

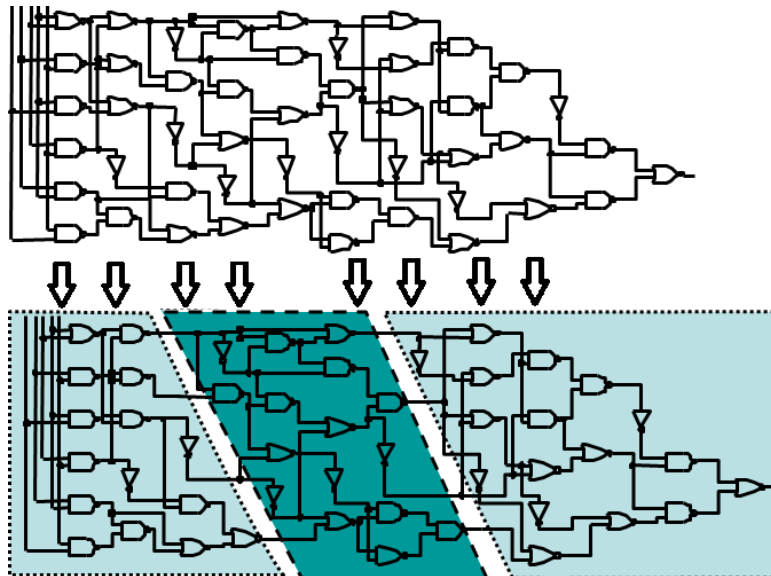


ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΚΑΤΑΜΕΡΙΣΜΟΥ ΕΙΓ



Επιβλέπων Καθηγητής: Σταμούλης Γεώργιος

Επιμέλεια: Σπανέλλης Δημήτριος

A.E.M:721

Βόλος 2013

Στην οικογένεια & στους φίλους μου

Ευχαριστίες

Με την περάτωση της παρούσας εργασίας, θα ήθελα να ευχαριστήσω θερμά τους επιβλέποντες της διπλωματικής εργασίας κ. Γεώργιο Σταμούλη και κ. Νέστορα Ευμορφόπουλο για την εμπιστοσύνη που επέδειξαν στο πρόσωπό μου, την άριστη συνεργασία, την συνεχή καθοδήγηση και τις ουσιώδεις υποδείξεις και παρεμβάσεις, τις οξυδερκείς συμβουλές και παρατηρήσεις, την σημαντική συμβολή, τις καίριες παρεμβάσεις, τα κατάλληλα και ουσιώδη κάθε φορά σχόλια, και γενικότερα για την εμπύχωση και υποστήριξή τους που διευκόλυναν την εκπόνηση της πτυχιακής εργασίας.

Επίσης θα ήθελα να εκφράσω την εκτίμησή μου σε όλα τα παιδιά με τα οποία συνεργάστηκα όλα αυτά τα χρόνια των προπτυχιακών σπουδών μου. Ιδιαίτερα θα ήθελα να ευχαριστήσω τον διδακτορικό φοιτητή Αντώνη Δαδαλιάρη και Κωσταντή Νταλούκα για την συνεργασία που είχαμε στην διπλωματική μου εργασία.

Τέλος, οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου, στην κοπέλα μου και στους φίλους μου για την αμέριστη υποστήριξη και την ανεκτίμητη βοήθεια που μου παρείχαν τόσο κατά την διάρκεια των σπουδών μου όσο και κατά την εκπόνηση της διπλωματικής εργασίας. Η άψογη συνεργασία μου μαζί τους οδήγησε στην δημιουργία ενός πολύ ευχάριστου κλίματος με αποτέλεσμα η εκπόνηση της πτυχιακής μου εργασίας να αποτελέσει για μένα μία καταπληκτική διαδικασία και εμπειρία που θα μου μείνει αξέχαστη!

Δημήτριος Σπανέλλης
Βόλος, 2013

Περιεχόμενα

Εισαγωγή.....	7
Περιγραφή του προβλήματος.....	8
Σκοπός της εργασίας.....	12
Αλγοριθμική μορφή του EIG.....	15
Η υλοποίηση του αλγορίθμου.....	16
Ο αλγόριθμος κατάτμησης με λεπτομερή ανάλυση.....	20
Αποτελέσματα.....	35
Συμπεράσματα.....	63
Βιβλιογραφία.....	64

Ευρετήριο Εικόνων

1.Κύκλωμα παραδείγματος.....	16
2.Γράφος κυκλώματος.....	17
3.Πίνακας συνδέσεων.....	18
4.Πίνακας αθροισμάτων.....	19
5.Πίνακας διαφορών.....	20
6.Ταξινόμηση κόμβων.....	22
7.Διαχωρισμός κόμβων.....	23

Ευρετήριο πινάκων

1.Κυκλώματα ISCAS 89 που χρησιμοποιήθηκαν.....	13
2.Αποτελέσματα κυκλώματος S208.....	31
3. Αποτελέσματα κυκλώματος S298.....	32
4. Αποτελέσματα κυκλώματος S344.....	33
5. Αποτελέσματα κυκλώματος S349.....	34
6. Αποτελέσματα κυκλώματος S382.....	35
7. Αποτελέσματα κυκλώματος S386.....	36
8. Αποτελέσματα κυκλώματος S400.....	37
9. Αποτελέσματα κυκλώματος S420.....	38
10. Αποτελέσματα κυκλώματος S440.....	39
11. Αποτελέσματα κυκλώματος S510.....	40
12. Αποτελέσματα κυκλώματος S526.....	41
13. Αποτελέσματα κυκλώματος S641.....	42
14. Αποτελέσματα κυκλώματος S713.....	43
15. Αποτελέσματα κυκλώματος S820.....	44
16. Αποτελέσματα κυκλώματος S832.....	45
17. Αποτελέσματα κυκλώματος S838.....	46
18. Αποτελέσματα κυκλώματος S953.....	47
19. Αποτελέσματα κυκλώματος S1196.....	48
20. Αποτελέσματα κυκλώματος S1238.....	49
21. Αποτελέσματα κυκλώματος S1423.....	50
22. Αποτελέσματα κυκλώματος S1488.....	51
23. Αποτελέσματα κυκλώματος S1494.....	52
24. Αποτελέσματα κυκλώματος S5378.....	53
25. Αποτελέσματα κυκλώματος S9234.....	54
26. Αποτελέσματα κυκλώματος S13207.....	55
27. Αποτελέσματα κυκλώματος S15850.....	56
28. Αποτελέσματα κυκλώματος S35932.....	57

Εισαγωγή

Το σύγχρονο ολοκληρωμένο κύκλωμα είναι μεταξύ των πιό σύνθετων προϊόντων εφαρμοσμένης μηχανικής που δημιουργήθηκε ποτέ από την ανθρωπότητα.

Από την εφεύρεσή του, περίπου τέσσερις δεκαετίες πριν (την δεκαετία του '60), ο αριθμός των τρανζίστορ ανά ολοκληρωμένο κύκλωμα διπλασιάζεται κάθε δύο έτη, σύμφωνα με το διάσημο νόμο του Moore. Ο σχεδιασμός των πολύ μεγάλης κλίμακας ολοκληρωμένων κυκλωμάτων (VLSI), εντούτοις, έχει γίνει μεγάλη πρόκληση, λαμβάνοντας υποψηφίους τις εκατοντάδες των σχεδιαστών και της εκτενούς χρήσης εργαλείων σχεδιασμού με την βοήθεια υπολογιστή (CAD). Ένα από τα παλαιότερα, αλλά σημαντικότερα προβλήματα CAD για τα κυκλώματα VLSI είναι η φυσική αυτοματοποίηση σχεδιασμού, όπου κάποιος πρέπει να υπολογίσει το καλύτερο φυσικό σχεδιάγραμμα των εκατομμυρίων δισεκατομμυρίων συστατικών του κυκλώματος σε μια μικροσκοπική επιφάνεια πυριτίου (λιγότερη από 5 cm^2), μια διαδικασία παρόμοια με την επίλυση ενός ιδιαίτερα σύνθετου παζλ με κομμάτια νανο-κλίμακας. Ο αρχικός στόχος ήταν η ελαχιστοποίηση περιοχής, για να προκύψει το συμπαγέστερο σχεδιάγραμμα. Πλέον η εστίαση δίνεται στην βελτιστοποίηση της απόδοσης κυκλωμάτων, και στην εξοικονόμηση ενέργειας καθώς η διάταξη των τμημάτων ενός κυκλώματος καθορίζει τις συνδέσεις και τα καλώδια που απαιτούνται για να συνδεθούν αυτά τα τμήματα μεταξύ τους, και με την σειρά τους οι συνδέσεις και τα καλώδια παίζουν σημαντικό ρόλο στην απόδοση και στην κατανάλωση ενέργειας ενός κυκλώματος. Μετά από σαράντα έτη, η αυτοματοποίηση σχεδιασμού συστημάτων VLSI παραμένει ακόμα ένας τομέας που ενδείκνυται και για έρευνα αλλά και για εμπορική ανάπτυξη. Πάνω από εκατό έρευνες παρουσιάζονται κάθε έτος, με νέες ιδέες να εξετάζουν τη συνεχώς αυξανόμενη πολυπλοκότητα και τους περιορισμούς κάθε σχεδιασμού. Καθώς η τεχνολογία VLSI εισχωρεί στην περιοχή των νανομέτρων, τα φαινόμενα της φυσικής πρέπει να εξετάζονται σε κάθε στάδια του σχεδιασμού.

Στο παρελθόν, το κοινό των βιβλίων σχεδιασμού ήταν περιορισμένο αρχικά στους σπουδαστές και τους ερευνητές που ενδιαφέρονταν για την ανάπτυξη των εργαλείων CAD για φυσικά σχεδιαγράμματα. Αυτό έχει αλλάξει. Σήμερα, υπάρχει μια ισχυρή απαίτηση γνώσης για την ανάπτυξη εργαλείων CAD σε ολόκληρο τον κόσμο. Βασικές τεχνικές σχεδιασμού μπορούν να τροποποιηθούν για να παράγουν εκτιμήσεις φυσικών φαινομένων ή για να χρησιμοποιηθούν για την ελαχιστοποίηση κατανάλωσης ενέργειας.

