

*“Χωροθέτηση Ολοκληρωμένων
Κυκλωμάτων με Παραμέτρους
Αξιοπιστίας”*

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ:

Δαδαλιάρης Αντώνιος

Βόλος, 2012

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Τμήμα Μηχανικών Η/Υ,

Τηλεπικοινωνιών & Δικτύων

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων και βασικό επιβλέποντα αυτής της διατριβής 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 τους συναδέλφους του γραφείου Ε5 στο οποίο εκπονήθηκε το μεγαλύτερο μέρος αυτής της διατριβής για την προγενέστερη δουλειά τους στα εργαλεία που χρησιμοποιήθηκαν και την μεταλαμπάδευση της γνώσης αυτής.

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

ΠΕΡΙΕΧΟΜΕΝΑ

Εισαγωγή.....	8
ΚΕΦΑΛΑΙΟ 1: ΣΧΕΔΙΑΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ	9
1.1 ΣΧΕΔΙΑΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ	9
1.2 ΡΟΗ ΣΧΕΔΙΑΣΗΣ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ	10
1.3 ΦΥΣΙΚΗ ΣΧΕΔΙΑΣΗ (PHYSICAL DESIGN).....	13
1.4 ΕΡΓΑΛΕΙΑ CAD (COMPUTER-AIDED DESIGN).....	19
1.5 ΕΡΓΑΛΕΙΑ EDA (ELECTRONIC DESIGN AUTOMATION)	19
1.5.1 ΙΣΤΟΡΙΑ ΤΟΥ EDA.....	20
1.5.2 ΤΟΜΕΙΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ EDA.....	21
1.6 ΣΤΟΧΟΣ ΤΗΣ ΔΙΑΤΡΙΒΗΣ.....	23
ΚΕΦΑΛΑΙΟ 2: PLACEMENT.....	24
2.1 ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ.....	24
2.2 GLOBAL PLACEMENT	27
2.2.1. GRAPH PARTITIONING.....	27
2.2.2 ANALYTIC-BASED PARTITIONING	30
2.2.3 FORCED-BASED METHODS	39
2.2.4 SIMULATED ANNEALING	40
2.2.5 CLUSTERING.....	40
2.3 FINAL PLACEMENT	41
2.3.1 SIMULATED ANNEALING	41
2.3.2 GREEDY APPROACHES	43

2.4. LEGALIZATION.....	46
ΚΕΦΑΛΑΙΟ 3: ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ	47
3.1. ΠΡΟΠΑΡΑΣΚΕΥΑΣΤΙΚΟ ΣΤΑΔΙΟ - ΔΟΜΕΣ ΑΠΟΘΗΚΕΥΣΗΣ ΔΕΔΟΜΕΝΩΝ	47
3.1.1. ΕΙΣΟΔΟΣ	48
3.1.2. ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΑ ΕΡΓΑΛΕΙΑ	51
3.1.3. ΕΞΟΔΟΣ.....	67
3.2. WIRELENGTH-DRIVEN PLACEMENT	69
3.3. IR-DROP BASED RE-PLACEMENT	76
ΚΕΦΑΛΑΙΟ 4: BENCHMARK CIRCUITS	80
ΚΕΦΑΛΑΙΟ 5: ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ & ΣΥΜΠΕΡΑΣΜΑΤΑ.....	84
5.1. Fixed Pin Positions, Random Initial Cell Placement, Tool-Calculated Area	84
5.2. Fixed Initial Cell Positions, Random Pin Positions, Tool-Calculated Area	87
5.3. Fixed Initial Cell Positions, Random Pin Positions, CAD Tool-Calculated Area.....	89
5.4. Fixed Pin Positions, Random Initial Cell Placement, CAD Tool-Calculated Area	92
5.5. IR-Drop Based Re-Placement	95
ΚΕΦΑΛΑΙΟ 6: ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	97

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

<i>Εικόνα 1.1: CPU Transistor Count – Moore’s Law</i>	10
<i>Εικόνα 1.2: VLSI Circuits Design Flow</i>	16
<i>Εικόνα 1.3: Design Flow & CAD Tools</i>	18
<i>Εικόνα 2.1: Layout Models</i>	25
<i>Εικόνα 2.2: Net Models</i>	30
<i>Εικόνα 2.3: Gordian Algorithm pseudocode</i>	32
<i>Εικόνα 2.4: Gordian Placement</i>	33
<i>Εικόνα 2.5: Simulated Annealing pseudocode</i>	42
<i>Εικόνα 2.6: Domino Placement</i>	44
<i>Εικόνα 2.7: Mongrel Algorithm Instance</i>	45
<i>Εικόνα 3.1: Design Compiler’s participation in a Typical Design Flow</i>	52
<i>Εικόνα 3.2: Synopsys Synthesis Flow</i>	55
<i>Εικόνα 3.3: Generic Synthesis Script</i>	56
<i>Εικόνα 3.4: Foundation Library Arithmetic Optimization</i>	59
<i>Εικόνα 3.5: Foundation Library Adder Implementation</i>	60
<i>Εικόνα 3.6: Analytical P&R Flow</i>	61
<i>Εικόνα 3.7: I/O, PWR/GND, Corner PAD Placement</i>	62

<i>Εικόνα 3.8: Floorplanning</i>	62
<i>Εικόνα 3.9: Amoeba Placement</i>	63
<i>Εικόνα 3.10: Power Planning</i>	64
<i>Εικόνα 3.11: Clock Tree Synthesis</i>	64
<i>Εικόνα 3.12: Power Analysis</i>	65
<i>Εικόνα 3.13: Power Routing</i>	66
<i>Εικόνα 3.14: Final Routing</i>	66
<i>Εικόνα 3.15: Random Initial Placement</i>	68
<i>Εικόνα 3.16: Final Placement</i>	68
<i>Εικόνα 3.17: Abacus legalization approach</i>	74
<i>Εικόνα 3.18: Wirelength-Driven Placement</i>	75
<i>Εικόνα 3.19: IC Supply Voltages by year</i>	76
<i>Εικόνα 3.20: IR-Drop Based Replacement</i>	79
<i>Εικόνα 4.1: ISCAS '89 benchmark circuits</i>	81
<i>Εικόνα 4.2: ITCC '99 benchmark circuits</i>	83
<i>Εικόνα 5.1: Fixed Pin Positions, Random Initial Cell Placement, Tool-Calculated Area, ISCAS results</i>	85
<i>Εικόνα 5.2: Fixed Pin Positions, Random Initial Cell Placement, Tool-Calculated Area, ITC results</i>	86

Εικόνα 5.3: Fixed Initial Cell Positions, Random Pin Positions, Tool-Calculated Area, ISCAS results... ..88

Εικόνα 5.4: Fixed Initial Cell Positions, Random Pin Positions, Tool-Calculated Area, ITC results... ..89

Εικόνα 5.5: Fixed Initial Cell Positions, Random Pin Positions, CAD Tool-Calculated Area, ISCAS results... ..90

Εικόνα 5.6: Fixed Initial Cell Positions, Random Pin Positions, CAD Tool-Calculated Area, ITC results... ..91

Εικόνα 5.7: Fixed Pin Positions, Random Initial Cell Placement, CAD Tool-Calculated Area, ISCAS results... ..92

Εικόνα 5.8: Fixed Pin Positions, Random Initial Cell Placement, CAD Tool-Calculated Area, ITC results... ..93

Εικόνα 5.9: IR-Drop Based Re-Placement, ISCAS & ITC results... ..95

Εισαγωγή

Σκοπός της παρούσας διατριβής είναι η παρουσίαση ενός νέου εργαλείου χωροθέτησης ολοκληρωμένων κυκλωμάτων. Το προκείμενο εργαλείο εντάσσεται στην διαδικασία της φυσικής σχεδίασης του κυκλώματος, εάν αναλογιστούμε μια απλουστευτική ροή σχεδίασης.

Κατά την ανάλυση της λειτουργίας του και την παρουσίαση των αποτελεσμάτων θα γίνει εμφανής η βελτίωση που παρουσιάζεται βάσει μετρικών που χρησιμοποιούνται σε αντίστοιχα εργαλεία, όπως το συνολικό μήκος καλωδίου αλλά και ο συνολικός καταλαμβανόμενος χώρος της σχεδίασης.

Επιπρόσθετα, θα γίνει εκτενής αναφορά στον τρόπο και την μεθοδολογία που ακολουθούμε προκειμένου να αδράξουμε ακόμη καλύτερα αποτελέσματα ως προς την ταχύτητα και τον χρονοισμό του τελικού κυκλώματος, εντάσσοντας στην διαδικασία βελτιστοποίησης ζητήματα που άπτονται των φαινομένων πτώσης τάσης που παρατηρούνται κατά την λειτουργία της εκάστοτε ψηφιακής σχεδίασης.

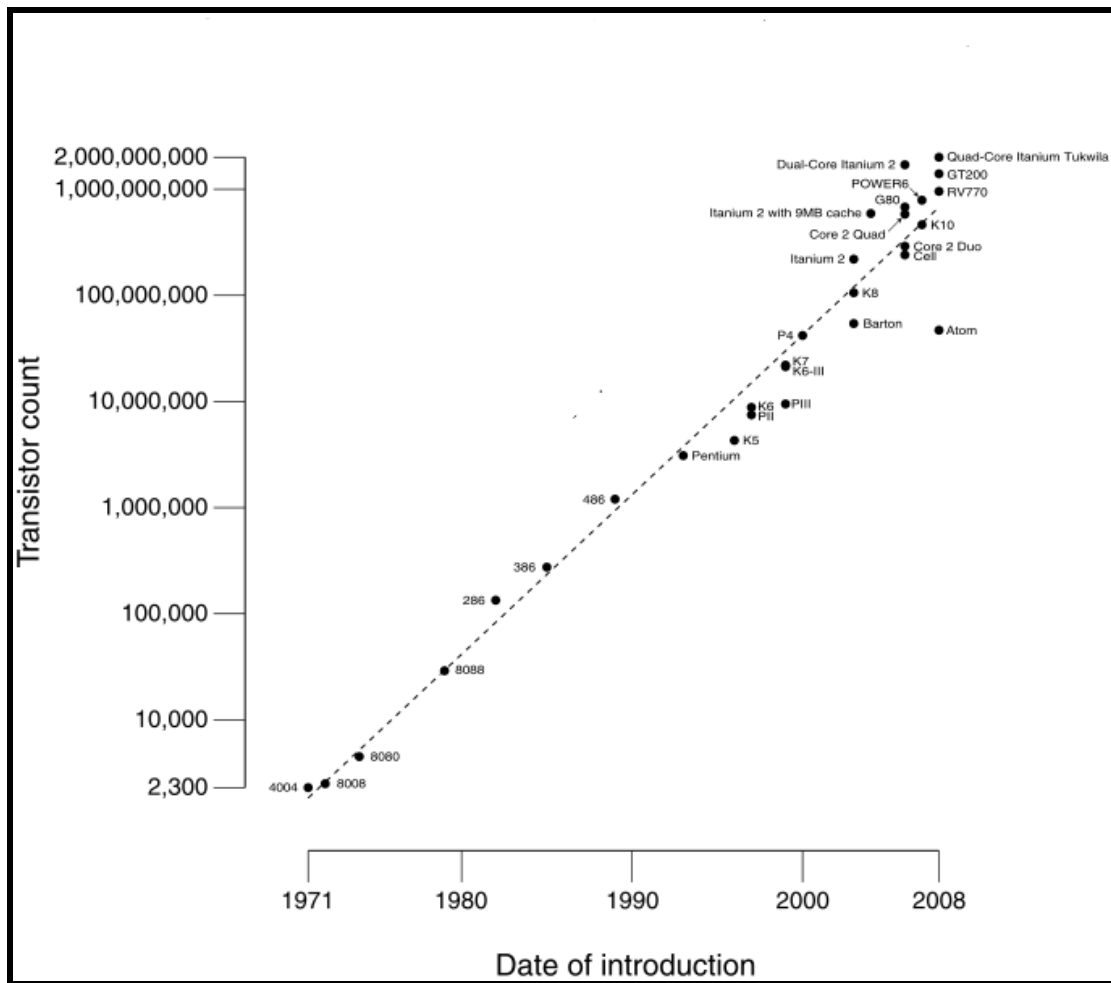
ΚΕΦΑΛΑΙΟ 1

ΣΧΕΔΙΑΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ

1.1 ΣΧΕΔΙΑΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ

Η σχεδίαση ολοκληρωμένων κυκλωμάτων (integrated circuit design) αποτελεί ένα υποσύνολο του γνωστικού αντικειμένου των Ηλεκτρολόγων Μηχανικών. Η ψηφιακή σχεδίαση ολοκληρωμένων κυκλωμάτων παράγει κυκλωματικά στοιχεία όπως μικροεπεξεργαστές (microprocessors), μνήμες (RAM memories, ROM memories, flash memories), FPGAs και ASICs (Application Specific Integrated Circuits). Τα σύγχρονα ολοκληρωμένα κυκλώματα είναι αρκούντως πολύπλοκα, με ένα chip μεγάλου μεγέθους να αποτελείται πλέον από πάνω από ένα δισεκατομμύριο τρανζίστορ. Το γεγονός αυτό, σε συνάρτηση με την πίεση της αγοράς για παραγωγή ολοκληρωμένων κυκλωμάτων στο μικρότερο δυνατό χρονικό διάστημα έχει οδηγήσει στην ανάπτυξη και χρήση εργαλείων αυτοματοποιημένης σχεδίασης (automated design tools). Η χρήση εργαλείων αυτής της κατηγορίας κρίνεται πλέον απαραίτητη στην πλειοψηφία των βημάτων που ακολουθούνται κατά την σχεδίαση ενός κυκλώματος.

Στην παρακάτω εικόνα παρουσιάζεται η ραγδαία αύξηση του αριθμού των τρανζίστορ σε ορισμένες από τις σημαντικότερες κεντρικές μονάδες επεξεργαστών που έχουν κατασκευαστεί την τελευταία τριακονταετία, σε συνάρτηση με τον νόμο του Moore που υποστηρίζει πως υπάρχει διπλασιασμός του αριθμού των τρανζίστορ που συνθέτουν ένα επεξεργαστή κάθε δύο χρόνια.



