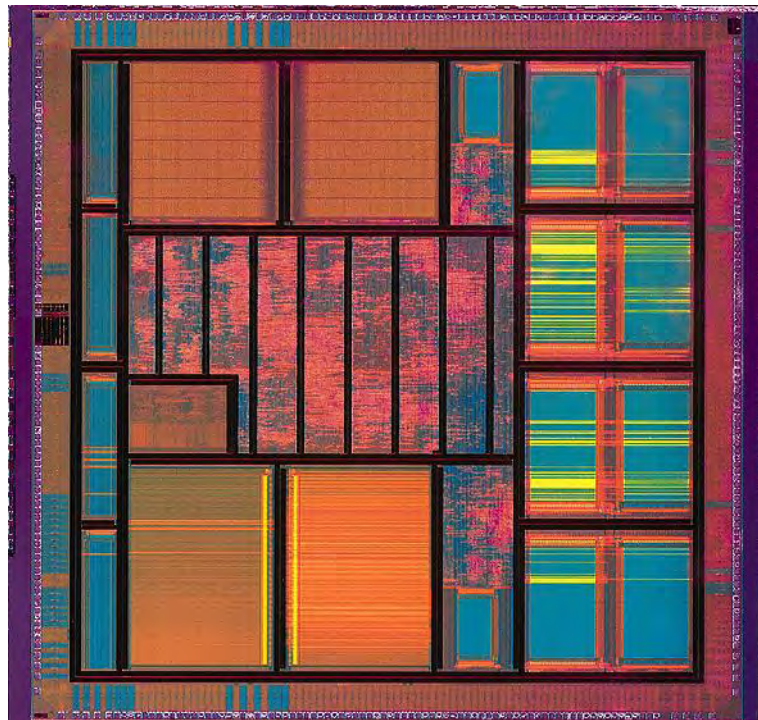


Υλοποίηση ενός Detail Placer αλγορίθμου για ολοκληρωμένα κυκλώματα

ΤΣΑΚΑΤΙΚΑ Π. ΗΛΙΤΣΑ

Βόλος, 10/13/2017



Copyright © Tsakatika P. Ilitsa, 2017
All rights reserved

Ευχαριστίες

Μετά το πέρας αυτής της διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές κ. Γεώργιο Σταμούλη και κ. Νέστωρ Ευμορφώπουλο, για την ευκαιρία που μου δασών να πραγματοποιήσω αυτή τη μελέτη, την εμπιστοσύνη που μου έδειξαν, την άριστη συνεργασία και την συνεχή καθοδήγησή τους. Επίσης, θα ήθελα να ευχαριστήσω τον διδακτορικό φοιτητή και φίλο Γιάννη Αρβανιτάκη, που με την ανεκτίμητη βοήθειά και υποστήριξη του, κατάφερα να πραγματοποιήσω αυτή τη μελέτη.

Τέλος, οφείλω ένα πολύ μεγάλο ευχαριστώ στην οικογένεια και τους φίλους μου για την αμέριστη υποστήριξη και ατελείωτη υπομονή που έδειξαν τόσο κατά τη διάρκεια των σπουδών μου, όσο και κατά την εκπόνηση αυτής της εργασίας.

Περιεχόμενα

1. Τι είναι Ολοκληρωμένο Κύκλωμα	1
2. Σχεδιασμός ολοκληρωμένων κυκλωμάτων	1
2.1 Φυσικός Σχεδιασμός	4
2.2 Εργαλεία CAD και EDA	11
3. Placement	15
3.1 Placement εντός της EDA ροής σχεδίασης	15
3.2 Στόχοι του Placement.....	16
3.3 Placement Flow.....	18
3.4 Μετρήσεις Καλωδίων	19
4. Detailed Placement και Αλγόριθμοι	21
4.1 Simulated Annealing	22
4.2 Branch and Bound.....	23
4.3 The Domino algorithm	26
4.4 The FastDP algorithm.....	29
5. Σύγκριση Detailed Placement αλγορίθμων	32
5.1 Detailed placer αλγόριθμοι προς σύγκριση	32
5.2 Περιβάλλον testing	35
5.3 Συμπεράσματα και Μελλοντική Έρευνα	37
6. Βιβλιογραφία.....	39

1. Τι είναι Ολοκληρωμένο Κύκλωμα

Ολοκληρωμένο κύκλωμα (integrated circuit) ονομάζεται ένα κύκλωμα αποτελούμενο από λογικές πύλες, που συνδέονται μεταξύ τους πάνω σε ένα φύλλο ημιαγωγού, κατά κύριο λόγο πυριτίου. Η ορολογία chip, που αποτελεί την εναλλακτική ονομασία του ολοκληρωμένου κυκλώματος, προκύπτει από την αγγλική ονομασία του φύλλου (chip) και όταν αναφερόμαστε σε chip της κλίμακας των μικρομέτρων, χρησιμοποιούμε τον όρο microchip.

Τα ολοκληρωμένα κυκλώματα χρησιμοποιούνται σχεδόν σε κάθε στοιχείο ηλεκτρονικού εξοπλισμού και θεωρούνται επανάσταση στον τομέα της ηλεκτρονικής.

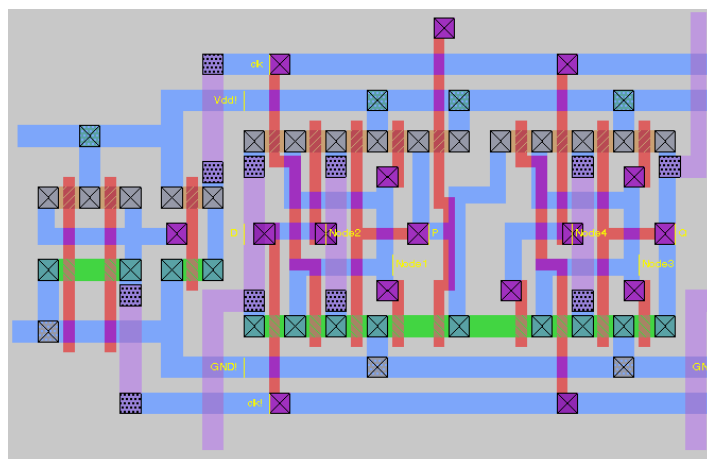
2. Σχεδιασμός ολοκληρωμένων κυκλωμάτων

Η σχεδίαση ολοκληρωμένου κυκλώματος (Integrated Circuit design ή IC design) είναι ένας τομέας της ηλεκτρονικής μηχανικής, που περιλαμβάνει τη συγκεκριμένη λογική και τις τεχνικές σχεδιασμού κυκλωμάτων που απαιτούνται για να σχεδιαστεί ένα ολοκληρωμένο κύκλωμα. Τα ολοκληρωμένα κυκλώματα είναι το αποτέλεσμα της ενσωμάτωσης μικροσκοπικών ηλεκτρονικών στοιχείων πάνω σε ένα ηλεκτρικό δίκτυο σε μονολιθικό υπόστρωμα ημιαγωγών με φωτολιθογραφία.

Η σχεδίαση ολοκληρωμένου κυκλώματος χωρίζεται σε δύο ευρείες κατηγορίες, την ψηφιακή και την αναλογική σχεδίαση. Ο ψηφιακός σχεδιασμός χρησιμοποιείται στην παραγωγή εξαρτημάτων μικροεπεξεργαστών FPGA, μνημών(RAM, ROM, flash) καθώς και στον σχεδιασμό ψηφιακών ASIC. Αυτού του είδους ο σχεδιασμός εστιάζει στην λογική ορθότητα, στο να υπάρχει όσο το δυνατόν μεγαλύτερη πυκνότητα κυκλώματος και την τοποθέτηση των κυκλωμάτων με τρόπο τέτοιο ώστε τα σήματα ρολογιού και χρονισμού να δρομολογούνται αποτελεσματικά.

Όσον αφορά την αναλογική σχεδίαση ολοκληρωμένου κυκλώματος. Αυτό το δεύτερο είδος σχεδίασης εξειδικεύεται στην ισχύ και την αναλογία συχνοτήτων των ολοκληρωμένων κυκλωμάτων. Η αναλογική σχεδίαση βρίσκει εφαρμογή στον σχεδιασμό γραμμικών ρυθμιστών, των βρόγχων κλειδωμένης φάσης, των ταλαντωτών και των ενεργών φίλτρων. Σε μεγαλύτερο βαθμό ασχολείται με τα φυσικά χαρακτηριστικά των συσκευών ημιαγωγών, όπως το κέρδος, η κατανάλωση ισχύος και η αντίσταση. Η πιστότητα της ενίσχυσης και του φιλτραρίσματος ενός αναλογικού σήματος συνήθως είναι κρίσιμη και γι αυτό το λόγο στην αναλογική σχεδίαση χρησιμοποιούνται μεγαλύτερης έκτασης ενεργές συσκευές και είναι λιγότερο πυκνά τα κυκλώματα σε σχέση με την ψηφιακή σχεδίαση.

Τα σύγχρονα ολοκληρωμένα κυκλώματα είναι εξαιρετικά περίπλοκα. Πλέον το μέσο chip ενός ηλεκτρονικού υπολογιστή υπολογίζεται πως έχει πάνω από ένα 1 δις τρανζίστορ και οι κανόνες σχετικά με το τι μπορεί να κατασκευαστεί και τι όχι, σε ένα ολοκληρωμένο κύκλωμα, είναι εξίσου πολύπλοκοι καθώς και υπεράριθμοι. Η πολυπλοκότητα της μοντέρνας σχεδίασης ολοκληρωμένων κυκλωμάτων καθώς επίσης και η πίεση της αγοράς για την ταχεία παραγωγή σχεδίων, έχει οδηγήσει στην εκτεταμένη χρήση αυτοματοποιημένων εργαλείων σχεδιασμού. Εν ολίγοις, η σχεδίαση ολοκληρωμένων κυκλωμάτων με τη χρήση αυτοματισμών επιτυγχάνει τη σχεδίαση, τον έλεγχο και την επαλήθευση των εντολών που καλείται να εκτελέσει το ολοκληρωμένο κύκλωμα.



Εικόνα 1: Layout view of a simple CMOS flip-flop

❖ Βασικές Αρχές

Η σχεδίαση ολοκληρωμένων κυκλωμάτων έχει να κάνει με τη δημιουργία ηλεκτρονικών εξαρτημάτων, όπως τρανζίστορ, αντιστάσεις, πυκνωτές και της μεταλλικής διασύνδεσης αυτών των στοιχείων σε ένα φύλλο, συνήθως πυριτίου.

