήση μεθόδων εντός της κλάσης υπολογίζονται τα άκρα αυτού του ορθογώνιου και κατ' επέκταση η ημιπερίμετρός του, χρησιμοποιώντας μια βασική μαθηματική εξίσωση υπολογισμού περιμέτρου ορθογώνιου, διαιρεμένης κατά δύο. Αρχικά να τονιστεί πως η μετρική του hpwl ή του συνολικού μήκους



καλωδίου διασύνδεσης (ας ονομαστεί `hrwl` για μελλοντικές αναφορές), που θα έχει όλο το κύκλωμα είναι το άθροισμα των επιμέρους `hrwl` του κάθε `net`. Έτσι, με βάση αυτό, ο συγκεκριμένος υπολογισμός θα επιταχύνει αρκετά την διαδικασία δημιουργίας του κυκλώματος στη μνήμη του υπολογιστή, γεγονός που υποστηρίζουν και τα αποτελέσματα στη συνέχεια. Επιπρόσθετα σε αυτό το σημείο, αυτομάτως ενημερώνεται κάθε `cell`, ώστε να περιέχει κι αυτό μια λίστα από τα `nets` των οποίων είναι μέρος, για γρηγορότερη αναζήτηση. Τέλος, η κλάση `Row`, όπως περιγράφει και το όνομά της, περιέχει όλες τις πληροφορίες από τα αρχεία `.scl`. Πρακτικά, μέσω αυτής σχηματίζεται το `die area` της σχεδίασης πάνω στο οποίο θα τοποθετηθούν τα `cells`. Επιπρόσθετα, για να υπάρχει μεγαλύτερη ταχύτητα σε περιπτώσεις `Detailed Placement`, σχηματίζονται και άλλες δομές εντός της κλάσης, όπως μια λίστα από τα `cells` που περιέχει κάθε γραμμή πάλι με στόχο την επιτάχυνση διαφόρων διαδικασιών όπως την αναδιάταξη των `cells` εντός μίας συγκεκριμένης γραμμής. Οι τρεις προαναφερθείσες κλάσεις αποτελούν την βάση της απεικόνισης ενός κυκλώματος στον υπολογιστή και όλες μαζί σε σύνολο λειτουργούν μέσω της `top-level` κλάσης που ονομάζεται `Benchmark`. Η συγκεκριμένη είναι αυτή που αυτοματοποιεί όλες τις διαδικασίες κλήσεις των υπολοίπων, συμπεριλαμβανομένου και του `parser` ώστε να σχηματιστεί το σύνολο της σχεδίασης.

Για την υλοποίηση του `BFP`, χρησιμοποιήθηκε η γλώσσα `Python`, και πιο συγκεκριμένα η έκδοση 3.x της γλώσσας. Ο λόγος γι' αυτό είναι πως ο προκάτοχός του, το `PyPUT` ήταν επίσης ανεπτυγμένο στην ίδια γλώσσα, κάτι το οποίο οδηγεί σε μια άμεση ενσωμάτωση ήδη υπάρχοντων συναρτήσεων στο εργαλείο με ελάχιστες τροποποιήσεις ώστε να προσαρμοστούν στο αντικειμενοστραφές μοντέλο. Επιπρόσθετα, η δυκνή φύση της γλώσσας που αναφέρθηκε και πριν, είναι αυτή που επίσης βοήθησε στον εύκολο επανασχεδιασμό και τη μετάβαση από τη συναρτησιακή στην αντικειμενοστραφή έκδοση. Μια ακόμη αιτία είναι οι ήδη υπάρχουσες δομές δεδομένων που προσφέρει η γλώσσα, οι οποίες παρέχουν ένα μεγάλο βαθμό ευλυγισίας δίνοντας εύκολη πρόσβαση και λειτουργίες πάνω στις δομές που δημιουργεί ένας προγραμματιστής, όπως αυτόματα ταξινόμηση σε λίστες αντικειμένων με βάση συγκεκριμένη ιδιότητά τους. Τέλος η `Python`, προσφέρει μια πληθώρα επιστημονικών πακέτων όπως το `NumPy` [8], το `SciPy` [9] και το `Pandas` [10] που διευκολύνουν σύνθετους μαθηματικούς υπολογισμούς ενώ παρέχεται μια γκάμα από τρόπους παρουσίασης των δεδομένων, με ή χωρίς τη χρήση τέτοιων πακέτων.

Περνώντας στην απόδοση που έχει το εργαλείο στα θέματα της συντήρησης, επεκτασιμότητας, ο σχεδιασμός ο οποίος δημιουργήθηκε εξυπηρετεί από μόνος του τους συγκεκριμένους σκοπούς. Η διαίρεση σε κλάσεις με βάση της οντότητες που πρέπει να σχηματιστούν προσφέρει μια διαφάνεια στο σύνολο του εργαλείου καθώς είναι πλέον ξεκάθαρα τα σημεία στα οποία θα πρέπει να γίνουν αλλαγές ή/και προσθήκες αναλόγως με τις ανάγκες του κάθε χρήστη. Σχετικά με την ευκολία χρήσης ή βελτίωση μπορεί να φανεί στο διάγραμμα ροής [Εικόνα 3] αρχικοποίησης μιας σχεδίασης του `BFP` σε αντίθεση με το `PyPUT`. Πλέον το μόνο που χρειάζεται είναι δύο εντολές κώδικα για την απεικόνιση ολόκληρης της σχεδίασης.

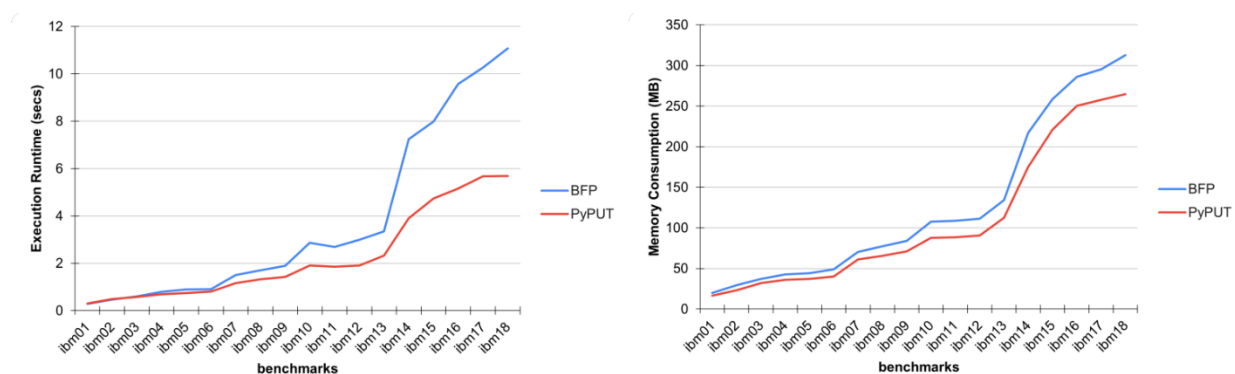


Εικόνα 3: Σενάριο Αρχικοποίησης, Πηγή 2

Σε αυτό το σημείο καλό θα ήταν να αναφερθούν κάποιες προεκτάσεις του εργαλείου οι οποίες δεν έχουν αλλάξει την δομή του, αλλά προσθέτουν στις λειτουργίες του ενώ άλλες τις βελτιώνουν. Οι συγκεκριμένες αλλαγές δεν αναφέρονται στο αντίστοιχο κομμάτι της πτυχιακής μου εργασίας, ούτε και στην σχετική δημοσίευση περί BFP καθώς έγιναν μεταγενέστερα. Η σημαντικότερη όλων ήταν μια επέκταση του υπολογισμού του thrwl. Αρχικά να τονιστεί πως ο υπολογισμός αυτού, αναλόγως με την υλοποίηση μπορεί να αποτελέσει μια από τις πιο χρονοβόρες διαδικασίες εκτέλεσης ενός τέτοιου εργαλείου, γεγονός που μπορεί να επηρεάσει και την εκτέλεση διαφόρων αλγορίθμων και placers που το χρησιμοποιούν ως βοηθό. Στην πρώτη έκδοση του εργαλείου, ο υπολογισμός σχεδιάστηκε ώστε να είναι αρκετά ταχύς αφού χρησιμοποιεί την ελάχιστη δυνατή πληροφορία για να γίνει. Όπως αναφέρθηκε παραπάνω, στις κλάσεις των net και cells έχουν δημιουργηθεί εκτενείς λίστες συσχέτισης οι οποίες πρακτικά προσπερνούν το κομμάτι της αναζήτησης. Ο συγκεκριμένος υπολογισμός σε σενάρια Detailed Placement, όπως τα swaps μεταξύ κελιών δεν χρειάζεται να ξαναγίνεται από την αρχή. Έτσι σε αυτό το σημείο δημιουργήθηκαν μέθοδοι οι οποίες αυτομάτως με κάθε αλλαγή υπολογίζουν την ενδεχόμενη αλλαγή των άκρων ενός net που επηρεάζεται, υπολογίζουν αν χρειαστεί το τοπικό hrwl και τέλος προσαυξάνουν το thrwl αναλόγως χωρίς να χρειαστεί να εξεταστούν και τα υπόλοιπα nets της σχεδίασης. Προφανώς ο συγκεκριμένος υπολογισμός επιταχύνει ακόμη περισσότερο τους υπολογισμούς ιδιαίτερος σε παρόμοια σενάρια. Επιπλέον στο εργαλείο προστέθηκε και ακόμη μια κλάση (Plotter) η οποία αφορά την οπτικοποίηση του κυκλώματος, π.χ. την πλήρη εικόνα του κυκλώματος ή κάποιο χρωματικό χάρτη πυκνότητας ο οποίος δείχνει τη συμφόρηση όλων περιοχών του chip.

Η αξιολόγηση του BFP έγινε μέσω της γλώσσας υλοποίησής του, που ήταν η Python 3.8.5, δημιουργώντας δύο σενάρια εκτέλεσης, ενώ έγινε χρήση εργαλείων αξιολόγησης λογισμικού που παρέχονται από την γλώσσα για την τελική σύγκριση του με το PyPUT. Για την εκτέλεση των σεναρίων χρησιμοποιήθηκαν τα πρότυπα κυκλώματα ISPD [11] τα οποία είναι γραμμένα σε Bookshelf Format.

Το πρώτο σενάριο εκτέλεσης αναφέρθηκε ήδη [Εικόνα 3] και πρόκειται για μια απλή αρχικοποίηση του κυκλώματος από τη στιγμή που διαβάζονται τα αρχεία ενός προτύπου κυκλώματος μέχρι τον πλήρη σχηματισμό όλων των απαραίτητων δομών για την αναπαράσταση της σχεδίασης στη μνήμη του υπολογιστή. Ακολουθώντας λοιπόν τα αντίστοιχα flows εκτέλεσης, τα αποτελέσματα σχετικά με τον χρόνο εκτέλεσης και κατανάλωσης χώρου μνήμης φαίνονται στα παρακάτω δύο γραφήματα [Εικόνα 4]. Και στις δύο περιπτώσεις βλέπουμε πως το PyPUT κρατάει ένα μικρό προβάδισμα σε χρόνο και σε κατανάλωση χώρου, κάτι το οποίο είναι απολύτως αναμενόμενο αν σκεφτεί κανείς πως καλεί μόνο πέντε συναρτήσεις και δημιουργεί αντίστοιχα πέντε δομές δεδομένων αποθήκευσης. Αν και φαινομενικά το BFP καλείται μόνο με δύο εντολές, θα πρέπει να τονιστεί πως αυτό είναι ζήτημα αυτοματισμού κι ευκολίας στη διαδικασία, και πως στο backend, οι δομές που δημιουργούνται είναι πιο πολύπλοκες καθώς και περισσότερες σε πλήθος, το ίδιο και οι κληθείσες μέθοδοι. Παρόλα αυτά οι πληροφορίες που παράγει και υπολογίζει το BFP, αν και στο συγκεκριμένο απλοϊκό σενάριο φαντάζει χειρότερο, είναι αρκετά πολύτιμες για το έργο ενός αλγορίθμου/placer, γεγονός το οποίο αναιρεί τα αρνητικά.