Περιγραφή του προβλήματος

Η αυτοματοποίηση της φυσικής σχεδίασης συστημάτων VLSI περιλαμβάνει κάποια στάδια από τα οποία περνάει το κάθε κύκλωμα μέχρι να φτάσει στην τελική του μορφή.

Τα στάδια αυτά είναι:

1.Συσταδοποίηση

Δοθέντος ενός κυκλώματος, ο στόχος της συσταδοποίησης του είναι να ομαδοποιηθούν οι πύλες σε συστάδες και να δημιουργηθεί ένα δίκτυο από συστάδες. Το μέγεθος της συστάδας από άποψη αριθμού συστάδων και αριθμού συνδέσεων θα είναι μικρότερο από αυτό του αρχικού κυκλώματος. Ένας χαρακτηριστικός στόχος είναι να ελαχιστοποιηθεί ο αριθμός συνδέσεων, ο μέγιστος αριθμός συνδέσεων σε οποιαδήποτε μονοπάτι, ενώ παράλληλα χαρακτηριστικοί περιορισμοί είναι το μέγιστο μέγεθος συστάδων, ο μέγιστος αριθμός από τις εξωτερικές συνδέσεις μιας συστάδας, κ.λ.π. Ο αριθμός συστάδων που πρέπει να δημιουργηθούν δεν διευκρινίζεται, και η ισορροπία έκτασης μεταξύ των συστάδων δεν είναι συνήθως συγκεκριμένη. Η συσταδοποίηση του κυκλώματος αποτελεί συνήθως την προεπεξεργασία για το διαχωρισμό του κυκλώματος και της χωροθέτησης για την μείωση της πολυπλοκότητας του προβλήματος.

Ορισμένοι από τους σημαντικότερους αλγορίθμους για την υλοποίηση της συσταδοποίησης είναι οι ακόλουθοι:

- α. Rajaraman and Wong
- β. Flow Map
- γ. Multi-level Coarsening

Οι πρώτοι δύο είναι βασισμένοι στην καθυστέρηση όπου η τελευταία του μεγαλύτερου μονοπατιού στα συσταδοποιημένα κυκλώματα ελαχιστοποιείται από ένα συγκεκριμένο μοντέλο καθυστέρησης κόμβων/ακμών. Ο τελευταίος είναι βασισμένος στο μέγεθος κοπής όπου το μέγεθος των συνδέσεων της κάθε συστάδας ελαχιστοποιείται.

2. Διαχωρισμός

Δοθέντος ενός κυκλώματος, ο στόχος είναι ο διαχωρισμός του σε K κατά προσέγγιση ίσα τμήματα. Ο απώτερος στόχος είναι η ελαχιστοποίηση του αριθμού των δικτύων που συνδέουν τις πύλες στο κάθε τμήμα, ο οποίος χαρακτηριστικά αποκαλείτε και μέγεθος αποκοπής. Άλλοι στόχοι είναι η ελαχιστοποίηση της καθυστέρησης στο κρίσιμο μονοπάτι και η ελαχιστοποίηση της συνολικής κατανάλωσης ισχύος.

Ορισμένοι από τους σημαντικότερους αλγόριθμους για την υλοποίηση του διαχωρισμού είναι οι ακόλουθοι:

- α. Kernighan and Lin
- β. Fiduccia and Mattheyses
- γ. Hagen and Kahng
- δ. Yang and Wong

Οι δύο πρώτοι αλγόριθμοι είναι βασισμένοι στην κίνηση των πυλών, όπου οι πύλες μετακινούνται από το ένα τμήμα στο άλλο όταν αποφασιστεί ο καλύτερος τρόπος διαχωρισμού. Ο τρίτος βασίζεται στο υπολογισμό ιδιοτιμών και ιδιοδιανυσμάτων των πινάκων ενός κυκλώματος. Ενώ ο τελευταίος αλγόριθμος υπολογίζει τη μέγιστη ροή για τον κάθε διαχωρισμό του κυκλώματος με σκοπό την επιλογή αυτού με την ελάχιστη.

3. Σχεδιασμός Βάσης

Τα δεδομένα εισόδου στο πρόβλημα της σχεδίασης βάσης είναι το σύνολο των μπλοκ (σταθερών ή μεταβλητών) και η λίστα περιγραφής του κυκλώματος. Ο στόχος στο στάδιο αυτό είναι να βρεθεί η θέση του κάθε μπλοκ έτσι ώστε να μην επικαλύπτεται το ένα με το άλλο. Σε περίπτωση μεταβλητών μπλοκ να αποφασιστεί το μέγεθος τους. Στόχοι είναι η ελαχιστοποίηση της περιοχής, του μεγέθους καλωδίων, της ανάπτυξης θερμότητας και του θορύβου.

Ορισμένοι από τους σημαντικότερους αλγόριθμους για την υλοποίηση του σχεδιασμού της βάσης είναι οι ακόλουθοι:

- α. Stockmeyer
- β. Normalized polish expression
- γ. ILP-based floor planning
- δ. Sequence pair representation

4. Χωροθέτηση

Η χωροθέτηση κυκλωμάτων είναι η διαδικασία που αποφασίζεται η θέση κάθε πύλης στο κύκλωμα. Οι στόχοι της είναι η ελαχιστοποίηση του μεγέθους των καλωδίων διασύνδεσης, του συγχρονισμού, και της συμφόρησης. Πρόσφατα, η ελαχιστοποίηση της θερμότητας, η κατανάλωση ισχύος, και τα ζητήματα θορύβου παροχής ηλεκτρικού ρεύματος αποτέλεσαν θέμα κατά την διαδικασία. Προκειμένου να είναι διαχειρίσιμα κυκλώματα μεγάλης κλίμακας, η χωροθέτηση γίνεται συνήθως σε δύο βήματα: στη γενική χωροθέτηση και στη λεπτομερή χωροθέτηση όπως στην περίπτωση με τη δρομολόγηση.

Ευρέως χρησιμοποιούμενοι αλγόριθμοι χωροθέτησης είναι οι εξής:

- α. Mincut
- β. Gordian
- γ. TimberWolf

5. Δρομολόγηση Steiner

Δοθέντος ενός συνόλου από σημεία P σε ένα 2D πλάνο, το πρόβλημα του δέντρου Steiner αναζητεί ένα σύνολο πρόσθετων σημείων S έτσι ώστε το μέγεθος καλωδίωσης του ελάχιστα εκτεινόμενου δέντρου (MST) του συνόλου $P \cup S$ να είναι ελάχιστο. Οι πρόσθετοι στόχοι της διαδικασίας είναι η ελαχιστοποίηση μετρικών όπως η ακτίνα και η καθυστέρηση.

Ορισμένοι από τους σημαντικότερους αλγορίθμους για την υλοποίηση της δρομολόγησης Steiner είναι οι ακόλουθοι:

- α. L-shaped Steiner routing
- β. 1-Steiner
- γ. BPRIM and BRBC
- δ. A-tree
- ε. ERT and SERT

6. Δρομολόγηση Multiunit

Δοθέντος μιας γραφικής παράστασης $G(V, E)$ (όπου G ο γράφος, V το σύνολο των κόμβων του και E το σύνολο των ακμών του) που αντιπροσωπεύει τους πόρους δρομολόγησης όπως τα κανάλια και τις περιοχές και ένα κύκλωμα NL που αποτελείται από πολλαπλά δίκτυα, όπου κάθε ένα από αυτά είναι ένα υποσύνολο κορυφών του V , ο στόχος της δρομολόγησης είναι να κατασκευάσει δέντρα δρομολόγησης όλων των δικτύων του NL έτσι ώστε ο περιορισμός χωρητικότητας που υπάρχει σε κάθε ακμή του E να είναι ικανοποιημένη. Ο στόχος είναι να ελαχιστοποιηθεί το συνολικό μήκος καλωδίων, η συμφόρηση της δρομολόγησης.

Ορισμένοι από τους σημαντικότερους αλγορίθμους για την υλοποίηση της δρομολόγησης Multiunit είναι οι ακόλουθοι:

- α. Steiner min-max tree
- β. Multi-commodity flow routing
- γ. Iterative deletion
- δ. Yoshimura and Kuh

Ο πρώτος δρομολογητής είναι ένας σειριακός δρομολογητής, όπου τα δίκτυα δρομολογούνται ένα ένα. Οι άλλοι δρομολογούν όλα τα δίκτυα ταυτόχρονα. Οι πρώτοι τρεις δρομολογητές είναι γενικοί δρομολογητές. Ο τελευταίος είναι ειδικός δρομολογητής.

Σκοπός της εργασίας

Στην παρούσα διπλωματική εργασία μελετάται ο αλγόριθμος EIG των Hagen and Kahng, ο οποίος χρησιμοποιείται στο στάδιο του διαχωρισμού. Έχοντας ήδη την υλοποίηση του αλγορίθμου Kernighan and Lin για το στάδιο του διαχωρισμού, και αποτελέσματα για συγκεκριμένα κυκλώματα, αυτό με το οποίο ασχοληθήκαμε στην παρούσα εργασία είναι η ανάλυση και η υλοποίηση του αλγορίθμου των Hagen and Kahng με σκοπό την σύγκριση των αποτελεσμάτων με αυτά του αλγορίθμου Kernighan and Lin και την εξαγωγή συμπερασμάτων σχετικά με τα αποτελέσματα και με την βελτίωση ή όχι της διαδικασίας της αυτοματοποιημένης σχεδίασης συστημάτων VLSI μεταξύ των δύο αυτών αλγορίθμων διαχωρισμού.

Συγκεκριμένα, μετά την υλοποίηση του αλγορίθμου EIG των Hagen and Kahng, αυτός και ο αλγόριθμος Kernighan and Lin εξετάστηκαν όσον αφορά την αποτελεσματικότητά τους και την βελτίωση της διαδικασίας της αυτοματοποιημένης σχεδίασης συστημάτων VLSI σε ένα σύνολο κυκλωμάτων που αναφέρονται παρακάτω.