❖ Βήματα Σχεδιασμού

Τα βήματα για το σχεδιασμό ενός τυπικού ολοκληρωμένου κυκλώματος είναι τα εξής:

- i. Μελέτη σκοπιμότητας και εκτίμηση μεγέθους
- ii. Ανάλυση λειτουργιών
- iii. Σχεδιασμός επιπέδου συστήματος
- iv. Αναλογικός σχεδιασμός, προσομοίωση και διάταξη
- v. Ψηφιακός σχεδιασμός, προσομοίωση και σύνθεση
- vi. Προσομοίωση συστήματος και επαλήθευση
- vii. Σχεδιασμός για test και την αυτόματη δημιουργία προτύπου
- viii. Σχεδιασμός για την δυνατότητα κατασκευής του ολοκληρωμένου κυκλώματος
- ix. Tape-in
- x. Προετοιμασία mask data
- xi. Tape-out
- xii. Κατασκευή wafer
- xiii. Δοκιμή της μήτρας
- xiv. Συσκευασία
- xv. Τοποθέτηση πυριτίου, επικύρωση και ολοκλήρωση
- xvi. Χαρακτηρισμός της συσκευής
- xvii. Τυποποίηση (όταν είναι απαραίτητο)
- xviii. Δημιουργία datasheet
- xix. Αναβάθμιση
- xx. Παράγωγή
- xxi. Ανάλυση απόδοσης, ανάλυση εγγύησης και αξιοπιστία
- xxii. Ανάλυση αποτυχίας σε κάθε επιστροφή

xxiii. Πλάνο για τα chip επόμενης γενιάς
χρησιμοποιώντας πληροφορίες από την παραγωγή
αν είναι δυνατόν

Η ψηφιακή σχεδίαση, εν ολίγης, χωρίζεται σε τρία μέρη:

- Ηλεκτρονικός σχεδιασμός σε επίπεδο συστήματος: εδώ δημιουργούνται οι ηλεκτρονικές προδιαγραφές του χρήστη.
- RTL σχεδιασμός: εδώ μετατρέπονται οι προδιαγραφές που όρισε ο χρήστης σε μία περιγραφή RTL. Στο RTL περιγράφεται η ακριβής συμπεριφορά των ψηφιακών κυκλωμάτων στο chip, καθώς και οι διασυνδέσεις στις εισόδους και εξόδους.
- Φυσικός σχεδιασμός: σε αυτό το βήμα παίρνουμε το RTL και μία βιβλιοθήκη των διαθέσιμων λογικών πυλών και δημιουργούμε το σχέδιο του ολοκληρωμένου κυκλώματος.

Να επισημάνουμε ότι το δεύτερο βήμα, όπου δημιουργείται το RTL, είναι απολύτως υπεύθυνο ώστε το chip να είναι λειτουργικό. Το τρίτο βήμα του φυσικού σχεδιασμού (αν γίνει σωστά) δεν επηρεάζει με κανένα τρόπο τη λειτουργικότητα, αλλά καθορίζει την αποδοτικότητα του chip και το κόστος αυτού. Αυτό είναι που θα μας απασχολήσει, κυρίως, σε αυτή τη διπλωματική εργασία. Η φυσική σχεδίαση των ολοκληρωμένων κυκλωμάτων.

2.1 Φυσικός Σχεδιασμός

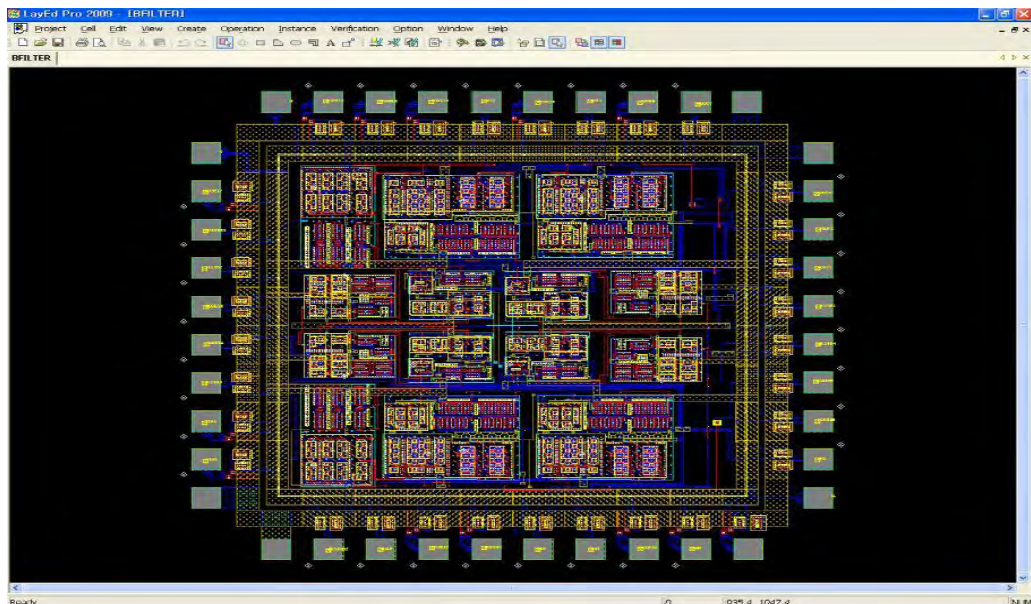
Στο στάδιο του φυσικού σχεδιασμού πραγματοποιείται η χαρτογράφηση του RTL σε πραγματικές γεωμετρικές αναπαραστάσεις όλων των ηλεκτρονικών συσκευών, όπως πυκνωτές, αντιστάσεις, λογικές πύλες και τρανζίστορ, που θα τοποθετηθούν στο ολοκληρωμένο κύκλωμα, οι οποίες, όταν κατασκευαστούν από τα αντίστοιχα υλικά, θα εξασφαλίσουν την απαιτούμενη λειτουργικότητα των στοιχείων. Οι γεωμετρικές αυτές αναπαραστάσεις συνθέτουν το σχέδιο του ολοκληρωμένου κυκλώματος. Η φυσική σχεδίαση συνήθως χωρίζεται σε ορισμένα

επιμέρους βήματα, τα οποία περιλαμβάνουν τόσο το σχεδιασμό όσο και την επαλήθευση και επικύρωση του layout.

2.1.1 Κατηγορίες Φυσικής Σχεδίασης

Γενικά, η φυσική σχεδίαση ολοκληρωμένων κυκλωμάτων κατηγοριοποιείται σε *Full Custom* σχεδίαση και *Semi Custom* σχεδίαση.

- **Full Custom:** η πλήρης ευελιξία του σχεδιαστή στο σχέδιο διάταξης και το ότι δεν χρησιμοποιούνται προκαθορισμένα κελιά είναι τα χαρακτηριστικά αυτής της κατηγορίας.
- **Semi Custom:** εδώ χρησιμοποιούνται προκαθορισμένες βιβλιοθήκες κελιών και η ευελιξία του σχεδιαστή περιορίζεται στην τοποθέτηση και διασύνδεση των κελιών.



Εικόνα 2: Φυσική Σχεδίαση Κυκλώματος

2.1.2 Ροή της Φυσικής Σχεδίασης

Τα βασικά βήματα της ροής της φυσικής σχεδίασης ASIC είναι τα παρακάτω:

- **Σχεδιασμός netlist:** ο φθισικός σχεδιασμός βασίζεται σε ένα netlist, το οποίο είναι το τελικό αποτέλεσμα της διαδικασίας της σύνθεσης. Η σύνθεση μετατρέπει το RTL, που συνήθως έχει κωδικοποιηθεί σε VHDL ή Verilog HDL, σε περιγραφές επιπέδου πύλης, τις οποίες το επόμενο set εργαλείων θα διαβάξει και κατανοεί. Το netlist αυτό περιέχει πληροφορίες όσον αφορά τα κελιά που χρησιμοποιούνται, τις διασυνδέσεις τους, την περιοχή που καταλαμβάνουν και χρησιμοποιούν και διάφορες άλλες λεπτομέρειες των κυκλωματικών στοιχείων.

Δύο εργαλεία που συναντώνται στη σύνθεση ψηφιακών κυκλωμάτων είναι ο Synopsis Design Compiler και ο Cadence RTL Compiler.

Κατά τη διαδικασία της σύνθεσης, για τη διασφάλιση ότι ο σχεδιασμός θα πληροί τις προδιαγραφές (απαιτούμενες λειτουργίες και ταχύτητα), εφαρμόζονται κάποιοι περιορισμοί. Τέλος, αφού γίνει η επαλήθευση του netlist για την ταχύτητα και την λειτουργικότητα του, το κύκλωμα προωθείται στην φυσική σχεδίαση.

- **Floorplanning:** Ουσιαστικά, το πρώτο βήμα στη ροή της φυσικής σχεδίασης είναι αυτό. Το floorplanning είναι η διαδικασία ταυτοποίησης των δομών που θα πρέπει να τοποθετηθούν και της δέσμησης χώρου που θα χρειαστούν αυτές, με τέτοιο τρόπο ώστε να ικανοποιούνται οι, πολλές φορές, αντικρουόμενες απαιτήσεις για διαθέσιμο χώρο, αποδοτικότητα και η επιθυμία να είναι όλα κοντά σε όλα. Στόχος του floorplanning είναι να χρησιμοποιήσουμε όσο το δυνατόν μικρότερη έκταση στο σχεδιασμό καθώς και η αύξηση της ταχύτητας του κυκλώματος, με αποτέλεσμα τη χρησιμοποίηση λιγότερων πόρων, η ελαχιστοποίηση του

μήκους καλωδίου που θα χρειαστεί και γρηγορότερες διαδρομές σήματος από το λεν άκρο στο άλλο. Συχνά απαραίτητος είναι ο επαναπροσδιορισμός της αρχιτεκτονικής που ορίστηκε, αφού η τοποθέτηση των στοιχείων στο χώρο γίνεται δυναμικά και πολλές φορές παρουσιάζει ψεγάδια.

