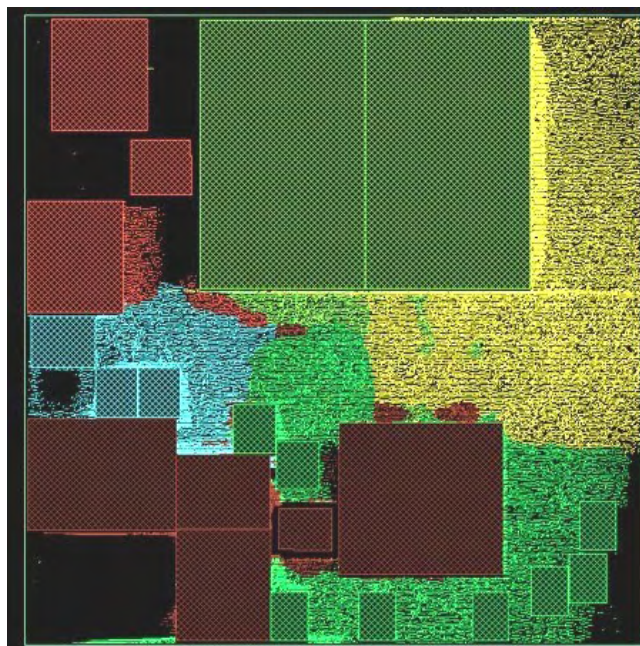


Ανάπτυξη ενός legalizer αλγορίθμου για ολοκληρωμένα κυκλώματα

ΛΑΖΑΡΙΔΟΥ ΚΩΝΣΤΑΝΤΙΝΑ

10/2/2015



Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών και βασικό επιβλέποντα αυτής της διπλωματικής εργασίας κ. Γεώργιο Σταμούλη που μου έδωσε την ευκαιρία να πραγματοποιήσω αυτή τη μελέτη. Η υποστήριξή του, η αμέριστη συμπαράστασή του, αλλά και οι διαρκείς και εύστοχες υποδείξεις του βοήθησαν στην έγκαιρη ολοκλήρωση αυτής της μελέτης.

Επιπρόσθετα, θα ήθελα να ευχαριστήσω τον διδακτορικό φοιτητή του τμήματος και καλό μου φίλο Γιάννη Αρβανιτάκη για την συμπαράστασή του και τη διευκόλυνση στη κατανόηση βασικών εννοιών του Placement .

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου, που μου συμπαραστάθηκαν καθ' όλη την διάρκεια των σπουδών μου.

Contents

1.	Σχεδίαση Ολοκληρωμένων Κυκλωμάτων	1
1.1	Φυσική Σχεδίαση	2
1.2	Εργαλεία CAD (Computer Aided Design).....	8
1.3	Εργαλεία EDA (Electronic Design Automation)	11
2.	Placement.....	15
2.1	Διατύπωση του προβλήματος	16
2.2	Το placement εντός της EDA ροής σχεδίασης	18
2.3	Ταξινόμηση των αλγορίθμων placement.....	19
2.4	Μετρήσεις Καλωδίων.....	21
3.	Μέθοδος Placement και αλγόριθμοι.....	25
3.1	Μέθοδος Global Placement.....	25
3.2	Μέθοδοι Final placement.....	30
4.	Τεχνικές Legalization.....	37
4.1	Flow και Diffusion-based legalization.....	37
4.2	Tetris-Based Legalization	38
4.3	Single-Row Dynamic Programming-Based Legalization	39
4.4	Όρια του Legalization και του Detailed Placement.....	42
5.	Σύγκριση Legalizer αλγορίθμων	45
5.1	Legalizer αλγόριθμοι προς σύγκριση	45
5.2	Περιβάλλον testing	46
6.	Συμπεράσματα και μελλοντική έρευνα.....	Error! Bookmark not defined.
7.	Βιβλιογραφία.....	Error! Bookmark not defined.

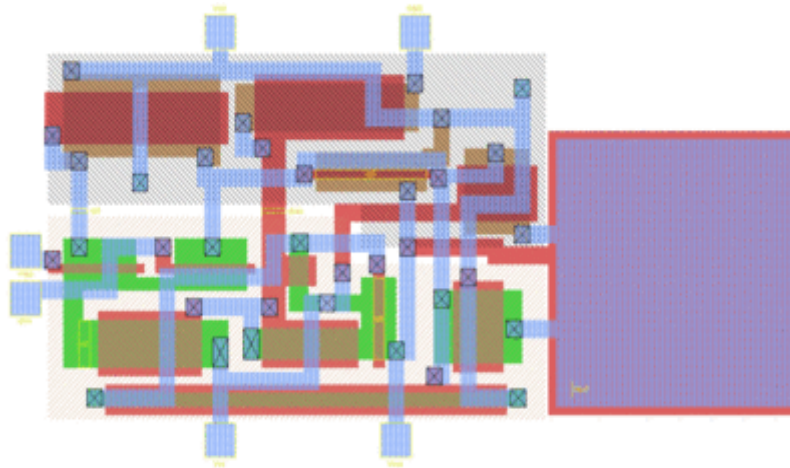
1. Σχεδίαση Ολοκληρωμένων Κυκλωμάτων

Η σχεδίαση ολοκληρωμένων κυκλωμάτων (integrated circuit design) ή IC σχεδίαση αποτελεί κλάδο των ηλεκτρολόγων μηχανικών, ο οποίος περιλαμβάνει ειδικότερα τεχνικές λογικής και σχεδίασης κυκλωμάτων, οι οποίες απαιτούνται για την σχεδίαση ολοκληρωμένων κυκλωμάτων. Τα ICs αποτελούνται από μικροσκοπικά ηλεκτρονικά στοιχεία ενσωματωμένα σε ένα ηλεκτρικό δίκτυο.

Η IC σχεδίαση μπορεί να χωριστεί σε δύο κατηγορίες την ψηφιακή(digital) και την αναλογική(analog) σχεδίαση. Η ψηφιακή παράγει κυκλώματα όπως μικροεπεξεργαστές (microprocessors), FPGAs, μνήμες(RAM, ROM, FLASH) και ψηφιακά ASICs(Application Specific Integrated Circuits). Επικεντρώνεται κυρίως στην λογική ορθότητα, την μεγιστοποίηση της πυκνότητας του κυκλώματος και την τοποθέτηση του έτσι ώστε τα σήματα ρολογιού και συγχρονισμού να δρομολογούνται αποτελεσματικά.

Η αναλογική σχεδίαση από την άλλη μεριά ασχολείται με την ισχύ και την αναλογία συχνότητας (ratio frequency) των ολοκληρωμένων κυκλωμάτων. Χρησιμοποιείται κυρίως στον σχεδιασμό γραμμικών ρυθμιστών (linear regulators), φάσεων κλειδωμένων βρόγχων (phase locked loops), ταλαντωτών (oscillators) και ενεργών φίλτρων (active filters) και ασχολείται περισσότερο με τα φυσικά χαρακτηριστικά των συσκευών ημιαγωγών όπως το κέρδος, η κατανάλωση ισχύος και η αντίσταση. Τέλος, συγκριτικά με την ψηφιακή σχεδίαση η αναλογική χρησιμοποιεί ευρύτερες περιοχές ενεργών συσκευών και συνήθως η πυκνότητα των κυκλωμάτων είναι μικρότερη.

Τα σύγχρονα ολοκληρωμένα κυκλώματα είναι εξαιρετικά περίπλοκα. Ενδεικτικά, ένα μεγάλο chip, από το 2009 έχει σχεδόν 1 δισ transistors. Η πολυπλοκότητα αυτή σε συνδυασμό με την συνεχώς αυξανόμενη πίεση της αγοράς για μεγάλη και σύντομη παραγωγή, είχαν ως αποτέλεσμα την εκτεταμένη χρήση εργαλείων αυτοματοποίησης (automated design tools) κατά την διαδικασία του σχεδιασμού IC, με στόχο τον σχεδιασμό, την δοκιμή και την επαλήθευση των εντολών που πρέπει να πραγματοποιεί το IC.



Εικόνα 1 Layout view of a simple CMOS Operational Amplifier(wiki.org)

1.1 Φυσική Σχεδίαση

Στη σχεδίαση ολοκληρωμένων κυκλωμάτων η φυσική σχεδίαση αποτελεί ένα βήμα του πρότυπου κύκλου σχεδίασης που ακολουθεί μετά τον σχεδιασμό του κυκλώματος. Σε αυτό το βήμα, οι κυκλωματικές αναπαραστάσεις των στοιχείων της σχεδίασης μετατρέπονται σε γεωμετρικές αναπαραστάσεις οι οποίες, όταν κατασκευαστούν από τα αντίστοιχα υλικά, θα εξασφαλίσουν την απαιτούμενη λειτουργικότητα των στοιχείων. Οι γεωμετρικές αυτές αναπαραστάσεις συνθέτουν το σχέδιο του ολοκληρωμένου κυκλώματος (integrated circuit layout). Η φυσική σχεδίαση συνήθως χωρίζεται σε ορισμένα επιμέρους βήματα τα οποία περιλαμβάνουν τόσο τον σχεδιασμό όσο και την επαλήθευση και επικύρωση του layout.

Η σχεδίαση ολοκληρωμένων κυκλωμάτων χωρίζεται σε :

- Σχεδίαση front-end χρησιμοποιώντας HDLs,
- Επαλήθευση (*Verification*),
- Σχεδίαση back-end ή φυσική σχεδίαση.

Κατηγορίες (divisions)

Τυπικά, η φυσική σχεδίαση των ολοκληρωμένων κυκλωμάτων χωρίζεται σε δύο κατηγορίες:

Full-custom: Ο σχεδιαστής έχει απόλυτη ευελιξία στο σχέδιο και δεν υπάρχουν προκαθορισμένα κελιά.

Semi-custom: Χρησιμοποιούνται προσχεδιασμένα κελιά και ο σχεδιαστής ασχολείται με την τοποθέτηση των κελιών και την διασύνδεση.

1.1.1 Ροή Φυσικής Σχεδίασης

Η φυσική σχεδίαση αποτελείται από βασικά αλλά και από περισσότερο λεπτομερή βήματα τα οποία χρησιμοποιούνται ανάλογα με τα εργαλεία που έχουν χρησιμοποιηθεί καθώς και με τη μεθοδολογία/τεχνολογία (FPGA/ASIC). Μερικά από τα εργαλεία/λογισμικά που χρησιμοποιούνται στη σχεδίαση back-end είναι:

- Cadence (Cadence Encounter RTL Compiler, Encounter Digital Implementation, Cadence Voltus IC Power Integrity Solution, Cadence Tempus Timing Signoff Solution)
- Synopsys (Design Compiler, IC Compiler)
- Magma (BlastFusion, etc.)
- Mentor Graphics (Olympus SoC, IC-Station, Calibre)

Τα βασικά βήματα μιας ASIC ροής της φυσικής σχεδίασης είναι τα εξής :

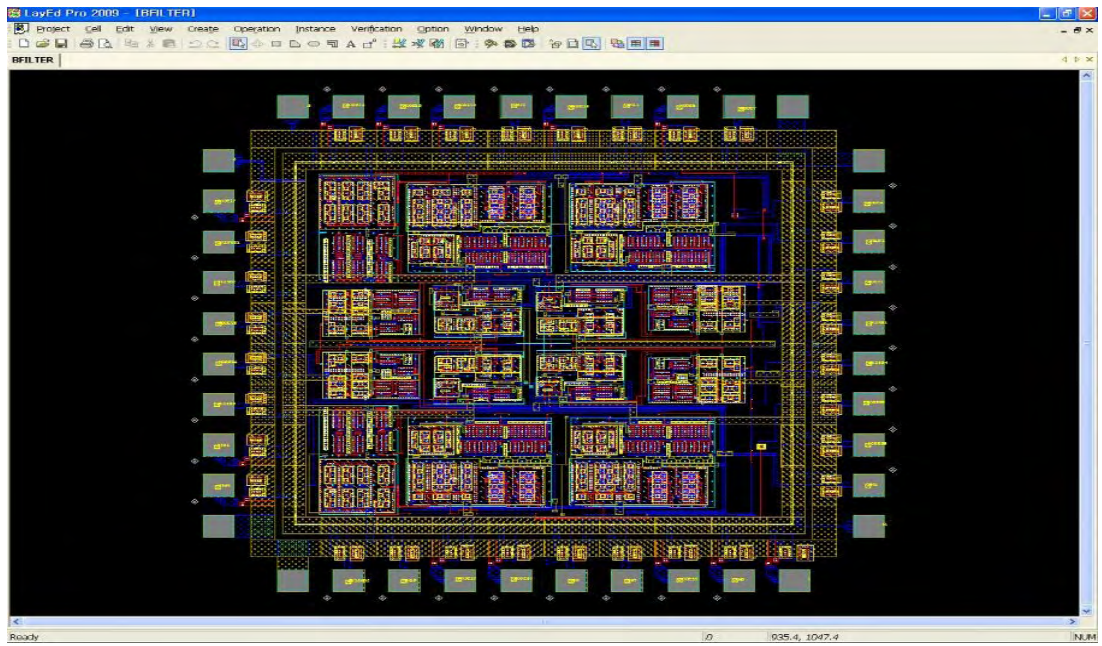
1. Design netlist : Η φυσική σχεδίαση βασίζεται σε ένα netlist το οποίο είναι το τελικό αποτέλεσμα της διαδικασίας σύνθεσης, δηλαδή την μετατροπή κάποιας γλώσσας περιγραφής υλικού (HDL) σε αντίστοιχες γλώσσες περιγραφής σε επίπεδο πηλών(π.χ. VHDL, Verilog). Το netlist περιέχει μεταξύ άλλων πληροφορίες σχετικά με τα cells που χρησιμοποιήθηκαν, την περιοχή τους και τις διασυνδέσεις μεταξύ τους. Δύο εργαλεία που συναντάμε συχνά στην σύνθεση ψηφιακών κυκλωμάτων είναι ο Synopsys Design Compiler και ο Cadence RTL Compiler. Κατά την διαδικασία της σύνθεσης εφαρμόζονται κάποιοι περιορισμοί ώστε να εξασφαλιστεί ότι ο σχεδιασμός ικανοποιεί την απαιτούμενη ταχύτητα και λειτουργικότητα. Στη συνέχεια γίνεται επαλήθευση για την λειτουργικότητα του κυκλώματος και το κύκλωμα προωθείται στην φυσική σχεδίαση.
2. Floorplaning : Το floorplaning αποτελεί το πρώτο ουσιαστικό βήμα ροής της φυσικής σχεδίασης και είναι η διαδικασία εντοπισμού των δομών οι οποίες

θα πρέπει τοποθετηθούν και στην συνέχεια να κατανεμηθούν, στον εκάστοτε χώρο, με τέτοιο τρόπο ώστε να πληρούν κάποιους στόχους έχοντας ως αποτέλεσμα τις απαιτούμενες επιδόσεις. Στόχος είναι η βελτιστοποίηση του χώρου ώστε να εξασφαλίζεται η ελάχιστη έκταση στον σχεδιασμό και κατ' επέκταση η αύξηση της ταχύτητας του κυκλώματος, καθώς στις περισσότερες μεθοδολογίες σχεδιασμού ο χώρος και η ταχύτητα του κυκλώματος είναι δύο έννοιες άρρηκτα συνδεδεμένες μεταξύ τους. Αυτό οδηγεί σε μικρότερες αποστάσεις διασύνδεσης, χρήση όσο το δυνατόν λιγότερων πόρων, γρηγορότερες διαδρομές σήματος από το ένα άκρο στο άλλο και ακόμη πιο γρήγορη και συνεπή τοποθέτηση και δρομολόγηση (place and route). Ο πλήρης καθορισμός του χώρου πραγματοποιείται δυναμικά καθιστώντας έτσι πολλές φορές αναγκαίο τον επαναπροσδιορισμό της αρχιτεκτονικής της σχεδίασης.