Έπίσης άλλος ένας σκοπός της εργασίας είναι και η εξαγωγή συμπερασμάτων για την αποδοτικότητα και την σύγκριση των δύο αυτών αλγορίθμων διαχωρισμού

Στην εργασία, λοιπόν, χρησιμοποιήθηκαν τα κυκλώματα ISCAS 89 προκειμένου να τρέξουμε τον αλγόριθμό μας και να εξάγουμε τα αντίστοιχα αποτελέσματα του καθενός κυκλώματος. Στον πίνακα που ακολουθεί γίνεται μία πλήρης αναφορά στα χαρακτηριστικά των κυκλωμάτων που χρησιμοποιήθηκαν.

Πρέπει να σημειωθεί ότι το γράμμα s στην ονομασία των κυκλωμάτων σημαίνει ότι τα κυκλώματα είναι σύγχρονα ακολουθιακά (synchronous sequential) και ο αριθμός που ακολουθεί το γράμμα s αντιπροσωπεύει τον αριθμό των γραμμών διασύνδεσης μεταξύ των απλών του κυκλωμάτων .

Circuit name	#of Primary Inputs	#of Primary Outputs	#of D-type flip-flops	#of AND/OR/NOT Gates
s208	11	2	8	96
s298	3	6	14	119
s344	9	11	15	160
s349	9	11	15	161
s382	3	6	21	158
s386	7	7	6	159
s400	3	6	21	162
s420	19	2	16	196
s444	3	6	21	181
s510	19	7	6	211
s526	3	6	21	193
s641	35	24	19	379
s713	35	23	19	393
s820	18	19	5	289
s832	18	19	5	287
s953	16	23	29	395
s1196	14	14	18	529
s1238	14	14	18	508
s1488	8	19	6	653

1.Κυκλώματα ISCAS 89 που χρησιμοποιήθηκαν

Αυτό που σήμερα γνωρίζουμε για το καθένα από τα παραπάνω κυκλώματα είναι:

- Το κύκλωμα s349 είναι ένας 4-bit πολλαπλασιαστής.
- Τα κυκλώματα s298, s400, s444 και s526 είναι φωτεινοί σηματοδότες ελέγχου κυκλοφορίας.
- Τα κυκλώματα s386, s510 και s953 είναι ελεγκτές, οι οποίοι συντίθενται από ένα υψηλό επίπεδο περιγραφής.
- Το κύκλωμα s1238 είναι ένα συνδυαστικό κύκλωμα με τυχαία παρεμβαλλόμενα flip-flops.
- Τα κυκλώματα s208, s420 και s838 είναι ψηφιακοί κλασματικοί πολλαπλασιαστές, που συντίθενται ιεραρχικά από ένα υψηλό επίπεδο περιγραφής.
- Τα κυκλώματα s298, s208, s713, s641 και s832, βασίζονται σε συσκευές PLD.
- Τα κυκλώματα s344, s382, s641, s820, s1196 και s1488 έχουν συντεθεί εκ νέου από τα κυκλώματα s349, s400, s526, s713, s832 και s1238, μετά την αφαίρεση όλων των απολύσεων σε πλήρη λειτουργία σάρωσης.

Αλγοριθμική Μορφή του EIG

Ο αλγόριθμος κατάτμησης των Hagen and Kahng που αναλύουμε μπορεί να γραφεί σε αλγοριθμική μορφή ως εξής:

1. Δημιούργησε Adjacency Matrix
2. Δημιούργησε Degree Matrix
3. Δημιούργησε Laplacian Matrix
4. Βρες δεύτερη μικρότερη ιδιοτιμή και αντίστοιχο ιδιοδιάνυσμα
5. Ταξινόμησε τους κόμβους
6. Για $n-1$ φορές:
 Προσθαφαίρεσε κόμβους
7. Βρες καλή λύση
8. Πιστοποίησε ότι $c \geq \lambda/n$

όπου:

- c είναι το ratio cut cost
- λ είναι η δεύτερη μικρότερη ιδιοτιμή
- n είναι ο αριθμός των κόμβων

Η υλοποίηση του αλγορίθμου

Η επιλογή της γλώσσας προγραμματισμού για την υλοποίηση του αλγορίθμου

Η υλοποίηση του αλγορίθμου έγινε χρησιμοποιώντας τη γλώσσα προγραμματισμού C. Οι λόγοι που προτιμήθηκε η συγκεκριμένη γλώσσα προγραμματισμού είναι διάφοροι.

Ο κυριότερος από αυτούς είναι ότι η γλώσσα C είναι προσανατολισμένη προς τις ανάγκες του προγραμματιστή, ο οποίος και έχει άμεση πρόσβαση στο υλικό. Με τη C έχουμε τη σπουδαία δυνατότητα να μπορούμε να χειριζόμαστε μεμονωμένα τα δυαδικά ψηφία (bits) της μνήμης. Μπορούμε δηλαδή να δεσμεύουμε και να αποδεσμεύουμε μνήμη ανάλογα με τις ανάγκες και τις απαιτήσεις μας. Άλλος ένας λόγος που προτιμήθηκε η C είναι το γεγονός ότι η C είναι μια φορητή γλώσσα, δηλ. τα προγράμματά της μπορούν να τρέξουν με λίγες ή και με καθόλου τροποποιήσεις και σε ένα άλλο σύστημα. Μάλιστα θεωρείται σαν η πιο φορητή γλώσσα. Επίσης ακόμη ένας λόγος που προτιμήθηκε η C είναι πως η C έχει μοντέρνες δομές ελέγχου για να μπορούμε να κάνουμε επαναληπτικές εργασίες και για εύκολη επιλογή εναλλακτικών τρόπων δράσης. Με το πλήθος των δομών δεδομένων που διαθέτει, μπορεί να αναπαραστήσει ένα μεγάλο σύνολο από διαφορετικούς τύπους πληροφοριών. Έχει και το μεγάλο πλεονέκτημα ότι επιβάλλει τη διάσπαση του προγράμματος σε αυτοδύναμες ενότητες, τις συναρτήσεις. Τελευταίος και σημαντικότερος λόγος που συντέλεσε για την επιλογή της είναι το ότι η C είναι δυναμική και ευέλικτη, δύο ιδιότητες που είναι αρκετά δημοφιλείς στους υπολογιστές. Όπως ξέρουμε, το μεγαλύτερο μέρος του δυναμικού και ευέλικτου λειτουργικού συστήματος Unix είναι γραμμένο σε C. Αυτό ισχύει και για επεξεργαστές κειμένων, μεταγλωττιστές (compilers) και ερμηνευτές (interpreters) γλωσσών προγραμματισμού. Η C διαθέτει μερικά από τα χαρακτηριστικά ελέγχου που συνήθως τα συναντάμε στη συμβολική γλώσσα (assembly language).

Οι δομές που δημιουργήθηκαν για τις ανάγκες του αλγορίθμου

Όπως αναφέρθηκε παραπάνω η γλώσσα C δίνει τη δυνατότητα στον προγραμματιστή να αναπαραστήσει ένα μεγάλο σύνολο από διαφορετικούς τύπους πληροφοριών. Προτού αναλυθεί η υλοποίηση του αλγορίθμου πρέπει να γίνει μία αναφορά στις δομές που χρειάστηκε να δημιουργηθούν με σκοπό την αναπαράσταση, την αποθήκευση και την διαχείριση των δεδομένων που χρησιμοποιεί αυτός. Έτσι λοιπόν δημιουργήθηκαν οι παρακάτω κύριες δομές:

- Struct Celli: Είναι η δομή που περιγράφει το κάθε κελί του διαχωρισμού. Έχει διάφορες μεταβλητές μέσα της, όπως το όνομα του κελιού, τις διαστάσεις του, τις συντεταγμένες του κελιού, το σύνολο των εισόδων και των εξόδων του, δείκτες για τις εισόδους του και τις εξόδους του, δείκτες για τα επόμενα και τα προηγούμενα κελιά στον τρέχον διαχωρισμό κτλπ.
- Struct Partition: Είναι η δομή η οποία περιγράφει τον κάθε διαχωρισμό. Οι μεταβλητές που έχει η συγκεκριμένη δομή είναι οι συντεταγμένες που καθορίζουν τον χώρο του διαχωρισμού, το όνομα του, το πόσα και ποια κελιά περιέχει ο διαχωρισμός, δείκτες για τον επόμενο και τον προηγούμενο διαχωρισμό κτλπ.

Να σημειωθεί ότι δημιουργήθηκαν και άλλες δομές, ωστόσο δεν αναφέρονται και δεν περιγράφονται διότι είναι συμπληρωματικές των δύο παραπάνω δομών.

Οι συναρτήσεις που δημιουργήθηκαν για τις ανάγκες του αλγορίθμου

Για να μπορέσει να υλοποιηθεί ο αλγόριθμος χρειάστηκε να δημιουργηθούν κάποιες συναρτήσεις. Οι κύριες συναρτήσεις που δημιουργήθηκαν αναφέρονται και περιγράφονται παρακάτω:

- `function parse_netlist`: Είναι συνάρτηση η οποία αναλύει συντακτικά κάποιο αρχείο με μορφή κειμένου με τη βοήθεια άλλων συναρτήσεων (όπως η `readline` που διαβάζει γραμμές) και αποθηκεύει τα δεδομένα που αναλύει στις αντίστοιχες δομές.
- `function create_adjacency_matrix`: Είναι η συνάρτηση που δοθέντος ενός δικτύου δημιουργεί τον πίνακα γειτνίασης.
- `function create_degree_matrix`: Είναι η συνάρτηση που δοθέντος του πίνακα γειτνίασης και του πίνακα συνδέσεων δημιουργεί τον διαγώνιο πίνακα.
- `function create_laplacian_matrix`: Είναι η συνάρτηση η οποία δοθέντος πίνακα γειτνίασης και διαγωνίου πίνακα δημιουργεί τον πίνακα της διαφοράς των δύο αυτών πινάκων.
- `function addCell`: Είναι η συνάρτηση που δοθέντος ενός κελιού και ενός διαχωρισμού τοποθετεί το συγκεκριμένο κελί στον συγκεκριμένο διαχωρισμό.