- **Partitioning:** Το partitioning είναι μία διαδικασία διαίρεσης του ολοκληρωμένου κυκλώματος σε μικρότερα blocks, με στόχο τον διαχωρισμό τους, ανάλογα με τις λειτουργίες που πραγματοποιούν, επιτυγχάνοντας έτσι την ευκολότερη τοποθέτηση και δρομολόγηση. Αυτό το είδος διαχωρισμού, συχνά, συναντάται ως λογικός διαχωρισμός.
- **Placement:** Πριν ξεκινήσει η διαδικασία της βελτιστοποίησης της τοποθέτησης, όλα τα μοντέλα φορτίου του καλωδίου (WLM), που υπήρχαν κατά τη διαδικασία της σύνθεσης, αφαιρούνται και η RC καθυστέρηση υπολογίζεται από την εικονική δρομολόγηση (VR), η οποία εφαρμόζει απόσταση Manhattan μεταξύ δύο σημείων. Τα VR έχουν μεγαλύτερη ακρίβεια από τα WLM.

Το placement εκτελείται σε τέσσερις φάσεις βελτιστοποίησης:

- **Placement optimization:** βελτιστοποιεί το netlist πριν το placement και μπορεί να περιορίσει το μέγεθος των κελιών
- **In-placement optimization:** βελτιστοποιεί εκ νέου τη λογική, βάση του VR. Η βελτιστοποίηση εκτελεί επαναληπτικά διόρθωση των ρυθμίσεων του σταδιακού timing και του

placement, που προκύπτει από την συμφόρηση των κελιών.

- **Post-placement optimization before Clock Tree Synthesis (CTS):** κάνει βελτιστοποίηση των netlist με ιδανικά ρολόγια και είναι βασισμένη στο global routing.
 - **Post-placement optimization after CTS**
-
- **Clock Tree Synthesis:** Στόχος αυτής της διαδικασίας είναι η ελαχιστοποίηση της καθυστέρησης και της καθυστέρησης κατά την εισαγωγή. Έτσι, σε αυτό το βήμα υπολογίζεται με μεγαλύτερη ακρίβεια ο χρονισμός του κυκλώματος.

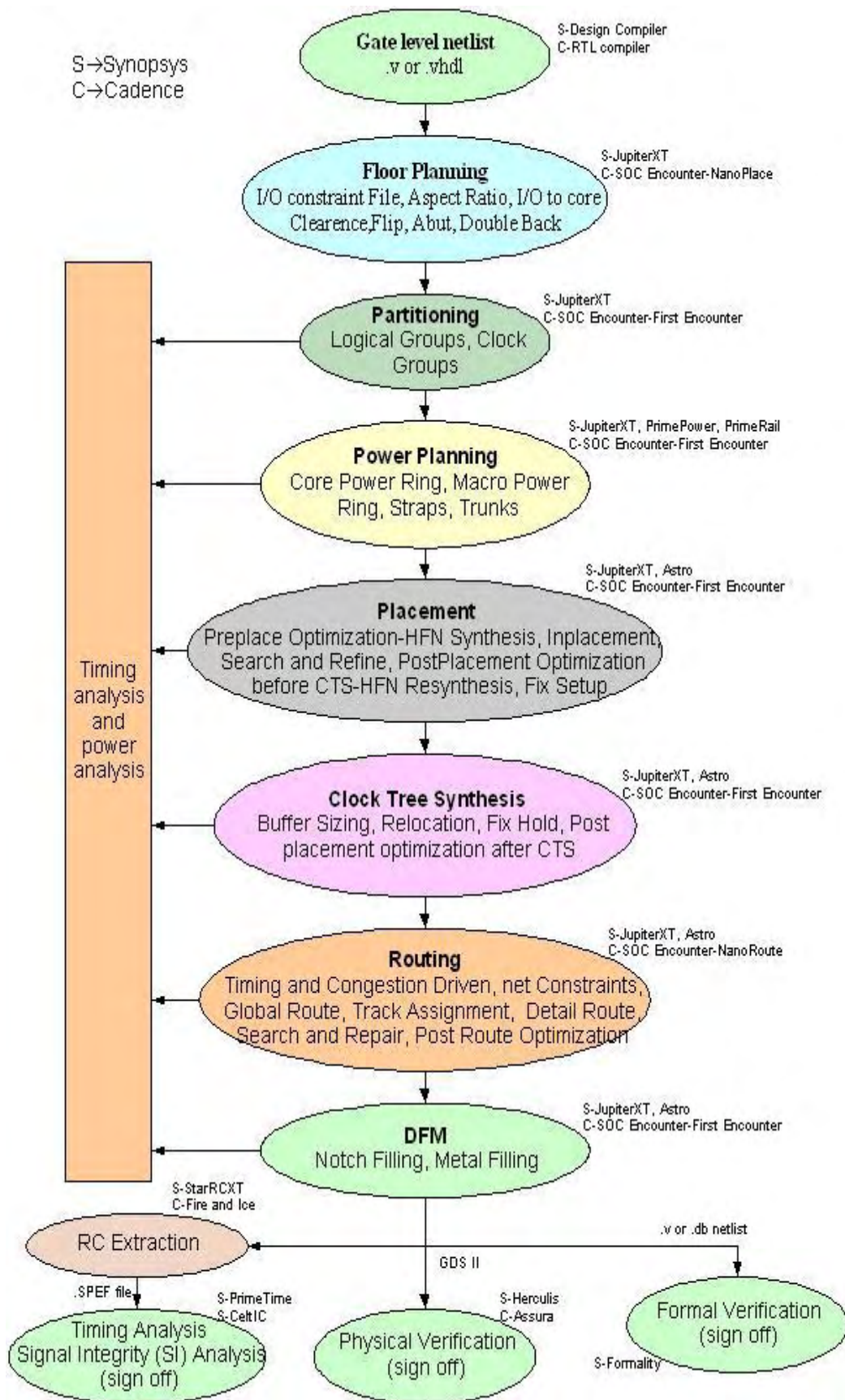
 - **Routing:** Όσον αφορά την δρομολόγηση υπάρχουν δύο τύποι, το global routing και το και το detailed routing.
 - Το global routing δεσμεύει τους πόρους δρομολόγησης, που χρησιμοποιούνται για τις συνδέσεις μεταξύ των στοιχείων του ολοκληρωμένου κυκλώματος.
 - Το detailed routing κάνει τις πραγματικές συνδέσεις. Κάποιοι απ' τους περιορισμούς που θα πρέπει να λαμβάνονται υπ' όψιν, κατά τη διάρκεια της δρομολόγησης είναι το μήκος καλωδίου, ο χρόνος κλπ.

 - **Physical Verification:** Η φυσική επαλήθευση ελέγχει την ορθότητα του layout που έχει παραχθεί. Αυτό σημαίνει ότι το layout:
 - Συμμορφώνεται πλήρως με όλους τους τεχνολογικούς κανόνες (Design Rule Checking-DRC)
 - Είναι σύμφωνο με την αρχική netlist (Layout vs Schematic-LVS)

- Δεν έχει Antenna effects (Antenna Rule Checking-ARC)
- Συμμορφώνεται με όλους τους ηλεκτρικούς κανόνες (Electrical Rule Checking-ERC)

Επίσης, θα πρέπει να γίνεται και επαλήθευση της πυκνότητας στο επίπεδο που το κύκλωμα έχει ολοκληρωθεί.

Αυτά τα βήματα είναι μόνο τα βασικά. Υπάρχουν κι άλλες πιο λεπτομερείς ροές που χρησιμοποιούνται ανάλογα με το εργαλείο που έχει χρησιμοποιηθεί και τις απαιτήσεις της εκάστοτε τεχνολογίας. Κάποια από τα εργαλεία που χρησιμοποιούνται στην back-end σχεδίαση είναι τα εξής: *Cadence*, *Synopsys*, *Magma*, *Mentor*.



Εικόνα 3: Ροή Φυσικής Σχεδίασης

2.2 Εργαλεία CAD και EDA

2.2.1 Computer-Aided Design (CAD)

Ως CAD ορίζεται η χρήση των ηλεκτρονικών υπολογιστικών συστημάτων για να βοηθήσουν στη δημιουργία, την τροποποίηση, την ανάλυση ή την βελτιστοποίηση ενός σχεδίου.

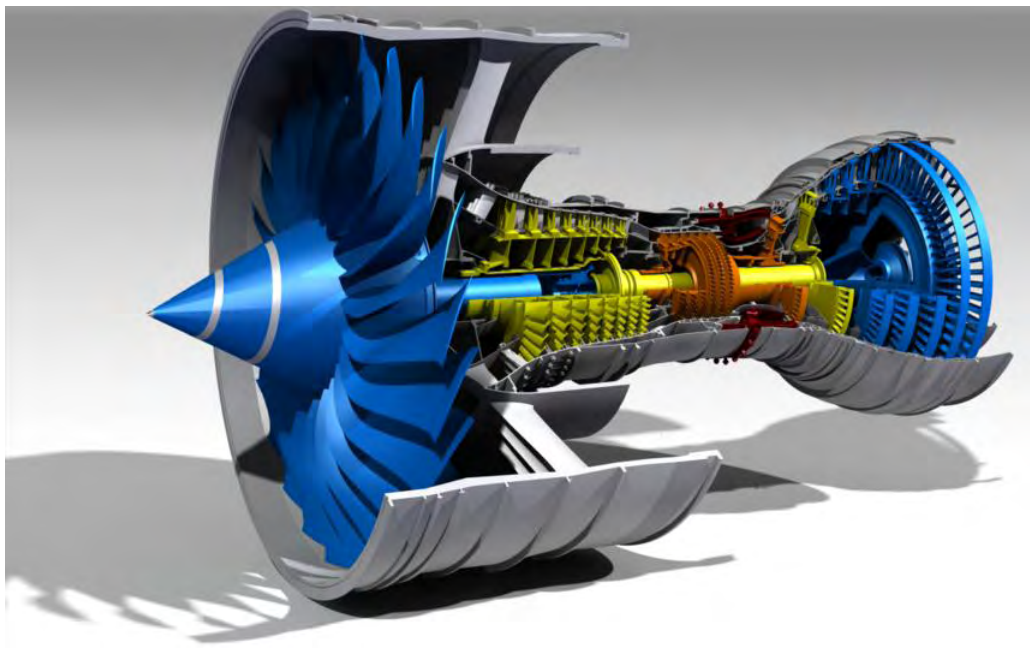
Το Computer-Aided program είναι μία τεχνολογία που σχεδιάζει και καταγράφει την διαδικασία του σχεδιασμού. Αυτό το εργαλείο μπορεί να διευκολύνει την διαδικασία κατασκευής, μεταφέροντας λεπτομερή διαγράμματα των εξαρτημάτων, τις διαδικασίες και τις ανοχές με συγκεκριμένες συμβάσεις, ανάλογα με το προϊόν που μας αφορά. Μπορεί ακόμα να χρησιμοποιηθεί για να παράγει είτε δισδιάστατα, είτε τρισδιάστατα σχέδια, τα οποία μπορούν να περιστρέφονται, ώστε να είναι εύκολη η παρατήρησή τους από οποιαδήποτε γωνία, ακόμα και από το εσωτερικό τους.