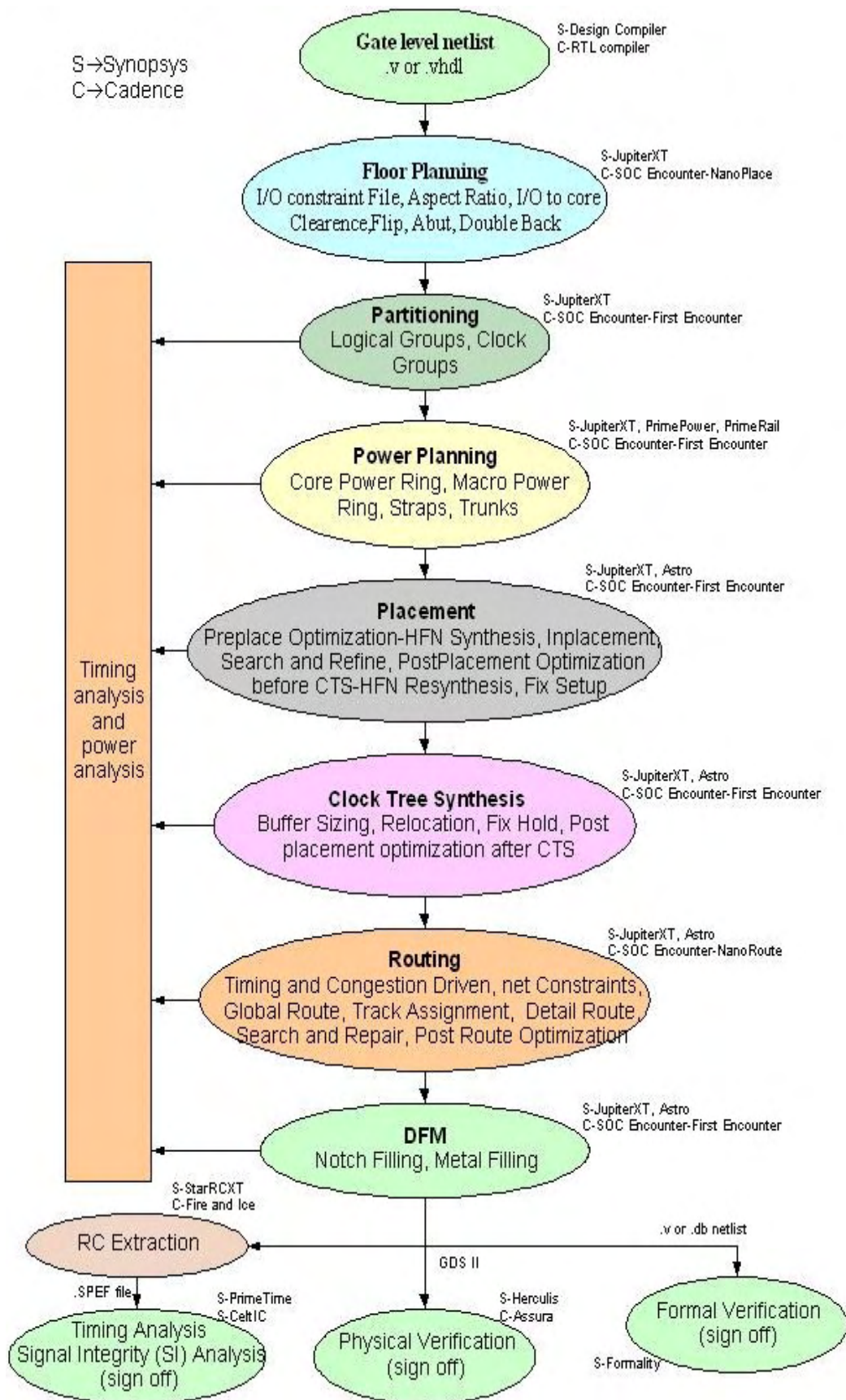
3. Partitioning : Το partitioning είναι μία διαδικασία διαίρεσης του τσιπ σε μικρότερα τμήματα από τον ίδιο τον μηχανικό σχεδιασμού με στόχο την ευκολότερη διαχείριση του χώρου διευκολύνοντας έτσι παράλληλα κάποιες παρακάτω διαδικασίες ροής της φυσικής σχεδίασης, όπως το place and route. Το είδος του διαχωρισμού αυτού αναφέρεται κυρίως ως λογικός διαχωρισμός (Logical Partitioning).
4. Placement : Το placement έχει ως στόχο την βέλτιστη τοποθέτηση των στοιχείων στον προκαθορισμένο χώρο της σχεδίασης. Πριν ξεκινήσει η διαδικασία βελτιστοποίησης όλα τα μοντέλα φορτίου του καλωδίου (Wire Load Models-WLM) που υπήρχαν κατά την διαδικασία της σύνθεσης αφαιρούνται και η RC καθυστέρηση υπολογίζεται από την εικονική δρομολόγηση (Virtual Route-VR) η οποία εφαρμόζει απόσταση Manhattan μεταξύ δύο σημείων. Η VR είναι πιο ακριβής από το WLM.
Η τοποθέτηση (placement) γίνεται σε τέσσερις φάσεις βελτιστοποίηση:
 - *Pre-placement Optimization* :
 - *In-placement optimization* :
 - *Post placement optimization* :
 - *Post placement optimization after Clock Tree Synthesis (CTS)*

5. Clock Tree Synthesis : Κατά τη διαδικασία του Clock Tree Synthesis παράγεται το δέντρο ρολογιού και στόχος είναι να επιτευχθεί η βελτιστοποίηση του skew και insertion delay. Μέχρι τώρα το ρολόι θεωρούνταν ιδανικό χωρίς καθυστέρηση. Σε αυτό το βήμα υπολογίζεται με μεγαλύτερη ακρίβεια ο χρονισμός του κυκλώματος. Υπάρχουν μονοπάτια που υπολογίζονται στο χρονισμό του κυκλώματος και άλλα που δεν το επηρεάζουν (false paths).
6. Routing : Υπάρχουν δύο είδη δρομολόγησης στη διαδικασία της φυσικής σχεδίασης, το global routing και το detailed routing. Το global routing δεσμεύει πηγές δρομολόγησης οι οποίες χρησιμοποιούνται για τις συνδέσεις μεταξύ των στοιχείων. Το detailed routing αναθέτει μονοπάτια δρομολόγησης σε συγκεκριμένα επίπεδα μετάλλου.
7. Physical verification : Το physical verification ελέγχει την ορθότητα του σχεδίου του κυκλώματος (layout). Ουσιαστικά επαληθεύει ότι το layout:
- Συμμορφώνεται πλήρως με όλους τους τεχνολογικούς κανόνες που έχουν τεθεί (Design Rule Checking-DRC)
 - Έχει την ίδια συμπεριφορά με το αρχικό netlist (Layout vs. Schematic-LVS)
 - Δεν επηρεάζεται από antenna effects (Antenna Rule Checking)
 - Συμμορφώνεται πλήρως με όλους τους ηλεκτρικούς κανόνες που έχουν τεθεί (Electrical Rule Checking-ERC)

Τέλος, το physical verification περιλαμβάνει τον έλεγχο της πυκνότητας σε όλα τα επίπεδα του chip.



Εικόνα 2: Φυσική Σχεδίαση Κυκλώματος



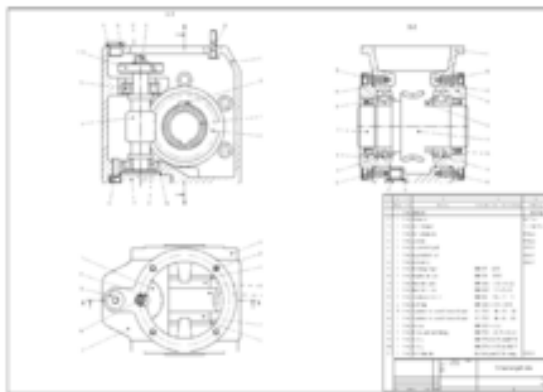
Εικόνα 3: Ροή Φυσικής Σχεδίασης

1.2 Εργαλεία CAD (Computer Aided Design)

Το CAD είναι η χρήση των υπολογιστικών συστημάτων που βοηθάει στη δημιουργία, μετατροπή, ανάλυση ή βελτιστοποίηση της σχεδίασης. Το λογισμικό CAD χρησιμοποιείται με στόχο να αυξήσει την παραγωγικότητα του σχεδιαστή, να βελτιώσει την ποιότητα της σχεδίασης, να βελτιώσει την επικοινωνία μέσω εγγράφων και να δημιουργήσει μια βάση δεδομένων για τη διαδικασία της κατασκευής. Η έξοδος του εργαλείου CAD έχει συνήθως τη μορφή ηλεκτρονικών αρχείων για εκτύπωση αλλά και για άλλες χρήσεις.

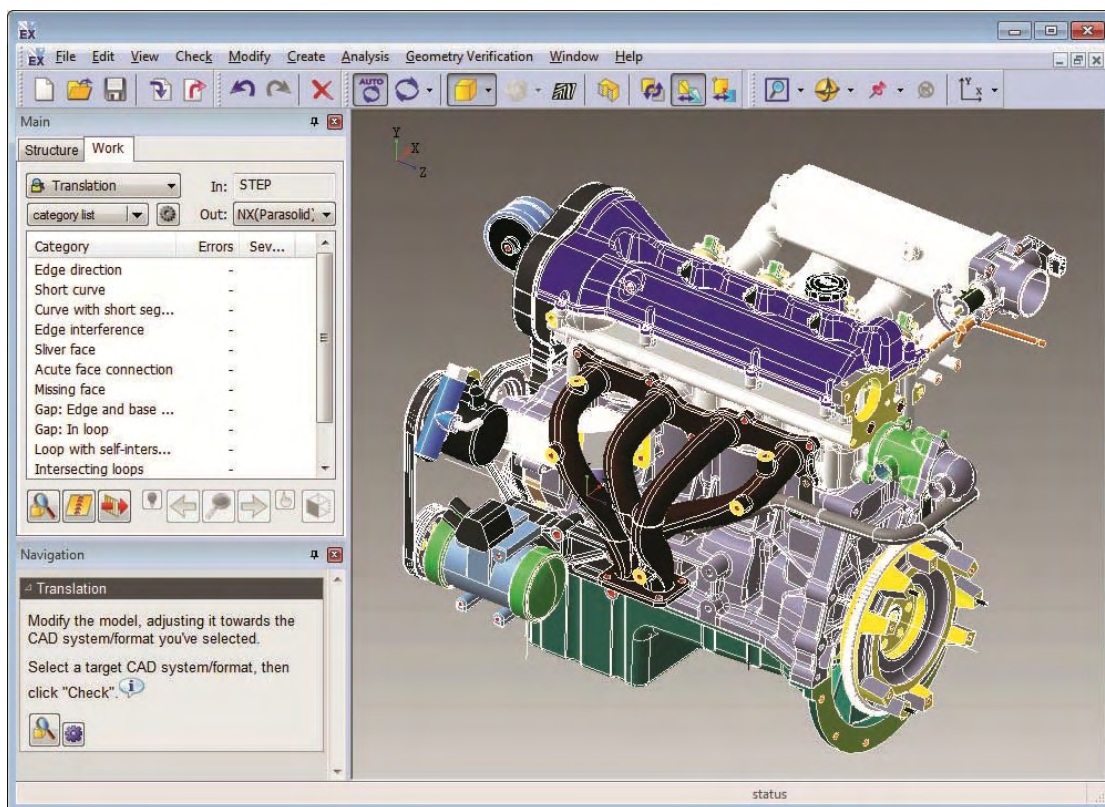
Το CAD έχει εφαρμογή σε πολλά επιστημονικά πεδία. Η χρήση του στη σχεδίαση ηλεκτρονικών συστημάτων είναι γνωστή και ως αυτοματοποιημένη ηλεκτρονική σχεδίαση (electronic design automation – EDA). Στη μηχανική σχεδίαση είναι γνωστή και ως αυτοματοποιημένη μηχανική σχεδίαση (mechanical design automation – MDA) ή CAD, η οποία περιλαμβάνει την διαδικασία δημιουργίας ενός τεχνικού σχεδίου με τη βοήθεια ενός λογισμικού.

Το λογισμικό CAD γενικότερα χρησιμοποιεί είτε διανυσματικά γραφικά για να αποτυπώσει τα πρότυπα αντικείμενα σχεδίασης, είτε μπορεί να παράγει rasters graphics δείχνοντας την γενική εμφάνιση των υπό σχεδίαση αντικειμένων. Παρόλα αυτά, δεν περιλαμβάνει μόνο σχήματα. Σύμφωνα με το εγχειρίδιο σύνταξης των τεχνικών και μηχανολογικών σχεδίων, η έξοδος του CAD πρέπει να περιέχει πληροφορίες που αφορούν υλικά, διαδικασίες καθώς και ανοχές σύμφωνα με τις συμβάσεις που ισχύουν για την συγκεκριμένη εφαρμογή.



Εικόνα 4: 2D CAD σχεδίαση

Το CAD μπορεί να χρησιμοποιηθεί τόσο για την σχεδίαση καμπυλών και σχεδίων σε χώρο δύο διαστάσεων, όσο και για την σχεδίαση καμπυλών, επιφανειών και στερεών σε τρισδιάστατο χώρο. Πρόκειται για μία πολύ σημαντική βιομηχανική τέχνη η οποία έχει πολλές εφαρμογές σε πολλούς κλάδους όπως στην αυτοκινητοβιομηχανία, στην ναυπηγική βιομηχανία, στον αρχιτεκτονικό σχεδιασμό και σε πολλούς άλλους. Λόγω της τεράστιας οικονομικής σημασίας του το CAD υπήρξε σημαντική κινητήρια δύναμη για την έρευνα στην υπολογιστική γεωμετρία, στα γραφικά υπολογιστών, τόσο στο hardware όσο και στο software, και σε άλλους τομείς.



Εικόνα 5: 3D CAD Σχέδιο κινητήρα

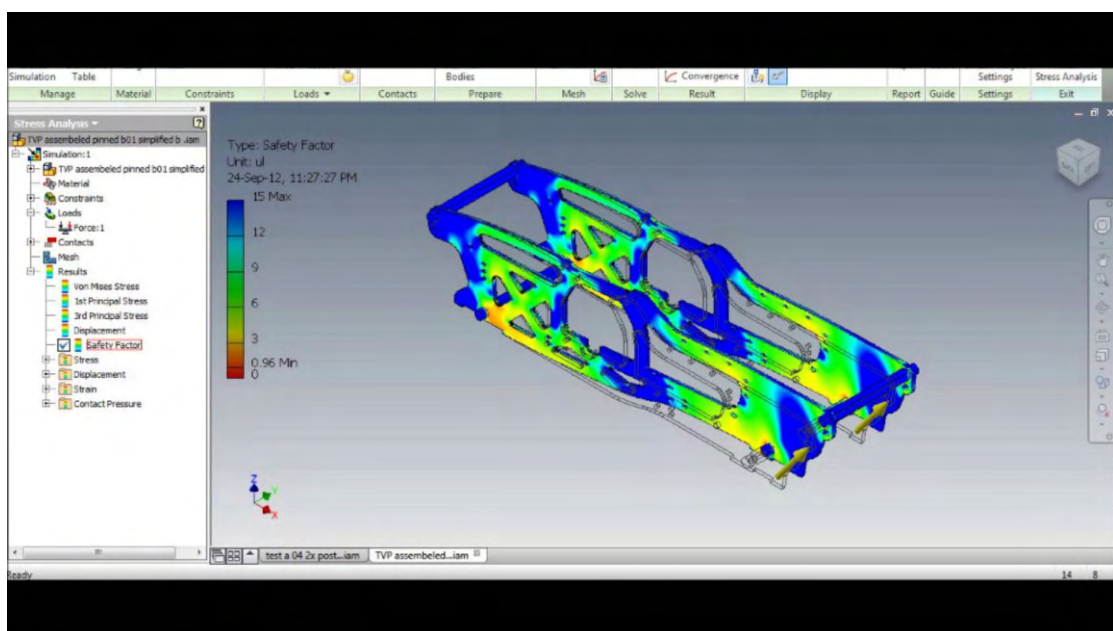
1.2.1 Χρήσεις του CAD

Το CAD είναι ένα από τα πολλά εργαλεία που χρησιμοποιούνται από τους μηχανικούς και τους σχεδιαστές. Χρησιμοποιείται με πολλούς τρόπους, ανάλογα με το επάγγελμα του χρήστη και τον τύπο του εν λόγω λογισμικού.

Το CAD αποτελεί ένα μέρος της συνολικής δραστηριότητας Digital Product Development(DPD) στο πλαίσιο των διαδικασιών του Product Lifecycle Management (PLM) και ως εκ τούτου χρησιμοποιείται σε συνδυασμό με άλλα εργαλεία όπως:

- Computer-aided engineering (CAE) και Finite element analysis (FEA)
- Computer-aided manufacturing (CAM) συμπεριλαμβανομένων των οδηγιών για τις Computer Numerical Control (CNC) μηχανές
- Photo realistic rendering και Motion Simulation
- Product Data Management (PDM) για την διαχείριση των εγγράφων και τον έλεγχο αναθεώρησης

Το CAD έχει αποδειχθεί χρήσιμο στους μηχανικούς καθώς χρησιμοποιεί τέσσερις ιδιότητες το ιστορικό της κατασκευής, τα χαρακτηριστικά της, τις παραμέτρους της και τους υψηλού επιπέδου περιορισμούς της. Το ιστορικό κατασκευής μπορεί να χρησιμοποιηθεί για να ανατρέξουμε σε ειδικά χαρακτηριστικά του μοντέλου ώστε να εργαστούμε σε μια συγκεκριμένη υπομονάδα του και όχι σε ολόκληρο το μοντέλο. Οι παράμετροι και οι περιορισμοί μπορούν να χρησιμοποιηθούν για να καθοριστεί το μέγεθος, το σχήμα και άλλα χαρακτηριστικά των διαφόρων στοιχείων μοντελοποίησης.