ΕΙΚΟΝΑ 1.1: CPU Transistor Count – Moore’s Law

1.2 ΡΟΗ ΣΧΕΔΙΑΣΗΣ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ

Η διαδικασία που ακολουθείται κατά την σχεδίαση ενός IC αποτελείται από πολλαπλά βήματα, με βασικότερα των οποίων τα ακόλουθα:

1. *Feasibility Study*: Ανάλυση του απώτερου στόχου, εκτίμηση των προβλημάτων που μπορεί να παρουσιαστούν και των πόρων που έχουμε στη διάθεσή μας για την υλοποίηση του τελικού αποτελέσματος.
2. *Die Size Estimation*: Εκτίμηση του χώρου που απαιτείται για την υλοποίηση του κυκλώματος.
3. *Functional Verification*: Επαλήθευση της λογικής της σχεδίασης. Η σχεδίασή μας πρέπει να πληρεί τις λειτουργικές προδιαγραφές που έχουν τεθεί.
4. *RTL Design*: Περιγραφή της λειτουργίας του κυκλώματος σε επίπεδο καταχωρητών (register-transfer level).
5. *RTL Simulation*: Προσομοίωση της λειτουργίας του κυκλώματος βάσει της περιγραφής που έχει δοθεί στο προηγούμενο βήμα.
6. *Logic Simulation*: Προσομοίωση της λειτουργίας της σχεδίασης με χρήση κατάλληλων λογισμικών.
7. *Floor Planning*: Σχηματική αναπαράσταση μιας «πρώιμης» τοπολογίας όλων των λειτουργικών τμημάτων της σχεδίασης.
8. *Layout*: Αναπαράσταση του ολοκληρωμένου κυκλώματος με μια σειρά γεωμετρικών σχημάτων τα οποία αντιστοιχούν στο σύνολο των στρωμάτων ημιαγωγού, μετάλλου και οξειδίου που απαρτίζουν την σχεδίαση.
9. *Static Timing Analysis*: Μελέτη του χρονισμού του ολοκληρωμένου κυκλώματος, η οποία δεν απαιτεί την εκ νέου προσομοίωση της λειτουργικότητάς του.
10. *Layout Review*: Επανεξέταση του layout που έχουμε σχεδιάσει. Η προκείμενη διαδικασία μπορεί να οδηγήσει στον επανασχεδιασμό του ολοκληρωμένου,

μπορεί δηλαδή να λειτουργήσει αναδρομικά και να επανατροφοδοτήσει με επιπλέον πληροφορίες προηγούμενα βήματα της διαδικασίας που έχουν παρέλθει χρονικά.

11. *Design For Test*: Χρήση συγκεκριμένων τεχνικών σχεδίασης οι οποίες προσδίδουν ιδιαίτερα χαρακτηριστικά στο κύκλωμα έτσι ώστε να γίνεται πιο εύκολη η διαδικασία ελέγχου της ορθότητάς του.
12. *Automatic Test Pattern Generation*: Εύρεση της κατάλληλης αλληλουχίας εισόδων, η οποία όταν εφαρμοσθεί βοηθάει στον εντοπισμό τυχών λαθών που μπορεί να υπάρχουν στη συμπεριφορά του κυκλώματος.
13. *Design For Manufacturability*: Εφαρμογή μιας σειράς τεχνικών οι οποίες τροποποιούν κατάλληλα το κύκλωμα έτσι ώστε η υλοποίησή του σε βιομηχανικό περιβάλλον να καθίσταται ευκολότερη.
14. *Mask Data Preparation*: Το βήμα κατά το οποίο η layout περιγραφή του κυκλώματος «μεταφράζεται» σε κάποια καταλληλότερη μορφή η οποία μπορεί να χρησιμοποιηθεί από έναν photomask writer.
15. *Wafer Fabrication*: Η διαδικασία κατά την οποία δημιουργείται το ολοκληρωμένο κύκλωμα, αποκτώντας την μορφή που γνωρίζουμε μελετώντας μια οποιαδήποτε ηλεκτρονική συσκευή.
16. *Packaging*: Το τελευταίο, πρακτικά, στάδιο της κατασκευής. Το κύκλωμα «συσκευάζεται» σε κάποιο κεραμικό ή πλαστικό υλικό προκειμένου να αποφευχθεί η φθορά του, αλλά και να διευκολυνθεί η ηλεκτρική σύνδεσή του και η ενσωμάτωσή του σε ένα ολοκληρωμένο σύστημα.

17. *Device Characterization*: Η διαδικασία κατά την οποία μέσω μιας σειράς μετρήσεων, με την βοήθεια των κατάλληλων οργάνων, συγκεντρώνουμε και παρουσιάζουμε τα ιδιαίτερα χαρακτηριστικά της τελικής υλοποίησης, βάσει ευρέως αποδεκτών μετρικών.

18. *Yield Analysis*: Συλλογή και ανάλυση των κατάλληλων δεδομένων που απαιτούνται για τον εντοπισμό και την διόρθωση αστοχιών που προκύπτουν κατά την λειτουργία του ολοκληρωμένου.

Η παραπάνω αλληλουχία βημάτων δύναται να χωριστεί σε τρία επιμέρους στάδια:

- I. System Level Design
- II. RTL Design
- III. Physical Design

Στην προκειμένη διατριβή θα ασχοληθούμε ενδελεχώς με το τρίτο μέρος, την φυσική σχεδίαση (physical design), ενός ολοκληρωμένου κυκλώματος.

1.3 ΦΥΣΙΚΗ ΣΧΕΔΙΑΣΗ (PHYSICAL DESIGN)

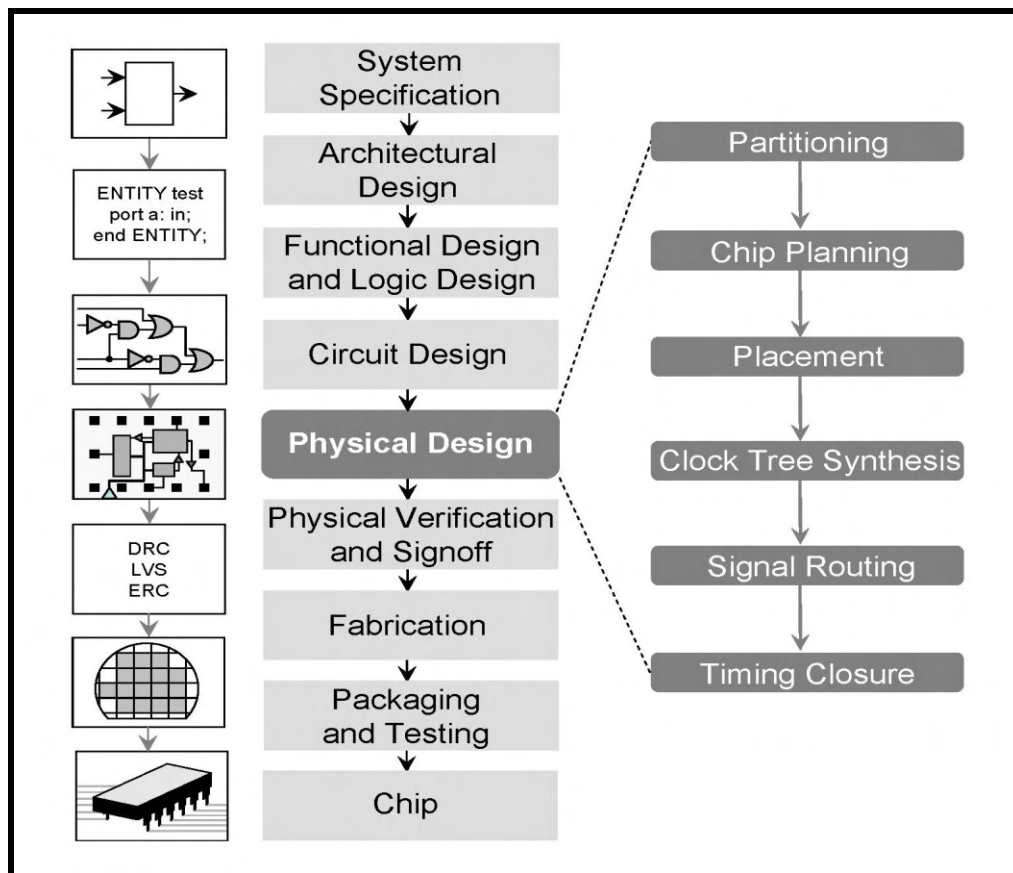
Η φυσική σχεδίαση ενός ολοκληρωμένου κυκλώματος αποτελείται από τα ακόλουθα βήματα:

1. *Design Netlist*: Ένα netlist σε επίπεδο πυλών (gate level netlist) είναι κατ'ουσίαν το αποτέλεσμα που προκύπτει από την διαδικασία της σύνθεσης (synthesis) ενός ψηφιακού κυκλώματος. Σύνθεση ονομάζεται η διαδικασία κατά την οποία μετατρέπουμε μια σχεδίαση η οποία μας δίνεται σε κάποια γλώσσα περιγραφής υλικού (hardware description language - HDL) σε ένα βέλτιστο netlist σε επίπεδο πυλών το οποίο προσδιορίζεται πλήρως από μια τεχνολογική βιβλιοθήκη. Δύο από τα πιο διαδεδομένα εργαλεία σύνθεσης ψηφιακών κυκλωμάτων είναι ο Synopsys Design Compiler και ο Cadence RTL Compiler.
2. *Floor Planning*: Το floorplanning είναι κατά πολλούς το πρώτο ουσιαστικό βήμα της φυσικής σχεδίασης. Στο βήμα αυτό κάνουμε μια πρώτη εκτίμηση του συνολικού χώρου που απαιτείται για τις βασικές δομικές μονάδες του chip και καθορίζουμε τις σχετικές τους θέσεις εντός του προκειμένου χώρου. Η διαδικασία είναι απαραίτητη διότι μέσω αυτής καθορίζεται το κατά πόσο η προς υλοποίηση σχεδίαση δύναται να χωρέσει εντός της διαθέσιμης περιοχής. Ο πλήρης καθορισμός της περιοχής πραγματοποιείται δυναμικά και μπορεί πολλές φορές να οδηγήσει στον επαναπροσδιορισμό της αρχιτεκτονικής της σχεδίασης.
3. *Partitioning*: Στο βήμα αυτό ο σχεδιαστής καλείται να βρει τον κατάλληλο τρόπο διαίρεσης της περιοχής, που καταλαμβάνει το chip, σε μικρότερες και ευκολότερα διαχειρίσιμες περιοχές. Αυτό γίνεται προκειμένου να διαχωρίσουμε τα functional blocks της σχεδίασης και να διευκολύνουμε περαιτέρω την διαδικασία του placement και του routing που περιγράφονται παρακάτω.
4. *Placement*: Η χωροθέτηση πραγματοποιείται σε τέσσερα βήματα, που απώτερο στόχο έχουν την βέλτιστη τοποθέτηση των κελιών της σχεδίασης στον προκαθορισμένο χώρο.

- a. *Pre-placement Optimization*: Στη φάση αυτή πραγματοποιούμε βελτιώσεις στην σχεδίασή μας σε επίπεδο netlist πριν αρχίσουμε την τοποθέτηση των κελιών πάνω στην επιθυμητή επιφάνεια (π.χ ελάτωση του μεγέθους των εμπλεκόμενων στη διαδικασία κελιών).
 - b. *In-Placement Optimization*: Εδώ, πραγματοποιείται επαναβελτιστοποίηση της λογικής της σχεδίασης μας. Το στάδιο αυτό εκτελείται κατ' επανάληψη και συμπεριλαμβάνει διαδικασίες βελτίωσης του κυκλώματος όπως cell sizing, cell moving, cell bypassing, gate duplication, buffer assertion και area recovery.
 - c. *Post-Placement Optimization*: Μετά την τοποθέτηση των πυλών διορθώνουμε όσες παραβιάσεις έχουν εμφανιστεί (π.χ setup, hold, max trans/cap violations).
 - d. *Post-Placement Optimization after Clock Tree Synthesis*: Στην τελευταία φάση της διαδικασίας του placement βελτιστοποιούμε τον χρονοισμό του κυκλώματος και προσπαθούμε να διατηρήσουμε το clock skew.
5. *Clock Tree Synthesis*: Ο στόχος του clock tree synthesis είναι να ελαχιστοποιήσουμε το skew και το insertion delay. Εάν το ρολόι της σχεδίασής μας διαμερίζεται σε πολλαπλά blocks τότε χρειάζεται επιμέρους skew analysis.
 6. *Routing*: Υπάρχουν δύο τύποι routing, το global routing και το detailed routing. Το global routing τοποθετεί τα routing resources τα οποία χρησιμοποιούνται για τις συνδέσεις μεταξύ των κελιών, ενώ το detailed routing αναθέτει συγκεκριμένα μονοπάτια (routes) σε συγκεκριμένα επίπεδα μετάλλου.
 7. *Physical Verification*: Στο τελευταίο στάδιο της φυσικής σχεδίασης ελέγχουμε την ορθότητα του layout που έχει παραχθεί. Αυτό περιλαμβάνει:
-

- a. Την πλήρη συμμόρφωση της σχεδίασής μας με όλες τις τεχνολογικές απαιτήσεις που έχουν τεθεί (Design Rule Checking - DRC).
- b. Την συνέπεια ως προς το αρχικό netlist (Layout vs. Schematic - LVS).
- c. Την εξάλειψη antenna effects (Antenna Rule Checking).
- d. Την πλήρη συμφωνία της σχεδίασης μας με τις ηλεκτρικές απαιτήσεις που έχουν τεθεί (Electrical Rule Checking - ERC).

Στην παρακάτω εικόνα παρουσιάζεται μια τυπική ροή σχεδίασης ενός VLSI κυκλώματος, με ιδιαίτερη έμφαση στο στάδιο της φυσικής σχεδίασης.