Εικόνα 4: Χρόνος εκτέλεσης εργαλείων & Κατανάλωση μνήμης, Πηγή 2

Το δεύτερο σενάριο, αποτελείται από την εκτέλεση ενός μονολιθικού αλγορίθμου Detailed Placement, κατά το οποίο τα cells όλων των σχεδιάσεων πραγματοποιούν ανταλλαγές θέσεων μεταξύ τους έως ότου βρεθεί για το καθένα μια καλύτερη θέση η οποία θα ελαττώνει το thrwl. Η διαδικασία επαναλαμβάνεται μέχρι να μην υπάρχει άλλη κίνηση χρησιμοποιώντας τον αλγόριθμο mPL6 [12] και το κύκλωμα νομιμοποιείται χρησιμοποιώντας το [13]. Στον πρώτο κατά σειρά πίνακα [Εικόνα 5] που ακολουθεί, σημειώνονται τα αποτελέσματα των εργαλείων για τις 10 πρώτες ανταλλαγές cells εντός μιας κοινής και τυχαίας γραμμής του κυκλώματος σε μετρικές χρόνου εκτέλεσης και κατανάλωσης μνήμης, ενώ στον δεύτερο πίνακα [Εικόνα 5] απεικονίζεται το πλήθος των cells που έχουν εξεταστεί για ενδεχόμενες ανταλλαγές σε ένα συγκεκριμένο χρονικό διάστημα.

design	BFP runtime (sec)	BFP mem (MB)	PyPUT runtime (sec)	PyPUT mem (MB)	design	BFP cells check (60 sec)	BFP cells check (120 sec)	PyPUT cells check (60 sec)	PyPUT cells check (120 sec)
ibm01	0.01	19.70	15.62	19.73	ibm01	114	211	1	1
ibm02	0.04	29.21	50.00	27.80	ibm02	38	77	1	1
ibm03	0.10	36.96	266.08	38.33	ibm03	64	123	1	1
ibm04	0.08	42.65	249.87	43.50	ibm04	57	108	2	2
ibm05	0.01	44.20	87.65	44.37	ibm05	27	54	5	5
ibm06	0.07	48.85	244.26	48.39	ibm06	34	71	1	1
ibm07	0.21	70.31	748.50	72.85	ibm07	32	62	2	2
ibm08	0.09	77.38	667.98	78.54	ibm08	20	41	2	2
ibm09	0.06	84.01	386.48	85.69	ibm09	20	42	1	1
ibm10	0.16	107.46	1021.63	106.36	ibm10	14	30	1	2

Εικόνα 5: Αριστερά, χρόνος εκτέλεσης και κατανάλωση μνήμης για cell swaps. Δεξιά, αριθμός cells swaps που ελέγχθηκαν σε συγκεκριμένο χρονικό διάστημα

Σχετικά με τον χρόνο εκτέλεσης, παρατηρείται πως το BFP νικάει συντριπτικά, αφού η πιο χρονοβόρα διαδικασία του υπολογισμού thrwl είναι ταχύτητα, για λόγους που εξηγήθηκαν, ενώ η προσέγγιση του υπολογισμού στο PyPUT αναγκάζεται να ελέγχει την αλλαγή στην τιμή του thrwl για κάθε ενδεχόμενη ανταλλαγή μεταξύ κελιών, υπολογίζοντας όλες τις διαφορές για κάθε net του κυκλώματος. Σχετικά, με την κατανάλωση μνήμης το BFP επίσης νικάει, δεδομένου ότι αν και ο σχεδιασμός του οδηγεί σε πολύπλοκες δομές δεδομένων με αρκετές συσχετίσεις μεταξύ κλάσεων, το τελικό αποτέλεσμα για κάθε σχεδίαση θα είναι σταθερό πάντα και ανάλογο του μεγέθους του κυκλώματος. Δεδομένο είναι επίσης πως αυτές οι δομές σχηματίζονται στην αρχή δημιουργίας της σχεδίασης λόγω του αντικειμενοστραφούς μοντέλου του εργαλείου και δεν αλλάζουν στο πέρας της εκτέλεσης του αλγορίθμου, κάτι το οποίο δεν συμβαίνει με το PyPUT, το οποίο θα πρέπει να δημιουργεί δομές ώστε να ελέγχει για ενδεχόμενες ανταλλαγές και την επαναφορά στην προηγούμενη κατάσταση σε περίπτωση αποτυχίας βελτίωσης του thrwl. Ο δεύτερος πίνακας, επιβεβαιώνει την καλή λειτουργία του BFP αφού στα συγκεκριμένα χρονικά διαστήματα είναι ικανό να ελέγξει πολύ περισσότερες ενδεχόμενες κινήσεις σε σχέση με τον προκάτοχό του. Το δεύτερο σενάριο, αποτελεί ένα πραγματικό παράδειγμα χρήσης το οποίο δείχνει πως τα πλεονεκτήματα και οι βελτιώσεις του BFP δεν βρίσκονται μόνο στους τομείς της βελτίωσης της χρήσης του αλλά και στον τρόπο με τον οποίο λειτουργεί ως βοήθημα σε σενάρια κανονικής εκτέλεσης αλγορίθμων placement, ιδιαίτερα αν ο τρόπος λειτουργίας τους βασίζεται κυρίως στον τρόπο με τον οποίο γίνεται η διαχείριση των δεδομένων των σχεδιάσεων στις οποίες εξετάζονται.

Ο τρίτος και τελευταίος κύκλος αξιολόγησης του BFP είναι με βάση τον τρόπο ανάπτυξης του και το τελικό αποτέλεσμα ως λογισμικό. Χρησιμοποιώντας το πακέτο της Python, radon 5.0.1 [14] έγινε αξιολόγηση του κώδικα του BFP και του PyPUT πάνω σε Cyclomatic Complexity (CC) [15], Halstead metrics [16] καθώς την μέτρηση του κώδικα στον τομέα της συντήρησης (maintainability index). Το CC, μετράει την πολυπλοκότητα ενός προγράμματος δημιουργώντας μια αναπαράσταση που έχει την μορφή γράφων και βασίζεται στην ακολουθία εκτέλεσης που έχει το πρόγραμμα. Πρακτικά, προσμετράται κάθε γραμμικά ανεξάρτητο τμήμα του γράφου ενώ κάθε τμήμα ελέγχου ροής του κώδικα (διακλαδώσεις, επαναλήψεις) που δημιουργούν μη-γραμμικά μονοπάτια στο γράφο, αυξάνουν την πολυπλοκότητα. Βάση αυτού ο χαμηλότερος βαθμός σημειώνει μικρότερη πολυπλοκότητα και πιο υψηλή βαθμολογία. Το BFP, βαθμολογήθηκε με A, έχοντας βαθμό ίσο με 2,8918 σε αντίθεση με το C που έλαβε το PyPUT και βαθμό 13. Τα Halstead metrics είναι μια σειρά από μετρήσεις πάνω στον κώδικα του προγράμματος. Η ανάλυση τους, παράγει μια πληθώρα πληροφοριών με την πιο βασική να είναι το πλήθος των λαθών που παράγει το λογισμικό σε μέσο όρο. Κάθε αρχείο που σχετίζεται με το BFP και το PyPUT εξετάστηκαν ξεχωριστά και βαθμολογήθηκαν με 0,23 και 0,52 αντίστοιχα με τον μικρότερο βαθμό να σημαίνει λιγότερες εμφανίσεις λαθών στον κώδικα των εργαλείων. Τέλος, το maintainability index, φανερώνει το μέγεθος της πολυπλοκότητας και της δυσκολίας να τροποποιηθεί, να επεκταθεί και να

συντηρηθεί. Στη συγκεκριμένη μέτρηση, όσο μεγαλύτερο ποσοστό επί τις εκατό επιστραφεί σημαίνει πως το πρόγραμμα έχει μικρότερη πολυπλοκότητα άρα είναι και καλύτερο. Εδώ, οι μέσοι όροι που επεστράφησαν ήταν 68,85% με βαθμό A για το BFP και 47,6% με ίδιο βαθμό για το PyPUT.

Συνολικά, όλοι οι τρόποι αξιολόγησης δείχνουν πως το BFP παρουσιάζει καλύτερη λειτουργικότητα από το PyPUT, ενώ παράλληλα επιτυγχάνει αυτά για τα οποία σχεδιάστηκε, ευκολία στη χρήση την επέκταση και τη συντήρηση. Παρόλα αυτά αν και η δομή του αποδίδει άρτια στον σκοπό της, παραμένει ένα βοηθητικό εργαλείο το οποίο θα επεκτείνει τις λειτουργίες του για να εξυπηρετεί τις ερευνητικές ανάγκες όσων ασχολούνται με τον σχεδιασμό και την ανάπτυξη αλγορίθμων κι ευριστικών μεθόδων Placement.

## ΚΕΦΑΛΑΙΟ 2: VARIATIONS ON A CONNECTIVITY-BASED LEGALIZER FOR STANDARD CELL DESIGN

---

### Εισαγωγή

---

Το κεφάλαιο που ακολουθεί, απαρτίζεται από τη μία εκ των τριών δημοσιεύσεων που συντάχθηκαν κατά τη διάρκεια των μεταπτυχιακών μου σπουδών. Πρόκειται για την ανάλυση, το σχεδιασμό, την υλοποίηση και την αξιολόγηση παραλλαγών ενός αλγορίθμου που έχει ως στόχο το Legalization κομμάτι μιας τυπικής διαδικασίας Placement. Το συγκεκριμένο paper [17] δημοσιεύθηκε την ίδια χρονική περίοδο με το BFP, του προηγούμενου κεφαλαίου, στο ίδιο συνέδριο. Πριν την ανάλυση όμως του έργου, παρατίθεται το μια μικρή βιβλιογραφική επισκόπηση καθώς και το υπόβαθρο (αναφερθείς αλγόριθμος) ώστε να υπάρχει συνοχή.

Η νομιμοποίηση (Legalization) είναι η ενδιάμεση διαδικασία μιας τυπικής ροής Placement κι έχει ως στόχο να διορθώσει διάφορα λάθη τα οποία εμφανιστήκαν μετά το Global Placement στάδιο. Τα λάθη αυτά είναι είτε διάφορες επικαλύψεις μεταξύ των cells της σχεδίασης (overlaps) και πιο σπάνια, το φαινόμενο όπου cells προεξέχουν από την περιφέρεια που ορίζει το die area, δημιουργώντας υπερχειλίσσεις (overflows). Αν και το βήμα του Legalization, θεωρείται τις περισσότερες φορές ξεχωριστό στη διαδικασία του Placement, πολλές φορές ενσωματώνεται στο προαναφερθέν στάδιο του Global Placement, το οποίο προηγείται.

Η διαδικασία του Legalization εφαρμόζεται πρακτικά μέσω διαφόρων αλγορίθμων ή ευριστικών διαδικασιών που έχουν ως στόχο να εξαλείψουν τις αστοχίες που δημιουργήθηκαν κατά το προηγούμενο στάδιο, με τους πιο σημαντικούς και αποτελεσματικούς αλγόριθμους του πεδίου να είναι οι Tetris [18] και Abacus [19]. Τα σημεία που τους ξεχωρίζουν από τους υπόλοιπους είναι ο συνολικός χρόνος εκτέλεσής τους καθώς και η δυνατότητα να ελαχιστοποιούν την απόσταση την οποία χρειάζεται να μετατοπιστούν τα cells μιας σχεδίασης, από τη θέση την οποία έλαβαν στο Global Placement, μέχρι την επόμενη διαθέσιμη ώστε να νομιμοποιηθούν. Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, μία από τις σημαντικότερες μετρικές που λαμβάνουν υπόψη οι αλγόριθμοι Placement είναι το συνολικό μήκος καλωδίου διασύνδεσης – Half Perimeter Wirelength (hpwl). Ο στόχος γενικότερα είναι η ελαχιστοποίηση της συγκεκριμένης τιμής και κατ' επέκταση η ελαχιστοποίηση του κόστους κατασκευής του κυκλώματος. Οι συγκεκριμένοι αλγόριθμοι όμως δεν συμπεριλαμβάνουν στις μεταβλητές τους τη συγκεκριμένη μετρική με αποτέλεσμα να χρειάζεται η εφαρμογή επιπλέον αλγορίθμων Detailed Placement με σκοπό να βελτιωθεί το αποτέλεσμά τους. Στην πηγή [20], αναλύεται ένα σύνολο ευριστικών μεθόδων που υιοθετούν τον τρόπο λειτουργίας του Tetris αλλά βελτιώνουν διάφορες μετρικές, σχετικές με το Placement, κάνοντας αλλαγές στον τρόπο που μετακινούνται τα cells καθώς και τη τμηματοποίηση της διαθέσιμης περιοχής του chip ώστε να περιοριστούν κινήσεις που θα αυξήσουν το hpwl. Επιπρόσθετα στη δημοσίευση [21] εφαρμόζεται μια διαδικασία επιδιόρθωσης των overlaps. Αρχικά ο αλγόριθμος, στο πρώτο βήμα του, τοποθετεί εξαρχής τα cells με κατακόρυφη διάταξη, έως ότου οι γραμμές του κυκλώματος βρεθούν στο όριο της χωρητικότητάς τους. Κατά το δεύτερο βήμα, τα overlaps εντός των γραμμών διορθώνονται, με την μοντελοποίηση του προβλήματος ως ένα κλασικό πρόβλημα ελάχιστης διαδρομής (shortest path).