- `function rmCell`: Είναι η συνάρτηση που δοθέντος ενός κελιού και ενός διαχωρισμού αφαιρεί το συγκεκριμένο κελί από τον συγκεκριμένο διαχωρισμό.
- `function total_wirelength`: Είναι η συνάρτηση που υπολογίζει το μήκος της καλωδίωσης ενός κυκλώματος. Η συνάρτηση αυτή δημιουργήθηκε για την σύγκριση του μήκους καλωδίωσης μεταξύ των δύο αλγορίθμων κατάτμησης.
- `function collision`: Είναι η συνάρτηση που υπολογίζει το πλήθος των συγκρούσεων ενός κυκλώματος. Η συνάρτηση αυτή δημιουργήθηκε για την σύγκριση του πλήθους των συγκρούσεων μεταξύ των δύο αλγορίθμων κατάτμησης.

Να σημειωθεί ότι δημιουργήθηκαν και άλλες συναρτήσεις, ωστόσο δεν αναφέρονται και δεν περιγράφονται διότι είναι συμπληρωματικές των παραπάνω συναρτήσεων.

Η είσοδος του προγράμματος

Το πρόγραμμα έχει σαν είσοδο αρχεία σε μορφή κειμένου τα οποία περιγράφουν με συγκεκριμένο τρόπο τα στοιχεία του κυκλώματος.

Η έξοδος του προγράμματος

Το πρόγραμμα έχει σαν έξοδο ένα αρχείο το οποίο περιγράφει την τροποποιημένη μορφή του κυκλώματος όπως αυτή έχει διαμορφωθεί με τον αλγόριθμο κατάτμησης. Επίσης έχει ως έξοδο και ένα γραφικό περιβάλλον στο οποίο απεικονίζεται η μορφή του κυκλώματος όπως αυτό είναι αρχικά και πως αυτό διαμορφώνεται μετά τις τροποποιήσεις που γίνονται με τον αλγόριθμο κατάτμησης.

Η συνάρτηση του αλγορίθμου

Ο αλγόριθμος υλοποιείται σε μια συνάρτηση. Η συνάρτηση αυτή ονομάστηκε `partitioningEIG`. Ο λόγος που προτιμήθηκε η υλοποίηση σε συνάρτηση είναι το γεγονός ότι ο αλγόριθμος εκτελείται πάνω από μία φορές, μέσα στο πρόγραμμα κατά την διαδικασία του `placement`, με αποτέλεσμα ο κώδικας της διαδικασίας του διαχωρισμού να εκτελείται και αυτός όσες φορές χρειαστεί. Έτσι ο αλγόριθμος υλοποιήθηκε σε συνάρτηση η οποία καλείται όσες φορές

χρειαστεί κατά τη διαδικασία του placement με τις μεταβλητές που δέχεται ως παραμέτρους να μπορούν να αλλάζουν κάθε φορά και έτσι η συνάρτηση να μπορεί να υλοποιεί την συνάρτηση που χρειάζεται την κάθε φορά. Οι μεταβλητές αυτές είναι:

part1, part2: Είναι τα δύο αρχικά τμήματα του διαχωρισμού που δίνονται ως είσοδο στην συνάρτηση, τα οποία ενώνονται σε ένα τμήμα, το οποίο και υλοποιείται η διαδικασία διαχωρισμού.

cutline: Είναι η γραμμή η οποία διαχωρίζει τα δύο τμήματα του διαχωρισμού.

direction: Είναι η κατεύθυνση που θα έχει η γραμμή (κάθετη ή οριζόντια).

adjacency_matrix: Είναι ο πίνακας γειτνίασης των κελιών

pin_connection_matrix: Είναι ο πίνακας συνδέσεων

degree_matrix: Είναι ο διαγώνιος πίνακας

laplacian_matrix: Είναι ο πίνακας διαφορών.

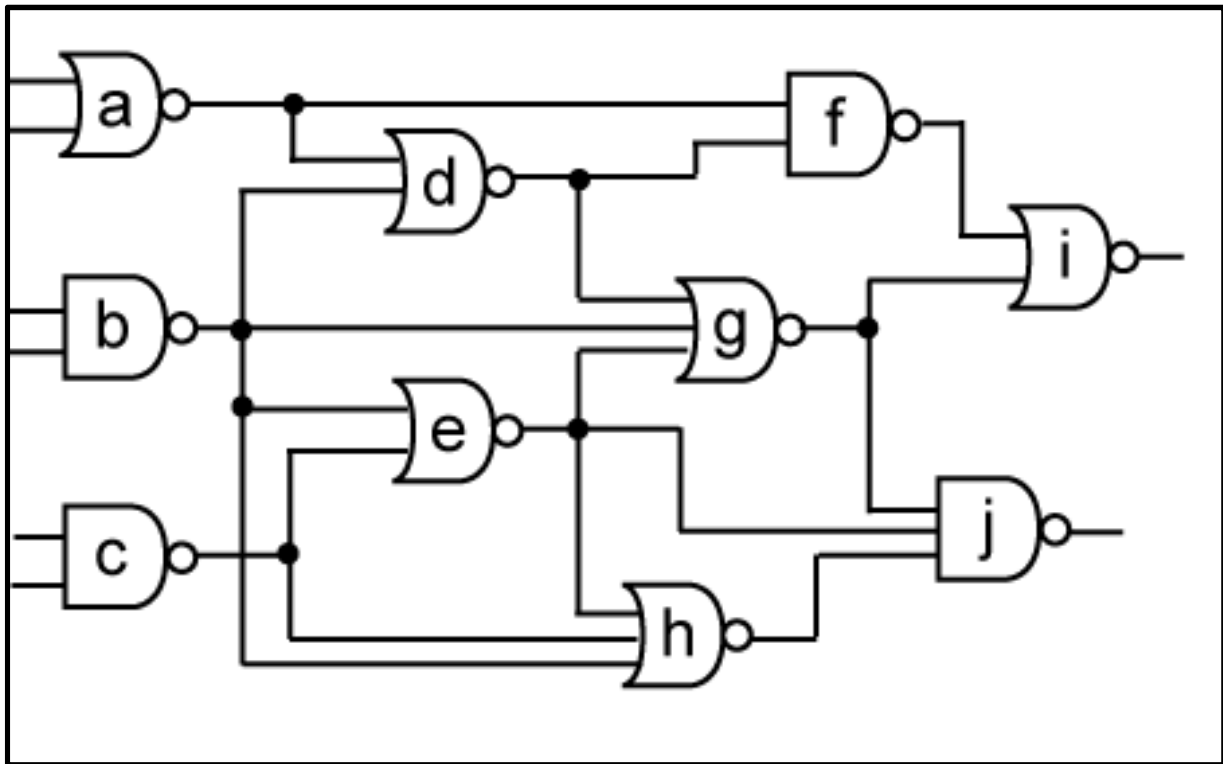
pin_matrix: Είναι η δομή που περιγράφει όλες της συνδέσεις.

Με βάση αυτές τις μεταβλητές η συνάρτηση υπολογίζει το καλύτερο δυνατό διαχωρισμό με κριτήριο το ελάχιστο μήκος καλωδίωσης που δύναται να υπάρχει στο κύκλωμα καθώς και τις ελάχιστες δυνατές συγκρούσεις που μπορούν να προκύψουν.

Ο αλγόριθμος κατάτμησης με λεπτομερή ανάλυση

Στη συνέχεια θα περιγράψουμε τον αλγόριθμο EIG των Hagen and Kahng αναλυτικότερα χρησιμοποιώντας ένα παράδειγμα για να γίνει περισσότερο κατανοητή η λειτουργία του.

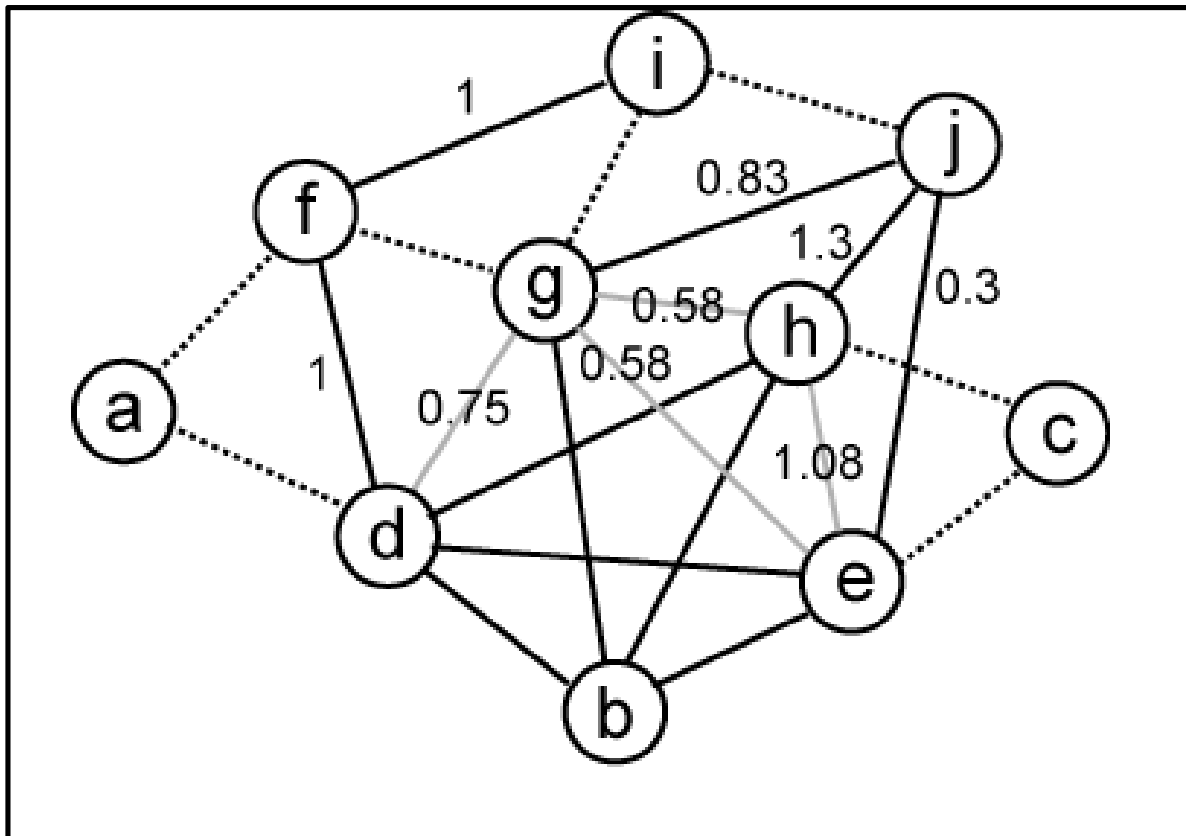
Το κύκλωμα του παραδείγματος στο οποίο θα εφαρμόσουμε τον αλγόριθμό μας είναι:



1.Κύκλωμα παραδείγματος

Το πρώτο πράγμα που πρέπει να γίνει είναι η μετατροπή του κυκλώματος σε ένα απλά γράφο με κορυφές και ακμές.