Για να εκτελεστούν οι λειτουργίες CAD το υπολογιστικό σύστημα θα πρέπει να συνδυάσει το απαραίτητο software και hardware. Το hardware αποτελείται από όλα τα υλικά στοιχεία του υπολογιστή, όπως τον επεξεργαστή, τη μητρική πλακέτα, την κάρτα γραφικών κα. Το software αποτελείται από τα πρόγραμμα που μπορούν να υλοποιήσουν γραφικά στο υπολογιστικό σύστημα και επίσης να εκτελέσουν πολυάριθμες λειτουργίες στον υπολογιστή. Το λογισμικό CAD μπορεί να υλοποιήσει λειτουργίες όπως η ανάλυση καταπόνησης στην οποία υποβάλλονται τα εξαρτήματα, δυναμική απόκριση των μηχανισμών, υπολογισμοί μεταφοράς θερμότητας κλπ. Δεν εκτελούν όλα τα λογισμικά CAD τις ίδιες λειτουργίες. Κάθε λογισμικό CAD είναι προγραμματισμένο να εκτελεί κάποια συγκεκριμένη λειτουργία. Ανάλογα με το αντικείμενό της, κάθε επιχείρηση θα επιλέξει το λογισμικό CAD θα είναι ικανό να εκτελέσει την συγκεκριμένη εφαρμογή που την ενδιαφέρει.

Πάραυτα, τα CAD software με τη μεγαλύτερη δημοτικότητα είναι αυτά που εκτελούν σχεδιαστικές λειτουργίες. Αυτά μπορούν να εκτελέσουν όλες τις λειτουργίες σχεδιασμού, όπως να κάνουν μεγάλο πλήθος υπολογισμών, να εκτελούν προσομοιώσεις των

ήδη σχεδιασμένων εξαρτημάτων κλπ. Ακόμα και τα ίδια τα σχέδια των εξαρτημάτων μπορούν να δημιουργηθούν χρησιμοποιώντας το λογισμικό CAD, γεγονός που βοηθάει στην αποφυγή του σχεδιασμού στο χέρι, μιας διαδικασίας μακράς και δυσκίνητης.

Οι εφαρμογές του CAD είναι πολυάριθμες. Η χρήση του στη σχεδίαση ηλεκτρονικών συστημάτων είναι ευρέως γνωστή ως Electronic Design Automation (EDA).



Εικόνα 4! 3D σχεδίαση στο CAD

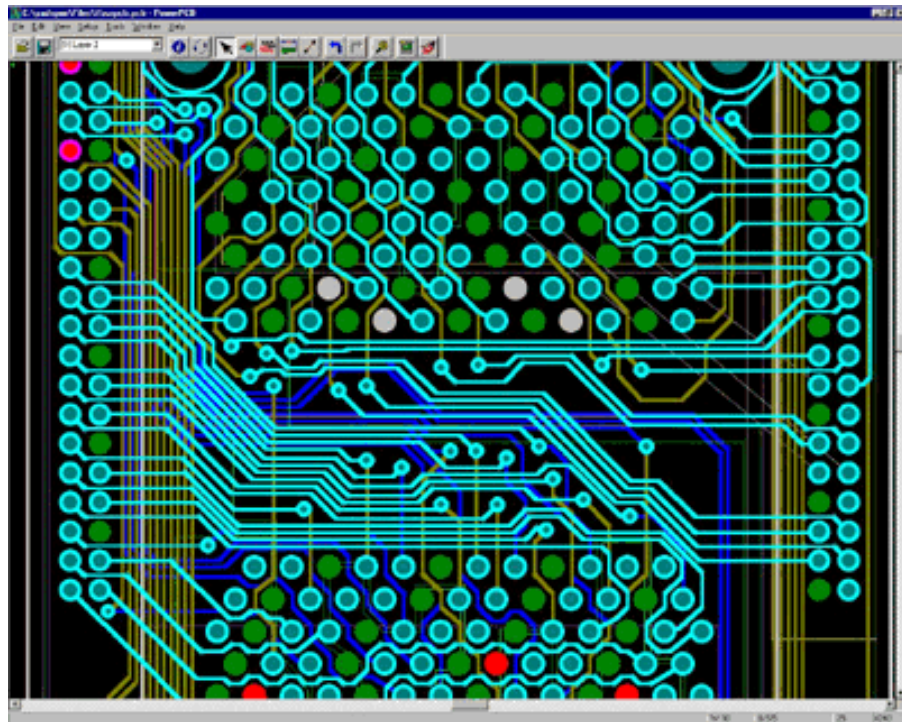
2.2.2 Electronic Design Automation (EDA)

Το Electronic Design Automation (EDA), που συχνά συναντάται και ως Electronic Computer-Aided Design (ECAD), είναι ένα σύνολο εργαλείων λογισμικού που σκοπό έχουν τη σχεδίαση ολοκληρωμένων κυκλωμάτων, πλακετών και άλλων ηλεκτρονικών συστημάτων. Τα λογισμικά του EDA συνεργάζονται σε μία ροή σχεδίασης, η οποία χρησιμοποιείται απ τους σχεδιαστές για τη σχεδίαση, αλλά και για την ανάλυση ολόκληρου του ολοκληρωμένου κυκλώματος. Δεδομένου ότι τα σύγχρονα ολοκληρωμένα κυκλώματα αποτελούνται από δισεκατομμύρια εξαρτημάτων, τα εργαλεία EDA είναι απαραίτητα για το σχεδιασμό τους.

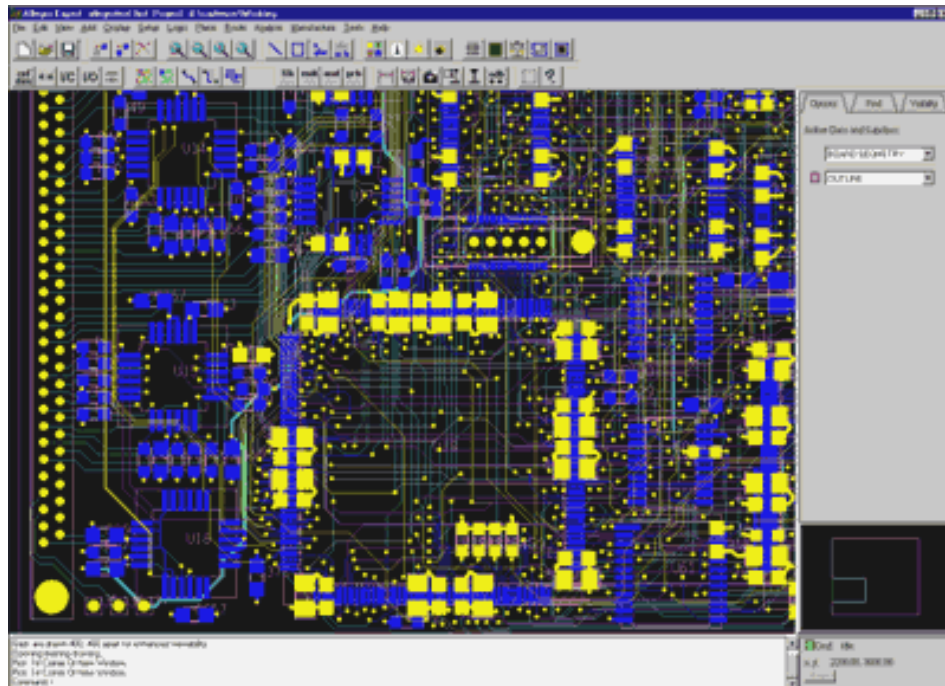
Το EDA χρησιμοποιείται κατά κόρον, διότι οι σύγχρονες ψηφιακές ροές είναι εξαιρετικά πολύπλοκες. Το front-end παράγει περιγραφές τυποποιημένων σχεδίων που μεταγλωττίζονται σε κελιά που τεχνολογικά είναι ανεξάρτητα. Τα κελιά αυτά υλοποιούν λογική ή ηλεκτρονικές διαδικασίες χρησιμοποιώντας συγκεκριμένη τεχνολογία. Οι κατασκευαστές, γενικά, παρέχουν βιβλιοθήκες των εξαρτημάτων, για τις διαδικασίες παραγωγής, με μοντέλα προσομοίωσης που ταιριάζουν με τα τυποποιημένα εργαλεία προσομοίωσης.

Τα EDA εργαλεία γίνονται όλο και πιο σημαντικά με την συνεχή κλιμάκωση των συσκευών των ημιαγωγών και την αυξανόμενη πολυπλοκότητα της χρήσης τους σε κυκλώματα και συστήματα. Οι απαιτήσεις για χαμηλή κατανάλωση ισχύος, μεγαλύτερη αξιοπιστία και πιο ευέλικτα ηλεκτρονικά συστήματα δημιουργούν νέες προκλήσεις τόσο για το σχεδιασμό, όσο και για τον αυτόματο σχεδιασμό τέτοιων συστημάτων.

Οι τομείς στους οποίους εστιάζει το λογισμικό EDA είναι η σχεδίαση, η προσομοίωση, η ανάλυση και επαλήθευση, η προετοιμασία κατασκευής και η ασφάλεια λειτουργίας.



Εικόνα 5: Στιγμιότυπο από το λογισμικό EDA Pads PowerPCB



Εικόνα 6: Στιγμιότυπο από το λογισμικό EDA Cadence Allegro Expert

3. Placement

3.1 Placement εντός της EDA ροής σχεδίασης

Το placement είναι ένα βασικό βήμα στη ροή ψηφιακής σχεδίασης, αφού αποδίδει την ακριβή τοποθεσία για τα διάφορα κυκλωματικά στοιχεία, εντός της περιοχής του πυρήνα του ολοκληρωμένου κυκλώματος. Μία όχι και τόσο καλή τοποθέτηση των στοιχείων δεν επηρεάζει μόνο την απόδοση του chip, αλλά μπορεί να το καταστήσει και μη υλοποιήσιμο κατασκευαστικά, λόγω της παραγωγής μεγάλου μήκους καλωδίου, που ξεπερνά τις διαθέσιμες πηγές δρομολόγησης. Συνεπώς, ένας placer θα πρέπει να εκτελεί την ανάθεση, ενώ ταυτόχρονα βελτιστοποιεί έναν αριθμό στόχων, έτσι ώστε να εξασφαλιστεί ότι το κύκλωμα θα ικανοποιεί τις απαιτήσεις απόδοσης που έχουν οριστεί. Τυπικά, τα αντικείμενα ενδιαφέροντος του placement συνοψίζονται στο συνολικό μήκος καλωδίου, τον χρονισμό, τη συμφόρηση και την ισχύ.