Εικόνα 6:Σχέδιο που έχου εισαχθεί παράμετροι ασφαλείας

1.2.2 Τεχνολογία του CAD

Αρχικά το hardware για το εργαλείο CAD αναπτύχθηκε σε γλώσσες προγραμματισμού όπως Fortran και ALGOL, το οποίο όμως με την εξέλιξη αντικειμενοστραφών γλωσσών άλλαξε ριζικά. Σήμερα ένα CAD σύστημα μπορεί να δημιουργηθεί από ένα graphical user interface (GUI), το οποίο βασίζεται σε modules της C με τα APIs και μία αναπαράσταση ορίων χρησιμοποιώντας πυρήνα γεωμετρικών μοντέλων. Τα σύγχρονα CAD εργαλεία υπάρχουν για όλες τις μεγάλες πλατφόρμες (Windows, Linux, UNIX και Mac OS X) και κάποια από αυτά διαθέτουν υποστήριξη για πολλαπλές πλατφόρμες. Δεν απαιτείται ειδικό hardware για την υποστήριξη CAD εργαλείων αλλά υπάρχει πιθανότητα να χρειάζεται ισχυρή κάρτα γραφικών, επεξεργαστής καθώς και μεγάλη χωρητικότητα σε RAM για απαιτητικές γραφικές και υπολογιστικές διεργασίες.

1.3 Εργαλεία EDA (Electronic Design Automation)

Η αυτοματοποιημένη ηλεκτρονική σχεδίαση (Electronic Design Automation-EDA) είναι μια κατηγορία εργαλείων λογισμικού, τα οποία χρησιμοποιούνται για το σχεδιασμό ηλεκτρονικών συστημάτων, όπως πλακέτες τυπωμένων κυκλωμάτων (printed circuit boards) και ολοκληρωμένων κυκλωμάτων (integrated circuits). Τα εργαλεία δουλεύουν μαζί σε μια ροή σχεδίασης την οποία οι σχεδιαστές χρησιμοποιούν με στόχο να σχεδιάσουν και να αναλύσουν ολόκληρο το τσιπ. Το EDA αναφέρεται συχνά και ως Electronic Computer-Aided Design (ECAD).

1.3.1 Η ιστορία του EDA

Πριν το EDA τα ολοκληρωμένα κυκλώματα σχεδιάζονταν με το χέρι. Πιο συγκεκριμένα, η διαδικασία του σχεδιασμού βασιζόταν κυρίως στην γραφική απεικόνιση του κυκλώματος και η μετατροπή από την ηλεκτρονική αναπαράσταση του στην γραφική γινόταν με το χέρι. Η πιο γνωστή εταιρία

εκείνης της εποχής ήταν η Calma η οποία δημιούργησε το format GDSII το οποίο χρησιμοποιείται μέχρι και σήμερα.

Από τα μέσα της δεκαετίας του 1970, οι προγραμματιστές άρχισαν να αυτοματοποιούν την σχεδίαση αναπτύσσοντας τα πρώτα εργαλεία τοποθέτησης και δρομολόγησης (place and route). Τα πρακτικά των επιστημονικών συνεδρίων της εποχής παρουσιάζουν αναλυτικά και καλύπτουν ένα μεγάλο μέρος της ροής σχεδίασης των ολοκληρωμένων κυκλωμάτων.

Η επόμενη εποχή για το EDA ξεκινά την περίοδο της δημοσίευσης του πρωτοποριακού άρθρου "Εισαγωγή σε VLSI συστήματα", το κείμενο του οποίου υποστήριζε τον σχεδιασμό τσιπ με γλώσσες προγραμματισμού που μεταγλωττίζονται σε υλικό. Το άμεσο αποτέλεσμα ήταν η σημαντική αύξηση της πολυπλοκότητας των τσιπ που μπορούσαν να σχεδιαστούν, με ιδιαίτερη βελτίωση στην σχεδίαση εργαλείων επαλήθευσης που χρησιμοποιούνται στην λογική προσομοίωση. Η αντίληψη πως η λειτουργικότητα του κυκλώματος μπορεί να περιγραφεί σε γλώσσα προγραμματισμού και να σχεδιαστεί λεπτομερώς φυσικά από εργαλείο αποτελεί τη βάση της ψηφιακής σχεδίασης IC και παραμένει μέχρι σήμερα αναλλοίωτη, παρά το γεγονός ότι οι γλώσσες και τα εργαλεία έχουν εξελιχθεί.

Τα πρώτα εργαλεία EDA παρήχθησαν σε ακαδημαϊκό περιβάλλον. Ένα από τα πιο διάσημα ήταν το "Berkeley VLSI Εργαλεία Tarball", το οποίο περιείχε ένα σύνολο εφαρμογών UNIX χρήσιμες για το σχεδιασμό συστημάτων VLSI.

Το 1981 σηματοδοτεί την αρχή του εμπορικού EDA. Μέχρι τότε πολλές εταιρίες όπως οι Hewlett Packard, Tektronix, και Intel παρήγαγαν το EDA εσωτερικά αλλά managers και προγραμματιστές ξέφυγαν από τις εν λόγω εταιρείες με σκοπό να ασχοληθούν με το EDA σε επιχειρηματικό επίπεδο οδηγώντας έτσι στην δημιουργία εταιριών όπως οι Daisy Systems, Mentor Graphics και Valid Logic Systems. Μέσα σε λίγα χρόνια υπήρχαν πολλές εταιρείες που ειδικεύονται στο EDA, η κάθε μία με ειδίκευση σε διαφορετικό τομέα.

1.3.3 Εφαρμογές του EDA

Το EDA βρίσκει εφαρμογή σε διάφορους τομείς που αφορούν τα ολοκληρωμένα κυκλώματα. Οι σημαντικότεροι από αυτούς είναι :

A. ΣΧΕΔΙΑΣΜΟΣ (DESIGN)

- High Level Synthesis
- Logic Synthesis
- Schematic Capture
- Layout

B. ΠΡΟΣΟΜΟΙΩΣΗ (SIMULATION)

- Transistor simulation
- Logic simulation
- Behavioral simulation
- Hardware emulation
- Technology CAD
- Electromagnetic field solvers

C. ΑΝΑΛΥΣΗ ΚΑΙ ΕΠΑΛΗΘΕΥΣΗ (ANALYSIS & VERIFICATION)

- Functional verification
- Clock Domain Crossing Verification
- Formal verification
- Equivalence checking
- Static timing analysis
- Physical verification

D. ΚΑΤΑΣΚΕΥΗ (MANUFACTURING)

- Mask data preparation
- Resolution enhancement techniques
- Optical proximity correction
- Mask generation
- Automatic test pattern generation
- Built-in self-test

2. Placement

Το placement αποτελεί ένα σημαντικό βήμα της αυτοματοποιημένης ηλεκτρονικής σχεδίασης. Έχει ως αντικείμενο το τμήμα της φυσικής ροής του σχεδιασμού που αποδίδει ακριβείς θέσεις στα διάφορα στοιχεία του κυκλώματος μέσα στο τσιπ. Αυτή η ακριβής θέση που θα δοθεί σε κάθε στοιχείο είναι σημαντική γιατί από αυτή θα εξαρτηθεί ο χρονισμός του κυκλώματος. Στη σύγχρονη εποχή, όπου η συνεχής κλιμάκωση μας έχει οδηγήσει στα deep submicron ηλεκτρονικά, το μήκος της εσωτερικής διασύνδεσης των πυλών καθώς και η καθυστέρηση που αυτό έχει, παίζει πολύ σημαντικό ρόλο στην τελική απόδοση του κυκλώματος.

Ο κύριος στόχος του placement είναι να μειωθεί το συνολικό μήκος καλωδίου που χρησιμοποιείται για τη διασύνδεση των πυλών. Καταφέροντας να μειωθεί αυτό μπορεί να μειωθεί η καθυστέρηση και η κατανάλωση του κυκλώματος, καθώς βελτιώνεται και η δυνατότητα δρομολόγησης αυτού. Υπάρχουν διάφοροι τρόποι μοντελοποίησης του μήκους καλωδίου για τη διασύνδεση των πυλών και θα τους αναφέρουμε σε επόμενη παράγραφο. Η δρομολόγηση είναι ένα βήμα μετά το placement. Στο τέλος δεν υπάρχει λόγος να παραχθεί ένα κύκλωμα το οποίο δεν έχει πιάσει το χρονισμό ή δεν έχει γίνει δρομολόγηση των πυλών.

Μία κακή ανάθεση κατά την διαδικασία του placement μπορεί να έχει ως αποτέλεσμα όχι μόνο την μείωση της απόδοσής του τσιπ αλλά και να καταστήσει αδύνατη την κατασκευή του. Αυτό οφείλεται στο γεγονός ότι απαιτούνται υπέρμετρα μήκη καλωδίων (wirelength) το οποίο μπορεί να ξεπεράσει τους διαθέσιμους πόρους δρομολόγησης. Συνεπώς, η ανάθεση θα πρέπει να πραγματοποιείται βελτιστοποιώντας παράλληλα μια σειρά από στόχους ώστε να εξασφαλιστεί πως το κύκλωμα ικανοποιεί τις απαιτήσεις των επιδόσεών του.

2.1 Διατύπωση του προβλήματος

Ο στόχος του placement είναι να βρεί τη βέλτιστη θέση για κάθε στοιχείο του κυκλώματος, σε μία συγκεκριμένη περιοχή έτσι ώστε να βελτιστοποιήσει μία συνάρτηση κόστους.

Πιο συγκεκριμένα δίνονται:

- Ένα netlist απο πύλες μιας βιβλιοθήκης ημιαγωγών.
- Μία μαθηματική έκφραση του netlist ως διάνυσμα.
- Η θέση των pins εισόδου/εξόδου καθώς και διαστάσεις του χώρου για το κύκλωμα.
- Ένα από τα επόμενα: περιορισμοί χρονισμού, μία λίστα από κρίσιμα μονοπάτια, περιορισμοί κατανάλωσης.

Η διαδικασία του placement επιστρέφει:

- Θέση των κελιών έτσι ώστε το κύκλωμα να εναρμονίζεται με τους περιορισμούς
- Δύο κελιά δεν επικαλύπτονται.
- Το κύκλωμα είναι δρομολογήσιμο

Οι κύριοι στόχοι του placement περιλαμβάνουν:

- Το συνολικό wirelength : Ο κύριος στόχος της διαδικασίας του placement είναι η ελαχιστοποίηση του συνολικού wirelength ή του αθροίσματος του μήκους όλων των καλωδίων του κυκλώματος. Αυτό όχι μόνο ελαχιστοποιεί το μέγεθος του τσιπ, άρα και το κόστος, αλλά ελαχιστοποιεί και την κατανάλωση ισχύος και την καθυστέρηση οι οποίες είναι ανάλογες του wirelength.
- Τον χρονισμό (Timing) : Ο κύκλος ρολογιού ενός τσιπ προσδιορίζεται από την καθυστέρηση του κρίσιμου μονοπατιού του (critical path), το οποίο συνήθως είναι το μακρύτερο μονοπάτι. Δεδομένων κάποιων προδιαγραφών απόδοσης δεν θα πρέπει να υπάρχει κανένα μονοπάτι, η

καθυστερήση του οποίου να ξεπερνάει το μέγιστο προκαθορισμένο όριο.

- Την συμφόρηση (Congestion) : Ενώ είναι αναγκαία η ελαχιστοποίηση του συνολικού wirelength, είναι επίσης αναγκαίο να μην υπάρχει συμφόρηση καλωδίων σε μια μικρή περιοχή του τσιπ, γιατί αυτό μπορεί να οδηγήσει σε πολλές παρακάμψεις δρομολόγησης, ή ακόμα και στο να καταστήσει αδύνατη την πραγματοποίηση όλων των δρομολογήσεων.
- Την κατανάλωση (Power) : Η ελαχιστοποίηση της κατανάλωσης ουσιαστικά περιλαμβάνει τη διανομή των περιοχών των στοιχείων των κελιών με τέτοιο τρόπο ώστε να μειώνεται η συνολική κατανάλωση ισχύος, να μειώνει τα hot spots και να ομαλοποιεί τις μεταβολές της θερμοκρασίας.
- Ένας δευτερεύον στόχος είναι η ελαχιστοποίηση του χρόνου πραγματοποίησης της διαδικασίας τοποθέτησης.

Το βέλτιστο είναι να ελαχιστοποιηθεί κάθε συνάρτηση, αλλά αυτό είναι αδύνατο καθώς μερικές συναρτήσεις είναι αντικρουόμενες, επομένως όταν ελαχιστοποιείται μία μεταβλητή του προβλήματος η άλλη μεταβλητή μεγιστοποιείται. Για παράδειγμα όταν μειώνεται η καθυστέρηση του κυκλώματος τότε η κατανάλωση ισχύος αυξάνεται. Μία καλή λύση του προβλήματος είναι αυτή που θα βελτιστοποιεί τις μεταβλητές που της έχουν δοθεί, αλλά θα έχει τηρήσει και τις ισορροπίες και στις υπόλοιπες κρίσιμες μεταβλητές.

Ένα άλλο πρόβλημα είναι ότι η εξέλιξη της τεχνολογίας αλλάζει και τους στόχους των συναρτήσεων υπολογισμού κόστους. Για παράδειγμα στα 90nm τρανζίστορ η καθυστέρηση προέρχονταν κυρίως από την καθυστέρηση των πυλών, αλλά τώρα η συνολική καθυστέρηση του κυκλώματος φαίνεται να μετατοπίζεται προς την καθυστέρηση των καλωδίων διασύνδεσης, για αυτό και συνολικό μήκος καλωδίου είναι ένα πολύ σημαντικό κριτήριο για τις συναρτήσεις υπολογισμού κόστους (cost function).

Το πρόβλημα του placement είναι ένα NP-hard πρόβλημα, που σημαίνει ότι δεν μπορούμε να έχουμε μία λύση σε πεπερασμένο χρόνο. Δεν θα ασχοληθούμε με την απόδειξη αυτή, καθώς δεν είναι στους στόχους της διπλωματικής αυτής εργασίας.

2.2 Το placement εντός της EDA ροής σχεδίασης

Κατά την διαδικασία του placement παίρνουμε ένα δεδομένο netlist και σε συνδυασμό με κάποια βιβλιοθήκη της εκάστοτε τεχνολογίας παράγεται μία έγκυρη διάταξη placement. Η διάταξη έχει βελτιστοποιηθεί σύμφωνα με τους προαναφερθέντες στόχους και είναι έτοιμη για αλλαγές και ρυθμίσεις του μεγέθους των κελιών το οποίο αποτελεί απαραίτητο βήμα για την ικανοποίηση της ακεραιότητας του χρονισμού και το σήματος. Ακολουθούν το Clock-tree synthesis και η δρομολόγησης, ολοκληρώνοντας έτσι την διαδικασία της φυσικής σχεδίασης. Σε πολλές περιπτώσεις, μέρη της ροής της φυσικής σχεδίασης ή ακόμα και ολόκληρη η φυσική σχεδίαση επαναλαμβάνεται αρκετές φορές έως ότου επιτευχθεί ο επιθυμητός σχεδιασμός.