Ο γράφος του κυκλώματος του παραδείγματός μας είναι:



2. Γράφος κυκλώματος

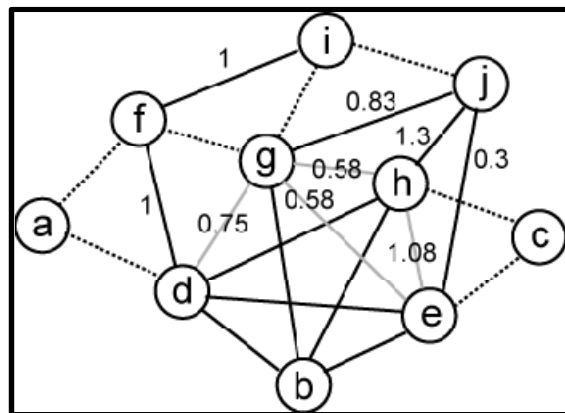
Να σημειωθεί πως ο αριθμός που αναγράφεται σε κάθε ακμή είναι το βάρος της. Οι ακμές με διακεκομμένη γραμμή έχουν βάρος 0.5 και οι γραμμές που δεν αναγράφουν αριθμό έχουν βάρος 0.25.

1. Δημιουργία Adjacency Matrix

Δημιουργούμε τον πίνακα που περιέχει τις συνδέσεις των κόμβων και το βάρος της κάθε μίας.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>
<i>a</i>	0	0	0	0.5	0	0.5	0	0	0	0
<i>b</i>	0	0	0	0.25	0.25	0	0.25	0.25	0	0
<i>c</i>	0	0	0	0	0.5	0	0	0.5	0	0
<i>d</i>	0.5	0.25	0	0	0.25	1.0	0.75	0.25	0	0
<i>e</i>	0	0.25	0.5	0.25	0	0	0.58	1.08	0	0.33
<i>f</i>	0.5	0	0	1.0	0	0	0.5	0	1.0	0
<i>g</i>	0	0.25	0	0.75	0.58	0.5	0	0.58	0.5	0.83
<i>h</i>	0	0.25	0.5	0.25	1.08	0	0.58	0	0	1.33
<i>i</i>	0	0	0	0	0	1.0	0.5	0	0	0.5
<i>j</i>	0	0	0	0	0.33	0	0.83	1.33	0.5	0

3. Πίνακας συνδέσεων

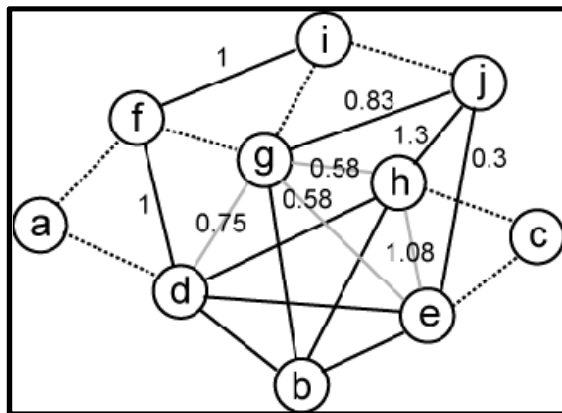


2. Δημιουργία Degree Matrix

Δημιουργούμε τον πίνακα που περιέχει το άθροισμα των γραμμών (ή στηλών) του Adjacency Matrix στη διαγώνιο ενώ τα υπόλοιπα στοιχεία ισούνται με μηδέν.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>
<i>a</i>	1.0	0	0	0	0	0	0	0	0	0
<i>b</i>	0	1.0	0	0	0	0	0	0	0	0
<i>c</i>	0	0	1.0	0	0	0	0	0	0	0
<i>d</i>	0	0	0	3.0	0	0	0	0	0	0
<i>e</i>	0	0	0	0	2.99	0	0	0	0	0
<i>f</i>	0	0	0	0	0	3.0	0	0	0	0
<i>g</i>	0	0	0	0	0	0	3.99	0	0	0
<i>h</i>	0	0	0	0	0	0	0	3.99	0	0
<i>i</i>	0	0	0	0	0	0	0	0	2.0	0
<i>j</i>	0	0	0	0	0	0	0	0	0	2.99

4. Πίνακας αθροισμάτων

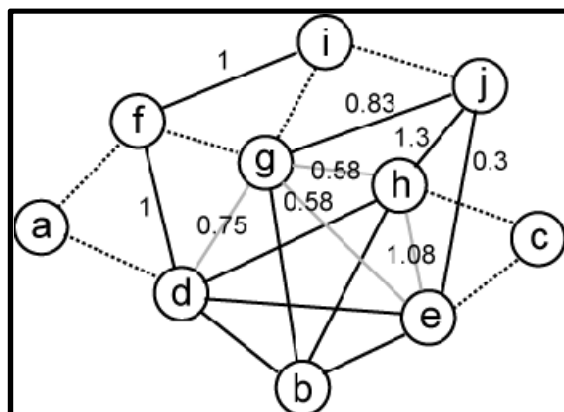


3. Δημιουργία Laplacian Matrix

Δημιουργούμε τον πίνακα που περιέχει τη διαφορά του Degree Matrix από τον Adjacency Matrix.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>
<i>a</i>	1.0	0	0	-0.5	0	-0.5	0	0	0	0
<i>b</i>	0	1.0	0	-0.25	-0.25	0	-0.25	-0.25	0	0
<i>c</i>	0	0	1.0	0	-0.5	0	0	-0.5	0	0
<i>d</i>	-0.5	-0.25	0	3.0	-0.25	-1.0	-0.75	-0.25	0	0
<i>e</i>	0	-0.25	-0.5	-0.25	2.99	0	-0.58	-1.08	0	-0.33
<i>f</i>	-0.5	0	0	-1.0	0	3.0	-0.5	0	-1.0	0
<i>g</i>	0	-0.25	0	-0.75	-0.58	-0.5	3.99	-0.58	-0.5	-0.83
<i>h</i>	0	-0.25	-0.5	-0.25	-1.08	0	-0.58	3.99	0	-1.33
<i>i</i>	0	0	0	0	0	-1.0	-0.5	0	2.0	-0.5
<i>j</i>	0	0	0	0	-0.33	0	-0.83	-1.33	-0.5	2.99

5. Πίνακας Διαφορών



4. Εύρεση δεύτερης μικρότερης ιδιοτιμής και ιδιοδιανύσματος

Βρίσκουμε τη δεύτερη μικρότερη ιδιοτιμή και το ιδιοδιάνυσμα που αναφέρεται σε αυτή.

Η δεύτερη μικρότερη ιδιοτιμή είναι

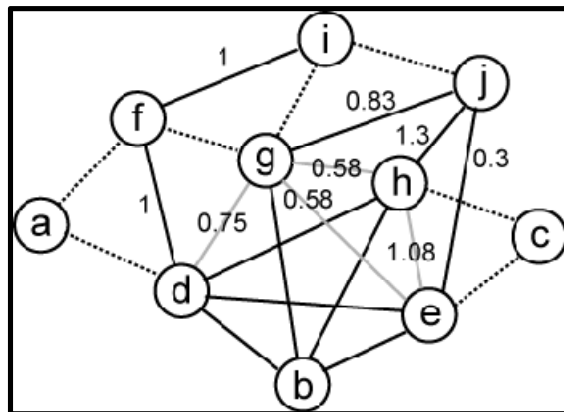
0.6281

και το ιδιοδιάνυσμα που αναφέρεται σε αυτή είναι

$[-0.6346, 0.1605, 0.5711, -0.1898, 0.2254, 0.2822, 0.0038, 0.1995, 0.1641, 0.1104]^T$

Παρατηρούμε τα εξής:

- Το τετραγωνικό άθροισμα των τιμών του ιδιοδιανύσματος είναι ίσο με 1.
- Οι τιμές των στοιχείων του ιδιοδιανύσματος ανήκουν στο διάστημα $[-1, 1]$



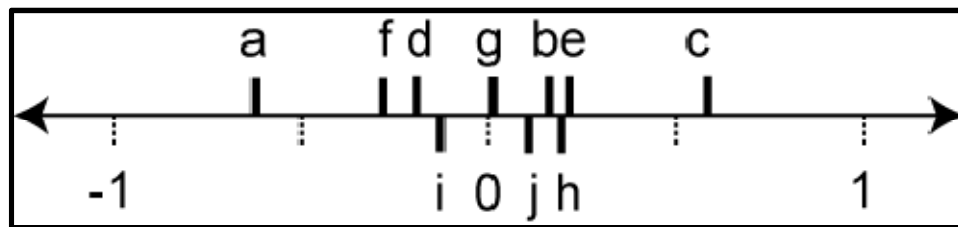
5. Ταξινόμηση Κόμβων

Κάνουμε ταξινόμηση των κόμβων με βάση τις τιμές του ιδιοδιανύσματος κατά αύξουσα σειρά.