Ουσιαστικά, η δουλειά που έχει ανατεθεί στον placer είναι να συνδυάζει ένα δεδομένο netlist με την κατάλληλη βιβλιοθήκη της τεχνολογίας που απαιτείται και να παράγει μία έγκυρη διάταξη placement. Αυτή η διάταξη βελτιστοποιείται σύμφωνα με τους στόχους και είναι έτοιμη για αναθεώρηση μεγέθους και buffering. Το βήμα που μόλις περιγράψαμε είναι απαραίτητο για την ικανοποίηση του χρονισμού και την ακεραιότητα του σήματος. Ο συγχρονισμός (clock tree synthesis) και η δρομολόγηση ακολουθούν το placement, συμπληρώνοντας έτσι τη διαδικασία της φυσικής σχεδίασης. Πολλές φορές κάποια τμήματα ή και ολόκληρη η ροή της φυσικής σχεδίασης επαναλαμβάνονται αρκετές φορές, μέχρι να επιτευχθεί το τελικό αποτέλεσμα του σχεδίου.

Το netlist ενός κυκλώματος συντίθεται από τα στοιχεία του κυκλώματος και από έναν αριθμό net, όπου ένα net συνδέει δύο ή περισσότερα στοιχεία μεταξύ τους. Με τον όρο net εννοούμε την απαραίτητη ηλεκτρική διασύνδεση ανάμεσα στα στοιχεία. Όσον αφορά τα ολοκληρωμένα κυκλώματα ειδικών εφαρμογών ή ASICs,

η περιοχή διαμόρφωσης του πυρήνα των chip αποτελείται από κάποιες γραμμές, με σταθερό και ίσο ύψος μεταξύ τους, με ή χωρίς κενό ανάμεσά τους. Κάθε γραμμή μπορεί να καλυφθεί από διάφορα στοιχεία, που μπορεί να έχουν αλλά και να μην έχουν κενά μεταξύ τους. Τα στοιχεία αυτά μπορεί να είναι είτε τυποποιημένα cells, είτε macroblocks, είτε input/output pads. Τα τυποποιημένα cells έχουν σταθερό ύψος, ίσο με εκείνο που έχουν και οι γραμμές, αλλά τα πλάτη τους ποικίλουν και αποτελούνται από έναν ακέραιο αριθμό θέσεων. Απ' την άλλη μεριά, τα macroblocks συνήθως είναι μεγαλύτερα απ' τα cells και τα ύψη τους διαφέρουν μεταξύ τους και συχνά καταλαμβάνουν πάνω από μία σειρά. Πολλές φορές συμβαίνει τα blocks να βρίσκονται σε προκαθορισμένες θέσεις (σταθερά block) και ο placer να μην έχει δικαίωμα να παρέμβει, τότε περιορίζεται μόνο στο placement των cells. Εάν όμως τα blocks δεν έχουν προκαθορισμένες θέσεις, τότε ο placer θα πρέπει να προβεί σε mixed-mode placement και να ορίσει θέσεις σε blocks, αλλά και σε κελιά.

3.2 Στόχοι του Placement

Στόχος του placement, συνήθως, είναι η βελτιστοποίηση υπό κάποιους περιορισμούς. Ο περιορισμός έγκειται στην εξάλειψη των επικαλύψεων των στοιχείων του ολοκληρωμένου κυκλώματος μεταξύ τους. Όμως, ο στόχος του placement μπορεί να είναι πολλαπλός και συνοψίζεται ως εξής:

- **Μήκος καλωδίου:** Η ελαχιστοποίηση του wirelength αποτελεί το κυριότερο μέλημα των περισσότερων σχεδιαστών, γιατί έτσι έχουμε ως αποτέλεσμα την μείωση του μεγέθους του ολοκληρωμένου κυκλώματος, άρα και του κόστους κατασκευής του, καθώς επίσης ελαχιστοποιείται η ισχύς, αλλά και η καθυστέρηση, τα οποία είναι ανάλογα του μήκους καλωδίου. Συνεπώς, ελαχιστοποιώντας το wirelength επιτυγχάνεται

βελτιστοποίηση της απόδοσης του ολοκληρωμένου κυκλώματος.

- **Χρονισμός:** Σε ένα ολοκληρωμένο κύκλωμα ο κύκλος ρολογιού προσδιορίζεται από την καθυστέρηση του μακρύτερο μονοπατιού του, γνωστό και ως *critical path*. Αν λάβουμε υπόψη τις δοθείσες προδιαγραφές απόδοσης, δεν θα πρέπει να υφίσταται μονοπάτι με καθυστέρηση που να υπερβαίνει την μέγιστη προκαθορισμένη.
- **Συμφόρηση:** Αν και απαραίτητη, η ελαχιστοποίηση του *wirelength* είναι, επίσης, απαραίτητο να μην υπάρχει συμφόρηση καλωδίων, σε μία μικρή περιοχή του *chip*, γιατί αυτό μπορεί να οδηγήσει σε πολλές παρακάμψεις δρομολόγησης ή και να καταστήσει αδύνατη την πραγματοποίηση όλων των δρομολογήσεων.
- **Ισχύς:** Η ελαχιστοποίηση της κατανάλωσης περιλαμβάνει την διανομή των τοποθεσιών των κελιών, έτσι ώστε να υπάρξει μείωση στη συνολική κατανάλωση ισχύος, να μειωθούν τα *hot spots* και να εξομαλυνθούν οι μεταβολές της θερμοκρασίας.
- Ένας δευτερεύων στόχος είναι και η ελαχιστοποίηση του χρόνου τοποθέτησης.

Το ιδανικό θα ήταν να ελαχιστοποιηθεί καθένας απ' τους παραπάνω παράγοντες, αυτό όμως δεν είναι εφικτό καθώς κάποιοι είναι αντιστρόφως ανάλογοι μεταξύ τους και όταν ο ένας μειώνεται ο άλλος αυξάνει. Επομένως, η λύση του προβλήματος βρίσκεται κάπου στη μέση, κρατώντας τις ισορροπίες ανάμεσα στις μεταβλητές.

3.3 Placement Flow

- Global placement

Στόχος του global placement είναι η δημιουργία μίας πρόχειρης λύσης placement, που αν και παραβιάζει κάποιους περιορισμούς, όπως είναι τα overlaps, ταυτόχρονα διατηρεί μία σφαιρική άποψη του συνόλου του netlist. Για την επίτευξη του βέλτιστου αποτελέσματος η διαδικασία εκτελείται επαναληπτικά.

- Final placement

Το Final placement βελτιστοποιεί τις θέσεις των cell, που έχουν προκύψει από το global placement, στο χώρο. Η διαδικασία αυτή εκτελείται, επίσης, επαναληπτικά και αυτό το βήμα τα κελιά τοποθετούνται ώστε να μην υπάρξει καμία επικάλυψη.

- Area minimization

Στόχος του area minimization είναι η ελαχιστοποίηση της περιοχής που καταλαμβάνουν τα cells. Αποτελεί, κατά βάση, ένα δισδιάστατο πρόβλημα placement σε περιορισμένο χώρο και ανήκει στην κατηγορία των NP-hard.

- Legalization

Σε περίπτωση επικαλύψεων θα πρέπει εκ νέου να εφαρμοστούν τεχνικές νομιμοποίησης των θέσεων, δηλαδή να γίνει legalization.

- Detailed placement

Σε αυτό το βήμα αποσκοπούμε στην περεταίρω βελτίωση, με επαναληπτικό τρόπο, κάνοντας αναδιάταξη μίας μικρής ομάδας των ενοτήτων σε μια local area, διατηρώντας παράλληλα όλες τις περιοχές σταθερές.

3.4 Μετρήσεις Καλωδίων

Αφού εφαρμοστεί η διαδικασία του placement, τα cells πλέον έχουν καθορισμένες συντεταγμένες που τα καθορίζουν στο εσωτερικό του πυρήνα. Αυτό σημαίνει ότι σχεδόν το συνολικό μήκος καλωδίου του κυκλώματος μπορεί να υπολογισθεί, αθροίζοντας τα μήκη όλων των καλωδίων. Αυτό όμως δεν μπορεί να γίνει χωρίς την βοήθεια κάποιων μετρητών, οι οποίοι υπολογίζουν και βελτιστοποιούν τα wirelength, λόγω του ότι οι αποστάσεις μεταξύ των cells δεν είναι συγκεκριμένες πριν την δρομολόγηση. Στους μετρητές απόστασης τα κελιά ορίζονται ως σημεία με συντεταγμένες είτε το κέντρο τους είτε το κατώτερο αριστερό άκρο τους.

Κάποια μοντέλα μετρητών είναι τα παρακάτω:

- **Steiner tree:** αποτελείται αποκλειστικά από οριζόντιες και κάθετες ακμές οι οποίες εκτείνονται προς εκείνες τις κατευθύνσεις που απαιτείται για να συμπεριληφθούν όλα τα εμπλεκόμενα, στο netlist, στοιχεία.
- **Minimum spanning tree:** αποτελεί υπογράφημα/δένδρο ενός μη κατευθυνόμενου γραφήματος και συνδέει όλες τις κορυφές.
- **Μοντέλο κλίκας:** Το μοντέλο αυτό βασίζεται στην κατάλληλη μετατροπή ενός net σε υπογράφημα κλίκας.
- **Star:** είναι παρόμοιο με το μοντέλο κλίκας, όπου οι υπερακμές του γραφήματος μετατρέπονται σε υπογράφους άστρου.
- **Bounding Box:** Η μέθοδος αυτή είναι η πιο απλή όλων, καθώς ορίζει ότι το συνολικό netlength ενός ολοκληρωμένου κυκλώματος είναι το άθροισμα της

ημικυκλίου του πλαισίου οριοθέτησης που περιβάλλει κάθε net.

$$BB_{(n)} = \max_{p \in n} (Ax(p)) - \min_{p \in n} (Ax(p)) + \max_{p \in n} (Ay(p)) - \min_{p \in n} (Ay(p))$$

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

4. Detailed Placement και Αλγόριθμοι

Μόλις όλα τα τυποποιημένα κελιά του ολοκληρωμένου κυκλώματος τοποθετηθούν globally, ένας detailed placement algorithm εκτελείται για να τελειοποιήσει την χωροθέτησή τους, με βάση τη συμφόρηση, το timing και/ή τις απαιτήσεις ισχύος.