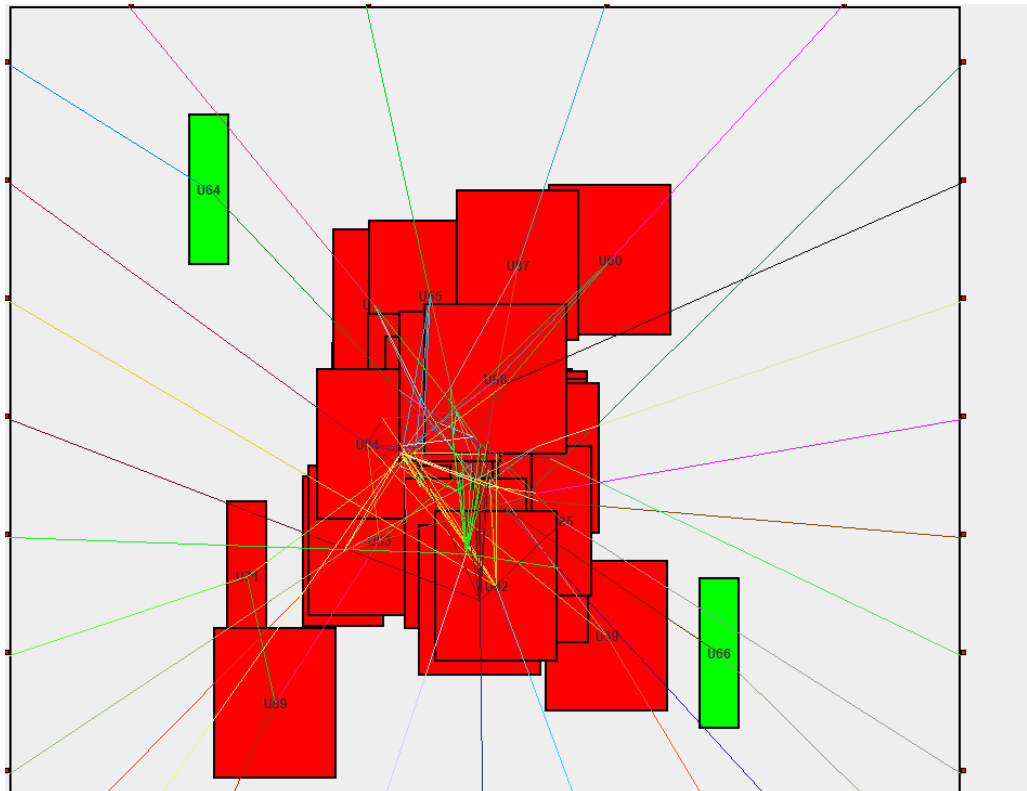
Στην περίπτωση των ολοκληρωμένων κυκλωμάτων ASICs, ο πυρήνας του χώρου σχεδίασης του τσιπ περιλαμβάνει έναν αριθμό γραμμών με σταθερό ύψος με λίγο ή καθόλου κενό μεταξύ τους. Κάθε γραμμή αποτελείται από θέσεις που καταλαμβάνονται από τα στοιχεία του κυκλώματος. Τα στοιχεία του κυκλώματος είναι πρότυπες πύλες, macro blocks, ή επιθέματα εισόδου εξόδου. Οι πρότυπες πύλες έχουν ένα σταθερό ύψος ίσο με το ύψος μιας γραμμής, αλλά διαφέρουν μεταξύ τους στο πλάτος, το οποίο αποτελείται από έναν ακέραιο αριθμό θέσεων. Από την άλλη πλευρά, τα blocks είναι συνήθως μεγαλύτερα από τις πύλες και έχουν μεταβλητά ύψη που καταλαμβάνουν πολλαπλό αριθμό σειρών. Τα blocks μπορούν να τοποθετηθούν είτε εκ των προτέρων είτε κατά την διάρκεια του placement. Όταν τα blocks έχουν προκαθορισμένες θέσεις πχ από το βήμα floorplanning process της φυσικής σχεδίασης τότε περιορίζουν τον placer να αναθέσει θέσεις στα κελιά. Αλλιώς τοποθετούνται μαζί με τα πρότυπα κελιά κι αυτό ονομάζεται mixed-cell placement.

2.3 Ταξινόμηση των αλγορίθμων placement

Το πρόβλημα του placement μπορεί να χωριστεί σε 2 κατηγορίες:

- Καθολικό placement(global placement):

Είναι το 1^ο βήμα του Placement στο οποίο χρησιμοποιούνται κατασκευαστικοί ή επαναληπτικοί αλγόριθμοι, με σκοπό να βρεθεί η κοντινότερη στο βέλτιστο θέση των κελιών. Σε αυτό το βήμα τα κελιά τοποθετούνται σε όλο το διαθέσιμο χώρο για το placement και υπάρχουν διάφοροι αλγόριθμοι που θα αναφέρουμε παρακάτω. Συνήθως ο χώρος χωρίζεται σε μικρότερα κομμάτια για να σπάει το πρόβλημα. Σε αυτό το βήμα μπορεί να υπάρχει επικάλυψη μεταξύ των κελιών.



Εικόνα 8: Καθολικό placement

- Τελικό placement(final placement):

Αναφέρεται ως και λεπτομερές placement. Έχει ως είσοδο το καθολικό placement κόβοντας την περιοχή σε μικρότερα κομμάτια εξετάζει τη βέλτιστη θέση των κελιών τοπικά. Συνήθως έχει ως έξοδο ένα legal placement(χωρίς επικάλυψη μεταξύ των κελιών). Εάν το output δεν είναι legal τότε ένας

placement και το λύνουν τοπικά με ακριβείς ή ευριστικούς τρόπους. Οι αλγόριθμοι αυτού του τύπου μπορούν να διακριθούν σε 2 κατηγορίες ανάλογα με το εάν εφαρμόζουν τυχαίους ή ντετερμινιστικούς τρόπους επίλυσης του προβλήματος. Οι επαναληπτικοί αλγόριθμοι που βασίζονται σε τυχαία κριτήρια επιλογής δεν απορρίπτουν ποτέ καλύτερες λύσεις αλλά αποδέχονται και χειρότερες λύσεις με λιγότερες πιθανότητες. Με αυτόν τον τρόπο μπορούν να ξεφύγουν από τοπικά ελάχιστα και να βρουν τη βέλτιστη λύση. Αυτό όμως δεν είναι πάντα πρακτικό, ιδίως για μεγάλα κυκλώματα.

Ένας άλλος διαχωρισμός των αλγορίθμων placement γίνεται επίσης και από την έξοδο που παράγουν. Μπορούν να διαχωριστούν, δηλαδή, σε ντετερμινιστικούς ή πιθανολογικούς αλγορίθμους ανάλογα με το εάν παράγουν το ίδιο αποτέλεσμα στην έξοδο όσες φορές και εάν εφαρμοστούν σε ένα κύκλωμα, ή εάν παράγουν διαφορετικά αποτελέσματα κάθε φορά ανάλογα.

2.4 Μετρήσεις Καλωδίων

Μετά την εφαρμογή της διαδικασίας του placement κατά την σχεδίαση, τα κελιά έχουν καθορισμένες συντεταγμένες μέσα στην περιοχή του πυρήνα. Αυτό σημαίνει ότι σχεδόν το συνολικό μήκος της καλωδίωσης του κυκλώματος μπορεί να υπολογιστεί, αθροίζοντας το μήκος κάθε καλωδίου. Το γεγονός ότι οι αποστάσεις μεταξύ των καλωδίων δεν είναι συγκεκριμένες πριν την διαδικασία του routing, καθιστά αναγκαίο τον ορισμό μερικών μετρητών οι οποίοι υπολογίζουν και βελτιστοποιούν τα wirelength (netlength metrics). Θα πρέπει να αναφερθεί πως στους μετρητές απόστασης τα κελιά ορίζονται ως σημεία και για αυτά τα σημεία χρησιμοποιούνται είτε οι συντεταγμένες του κατώτερου αριστερού σημείου είτε του κέντρου τους.

2.4.1 L_k -norm επαγόμενη μετρική απόσταση

$$dk(p, q) = \sqrt[k]{|p_x - q_x|^k + |p_y - q_y|^k}, \quad p, q \in R^2, \quad k \in \mathbb{N}$$

Έστω ότι τα p και q είναι δύο modules, με συντεταγμένες p_x , p_y , q_x , q_y στους άξονες x και y αντίστοιχα. L_1 είναι η απόσταση Manhattan και L_2 είναι

η Ευκλείδεια απόσταση, οι οποίες χρησιμοποιούνται συχνότερα ως μετρητές απόστασης καλωδίωσης. Επίσης, όταν $k \rightarrow \infty$ δίνει ως αποτέλεσμα:

$$d_q(p, q) = \lim_{k \rightarrow \infty} d_k(p, q) = \max(|p_x - q_x|, |p_y - q_y|)$$

2.4.2 Steiner-tree και ελάχιστο επικαλύπτων δέντρο

Δεδομένων n στοιχείων στο επίπεδο απαιτείται η διασύνδεσή τους να πραγματοποιείται με το ελάχιστο δυνατό δίκτυο καλωδίων, το οποίο αποτελείται μόνο από οριζόντια και κάθετα ευθύγραμμα τμήματα. Μπορεί να αποδειχτεί ότι ένα τέτοιο δίκτυο καλωδίωσης ουσιαστικά αποτελεί ένα δέντρο του οποίου οι κορυφές είναι τα σημεία εισόδου μαζί με κάποια άλλα επιπλέον σημεία (Steiner points).

Υποθέτοντας ότι μέχρι σήμερα χρησιμοποιούνται μόνο οριζόντια και κάθετα καλώδια, το ελάχιστο δυνατό netlength επιτυγχάνεται συνδέοντας τα pins του κάθε καλωδίου με ένα δέντρο Steiner με τέτοιο τρόπο ώστε να μην τέμνονται τα δέντρα μεταξύ τους. Αν και το routing ενός ολόκληρου net με πολλαπλούς κόμβους απεικονίζεται καλύτερα με ένα ευθύγραμμο δέντρο Steiner και μπορεί να θεωρηθεί πως είναι το μικρότερο wirelength, το ελάχιστο ευθύγραμμο spanning tree (rectilinear minimum spanning tree - RMST) παρέχει μια ικανοποιητική προσέγγιση και εκτίμηση του wirelength. Το RMST μπορεί να θεωρηθεί και ως ένα RSTM χωρίς επιπλέον σημεία Steiner, έτσι ώστε κάθε κόμβος να είναι συνδεδεμένος με το ελάχιστο network το οποίο αποτελείται μόνο από οριζόντια και κάθετα ευθύγραμμα τμήματα. Για τον ίδιο αριθμό κόμβων με συγκεκριμένες συντεταγμένες, ο Hwang απέδειξε ότι :

$$RMSTlength \leq \frac{3}{2} RSMTlength$$

2.4.3 Μοντέλο Κλίκας

Το μοντέλο κλίκας είναι πολύ δημοφιλές σε πολλά προβλήματα VLSI. Το μοντέλο αυτό βασίζεται στην κατάλληλη μετατροπή των nets σε υπογράφους

κλίκας. Το δεδομένο βάρος για κάθε ακμή που κατασκευάζει το μοντέλο κλίκας υπολογίζεται από τον ακόλουθο τύπο :

$$w(i, j) = \frac{2}{n}$$

Όπου i, j οι κόμβοι που είναι συνδεδεμένοι με μία ακμή και n ο συνολικός αριθμός των κόμβων κλίκας. Ο τύπος αυτός βεβαιώνει ότι το αποτέλεσμα του συνολικού βάρους μεγάλων κλικών είναι σχετικά μειωμένο σε σχέση με αυτό των μικρών κλικών οι οποίες χρησιμοποιούνται περισσότερο στην φυσική σχεδίαση. Από την άλλη πλευρά αυτού του τύπου το μοντέλο εξασφαλίζει ότι το αποτέλεσμα του συνολικού βάρους εξισορροπείται με το μέγεθος κάθε κλίκας.

Το μήκος ενός net $n \in N$ είναι η συνολική απόσταση των ζευγαριών των pins στο net :

$$CL_k(n) = \frac{1}{|n| - 1} \sum_{p \in n} \sum_{q \in n} d_k(A(p), A(q))^k$$

Το $netlength$ της κλίκας μπορεί να υπολογιστεί σε χρόνο $O(|n|^2)$. Για μια βελτιωμένη προσέγγιση του γραμμικού μοντέλου net σε σχέση με το τετραγωνικό μοντέλο, ο Gordian-L (Sigl, Doll and Johannes [1991]) χρησιμοποιεί το ακόλουθο επαναληπτικό μοντέλο net : στην επανάληψη όταν $k = 2 \dots$ βελτιστοποιείται:

$$GordianL_x^{(k)}(n) = \frac{1}{|n| - 1} \sum_{p, q \in n} \frac{(x_k(p) - x_k(q))^2}{|x_{k-1}(p) - x_{k-1}(q)|}$$

$$και \quad L^k(n) = GordianL_x^{(k)}(n) + GordianL_y^{(k)}(n)$$

Σημειώνεται ότι η γραμμική κλίκα αποτελούσε το επιλεγόμενο μοντέλο για τις a-priori τοπολογίες, οπότε στο όριο κάποιος μπορεί να καταλήξει στο μέγιστο δυνατό αποτέλεσμα. Τελικά, όμως, δεν μπορεί να επιτευχθεί κάτι καλύτερο από γραμμική σύγκλιση της μεθόδου, οπότε γενικά αυτή η μέθοδος προσέγγισης θα είναι πολύ αργή.

2.4.4 Αστρο

Το μοντέλο είναι παρόμοιο με το μοντέλο κλίκα, όπου οι υπερακμές του γραφήματος μετατρέπονται σε υπογράφους άστρου.

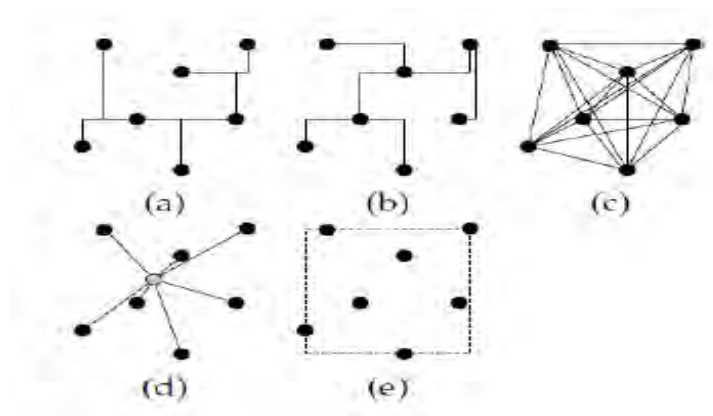
2.4.5 Bounding Box

Η μέθοδος bounding box είναι μακράν η πιο απλή μέθοδος μέτρησης του net, καθώς ορίζει ότι το συνολικό netlength ενός κυκλώματος είναι το άθροισμα της ημι-περιμέτρου του πλαισίου οριοθέτησης που περιβάλλει κάθε net.

$$BB(n) = \max_{p \in n} (A_x(p)) - \min_{p \in n} (A_x(p)) + \max_{p \in n} (A_y(p)) - \min_{p \in n} (A_y(p))$$

Η πολυπλοκότητα του υπολογισμού του συνολικού netlength χρησιμοποιώντας το μετρητή bounding box είναι $O(n)$.

Θεωρώντας και συγκρίνοντας τώρα την πολυπλοκότητα του μετρητή bounding box με αυτήν του μετρητή κλίκα, είναι προφανές ότι οι αλγόριθμοι που χρησιμοποιούν την πρώτη μέθοδο αποκαλούνται αλγόριθμοι γραμμικής ελαχιστοποίησης, ενώ οι αλγόριθμοι που χρησιμοποιούν τη δεύτερη μέθοδο αποκαλούνται αλγόριθμοι τετραγωνικής ελαχιστοποίησης. Τα σχήματα που παρουσιάζονται παρακάτω μας δείχνουν το είδος του γραφήματος κάθε μετρητή και τη διαφορά μεταξύ του γραμμικού και του τετραγωνικού υπολογισμού.



Εικόνα 10: Steiner Tree and Ελάχιστο επικαλύπτον δέντρο(a,b), κλίκα(c), άστρο(d), bounding box(e)

3. Μέθοδος Placement και αλγόριθμοι

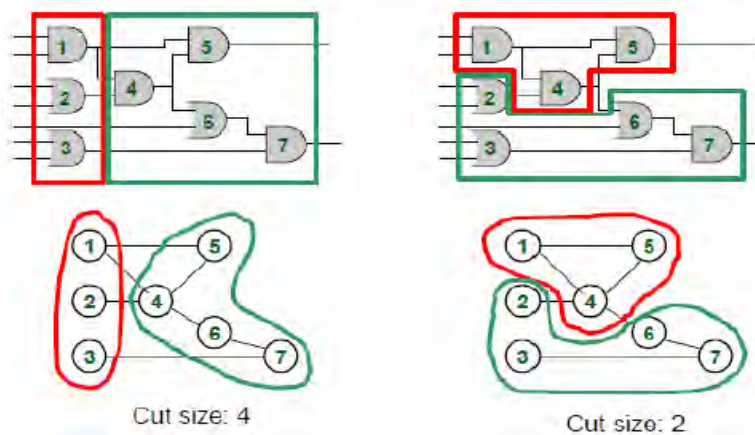
Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, το placement χωρίζεται σε δύο στάδια, το global placement και final placement. Σε αυτό το κεφάλαιο παρουσιάζονται, οι μέθοδοι των δύο φάσεων και οι πιο γνωστοί αλγόριθμοι.

3.1 Μέθοδος Global Placement

Ο κύριος σκοπός της μεθόδου του global placement είναι να διανείμει τα cells ομοιόμορφα στην δεδομένη περιοχή και να βελτιστοποιήσει την τοποθέτησή τους σύμφωνα με ορισμένους στόχους που έχουν τεθεί όπως το wirelength, ο χρονισμός, η ισχύς κτλ.

3.1.1 Διαχωρισμός Γράφων (Graph partitioning)

Η πλέον χρησιμοποιούμενη τεχνική για global placement βασίζεται στο διαχωρισμό γράφων. Δεδομένου ενός κυκλώματος σε επίπεδο πυλών, ο στόχος του προβλήματος διαχωρισμού του κυκλώματος είναι να διαιρέσουμε το κύκλωμα σε K περίπου ίσου μεγέθους τμήματα. Ο κύριος στόχος της μεθόδου αυτής είναι να ελαχιστοποιηθεί ο αριθμός των υπερακμών που συνδέουν τους κόμβους από το ένα τμήμα στο άλλο, το οποίο τυπικά ονομάζεται "cutsizes". Άλλοι στόχοι περιλαμβάνουν κρίσιμα μονοπάτια καθυστέρησης, άλλοι την συνολική κατανάλωση ενέργειας, κ.λπ. Λόγω αυτών των στόχων, οι partitioning αλγόριθμοι μπορεί να χωριστούν σε γραμμικούς και αναλυτικούς, το οποίο σημαίνει πως οι αλγόριθμοι προσεγγίζουν γραμμικά το "cutsizes" και τετραγωνικά υπολογίζουν τους άλλους στόχους.