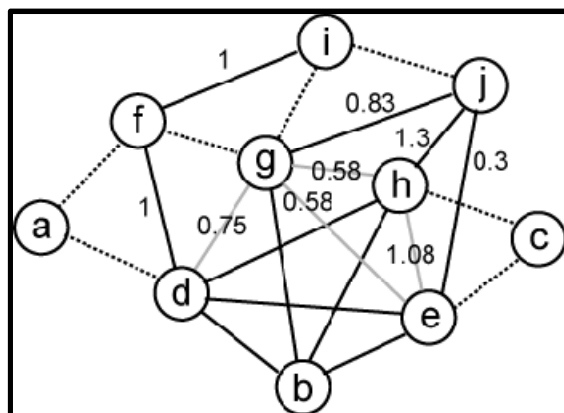
Οι τιμές αυτές μας δίνουν την ακόλουθη σειρά των κόμβων:

$$Z = \{a, f, d, i, g, j, b, h, e, c\}$$

Οι οποίοι ταξινομούνται στο διάστημα $[-1,1]$ ως εξής:



6. Ταξινόμηση Κόμβων



6. Προσθαφαίρεση Κόμβων

Στο στάδιο αυτό εκτελούνται $n-1$ ($=9$) φορές οι ακόλουθες πράξεις:

- Πρόσθεση στο πρώτο Partition και αφαίρεση από το δεύτερο Partition του επόμενου κόμβου από την ταξινόμηση του προηγούμενου βήματος
- Υπολογισμός του cut size και του ratio cut ανάμεσα στα δύο Partitions

Έχουμε:

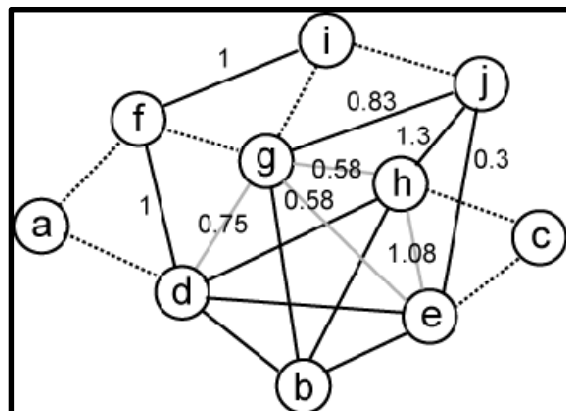
1^η φορά:

Διαχωρισμός: $\{a\}, \{f, d, i, g, j, b, h, e, c\}$

Ακμές σύνδεσης διαχωρισμού: (a, f) και (a, d)

Cutsize: $0.5 + 0.5 = 1.0$

Ratiocut: $1.0 / (1 \cdot 9) = 0.1111$



2^η φορά:

Διαχωρισμός:({a, f}, {d, i, g, j, b, h, e, c})

Ακμές σύνδεσης διαχωρισμού:(f, i), (f, g), (f, d) και (a, d)

Cutsize:1.0+0.5+1.0+0.5 = 3.0

Ratiocut:3.0/(2 · 8) = 0.1875

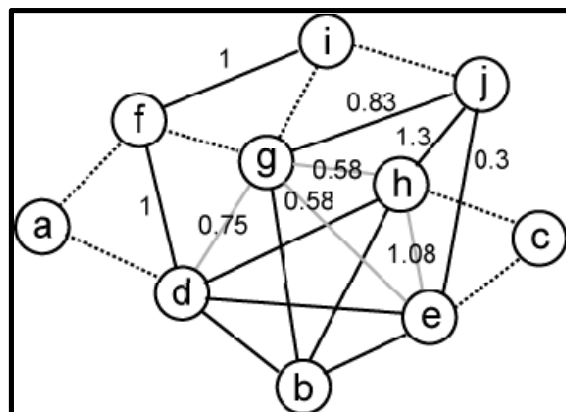
3^η φορά:

Διαχωρισμός:({a, f, d}, {i, g, j, b, h, e, c})

Ακμές σύνδεσης διαχωρισμού:(f, i), (f, g), (d, g),(d, h), (d, e), και (d, b)

Cutsize:1.0+0.5+0.75+3·0.25 =3.0

Ratiocut:3.0/(3 · 7) = 0.1429



4^η φορά:

Διαχωρισμός:({a, f, d, i}, {g, j, b, h, e, c})

Ακμές σύνδεσης διαχωρισμού:(i, j), (i, g), (f, g), (d, g), (d, h), (d, e), και (d, b)

Cutsize: $0.5 \cdot 3 + 0.75 + 3 \cdot 0.25 = 3.0$

Ratio cut: $3.0 / (4 \cdot 6) = 0.125$

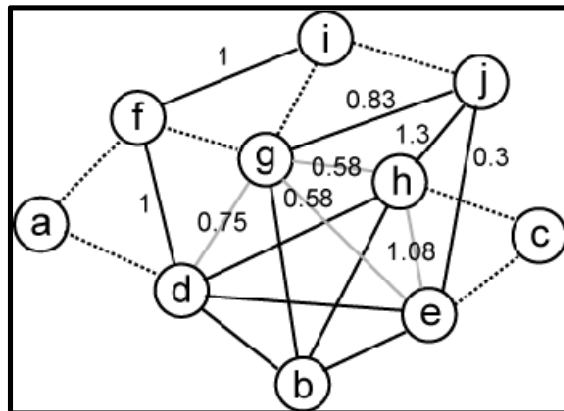
5^η φορά:

Διαχωρισμός:({a, f, d, i, g}, {j, b, h, e, c})

Ακμές σύνδεσης διαχωρισμού:(i, j), (g, j), (g, h), (g, e), (g, b), (d, h), (d, e), και (d, b)

Cutsize: $0.5 + 0.83 + 0.58 \cdot 2 + 0.25 \cdot 4 = 3.49$

Ratio cut: $3.49 / (5 \cdot 5) = 0.1396$



6^η φορά:

Διαχωρισμός:({a, f, d, i, g, j}, {b, h, e, c})

Ακμές σύνδεσης διαχωρισμού:(j, e), (j, h), (g, h), (g, e), (g, b), (d, h), (d, e), και (d, b)

Cutsize: $0.33+1.33+ 0.58 \cdot 2+0.25 \cdot 4 = 3.82$

Ratio cut: $3.82/(6 \cdot 4) = 0.1592$

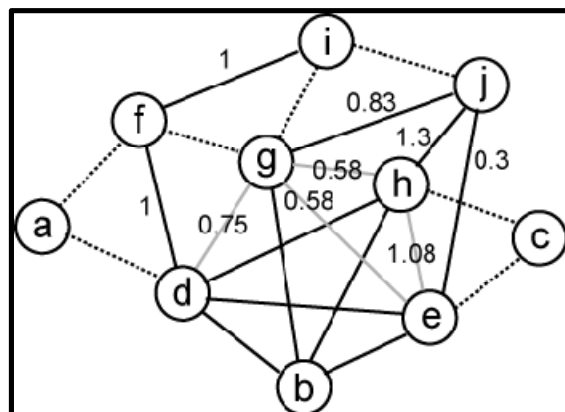
7^η φορά:

Διαχωρισμός:({a, f, d, i, g, j, b}, {h, e, c})

Ακμές σύνδεσης διαχωρισμού:(j, e), (j, h), (g, h),(g, e), (d, h), (d, e), (b, h), και (b, e)

Cutsize: $0.33+1.33+0.58 \cdot 2+0.25 \cdot 4 = 3.82$

Ratio cut: $3.82/(7 \cdot 3) = 0.1819$



8^η φορά:

Διαχωρισμός:({a, f, d, i, g, j, b, h}, {e, c})

Ακμές σύνδεσης διαχωρισμού:(h, c), (h, e), (j, e), (g, e), (d, e), και (b, e)

Cutsize:0.5+1.08+0.33+0.58+ 0.25 + 0.25 = 2.99

Ratiocut:2.99/(8 · 2) = 0.1869

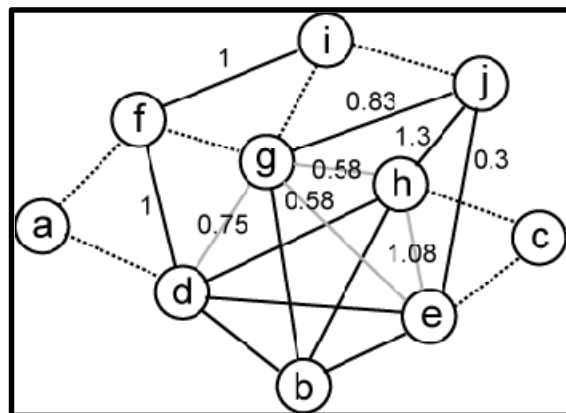
9^η φορά:

Διαχωρισμός:({a, f, d, i, g, j, b, h, e}, {c})

Ακμές σύνδεσης διαχωρισμού:(h, c) και (e, c)

Cutsize:0.5 + 0.5 = 1.0

Ratiocut:1.0/(9 · 1) = 0.1111

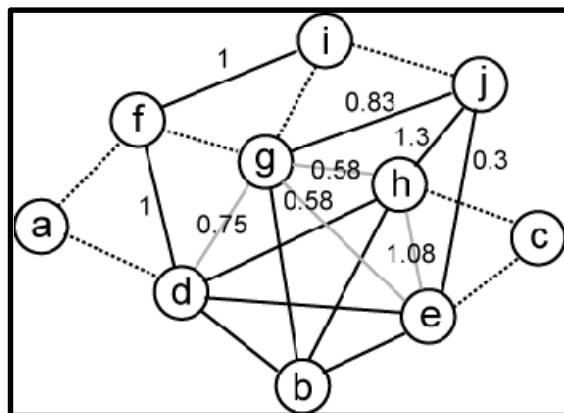


7. Εύρεση καλής λύσης

Μέσω ευριστικής λύσης βρίσκουμε πως μία ισοροπημένη λύση με χαμηλό κόστος σύνδεσης διαχωρισμού είναι ο διαχωρισμός: $\{(a,f,d,g,i), (j,b,h,e,c)\}$

P_A	P_B	cutsize	ratio cut
$\{a\}$	$\{f, d, i, g, j, b, h, e, c\}$	1.0	$1.0/(1 \cdot 9) = 0.1111$
$\{a, f\}$	$\{d, i, g, j, b, h, e, c\}$	3.0	$3.0/(2 \cdot 8) = 0.1875$
$\{a, f, d\}$	$\{i, g, j, b, h, e, c\}$	3.0	$3.0/(3 \cdot 7) = 0.1429$
$\{a, f, d, i\}$	$\{g, j, b, h, e, c\}$	3.0	$3.0/(4 \cdot 6) = 0.125$
$\{a, f, d, i, g\}$	$\{j, b, h, e, c\}$	3.49	$3.49/(5 \cdot 5) = 0.1396$
$\{a, f, d, i, g, j\}$	$\{b, h, e, c\}$	3.82	$3.82/(6 \cdot 4) = 0.1592$
$\{a, f, d, i, g, j, b\}$	$\{h, e, c\}$	3.82	$3.82/(7 \cdot 3) = 0.1819$
$\{a, f, d, i, g, j, b, h\}$	$\{e, c\}$	2.99	$2.99/(8 \cdot 2) = 0.1869$
$\{a, f, d, i, g, j, b, h, e\}$	$\{c\}$	1.0	$1.0/(9 \cdot 1) = 0.1111$

7.Δυνατοί διαχωρισμοί



8. Πιστοποίηση θεωρήματος

Πιστοποιούμε ότι η δεύτερη μικρότερη ιδιοτιμή είναι το κατώτερο όριο σε όλα τα κόστη σύνδεσης διαχωρισμού.