EIKONA 1.2: VLSI Circuits Design Flow

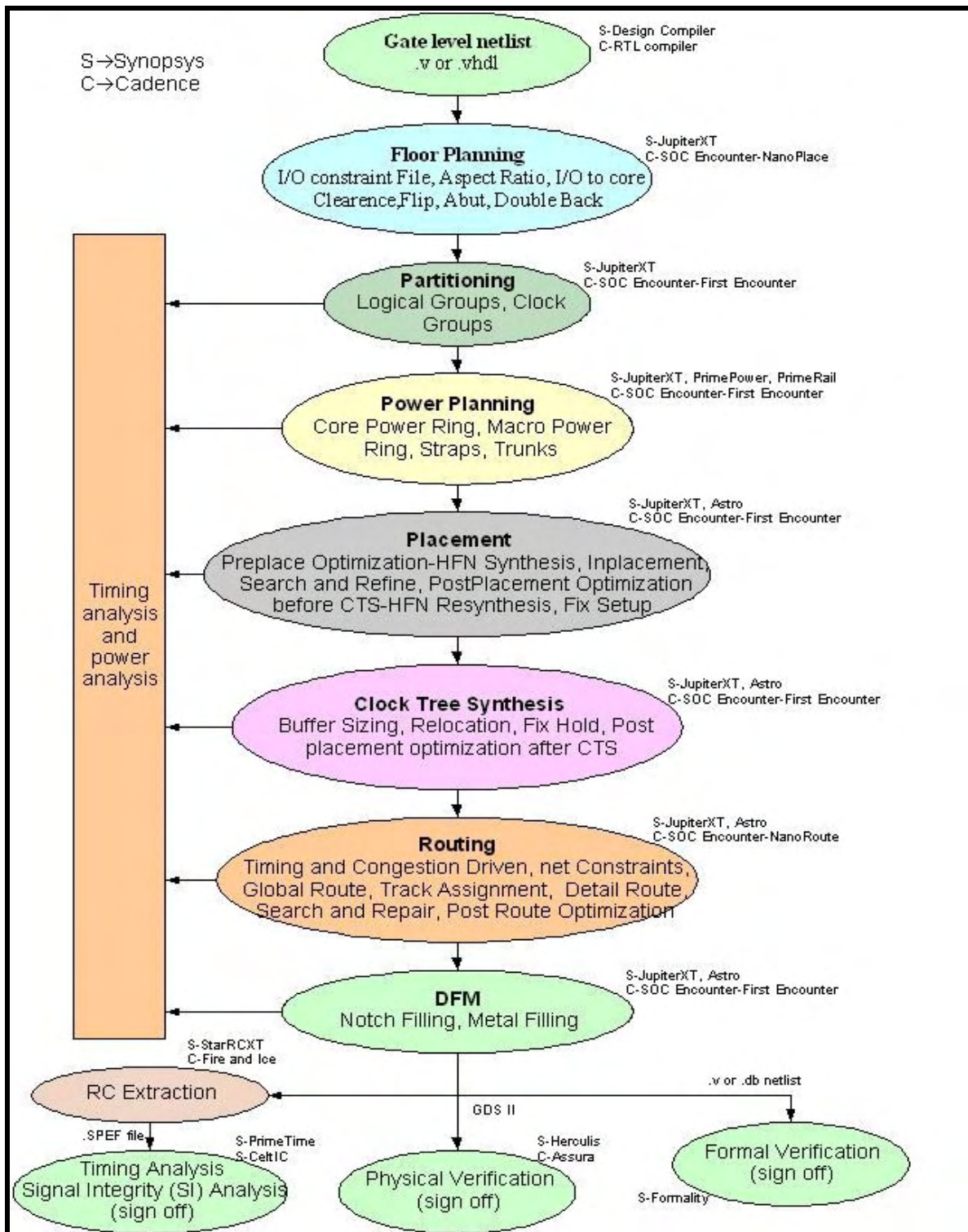
Τα βήματα που περιγράψαμε παραπάνω είναι κατ' ουσίαν μια αφαιρετική περιγραφή του τελευταίου σταδίου σχεδίασης ενός ολοκληρωμένου κυκλώματος. Πιο αναλυτικές ροές σχεδίασης μπορούν να προκύψουν ανάλογα με τα λογισμικά που χρησιμοποιούνται και την επιλογή των κατάλληλων μεθοδολογιών. Μερικά από τα πιο ευρέως διαδεδομένα λογισμικά που χρησιμοποιούνται στην βιομηχανία είναι τα ακόλουθα:

- Cadence (SoC Encounter, NanoRoute)
- Synopsys (Design Compiler)
- Magma (Blast Fusion)
- Mentor (Olympus SoC, Calibre)

Μια πιο λεπτομερής ροή της φυσικής σχεδίασης παρουσιάζεται στην παρακάτω εικόνα, ενώ παράλληλα αναφέρονται τα ακριβή βήματα και τα εργαλεία που χρησιμοποιούνται σε κάθε ένα από αυτά.

Στο σημείο αυτό πρέπει να αναφέρουμε πως κατά την διαδικασία σχεδίασης ενός ολοκληρωμένου κυκλώματος δύνανται να ακολουθηθούν διαφορετικές ροές σχεδίασης (είτε πιο λεπτομερείς είτε πιο γενικευμένες), ανάλογα με την σημαντικότητα και τη βαρύτητα που δίνεται σε κάθε ένα από τα παραπάνω βήματα.

Η παρουσίαση της παρακείμενης ροής σχεδίασης, με χρήση των συγκεκριμένων εργαλείων, γίνεται λόγω της ευρείας χρήσης της από την πλειοψηφία των, σχετιζόμενων με τον κλάδο, του σχεδιασμού ολοκληρωμένων κυκλωμάτων, εταιρειών.



EIKONA 1.3: Design Flow & CAD Tools

1.4 ΕΡΓΑΛΕΙΑ CAD (COMPUTER-AIDED DESIGN)

Όταν αναφερόμαστε στο όρο Computer-Aided Design (CAD), αναφερόμαστε πρακτικά στην χρησιμοποίηση της τεχνολογίας των υπολογιστών για την διευκόλυνση της διαδικασίας σχεδιασμού ενός αντικειμένου.

Τα προγράμματα CAD πρωτοεμφανίστηκαν την δεκαετία του '80 και οδήγησαν στο να μειωθεί η άμεση ανάγκη επιπλέον εξειδικευμένου προσωπικού, δίνοντας περαιτέρω δυνατότητες ανάπτυξης σε εταιρείες μεσαίου μεγέθους. Το γεγονός πως, σε σύντομο σχετικά χρονικό διάστημα, από την πρώτη εμφάνισή τους, οι εταιρείες κατέστησαν οικονομικά προσιτή την απόκτησή τους και την χρήση τους στον προσωπικό υπολογιστή ενός μέσου χρήστη, βοήθησε στο μέγιστο βαθμό τους σχετιζόμενους με το αντικείμενο κλάδους μηχανικής.

1.5 ΕΡΓΑΛΕΙΑ EDA (ELECTRONIC DESIGN AUTOMATION)

Η αυτοματοποιημένη ηλεκτρονική σχεδίαση (Electronic Design Automation - EDA), είναι κατ'ουσίαν, η υλοποίηση ενός κυκλώματος με παράλληλη χρήση ειδικών λογισμικών τα οποία έχουν δημιουργηθεί κατά περίπτωση για την υποβοήθηση της διαδικασίας. Στην περίπτωση που εξετάζουμε, στον σχεδιασμό, δηλαδή, ενός ολοκληρωμένου κυκλώματος, η εν σειρά χρησιμοποίηση μιας ομάδας εργαλείων για την παραγωγή της τελικής σχεδίασης στοιχειοθετεί μια ροή σχεδιασμού (design flow).

1.5.1 ΙΣΤΟΡΙΑ ΤΟΥ EDA

Πριν από την αυτοματοποίηση της διαδικασίας, ο σχεδιασμός των ολοκληρωμένων κυκλωμάτων πραγματοποιούνταν χειρονακτικά. Ο σχεδιασμός βασιζόταν στην γραφική απεικόνιση του κυκλώματος και πιο συγκεκριμένα στην τροποποίηση της ηλεκτρονικής περιγραφής του κυκλώματος στην γραφική αναπαράστασή του. Μια από τις πιο γνωστές εταιρείες της πρώιμης αυτής περιόδου, ήταν η Calma η οποία δημιούργησε το format GDSII το οποίο χρησιμοποιείται ακόμη και σήμερα.

Στα μέσα της δεκαετίας του '70 ξεκίνησε η ανάπτυξη των πρώτων εργαλείων για την κατάλληλη χωροθέτηση ενός κυκλώματος (placement). Τα πρακτικά των επιστημονικών συνεδρίων της εποχής παρουσιάζουν ανάγλυφα τις καινοτομίες που εισήχθησαν στην ροή σχεδίασης ολοκληρωμένων κυκλωμάτων και αποτέλεσαν την βάση για την έρευνα στον προκείμενο τομέα τα χρόνια που ακολούθησαν.

Τα πρώτα ολοκληρωμένα εργαλεία EDA, παρόλαυτα, εμφανίστηκαν αρχικά εντός ακαδημαϊκού περιβάλλοντος. Ένα από τα δημοφιλέστερα εργαλεία της εποχής αυτής ήταν το VLSI Tools Tarball το οποίο αναπτύχθηκε στο πανεπιστήμιο του Berkeley, και ήταν, κατ'ουσίαν, μια συλλογή εφαρμογών σε περιβάλλον UNIX για τον σχεδιασμό VLSI συστημάτων.

Στις αρχές της δεκαετίας του '80 αρχίζει η βιομηχανική παραγωγή εργαλείων EDA. Μέχρι εκείνη την χρονική στιγμή οι μεγαλύτερες εταιρείες του κλάδου ανέπτυσαν εργαλεία αυτής της κατηγορίας προς ιδίαν χρήση, χωρίς να παρέχεται η δυνατότητα απόκτησής τους από άλλους φορείς. Πολλοί, όμως σχεδιαστές, κρίνοντας πως ο τομέας παρουσίαζε μεγάλο τεχνολογικό ενδιαφέρον και είχε δυνατότητες περαιτέρω οικονομικής ανάπτυξης αποφάσισαν να δημιουργήσουν νέες αυτόνομες εταιρείες με αποκλειστικό στόχο την δημιουργία EDA εργαλείων. Την προκείμενη

περίοδο ιδρύθηκαν σημαντικές εταιρείες όπως η Mentor Graphics και η Valid Logic Systems.

Επιπρόσθετα, το 1986 και το 1987 παρουσιάστηκαν δύο υψηλού επιπέδου γλώσσες περιγραφής υλικού, η Verilog και η VHDL αντίστοιχα, οι οποίες χρησιμοποιήθηκαν άμεσα από την πλειοψηφία των προσομοιωτών λειτουργίας ψηφιακών κυκλωμάτων (simulators) προκειμένου να διευκολυνθεί ο άμεσος έλεγχος της ορθότητας της εκάστοτε σχεδίασης, γεγονός το οποίο με τη σειρά του οδήγησε στην δημιουργία των πρώτων εργαλείων λογικής σύνθεσης (logic synthesis).

Οι σύγχρονες ροές σχεδίασης ολοκληρωμένων κυκλωμάτων αποτελούνται πλέον από πολλαπλά βήματα, σε κάθε ένα από τα οποία γίνεται χρήση του κατάλληλου εργαλείου. Στην αρχή της προαναφερθείσας διαδικασίας έχουμε κατά κανόνα την δημιουργία μιας περιγραφής του κυκλώματος βάσει κάποιας HDL γλώσσας σε επίπεδο κελιών, τα οποία είναι τεχνολογικά ανεξάρτητα. Ακολούθως, ο σχεδιαστής παρέχει τις κατάλληλες τεχνολογικές βιβλιοθήκες οι οποίες συνδράμουν στην επιτυχή και λεπτομερή προσομοίωση της λειτουργίας του κυκλώματος, ενώ κατά το τελευταίο στάδιο παρέχονται στον σχεδιαστή οι τελικές προδιαγραφές για τις συνθήκες λειτουργίας του.

1.5.2 ΤΟΜΕΙΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ EDA

Οι βασικότεροι τομείς της σχεδίασης ενός ολοκληρωμένου κυκλώματος, για τους οποίους έχουν αναπτυχθεί κατάλληλα λογισμικά τα οποία εμπίπτουν στην κατηγορία των EDA εργαλείων είναι οι ακόλουθοι:

- ΣΧΕΔΙΑΣΜΟΣ (*DESIGN*)
 - *High Level Synthesis*
 - *Logic Synthesis*
 - *Schematic Capture*
 - *Layout*

- ΠΡΟΣΟΜΟΙΩΣΗ (*SIMULATION*)
 - *Logic Simulation*
 - *Behavioral Simulation*
 - *Hardware Emulation*

- ΑΝΑΛΥΣΗ ΚΑΙ ΕΠΑΛΗΘΕΥΣΗ (*ANALYSIS & VERIFICATION*)
 - *Functional Verification*
 - *Formal Verification*
 - *Equivalence Checking*
 - *Static Timing Analysis*
 - *Physical Verification*

- ΚΑΤΑΣΚΕΥΗ (*MANUFACTURING*)
 - *Mask Data Preparation*
 - *Mask Generation*

- *Automatic Test Pattern Generation*
- *Built-In Self-Test*

1.6 ΣΤΟΧΟΣ ΤΗΣ ΔΙΑΤΡΙΒΗΣ

Στόχος αυτής της διατριβής είναι η παρουσίαση ενός νέου εργαλείου Placement. Ενός εργαλείου το οποίο τοποθετεί στις βέλτιστες θέσεις τους τα κυκλωματικά στοιχεία μιας σχεδίασης προκειμένου να ελαχιστοποιηθεί ο συνολικός χώρος που καταλαμβάνουν όπως και το συνολικό μήκος καλωδίου που απαιτείται για την διασύνδεσή τους. Επιπρόσθετα, παρουσιάζεται μία νέα ιδέα για την υλοποίηση ενός IR-Drop Driven re-Placement μιας σχεδίασης με απώτερο στόχο την βελτίωση του χρόνισμού της.

Τέλος, στο σημείο αυτό πρέπει να αναφέρουμε πως το προκείμενο εργαλείο εντάσσεται και εναρμονίζεται πλήρως με τις υπάρχουσες ροές σχεδίασης ψηφιακών κυκλωμάτων, αφού τα αρχεία εισόδου και εξόδου μπορούν να χρησιμοποιηθούν για την ανατροφοδότηση εταιρικών εργαλείων που χρησιμοποιούνται κατά κόρον στην βιομηχανία.

ΚΕΦΑΛΑΙΟ 2

PLACEMENT

Στο κεφάλαιο αυτό θα παρουσιάσουμε αναλυτικά το θεωρητικό υπόβαθρο του προβλήματος του Placement και τις μελέτες που έχουν γίνει στο συγκεκριμένο ζήτημα έως σήμερα.