Οι αλγόριθμοι OAL [22] και Domocus [23] μπορούν να χαρακτηριστούν ως παράγωγα του Abacus, με τον πρώτο να υποθέτει πως είναι εφικτό ένα γραμμικό wire length μοντέλο πάνω σε σχεδιάσεις που περιέχουν εμπόδια, ενώ ο δεύτερος εφαρμόζει πρακτικές παραλληλισμού ώστε να μετριάσει τον χρόνο εκτέλεσης και την κατανάλωση μνήμης του

Abacus. Για να γίνει αυτό, οι συγγραφείς, χωρίζουν την σχεδίαση σε ισομεγέθεις και κατακόρυφες ζώνες. Έπειτα ο αλγόριθμος τρέχει παράλληλα για κάθε ζώνη ξεχωριστά συγκεκριμένα μόνο για τα cells που περιέχονται εκεί. Έτσι, αποφεύγονται περεταίρω καθυστερήσεις από επιπλέον μεθοδολογίες συγχρονισμού αφού κάθε περιοχή εξετάζεται ξεχωριστά.

Τέλος, στην δημοσίευση [24], αναλύεται μια προσέγγιση η οποία εκμεταλλεύεται την σύνδεση μεταξύ των cells σε κάθε επανάληψη μιας διαδικασίας Legalization. Τα cells, ελέγχονται σταδιακά, ταξινομημένα σε μια αύξουσα σειρά η οποία προκύπτει από τις συντεταγμένες τους, ξεκινώντας από το αριστερό μέρος κάθε γραμμής που έχει ένα κύκλωμα. Για κάθε cell υπολογίζεται η τιμή του κόστους τοποθέτησής του σε μια γραμμή βάση του displacement που αναλύθηκε προηγουμένως. Ως νέο σημείο τοποθέτησης, επιλέγεται αυτό με το ελάχιστο displacement. Μετά την ολοκλήρωση της μετακίνησης του εκάστοτε cell, επαναλαμβάνουν ακριβώς την ίδια μετακίνηση (απόσταση και κατεύθυνση) όλα τα cells τα οποία συνδέονται μαζί του (ανήκουν στο ίδιο net), εξαιρώντας αυτά τα οποία έχουν ήδη τοποθετηθεί σε προηγούμενη βήμα ή αυτά τα οποία θα βγούνε εκτός των ορίων της περιφέρειας της σχεδίασης αν επαναλάβουν την ίδια κίνηση. Η διαδικασία σταματάει όταν όλα τα cells έχουν λάβει μία ακριβώς φορά τον ρόλο του leading cell, δηλαδή αυτού που έχει κάνει την πρώτη κίνηση κι έχουν οριστικά νομιμοποιηθεί.

Βάση του συγκεκριμένου αλγορίθμου, στην δημοσίευση η οποία είναι θέμα αυτού του κεφαλαίου, προτείνονται τέσσερις παραλλαγές του οι οποίες μπορούν να εφαρμοστούν σε κάθε είδους σχεδίαση. Οι συγκεκριμένες προσεγγίσεις καταφέρνουν να μειώσουν την μετρική hrwl πραγματοποιώντας αλλαγές στον τρόπο κατά τον οποίο μετακινούνται τα cells, διατηρώντας πάντα την εύρυθμη λειτουργία του αρχικού αλγορίθμου, παράγοντας δηλαδή νόμιμες σχεδιάσεις.

## Σχεδιασμός των Παραλλαγών

---

Το κύριο πλεονέκτημα που έχει ο αλγόριθμος της δημοσίευσης [24] είναι οι συνδέσεις που σχηματίζουν τα cells μεταξύ τους, καθώς το αρχικό cell (leading cell) οδηγεί τα υπόλοιπα με τα οποία συνδέεται με αποτέλεσμα να μην αυξάνεται επιπλέον το συνολικό μήκος του καλωδίου διασύνδεσης. Όμως ο συγκεκριμένος αλγόριθμος, εξετάζοντας τα cells κάθε γραμμής του κυκλώματος από τα αριστερά στα δεξιά, επιλέγει συνήθως μετακινήσεις κελιών προς τα αριστερά, δημιουργώντας έτσι αρκετά πυκνές περιοχές πάνω στη σχεδίαση. Επιπλέον, δεν υπάρχει κανένα όριο στην πυκνότητα κάθε γραμμής γεγονός το οποίο σημαίνει πως υπάρχει ενδεχόμενο κάποιες γραμμές να μην μπορούν να συνδεθούν με τις απολήξεις του κυκλώματος (unroutable).

Οι παραλλαγές του αρχικού αλγορίθμου, που περιγράφονται στη συνέχεια, στοχεύουν σε standard cell σχεδιάσεις, με τα περιεχόμενα να είναι cells που χαρακτηρίζονται και ως nodes τα οποία ανάλογα με το αν είναι μετακινήσιμα ή όχι χαρακτηρίζονται non-terminal ή terminal αντιστοίχως.

Η πρώτη παραλλαγή ονομάζεται unbounded\_bidirectional (ub). Στην συγκεκριμένη, ο έλεγχος των κελιών για ενδεχόμενη μετακίνηση, ξεκινάει ακριβώς όπως και ο αρχικός αλγόριθμος, δηλαδή σε αύξουσα κατά x-συντεταγμένη σειρά, τον οποίο λόγω του τρόπου λειτουργίας του μπορούμε να τον ονομάσουμε άτυπα ως NetTetris για επόμενες αναφορές. Η διαφοροποίηση της συγκεκριμένης προσέγγισης ξεκινάει από τον τρόπο με τον οποίο βρίσκει τις διαθέσιμες περιοχές στις οποίες δύναται να τοποθετήσει το εκάστοτε cell. Ενώ ο NetTetris εξετάζει τις ελεύθερες περιοχές μιας γραμμής από τα αριστερά προς τα δεξιά, η παραλλαγή ub,

ελέγχει για διαθέσιμες περιοχές και στις δύο κατευθύνσεις. Το κριτήριο επιλογής παραμένει η ελάχιστη μετατόπιση από την αρχική θέση του cell, δηλαδή το ελάχιστο displacement. Αφού γίνει η επιλογή της νέας θέσης, το cell μετακινείται σε αυτή και πλέον θεωρείται ως leading cell, βάση του οποίου όλα τα υπόλοιπα cells που είναι συνδεδεμένα με αυτό (ανήκουν στο ίδιο net) θα μετακινηθούν με τον ίδιο ακριβώς τρόπο, κατά την ίδια κατεύθυνση και απόσταση, αν αυτό δεν δημιουργεί κάποια παράβαση όπως αναφέρθηκε και προηγουμένως. Από αυτή την διαδικασία εξαιρούνται cells τα οποία ήδη τοποθετηθήκαν σε μια καινούρια θέση, σε προηγούμενο βήμα εκτέλεσης της διαδικασίας. Να σημειωθεί επίσης πως αν ένα cell αποτελεί κομμάτι του ίδιου net με το leading cell πάνω από μια φορά θα συνεχίσει να μετακινείται αναλόγως, όσες φορές είναι το πλήθος αυτό.

Η δεύτερη παραλλαγή του NetTetris είναι η bounded\_bidirectional\_dens\_limit (bd). Η διαδικασία της συγκεκριμένης παραλλαγής είναι όμοια με την ub, με την διαφορά πως πλέον τίθεται όριο στην πυκνότητα πληθυσμού cells που μπορεί να φτάσει κάθε γραμμή της σχεδίασης. Έτσι αν μια ενδεχόμενη βέλτιστη κατά displacement κίνηση αυξάνει τη συγκεκριμένη τιμή άνω του ορίου, τότε η συγκεκριμένη κίνηση απορρίπτεται και η διαδικασία επαναλαμβάνεται έως ότου να πληρούνται και τα δύο κριτήρια, αυτά της πυκνότητας και του ελάχιστου displacement, με το πρώτο να έχει προτεραιότητα έναντι του δεύτερου.

Η επόμενη παραλλαγή ονομάζεται unbounded\_bidirectional\_div (ubd). Η συγκεκριμένη μέθοδος ακολουθεί επίσης τον τρόπο ελέγχου και των δύο κατευθύνσεων που έχει η ub καθώς και τον τρόπο επιλογής με μόνο κριτήριο να είναι το ελάχιστο displacement. Η διαφορά έγκειται στον τρόπο κίνησης των διασυνδεδεμένων cells με το leading. Αν ένα τέτοιο cell είναι κομμάτι του ίδιου net με το leading πάνω από μία φορά, τότε το cell αυτό μετακινείται κατά την ίδια κατεύθυνση αλλά με απόσταση η οποία θα υπολογίζεται από την παρακάτω εξίσωση.

$$total\_disp = \sum_0^{n-1} \frac{orig\_disp}{2^n}$$

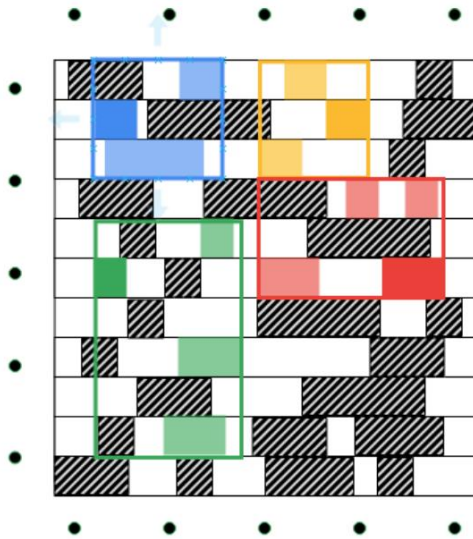
Στην συγκεκριμένη εξίσωση,  $n$  είναι ο αριθμός των κοινών nets και  $orig\_disp$  είναι η τιμή της μετατόπισης που έχει λάβει το leading cell. Έτσι πρακτικά, κάθε φορά που το διασυνδεδεμένο cell θα πρέπει να ακολουθήσει την ίδια κίνηση με το leading πρακτικά θα μειώνει την απόσταση της μετατόπισής του κατά το μισό της απόστασης που διένυσε σε προηγούμενη κίνηση. Αν για παράδειγμα ένα cell πρέπει να ακολουθήσει το leading τρεις φορές (ανήκουν σε τρία κοινά nets) τότε την πρώτη φορά θα μετακινηθεί όσο το leading cell, έστω  $x$ , την δεύτερη φορά  $x/2$  και την τελευταία φορά  $x/4$ .

Η τέταρτη και τελευταία προσέγγιση που σχεδιάστηκε και αναπτύχθηκε στα πλαίσια της δημοσίευσης ονομάζεται bounded\_bidirectional\_div\_dens\_limit (bdd). Πρακτικά αποτελεί τον συνδυασμό των δύο προηγούμενων παραλλαγών, bd και ubd εφαρμόζοντας τους ελέγχους περί πυκνότητας γραμμής και μετατόπισης cell ενώ παράλληλα υιοθετείται και ο τρόπος υπολογισμού της απόστασης μετατόπισης των διασυνδεδεμένων cells που αναφέρθηκε.