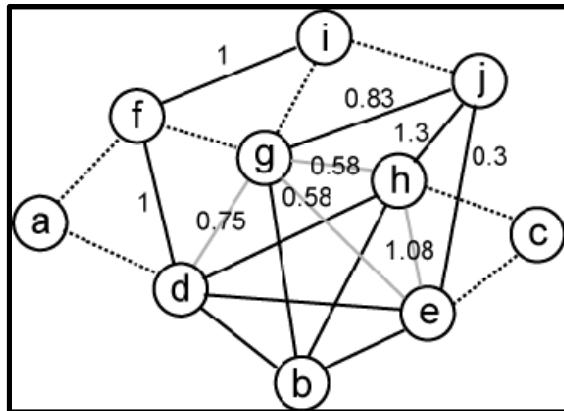
Η ιδιοτιμή αυτή είναι $\lambda=0.6281$.

Βλέπουμε πως $c \geq \lambda/n$

όπου:

- c είναι το ratio cut cost
- n είναι ο αριθμός των κόμβων

Στην περίπτωση μας έχουμε $n=10$, οπότε $\lambda/n=0.06281$ είναι μικρότερο από όλα τα κόστη σύνδεσης διαχωρισμού.



Αποτελέσματα

Σε αυτό το κεφάλαιο παρουσιάζονται τα αποτελέσματα του αλγορίθμου μας, τον οποίο τρέξαμε χρησιμοποιώντας διαφορετικά κυκλώματα και φυσικά θα προχωρήσουμε στον περαιτέρω σχολιασμό αυτών των αποτελεσμάτων.

Να σημειωθεί πως τα κυκλώματα που χρησιμοποιήθηκαν είναι αυτά που αναφέρονται στην σελίδα 12.

Τα αποτελέσματα παρουσιάζονται σε μορφή πινάκων.

Ουσιαστικά αυτό που παρουσιάζεται είναι οι διαφορές ανάμεσα στον ήδη υλοποιημένο αλγόριθμο διαχωρισμού Kernighan and Lin και του αλγορίθμου διαχωρισμού EIG που υλοποιήθηκε στα πλαίσια της παρούσας εργασίας, όσον αφορά τις μεταβλητές:

- Total wirelength
- Collisions
- Wirelength after optimization(euc)
- Wirelength after optimization(hp)

Οι μεταβλητές wirelength (euc) και wirelength (hp) συγκρίνονται στις τιμές που έχουν μετά τη βελτιστοποίησή τους, καθώς οι τιμές αυτές είναι ίδιες πριν τη βελτιστοποίηση και για τους δύο αλγορίθμους διαχωρισμού.

Στη συνέχεια, παρουσιάζονται οι πίνακες με τα αποτελέσματα κάθε κυκλώματος ξεχωριστά.

Για το κύκλωμα s208

S208		
Αλγόριθμος	KL	EIG
Total wirelength	697.012414	644.838521
Collisions	317	316
Wirelength before optimization(eucl)	653.352087	653.352087
Wirelength after optimization(eucl)	684.489444	628.161864
Wirelength before optimization(hp)	560.333300	560.333300
Wirelength after optimization(hp)	529.863571	498.423571

2.Αποτελέσματα κυκλώματος S208

Για το κύκλωμα s298 :

S298		
Αλγόριθμος	KL	EIG
Total wirelength	1295.178883	1020.650825
Collisions	554	541
Wirelength before optimization(eucl)	1371.347890	1371.347890
Wirelength after optimization(eucl)	1179.856907	1176.276062
Wirelength before optimization(hp)	1149.323565	1149.323565
Wirelength after optimization(hp)	964.965333	920.735333

3. Αποτελέσματα κυκλώματος S298

Για το κύκλωμα s344 :

S344		
Αλγόριθμος	KL	EIG
Total wirelength	3311.830441	2870.055974
Collisions	1198	1149
Wirelength before optimization(eucl)	3885.235494	3885.235494
Wirelength after optimization(eucl)	3212.917103	3204.350720
Wirelength before optimization(hp)	2949.537650	2949.537650
Wirelength after optimization(hp)	2541.055000	2374.785000

4.Αποτελέσματα κυκλώματος S344

Για το κύκλωμα s349 :

S349		
Αλγόριθμος	KL	EIG
Total wirelength	2946.880512	2410.750329
Collisions	1082	1011
Wirelength before optimization(eucl)	3036.605608	3036.605608
Wirelength after optimization(eucl)	2913.892199	2601.266346
Wirelength before optimization(hp)	2513.561274	2513.561274
Wirelength after optimization(hp)	2331.459000	1914.109000

5.Αποτελέσματα κυκλώματος S349

Για το κύκλωμα s382 :

S382		
Αλγόριθμος	KL	EIG
Total wirelength	2079.946351	1764.006075
Collisions	870	802
Wirelength before optimization(eucl)	2530.147627	2530.147627
Wirelength after optimization(eucl)	2224.042935	2032.550327
Wirelength before optimization(hp)	2048.251486	2048.251486
Wirelength after optimization(hp)	1749.024400	1534.834400

6.Αποτελέσματα κυκλώματος S382

Για το κύκλωμα s386 :

S386		
Αλγόριθμος	KL	EIG
Total wirelength	1920.303157	1903.858763
Collisions	851	816
Wirelength before optimization(eucl)	2183.402201	2183.402201
Wirelength after optimization(eucl)	1951.068997	1907.823676
Wirelength before optimization(hp)	1694.579943	1694.579943
Wirelength after optimization(hp)	1456.582800	1403.167800

7.Αποτελέσματα κυκλώματος S386

Για το κύκλωμα s400 :

S400		
Αλγόριθμος	KL	EIG
Total wirelength	2222.672009	1777.090894
Collisions	815	812
Wirelength before optimization(eucl)	2243.870179	2243.870179
Wirelength after optimization(eucl)	2162.123798	2012.373937
Wirelength before optimization(hp)	1912.938660	1912.938660
Wirelength after optimization(hp)	1770.951600	1658.536600

8.Αποτελέσματα κυκλώματος S400

Για το κύκλωμα s420 :

S420		
Αλγόριθμος	KL	EIG
Total wirelength	4117.071508	3063.277072
Collisions	1295	1216
Wirelength before optimization(eucl)	4038.776351	4038.776351
Wirelength after optimization(eucl)	3878.984475	3565.276813
Wirelength before optimization(hp)	3261.148442	3261.148442
Wirelength after optimization(hp)	2988.031250	2642.296250

9.Αποτελέσματα κυκλώματος S420

Για το κύκλωμα s444 :

S444		
Αλγόριθμος	KL	EIG
Total wirelength	3042.139550	2090.843034
Collisions	1041	1005
Wirelength before optimization(eucl)	3081.352060	3081.352060
Wirelength after optimization(eucl)	2787.918085	2407.321320
Wirelength before optimization(hp)	2536.685510	2536.685510
Wirelength after optimization(hp)	2324.078000	1889.963000

10.Αποτελέσματα κυκλώματος S444

Για το κύκλωμα s510 :

S510		
Αλγόριθμος	KL	EIG
Total wirelength	6079.340828	4374.113612
Collisions	1815	1742
Wirelength before optimization(eucl)	6158.005075	6158.005075
Wirelength after optimization(eucl)	5524.875155	5223.387599
Wirelength before optimization(hp)	4546.623021	4546.623021
Wirelength after optimization(hp)	4046.688067	3322.373067

11.Αποτελέσματα κυκλώματος S510

Για το κύκλωμα s526 :

S526		
Αλγόριθμος	KL	EIG
Total wirelength	3084.656644	1705.325504
Collisions	1054	1015
Wirelength before optimization(eucl)	3200.761768	3200.761768
Wirelength after optimization(eucl)	3063.047057	2199.083100
Wirelength before optimization(hp)	2536.408014	2536.408014
Wirelength after optimization(hp)	2451.905000	1581.330000

12.Αποτελέσματα κυκλώματος S526

Για το κύκλωμα s641 :

S641		
Αλγόριθμος	KL	EIG
Total wirelength	6626.625902	4375.627727
Collisions	1717	1704
Wirelength before optimization(eucl)	6049.397131	6049.397131
Wirelength after optimization(eucl)	5995.208320	4763.069746
Wirelength before optimization(hp)	5423.929792	5423.929792
Wirelength after optimization(hp)	5214.114286	4105.904286

13.Αποτελέσματα κυκλώματος S641

Για το κύκλωμα s713 :

S713		
Αλγόριθμος	KL	EIG
Total wirelength	6485.299483	6026.625902
Collisions	1718	1717
Wirelength before optimization(eucl)	5926.603933	5926.603933
Wirelength after optimization(eucl)	5946.428296	4729.076962
Wirelength before optimization(hp)	5325.458890	5325.458890
Wirelength after optimization(hp)	5265.226929	4120.766929

14.Αποτελέσματα κυκλώματος S713

Για το κύκλωμα s820 :

S820		
Αλγόριθμος	KL	EIG
Total wirelength	5024.005247	4197.672164
Collisions	1688	1547
Wirelength before optimization(eucl)	5628.746522	5628.746522
Wirelength after optimization(eucl)	4809.099215	4689.495445
Wirelength before optimization(hp)	4175.719942	4175.719942
Wirelength after optimization(hp)	3574.276154	3236.591154

15.Αποτελέσματα κυκλώματος S820

Για το κύκλωμα s832 :