Εικόνα 11: Παράδειγμα Cut-size

3.1.2 Analytic and Relaxation Based Placement

Οι φόρμουλες του τετραγωνικού netlength μπορούν να επιλυθούν γρήγορα με τη χρήση αριθμητικών γραμμικών λύσεων των εξισώσεων, με την πιο δημοφιλή μέθοδο να είναι η μέθοδος Conjugate Gradient, ενώ το μοντέλο bounding-box απαιτεί τετραγωνικό χρόνο λειτουργίας των μεθόδων δικτύου-ροής.

3.1.3 Μέθοδοι Force-Based

Οι μέθοδοι force-based μπορούν να ορισθούν ως μέθοδοι που χειρίζονται τα modules ως αντικείμενα και τα nets ως ελατήρια που συνδέουν τα αντικείμενα, και προσπαθούν να ελαχιστοποιήσουν το netlength διασύνδεσης θέτοντας “spring system” στο ισοζύγιο, σύμφωνα με το νόμο του Hooke. Αυτό το είδος της ελαχιστοποίησης του netlength εισήχθη από τους Eisenmann και Johannes το 1998. Η μέθοδος force-based χρησιμοποιεί τεχνικές αναλυτικού placement, με στόχο να επιτύχει placement στο οποίο τα κελιά επικαλύπτονται (overlapping placement), το οποίο ελαχιστοποιεί το τετραγωνικό netlength. Η ευριστική αυτή μέθοδος προχωράει επαναληπτικά εισάγοντας “repelling forces” μεταξύ των modules και των bins τα οποία βρίσκονται στην περιοχή του placement και παρουσιάζουν επικάλυψη. Σε κάθε επανάληψη οι τρέχουσες επικαλυπτόμενες δυνάμεις προστίθενται στο τετραγωνικό netlength και η συνδυασμένη τετραγωνική συνάρτηση ελαχιστοποιείται.

Η δύναμη σε μια περιοχή $(x,y)^t$ βρίσκεται ως εξής:

$$f(x, y) = \frac{k}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} D(x', y') \frac{(r - r')}{|r - r'|^2} dx' dy'$$

Για να βελτιωθεί το netlength, οι Eisenmann και Johannes χρησιμοποίησαν επίσης μια παραλλαγή της μεθόδου γραμμικοποίησης. Τέλος, η τοπική μέθοδος αναζήτησης Domino η οποία θα συζητηθεί αργότερα, χρησιμοποιούνταν για τελικό placement. Οι Eisenmann και Johannes χρησιμοποιούσαν την ευριστική μέθοδο μόνο στα standard-cell κυκλώματα, αλλά ο Mo επέκτεινε την ευρεστική force-based μέθοδο στα macro-cells χωρίς κάποιο σχέδιο γραμμικοποίησης. Επιπλέον, αντί για την εισαγωγή των δυνάμεων απώθησης (repulsive forces), δημιούργησε μια δύναμη πλήρωσης ανάμεσα στα modules και τις κενές περιοχές του placement. Ο Mo επίσης εισάγει και προσανατολισμό στα macro-cells. Για κάθε κελί ο καινούριος

προσανατολισμός καθορίζεται θεωρώντας και τους οχτώ πιθανούς προσανατολισμούς και διαλέγοντας αυτόν ο οποίος ελαχιστοποιεί την εξωτερική δύναμη. Δεδομένου ότι η αλλαγή προσανατολισμού σε ένα κελί επηρεάζει αυτόματα και τη δύναμη των άλλων κελιών, δεν θα πρέπει να γίνεται ταυτόχρονα αλλαγή προσανατολισμού σε πολλά κελιά.

Στην συνέχεια οι Hu και Marek-Sadowska άλλαξαν λίγο την μέθοδο των Eisenmann και Johannes και εισήγαγαν το Fixed-point Addition and Relaxation (FAR). Σε αυτή τη μέθοδο δημιουργούν ένα ψευτο-στοιχείο για κάθε στοιχείο και μια ψευτοσύνδεση μεταξύ του στοιχείου και του ψευτο-στοιχείου. Παρατήρησαν ότι τα συγκεκριμένα σημεία είναι πιο εύκολα διαχειρίσιμα από τις αποθητικές δυνάμεις, καθώς επίσης και το αναλυτικό placement μπορεί να περιέχεται σε συγκεκριμένη περιοχή, σε σύγκριση με τις συνεχείς δυνάμεις.

Μια άλλη μέθοδος force-based είναι και η Attractor Repellor Approach (ARP), η οποία προτάθηκε από τον Etawil το 1999. Αποτελεί μια εντελώς διαφορετική μέθοδο σε σχέση με την προηγούμενη καθώς αντί να παρακολουθεί τις περιοχές όπου υπάρχει επικάλυψη, εισάγει μια δύναμη απώθησης η οποία υπάρχει ανάμεσα στα άμεσα συνδεδεμένα modules. Αυτή η εναλλακτική ιδέα έχει αρκετά ελαττώματα, όπως για παράδειγμα ότι υπάρχει έντονη επικάλυψη με αποτέλεσμα να αυξάνεται το συνολικό wirelength στο στάδιο legalization. Για το λόγο αυτό, ο Etawil εισάγει δυνάμεις έλξης σε περιοχές χαμηλής πυκνότητας.

3.1.4 Simulated Annealing για Global Placement

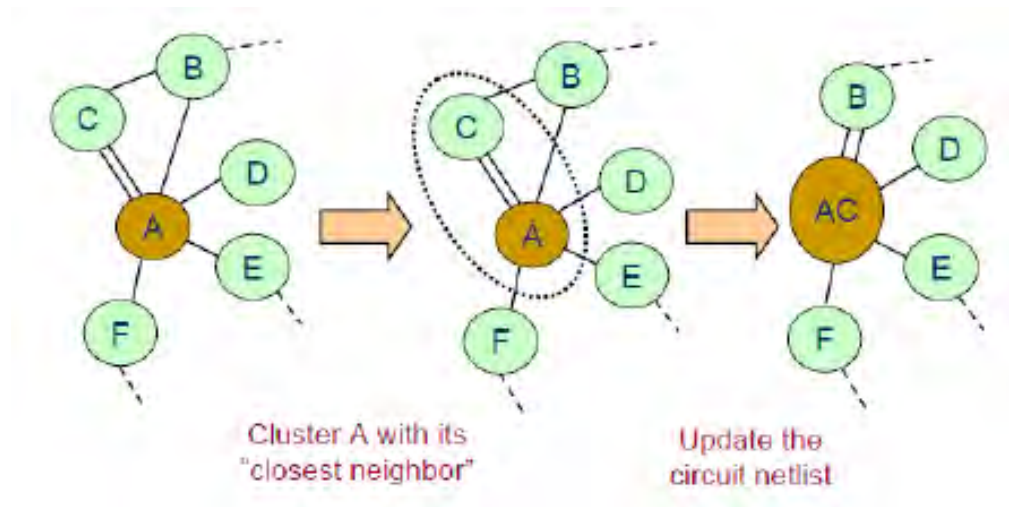
Η μέθοδος simulated annealing (SA) είναι μια γενικευμένη πιθανολογική μέθοδος η οποία ασχολείται με το πρόβλημα του global optimization. Αρχικά προτάθηκε από τους Kirkpatrick, Gelett και Vecchi (1983) και Cerny (1985). Το αποτέλεσμα της είναι μια καλή προσέγγιση του ολικού βέλτιστου μιας συνάρτησης με συγκεκριμένο στόχο σε ένα μεγάλο, διακριτό χώρο αναζήτησης με πολλά ελάχιστα. Σε πολλές περιπτώσεις η μέθοδος αυτή μπορεί να είναι περισσότερο αποδοτική από κάθε άλλη ευρεστική μέθοδο.

Εξαιτίας της απλότητάς της, έχει γίνει η πιο διάσημη μετα-ευρεστική μέθοδος η οποία χρησιμοποιείται σε πολλές εφαρμογές, περιλαμβάνοντας και τη φυσική σχεδίαση όταν όλες οι άλλες μέθοδοι αποτυγχάνουν. Ωστόσο, η εφαρμογή της Simulated Annealing στο πρόβλημα του placement έχει αποδείξει ότι είναι αρκετά χρονοβόρα όταν χρησιμοποιείται για κυκλώματα μεγάλης κλίμακας. Παρόλα αυτά, χρησιμοποιείται από τον Dragon placer, για το καθολικό placement καθώς και για το τελικό placement, ακόμα και όταν χρησιμοποιείται σε μικρά κομμάτια (partitions) του global placement.

3.1.5 Ομαδοποίηση (Clustering)

Μερικοί αλγόριθμοι τοποθέτησης συνδυάζουν τα modules σε ομάδες. Η ομαδοποίηση (clustering) είναι μια ισχυρή μέθοδος η οποία ασχολείται με τα υπο-κυκλώματα, και μπορεί να μειώσει τον χρόνο λειτουργίας (running time) εφόσον υπάρχουν λιγότερα modules. Επιπλέον, η ομαδοποίηση μπορεί να μετακινήσει ένα αλγόριθμο απλής τοποθέτησης εκτός ενός τοπικού ελαχίστου καθώς χρησιμοποιεί πιο πολύπλοκη κίνηση των ατομικών κελιών. Οι μέθοδοι ομαδοποίησης έχουν χρησιμοποιηθεί μαζί με αλγορίθμους διαμέρισης γράφων με στόχο να βελτιωθεί ο χρόνος της διαμέρισης. Ωστόσο, οι ομάδες (clusters) χρησιμοποιούνται και σε λεπτομερείς τοποθετήσεις (detailed placement). Δεδομένου ενός netlist, ο στόχος του clustering στο κύκλωμα είναι η ομαδοποίηση των πυλών και των διασυνδέσεών τους σε clusters, και η εύρεση αυτού του δικτύου των clusters. Έτσι, μετά την ομαδοποίηση η ομάδα (cluster) γίνεται ένα μέρος του γράφου, οδηγώντας σε μειωμένο αριθμό κόμβων και διασυνδέσεων. Ακόμα, η ομαδοποίηση χρησιμοποιείται πριν τους αλγορίθμους τοποθέτησης με στόχο να μειώσει την πολυπλοκότητα του χρόνου λειτουργίας (run time) κάθε ευρεστικής μεθόδου που υλοποιείται μετά από αυτήν. Ο στόχος των συναρτήσεων ομαδοποίησης που χρησιμοποιούνται είναι να ελαχιστοποιήσουν τις συνδέσεις μεταξύ των clusters, μεγιστοποιώντας τις συνδέσεις μέσα στα clusters, καθώς και να μεγιστοποιήσουν τον αριθμό των inter-cluster συνδέσεων σε οποιοδήποτε μονοπάτι. Τυπικοί περιορισμοί αυτής της μεθόδου είναι το μέγιστο μέγεθος ενός cluster και ο μέγιστος αριθμός των εξωτερικών συνδέσεων ενός cluster. Ο αριθμός των clusters που πρόκειται να

ληφθεί δεν είναι συγκεκριμένος, και η περιοχή ισορροπίας ανάμεσα στα clusters συνήθως δεν είναι απαραίτητη.



Εικόνα 12: Παράδειγμα Clustering

Υπάρχουν δύο τρόποι θεώρησης του προβλήματος ομαδοποίησης στο physical placement, με στόχο να οργανωθούν καλύτερα τα clusters για το συγκεκριμένο πρόβλημα:

- Η παραγωγή των clusters ακολουθώντας αυστηρά την απεικόνιση του hypergraph του κυκλώματος, σαν ένα γενικό πρόβλημα χωρίς να ασχολούμαστε με το placement.
- Η παραγωγή των clusters ανάλογα με ευρεστικές τοποθέτησης και η μετατροπή βάρους (weights) ανάλογα με τον στόχο της συνάρτησης.

Οι πιο διαδεδομένοι αλγόριθμοι που χρησιμοποιούν τη μέθοδο ομαδοποίησης βάσει των ευρεστικών τοποθέτησης είναι οι εξής:

- ❖ Αλγόριθμος Rajaraman και Wong (1995),
- ❖ Αλγόριθμος FlowMap (Cong and Ding, 1992),
- ❖ Αλγόριθμος Multi-level Coarsening (Karypis et al., 1997)

Οι πρώτοι δύο είναι αλγόριθμοι timing driven, όπου η μεγαλύτερη καθυστέρηση ενός μονοπατιού μέσα σε ένα cluster ελαχιστοποιείται υπό την επίβλεψη ενός συγκεκριμένου μοντέλου καθυστέρησης. Ο τελευταίος αλγόριθμος ελαχιστοποιεί τον αριθμό των συνδέσεων μεταξύ των clusters.

3.2 Μέθοδοι Final placement

Ο σκοπός του υπο-προβλήματος του final placement είναι η περαιτέρω βελτίωση της ποιότητας της λύσης του placement. Είναι πιο περιορισμένο από το global placement καθώς βελτιστοποιεί τους στόχους και παράγει ένα επιτρεπτό (legal) placement. Χρησιμοποιεί συνήθως περισσότερο ακριβή μοντέλα wirelength όπως η ημιπερίμετρος. Η δουλειά και η έρευνα που έχει γίνει πάνω στο final placement είναι σημαντικά λιγότερη συγκριτικά με αυτή του global placement. Στις παραγράφους που ακολουθούν θα παρουσιαστούν οι πιο σημαντικές μέθοδοι του final placement.

Η ομαδοποίηση της μεθόδου του final placement είναι δύσκολη διότι ενδέχεται να χρησιμοποιεί μια σειρά διαφορετικών αλγορίθμων μέσα στην ίδια μέθοδο, και ένα ποσοστό του legalization μπορεί να συνυπάρχει στη διαδικασία του global placement. Τείνουν να υπάρχουν πέντε κύριες κατηγορίες τεχνικών που χρησιμοποιούνται πιο συχνά, ή συνδυασμοί αυτών.

- Η πρώτη κατηγορία είναι οι μέθοδοι network flow. Σε αυτές τις μεθόδους το πρόβλημα μεταφοράς επιλύεται χρησιμοποιώντας την ελάχιστη ροή κόστους ή τον αλγόριθμο με το συντομότερο μονοπάτι. Η περιοχή του placement χωρίζεται σε bins, και οι τιμές της πυκνότητας υπολογίζονται για κάθε bin βάσει των κελιών που υπάρχουν στην περιοχή του bin. Το κόστος της μετακίνησης ενός κελιού σε ένα άλλο bin υπολογίζεται σύμφωνα με το στόχο της συνάρτησης. Γενικά, τα κελιά μετακινούνται από το πυκνότερο bin στα λιγότερο πυκνά, ακολουθώντας διαδρομές που έχουν ως αποτέλεσμα την μικρότερη υποβάθμιση της ποιότητας του placement.
- Η δεύτερη κατηγορία είναι οι μέθοδοι min-cut, οι οποίες διαχωρίζουν το global placement σε σύνολα ίσου αριθμού μονοπατιών και χρησιμοποιούν τεχνικές partitioning για την ελαχιστοποίηση των διασυνδέσεων.
- Η τρίτη κατηγορία, οι γραμμικές μέθοδοι τοποθέτησης, βελτιστοποιεί τη γραμμική διευθέτηση των κελιών εντός μιας σειράς.

- Μια άλλη βασική κατηγορία, οι τυχαίες μέθοδοι όπως αυτή που βασίζεται στο simulated annealing, κινείται τυχαία τοπικά ώστε να επιτύχει τη βελτιστοποίηση του στόχου της συνάρτησης.
- Τέλος, κάποιες άπληστες προσεγγίσεις χρησιμοποιούνται για τη βελτιστοποίηση του wirelength κατά τη διαδικασία του final placement.

Πιο συγκεκριμένα, το final placement μπορεί να χωριστεί σε δύο βήματα, το legalization και το detailed placement wirelength optimization.

3.2.1 Legalization

Δεδομένου ενός illegal placement, το legalization είναι η διαδικασία η οποία εξουδετερώνει όλες τις επικαλύψεις διαταράσσοντας τα modules όσο λιγότερο γίνεται.

Η αφαίρεση των επικαλύψεων αποτελεί την πρώτη δοκιμασία για έναν detailed αλγόριθμο. Η κοινή τεχνική για την αφαίρεση των επικαλύψεων είναι στοιβάζοντας τα κελιά σε σειρές ή απλώνοντάς τα στο εσωτερικό όλων των placement bins. Μια άλλη προσέγγιση είναι στοιβάζοντας τα κελιά σειρά - σειρά χωρίς την εισαγωγή λευκού κενού (white space) και χωρίς να λαμβάνεται υπόψιν η ανισορροπία. Την προσέγγιση αυτή συναντάμε για πρώτη φορά στον αλγόριθμο Feng Shui.