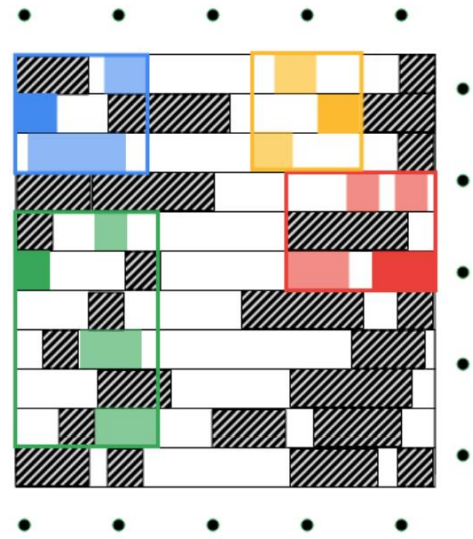
Για να γίνει λίγο πιο κατανοητός ο τρόπος λειτουργίας των παραλλαγών, ακολουθούν κάποιες αναπαραστάσεις της διαδικασίας πάνω σε κύκλωμα μικρής κλίμακας. Στην πρώτη εικόνα [Εικόνα 6] φαίνονται τέσσερα διαφορετικά nets στα οποία έχουν σημειωθεί τα leading cells με έντονο χρώμα και τα following cells με ένα ποσοστό διαφάνειας στο χρώμα. Στην δεύτερη εικόνα [Εικόνα 7] παρουσιάζεται ένα επόμενο στιγμιότυπο της διαδικασίας κατά το οποίο έχουν ήδη γίνει οι έλεγχοι για τις νέες τοποθεσίες κι έχουν γίνει οι απαραίτητες μετακινήσεις, αρχικά των leading cells κι έπειτα των υπόλοιπων διασυνδεδεμένων στο ίδιο net. Προφανώς στην ίδια διαδικασία μετακινούνται και τα υπόλοιπα (ασπρόμαυρα) cells τα οποία για χάρη του παραδείγματος δεν έχουν σημειωθεί αλλά αποτελούν κομμάτι κάποιων άλλων nets, άσχετων με το στιγμιότυπο. Στην τελευταία εικόνα [Εικόνα 8] τονίζεται ο τρόπος ελέγχου της πυκνότητας γραμμών που υλοποιείται στις παραλλαγές bd και bdd. Για κάθε σημειωμένο



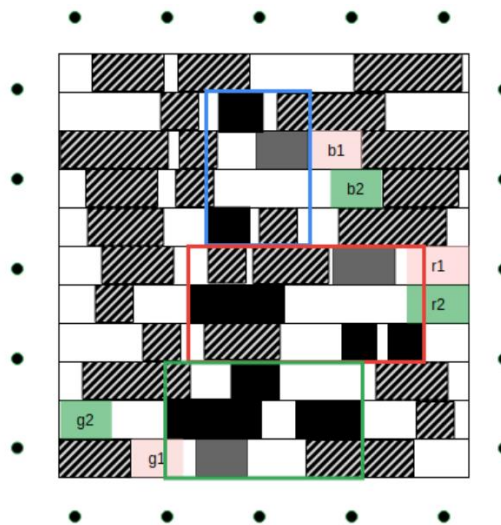
cell εμφανίζονται δύο διαθέσιμες τοποθεσίες εκ των οποίων επιλέγεται εν τέλει ως βιώσιμη λύση για κάθε cell η πράσινη.



Εικόνα 6: Επιλογή Nets, Πηγή 17



Εικόνα 7: Επανατοποθέτηση Nets μέσω ubd, Πηγή 17



Εικόνα 8: Επανατοποθέτηση Nets μέσω bd, Πηγή 17

## Αξιολόγηση & Συμπεράσματα

Όλες οι παραλλαγές του αλγορίθμου NetTetris υλοποιήθηκαν σε γλώσσα Python, ομοίως με τον αρχικό ο οποίος χρησιμοποιείται ως μέτρο σύγκρισης της λειτουργίας τους. Για την διαδικασία της αξιολόγησης γίνεται εκτέλεση της κάθε παραλλαγής πάνω στα πρότυπα κυκλώματα της αναφοράς [11], ενώ το βήμα του Global Placement έχει γίνει με τη χρήση του αλγορίθμου mPL6 [12]. Για κάθε κύκλωμα και παραλλαγή καταγράφηκαν το ποσοστό βελτίωσης (A) που παρουσιάζει το κύκλωμα έναντι του αρχικού αλγορίθμου (B) στις μετρικές των hpwl, displacement και χρόνο εκτέλεσης χρησιμοποιώντας την φόρμουλα που ακολουθεί.

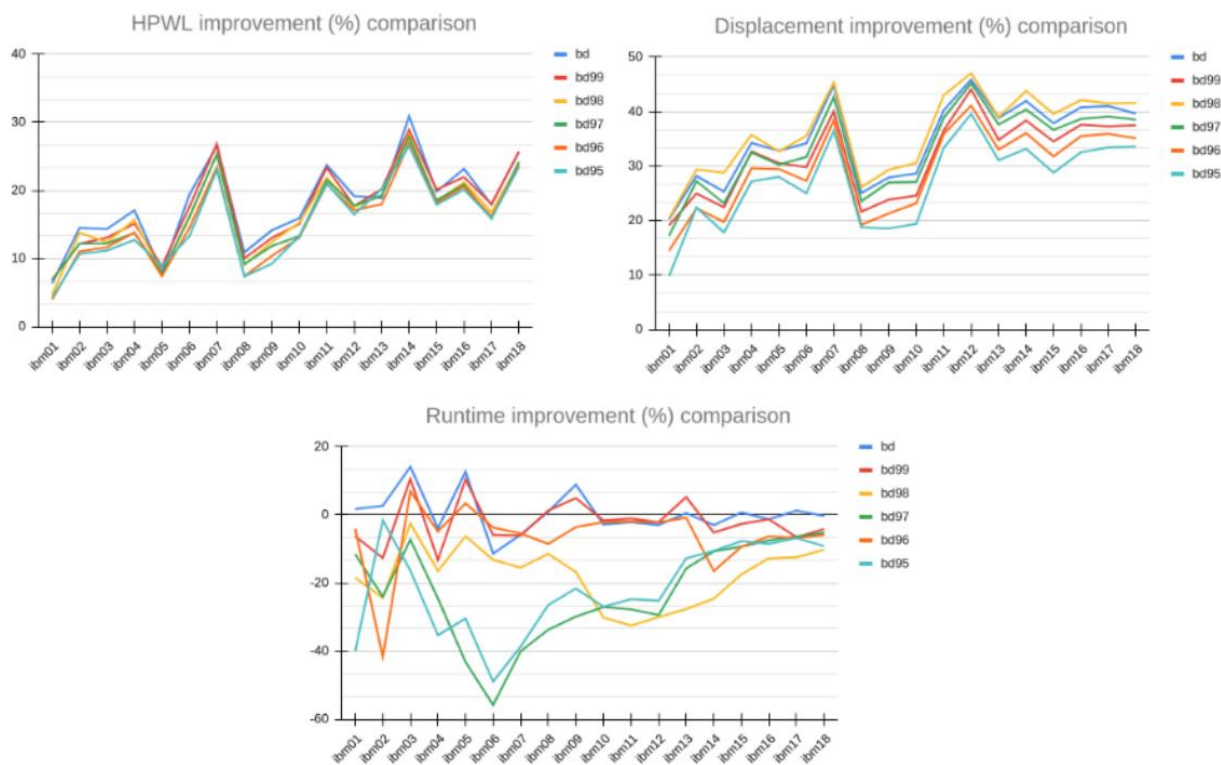
$$improvement = \frac{performance(B) - performance(A)}{performance(B)}$$

Στον πρώτο πίνακα [Εικόνα 9] εμφανίζονται συνολικά τα αποτελέσματα της παραλλαγής ub στην οποία εμφανίζεται αρκετά μεγάλη μείωση στο συνολικό μήκος καλωδίου διασύνδεσης (hpwl) καθώς και στην συνολική μετατόπιση (displacement) των cells, εις βάρος όμως του χρόνου εκτέλεσης, με σχεδόν ασήμαντες αυξήσεις στις περισσότερες περιπτώσεις.

design	HPWL	Displacement	Time
ibm01	6.3%	20.4%	1.7%
ibm02	14.5%	29.3%	2.6%
ibm03	14.3%	28.7%	14.1%
ibm04	17%	35.7%	-3.9%
ibm05	7.9%	32.7%	12.6%
ibm06	19.38%	35.5%	-11.3%
ibm07	26.5%	45.3%	-5.7%
ibm08	10.9%	26.1%	1%
ibm09	14.1%	29.3%	8.8%
ibm10	15.9%	30.6%	-2.8%
ibm11	23.7%	42.3%	-2.1%
ibm12	19.1%	47%	-2.9%
ibm13	18.9%	38.9%	0.5%
ibm14	30.9%	43.8%	3%
ibm15	19.8%	39.5%	7.2%
ibm16	23.1%	42.1%	1.3%
ibm17	17.9%	41.4%	1.2%
ibm18	25.6%	41.6%	-0.3%

Εικόνα 9: Αποτελέσματα ub, Πηγή 17

Στην σειρά γραφημάτων [Εικόνα 10] που ακολουθεί παρατίθενται τα αποτελέσματα της παραλλαγής bd, κατά το οποία έχουν τεθεί διάφορα όρια στην πυκνότητα κάθε γραμμής της σχεδίασης. Λόγω της μεγάλης πυκνότητας των πρότυπων κυκλωμάτων που εξετάζονται, ενδεχόμενη αύξηση του ορίου αυτού οδηγεί σε λύσεις που εμφανίζουν λάθη στη σχεδίαση. Από τα γραφήματα φαίνεται μια αρκετά σημαντική βελτίωση στο hrwl καθώς και στο displacement, πάλι σε βάρος του χρόνου εκτέλεσης το οποίο έχει αυξηθεί αρκετά.



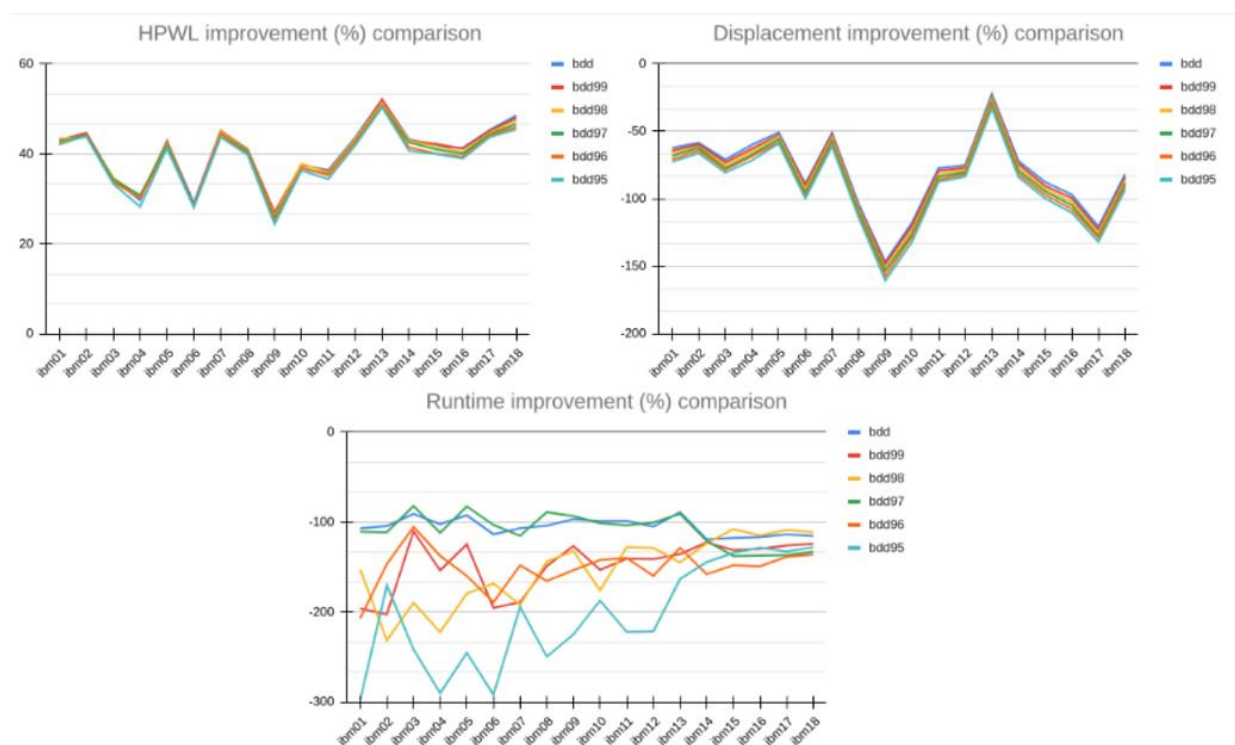
Εικόνα 10: Αποτελέσματα bd, Πηγή 17

Στον επόμενο πίνακα [Εικόνα 11], ακολουθούν τα αποτελέσματα της προσέγγισης ubd. Το hrwl στην συγκεκριμένη περίπτωση έχει μειωθεί εξαιρετικά σε σχέση με την πρώτη παραλλαγή ub, σε όλες τις περιπτώσεις. Αυτό αποδίδεται στο γεγονός πως κάθε κίνηση που πραγματοποιεί ένα following cell, υποδιπλασιάζεται σε απόσταση με βάση την ανάλυση που έγινε στο προηγούμενο κομμάτι του κεφαλαίου. Έτσι οι κινήσεις που ευθύνονται για την αύξηση του hrwl, σε αυτή την παραλλαγή μειώνονται εξαιρετικά πιο πολύ. Δυστυχώς όμως, η ανταλλαγή αυτού είναι πως σε αυτό το σενάριο το displacement είναι χειρότερο ενώ και σε αυτή την προσέγγιση ο χρόνος εκτέλεσης επίσης χειροτερεύει.

design	HPWL	Displacement	Time
ibm01	43.17%	-62.54%	-106.96%
ibm02	44.47%	-58.42%	-104.29%
ibm03	34.39%	-70.91%	-90.85%
ibm04	29.73%	-59.93%	-102.21%
ibm05	42.78%	-50.89%	-92.35%
ibm06	29.09%	-88.82%	-113.54%
ibm07	45.17%	-51.13%	-106.73%
ibm08	40.89%	-103.86%	-103.96%
ibm09	26.84%	-146.62%	-96.76%
ibm10	37.53%	-117.84%	-98.83%
ibm11	36.39%	-77.32%	-98.67%
ibm12	43.64%	-75.11%	-105.02%
ibm13	52.13%	-22.39%	-88.90%
ibm14	43.24%	-72.01%	-119.06%
ibm15	41.76%	-87.62%	-117.52%
ibm16	41.34%	-96.89%	-116.66%
ibm17	45.32%	-120.701%	-113.49%
ibm18	48.55%	-81.73%	-115.31%

Εικόνα 11: Αποτελέσματα ubd, Πηγή 17

Με παρόμοιο τρόπο αναλύονται και τα αποτελέσματα της προσέγγισης bdd η οποία επίσης εφαρμόζει ένα όριο πυκνότητας των γραμμών. Στο γράφημα [Εικόνα 12] παρουσιάζονται παρόμοια φαινόμενα με πριν. Με τον περιορισμό της πυκνότητας χειροτερεύουν οι μετρικές των hpwl και displacement.