Η βελτίωση της συμφόρησης ή το placement ανάλογα τη συμφόρηση είναι πιο επικερδής σε ASIC σχεδιάσεις με πολύ υψηλή πυκνότητα και ο στόχος του detailed placer είναι να απομακρύνει τα τυποποιημένα cell μεταξύ τους, έτσι ώστε περισσότερα τμήματα routing να δημιουργηθούν ανάμεσά τους. Η ποιότητα του congestion placement σχετίζεται άμεσα με το πόσο καλά ο global Placer έχει χωρίσει το σχέδιο και μπορεί να έχει άσχημο αντίκτυπο στο μέγεθος και την απόδοση της συσκευής. Για μικρότερο μέγεθος συσκευής μπορεί να χρησιμοποιηθούν περισσότερα routing layers. Κατά τον προσδιορισμό του συνολικού πληθυσμού των routing layers που πρέπει να χρησιμοποιηθούν, είναι επιτακτική η εξέταση του trade-off ανάμεσα στην αύξηση του μεγέθους της συσκευής και της χρήσης επιπλέον routing layers, γιατί σε μερικές περιπτώσεις, μπορεί να είναι πιο οικονομικό να αυξηθεί το μέγεθος της συσκευής απ' το να προστεθούν εξτρά επίπεδα δρομολόγησης.

Οι αλγόριθμοι placement που βασίζονται στο timing, έχουν ταξινομηθεί είτε ως net based είτε ως path based. Οι net based προσπαθούν να περιορίσουν την καθυστέρηση ενός path επιβάλλοντας μια upper bound καθυστέρηση ή αναθέτοντας ένα βάρος σε κάθε net. Οι path based εφαρμόζουν περιορισμούς για να καθυστερήσουν τα path των μικρών υποκυκλωμάτων. Το μειονέκτημα των path based αλγορίθμων είναι ότι είναι αδύνατο να απαριθμιστούν όλα τα path ενός σχεδίου. Η μεγάλη πρόκληση στο timing based placement είναι η βελτιστοποίηση ενός μεγάλου συνόλου καθυστερήσεων των path χωρίς την απαρίθμησή τους στο ASIC σχέδιο. Αυτή η βελτιστοποίηση επιτυγχάνεται με την παρεμβολή weighted connectivity based placement, με timing analysis, που σχολιάζει μεμονωμένες περιπτώσεις, nets και καθυστερήσεις με πληροφορίες σχετικές με το σχεδιασμό.

Για να καταλήξουμε σε αυτούς τους τύπου των περιορισμών διάφορες τεχνικές placement έχουν προταθεί και χρησιμοποιηθεί. Η πιο γνωστή μέθοδος detailed placement είναι η Simulated annealing. Αυτή η μέθοδος δεν είναι μόνο αποτελεσματική, αλλά μπορεί να διαχειριστεί και περίπλοκους σχεδιαστικούς περιορισμούς.