2.1 ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

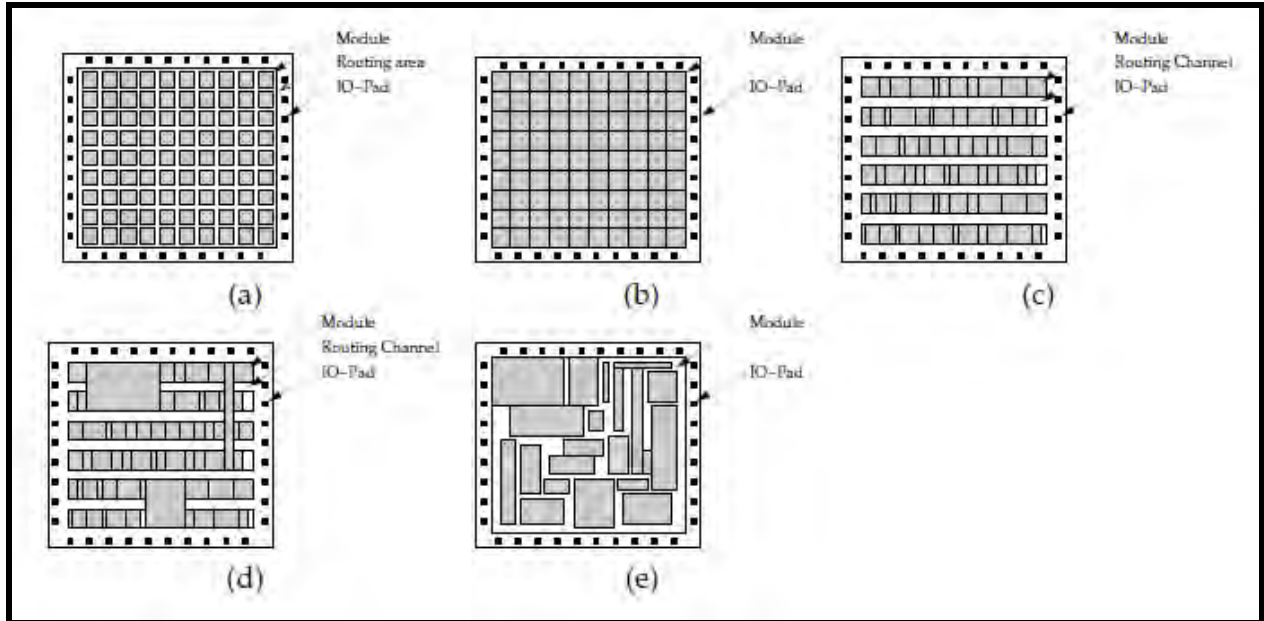
Η διαδικασία του Placement μπορεί να οριστεί ως το πρόβλημα χωροθέτησης (εύρεσης του κατάλληλου τρόπου τοποθέτησης) μιας ομάδας αντικειμένων σε έναν πλήρως καθορισμένο χώρο, προκειμένου να ελαχιστοποιηθεί η τιμή μιας αντικειμενικής συνάρτησης. Στην περίπτωση που καλούμαστε να τοποθετήσουμε «ιδανικά» ένα σύνολο κελιών τα οποία αποτελούν ένα ολοκληρωμένο κύκλωμα, η αντικειμενική αυτή συνάρτηση αναφέρεται, στην πλειοψηφία των περιπτώσεων, είτε στο συνολικό μήκος καλωδίου που χρησιμοποιείται για τη σύνδεση των κελιών, είτε στην συνολική επιφάνεια που αυτά καταλαμβάνουν.

Το τελικό αποτέλεσμα που προκύπτει μετά την ολοκλήρωση της προκειμένης διαδικασίας αποτυπώνεται σε μορφή layout. Υπάρχουν πέντε βασικές κατηγορίες

layout οι οποίες διαχωρίζονται, κατά κύριο λόγο, από την ελευθερία που παρέχεται στον σχεδιαστή κατά την υλοποίηση του τελικού αποτελέσματος:

1. Gate Array
2. Sea Of Gates
3. Standard Cell
4. Mixed Cell
5. General Cell (Macros)

Στην εικόνα που ακολουθεί παρουσιάζονται ενδεικτικά παραδείγματα των παραπάνω κατηγοριών.



EIKONA 2.1: Layout Models, a.gate array, b.sea of gates, c.standard cell, d.mixed cell, e.general cell

Στην παρούσα διατριβή θα ασχοληθούμε με την τρίτη μορφή, το standard cell layout. Σε αυτή την περίπτωση όλα τα κελιά έχουν το ίδιο ύψος και διαφοροποιούνται μόνο ως προς το μήκος τους. Τα κελιά τοποθετούνται εν σειρά, σε προκαθορισμένες από τον σχεδιαστή γραμμές οι οποίες απέχουν μεταξύ τους μια συγκεκριμένη απόσταση.

Στη διεθνή βιβλιογραφία το πρόβλημα του Placement διαιρείται σε πέντε υποκατηγορίες - βήματα:

1. *Global Placement*: Το global placement παράγει ένα αρχικό placement στο οποίο υπάρχει αλληλεπικάλυψη (overlap) μεταξύ των κελιών. Η διαδικασία του global placement μπορεί να εκτελεστεί επαναληπτικά, προκειμένου να παραχθεί κάποιο καλύτερο αποτέλεσμα.
2. *Final Placement*: Το final placement βελτιστοποιεί τις θέσεις των κελιών που έχουν προκύψει από το παραπάνω βήμα. Η διαδικασία είναι πάντοτε επαναληπτική και πρακτικά δίνει αποτελέσματα τα οποία κινούνται εντός ενός συγκεκριμένου συνόλου λύσεων. Η τελική λύση που προκύπτει είναι, συνήθως, μια συγκεκριμένη χωροθέτηση των κελιών χωρίς κανένα φαινόμενο επικάλυψης.
3. *Area Minimization*: Το πρόβλημα ελαχιστοποίησης της καταλαμβανόμενης από τα κελιά περιοχής είναι κατά βάση ένα διοδιάστατο πρόβλημα τοποθέτησης αντικειμένων σε περιορισμένο χώρο και ανήκει στα NP-hard προβλήματα.
4. *Legalization*: Εάν το τελικό κύκλωμα εξακολουθεί να παρουσιάζει αλληλεπικάλυψη μεταξύ των κελιών, πρέπει εκ νέου να εφαρμοστούν τεχνικές «νομιμοποίησης» των θέσεών τους.

5. *Post-Placement Optimization*: Σε ορισμένες περιπτώσεις η τελική λύση του προβλήματος μπορεί να βελτιωθεί αισθητά, εφαρμόζοντας επιπλέον τεχνικές οι οποίες αφήνουν ανέπαφη την πλειοψηφία των κελιών.

2.2 GLOBAL PLACEMENT

Εάν ανατρέξει κάποιος στη σύγχρονη βιβλιογραφία για το ζήτημα του global placement θα παρατηρήσει πως οι προσεγγίσεις που έχουν παρουσιαστεί μέχρι και σήμερα είναι ελάχιστες. Οι βασικότερες από αυτές είναι ο διαχωρισμός του κυκλώματος βάσει του γράφου διασυνδεσιμότητας των στοιχείων του (graph partitioning) και ο διαχωρισμός με χρήση αναλυτικών μεθόδων (analytic based placement). Στα κεφάλαια που ακολουθούν θα παρουσιαστούν οι προαναφερθείσες προσεγγίσεις ενώ, επιπρόσθετα θα γίνει αναφορά και σε κάποιες σπανιότερα εφαρμοζόμενες τεχνικές.

2.2.1. GRAPH PARTITIONING

Η συγκεκριμένη προσέγγιση βασίζεται στην εύρεση του κατάλληλου τρόπου διαχωρισμού του υπεργράφου (hypergraph) που σχηματίζεται εάν λάβουμε υπόψη μας όλες τις συνδέσεις μεταξύ των στοιχείων που απαρτίζουν το προς εξέταση κύκλωμα.

Η διεργασία διαχωρισμού (partitioning) εκτελείται αναδρομικά και τμηματοποιεί την σχεδίαση βάσει προεπιλεγμένων κριτηρίων.

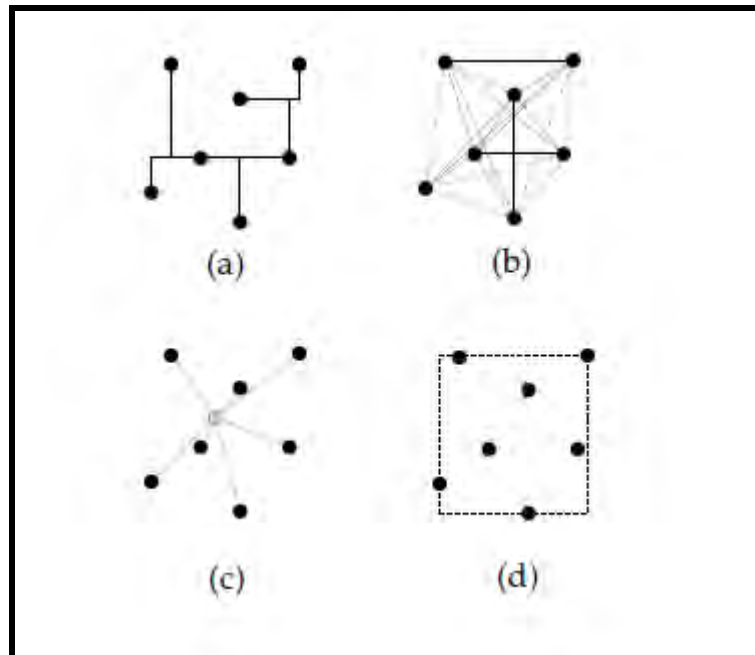
Στην πλειοψηφία των περιπτώσεων βασικό κριτήριο αποτελεί η ισόποση κατανομή των βαρών των ακμών του γράφου στις δύο νέες περιοχές που δημιουργούνται. Όταν αναφερόμαστε στο βάρος μιας ακμής αναφερόμαστε κατ'ουσίαν σε μια τιμή που της έχει ανατεθεί σε σχέση με την βαρύτητα που παρουσιάζει ως προς την σημαντικότητα της σύνδεσης των προκείμενων κελιών. Ο τρόπος υπολογισμού έχει άμεση σχέση με το net model που χρησιμοποιείται για την αναπαράσταση του κυκλώματος σε μορφή γράφου. Τα σημαντικότερα μοντέλα είναι τα ακόλουθα:

1. Distance Based: Για τον προσδιορισμό του μήκους καλωδίου του εκάστοτε net υπολογίζεται το άθροισμα της ευκλείδειας απόστασης μεταξύ όλων εκείνων των στοιχείων του κυκλώματος που συνδέονται μεταξύ τους και το απαρτίζουν.
2. Steiner Tree: Στην πλειοψηφία των αλγορίθμων που χρησιμοποιούν αυτή τη μορφή αναπαράστασης των net παρουσιάζεται επιπρόσθετα και μια επέκταση αυτού του μοντέλου το ευθύγραμμο (rectilinear) Steiner Tree. Η χρήση ενός rectilinear Steiner Tree παρουσιάζει σημαντικά πλεονεκτήματα εάν αναλογιστούμε πως την διαδικασία του placement διαδέχεται η διαδικασία του routing. Ένα δέντρο της προκείμενης μορφής αποτελείται αποκλειστικά από οριζόντιες και κάθετες ακμές οι οποίες εκτείνονται προς εκείνες τις κατευθύνσεις που απαιτείται για να συμπεριληφθούν όλα τα εμπλεκόμενα στο net στοιχεία. Ο αριθμός των ακμών είθισται να είναι ο μικρότερος δυνατός βάσει του οποίου ελαχιστοποιείται το συνολικό μήκος καλωδίου που απαιτείται για την διασυνδεσιμότητα των κελιών και των pins που αποτελούν το net.
3. Clique: Η χρήση του μοντέλου κλίκας είναι αρκούντως διαδεδομένη στα εργαλεία που υλοποιούν αλγορίθμους global placement. Το συγκεκριμένο

μοντέλο βασίζεται στην κατάλληλη μετατροπή ενός net σε ένα υπογράφο κλίκας. Για τον υπολογισμό του βάρους των ακμών που συνθέτουν το net χρησιμοποιείται ο μαθηματικός τύπος : $w_{(i,j)}=2/k$. Το βάρος, δηλαδή, μιας υπάρχουσας ακμής που προσδιορίζει την σύνδεση μεταξύ ενός στοιχείου i και ενός στοιχείου j ισούται με το αποτέλεσμα της διαίρεσης του δύο με το συνολικό αριθμό στοιχείων που απαρτίζουν την κλίκα. Ο υπολογισμός γίνεται βάσει του παραπάνω τύπου προκειμένου να ελαττώσουμε, χωρίς να διαταράξουμε την συνολική, επιρροή των μεγάλων δικτυωμάτων έναντι των μικρότερων που αποτελούν συνήθως την πλειοψηφία σε μια μέση ψηφιακή σχεδίαση. Βασικό χαρακτηριστικό και πλεονέκτημα του μοντέλου είναι η μικρή πολυπλοκότητα υπολογισμού των βαρών.

4. Star: Το μοντέλο αυτό είναι μια παραλλαγή του μοντέλου κλίκας, όπου οι υπερακμές του σχηματιζόμενου υπεργράφου, μετατρέπονται σε υπογράφους σε σχηματισμό αστέρα.
5. Bounding Box: Ο υπολογισμός του συνολικού μήκους καλωδίου γίνεται βάσει της ημiperιμέτρου του μικρότερου παραλληλόγραμμου που περιβάλλει κάθε net.

Στην παρακάτω εικόνα παρουσιάζονται χαρακτηριστικά παραδείγματα των παραπάνω μοντέλων.



ΕΙΚΟΝΑ 2.2: Net Models, a.Rectilinear Steiner Tree b.Clique c.Star d.Bounding Box

2.2.2 ANALYTIC-BASED PARTITIONING

Η χρήση αναλυτικών μεθόδων για το κατάλληλο partitioning του κυκλώματος σχετίζεται άμεσα με το graph partitioning. Η βασική τους, όμως, διαφορά έγκειται στο ότι ο βασικός στόχος, πλέον, δεν είναι η εύρεση του min cut, αλλά ο προσδιορισμός της καλύτερης δυνατής θέσης για το cut line. Ο σημαντικότερος αλγόριθμος αυτής της κατηγορίας είναι ο Gordian. Στα κεφάλαια που ακολουθούν θα παρουσιαστεί τόσο ο προαναφερθείς αλγόριθμος όσο και μια από τις βασικότερες παραλλαγές του ο Gordian-L.