Εικόνα 12: Αποτελέσματα bdd, Πηγή 17

Συνολικά, η πιο ισορροπημένη παραλλαγή φαίνεται να είναι η ub, δεδομένου των αποτελεσμάτων της. Όμως σε περιπτώσεις που το Global Placement δεν έχει γίνει με αποδοτικό τρόπο αλλά και σε περιπτώσεις που για το Legalization χρειάζεται μία προσέγγιση με βαρύτητα στις συνδέσεις των cells, ανεξαρτήτως του χρόνου εκτέλεσης, η προσέγγιση ubd φαίνεται να είναι η καλύτερη.

Συνοψίζοντας, στο συγκεκριμένο κεφάλαιο παρουσιάστηκαν οι παραλλαγές ενός υπάρχοντος αλγορίθμου ο οποίος δίνει αρκετή βαρύτητα στις διασυνδέσεις που παρουσιάζονται μεταξύ των cells. Κάθε παραλλαγή σκοπεύει στην βελτίωση του αρχικού αλγορίθμου, εκμεταλλευόμενες διάφορα «αδύναμα» σημεία του. Σύμφωνα με τα αποτελέσματα, η προσέγγιση ubd φαίνεται πως είναι η καλύτερη, βάση του hwpl με την μέση βελτίωση να είναι 40.36% αλλά έχοντας παράλληλα αρκετά μεγάλη αύξηση στον χρόνο εκτέλεσης. Από την άλλη πλευρά, η προσέγγιση ub φαίνεται να είναι πιο ισορροπημένη με χειρότερα αποτελέσματα από την ubd σε σχέση με το hwpl αλλά αρκετά καλύτερο χρόνο. Ως γενικό αποτέλεσμα, η απόκλιση από τις κλασσικές μεθόδους Legalization με σκοπό την αποφυγή των αρνητικών τους σημείων και την παράλληλη εφαρμογή ευριστικών διαδικασιών μπορεί να επιτευχθούν αρκετές βελτιώσεις πάνω στο αποτέλεσμα της διαδικασίας. Επιπρόσθετα, με μελλοντικές και πιο περίπλοκες μεθόδους που θα εμβαθύνουν στον τρόπο κίνησης των following cells καθώς και την εφαρμογή μεθόδων παραλληλοποίησης που θα βελτιώνουν αρκετά τον χρόνο εκτέλεσης, οι παραγόμενες παραλλαγές θα ήταν αρκετά πιο χρήσιμες. Τέλος, οι τέσσερις προτεινόμενες παραλλαγές εκτός από σενάρια Legalization θα μπορούσαν να θεωρηθούν και κομμάτια ενός Detail placer, βελτιώνοντας το αποτέλεσμα άλλων Legalizer.

## ΚΕΦΑΛΑΙΟ 3: EXPLOITING NET CONNECTIVITY IN LEGALIZATION AND DETAILED PLACEMENT SCENARIOS

---

### Εισαγωγή

---

Μετά την ολοκλήρωση της δημοσίευσης [17], που αποτελεί και το αντικείμενο του προηγούμενου κεφαλαίου, μας ζητήθηκε από τη συντακτική ομάδα του περιοδικού Information του MDPI, επέκταση πάνω στην συγκεκριμένη δουλειά, ώστε να δημοσιευθεί εκ νέου ως άρθρο στο περιοδικό τους. Έτσι κληθήκαμε να επεκτείνουμε τις παραλλαγές τους αλγόριθμου [24] περεταίρω. Αν και στο αντίστοιχο άρθρο του περιοδικού [25] περιγράφεται η διαδικασία δημιουργίας των νέων και των προηγούμενων παραλλαγών, εφόσον αυτές ήταν κομμάτι του προηγούμενου κεφαλαίου, δεν υπάρχει λόγος να αναλυθούν εξαρχής παρά μόνο όπου κρίνεται απαραίτητο λόγω άμεσης συσχέτισης.

### Νέες Προσεγγίσεις

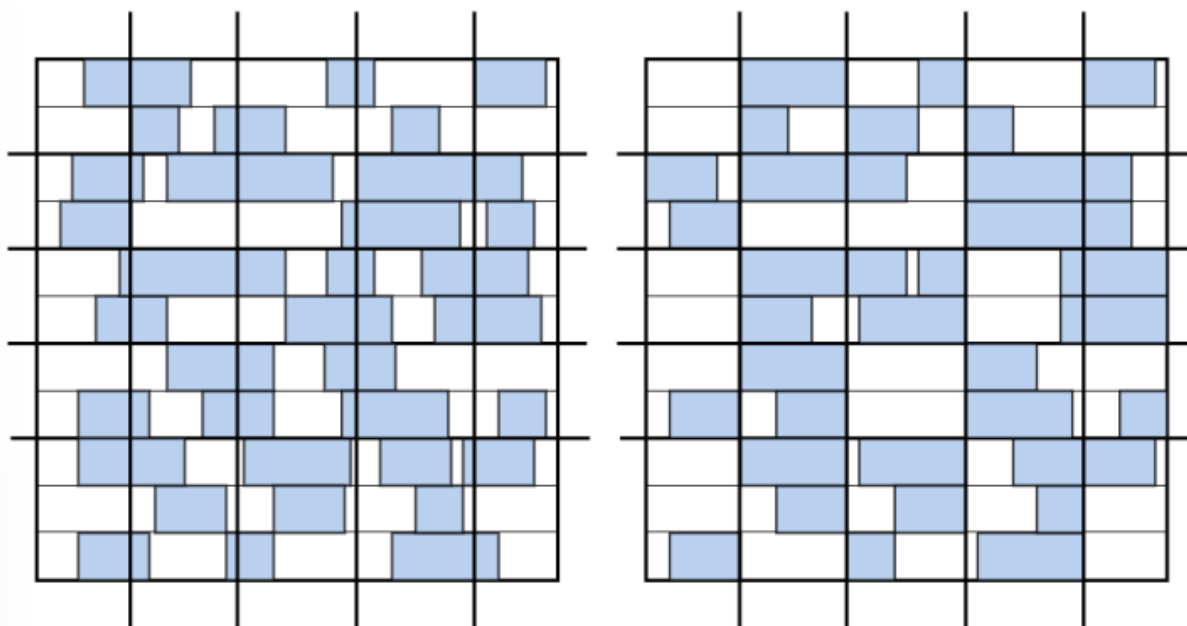
---

Υπάρχουν αρκετοί αλγόριθμοι κι ευριστικές διαδικασίες, οι οποίες καθόλα τα βήματα του Physical Design και πιο συγκεκριμένα του Placement, μπορούν να παράγουν αποδεκτές λύσεις, δηλαδή κυκλώματα που μπορούν να κατασκευαστούν. Το Placement όμως μπορεί να θεωρεί και ως NP πρόβλημα [26][27] με αποτέλεσμα να είναι αντικείμενο συνεχούς μελέτης κι έρευνας με στόχο την συνεχόμενη βελτίωση των διαθέσιμων λύσεων.

Οι πιο συνηθισμένες μετρικές τιμές με τις οποίες αξιολογείται η απόδοση των λύσεων είναι το συνολικό μήκος καλωδίου διασύνδεσης (hprwl) και η συνολική μετατόπιση (displacement) των δομικών μονάδων της σχεδίασης (cells). Η πρώτη τιμή, σχετίζεται άμεσα με την διασυνδεσιμότητα και τον χρονισμό του κυκλώματος ενώ η δεύτερη δείχνει την απόκλιση από την αρχική, αλλά παράλληλα μη-νόμιμη, λύση που έχει σχηματιστεί στο βήμα του Global Placement.

Συνοπτικά, οι τέσσερις παραλλαγές που προτάθηκαν στο [17] λειτουργούν ως Legalizers, δηλαδή διαδικασίες οι οποίες νομιμοποιούν τυχόν ανομίες που παρουσιάστηκαν στο κύκλωμα μετά το Global Placement. Όλες τους, λειτουργούν ελέγχοντας ένα-ένα τα cells μιας σχεδίασης, ψάχνοντας διαθέσιμες θέσεις πάνω στις γραμμές του κυκλώματος με σκοπό να εξαλειφθούν ενδεχόμενα overlaps, με τις διαφορές τους να είναι τα όρια που έχουν οριστεί για την λειτουργία τους καθώς και ο τρόπος με τον οποίο μετακινούν τα εν λόγω cells. Οι καινούριες προσθήκες εκμεταλλεύονται τις ήδη υπάρχουσες ενώ παράλληλα προσθέτουν την δυνατότητα να γίνεται διαχωρισμός του κυκλώματος σε περιοχές ίδιων διαστάσεων, δημιουργώντας ένα πλέγμα πάνω στο οποίο μπορούν να λειτουργήσουν οι προηγούμενες παραλλαγές. Ο αριθμός των περιοχών που δύναται να δημιουργηθούν ορίζεται από τον χρήστη και παραμένει στατικός για όλη τη διάρκεια εκτέλεσης κάθε προσέγγισης. Αν για παράδειγμα δοθεί ως είσοδος το μέγεθος 4, θα δημιουργηθούν 16 περιοχές πάνω στο αρχικό κύκλωμα. Όσον αφορά το πλήθος τιμών, δεν μπορεί να οριστεί μεταξύ αυστηρών ορίων καθώς αυτό εξαρτάται από το συνολικό μέγεθος της σχεδίασης και τις υπομονάδες της. Ο αριθμός αυτός προφανώς δεν μπορεί να είναι το 1, αφού αυτό ορίζει όλη τη σχεδίαση και θεωρητικά δεν πρέπει να δημιουργεί περιοχές μικρότερες σε διαστάσεις από τα cells της σχεδίασης.

Οι δύο καινούριες παραλλαγές έχουν ονομαστεί ως `unbounded_bidirectional_grid` (`ubg`) και `unbounded_bidirectional_div_grid` (`ubdg`). Και οι δύο παραλλαγές χρησιμοποιούν τις `ub` και `ubd` αντίστοιχως ως τον αλγόριθμο του `Legalization` προσαρμόζοντάς την λειτουργικότητά τους στα πλαίσια που ορίζει το πλέγμα. Ένα οπτικό παράδειγμα λειτουργίας παρουσιάζεται στην παρακάτω εικόνα [Εικόνα 13], στην οποία παρουσιάζονται η αρχική κατάσταση κατά τη δημιουργία του πλέγματος και η τελική μετά την εκτέλεση της `ubg`. Να σημειωθεί πως και οι δύο μέθοδοι επιτρέπουν την τοποθέτηση των `cells` μόνο εντός των περιοχών που έχουν οριστεί από το πλέγμα, με αποτέλεσμα να μειώνεται η απόσταση από την τελική τους τοποθεσία (`displacement`), ενώ παράλληλα μειώνεται και η συνολική συγκέντρωση πολλών `cells` σε συγκεκριμένες περιοχές του `chip`.