4.1 Simulated Annealing

Η μέθοδος simulated annealing είναι μια τεχνική placement που βασίζεται την προσομοίωση και χρησιμοποιείται ως αλγόριθμος επαναληπτικής βελτίωσης κατά τη διαδικασία του detailed placement. Ο στόχος της τεχνικής αυτής είναι να βρει ένα βέλτιστο ή σχεδόν βέλτιστο placement για κάθε τυποποιημένο cell που είναι ήδη τοποθετημένο. 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 μεταευρεστική 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.

```
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 7: Αλγόριθμος Simulated annealing

Δοσμένου ενός αρχείου placement, οι αλλαγές γίνονται είτε μετακινώντας ένα cell είτε κάνοντας swap δυο cells μεταξύ τους. Όλες οι κινήσεις οι οποίες οδηγούν σε μείωση του κόστους είναι αποδεκτές. Οι κινήσεις που οδηγούν σε αύξηση του κόστους είναι αποδεκτές μόνο όταν υπάρχει η πιθανότητα να μειωθεί το κόστος μετά από κάποιες επαναλήψεις. Η αναλογία με την πραγματική διαδικασία annealing ενισχύεται με τη χρήση μίας 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 έτσι ώστε οι κινήσεις που αυξάνουν το κόστος να έχουν μικρότερη πιθανότητα να γίνουν αποδέκτες. Εν τέλει η θερμοκρασία μειώνεται σε μία πολύ χαμηλή τιμή και έτσι οι μόνες κινήσεις που γίνονται αποδεκτές είναι αυτές που οδηγούν σε μειωμένο κόστος.

4.2 Branch and Bound

Το Branch and Bound (BB ή B&B ή BnB) είναι ένας σχεδιαστικός αλγόριθμος για διακριτά και συνδυαστικά προβλήματα βελτιστοποίησης, καθώς επίσης και μαθηματικής βελτιστοποίησης. Αυτός ο αλγόριθμος αποτελείται από μία συστηματικά αρίθμηση των υποψήφιων λύσεων, μέσω της αναζήτησης χώρου. Το σύνολο των υποψήφιων λύσεων θεωρείται ότι σχηματίζει ένα rooted tree με ρίζα το πλήρες σύνολο. Ο αλγόριθμος BB διερευνά τα κλαδιά αυτού του δέντρου, που αντιπροσωπεύουν τα υποσύνολα του συνόλου των λύσεων. Πριν φτάσουμε στην απαρίθμηση των υποψήφιων λύσεων ενός κλαδιού, το κλαδί ελέγχεται έναντι των ανώτερων και κατώτερων εκτιμώμενων ορίων στη βέλτιστη λύση και απορρίπτεται εάν δεν μπορεί να παράγει μία καλύτερη λύση από τη βέλτιστη που έχει βρεθεί μέχρι στιγμής.

Ο αλγόριθμος εξαρτάται από την αποτελεσματική εκτίμηση του κατώτερου και του ανώτερου ορίου μίας περιοχής του χώρου

αναζήτησης και προσεγγίζει την εξαντλητική απαρίθμηση καθώς το μέγεθος της περιοχής τείνει στο μηδέν.

Ο στόχος ενός Branch and Bound αλγορίθμου είναι να βρεθεί μία τιμή x που μεγιστοποιεί ή ελαχιστοποιεί την τιμή μίας συνάρτησης πραγματικών τιμών $f(x)$, που ονομάζεται objective function, μεταξύ μερικών συνόλων S των αποδεκτών ή υποψήφιων λύσεων. Το σύνολο S ονομάζεται χώρος αναζήτησης. Στη συνέχεια, υποθέτουμε ότι η ελαχιστοποίηση του $f(x)$ είναι επιθυμητή. Αυτή η υπόθεση έρχεται χωρίς απώλεια της γενικότητας, αφού μπορούμε να βρούμε τη μέγιστη τιμή του $f(x)$ βρίσκοντας το ελάχιστο $g(x) = -f(x)$.

Ένας αλγόριθμος BB λειτουργεί σύμφωνα με δύο αρχές:

- Χωρίζει αναδρομικά τον χώρο αναζήτησης σε μικρότερους χώρους, ελαχιστοποιώντας το $f(x)$ σε αυτούς τους μικρότερους χώρους. Ο διαχωρισμός αυτός ονομάζεται *branching*.
- Η διακλάδωση από μόνης της θα ισοδυναμούσε με την καταμέτρηση των υποψήφιων λύσεων και τη δοκιμή όλων. Για να βελτιωθεί η απόδοση της *brute-force* αναζήτησης, ένας BB αλγόριθμος παρακολουθεί τα ελάχιστα όρια αυτού που ψάχνει και τα χρησιμοποιεί για να μειώσει τον χώρο αναζήτησης, εξαλείφοντας τις υποψήφιες λύσεις που μπορεί να αποδείξει ότι δεν εμπεριέχουν κάποια βέλτιστη λύση.

Η μετατροπή αυτών των αρχών σε ένα συγκεκριμένο αλγόριθμο για ένα συγκεκριμένο πρόβλημα βελτιστοποίησης απαιτεί κάποιο είδος δομής δεδομένων που αντιπροσωπεύει σύνολα υποψήφιων λύσεων. Μία τέτοια αναπαράσταση ονομάζεται παράδειγμα του προβλήματος. Δηλώνουμε το σύνολο των υποψήφιων λύσεων ενός παραδείγματος I ως S_I . Η αναπαράσταση του παραδείγματος πρέπει να περιλαμβάνει τα εξής:

- *Branch(I)*: παράγει δύο ή περισσότερα στιγμιότυπα, που το καθένα αντιπροσωπεύει ένα υποσύνολο S_i . Ουσιαστικά, τα υποσύνολα

είναι ασύνδετα, για να αποτρέψουν τον αλγόριθμο να διαπεράσει την ίδια υποψήφια λύση δεύτερη φορά, αλλά αυτό δεν είναι προαπαιτούμενο. Η μόνη απαίτηση για έναν σωστό BnB αλγόριθμο είναι ότι η βέλτιστη λύση του S_i περιλαμβάνεται σε τουλάχιστον ένα απ' τα υποσύνολα.

- $Bound(I)$: υπολογίζει ένα κατώτερο όριο στην τιμή κάθε υποψήφιας λύσης στον χώρο που αναπαριστά το I , δηλαδή

$$Bound(I) \leq f(x), \text{ για κάθε } x \in S_i$$

- $Solution(I)$: προσδιορίζει εάν το I αντιπροσωπεύει μία μοναδική υποψήφια λύση.

Χρησιμοποιώντας αυτές τις λειτουργίες, ένας BB αλγόριθμος εκτελεί μία top-down αναδρομική αναζήτηση εντός των στιγμιότυπων του δέντρου που δημιουργούνται απ' την $branch(I)$. Κατά την επίσκεψη σε ένα στιγμιότυπο I , ελέγχει εάν είναι το $bound(I)$ μεγαλύτερο από το ανώτερο όριο κάποιας άλλης περίπτωσης που έχει ήδη εξεταστεί, αν είναι το I μπορεί ασφαλώς να απορριφθεί από την αναζήτηση και η αναδρομή να σταματήσει. Αυτό το βήμα συνήθως υλοποιείται διατηρώντας μία global μεταβλητή που καταγράφει το ελάχιστο άνω όριο όλων των περιπτώσεων που έχουν μέχρι στιγμής εξετασθεί.

1. Using a [heuristic](#), find a solution x_h to the optimization problem. Store its value, $B = f(x_h)$. (If no heuristic is available, set B to infinity.) B will denote the best solution found so far, and will be used as an upper bound on candidate solutions.
2. Initialize a queue to hold a partial solution with none of the variables of the problem assigned.
3. Loop until the queue is empty:
 1. Take a node N off the queue.
 2. If N represents a single candidate solution x and $f(x) < B$, then x is the best solution so far. Record it and set $B \leftarrow f(x)$.
 3. Else, $branch$ on N to produce new nodes N_i . For each of these:
 1. If $g(N_i) > B$, do nothing; since the lower bound on this node is greater than the upper bound of the problem, it will never lead to the optimal solution, and can be discarded.
 2. Else, store N_i on the queue.

4.3 The Domino algorithm

Ένας πολύ υψηλής ποιότητας, καθώς, και αποτελεσματικός detailed placer είναι ο αλγόριθμος Domino. Ο Domino χρησιμοποιεί μία προσέγγιση sliding window για να βελτιώσει διαδοχικά μία μικρή περιοχή. Για κάθε περιοχή, το πρόβλημα της ανάθεσης των κελιών σε νέες τοποθεσίες διατυπώνεται ως πρόβλημα μεταφοράς. Για να υπολογιστούν τα διάφορα πλάτη των κελιών, κάθε κελί l με πλάτος w_l διαιρείται σε w_l υποκελιά, μοναδιαίου πλάτους. Τότε το πρόβλημα είναι η ταυτόχρονη μεταφορά των υποκελιών σε θέσεις μοναδιαίου πλάτους, με τρόπο που δεν θα υπάρχουν overlaps και που θα ελαχιστοποιεί μία συνάρτηση κόστους που προσεγγίζει το μήκος καλωδίου. Αυτό το πρόβλημα μπορεί να μετατραπεί σε πρόβλημα min-cost – max-flow σε ένα δίκτυο

Αυτό το δίκτυο αποτελείται από μία πηγή S , ένα σύνολο cell-κόμβων i , ένα σύνολο κόμβων θέσης k και έναν κόμβο προορισμού D . Η χωρητικότητα της ακμής μεταξύ της πηγής S και των cell-κόμβων i είναι w_i . Επειδή κάθε τοποθεσία μπορεί να κρατήσει το πολύ ένα υποκελί, όλες οι χωρητικότητες των ακμών, που οδηγούν από κόμβους τοποθεσίας στον κόμβο προορισμού D , είναι ίσες με ένα. Το κόστος ανάθεσης ενός υποκελιού ενός κελιού i σε μία τοποθεσία k είναι C_{ik} .

Με την επίλυση του προβλήματος ροής δικτύου, τα subsells αντιστοιχίζονται σε τοποθεσίες. Έτσι, όλα τα υποκελιά ενός cell, καθώς συνδέονται με το ίδιο κόστος μεταφοράς και έλκονται προς την φθηνότερη τοποθεσία από τον αλγόριθμο μεταφοράς, τείνουν να βρίσκονται δίπλα-δίπλα. Κάθε κελί τοποθετείται στη γραμμή που έχει τα περισσότερα υποκελιά του. Η συντεταγμένη x του cell καθορίζεται από το κέντρο βάρους των subcells. Εν τέλη τα cells κάθε γραμμής της περιοχής πακετάρονται σύμφωνα με τις συντεταγμένες x που έχουν, για την αποφυγή overlap και ανεκμετάλλευτου χώρου.

Το κόστος c_{ik} για την ανάθεση ενός υποκελιού ενός κελιού i σε μία τοποθεσία k είναι το συνολικό HPWL (half perimeter wirelength) όλων που είναι συνδεδεμένα με το cell i . Κατά την

αξιολόγηση του κόστους c_{ik} , το cell i θεωρείται ότι είναι στην τοποθεσία k . Ωστόσο, οι τοποθεσίες των άλλων κελιών στην περιοχή εξακολουθούν να είναι άγνωστες. Ως εκ τούτου, το HPWL των nets εκτιμάται σύμφωνα με το ακόλουθο μοντέλο δικτύου. Για να υπολογιστεί το HPWL ενός net e συνδεδεμένου σε ένα cell i , έστω ότι e_i είναι το υποσύνολο των συνδεδεμένων κελιών στο net e εντός της περιοχής. Ας εξετάσουμε τις τρεις παρακάτω περιπτώσεις:

- $|e_i| = 1$
 Σε αυτή την περίπτωση, το μόνο κελί, μέσα στην περιοχή, που συνδέεται το net e είναι το κελί i . Επομένως, το HPWL του net e μπορεί να υπολογιστεί με ακρίβεια.
- $1 < |e_i| < |e|$
 Σε αυτή την περίπτωση, το net e επανασυνδέεται με κάποια cells διαφορετικά του cell i μέσα και εκτός της περιοχής. Οι άγνωστες θέσεις των κελιών στο $e_i - \{i\}$ προσδιορίζονται απ' τις συντεταγμένες στην τρέχουσα χωροθέτηση. Στη συνέχεια το HPWL του net e μπορεί να υπολογιστεί.
- $|e_i| = |e|$
 Στην περίπτωση αυτή, το net e συνδέεται με cells εντός της περιοχής μόνο. Επίσης, οι θέσεις των cells στο $e_i - \{i\}$ προσδιορίζονται απ' τις συντεταγμένες τους στην τρέχουσα χωροθέτηση. Εκτός αυτού, ένα εικονικό κελί εισάγεται στο κέντρο βάρους των κελιών στο e_i σε σχέση με την τρέχουσα τοποθέτηση. Το HPWL όλων των cells του net e μαζί με το εικονικό cell, χρησιμοποιούνται για την εκτίμηση του HPWL του net e .

Ένα πλεονέκτημα του Domino σε σχέση με τους branch and Bound αλγορίθμους είναι ότι το πρόβλημα ροής δικτύου έχει πολύ

μικρότερη υπολογιστική πολυπλοκότητα και επομένως μπορούν να χρησιμοποιηθούν πολύ μεγαλύτερα παράθυρα. Ένα μεγαλύτερο παράθυρο επιτρέπει σε περισσότερα cells να τοποθετηθούν ταυτόχρονα και ενδεχομένως βελτιώνει το μήκος καλωδίου. Ωστόσο, αυξάνει το χρόνο εκτέλεσης και οδηγεί σε μία λιγότερο ακριβή εκτίμηση του HPWL στη συνάρτηση κόστους. Στην πράξη, ένα παράθυρο περίπου 20 έως 30 κελιών ανά περιοχή αποδίδει μία καλή συμφωνία μεταξύ του wirelength και του χρόνου εκτέλεσης.