2.2.2.1 Gordian

Ο αλγόριθμος Gordian έχει δύο βασικά χαρακτηριστικά, πραγματοποιεί global optimization σε κάθε βήμα και κατά το partitioning δημιουργεί αποκλειστικά ορθογώνιες περιοχές.

Συνοπτικά τα βήματα του αλγορίθμου είναι τα εξής:

- Τοποθετούμε όλα τα στοιχεία του κυκλώματος εντός της προβλεπόμενης περιοχής που έχουμε στη διάθεσή μας.
- Χρησιμοποιούμε κατάλληλες μεθόδους τμηματοποίησης για να δημιουργήσουμε ομάδες στοιχείων ενώ παράλληλα ακολουθούμε ρητά τους κανόνες καθολικής βελτιστοποίησης που έχουν τεθεί.
- Όταν οι ομάδες στοιχείων που έχουν σχηματιστεί έχουν πλήθος στοιχείων μικρότερο από μια μεταβλητή, η τιμή της οποίας έχει προαποφασιστεί στην αρχή της εκτέλεσης του αλγορίθμου, τοποθετούμε καταλλήλως τα στοιχεία στον διαθέσιμο χώρο.

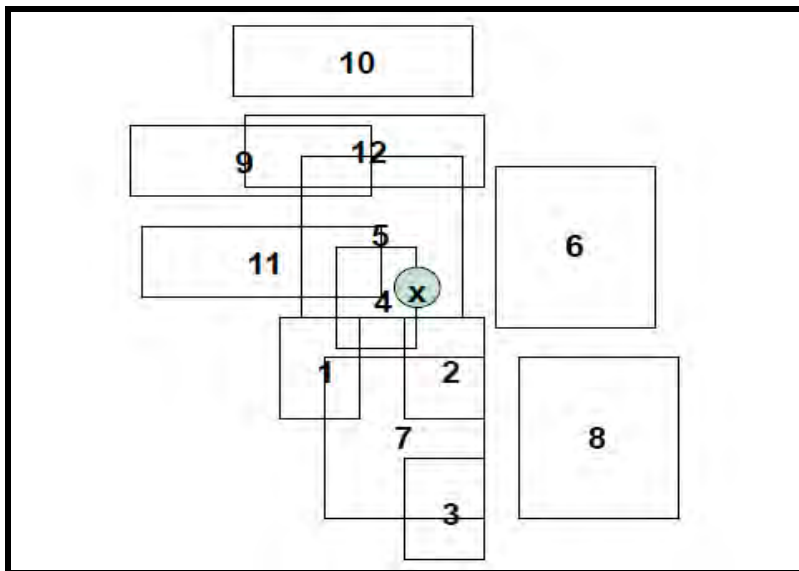
Η συνολική λειτουργία του αλγορίθμου μπορεί να αποτυπωθεί υπό την μορφή ψευδοκώδικα, μία τυποποιημένη εκδοχή του οποίου παρουσιάζεται στην εικόνα που ακολουθεί.


```
Gordian Procedure:  
l := 1  
global_optimize( l );  
while ( there exists  $|M_l| > K$  )  
  for each r  
    partition( r, r', r'' );  
  l ++;  
  setup_constraints( l );  
  global_optimize( l );  
  re-partition( l );  
final_placement( l );  
end_procedure;
```

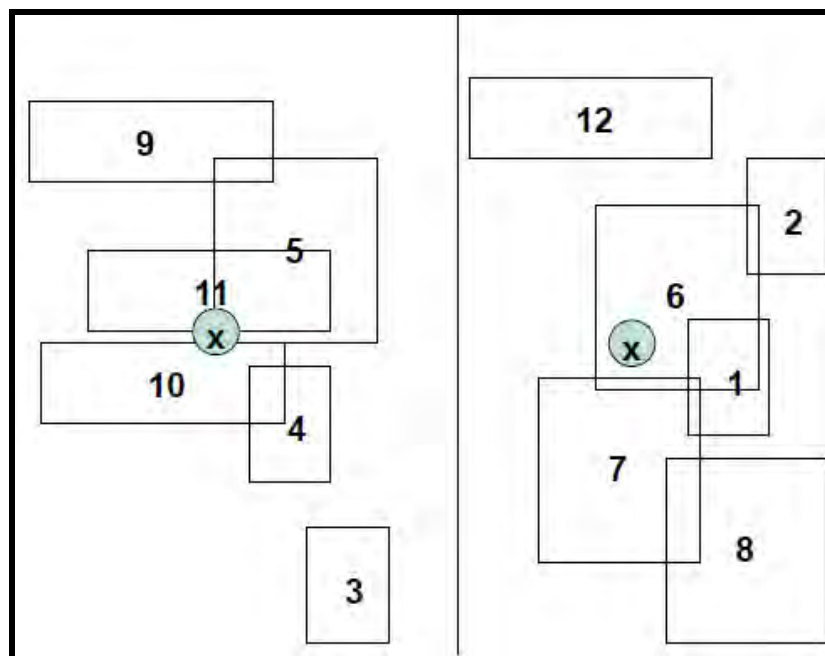
EIKONA 2.3: Gordian Algorithm pseudocode

όπου l είναι το προκείμενο βήμα εκτέλεσης και r είναι το προς εξέταση partition.

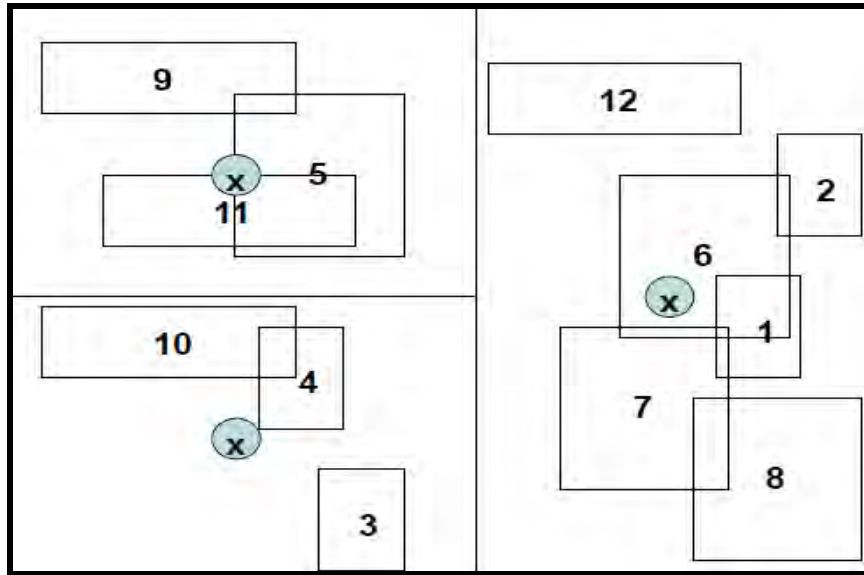
Στις εικόνες που ακολουθούν παρουσιάζεται ένα απλό παράδειγμα εφαρμογής του αλγορίθμου, για την χωροθέτηση ενός κυκλώματος έντεκα πυλών με ελάχιστο αριθμό κελιών ανά εξεταζόμενη περιοχή ίσο με τρία.



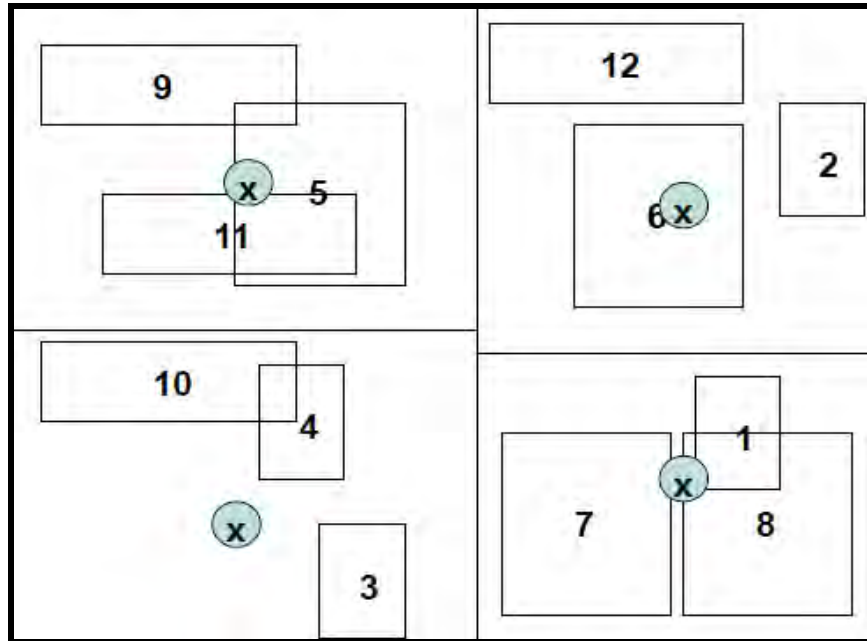
a) Τυχαία αρχική τοποθέτηση των κελιών



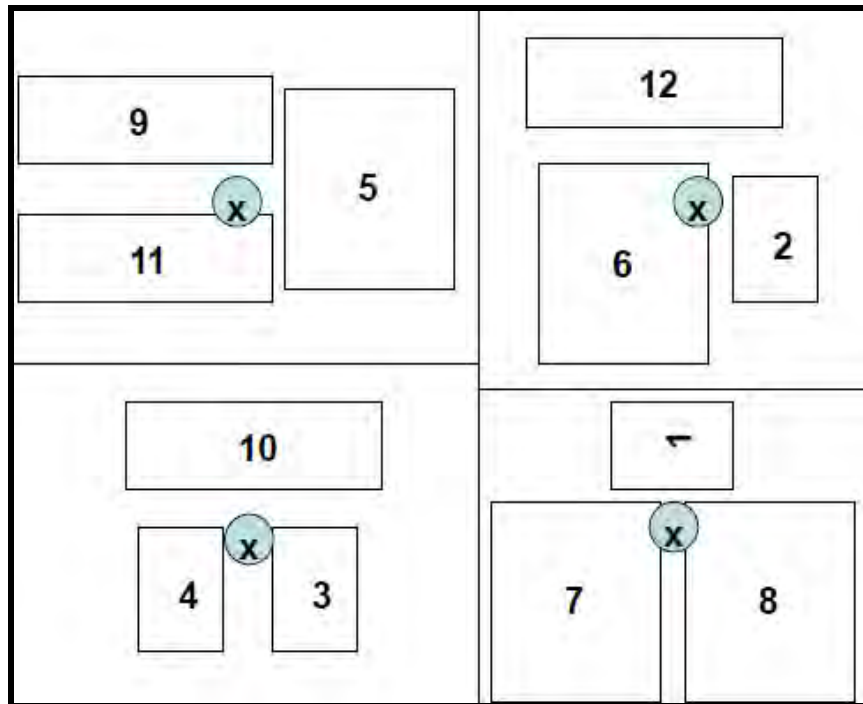
b) 1^ο partitioning και αναπροσαρμογή των θέσεων



c) 2^ο partitioning και αναπροσαρμογή των θέσεων



d) 3^ο partitioning και αναπροσαρμογή των θέσεων



e) 4^ο partitioning, αναπροσαρμογή των θέσεων και legalization

EIKONA 2.4: Gordian Placement

Στο σημείο αυτό θα διατυπώσουμε αναλυτικότερα την ακολουθία των βημάτων που έχουν αναφερθεί παραπάνω. Αρχικά, λοιπόν, χρησιμοποιώντας το clique net model σχηματίζουμε έναν υπεργράφο ο οποίος αναπαριστά, λεπτομερώς, τον τρόπο σύνδεσης των στοιχείων που απαρτίζουν το κύκλωμα. Για τον υπολογισμό του βάρους της εκάστοτε ακμής χρησιμοποιείται ο τύπος $2/K$ που έχει αναφερθεί σε προηγούμενη ενότητα. Επιπρόσθετα, τοποθετούμε τα pin της σχεδίασης στις κατάλληλες θέσεις (μια σύμβαση που ακολουθείται στα περισσότερα κυκλώματα είναι να τοποθετούμε τα input pins στην αριστερή και στην πάνω πλευρά του παραλληλογράμμου που αναπαριστά

τον χώρο που έχουμε στη διάθεσή μας και τα output pins στην κάτω και στην δεξιά πλευρά αυτού).

Ακολούθως, προχωρούμε στον υπολογισμό μιας σειράς πινάκων οι οποίοι θα μας βοηθήσουν, δυνητικά, στην μαθηματική διατύπωση του προκείμενου προβλήματος. Ο πρώτος πίνακας που υπολογίζουμε είναι ο πίνακας διασυνδεσιμότητας κελιών (adjacency matrix), ο οποίος έχει διαστάσεις $n \times n$, όπου n ο αριθμός των κελιών του κυκλώματος. Κάθε στοιχείο του πίνακα έχει την τιμή του βάρους της ακμής που συνδέει τα κελιά που προσδιορίζονται από την θέση (γραμμή, στήλη) του στοιχείου. Η μη ύπαρξη ακμής στον γράφο συνεπάγεται την τοποθέτηση μηδενικής τιμής στην, προς συμπλήρωση, θέση του πίνακα.

Ο επόμενος πίνακας που υπολογίζουμε είναι ο pin connection matrix, ο πίνακας, δηλαδή, που περιγράφει τις συνδέσεις των κελιών με τα input και output pins. Οι διαστάσεις του είναι $n \times m$ όπου m ο συνολικός αριθμός των pins. Ο τρόπος υπολογισμού της κάθε θέσης του πίνακα γίνεται με όμοιο τρόπο όπως με τον προηγούμενο πίνακα.

Στη συνέχεια, υπολογίζεται ο degree matrix. Όλα τα στοιχεία του $n \times n$ πίνακα είναι μηδενικά εκτός από τα στοιχεία της κύριας διαγωνίου. Κάθε ένα από αυτά προκύπτει προσθέτοντας όλες τις τιμές των αντίστοιχων γραμμών των πινάκων adjacency matrix και pin connection matrix.