Εικόνα 13: Αρχική και τελική κατάσταση μετά την εφαρμογή `ubg`, Πηγή 25

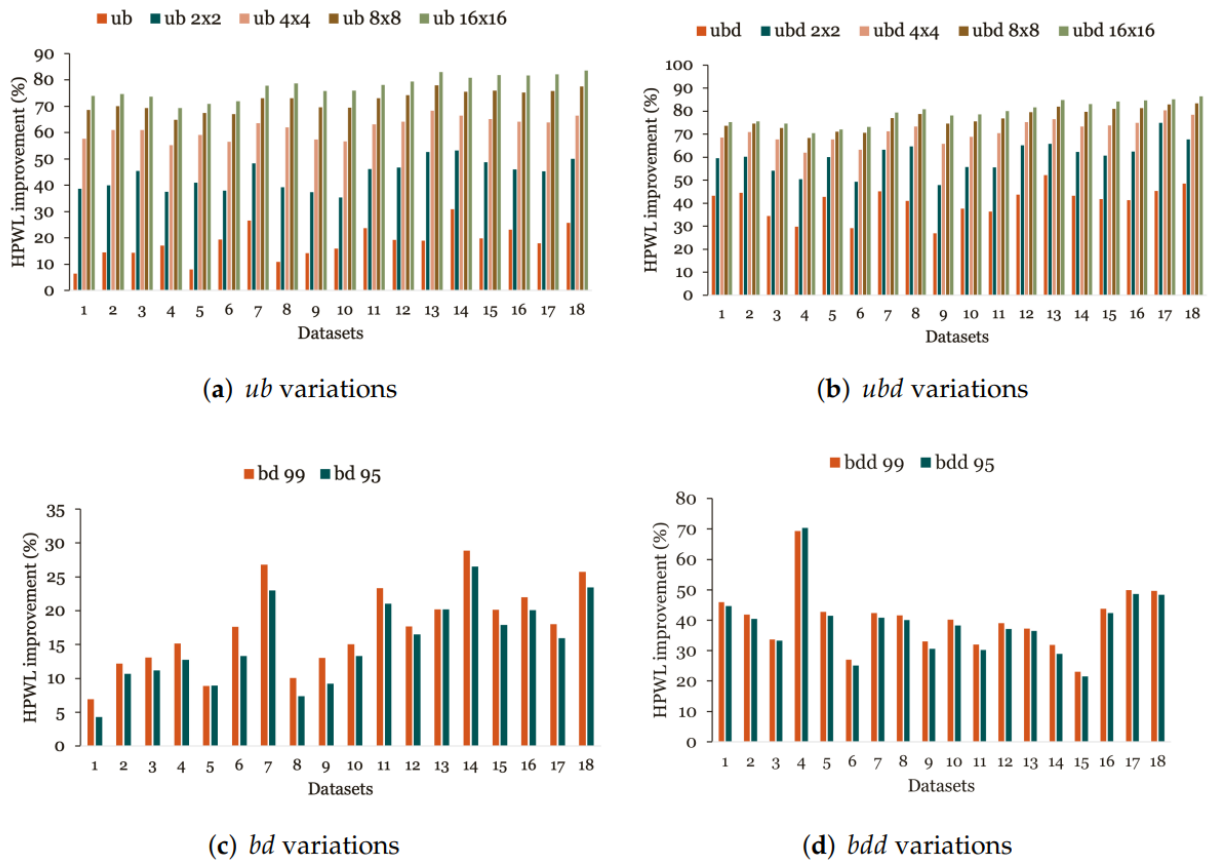
Τέλος θα πρέπει να σημειωθεί πως αν και οι αντίστοιχες παραλλαγές πλέγματος για τις `bd` και `bdd` περιπτώσεις έχουν δημιουργηθεί, τα αποτελέσματά τους φανερώνουν πως δεν είναι κατάλληλες για βελτιστοποίηση των κυκλωμάτων, πράγμα αρκετά λογικό αν σκεφτεί κανείς πως η τμηματοποίηση του κυκλώματος καθώς και η δέσμευση τοποθέτησης `cells` εντός των περιοχών του πλέγματος δεν ευνοούν τον περεταίρω περιορισμό της πυκνότητας αυτών των περιοχών.

## Αποτελέσματα & Σχολιασμός

Για τους σκοπούς της εξέτασης των νέων μεθόδων συγκριτικά με τις προηγούμενες παραλλαγές χρησιμοποιήθηκαν τα πρότυπα κυκλώματα της αναφοράς [11] και [28], συνολικά 22 σε αριθμό. Η μέτρηση οποιασδήποτε βελτίωσης υπολογίζεται με την εξίσωση που αναγράφεται στο προηγούμενο κεφάλαιο ενώ η υλοποίηση των αλγορίθμων έγινε με χρήση της γλώσσας `Python`.

Σχετικά με την μεθοδολογία της αξιολόγησης, γίνεται αρχικά χρήση του αλγορίθμου `mPL6` [12] για το βήμα του `Global Placement`, κι έπειτα για την βήμα του `Legalization` γίνεται χρήση των προτεινόμενων μεθοδολογιών για το πρώτο σετ πειραμάτων. Στην εικόνα 14

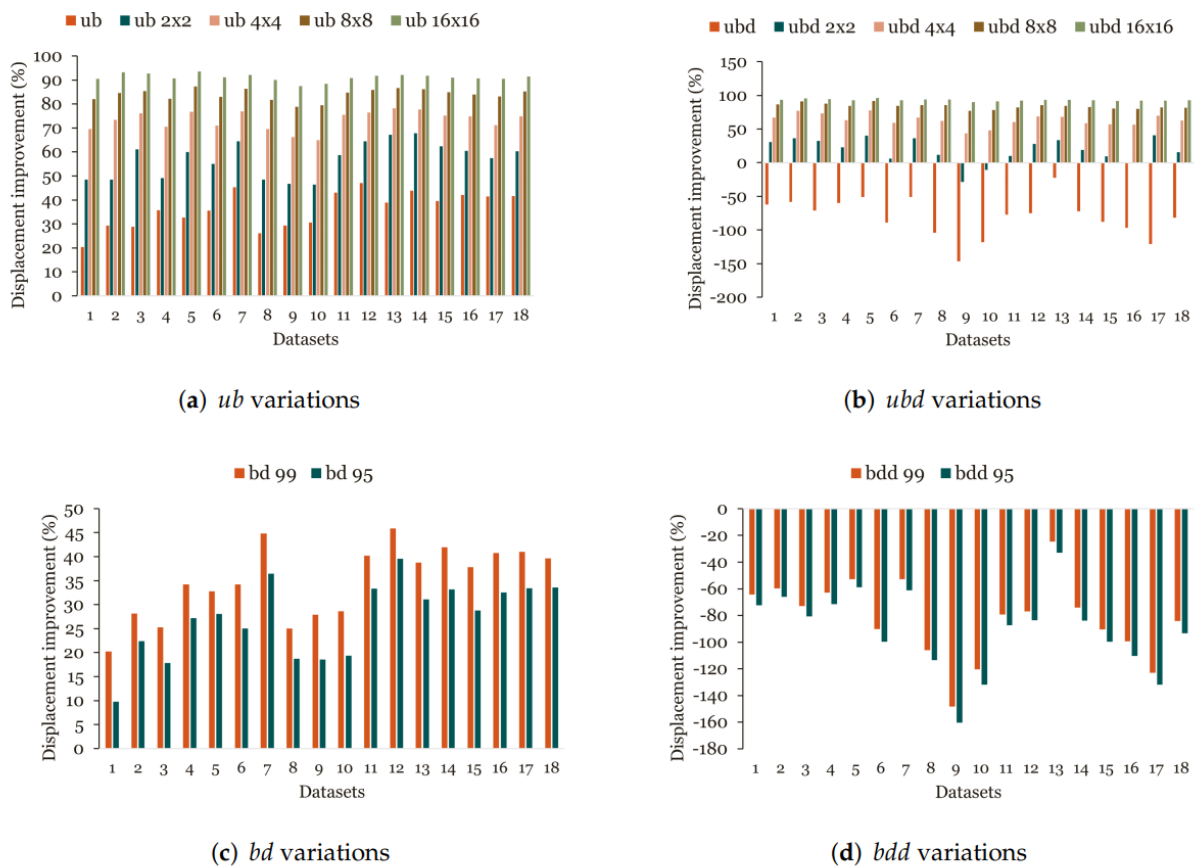
παρουσιάζονται τα γραφήματα σχετικά με την ποσοστιαία βελτίωση του hpwl. Κάθε υπογράφημα σημειώνει την απόδοση των παλαιών και των νέων μεθοδολογιών για σύγκριση. Σχετικά με το hpwl, παρατηρείται βελτίωση σε όλες τις περιπτώσεις, με τις ub, bd, ubd και bdd να παρουσιάζουν βελτίωση ανάλογη με τον μέγεθος του κυκλώματος αφού πραγματοποιούν αλλαγές σε όλη τη σχεδίαση. Η απόδοση των συγκεκριμένων παραλλαγών δεν θα σχολιαστεί περαιτέρω καθώς αυτό αποτελεί κομμάτι του προηγούμενου κεφαλαίου. Επιπρόσθετα, στην νέα δημοσίευση χωρίς κάποια αλλαγή δεν δύναται να παρουσιάσουν κάποιο διαφορετικό αποτέλεσμα.



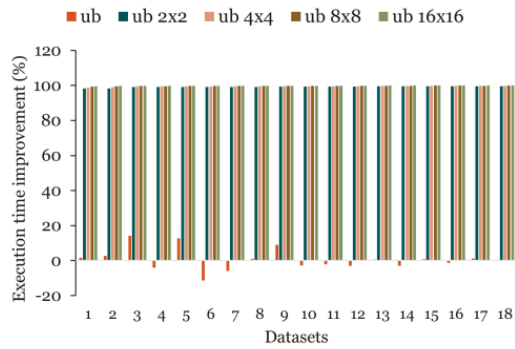
Εικόνα 14: Αποτελέσματα HWPL, Πηγή 25



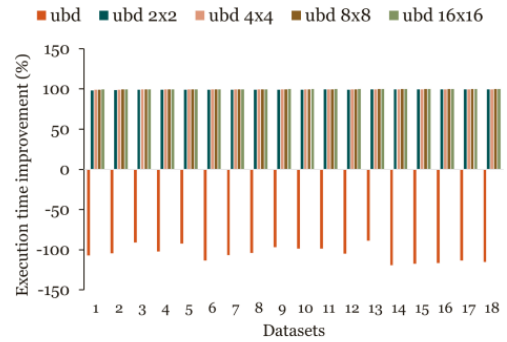
Οι νέες μεθοδολογίες (ubg, ubdg), οι οποίες παρουσιάζονται στα γραφήματα με τα μεγέθη του αντίστοιχου πλέγματος που σχηματίζουν, φαίνεται να σημειώνουν αρκετά καλύτερες επιδόσεις σχετικά με το χρόνο, πράγμα που αποδεικνύεται από τα γραφήματα της εικόνας 16. Ο διαχωρισμός της σχεδίασης σε τμήματα οδηγεί σε ένα αρκετά μικρότερο χώρο αναζήτησης καθώς το πλήθος των cells που εξετάζονται αλλά και οι διαθέσιμες νέες θέσεις είναι αρκετά μικρότερες σε πλήθος. Επιπρόσθετα ανάλογες θα είναι και οι βελτιώσεις σε displacement και hrwl (εικόνες 14, 15) αφού πλέον σε μικρότερο χώρο συνολικών κινήσεων πραγματοποιούνται και μικρότερες σε απόσταση μετατοπίσεις, γεγονός που έχει άμεση επιρροή και στις δύο τιμές.