Σε μια διαφορετική μέθοδο την οποία συναντάμε στον αλγόριθμο Domino, ένα συμπαγές placement χωρίς επικαλύψεις, κατασκευάζεται αναπτύσσοντας το placement σαν κρύσταλλο ανά περιοχή από τα αριστερά προς τα δεξιά.

Μια άλλη προσέγγιση διαπερνά το placement ανά κελί από αριστερά προς τα δεξιά και πραγματοποιεί μια λιγότερο απαιτητική έκδοση των packing cells στα αριστερά, η οποία έχει εμφανιστεί στον αλγόριθμο του Tetris. Μερικές φορές το detailed placement χωρίζεται σε δύο στάδια με ένα coarse legalizer σαν το Mongrel, προβαίνοντας σε σημαντικές βελτιώσεις στην αφαίρεση των επικαλύψεων πριν χρησιμοποιηθεί ένας fine-grain legalizer.

3.2.2 Τοπική βελτίωση

Η ελαχιστοποίηση του wirelength ή άλλοι στόχοι που έχουν τεθεί κατά τη διάρκεια της διαδικασίας του detailed placement συχνά πραγματοποιείται ανταλλάσσοντας κελιά στόχους μεταξύ διαφορετικών σειρών. Άλλα κελιά στις γραμμές αυτές θα πρέπει ενδεχομένως να μετατοπιστούν έτσι ώστε να αφαιρεθεί η ενδεχόμενη επικάλυψη ή το λευκό κενό (white space).

Μερικά εργαλεία τοποθέτησης δεν εξετάζουν μόνο οποιονδήποτε συνδυασμό της σειράς των κελιών εντός μιας σειράς. Η μέθοδος Domino βελτιώνει την τρέχουσα λύση του placement επαναληπτικά με βάση το wirelength. Επίσης, μειώνει τις επικαλύψεις διαχωρίζοντας την περιοχή του πυρήνα σε bins και επιλύοντας το πρόβλημα τοπικά, το καθένα από τα οποία διαμορφώνεται ως πρόβλημα μεταφοράς και επιλύεται με μεθόδους network flow. Έτσι τα swaps μεταξύ των κελιών επιτρέπονται μόνο μέσα σε μία συγκεκριμένη σειρά, αποφεύγοντας την εκτεταμένη μετατόπιση των κελιών. Μια άλλη τεχνική placement προτείνει τη χρήση προσεγγίσεων δυναμικού προγραμματισμού για το διαχωρισμό των κελιών εντός μιας σειράς σε δύο ομάδες οι οποίες αλλάζουν δυναμικά, και τέλος τις αναδιατάσσει με τον κοντινότερο βέλτιστο τρόπο στη σειρά. Παρόλα αυτά, η τεχνική αυτή έχει επιπλέον πολυπλοκότητα για πολλαπλούς αριθμούς σειρών. Ο Capo χρησιμοποιεί ένα branch-and-bound end placer ώστε να βρει την καλύτερη μετάθεση των κελιών σε μια μικρή περιοχή. Διαχωρίζει και «ράβει» το white space με τα κελιά, εάν υπάρχει white space στην περιοχή, ώστε να γίνει ομοιόμορφη διανομή των κελιών.

Μερικά εργαλεία Placement χρησιμοποιούν την ανταλλαγή των κελιών σε διαφορετικές σειρές. Για παράδειγμα κατά τη διάρκεια του detailed placement ο Dragon2000 χρησιμοποιεί την ανταλλαγή των cells σε διαφορετική σειρά χωρίς να υπολογίζει την αλλαγή του συνάρτησης κόστους και σε αρκετές περιπτώσεις αυτό οδηγεί στη μετατόπιση των διπλών κελιών με αποτέλεσμα χειρότερες λύσεις placement. Μία άλλη προσέγγιση στον FengShui είναι η ανταλλαγή των cells σε διαφορετική σειρά μόνο όταν δεν δημιουργείται overlap με άλλα cells. Από την άλλη, αυτές οι κινήσεις περιορίζουν την αποτελεσματικότητα των ανταλλαγών στην ίδια σειρά. Ένα άλλο σχετικό θέμα

είναι ο ακριβής υπολογισμός της αλλαγής του wirelength για τα μετατοπισμένα κελιά. Ο υπολογισμός αυτός μπορεί να είναι υπολογιστικά ακριβός και για αυτό ο Timberwolf, για παράδειγμα, χρησιμοποιεί στατιστικό υπολογισμό του wirelength ώστε να μειώσει τον υπολογιστικό χρόνο.

3.2.3 *Simulated annealing*

Όπως είδαμε και σε προηγούμενη παράγραφο, η μέθοδος Simulated annealing (SA) αποτελεί μια γενικευμένη πιθανολογική μέθοδο για το πρόβλημα του global optimization. Το αποτέλεσμα της είναι μια καλή προσέγγιση του ολικού βέλτιστου μιας συνάρτησης με συγκεκριμένο στόχο σε ένα μεγάλο, διακριτό χώρο αναζήτησης με πολλά ελάχιστα. Σε πολλές περιπτώσεις η μέθοδος αυτή μπορεί να είναι περισσότερο αποδοτική από κάθε άλλη ευρεστική μέθοδο.

Η κεντρική ιδέα αυτού του αλγορίθμου προέρχεται από την τεχνική annealing που υπάρχει στον τομέα της μεταλλουργίας. Η τεχνική αυτή ανεβάζει τη θερμοκρασία ενός υλικού μέχρι μια κρίσιμη τιμή και στη συνέχεια το ψύχει ελεγχόμενα με σκοπό να αυξήσει το μέγεθος των κρυστάλλων του και να αλλάξει τις ιδιότητές του όπως τη σκληρότητα και την ολκιμότητά του. Η αύξηση της θερμοκρασίας κάνει τα άτομα να ξεκολλήσουν από το νέφος ηλεκτρονίων με την ελάχιστη εσωτερική ενέργεια και να περιπλανιώνται τυχαία μεταξύ καταστάσεων υψηλότερης ενέργειας. Ύστερα, με την ελεγχόμενη ψύξη τα άτομα προσπαθούν να επιστρέψουν σε χαμηλές εσωτερικές καταστάσεις ενέργειας, ίσως και σε κατάσταση με χαμηλότερη ενέργεια και από την αρχική κατάσταση στην οποία βρισκόταν πριν της διαδικασίας αυτής.

Η μέθοδος Simulated annealing αποτελεί μια από τις πιο αναπτυγμένες μεθόδους placement από αυτές που είναι διαθέσιμες στην αγορά. Χρησιμοποιείται στο placement ως ένας αλγόριθμος επαναληπτικής βελτίωσης. Δοσμένου ενός αρχικού placement, η αλλαγή γίνεται μετακινώντας ένα στοιχείο ή με την εναλλαγή των θέσεων δύο στοιχείων.

```

Algorithm Simulated-Annealing
Begin
  temp=Init-Temp;
  place=Init-Placement;
  while (temp>Final-Temp) do
    while (inner_loop_criterion = False)
      new_place=Pertrub(place);
      ΔC=Cost(new_place)-Cost(place);
      if (ΔC<0) then
        place=new_place;
      else if (Random(0,1)>eΔC/temp) then
        place=new_place;
      temp=Schedule(temp);
  End

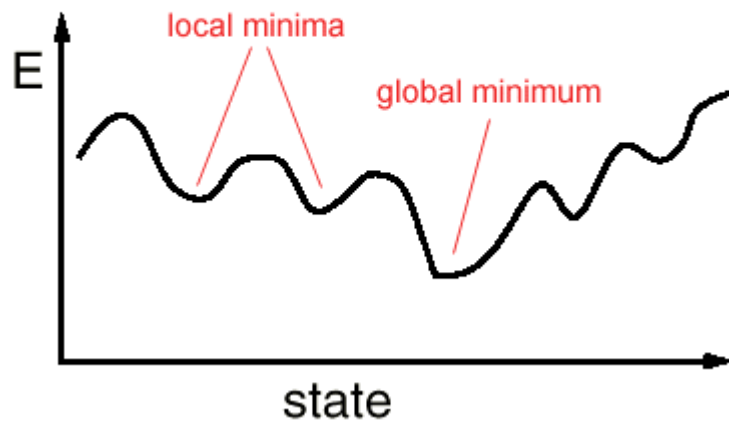
```

Εικόνα 13: Αλγόριθμος Simulated annealing

Όλες οι κινήσεις οι οποίες οδηγούν σε μείωση του κόστους είναι προφανώς αποδεκτές. Οι κινήσεις οι οποίες οδηγούν σε αύξηση του κόστους είναι αποδεκτές μόνο όταν υπάρχει το ενδεχόμενο να μειωθεί το κόστος μετά από κάποιες επαναλήψεις. Η αναλογία με την πραγματική διαδικασία annealing ενισχύεται με τη χρήση μιας παραμέτρου που ονομάζεται θερμοκρασία T , η οποία ελέγχει την πιθανότητα αποδοχής κινήσεων με αυξημένο κόστος. Η αρχική τιμή της θερμοκρασίας στον αλγόριθμο είναι πολύ υψηλή και σταδιακά μειώνεται κατά τη διάρκεια της διαδικασίας έτσι ώστε οι κινήσεις που αυξάνουν το κόστος να έχουν μικρότερη πιθανότητα να γίνουν αποδεκτές. Τέλος, η θερμοκρασία μειώνεται σε μια πολύ χαμηλή τιμή και έτσι οι μόνες κινήσεις που γίνονται αποδεκτές είναι αυτές που οδηγούν σε μειωμένο κόστος.

Χρησιμοποιώντας αυτή τη διαδικασία, η Simulated annealing μπορεί να υπερβεί τα τοπικά ελάχιστα και να βρει ένα ολικό ελάχιστο ή λύσεις κοντά σε αυτό. Οι παράμετροι και οι συναρτήσεις που χρησιμοποιούνται σε έναν αλγόριθμο simulated annealing καθορίζουν την ποιότητα του placement. Αυτές περιλαμβάνουν το χρονοδιάγραμμα ψύξης το οποίο αποτελείται από την αρχική θερμοκρασία, την τελική θερμοκρασία και τη συνάρτηση για την αλλαγή της θερμοκρασίας. Μια καλή επιλογή παραμέτρων και συναρτήσεων μπορεί να οδηγήσει σε ένα καλό placement και σχετικά μικρό χρονικό διάστημα.

Simulated Annealing



Εικόνα 14: Graph of simulated annealing

Η μέθοδος αυτή είχε περιγραφεί ανεξάρτητα από τους Scott Kirkpatrick, Daniel Gelatt και Mario P. Vecchi το 1983, και από τον Vlado Cerny το 1985. Η μέθοδος αυτή είναι μια αναπροσαρμογή του αλγορίθμου Metropolis-Hastings, μιας μεθόδου Monte Carlo η οποία παράγει sample states ενός θερμοδυναμικού συστήματος, η οποία εφευρέθηκε από τον M. N. Rosenbluth το 1953.

Η simulated annealing χρησιμοποιείται κυρίως στο detailed placement. Παρόλα αυτά, οι Wang και Sarrafzadeh χρησιμοποιούν αυτή τη μέθοδο για το global placement αφού η περιοχή του placement έχει χωριστεί πρώτα σε bins. Κάθε κίνηση στην εφαρμογή του simulated annealing σε αυτά τα bins είναι μια εναλλαγή κελιών μεταξύ των bins. Το μέγεθος του δικτύου καθορίζεται από την ανάλυση ενός κυκλώματος.

Η μέθοδος χρησιμοποιούταν, επίσης, στον αλγόριθμο Dragon από τους Wang και Sarrafzadeh στο στάδιο του detailed placement. Το πρώτο στάδιο του placer αποτελείται από μια ιεραρχική προσέγγιση. Η περιοχή του placement διαιρείται αναδρομικά σε δομές bin 2x2. Αυτή η αναδρομή ολοκληρώνεται όταν κάθε bin αποτελείται από μόνο εφτά στοιχεία. Σε κάθε επίπεδο του ιεραρχικού αλγορίθμου, ένας αλγόριθμος εναλλαγής των bins επιδιώκει να αλλάξει τα περιεχόμενα μεταξύ γειτονικών bins με στόχο να βελτιώσει το placement.

3.2.4 Greedy Approaches

Οι greedy approaches συνήθως ελέγχουν όλες τις πιθανές εναλλαγές ενός κελιού ή μιας ομάδας κελιών, έτσι ώστε να βρουν την καλύτερη τοποθέτηση για αυτά δεδομένου ενός set τοπικών περιορισμών.

Το detailed placement αναφέρεται επίσης και ως τελικό ή απόλυτο γεωμετρικό placement. Το ποσοστό του detailed placement εξαρτάται από τον τύπο του αλγορίθμου global placement που χρησιμοποιείται. Σε αντίθεση με το στάδιο του global placement, το detailed placement δεν προσελκύει μεγάλη προσοχή εξαιτίας της σχετικά μικρής πολυπλοκότητας του χρόνου λειτουργίας (run-time). Παρόλα αυτά, ένα κακό detailed placement μπορεί να αποτύχει να διατηρήσει ή να βελτιώσει την ποιότητα ενός καλού global placement.

4. Τεχνικές Legalization

Το legalization αποτελεί ένα βασικό βήμα του placement για μια επιτυχημένη και πλήρως αναλυτική τοποθέτηση. Σε ένα legal placement, όλα τα κελιά πρέπει να είναι ευθυγραμμισμένα στα όρια της σειράς και να μην υπάρχει καμία επικάλυψη μεταξύ τους.

Η αναφορά μας ξεκινάει με μια μέθοδο flow-based, η οποία προέρχεται από τις μεθόδους space management. Εάν ένα placement έχει εξαπλωθεί αποτελεσματικά, έτσι ώστε να μην υπάρχει περιοχή η οποία να απαιτεί περισσότερο από τον διαθέσιμο χώρο, τότε η διαδικασία του legalization είναι ασήμαντη.

Η μέθοδος Tetris είναι εξαιρετικά απλή και εύκολη στην εφαρμογή, θα πρέπει να παρέχει καλή διαίσθηση για το πρόβλημα. Ενώ η μέθοδος λειτουργεί καλά σε κάποια προβλήματα, μπορεί να αποτύχει δραματικά σε κάποια άλλα: το οποίο μας δίνει να καταλάβουμε το γιατί οι πιο ισχυρές μέθοδοι όπως αυτές που βασίζονται στην ελάχιστη ροή κόστους είναι αναγκαίες.

4.1 Flow και Diffusion-based legalization

Χρησιμοποιώντας είτε το μοντέλο ελάχιστης ροής κόστους είτε της διάχυσης (diffusion), ένα placement το οποίο έχει μικρές τοπικές περιοχές επικάλυψης μπορεί να επιτευχθεί. Για σχεδιάσεις οι οποίες περιλαμβάνουν standard cells (ή στις οποίες τα macroblocks είναι σταθερά), το legalization μπορεί να επιτευχθεί εύκολα.

Με τη δημιουργία ζωνών, bins, ή περιοχών που έχουν ύψος ένα μόνο κελί και στη συνέχεια διασφαλίζοντας ότι κάθε ζώνη βρίσκεται εντός του περιορισμού πυκνότητας, ένα legal placement μπορεί να βρεθεί τοποθετώντας κελιά σε κάθε ζώνη. Μια σωστή λύση εγγυημένα υπάρχει: η διανομή των κελιών εντός μιας ζώνης μπορεί να βελτιώσει περαιτέρω τα αποτελέσματα.

Οι μέθοδοι diffusion-based και flow, εκτός από μέθοδοι διαχείρισης του χώρου, μπορεί να θεωρηθούν και ως τεχνικές legalization. Μια καλή διαχείριση χώρου καθιστά το legalization ασήμαντο.

4.2 Tetris-Based Legalization

Αντί να διανέμεις τα στοιχεία του κυκλώματος με μεθόδους ροής, το στοίβαγμά τους είναι μια δυνατή εναλλακτική προσέγγιση. Η μέθοδος του Tetris legalization από τον Hill είναι εξαιρετικά απλή, και εύκολη στην εφαρμογή. Πρώτα αναφέρουμε την προσέγγιση στο πλαίσιο της σχεδίασης των standard cells, και στη συνέχεια δείχνουμε πώς επεκτείνεται με σκοπό να χειριστεί ένα συνδυασμό από standard cells και macroblocks. Για κάθε κελί c_i , έχουμε μια επιθυμητή θέση (x_i, y_i) : τα κελιά πρέπει να είναι τοποθετημένα σε τυπικές γραμμές κελιών $R = \{r_1, r_2, \dots, r_k\}$, και θεωρούμε ότι η αριστερότερη διαθέσιμη θέση της κάθε γραμμής είναι γνωστή.