Αφού ολοκληρώσουμε, τον παραπάνω υπολογισμό, διαμορφώνουμε τον laplacian matrix, ο οποίος χρησιμοποιείται άμεσα κατά την μορφοποίηση του μαθηματικού προβλήματος που καλούμαστε να λύσουμε. Ο συγκεκριμένος πίνακας είναι το αποτέλεσμα της αφαίρεσης του adjacency matrix από τον degree matrix.

Στο επόμενο βήμα διαμορφώνουμε τα fixed pin vectors. Τα fixed pins vectors εξαρτώνται από το pin connection matrix και την θέση των pins στην περιφέρεια της, προς χρήση, συνολικής περιοχής. Ο μαθηματικός τύπος υπολογισμού κάθε θέσης του διανύσματος είναι ο ακόλουθος: $d_{x,i} = -\sum p_{ij} \cdot x(p_j)$, όπου p_{ij} είναι η τιμή της αντίστοιχης θέσης του pin connection matrix και $x(p_j)$ η συντεταγμένη x του pin j . Με τον ίδιο τρόπο υπολογίζουμε και το αντίστοιχο διάνυσμα για την κατεύθυνση y .

Στο σημείο αυτό μπορούμε να διατυπώσουμε το quadratic problem που καλούμαστε να επιλύσουμε. Θέτουμε ως (x_i, y_i) τις συντεταγμένες του κέντρου ενός κελιού i , w_{ij} το βάρος της ακμής που ενώνει δύο κελιά i και j , και x, y , τα διανύσματα που θα αποθηκευτούν οι τελικές συντεταγμένες. Το κόστος της προκειμένης σύνδεσης δίνεται από τον ακόλουθο τύπο: $0,5 * w_{ij} * ((x_i - x_j)^2 + (y_i - y_j)^2)$, και βάσει αυτού προκύπτει η συνάρτηση που περιγράφει το συνολικό κόστος:

$$f(x, y) = 0.5 * \mathbf{x}^T * L * \mathbf{x} + d_x^T * \mathbf{x} + 0.5 * \mathbf{y}^T * L * \mathbf{y} + d_y^T * \mathbf{y} + \text{const}$$

όπου x, y είναι τα διανύσματα που εσωκλείουν τις συντεταγμένες όλων των κελιών του κυκλώματος, L ο laplacian matrix, τον τρόπο υπολογισμό του οποίου περιγράψαμε παραπάνω, d_x, d_y τα fixed pins vectors και const μια σταθερά η τιμή της οποίας υπολογίζεται δυναμικά μέσα από αλληπάλληλες πειραματικές εκτελέσεις του αλγορίθμου.

Το ζητούμενο πλέον είναι η ελαχιστοποίηση της παραπάνω συνάρτησης. Υπάρχουν, όμως, επιπλέον περιορισμοί τους οποίους πρέπει να λάβουμε υπόψη. Κατά την δημιουργία των νέων partition, πρέπει τα κέντρα των νέων περιοχών να

«συγκρατούν» εντός της υφιστάμενης περιοχής τα αντίστοιχα κελιά. Η μαθηματική διατύπωση του περιορισμού είναι η ακόλουθη:

$$\mathbf{A}^{(i)} * \mathbf{x} = \mathbf{u}^{(i)}$$

Αντίστοιχα προσδιορίζεται ο περιορισμός και για την έτερη κατεύθυνση. Στο στάδιο i , λοιπόν, της βελτιστοποίησης πρέπει το διάνυσμα των συντεταγμένων των κελιών να επιβεβαιώνει την παραπάνω εξίσωση. Το διάνυσμα \mathbf{u} αποτελείται από τις συντεταγμένες του κέντρου του υπό εξέταση partition, ενώ ο πίνακας \mathbf{A} , έχει όλα τα στοιχεία του μηδενικά εκτός από αυτά που αντιστοιχούν στην περιοχή στην οποία ανήκει το εκάστοτε κελί.

Ο Gordian χρησιμοποιεί το partitioning για να περιορίσει την ελευθερία κίνησης των κελιών, και όχι για να ελαττώσει το μέγεθος του προβλήματος. Ένας άτυπος περιορισμός για το partitioning είναι πως οι δύο νέες περιοχές, που δημιουργούνται στο εκάστοτε βήμα, πρέπει να έχουν αναλογία $1/2$ στην χειρότερη των περιπτώσεων.

Το final placement είναι, ίσως, το σημαντικότερο βήμα του Gordian. Στις standard cell σχεδιάσεις τα κελιά έχουν το ίδιο ύψος, το μήκος τους, όμως, μπορεί να διαφέρει αισθητά. Ο στόχος είναι να δημιουργήσουμε σειρές στις οποίες θα τοποθετηθούν τα κελιά, οι οποίες θα απέχουν την μικρότερη δυνατή απόσταση μεταξύ τους. Επιπρόσθετα, το μήκος συμπληρωμένων από κελιά σειρών δεν πρέπει να διαφέρει πάνω από 1-5%, για να έχουμε αισθητή βελτίωση στο τελευταίο στάδιο εφαρμογής του αλγορίθμου.

2.2.2.2 Gordian-L

Ο αλγόριθμος Gordian-L ακολουθεί την βασική στρατηγική του Gordian, τροποποιεί μερικώς, όμως, το global placement και τον τρόπο που γίνεται το partitioning. Οι βασικότερες τροποποιήσεις είναι η μετατροπή της αντικειμενικής συνάρτησης σε γραμμική και ο επαναληπτικός διαχωρισμός της εκάστοτε περιοχής σε ίσους χώρους.

Η σύγκριση των αποτελεσμάτων της εφαρμογής των δύο αλγορίθμων μας οδηγεί στο συμπέρασμα πως ο Gordian-L παρουσιάζει ελαφρώς καλύτερα αποτελέσματα ως προς την ελαχιστοποίηση της απαιτούμενης περιοχής σε κυκλώματα με μικρό αριθμό πυλών, ενώ ο Gordian παρουσιάζει καλύτερα αποτελέσματα ως προς το συνολικό μήκος καλωδίου σε κυκλώματα με μεγάλο αριθμό πυλών.

2.2.3 FORCED-BASED METHODS

Οι forced-based μέθοδοι είναι κατ'ουσίαν μια υποκατηγορία της χωροθέτησης πυλών με χρήση αναλυτικών μεθόδων. Το όνομά τους προκύπτει από την διαφορετική ερμηνεία του προβλήματος του quadratic placement. Πιο συγκεκριμένα, η ονομασία τους προκύπτει εάν θεωρήσουμε τα στοιχεία του κυκλώματος ως αντικείμενα και τα nets ως ελάσματα που τα συνδέουν. Με βάση αυτή την τροποποίηση το ζητούμενο του προβλήματος, η ελαχιστοποίηση δηλαδή του συνολικού μήκους καλωδίου, μετατρέπεται πλέον στην εύρεση του κατάλληλου τρόπου χωροθέτησης που θέτει το σύστημα που δημιουργείται σε κατάσταση ισορροπίας.

2.2.4 SIMULATED ANNEALING

Η μέθοδος του simulated annealing, παρόλο που χαρακτηρίζεται από την απλότητά της, χρησιμοποιείται σπανίως στην διαδικασία του global placement, όταν δεν έχουν παραχθεί τα επιθυμητά αποτελέσματα, κυρίως λόγω του ότι ο συνολικός χώρος λύσεων είναι υπερβολικά μεγάλος. Παρόλαυτα, χρησιμοποιείται σε ένα ευρέως διαδεδομένο placer, τον Dragon, στο στάδιο του global placement, χωρίς να δίνονται συγκεκριμένες λεπτομέρειες για τον τρόπο εφαρμογής της στην επίσημη περιγραφή του συγκεκριμένου εργαλείου.

2.2.5 CLUSTERING

Μία διαφορετική προσέγγιση πάνω στο πρόβλημα του placement, είναι η ομαδοποίηση των κελιών του κυκλώματος σε συστάδες (clustering). Το clustering παρουσιάζει δύο βασικά πλεονεκτήματα:

- I. Βελτιώνει την ταχύτητα εκτέλεσης λόγω της ελάττωσης του συνόλου αντικειμένων που καλούμαστε να χωροθετήσουμε.
- II. Βοηθάει στην διαφυγή από τοπικά ελάχιστα που μπορούν να αλλοιώσουν την ποιότητα της τελικής λύσης.

Το βασικότερο μειονέκτημα του clustering είναι ο τρόπος επιλογής των κελιών που θα ομαδοποιηθούν. Οι δύο βασικές προσεγγίσεις είναι είτε μια προεπιλογή που βασίζεται στον αρχικό υπεργράφο είτε η επιλογή βάσει ενός αρχικού placement. Σημαντικότερο παράδειγμα αυτής της κατηγορίας είναι ο αλγόριθμος Timberwolf, ο οποίος χρησιμοποιεί ένα συνδυασμό simulated annealing και clustering.

2.3 FINAL PLACEMENT

Το υποπρόβλημα του final placement περιορίζεται κατ'ουσίαν στην εύρεση του κατάλληλου τρόπου αναπροσαρμογής των θέσεων των κελιών έτσι ώστε να τοποθετηθούν στις βέλτιστες δυνατές θέσεις εξαλείφοντας τυχών φαινόμενα επικάλυψης. Στα κεφάλαια που ακολουθούν θα παρουσιαστούν οι βασικότερες προσεγγίσεις που άπτονται του προκείμενου προβλήματος.

2.3.1 SIMULATED ANNEALING

Η διαδικασία του simulated annealing εκκινεί με έναν τυχαίο αρχικό διαχωρισμό του κυκλώματος. Ακολούθως, πραγματοποιείται ένας νέος διαχωρισμός των στοιχείων του κυκλώματος ανταλλάσσοντας τις θέσεις ορισμένων κελιών των νέων περιοχών που έχουν δημιουργηθεί. Για κάθε μια από τις αλλαγές αυτές, υπολογίζεται κάποιο κόστος. Εάν το κόστος της κίνησης των κελιών είναι αρνητικό τότε η προκείμενη κίνηση γίνεται δεκτή, εάν είναι ίσο με το μηδέν ή θετικό τότε η κίνηση αυτή γίνεται δεκτή υπό προϋποθέσεις. Αυτή η διαφοροποίηση είναι και ο βασικός λόγος που το simulated annealing μπορεί να υπερβεί τοπικά ελάχιστα στην διαδικασία εύρεσης του καθολικού ελάχιστου. Στην εικόνα που ακολουθεί παρουσιάζεται μια περιγραφή σε ψευδοκώδικα της διαδικασίας του simulated annealing:

Simulated Annealing Procedure:

```
t := t0;  
cur_part := init_part;  
cur_score := SCORE( cur_part );  
repeat  
  repeat  
    comp1 := SELECT( part1 );  
    comp2 := SEIECT( part2 );  
    trial_part :=EXCHANGE( comp1, comp2, cur_part );  
    trial_score := SCORE( trial_part );  
    ds := trial_score – cur_score;  
  
    if ( ds < 0 ) then  
      cur_score := trial_score;  
      cur_part := MOVE( comp1, comp2 );  
    else  
      r := RANDOM( 0, 1 );  
      if( r < e-ds/t ) then  
        cur_score := trial_score;  
        cur_part := MOVE( comp1, comp2 );  
  
    until( equilibrium )  
  t := a*t;    //( 0 < a < 1 )  
until( t near 0 )  
end_procedure;
```

EIKONA 2.5: Simulates Annealing pseudocode

Η συνάρτηση SELECT χρησιμοποιείται για την επιλογή δύο τυχαίων κελιών, ένα από κάθε partition που σχηματίζεται. Τα κελιά αυτά είναι υποψήφια προς ανταλλαγή θέσης. Η συνάρτηση EXCHANGE που καλείται στη συνέχεια δημιουργεί δύο δοκιμαστικά partition χωρίς να εκτελεί την ανταλλαγή. Η συνάρτησης SCORE υπολογίζει το κόστος της κίνησης αυτής, και εάν το κόστος μειώνεται τότε και μόνο τότε πραγματοποιείται η ανταλλαγή κελιών μέσω της κλήσης της MOVE. Το κόστος που υπολογίζεται μπορεί να είναι είτε το cutsize του partitioning, είτε το cutsize σε συνάρτηση με άλλους παράγοντες που έχουμε θέσει προς βελτιστοποίηση.

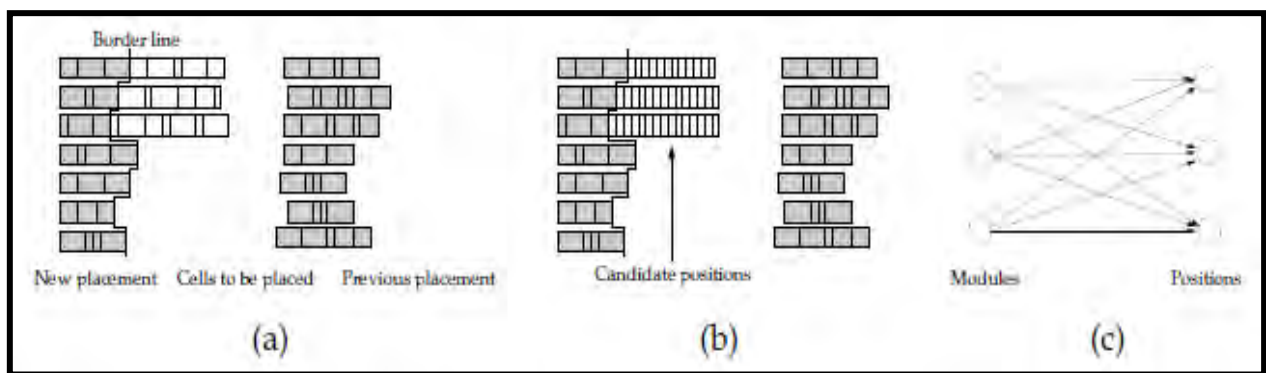
Στην περίπτωση που το τρέχων κόστος είναι μεγαλύτερο από το κόστος πριν από την επιλογή των προς ανταλλαγή κελιών, υπολογίζεται η πιθανότητα αποδοχής του συγκεκριμένου ζεύγους με χρήση της συνάρτησης RANDOM. Εάν η κίνηση γίνει δεκτή καλείται και πάλι η συνάρτηση MOVE για να εκτελέσει την ανταλλαγή.