```
Initial_placement;  
While (improvement)  
    Generate_improved_placement;  
End  
Adjust_row_lengths;  
Swap_cells,
```

Αλγόριθμος Domino

4.4 The FastDP algorithm

Ο αλγόριθμος FastDP είναι άπληστος ευρεστικός, που μπορεί να δημιουργήσει καλύτερες λύσεις απ' τον Domino και είναι μία τάξη μεγέθους πιο γρήγορος. Ο FastDP αποτελείται από τέσσερις βασικές τεχνικές: global swap, vertical swap, local re-ordering και single segment clustering. Η ροή του αλγορίθμου φαίνεται παρακάτω.

The FastDP Detailed Placement Algorithm

- Perform single-segment clustering
- Repeat
- Perform global swap
 - Perform vertical swap
 - Perform local reordering
- Until no significant improvement in wirelength
- Repeat
- Perform single-segment clustering
- Until no significant improvement in wirelength

Global swap είναι η τεχνική που δίνει τη μεγαλύτερη μείωση wirelength. Εξετάζει όλα τα cells ένα προς ένα. Για κάθε κελί i , ο στόχος είναι να το μετακινήσουμε στην optimal region του. Για μία δεδομένη χωροθέτηση, ως optimal region ενός cell ορίζεται η περιοχή στην οποία αν τοποθετηθεί το cell το wirelength θα είναι βέλτιστο. Αυτό μπορεί αν καθοριστεί με βάση το median. Έστω ότι E_i είναι το σύνολο των συνδεδεμένων nets στο cell i . Για κάθε net $e \in E_i$, υπολογίζεται το bounding box εξαίρουμένου του cell i . Έστω x_L^e και x_R^e είναι οι x συντεταγμένες των αριστερών και δεξιών ορίων και y_L^e και y_U^e οι y συντεταγμένες των κάτω και πάνω ορίων, αντίστοιχα. Στη συνέχεια, η βέλτιστη x συντεταγμένη για το cell i δίνεται από το median του συνόλου των boundary συντεταγμένων $x \{ x_L^e : e \in E_i \} \cup \{ x_R^e : e \in E_i \}$. Γενικά, η βέλτιστη συντεταγμένη για το cell i είναι μία περιοχή αντί για ένα σημείο, καθώς ο αριθμός των στοιχείων του συνόλου $\{ x_L^e : e \in E_i \} \cup \{ x_R^e : e \in E_i \}$ είναι άρτιος.

Παρομοίως, η βέλτιστη γ συντεταγμένη για το cell i δίνεται από το median του συνόλου των boundary γ συντεταγμένων $\{ \gamma_L^e : e \in E_i \} \cup \{ \gamma_U^e : e \in E_i \}$.

Αν και είναι επιθυμητό, στα όρια του wirelength, να μετακινήσουμε το cell i στην optimal region του, αυτή μπορεί να μην έχει αρκετό χώρο για να υποδεχτεί το κελί. Οπότε, στο global swar, το cell i θα αλλάξει θέση με ένα άλλο κελί η με ένα space εντός της optimal region. Μία συνάρτηση κέρδους υπολογίζεται για κάθε cell και κάθε space της optimal region. Αν υπάρχει κάποιο cell ή κάποιο space με θετικό κέρδος, εκείνο με το μεγαλύτερο κέρδος θα κάνει swar με το cell i . Η συνάρτηση benefit αποτελείται από δύο μέρη. Το πρώτο μέρος είναι η βελτίωση του συνολικού wirelength αν εκτελεστεί το swar. Το δεύτερο μέρος είναι ένα penalty. Κάνοντας swar κελιά διαφορετικών μεγεθών ή κελί μεγάλου μεγέθους με μικρό space μπορεί να δημιουργηθεί overlap. Το overlap επιλύεται μετακινώντας τα διπλανά κελιά. Το penalty είναι μια συνάρτηση της ελάχιστης μετακίνησης που είναι απαραίτητη για την επίλυση του overlap.

Το vertical swar είναι παρόμοιο με το global swar. Η διαφορά έγκειται στο ότι το cell i πρόκειται να κάνει swar με τα διπλανά κελιά που είναι μία γραμμή πάνω ή κάτω απ' την τρέχουσα θέση του. Μερικές φορές ένα cell αποτυγχάνει να κάνει swar μέσω του global swar επειδή δεν υπάρχει ούτε κελί ούτε ελεύθερος χώρος, στην optimal region, που να γίνουν κάποιο θετικό benefit. Το vertical swar επιτρέπει στο cell να κινηθεί προς την optimal region, ώστε να μειωθεί το κάθετο μήκος καλωδίου. Επιπρόσθετα, το vertical swar είναι πολύ γρηγορότερο από το global swar, γιατί ο αριθμός των υποψήφια για swar είναι πολύ μικρότερος.

Το local re-ordering παίρνει κάθε πιθανό group n διαδοχικών cells σε μία γραμμή. Για κάθε group, δοκιμάζονται όλες οι πιθανές αλλαγές των cells δεξιά και αριστερά και επιλέγεται το swar που δίνει το βέλτιστο wirelength. Το local re-ordering είναι πολύ φθηνή διαδικασία για την ελαχιστοποίηση του οριζόντιου wirelength. Πρακτικά, το n ορίζεται ίσο με 3. Δεν είναι απαραίτητο να χρησιμοποιήσουμε μεγαλύτερο n , γιατί είναι πιο

αποτελεσματική η επιδιόρθωση μη τοπικών σφαλμάτων με global swap.

Το single-segment clustering είναι μία τεχνική ελχιστοποίησης του οριζόντιου wirelength μετατοπίζοντας τα cells ως segments χωρίς να διαταράσσεται η σειρά τους. Στον FastDP, ένα segment είναι το μέγιστο αδιάσπαστο τμήμα μίας πρότυπης γραμμής κελιών. Το single-segment clustering εξετάζει κάθε segment ένα προς ένα. Όταν εξετάζουμε ένα segment, οι θέσεις όλων των καλιών στα υπόλοιπα segments είναι σταθερές.

5. Σύγκριση Detailed Placement αλγορίθμων

Σε αυτό το κεφάλαιο περιγράφονται οι αλγόριθμοι που γράφτηκαν για αυτή τη διπλωματική εργασία, η διαδικασία για το testing αυτών, καθώς επίσης, και τα συμπεράσματα στα οποία καταλήξαμε μετά το πέρας της.

5.1 Detailed placer αλγόριθμοι προς σύγκριση

Οι αλγόριθμοι που περιγράφονται παρακάτω είναι βασισμένοι στις τεχνικές global swap, vertical swap και local re-ordering, που περιγράφονται πιο πάνω στον αλγόριθμο FastDP, και είναι παραλλαγές αυτού, αντικαθιστώντας το single-segment clustering με την legalize τεχνική tetris, που περιγράφεται πιο κάτω.

❖ *Tetris legalizer*

Η μέθοδος του Tetris legalization από τον Hill είναι εξαιρετικά απλή και εύκολη στην εφαρμογή. Πρώτα αναφέρουμε την προσέγγιση στο πλαίσιο της σχεδίασης των πρότυπων cells και στη συνέχεια δείχνουμε πως επεκτείνεται με σκοπό να χειριστεί ένα συνδυασμό από πρότυπα κελιά και macroblocks. Για κάθε κελί c_i , έχουμε μία 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.

Αυτή η μέθοδος πρώτα ταξινομεί τα κελιά c ανάλογα με τη θέση x και μετά εισάγει αυτά στην αριστερή πλευρά της γραμμής με greedy τρόπο, τέτοιο ώστε η μετατόπιση του κάθε cell να είναι ελάχιστη.

```

Algorithm 4 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

```

➤ **Αλγόριθμος 1**

- Tetris legalization

Repeat

- Simple global swap
- Vertical swap
- Local re-ordering

Until no significant improvement in wirelength

- Tetris legalization

➤ **Αλγόριθμος 2**

- Tetris legalization

Repeat

- Global swap
- Vertical swap
- Local re-ordering

Until no significant improvement in wirelength

- Tetris legalization

➤ **Αλγόριθμος 3**

- *Tetris legalization*

Repeat

- *Global swap*
- *Tetris legalization*
- *Vertical swap*
- *Local re-ordering*

Until no significant improvement in wirelength

- *Tetris legalization*

➤ **Αλγόριθμος 4**

- *Tetris legalization*

Repeat

- *Global swap*
- *Vertical swap*
- *Local re-ordering*
- *Tetris legalization*

Until no significant improvement in wirelength

5.2 Περιβάλλον testing

Για το testing των αλγορίθμων χρησιμοποιήθηκαν τα κυκλώματα IBM. Αυτά δόθηκαν ως input σε έναν Gordian placer και το output είναι οι θέσεις των cell στο χώρο μετά το detailed placement που κάναμε. Τα cells, σε πρώτη φάση, έχουν αρκετά μεγάλη επικάλυψη μεταξύ τους και δεν έχουν διανεμηθεί στο χώρο. Αυτό οδηγεί τον detailed placer να έχει μεγάλο βάρος για το τελικό placement των κελιών. Ο χώρος που χρησιμοποιείται είναι στο 110% του συνολικού εμβαδού που καταλαμβάνουν τα cells, επομένως υπάρχει και αρκετά μεγάλος περιορισμός στο χώρο.

Τα αποτελέσματα που θα παρουσιάσουμε σε αυτή τη διπλωματική είναι η ποσοστιαία μείωση του wirelength μετά τη διαδικασία του detailed placement. Ο χρόνος που χρειάστηκε στους αλγόριθμους για να δώσουν την τελική θέση στα cells είναι σχεδόν ίδιος και η πολυπλοκότητάς τους παρόμοια, οπότε δεν θα ασχοληθούμε περαιτέρω.

	<i>Legalizer</i>	<i>Algo1</i>	<i>Algo2</i>	<i>Algo3</i>	<i>Algo4</i>
<i>lbm01</i>	1.908	-27.77%	-34.80%	-23.11%	-27.30%
<i>lbm02</i>	3.184	-24.81%	-31.25%	-19.84%	-25.14%
<i>lbm03</i>	4.537	-31.14%	-33.92%	-23.12%	-31.35%
<i>lbm04</i>	5.923	-32.86%	-34.97%	-25.18%	-31.67%
<i>lbm05</i>	9.931	-16.85%	-17.51%	-17.81%	-23.58%
<i>lbm06</i>	5.584	-27.81%	-29.52%	-22.53%	-26.17%
<i>lbm07</i>	8.741	-26.93%	-29.44%	-22.14%	-25.73%
<i>lbm08</i>	8.954	-26.55%	-28.92%	-21.98%	-24.95%
<i>lbm09</i>	9.753	-36.72%	-39.21%	-29.52%	-33.15%
<i>lbm10</i>	17.827	-33.12%	-37.92%	-28.14%	-31.28%
<i>lbm11</i>	14.222	-35.48%	-39.82%	-30.52%	-33.14%
<i>lbm12</i>	22.753	-31.91%	-34.14%	-26.17%	-30.02%
<i>lbm13</i>	16.873	-35.84%	-39.82%	-32.78%	-33.48%
<i>lbm14</i>	32.247	-31.12%	-35.93%	-29.13%	-30.28%
<i>lbm15</i>	39.128	-34.51%	-38.72%	-33.20%	-33.57%
<i>lbm16</i>	44.015	-36.82%	-42.53%	-35.68%	-36.12%
<i>lbm17</i>	61.878	-32.12%	-37.73%	-34.28%	-35.18%
<i>lbm18</i>	41.461	-36.87%	-42.81%	-37.92%	-39.88%
		-31.06%	-34.92%	-27.39%	-30.66%

Πίνακας 1: Ποσοστιαία μείωση του wirelength

5.3 Συμπεράσματα και Μελλοντική Έρευνα

Μετά από την εκτέλεση των τεσσάρων αλγορίθμων που παρουσιάστηκαν πιο πάνω, για καθένα από τα κυκλώματα IBM, συμπεραίνουμε ότι οι αποκλίσεις του είναι μικρότερες της τάξης του 7.50%. Μεγαλύτερη βελτιστοποίηση στο wirelength παρουσιάζει ο Αλγόριθμος 2, που είναι και ο πιο όμοιος με τον FastDP. Χαμηλότερη αποδοτικότητα δείχνει να παρουσιάζει ο Αλγόριθμος 3, αποδεικνύοντας έτσι ότι η παρεμβολή legalization, ανάμεσα στα βήματα global swar και vertical swar, όχι μόνο είναι περιττή, αλλά φέρνει και αρνητικά αποτελέσματα. Ο αλγόριθμος 1 με τον αλγόριθμο 2 διαφέρουν όσον αφορά το global swar. Ουσιαστικά, ο Αλγόριθμος 2 ψάχνει εκτενέστερα την optimal region του κάθε cell, σε αντίθεση με τον Αλγόριθμο 1 που αν αποτύχει η πρώτη αναζήτηση στην optimal region, προχωράει παρακάτω. Αν και εκ πρώτης όψης, αυτή η διαφοροποίηση του Αλγορίθμου 2 απ' τον Αλγόριθμο 1 δείχνει να αυξάνει το χρόνο εκτέλεσης, τα αποτελέσματα δείχνουν ότι κάτι τέτοιο δεν ισχύει και πολλές φορές ο αλγόριθμος 2 ήταν και πιο γρήγορος.

Για να υπάρξουν πιο αξιόπιστα στοιχεία και πιο έγκυρες συγκρίσεις, σχετικά με αυτούς τους τέσσερις αλγορίθμους, θα πρέπει να εκτελεστούν και για μεγαλύτερα κυκλώματα. Επίσης, ενδιαφέρον παρουσιάζει και η εκτέλεση με διαφορετικούς legalizers.

6. 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

transactions 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