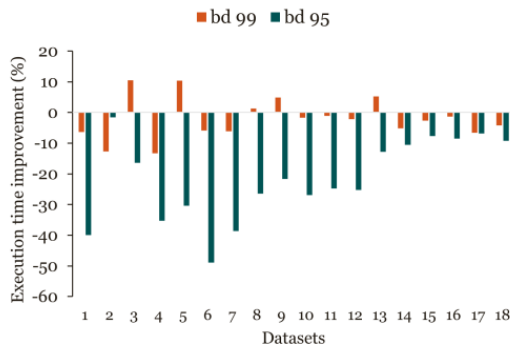
Εικόνα 15: Αποτελέσματα Displacement, Πηγή 25



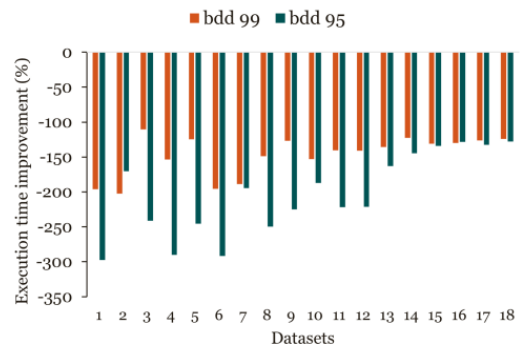
(a) *ub* variations



(b) *ubd* variations



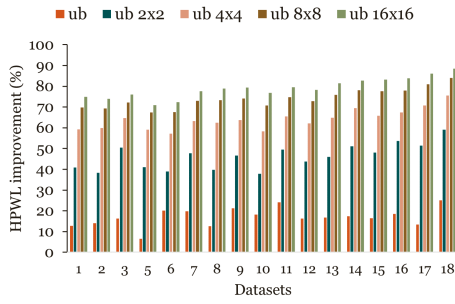
(c) *bd* variations



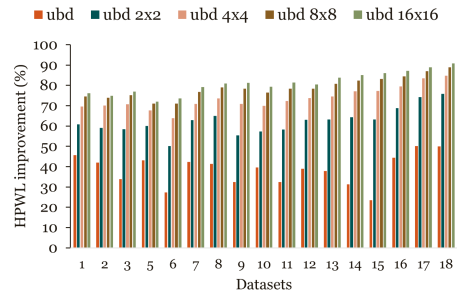
(d) *bdd* variations

Εικόνα 16: Αποτελέσματα χρόνου εκτέλεσης, Πηγή 25

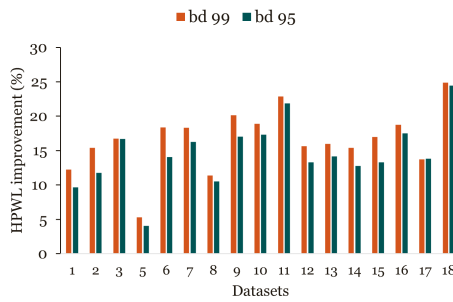
Για το δεύτερο σετ πειραμάτων έγινε χρήση του αλγορίθμου ePlace [29] για το βήμα του Global Placement, επαναλαμβάνοντας την ίδια διαδικασία για το βήμα του Legalization για περαιτέρω επιβεβαίωση των υπάρχοντων αποτελεσμάτων. Τα αποτελέσματα των γραφημάτων που περιέχονται στη δημοσίευση είναι αρκετά παρόμοια σε όλες τις περιπτώσεις και για όλες τις μετρικές, γεγονός που βεβαιώνει πως όλες οι παραλλαγές λειτουργούν πιο αποδοτικά από τον αρχικό αλγόριθμο [24] χωρίς να εξαρτώνται από το αποτέλεσμα του Global Placer που χρησιμοποιείται. Οι εικόνες 17, 18 και 19 παρουσιάζουν τα αντίστοιχα γραφήματα.



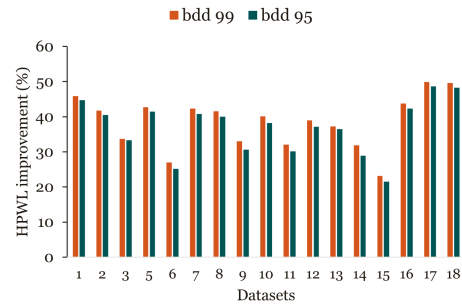
(a) *ub* variations



(b) *ubd* variations

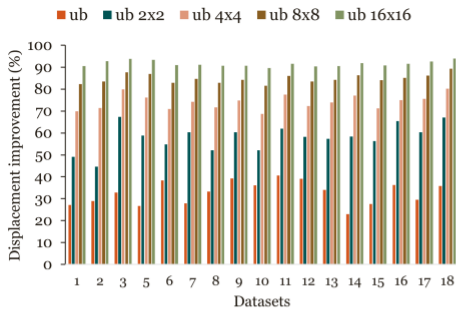


(c) *bd* variations

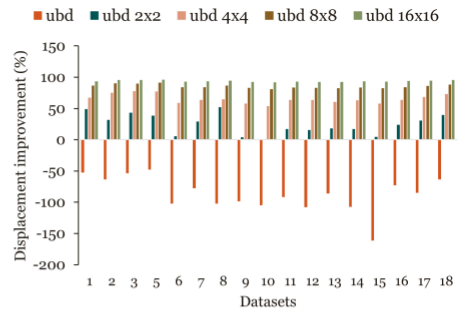


(d) *bdd* variations

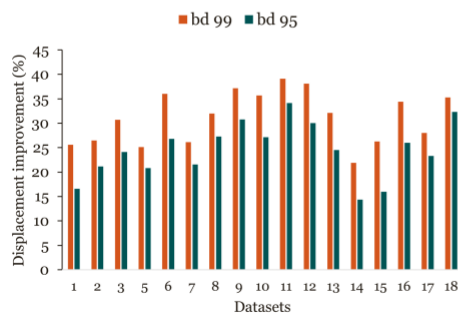
Εικόνα 17: Αποτελέσματα HPWL, Πηγή 25



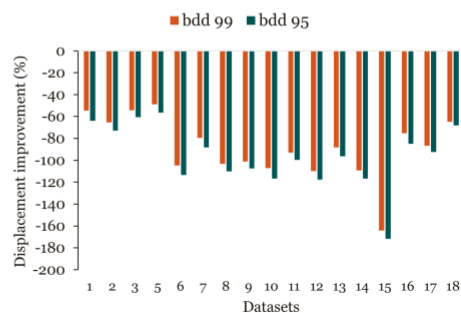
(a) *ub* variations



(b) *ubd* variations

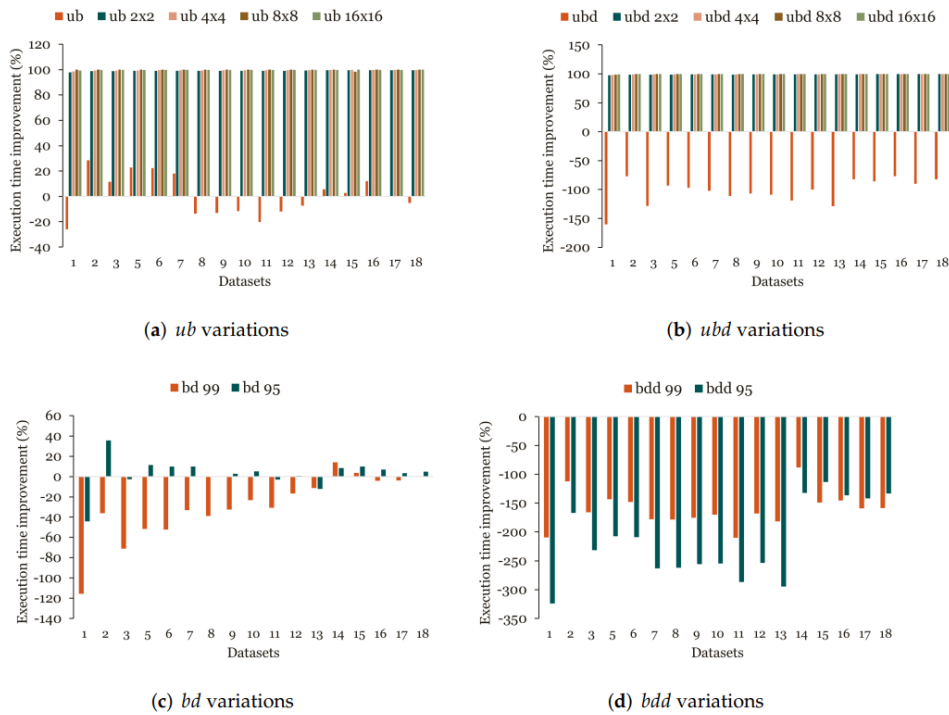


(c) *bd* variations



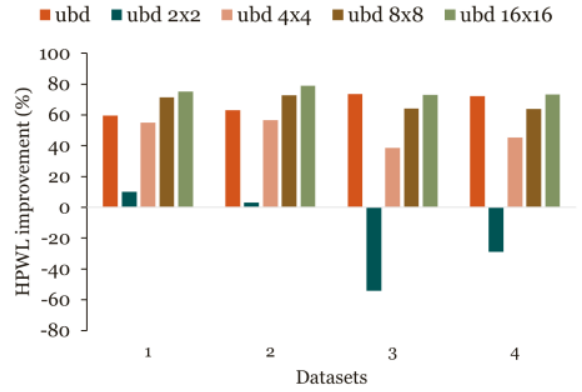
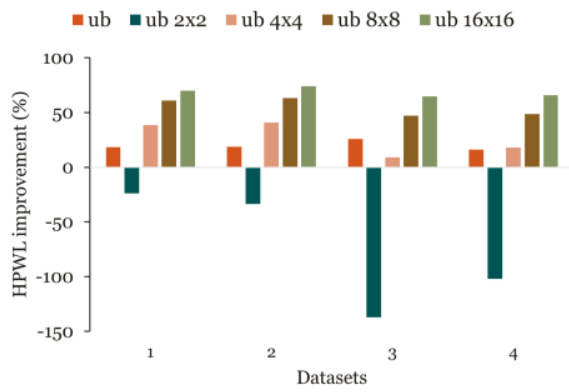
(d) *bdd* variations

Εικόνα 18: Αποτελέσματα Displacement, Πηγή 25



Εικόνα 19: Αποτελέσματα χρόνου εκτέλεσης, Πηγή 25

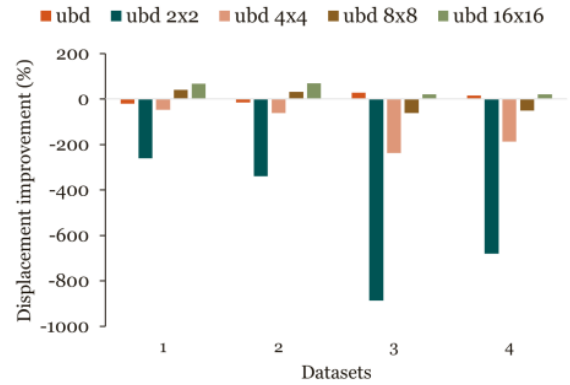
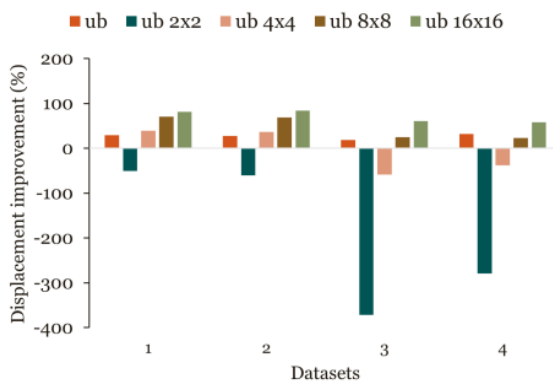
Το τελευταία σει πειραμάτων αφορά την εφαρμογή όλων των παραπάνω παραλλαγών σε Legalized κυκλώματα, πρακτικά εξετάζοντας την απόδοσή τους ως Detailed Placement διαδικασίες. Τα κυκλώματα της αναφοράς [28] έγιναν Globally Placed με τυχαίο τρόπο, ενώ χρησιμοποιήθηκε ο αλγόριθμος [24] για το Legalization κομμάτι τους, ο ίδιος αλγόριθμος που αποτελεί την βάση όλων των παραλλαγών. Να σημειωθεί πως οι συγκεκριμένες σχεδιάσεις περιέχουν και cells τα οποία θεωρούνται αμετακίνητα (macros) και εξαιτίας αυτού θα πρέπει να εξαιρεθούν των διαδικασιών. Το μέγεθος των συγκεκριμένων macros καθιστά τις παραλλαγές με τα όρια πυκνότητας αδύνατες στην εκτέλεση, καθώς κάποιες φορές καταλαμβάνουν ολόκληρες γραμμές της σχεδίασης. Οι παραλλαγές ub και ubd βελτιώνουν το hwpl όπως ήταν αναμενόμενο. Σχετικά τις εκδόσεις που χρησιμοποιούν πλέγμα, φαίνεται πως η 2x2 δεν μπορεί να παράγει καλά αποτελέσματα, κυρίως εξαιτίας του τρόπου που χωρίζεται σε πλέγμα η συγκεκριμένη σχεδίαση. Με την δημιουργία πλεγμάτων περισσότερων περιοχών φαίνεται πως τα αποτελέσματα είναι πάλι θετικά και βελτιώνουν το τελικό αποτέλεσμα των κυκλωμάτων (εικόνα 20). Παρόμοια φαίνεται πως είναι και τα αποτελέσματα σχετικά με το displacement και τον χρόνο εκτέλεσης των διαδικασιών (εικόνες 21, 22) με την γενικότερη εικόνα να δείχνει πως σε σχεδιάσεις αρκετά μεγαλύτερου μεγέθους είναι πιο λειτουργικά τα πλέγματα που δημιουργούν περισσότερες περιοχές σε σύνολο με τις περιπτώσεις 8x8 και 16x16 να παρουσιάζουν σχεδόν σε όλα βελτίωση.