Η ποιότητα της λύσης που παράγεται από την εφαρμογή του simulated annealing εξαρτάται από την επιλογή της μεταβλητής t που εΐθισται να αναπαριστά την θερμοκρασία και το βήμα μείωσης της προκειμένης θερμοκρασίας (την μεταβλητή a εν προκειμένω). Όσο μεγαλύτερη είναι η αρχική θερμοκρασία και όσο μικρότερος είναι ο ρυθμός ελάττωσής της, τόσο καλύτερα είναι τα αποτελέσματα που προκύπτουν. Το μειονέκτημα, όμως, είναι πως ο χρόνος που απαιτείται για τον σχηματισμό της κατάλληλης λύσης είναι ανάλογος του συνολικού αριθμού βημάτων μείωσης της θερμοκρασίας.

2.3.2 GREEDY APPROACHES

Ένας από τους πιο γνωστούς και ευρέως χρησιμοποιούμενους αλγόριθμους, που ακολουθούν μια άπληστη προσέγγιση ως προς τον τρόπο χωροθέτησης των κελιών είναι

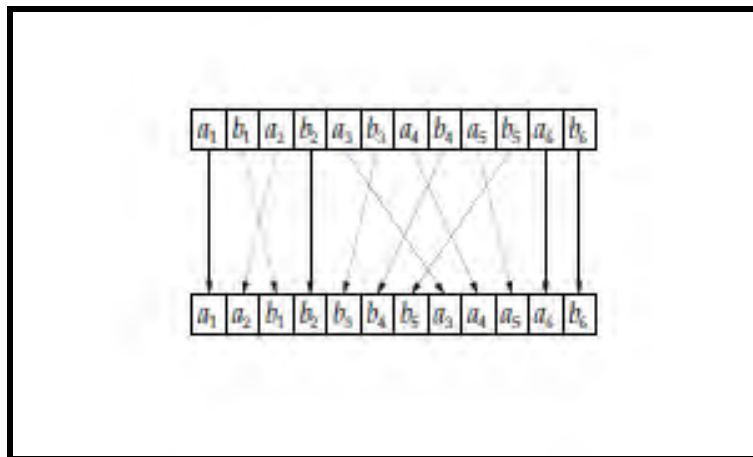
ο Domino. Ο αλγόριθμος δέχεται ως είσοδο ένα placement, δεν απαιτεί, όμως, η προκειμένη χωροθέτηση να είναι legalized. Ακολουθώς, διαχωρίζει τον συνολικό χώρο σε υποπεριοχές, βάσει συγκεκριμένων κριτηρίων, και τις εξετάζει μία προς μία. Κατά τη διάρκεια τροποποίησης της εκάστοτε περιοχής οι υπόλοιπες θεωρούνται ανενεργές. Στην επιλεγμένη περιοχή, ο Domino εκκινεί την εξέταση των κελιών από αριστερά προς τα δεξιά. Κάθε κελί διαχωρίζεται σε μια ομάδα υποκελιών, ίσου μεγέθους, για τα οποία υπολογίζονται νέες «νόμιμες» θέσεις σε κάθε μια από τις προκαθορισμένες σειρές της περιοχής. Βάσει αυτών των βημάτων διατυπώνεται, σε μαθηματική μορφή, ένα transportation problem. Το κόστος της κάθε μεταφοράς σχετίζεται άμεσα με το netlength του εκάστοτε υποκελιού. Την επίλυση του παραπάνω προβλήματος ακολουθεί η επιλογή της νέας θέσης του κελιού. Η θέση αυτή ταυτίζεται με τη θέση όπου έχουν τοποθετηθεί τα περισσότερα υποκελιά του. Το βασικό πλεονέκτημα του Domino είναι πως λόγω της επαναληπτικής διαδικασίας που ακολουθείται παρουσιάζει την ιδιότητα να αποφεύγει τοπικά ελάχιστα που μπορούν να εμφανιστούν και να αλλοιώσουν την ποιότητα της τελικής λύσης. Στην εικόνα που ακολουθεί παρουσιάζεται η χρήση του αλγορίθμου για την εύρεση του κατάλληλου placement μιας υποπεριοχής.



EIKONA 2.6: Domino Placement

Μια, ακόμη, ενδιαφέρουσα προσέγγιση είναι και ο αλγόριθμος Dragon. Ο προκείμενος αλγόριθμος αποδέχεται την global λύση ως ιδανική, οπότε αντί της χρήσης simulated annealing εφαρμόζει μια άπληστη προσέγγιση επίλυσης του προβλήματος χωροθέτησης. Για κάθε ένα από τα κελιά ερευνούνται τα γειτονικά του, έτσι ώστε να γίνει η επιλογή του καταλληλότερου προς ανταλλαγή κελιού. Το σύνολο των γειτονικών κελιών είναι σχετικά μικρό με αποτέλεσμα ο αριθμός των προς ανταλλαγή κελιών να είναι μικρός.

Τέλος, κρίνεται σκόπιμη η αναφορά στον placer Mongrel. Ο συγκεκριμένος placer ασχολείται μόνο με την μετακίνηση των κελιών εντός του εκάστοτε row. Διατηρεί την σχετική σειρά των κελιών και προσπαθεί να βρει τις κατάλληλες υποακολουθίες αυτής που μπορούν να αλλάξουν εκατέρωθεν θέσεις. Η εύρεση του ζεύγους υποακολουθιών έχει αποδειχθεί πως μπορεί να καταστεί δυνατή με χρήση δυναμικού προγραμματισμού εντός πολυωνυμικού χρόνου, αναλόγου με τον συνολικό αριθμό των pins των προς εξέταση κελιών. Στην εικόνα που ακολουθεί παρουσιάζεται η βελτίωση που εμφανίζεται μέσω της εφαρμογής του αλγορίθμου.



EIKONA 2.7: Mongrel Algorithm Instance

2.4. LEGALIZATION

Οι αλγόριθμοι που παρουσιάστηκαν στην παραπάνω ενότητα δεν δίνουν πάντοτε στην έξοδό τους ένα legal placement. Γι' αυτό και στην συνολική διαδικασία χωροθέτησης ενός κυκλώματος εισάγεται και ένα τελικό βήμα «νομιμοποίησης» της σχεδίασης.

Στην περίπτωση του standard-cell placement η διαδικασία που ακολουθείται, στην πλειοψηφία των περιπτώσεων, είναι να γίνεται ένας αρχικός διαχωρισμός των κελιών ως προς τις συντεταγμένες x και y , και ακολούθως να τοποθετούνται στις σειρές που έχουν προκαθοριστεί χρησιμοποιώντας την συντεταγμένη y για την επιλογή της κατάλληλης σειράς και την συντεταγμένη x για την θέση τους εντός της προκείμενης σειράς.

ΚΕΦΑΛΑΙΟ 3

ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ

Στο κεφάλαιο που ακολουθεί θα παρουσιαστούν τα βασικότερα χαρακτηριστικά και οι σημαντικότερες ιδιότητες του κώδικα που υλοποιήθηκε προκειμένου να αντιμετωπιστούν κάποια από τα σπουδαιότερα προβλήματα που παρατηρούνται στις σύγχρονες προσεγγίσεις του προβλήματος χωροθέτησης ολοκληρωμένων κυκλωμάτων. Ακολούθως, θα παρουσιαστούν, λεπτομερώς, οι προσεγγίσεις μας πάνω σε θέματα που άπτονται του placement, όπως η μείωση του συνολικού απαιτούμενου μήκους καλωδίου διασύνδεσης των στοιχείων του κυκλώματος, η ελαχιστοποίηση του καταλαμβανόμενου, από τα κελιά του κυκλώματος, χώρου και η αναδιάταξη του critical path ενός placed κυκλώματος βάσει της ανάλυσης φαινομένων πτώσης τάσης που παρατηρούνται.

3.1. ΠΡΟΠΑΡΑΣΚΕΥΑΣΤΙΚΟ ΣΤΑΔΙΟ - ΔΟΜΕΣ ΑΠΟΘΗΚΕΥΣΗΣ ΔΕΔΟΜΕΝΩΝ

Στις παρακάτω ενότητες θα γίνει αναλυτική περιγραφή των αρχείων που δέχεται ως είσοδο η πλατφόρμα που αναπτύξαμε, της εξόδου που παράγει και των βιομηχανικών εργαλείων που χρησιμοποιήθηκαν προκειμένου να εντάσσεται η

προκείμενη διαδικασία στην πλειοψηφία των ακολουθούμενων ροών σχεδίασης ολοκληρωμένων κυκλωμάτων.

Ο αλγόριθμος υλοποιήθηκε χρησιμοποιώντας την γλώσσα προγραμματισμού C. Η επιλογή της συγκεκριμένης γλώσσας βασίστηκε στα δύο βασικά χαρακτηριστικά της την μεταφερσιμότητα που παρουσιάζει ο κώδικας (την δυνατότητα να εκτελεστεί σε οποιοδήποτε λειτουργικό σύστημα) και την παρεχόμενη στο χρήστη δυνατότητα δυναμικής διαχείρισης της απαιτούμενης μνήμης.

3.1.1. ΕΙΣΟΔΟΣ

Η είσοδος του αλγορίθμου καθορίζεται από την επιλογή της λειτουργίας που επιθυμούμε να διατελέσει. Στην ενότητα αυτή θα γίνει εκτενής αναφορά σε όλες τις πιθανές εισόδους:

- VHDL netlist: Η περιγραφή της σχεδίασης που καλούμαστε να χωροθετήσουμε δίνεται σε αρχεία VHDL. Η αρχική περιγραφή, όμως, είναι σε επίπεδο πυλών, και προκειμένου να λάβουμε όλα τα απαραίτητα για την διαδικασία του placement δεδομένα, πρέπει να «διαβάσουμε» μια περιγραφή σε επίπεδο κελιών. Γι' αυτό το λόγο η αρχική σχεδίαση υφίσταται την διαδικασία της σύνθεσης, μέσω της χρήσης κατάλληλων EDA εργαλείων. Το λογισμικό που επιλέξαμε, όπως και ο τρόπος που πραγματοποιείται η synthesis των κυκλωμάτων θα αναφερθούν σε επόμενη ενότητα. Όταν, λοιπόν, ολοκληρωθεί η παραπάνω διαδικασία το αρχείο .vhd που προκύπτει υφίσταται το κατάλληλο parsing έτσι

ώστε να αποθηκευτούν πληροφορίες που θα χρησιμοποιηθούν κατά την εκτέλεση του αλγορίθμου. Πιο συγκεκριμένα, αποθηκεύεται ο τρόπος σύνδεσης των κελιών, η διασυνδεσιμότητά τους δηλαδή, τα nets που σχηματίζονται όπως και η διαδρομή του κυκλώματος που παρουσιάζει την μέγιστη καθυστέρηση (critical path).

- Technology Library: Η τεχνολογική βιβλιοθήκη που χρησιμοποιούμε είναι απαραίτητη για την εξαγωγή επιπλέον στοιχείων και ιδιοτήτων των κελιών που απαρτίζουν την σχεδίαση. Από την προκείμενη βιβλιοθήκη αποθηκεύουμε στοιχεία όπως το πλάτος των κελιών, τα pins που διαθέτουν, το fanin και το fanout τους όπως και επιπρόσθετες πληροφορίες που αφορούν τον χρονισμό τους και την ηλεκτρική συμπεριφορά τους. Η βιβλιοθήκη που χρησιμοποιήθηκε στην παρούσα διατριβή είναι η Nangate 45nm Open Cell Library, η οποία είναι open source και διανέμεται δωρεάν για ερευνητικούς σκοπούς.
- Initial Cell Placement Seed: Στην περίπτωση που θέλουμε να μελετήσουμε την συμπεριφορά του αλγορίθμου σε πολλαπλά του στιγμιότυπα βάσει της αρχικής τοποθέτησης των κελιών, δίνεται η δυνατότητα δημιουργίας ενός πλήθους ψευδοτυχαίων αρχικών placement που θα αποτελέσουν την αρχική κατάσταση του προς επίλυση προβλήματος.
- Initial Pin Placement Seed: Οι θέσεις των input και output pins του κυκλώματος μπορούν να επηρεάσουν αισθητά την ποιότητα της τελικής λύσης του προβλήματος του placement. Δίνεται, λοιπόν, η δυνατότητα στον χρήστη να

μελετήσει την διακύμανση των αποτελεσμάτων αυτών δημιουργώντας ψευδοτυχαίες αρχικές θέσεις των I/O ports της σχεδίασης.

- Algorithm Selection: Παρέχεται στον χρήστη η δυνατότητα επιλογής του αλγορίθμου επίλυσης του προβλήματος που θα χρησιμοποιηθεί. Οι δυνατές επιλογές είναι, είτε η αντιμετώπιση του quadratic problem ως ενός μαθηματικού τρόπου υλοποίησης ενός global placement, είτε η μορφοποίησή του σε μια αναδρομική διαδικασία βέλτιστης τοποθέτησης των κελιών που απαρτίζουν την σχεδίαση σε υποχώρους του συνολικού διαθέσιμου χώρου που σχηματίζονται με χρήση του κατάλληλου partitioner. Επιπρόσθετα, δίνεται η δυνατότητα κλήσης του Matlab για την επίλυση του σχηματιζόμενου συστήματος εξισώσεων.
- DEF file: Ο αλγόριθμος μπορεί να λάβει ως είσοδο ένα .def αρχείο το οποίο θα περιγράφει λεπτομερώς τις θέσεις των κελιών και των pins της σχεδίασης, από το οποίο θα αντληθούν οι κατάλληλες πληροφορίες για την δημιουργία μιας re-placed σχεδίασης βάσει κριτηρίων που θα προεπιλεγθούν από τον χρήστη. Τα .def files που χρησιμοποιήσαμε στην προκείμενη περίπτωση παράχθησαν από γνωστά και ευρέως χρησιμοποιούμενα βιομηχανικά εργαλεία αυτοματοποιημένης χωροθέτησης.