Η μέθοδος legalization πρώτα ταξινομεί τα κελιά C ανάλογα με την θέση x , και μετά εισάγει αυτά στην αριστερή πλευρά της γραμμής με greedy τρόπο, τέτοιο ώστε η μετατόπιση του κάθε κελιού να είναι η ελάχιστη.

```

Algorithm The Tetris legalizer by Hill
{C}= All cells to be legalized;
Sort the cells in C by their X-coordinates to get  $L_s$ ;
 $l_j$  = left-most position of each row  $r_j$ ;
for  $i = 1$  to the number of cells do
     $best = \text{limsup}$ ;
    for  $j = 1$  to the number of rows do
         $cost = \text{displacement of moving cell } i \text{ in } L_s \text{ to } l_j$ ;
        if  $cost \leq best$  then
             $best = cost$ ;
             $best\_row = j$ ;
        end if
    end for
    Move cell  $i$  in  $L_s$  to the row  $best\_row$ ;
     $l_{best\_row} = l_{best\_row} + width_i$ ;
end for

```

Εικόνα 15: Αλγόριθμος Tetris

Ο ψευδοκώδικας για την προσέγγιση αυτή παρουσιάζεται στον παραπάνω αλγόριθμο. Αν περιστρέψουμε την περιοχή του placement αριστερόστροφα τότε η διαδικασία του legalization μπορεί να μοιάζει πολύ με ένα παιχνίδι Tetris, από το οποίο πήρε το όνομά του.

Ο κώδικας του παραδείγματός μας μπορεί να τρέξει γρηγορότερα εξετάζοντας μόνο τις σειρές που βρίσκονται κοντά στην επιθυμητή θέση του κελιού. Στην πράξη, ο χρόνος εκτέλεσης του αλγορίθμου είναι γραμμικός σε σχέση με τον αριθμό των κελιών στα οποία πρόκειται να επέλθει legalization. Το παράδειγμά μας, επίσης, τοποθετεί τα κελιά στα αριστερά, αλλά αυτό δεν είναι απαραίτητο. Εάν ο χώρος έχει δεσμευτεί για δρομολόγηση, ίσως είναι προτιμότερο να γίνει legalization σε μια θέση που δεν βρίσκεται προς τα αριστερά.

Άλλες προφανείς παραλλαγές είναι το legalization μπορεί να πραγματοποιείται τόσο από την αριστερή όσο και από τη δεξιά πλευρά, και η ταξινόμηση των κελιών να γίνεται με το αριστερότερο όριο τους, αντί για το κέντρο τους. Η μέθοδος μπορεί ακόμα να προσαρμοστεί σε σχέδια τα οποία περιέχουν macroblocks, προσθέτοντας απλά έναν έλεγχο για την κάθε γραμμή όπου υπάρχει ένα macroblock, για να βρεθεί η αριστερότερη θέση που δεν οδηγεί σε επικάλυψη.

Αν και η μέθοδος Tetris δουλεύει καλά στην πράξη για τα global placements, τα οποία διαθέτουν λογικά στοιχεία τα οποία κατανέμονται ομοιόμορφα, ακόμα και μικρές περιοχές με επικάλυψη μπορούν να προκαλέσουν σημαντικά προβλήματα. Όταν ένα block πρέπει να μετακινηθεί κατά τη διαδικασία του legalization, ίσως προκαλέσει αλυσιδωτές αντιδράσεις με αποτέλεσμα να μετακινηθούν και άλλα blocks. Σε μερικές περιπτώσεις, τα wirelengths μπορούν να αυξηθούν πάνω από 30 τοις εκατό. Η μέθοδος είναι γρήγορη και αποτελεσματική για εύκολα προβλήματα legalization αλλά έχει χαμηλή επίδοση στις πιο δύσκολες περιπτώσεις.

4.3 Single-Row Dynamic Programming-Based Legalization

Σε αντίθεση με την απλή μέθοδο Tetris η οποία είναι μέθοδος που λειτουργεί με βάση σειρά-σειρά, η single-row dynamic programming-based legalization αναπτύχθηκε για standard cell placements, και διαλέγει ένα υποσύνολο κελιών για να τα τοποθετήσει σε σειρά χρησιμοποιώντας τεχνικές δυναμικού προγραμματισμού.

Μια ιδέα της τεχνικής δίνεται στον αλγόριθμο που παρουσιάζεται πιο κάτω. Το legalization πραγματοποιείται από την κάτω γραμμή στην πάνω (αν και μπορεί να συμβεί και το αντίστροφο). Τα κελιά πρώτα ταξινομούνται με βάση τις αρχικές Y-συντεταγμένες σε μη-φθίνουσα σειρά. Έστω ότι το L_s δηλώνει την ταξινομημένη λίστα των κελιών. Ένα σύνολο από υποψήφια κελιά C_{cand}^i επιλέγονται στη συνέχεια για ανατεθούν σε μια συγκεκριμένη σειρά, έστω R_i . Το συνολικό πλάτος των κελιών στο C_{cand}^i είναι μεγαλύτερο της χωρητικότητας του R_i και ικανοποιεί την ακόλουθη σχέση:

$$[k_1 * width(R_i) \leq width(C_{cand}^i) = \sum_{j=1}^m width(c_j) \leq [k_2 * width(R_i)]$$

όπου

$$k_1, k_2 \in \mathcal{R} \text{ και είναι σταθερές έτσι ώστε } 1 < k_1 \leq k_2$$

$width(R_i)$ είναι το πλάτος της γραμμής R_i

$width(C_{cand}^j)$ είναι το συνολικό πλάτος των κελιών στο C_{cand}^j

$width(c_j)$ είναι το πλάτος του κελιού $c_j \in C_{cand}^j$

$$m = |C_{cand}^j|$$

Να σημειωθεί ότι τα κελιά στο C_{cand}^i είναι τα ελάχιστα ικανά κελιά m του L , τα οποία ικανοποιούν την παραπάνω σχέση.

Ο αλγόριθμος Tetris τοποθετεί ένα κελί τη φορά στη μια πλευρά της περιοχής του placement, όπου αυτή η προσέγγιση βρίσκει ένα σύνολο κελιών για να γεμίσει μια ολόκληρη σειρά. Η μέθοδος που χρησιμοποιείται για να βρεθεί το σύνολο των κελιών είναι παρόμοια με την dynamic programming technique για την επίλυση του κλασικού προβλήματος 0-1 σακιδίου.

Στο πρόβλημα σακιδίου, υπάρχει ένα όριο στο βάρος του σακιδίου. Η αναλογία του περιορισμού αυτού είναι το όριο στο συνολικό πλάτος κελιού στη γραμμή. Ομοίως, η τιμή ενός στοιχείου στο πρόβλημα του σακιδίου διαμορφώνεται από τη φυσική μετατόπιση του κάθε κελιού.

Μετά τον προσδιορισμό του συνόλου των υποψηφίων κελιών C_{cand}^i , ταξινομούνται από τις X-συντεταγμένες σε μια μη-φθίνουσα σειρά. Τα κελιά αυτά θεωρείται πως εντάσσονται στη R_l με αυτή τη σειρά ταξινόμησης. Το πρόβλημα του σακιδίου επιλύεται χρησιμοποιώντας dynamic programming, με τα επιλεγμένα κελιά να είναι ήδη τοποθετημένα και τα εναπομείναντα κελιά να είναι διαθέσιμα για legalization στην επόμενη γραμμή. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου να επέλθει legalization σε όλα τα κελιά. Ο παρακάτω αλγόριθμος απεικονίζει αυτή την προσέγγιση.

Algorithm Row-by-row dynamic programming based legalization.
 (The overall approach is a variation of the classic method for the 0-1 knapsack problem)
 {C}= All cells to be legalized;
 Sort the cells in C by their Y-coordinates to get L_s ;
for $i=1$ to N_{Y-1} **do**
 Select a set of candidate cells (C_{cand}^i) from L_s ;
 Assign a subset of cells C_{assign}^i from C_{cand}^i to row R_i ;
 $L_s = L_s \setminus C_{assign}^i$;
end for
 Assign the remaining cells in L_s to row R_{N_Y} ;

Εικόνα 16: Αλγόριθμος Dynamic programming με 0-1 knapshack

Κάθε κελί c_j έχει δύο παράγοντες που συσχετίζονται με αυτό: (1) το κόστος της ανάθεσης του c_j στη γραμμή R_i , που συμβολίζεται με w_j^i και (2) η ποινή μη ανάθεσης του c_j στη γραμμή R_i , που συμβολίζεται με p_j^i . Είναι προφανές ότι εάν το c_j ανατεθεί στη γραμμή R_i , τότε $p_j^i = 0$ ενώ αν δεν ανατεθεί τότε $w_j^i = 0$.

Αν το κελί c_j πρόκειται να ανατεθεί στη γραμμή R_i , τότε το κόστος της ανάθεσης δίνεται από την παρακάτω εξίσωση:

$$w_j^i = (x_j^i - x_j)^2 + (y_j^i - y_j)^2$$

όπου

(x_j^i, y_j^i) είναι η περιοχή του R_i στην οποία πρόκειται να τοποθετηθεί το c_j

(x_j, y_j) είναι η αρχική τοποθέτηση του c_j

Από την άλλη μεριά, η ποινή της μη ανάθεσης του c_j στη γραμμή R_i δίνεται από τον τύπο:

$$p_i^j = (y_j^{i+1} - y_i)^2$$

όπου

y_j^{i+1} είναι η y – συντεταγμένη του κέντρου της γραμμής R_{i+1}

y_j είναι η αρχική y – συντεταγμένη του c_j

4.4 Όρια του Legalization και του Detailed Placement

Όλες οι λειτουργίες που πραγματοποιούνται στο legalization και το detailed placement είναι κατά μία έννοια τοπικές, και συνεπώς αποτυγχάνουν να αντιμετωπίζουν στόχους global optimization. Αυτοί οι αλγόριθμοι τυπικά εφαρμόζονται με έναν διαδραστικό τρόπο. Πολλές ομάδες έχουν πειραματιστεί με διάφορους legalizers, όπως windows sizes για αναδιάταξη, μεθόδους κατανομής του χώρου και άλλα.

Δεν υπάρχει απλή μέθοδος και μία σωστή στρατηγική. Ευτυχώς, τα περισσότερα εργαλεία placement από ακαδημαϊκές ομάδες μπορούν να τρέξουν σε legalization mode αλλά και σε detailed placement mode, καθιστώντας έτσι εύκολη την χρήση διαφορετικών τεχνικών. Όταν συγκρίνονται με τους χρόνους τρεξίματος για το global placement, οι χρόνοι του detailed placement είναι συνήθως μικροί, για αυτό και αξίζει να γίνει διερεύνηση για διαφορετικούς συνδυασμούς.

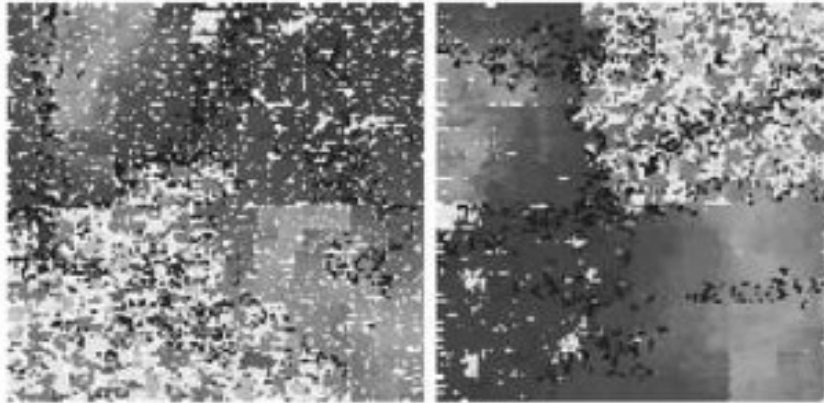
Η πρώτη βελτιστοποίηση που εφαρμόστηκε φυσιολογικά θα ήταν η εισαγωγή χώρου με στόχο την μείωση του routing congestion. Ως γενικός κανόνας, θα ήταν επιθυμητό η απαίτηση για δρομολόγηση σε οποιαδήποτε περιοχή να είναι λιγότερη από το 70 τοις εκατό των διαθέσιμων πόρων για δρομολόγηση. Το dense routing συνήθως καταλήγει στο να αποτύχει να δρομολογηθούν όλα τα nets, ή στη δημιουργία nets με μεγάλες παρακάμψεις. Το routing density που είναι δυνατό να επιτευχθεί εξαρτάται από το συγκεκριμένο εργαλείο δρομολόγησης που χρησιμοποιείται κάθε φορά. Η αύξηση του wirelength εξαιτίας της εισαγωγής χώρου (space insertion) μπορεί να είναι μικρότερη από αυτή που θα περιμέναμε. Αν τα στοιχεία του

κυκλώματος είναι διασπαρμένα. με την αύξηση της περιοχής του Placement κατά 10 τοις εκατό, το πραγματικό spreading που πραγματοποιείται είναι λιγότερο από 5 τοις εκατό σε οριζόντιες κατευθύνσεις. Σε αυτή την περίπτωση, το χειρότερο σενάριο είναι να υπάρξει αύξηση 5 τοις εκατό στο συνολικό wirelength.

Πηγαίνοντας από ένα global placement σε ένα legalized παρατηρείται μια μικρή αύξηση στο wirelength. Τα global placements που έχουν σχετικά μικρή επικάλυψη γενικά παρουσιάζουν και μικρότερες αυξήσεις wirelength. Αν υπάρχει περίσσεια χώρου για δρομολόγηση, είναι πολύ πιθανό το legalized placement να οδηγήσει σε μείωση του wirelength. Αν αντίθετα υπάρξει αύξηση τότε χρησιμοποιώντας αναδιάταξη, κατοπτρισμό κελιών και βελτιστοποιημένο γραμμικό placement μπορούμε να ανακτήσουμε ένα μέρος του χαμένου wirelength.

Παρόλα αυτά, ύστερα από λίγες επαναλήψεις οι βελτιώσεις οποιασδήποτε τεχνικής γίνεται ασυμπτωτικά μικρή. Τα πρώτα ποσοστά βελτίωσης του wirelength προκύπτουν εύκολα, αλλά στη συνέχεια η πρόοδος είναι πιο αργή. Για διάφορους λόγους, είναι απόφαση του σχεδιαστή του σχεδίου το κατά πόσον θα εφαρμόσει detailed placement. Για μερικούς, μια μικρή βελτίωση στο wirelength έχει μεγάλη σημασία, δικαιολογώντας τον επιπλέον χρόνο υπολογισμού. Για άλλους, μια γρήγορη λύση ίσως είναι ικανοποιητική.

Θα πρέπει να αναφερθεί εδώ ότι μια από τις πιο κρίσιμες εκτιμήσεις είναι η ελαχιστοποίηση του ποσοστού της επικάλυψης κατά τη διάρκεια του global placement, καθώς και η διασφάλιση ότι δεν υπάρχουν μεγάλα τμήματα της σχεδίασης τα οποία έχουν απαιτήσεις περιοχών που να ξεπερνάνε τον διαθέσιμο χώρο. Οι μέθοδοι legalization διαχειρίζονται τη διανομή του χώρου ατελώς, για αυτό και όταν τα προβλήματα μπορούν να επιλυθούν με global placement τότε τα αποτελέσματά τους είναι σχεδόν σίγουρα καλύτερα. Η σύγκριση των αποτελεσμάτων του placement πριν και μετά το legalization μπορεί να μας δώσει επίγνωση για το τι συμβαίνει κατά τη διάρκεια του legalization. Υπάρχει ένας αριθμός εργαλείων οπτικοποίησης για τα placements. Κάθε περιοχή η οποία αλλάζει σημαντικά κατά τη διάρκεια του legalization πρέπει να εξετάζεται προσεκτικά.



Εικόνα 17: Αποτελέσματα από το CAPO και το Feng Shui

Αν και έχουν πραγματοποιηθεί σημαντικές βελτιώσεις στο placement, οι λύσεις που δίνονται δεν είναι μακριά από την βέλτιστη. Πρόσφατα, ο Chang παρουσίασε ένα σύνολο από benchmark circuits με γνωστές βέλτιστες διαμορφώσεις (PEKO). Αυτά τα παραδείγματα του placement με γνωστές βέλτιστες σχεδιάσεις προσέλκυσαν μεγάλο ενδιαφέρον, και τα αποτελέσματα πολλών εργαλείων placement στα benchmarks ήταν απρόσμενα. Στην παραπάνω φωτογραφία παρουσιάζονται τα αποτελέσματα της χαρτογράφησης μιας εικόνας μέσα στο βέλτιστο placement ενός σύνθετου PEKO benchmark, και στη συνέχεια η αναδιάταξη των placements με στόχο να αντιστοιχούν στα αποτελέσματα ενός αριθμού ακαδημαϊκών εργαλείων.