(a) *ub* variations

(b) *ubd* variations

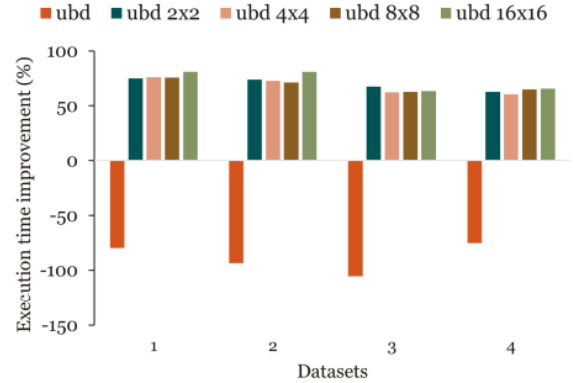
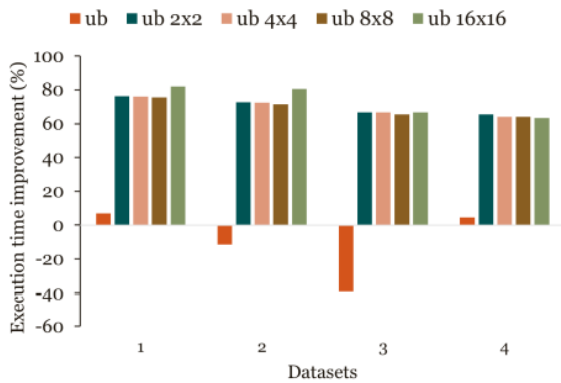
Εικόνα 20: Αποτελέσματα HPWL, Πηγή 25



(a) *ub* variations

(b) *ubd* variations

Εικόνα 21: Αποτελέσματα Displacement, Πηγή 25



(a) *ub* variations

(b) *ubd* variations

Εικόνα 22: Αποτελέσματα χρόνου εκτέλεσης, Πηγή 25

## ΣΥΝΟΨΗ

---

Η διαδικασία του Placement αποτελείται από τρία διακριτά βήματα, το Global Placement, το Legalization και το Detailed Placement. Υπάρχει μια αρκετά εκτενής βιβλιογραφία από αλγορίθμους κι ευριστικές διαδικασίες που λειτουργούν πάνω σε ένα, είτε και περισσότερα βήματα αυτής. Για την υλοποίηση όμως αυτών των αλγορίθμων συνήθως προηγείται μια διαδικασία ανάπτυξης και σχεδιασμού κατά την οποία αξιοποιούνται διάφορα πρότυπα κυκλώματα για τον πειραματισμό και την αξιολόγησή τους. Έτσι γίνεται αυτομάτως κατανοητό, πως υπάρχει η ανάγκη για εργαλεία που θα αυτοματοποιούν την διαδικασία της αναπαράστασης των συγκεκριμένων σχεδιάσεων σε μορφή συμβατή για την υλοποίηση οποιουδήποτε αλγορίθμου, αλλά και για την ικανοποίηση άλλων αναγκών που θα προκύψουν κατά την ανάπτυξη.

Ένα τέτοιο βοηθητικό εργαλείο είναι και το BFP, ένας δομικός επανασχεδιασμός του προκατόχου του PyPUT, ο οποίος φαίνεται συγκριτικά καλύτερος κι έχει ήδη αξιοποιηθεί σε ακαδημαϊκό έργο. Στόχος του είναι η ευκολία στη χρήση, την συντήρηση αλλά και την επέκταση ώστε να είναι πάντα συμβατό στις ανάγκες οποιουδήποτε χρήστη, κάτι το οποίο υποστηρίζουν και τα συγκριτικά αποτελέσματα που παρουσιάστηκαν.

Κατά τη φάση του Legalization, γίνεται χρήση αλγορίθμων οι οποίοι έχουν ως στόχο την επίλυση συγκεκριμένων αστοχιών οι οποίες προέκυψαν κατά το βήμα του Global Placement. Η μεθοδολογία που ακολουθείται όμως κατά την επίλυση συνήθως δημιουργεί αποκλίσεις από τις μετρικές που παρουσιάζει η αρχική χωροθέτηση δυσχεραίνοντας ποιοτικά το τελικό αποτέλεσμα. Οι παραλλαγές που προτείνονται στα κεφάλαια 2 και 3 αποτελούν μια προσπάθεια βελτίωσης των συγκεκριμένων σημείων ενός ήδη υπάρχοντος και δημοσιευμένου αλγορίθμου. Οι προτεινόμενες βελτιώσεις στην πρώτη τους μορφή παρουσιάζουν αρκετά μεγάλη βελτίωση σε συγκεκριμένα σημεία αλλά υστερούν κυρίως σε θέματα χρόνου εκτέλεσης. Οι μεταγενέστερες εκδόσεις αυτών, δείχνουν να βελτιώνονται στο συγκεκριμένο σημείο, έχοντας πλέον πετύχει αποτελεσματικά τον στόχο των προκατόχων τους.

- [1] Κρανάς, Γ., 2021, Υλοποίηση και Ανάπτυξη Αλγορίθμων Λεπτομερούς Χωροθέτησης Ολοκληρωμένων Κυκλωμάτων, Πτυχιακή εργασία, Πανεπιστήμιο Θεσσαλίας, Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Λαμία.
- [2] Kranas, G. K., Dadaliaris, A. N., Oikonomou, P., Floros, G., & Dossis, M. (2021, September). Redesign, Extensibility & Evaluation of a Placement Utilities Toolset. In *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)* (pp. 1-5). IEEE.
- [3] Kranas, G. K., Kouskouras, T. G., Dimitriadis, V., Dossis, M., Oikonomou, P., & Dadaliaris, A. N. (2020, September). A novel genetic algorithm for i/o pad planning retaining former cell positions. In *2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)* (pp. 1-5). IEEE.
- [4] Dadaliaris, A. N., Kranas, G. K., Oikonomou, P., & Dossis, M. (2021, September). Variations on a Connectivity-based Legalizer for Standard Cell Design. In *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)* (pp. 1-5). IEEE.
- [5] Dadaliaris, A., Kranas, G., Oikonomou, P., Floros, G., & Dossis, M. (2022). Exploiting Net Connectivity in Legalization and Detailed Placement Scenarios. *Information*, 13(5), 212.
- [6] A. Caldwell, A. Kahng, and I. Markov, "Gsrc bookshelf for vlsi cad algorithms," Available: HTTP: <http://vlsicad.cs.ucla.edu/GSR>
- [7] Kranas, G., Tsalamagkakakis, G. C., Oikonomou, P., & Dadaliaris, A. N. (2018, September). Pypot: Python-based placement utilities toolset. In *2018 South-Eastern European Design Automation, Computer Engineering, Computer Networks and Society Media Conference (SEEDA-CECNSM)* (pp. 1-4). IEEE.
- [8] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernandez del Río, M. Wiebe, P. Peterson, P. Gerard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, p. 357–362, 2020.
- [9] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [10] W. McKinney et al., "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, Austin, TX, 2010.
- [11] ISPD98 benchmark circuits, Available at: [http://vlsicad.eecs.umich.edu/BK/Slots/cache/www.public.iastate.edu/~nataraj/ISPD04\\_Benchmark.html](http://vlsicad.eecs.umich.edu/BK/Slots/cache/www.public.iastate.edu/~nataraj/ISPD04_Benchmark.html)
- [12] T. F. Chan, J. Cong, J. R. Shinnerl, K. Sze, and M. Xie, "mpl6: Enhanced multilevel mixed-size placement," in *Proceedings of the 2006 international symposium on Physical design*, pp. 212–214, 2006.
- [13] A. N. Dadaliaris, P. Oikonomou, M. G. Koziri, E. Nerantzaki, T. Loukopoulos, G. I. Stamoulis, et al., "A connectivity-based legalization scheme for standard cell placement," *Circuits and Systems*, vol. 8, no. 08, p. 191, 2017.
- [14] M. Lacchia, "radon 5.0.1," 2021.
- [15] T. J. McCabe, "A complexity measure," *IEEE Transactions on software Engineering*, no. 4, pp. 308–320, 1976
- [16] M. H. Halstead, "Elements of software science," 1977.
- [17] Dadaliaris, A. N., Kranas, G. K., Oikonomou, P., & Dossis, M. (2021, September). Variations on a Connectivity-based Legalizer for Standard Cell Design. In *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)* (pp. 1-5). IEEE.
- [18] D. Hill, "Method and system for high speed detailed placement of cells within an integrated circuit design," Apr. 9 2002. US Patent 6,370,673.
- [19] P. Spindler, U. Schlichtmann, and F. M. Johannes, "Abacus: Fast legalization of standard cell circuits with minimal movement," in *Proceedings of the 2008 international symposium on Physical design*, pp. 47–53, 2008.
- [20] A. N. Dadaliaris, P. Oikonomou, M. G. Koziri, E. Nerantzaki, Y. Hatzaras, D. Garyfallou, T. Loukopoulos, and G. I. Stamoulis, "Heuristics to augment the performance of tetris legalization: making a fast but inferior method competitive," *Journal of Low Power Electronics*, vol. 13, no. 2, pp. 220–230, 2017.

- [21] L. He, A. B. Kahng, K. H. Tam, and J. Xiong, "Variability-driven considerations in the design of integrated-circuit global interconnects," in Proc. 21th Intl. VLSI Multilevel Interconnection (VMIC) Conf, pp. 214–221, Citeseer, 2004.
- [22] S. Chou and T.-Y. Ho, "Oal: An obstacle-aware legalization in standard cell placement with displacement minimization," in 2009 IEEE International SOC Conference (SOCC), pp. 329–332, IEEE, 2009.
- [23] P. Oikonomou, M. G. Koziri, A. N. Dadaliaris, T. Loukopoulos, and G. I. Stamoulis, "Domocus: Lock free parallel legalization in standard cell placement," in 2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST), pp. 1–4, IEEE, 2017.
- [24] A. N. Dadaliaris, P. Oikonomou, M. G. Koziri, E. Nerantzaki, T. Loukopoulos, G. I. Stamoulis, et al., "A connectivity-based legalization scheme for standard cell placement," *Circuits and Systems*, vol. 8, no. 08, p. 191, 2017.
- [25] Dadaliaris, A., Kranas, G., Oikonomou, P., Floros, G., & Dossis, M. (2022). Exploiting Net Connectivity in Legalization and Detailed Placement Scenarios. *Information*, 13(5), 212.
- [26] Garey, M.R.; Johnson, D.S. *Computers and Intractability*; Freeman: San Francisco, CA, USA, 1979; Volume 174.
- [27] Garey, M.R.; Johnson, D.S.; Stockmeyer, L. Some simplified NP-complete problems. In Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, 30 April–2 May 1974; pp. 47–63
- [28] Nam, G.J.; Alpert, C.J.; Villarrubia, P.; Winter, B.; Yildiz, M. The ISPD2005 placement contest and benchmark suite. In Proceedings of the 2005 International Symposium on Physical Design, San Francisco, CA, USA, 3–6 April 2005; pp. 216–220.
- [29] Lu, J.; Chen, P.; Chang, C.C.; Sha, L.; Dennis, J.; Huang, H.; Teng, C.C.; Cheng, C.K. ePlace: Electrostatics based placement using Nesterov's method. In Proceedings of the 51st ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 1–5 June 2014; pp. 1–6.
- [30] C. Sechen and A. Sangiovanni-Vincentelli. TimberWolf3.2: A New Standard Cell Placement and Global Routing Package. In DAC, pages 432–439, 1986
- [31] Caldwell, A.E.; Kahng, A.B.; Markov, I.L. (June 2000). "Can recursive bisection alone produce routable placements? ". Proceedings of the 37th Design Automation Conference. pp. 477–482.
- [32] J. M. Kleinhans, G. Sigl, F. M. Johannes, and K. J. Antreich, "GORDIAN: VLSI placement by quadratic programming and slicing optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 356-365, March 1991
- [33] H. Eisenmann and F. M. Johannes. Generic Global Placement and Floorplanning. In DAC, pages 269–274, 1998.
- [34] P. Spindler, U. Schlichtmann, and F. M. Johannes. Kraftwerk2 - A Fast Force-Directed Quadratic Placement Approach Using an Accurate Net Model. *IEEE TCAD*, 27(8):1398–1411, 2008.
- [35] N. Viswanathan, and C. N. Chu, "FastPlace: efficient analytical placement using cell shifting, iterative local refinement, and a hybrid net model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 722-733, May 2005
- [36] M. C. Kim, D. J. Lee, and I. L. Markov, "SimPL: An effective placement algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 50-60, January 2012
- [37] W. C. Naylor, R. Donnelly, and L. Sha. Non-Linear Optimization System and Method for Wire Length and Delay Optimization for an Automatic Electric Circuit Placer. In US Patent 6301693, 2001.
- [38] A. B. Kahng, S. Reda and Q. Wang, "Architecture and Details of a High Quality, Large-Scale Analytical Placer", In ICCAD 2005, pp. 891-898
- [39] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang. NTUPlace3: An Analytical Placer for Large-Scale Mixed-Size Designs with Preplaced Blocks and Density Constraint. *IEEE TCAD*, 27(7):1228–1240, 2008