S832		
Αλγόριθμος	KL	EIG
Total wirelength	7496.843316	5928.767115
Collisions	1871	1768
Wirelength before optimization(eucl)	7530.934238	7530.934238
Wirelength after optimization(eucl)	7214.451137	6503.718755
Wirelength before optimization(hp)	5422.535437	5422.535437
Wirelength after optimization(hp)	5241.156200	4361.991200

16.Αποτελέσματα κυκλώματος S832

Για το κύκλωμα s838 :

S838		
Αλγόριθμος	KL	EIG
Total wirelength	8434.495357	6077.250164
Collisions	2023	1912
Wirelength before optimization(eucl)	7888.067637	7888.067637
Wirelength after optimization(eucl)	8010.368877	6592.782733
Wirelength before optimization(hp)	6373.085004	6373.085004
Wirelength after optimization(hp)	6430.296000	5111.261000

17.Αποτελέσματα κυκλώματος S838

Για το κύκλωμα s953 :

S953		
Αλγόριθμος	KL	EIG
Total wirelength	10006.593591	7965.699390
Collisions	2573	2521
Wirelength before optimization(eucl)	11000.048028	11000.048028
Wirelength after optimization(eucl)	9926.822373	9446.490911
Wirelength before optimization(hp)	8450.264192	8450.264192
Wirelength after optimization(hp)	7654.749375	6890.759375

19.Αποτελέσματα κυκλώματος S953

Για το κύκλωμα s1196 :

S1196		
Αλγόριθμος	KL	EIG
Total wirelength	13930.052922	10678.778698
Collisions	3216	3090
Wirelength before optimization(eucl)	15184.279693	15184.279693
Wirelength after optimization(eucl)	13280.017612	12756.628037
Wirelength before optimization(hp)	11301.279888	11301.279888
Wirelength after optimization(hp)	9928.500556	9074.670556

20.Αποτελέσματα κυκλώματος S1196

Για το κύκλωμα s1238 :

s1238		
Αλγόριθμος	KL	EIG
Total wirelength	18254.114284	13364.696214
Collisions	3518	3288
Wirelength before optimization(eucl)	18073.550343	18073.550343
Wirelength after optimization(eucl)	16461.349827	14958.851750
Wirelength before optimization(hp)	13514.866228	13514.866228
Wirelength after optimization(hp)	12388.498500	10072.863500

21.Αποτελέσματα κυκλώματος S1238

Για το κύκλωμα s1423 :

S1423		
Αλγόριθμος	KL	EIG
Total wirelength	25048.640121	17386.739642
Collisions	4629	4474
Wirelength before optimization(eucl)	25471.394286	25471.394286
Wirelength after optimization(eucl)	23876.677286	20346.172229
Wirelength before optimization(hp)	21653.983122	21653.983122
Wirelength after optimization(hp)	19748.737000	15745.822000

22.Αποτελέσματα κυκλώματος S1423

Για το κύκλωμα s1488 :

S1488		
Αλγόριθμος	KL	EIG
Total wirelength	16272.800873	10284.207135
Collisions	3329	3150
Wirelength before optimization(eucl)	15502.785719	15502.785719
Wirelength after optimization(eucl)	15177.433808	12417.025766
Wirelength before optimization(hp)	11309.062573	11309.062573
Wirelength after optimization(hp)	11053.793889	8337.843889

23.Αποτελέσματα κυκλώματος S1488

Για το κύκλωμα s1494 :

S1494		
Αλγόριθμος	KL	EIG
Total wirelength	21506.899152	14829.457174
Collisions	3651	3062
Wirelength before optimization(eucl)	20122.707226	20122.707226
Wirelength after optimization(eucl)	19940.567673	16930.039121
Wirelength before optimization(hp)	13967.643959	13967.643959
Wirelength after optimization(hp)	13662.923700	10255.273700

24.Αποτελέσματα κυκλώματος S1494

Για το κύκλωμα s5378 :

S5378		
Αλγόριθμος	KL	EIG
Total wirelength	49786.477908	36094.508353
Collisions	6552	6243
Wirelength before optimization(eucl)	45639.869247	45639.869247
Wirelength after optimization(eucl)	43821.106907	38100.029679
Wirelength before optimization(hp)	38749.340548	38749.340548
Wirelength after optimization(hp)	37213.912440	32622.262440

25.Αποτελέσματα κυκλώματος S5378

Για το κύκλωμα s9234 :

S9234		
Αλγόριθμος	KL	EIG
Total wirelength	78410.263427	63570.901046
Collisions	9675	9489
Wirelength before optimization(eucl)	78385.626721	78385.626721
Wirelength after optimization(eucl)	73757.248108	68644.154733
Wirelength before optimization(hp)	66915.258270	66915.258270
Wirelength after optimization(hp)	62090.773516	57492.848516

26.Αποτελέσματα κυκλώματος S9234

Για το κύκλωμα s13207 :

S13207		
Αλγόριθμος	KL	EIG
Total wirelength	145417.909455	112815.376472
Collisions	13666	13387
Wirelength before optimization(eucl)	132108.529241	132108.529241
Wirelength after optimization(eucl)	129627.054863	115883.305024
Wirelength before optimization(hp)	119528.496848	119528.496848
Wirelength after optimization(hp)	115246.085943	104362.765943

27.Αποτελέσματα κυκλώματος S13207

Για το κύκλωμα s15850 :

S15850		
Αλγόριθμος	KL	EIG
Total wirelength	213041.058848	145719.958071
Collisions	17904	17264
Wirelength before optimization(eucl)	191175.728785	191175.728785
Wirelength after optimization(eucl)	189257.107907	151165.629104
Wirelength before optimization(hp)	167195.240807	167195.240807
Wirelength after optimization(hp)	158086.548200	128417.383200

28. Αποτελέσματα κυκλώματος S15850

Για το κύκλωμα s35932 :

S35932		
Αλγόριθμος	KL	EIG
Totalwirelength	733444.406966	389673.165859
Collisions	32010	31572
Wirelength before optimization(eucl)	501579.562516	501579.562516
Wirelength after optimization(hp)	492927.126317	395316.357500
Wirelength before optimization(hp)	444209.246135	444209.246135
Wirelength after optimization(hp)	424981.470000	353935.275000

29.Αποτελέσματα κυκλώματος S35932

Αφού έχουμε ολοκληρώσει όλα τα πειράματα μπορούμε να σχολιάσουμε τους πίνακες των αποτελεσμάτων τους.

Αυτό που παρατηρούμε από τους πίνακες των αποτελεσμάτων είναι το γεγονός ότι για όλες τις περιπτώσεις των διαφόρων κυκλωμάτων έχουμε βελτίωση. Έχει μειωθεί το συνολικό μήκος καλωδίωσης (Total wirelength), οι συγκρούσεις (Collisions), το μήκος καλωδίωσης (eucl) μετά την βελτιστοποίηση(Wirelength before optimization(eucl)) και το μήκος καλωδίωσης (hp) μετά την βελτιστοποίηση(Wirelength after optimization(hp)) .

Δηλαδή, ο αλγόριθμος EIG κατάφερε, όντως, να βελτιώσει την διαδικασία του διαχωρισμού των πυλών και κατ' επέκταση τη διαδικασία της χωροθέτησης των πυλών σε όλα τα κυκλώματα που τον τρέξαμε.

Συμπεράσματα

Παρατηρώντας την μεγάλη τεχνολογική ανάπτυξη στον κόσμο των ηλεκτρονικών, μπορούμε να συμπεράνουμε ότι η έρευνα για τη βελτιστοποίηση της χωροθέτησης των πυλών στην επιφάνεια ενός τσίπ αποτελεί ένα πολύ σημαντικό θέμα.

Αυτό, λοιπόν που μπορούμε να πούμε κατ'αρχήν είναι πως ο αλγόριθμος με τον οποίο ασχοληθήκαμε σε όλη τη διάρκεια της διπλωματικής εργασίας είναι σημαντικότερος κατά τη διαδικασία του placement.

Η μείωση του μήκους καλωδίωσης είναι ιδιαίτερα αισθητή σε σχέση με το αλγόριθμο KL καθώς και οι μείωση των συγκρούσεων.

Οι διαφορές που προκύπτουν στη μείωση των παραπάνω μεταβλητών είναι περισσότερο αισθητές σε μεγάλα κυκλώματα.

Αν προσπαθήσουμε να κατανοήσουμε το λόγο της τόσο μεγάλης μείωσης του μήκους καλωδίωσης, και της μείωσης του αριθμού των συγκρούσεων τότε αυτό που θα δούμε είναι πως ο λόγος είναι το γεγονός ότι ο αλγόριθμος EIG στηρίζει τη λειτουργία του στον πίνακα που καθορίζεται από τις συνδέσεις μεταξύ των πυλών.

Όλα τα παραπάνω μας οδηγούν στο συμπέρασμα ότι ο αλγόριθμος EIG πρέπει να προτιμάται έναντι του αλγορίθμου KL κατά τη διαδικασία του placement.

Υπάρχουν, όμως, σημαντικές δυνατότητες για περαιτέρω βελτίωση της διαδικασίας του διαχωρισμού των πυλών ακόμη. Ευχή μας είναι να βελτιωθεί ακόμα περισσότερο διότι στην εποχή μας, η λειτουργία πολλών ηλεκτρονικών συσκευών, και πολλών προϊόντων που χρησιμοποιούμε στην καθημερινότητα μας στηρίζεται σε αυτό. Χάρη στην βελτίωση της διαδικασίας διαχωρισμού το κόστος των παραπάνω ηλεκτρικών συσκευών και προϊόντων θα μειωθεί, με αποτέλεσμα να γίνουν πιο προσιτά στους καταναλωτές. Αγοράζοντας και χρησιμοποιώντας τις συσκευές αυτές θα υπάρξει άμεσο αντίκτυπο στη ζωή τους, παρέχοντας τους πολλές διευκολύνσεις σε διάφορους τομείς της καθημερινότητάς τους.

Βιβλιογραφία

- Practical Problems in VLSI Physical Design Automation
Sung Kyu Lim
- New Spectral Methods for Ratio Cut Partitioning and Clustering
Lars Hagen, Andrew B. Kahng
- Fast Spectral Methods for Ratio Cut Partitioning and Clustering
Lars Hagen and Andrew Kahng
- Combinational profiles of sequential benchmark circuits. FrancBrglez, David Bryan, and Krzysztof Kozminski