Τα benchmarks του PEKO δεν περιλαμβάνουν pads, και για το λόγο αυτό υπάρχουν πολλαπλές βέλτιστες διαμορφώσεις, οι οποίες αντιστοιχούν στον κατοπτρισμό και στην αναστροφή του σχεδίου. Από τις στρεβλώσεις των εικόνων, θα έπρεπε να είναι σαφές ότι πολλά εργαλεία placement είναι σε γενικά σωστά. Σε πολύ υψηλό επίπεδο τα placements ταυτίζονται με το βέλτιστο αποτέλεσμα. Σε τοπικό επίπεδο, ωστόσο, υπάρχει ένα μεγάλο ποσοστό με αποτελέσματα κάτω από το βέλτιστο. Τα τμήματα του κάθε placement τεντώνονται και κυρτώνουν καταλήγοντας έτσι σε μεγαλύτερα μήκη διασύνδεσης.

Αν και υπάρχει κάποια αμφιβολία σχετικά με το πόσο κοντά βρίσκονται τα benchmarks του συνθετικού PEKO με το πραγματικό κύκλωμα, είναι προφανές ότι τα αποτελέσματα του placement μπορούν να βελτιωθούν, και η βελτίωση αυτή πρέπει να καλύπτει τόσο τα τοπικά όσο και τα γενικά επίπεδα.

5. Σύγκριση Legalizer αλγορίθμων

Σε αυτό το κεφάλαιο περιγράφονται οι αλγόριθμοι που γράφτηκαν για αυτή τη διπλωματική καθώς και η διαδικασία για το testing αυτών.

5.1 Legalizer αλγόριθμοι προς σύγκριση

Υλοποιήθηκαν οι παρακάτω αλγόριθμοι legalization:

Legalizer αλγόριθμος 1

1. Τοποθέτησε τα cells στη σειρά που βρίσκονται πιο κοντά
2. αξιόμησε τα cells κάθε σειράς ως προς το μέγεθος τους. Φθίνουσα ταξινόμηση
3. Τοποθέτησε τα cells κάθε σειράς. Εάν υπερβαίνουν το όριο τότε τοποθέτησε τα στην πιο κοντινή σειρά που χωράνε.

Legalizer αλγόριθμος 2

1. Τοποθέτησε τα cells στη σειρά που βρίσκονται πιο κοντά
2. Ταξινόμησε τα cells κάθε σειράς ως προς την κάτω αριστερά συντεταγμένη. Αύξουσα ταξινόμηση.
3. Τοποθέτησε τα cells κάθε σειράς. Εάν υπερβαίνουν το όριο τότε τοποθέτησε τα στην πιο κοντινή σειρά που χωράνε.

Legalizer αλγόριθμος 3

1. Τοποθέτησε τα cells στη σειρά που βρίσκονται πιο κοντά
2. Ταξινόμησε τα cells κάθε σειράς ως προς το μέγεθος τους. Αύξουσα ταξινόμηση.
3. Τοποθέτησε τα cells κάθε σειράς. Εάν υπερβαίνουν το όριο τότε τοποθέτησε τα στην πιο κοντινή σειρά που χωράνε.

Ο αλγόριθμος 1 χρησιμοποιείται στον NTUPlace και υπόλοιποι είναι παραλλαγές του.

Επίσης υλοποιήθηκε ο αλγόριθμος Tetris όπως περιγράφεται στο κεφάλαιο 4. 2.

Τέλος υλοποιήθηκε και ένας δικός μας αλγόριθμος:

Αλγόριθμος 5

1. Τοποθέτησε τα cells στη σειρά που βρίσκονται πιο κοντά
2. Ταξινόμησε τα cells κάθε σειράς ως προς την κάτω αριστερά συντεταγμένη. Αύξουσα ταξινόμηση.
3. Εάν τα cells δεν χωράνε στη σειρά, ψάξε τις σειρές πάνω-κάτω, και ταξινόμησε τα ως προς την κάτω αριστερά συντεταγμένη στην σειρά που χωράνε.
4. Τοποθέτησε τα cells κάθε σειράς, ώστε να μην υπάρχει επικάλυψη.

5.2. Περιβάλλον testing

Για το testing των αλγορίθμων χρησιμοποιήθηκαν τα κυκλώματα ISCAS '89. Αυτά δόθηκαν ως input σε έναν GORDIAN placer και το output ήταν οι θέσεις των cells στο χώρο μετά από την πρώτη επανάληψη. Τα cells σε αυτή τη φάση του placement έχουν αρκετά μεγάλη επικάλυψη μεταξύ τους και δεν έχουν διανεμηθεί στο χώρο αρκετά. Αυτό οδηγεί τους legalizers να έχουν αρκετά μεγάλο βάρος για τη τελική ταξινόμηση των cells. Επιπλέον ο χώρος που χρησιμοποιείται είναι στο 110% του συνολικού εμβαδού που καταλαμβάνουν τα cells, επομένως και ο περιορισμός στο χώρο είναι αρκετά μεγάλος.

Τα αποτελέσματα που παρουσιάζονται σε αυτή τη διπλωματική είναι το τελικό wirelength που έχουν τα κυκλώματα μετά τη διαδικασία του legalization (βλ. Πίνακα 1) και την ποσοστιαία αύξηση του (βλ. Πίνακα 2). Η μέτρηση του wirelength έγινε με χρήση bounding-box. Ο χρόνος που χρειάστηκε στους legalizers για να δώσουν την τελική θέση στα cells είναι σχεδόν ίδιος και οι

αλγόριθμοι έχουν παρόμοια πολυπλοκότητα, οπότε δεν χρειάζεται να αναφερθούμε.

	Initial	Algor. 1	Algor. 2	Algor. 3	Algor. 4	Algor. 5
S27	55.379698	99.734167	89.117334	96.844840	105.965667	82.435061
S208	227.470812	472.794373	522.290384	489.267269	528.135970	442.167826
S298	359.624464	911.243671	840.031050	892.122219	987.507931	838.789742
S344	682.205950	2182.638069	2013.598356	2104.380058	2337.491851	2041.42553
S349	614.241067	1831.468523	1748.573277	1889.407141	2011.752367	1753.78492
S382	650.876732	1496.588634	1470.693534	1605.159140	1615.654168	1439.19326
S386	378.448846	1190.317355	1102.645919	1184.602554	1559.607460	1109.09490
S400	624.724891	1629.675964	1557.829316	1593.440830	1547.468423	1427.03691
S420	897.148331	2391.115900	2219.490605	2395.956367	2755.529728	2062.34249
S444	655.467966	1878.111251	1612.024447	1809.067610	2096.326226	1668.43583
S510	847.728569	3372.688000	3237.952651	3310.802802	4198.506850	2984.18572
S526	662.402475	1765.086902	1569.889712	1708.546408	1868.417030	1621.04695
S641	1626.89259	4234.493114	3697.977322	4225.703994	4690.298282	3764.10078
S713	1606.97303	4342.250571	3698.057762	4254.629870	4534.836546	4030.93399
S820	882.556443	2982.792304	2946.765375	3078.116702	3817.668410	2708.85513
S832	1075.32001	3934.188008	3785.633614	4256.122131	4851.501488	3717.03841
S838	1890.47179	5635.236216	5009.743764	5293.642621	5429.814519	4565.09936
S953	1423.94483	5800.865748	5508.055138	5516.155066	7823.445779	5419.35085
S1196	1853.84581	8111.993565	7638.795339	8807.080071	10230.976214	7669.98044
S1238	2005.36308	9832.053680	9319.706178	10498.26188	11651.917729	8949.06179
S1423	4817.19139	14822.35826	13532.78589	15662.13295	16069.879527	14855.6466
S1488	1334.59576	7620.486107	7338.050550	7814.727037	10766.018699	6915.17213
S1494	1485.3915	9182.232898	9344.545760	9316.235674	13575.538690	8864.68051

Πίνακας 1: Αποτελέσματα wirelength μετά την εφαρμογή των legalizers.

	Algor. 1 %	Algor. 2 %	Algor. 3 %	Algor. 4 %	Algor. 5 %
S27	80.091569	60.920585	74.874265	91.343887	48.854298
S208	107.848369	129.607649	115.090132	132.177467	94.384423
S298	153.387565	133.585624	148.070504	174.594203	133.240457
S344	219.938293	195.159894	208.466975	242.637271	199.238893
S349	198.167710	184.672154	207.600264	227.518376	185.520622
S382	129.934266	125.955770	146.614921	148.227366	121.116103
S386	214.525297	191.359303	213.015237	312.105223	193.063358
S400	160.862979	149.362454	155.062805	147.703981	128.426452
S420	166.524031	147.393940	167.063571	207.143159	129.877537
S444	186.529830	145.934894	175.996342	219.821309	154.541170
S510	297.849987	281.956297	290.549867	395.265466	252.021369
S526	166.467438	136.999373	157.931767	182.066734	144.722357
S641	160.281050	127.303102	159.740810	188.297968	131.367504
S713	170.213034	130.125690	164.760503	182.197428	150.840177
S820	237.971846	233.889736	248.772787	332.569321	206.932792
S832	265.862067	252.047166	295.800515	351.168159	245.668115
S838	198.086236	164.999657	180.017012	187.220076	141.479369
S953	307.379951	286.816612	287.385449	449.420569	280.587137
S1196	337.576499	312.051278	375.070799	451.878489	313.733462
S1238	390.287958	364.739093	423.509283	481.037811	346.256435
S1423	207.697101	180.926889	225.129971	233.594375	208.388133
S1488	470.995828	449.833198	485.550117	706.687614	418.147317
S1494	518.169210	529.096488	527.190588	813.936743	496.790847

Πίνακας 2: Ποσοστό αύξησης wirelength κατά τη διαδικασία του legalization

6. Συμπεράσματα και μελλοντική έρευνα

Το πρώτο συμπέρασμα στο οποίο μπορεί να οδηγηθεί κανείς είναι αυτό που έχουμε επαναλάβει για τους legalizers, ότι χειροτερεύουν το συνολικό wirelength. Στη συγκεκριμένη περίπτωση κατά πολύ καθώς η αρχική θέση των cells προέρχεται από την πρώτη επανάληψη του Gordian placer. Αυτό σημαίνει ότι τα cells είναι συγκεντρωμένα στο κέντρο του κυκλώματος και δεν υπάρχει σε αυτό το στάδιο καλώς διαχωρισμός μεταξύ τους.

Αυτό μας οδηγεί στο συμπέρασμα πως για να είναι ένας legalizer αποδοτικός θα πρέπει να έχει αρκετά καλό initial placement και πως ουσιαστικά από αυτό εξαρτάται το τελικό placement. Δεν θα πρέπει ο legalizer να αποφασίζει σε μεγάλο βαθμό για τη θέση των cells στο χώρο.

Τέλος ο αλγόριθμος μας φαίνεται να έχει τα καλύτερα αποτελέσματα από τους υπόλοιπους αλγορίθμους. Αυτό θα πρέπει να επαληθευτεί και για μεγαλύτερα κυκλώματα.

Το επόμενο βήμα που πρέπει επομένως να γίνει είναι να επαληθευτούν τα αποτελέσματα για μεγαλύτερα κυκλώματα. Επίσης το testing αυτό θα πρέπει να επαναληφθεί και για καλύτερη αρχική θέση από global placer. Ίσως με αυτόν τον τρόπο τα αποτελέσματα των legalizers δεν θα έχουν τόσο μεγάλη ποσοστιαία απόκλιση. Επιπλέον καλό θα ήταν να γίνει μία έρευνα κατά πόσο και εάν το profile των κυκλωμάτων επηρεάζουν την απόδοση των legalizers.

Ένα ακόμα ενδιαφέρον αποτέλεσμα που θα πρέπει να μελετηθεί είναι ο λόγος για τον οποίο ενώ ο legalizer μας σχεδόν σε όλα τα κυκλώματα είναι καλύτερος τι το επηρεάζει σε ορισμένα και ο αλγόριθμος 2 παρουσιάζει καλύτερα αποτελέσματα.

Τέλος, ο αλγόριθμος μπορεί να βελτιωθεί χρησιμοποιώντας ένα μικρό κομμάτι επανάληψης ώστε να παρουσιάσει τοπική βελτίωση. Ή ακόμα θα μπορούσε να υπολογίζει το wirelength κατά τη μετακίνηση των cells ώστε να αποφασίζει εάν είναι συμφέρουσα ή όχι. Μία τελευταία λύση θα μπορούσε να είναι ένα εργαλείο το οποίο τρέχει παράλληλα όλους τους πιθανούς legalizers και να βρίσκει το καλύτερο αποτέλεσμα. Ούτως ή άλλως στη βιβλιογραφία έχει παρουσιαστεί ως πρόταση να τρέχουν όλοι οι πιθανοί τρόποι σε έναν αλγόριθμο, αφού ο καθένας παρουσιάζει διαφορετικά αποτελέσματα.

7. Βιβλιογραφία

- [1] Myung-Chul Kim, Dong-Jin Lee and Igor L. Markov. SimPL: An Effective Placement Algorithm. , Department of EECS, University of Michigan. IEEE, 2010.
- [2] Charles J. Alpert , Dinesh P.Mehta and Sachin S. Sapatnekar. Handbook of Algorithms for Physical Design Automation. Chapter 21 Timing driven placement. Auerbach Publications 2008.
- [3]Jens Egeblad. Placement Techniques for VLSI Layout Using Sequence-Pair Legalization. Master of Science Thesis,Department of Computer Science University of Copenhagen. 2003.
- [4] Ameya R. Agnihotri , Satoshi Ono and Patrick H. Madden. Recursive Bisection Placement: Feng Shui 5.0 Implementation Details. SUNY Binghamton CSD and university of Kitakyushu. 2005.
- [5] Natarajan Viswanathan and Chris Chong-Nuen 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, vol 24, no 5, May 2005.
- [6] Jens Vygen and Bernhard Korte. Flow-based Partitioning and Fast Global Placement in Chip Design Dissertation. Rheinischen Friedrich-Wilhelms university, Bonn, Juli 2010.
- [7] Ameya Agnihotri, Mehmet Can, YILDIZ Ateen Khatkhate, Ajita Mathur, Satoshi Ono, Patrick H. Madden. Fractional cut: Improved recursive bisection placement. SYNY Binghamton Computer Science Department. 2003.
- [8] Naveed Sherwani. Algorithms for VLSI Physical Design Automation – Third edition. Chapters 5-7. eBook ISBN 0-306-47509-X. 1999.
- [9] Konrad Doll, Frank M. Johannes, and Kurt J. Antreich. Iterative Placement Improvement by Network Flow Methods. IEEE transations on computer-aided design of integrated circuits and systems, vol 13, no 10. October 1994.
- [10] Min Pan, Natarajan Viswanathan and Chris Chu. An Efficient and Effective Detailed Placement Algorithm. Department of Electrical and Computer Engineering, Iowa State University. 2005.

- [11] Ulrich Brenner, Anna Pauli, and Jens Vygen. Almost Optimum Placement Legalization by Minimum Cost Flow and Dynamic Programming. Research Institute for Discrete Mathematics, University of Bonn. 2004.
- [12] Sung Kyu Lim. Practical problems in VLSI physical design automation. Chapters 1-4. e-ISBN 978-1-4020-6627-6. 2008.
- [13] Zhe-Wei Jiang, Hsin-Chen Chen, Tung-Chieh Chen, and Yao-Wen Chang. Challenges and Solutions in Modern VLSI Placement. Institute of Electronics Engineering, National Taiwan University. 2007.
- [14] Ameya R Agnihotri. Combinational Optimization techniques for VLSI Placement. Chapters 1-3. 2007.
- [15] Brent Goplen. Advanced placement techniques for future VLSI circuits. Thesis, University of Minnesota. 2006.
- [16] Yao-Wen Chang, Zhe-Wei Jiang, and Tung-Chieh Chen. Essential Issues in Analytical Placement Algorithms. 2007.
- [17] Hidetoshi Onoderat, Yo Taniguchi, and Keikichi Tamaru. Branch-and-Bound Placement for Building Block Layout. 1991.
- [18] U.Brenner J.Vygen. Faster Optimal Single-Row Placement with Fixed Ordering. 2000.
- [19] Peter Spindler, Ulf Schlichtmann and Frank M. Johannes. Abacus: Fast Legalization of Standard Cell Circuits with Minimal Movement. Insitute for Electronic Design Automation Munich. 2008.
- [20] Yu-Min Lee, Tsung-You Wu, and Po-Yi Chiang. A Hierarchical Bin-Based Legalizerfor Standard-Cell Designs with Minimal Disturbance. Department of Electrical Engineering National ChiaoTungUniversity. ASPDAC, January 2010.
- [21] Ulrich Brenner. Theory and Practice of VLSI Placement. Phd thesis.2006.
- [22] Tony Chan, Jason Cong, Kenton Sze. Multilevel generalized force-directed method for circuit placement. UCLA. 2005.
- [23] Tsung-Yi Ho, Sheng-Hung Liu. Fast Legalization for Standard Cell Placement with Simultaneous Wirelength and Displacement Minimization. 2012.
- [24] Igor L. Markov, Jin Hu and Myung-Chul Kim. Progress and Challenges in VLSI Placement Research. University of Michigan. 2012.
- [25] www.wikipedia.com
- [26] www.asic-world.com