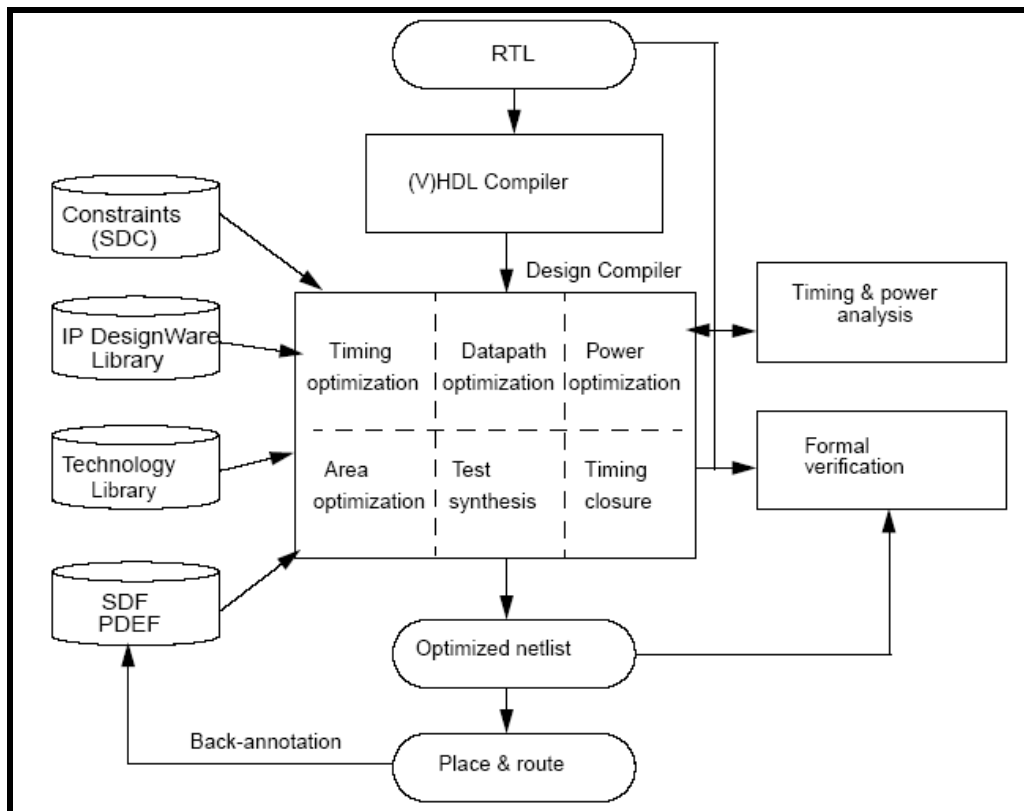
3.1.2. ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΑ ΕΡΓΑΛΕΙΑ

Τα βασικά εργαλεία που χρησιμοποιήθηκαν για την υποβοήθηση της προκείμενης διατριβής, ήταν ο Design Compiler της Synopsys και ένα σύνολο placement tools τα οποία χρησιμοποιούνται κατά κόρον στις σύγχρονες ροές σχεδίασης ψηφιακών κυκλωμάτων, και θα παρουσιαστούν αναλυτικά στις ενότητες που ακολουθούν. Επιπρόσθετα, χρησιμοποιήθηκε ο Library Compiler της Synopsys για την μετατροπή της τεχνολογικής βιβλιοθήκης σε μορφή κατάλληλη ώστε να μπορεί να δοθεί ως είσοδος στα εργαλεία της Synopsys και το Matlab προς επαλήθευση των αποτελεσμάτων των μαθηματικών συναρτήσεων επίλυσης του quadratic problem που υλοποιήσαμε. Η συμμετοχή τους, όμως, στην υλοποίηση του τελικού αποτελέσματος είναι επιδερμική και γι' αυτό δεν κρίνεται σκόπιμη η περαιτέρω ανάλυση του τρόπου λειτουργίας τους.

3.1.2.1. Design Compiler

Ο Design Compiler (DC) της Synopsys είναι ένα από τα συχνότερα χρησιμοποιούμενα λογισμικά σύνθεσης ψηφιακών κυκλωμάτων. Ο DC βελτιστοποιεί τις σχεδιάσεις με απώτερο στόχο την παροχή μικρότερων και γρηγορότερων αναπαραστάσεων μιας λογικής συνάρτησης. Περιέχει υποεργαλεία τα οποία συνθέτουν τις HDL σχεδιάσεις σε τεχνολογικά εξαρτημένα κυκλώματα σε επίπεδο κελιών, βελτιώνοντας την ταχύτητα απόκρισής τους, τον χώρο που καταλαμβάνουν και την ισχύ που καταναλώνουν.

Στην παρακάτω εικόνα παρουσιάζεται ο τρόπος με τον οποίο ο DC συμβάλλει στη συνολική ροή σχεδίασης ενός ολοκληρωμένου κυκλώματος.



EIKONA 3.1: Design Compiler's participation in a Typical Design Flow

Τα βήματα που ακολουθούνται κατά την διαδικασία της σύνθεσης ενός ψηφιακού κυκλώματος, σε άμεση αντιστοιχία με το παραπάνω σχήμα, είναι τα εξής:

- 1) Ο DC δέχεται ως είσοδο, αρχεία τα οποία περιγράφουν το κύκλωμα σε επίπεδο καταχωρητών (RTL), δηλαδή σε κάποια γλώσσα περιγραφής υλικού, όπως η Verilog και η VHDL.
- 2) Κατά την διάρκεια της σύνθεσης, μεταφράζει την HDL περιγραφή σε συνθετικά στοιχεία της DesignWare Library (τα οποία στο εξής θα αναφέρουμε ως genetic Boolean components – GTECH components). Τα

GTECH components δεν εμπεριέχουν πληροφορίες όσον αφορά τον χρονισμό και την δυνατότητα οδήγησης και δεν ανταποκρίνονται σε κάποια πραγματική τεχνολογική βιβλιοθήκη. Ο DC χρησιμοποιεί τεχνολογικές (technology), συνθετικές (synthetic) και συμβολικές (symbolic) βιβλιοθήκες από το σημείο αυτό και μετέπειτα για την υλοποίηση του netlist και την παρουσίασή του ως γραφική αναπαράσταση.

- 3) Αφού ολοκληρωθεί η μετατροπή του HDL κώδικα σε περιγραφή σε επίπεδο πυλών, ο DC βελτιστοποιεί την σχεδίαση και αντιστοιχίζει τα στοιχεία που την αποτελούν σε ένα σύνολο από πλήρως καθορισμένα κελιά μιας τεχνολογικής βιβλιοθήκης, βασιζόμενος στις επιλογές του χρήστη και τους περιορισμούς που αυτός έχει εισάγει. Οι περιορισμοί αποτελούν στην ουσία τις σχεδιαστικές απαιτήσεις του χρήστη για τους στόχους που καλείται να εκπληρώσει, αναφέρονται, δηλαδή, στους χωρικούς και χρονικούς περιορισμούς της τελικής μορφής της σχεδίασης.
- 4) Ακολούθως, ελέγχεται το αποτέλεσμα της σύνθεσης προκειμένου να επιβεβαιωθεί πως βρίσκεται εντός των σχεδιαστικών στόχων, έτσι ώστε στην περίπτωση εμφάνισης κάποιου προβλήματος να υπάρχει η δυνατότητα επίλυσης του σε όσο το δυνατό πιο πρώιμο στάδιο του κύκλου σχεδίασης.
- 5) Μετά το πέρας του παραπάνω σταδίου, είναι πλέον εφικτό το placement της σχεδίασης. Ο σχεδιαστής στο σημείο αυτό, έχει την δυνατότητα να υποσημειώσει επιπρόσθετα χαρακτηριστικά των κελιών της σχεδίασης, όπως οι καθυστερήσεις που παρουσιάζονται στις εσωτερικές διασυνδέσεις (interconnection delays) και να επανατροφοδοτήσει τον DC με νέα

δεδομένα εισόδου, προκειμένου να γίνει μια πιο ακριβής ανάλυση του χρονισμού του κυκλώματος.

Στο σημείο αυτό πρέπει να αναφέρουμε πως τα αρχεία εξόδου που παράγονται σε κάθε ένα από τα παραπάνω στάδια, είναι σε κατάλληλη μορφή τεχνολογικών standards τα οποία προσφέρονται για άμεση διασύνδεση με άλλα EDA εργαλεία.

Στις υποενότητες που ακολουθούν θα παρουσιαστεί το Design Vision, ένα εργαλείο της Synopsys που αποτελεί υποσύνολο του DC, και χρησιμοποιήθηκε για την σύνθεση των εμπλεκόμενων με την διατριβή κυκλωμάτων και οι DesignWare Libraries που αυτό χρησιμοποιεί και αποτελούν ένα ενδιάμεσο στάδιο αφαιρετικής περιγραφής των δομικών μονάδων του εκάστοτε κυκλώματος.

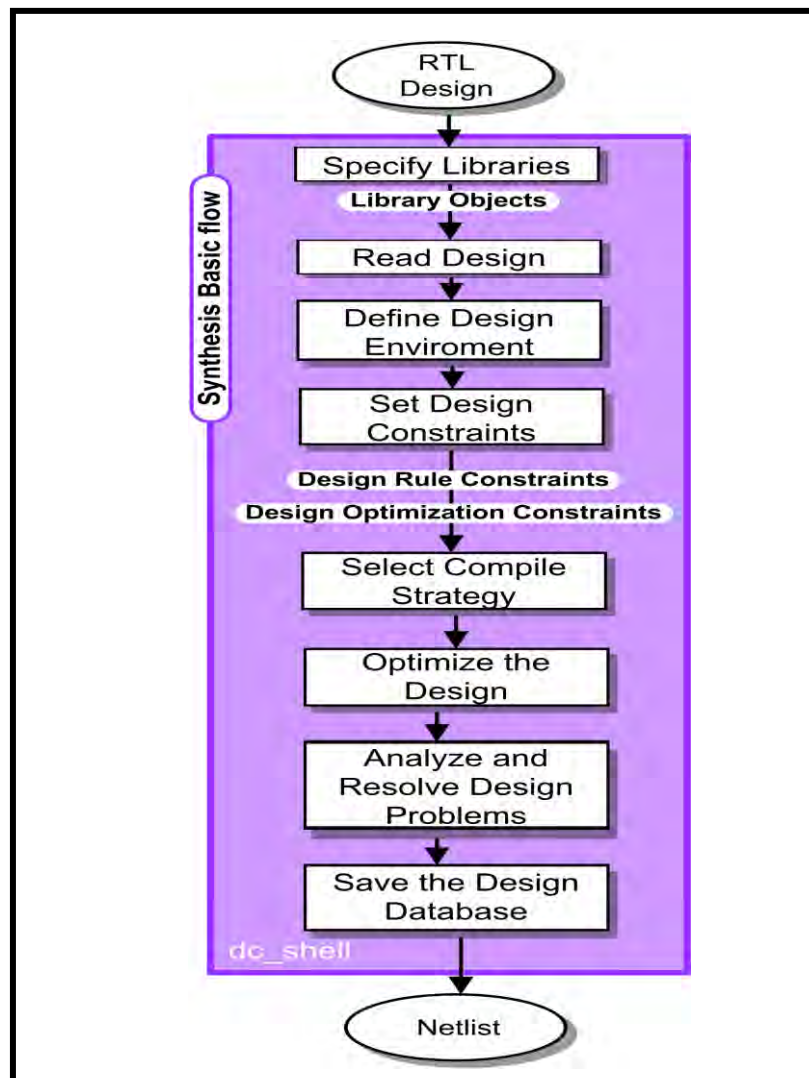
Design Vision:

Το Design Vision είναι μια γραφική διεπαφή για το περιβάλλον σύνθεσης της Synopsys και παράλληλα ένα εργαλείο το οποίο μπορεί να χρησιμοποιηθεί για την ανάλυση της προς εξέταση σχεδίασης και την μετατροπή της σε ένα netlist σε επίπεδο πυλών εξαρτημένο από τα modules της GTECH βιβλιοθήκης.

Το γραφικό περιβάλλον του Design Vision περιέχει μενού επιλογών τα οποία μας δίνουν την δυνατότητα να εκτελέσουμε το μεγαλύτερο ποσοστό των εντολών που μπορούν να εκτελεστούν μέσα από το dc_shell. Με άλλα λόγια, το προκείμενο εργαλείο αποτελεί μια μετεξέλιξη του Design Analyzer, το οποίο αποτελεί μια επιπρόσθετη γραφική διεπαφή επικοινωνίας με τον πυρήνα των εργαλείων της Synopsys.

Στο σημείο αυτό πρέπει να αναφέρουμε πως όλες οι δυνατότητες του DC μπορούν να χρησιμοποιηθούν μέσα από την κονσόλα που μας προσφέρει το Design Vision, και αυτό συμβαίνει διότι τόσο το Design Vision όσο και ο Design Compiler χρησιμοποιούν την ίδια «μηχανή» στατικής ανάλυσης χρονισμού.

Η μεθοδολογία που ακολουθήθηκε είναι παραπλήσια με αυτή που έχει περιγραφεί παραπάνω και παρουσιάζεται στην ακόλουθη εικόνα:



EIKONA 3.2: Synopsys Sythesis Flow

Ακολούθως, θα αναλύσουμε βήμα-βήμα την διαδικασία σύνθεσης που εφαρμόσαμε και βασίζεται στο generic synthesis script που παρουσιάζεται στην εικόνα που ακολουθεί.

```
set search_path { /path }
set target_library { /path/library.db }
set link_library { /path/library.db }
analyze -f vhdl { /path/circuit_behavioral.vhd }
elaborate circuit_behavioral -library WORK
current_design circuit
set_fix_multiple_port_nets -all -feedthroughs -outputs
set_max_delay 0 [all_outputs]
set_max_transition 0.1 [all_inputs]
compile -map_effort high -ungroup_all -exact_map
write -format verilog -hierarchy -output b01_netlist.v
write_parasitics -output b01_parasitics.mapped.spef
write_sdc -nosplit b01.mapped.sdc
```

EIKONA 3.3: Generic Synthesis Script

Οι εντολές, με την σειρά που αναγράφονται, αποτελούν ένα τυπικό παράδειγμα ενός script το οποίο «καθοδηγεί» το Design Vision στην σύνθεση ενός ψηφιακού κυκλώματος.

Με την πρώτη εντολή καθορίζεται η θέση στο δίσκο στην οποία είναι αποθηκευμένη η περιγραφή του κυκλώματος. Οι δύο επόμενες εντολές (set link_library

και `set target_library`), προσδιορίζουν την θέση όπου βρίσκονται οι τεχνολογικές βιβλιοθήκες βάσει των οποίων θα δημιουργηθεί το τεχνολογικά εξαρτημένο netlist.

Η εντολή `analyze` εξετάζει το HDL αρχείο, προκειμένου να διαπιστώσει εάν βρίσκεται σε ορθή συντακτική και λογική μορφή. Επιπρόσθετα, μεταφράζει τα αρχεία σε ένα ενδιάμεσο `format` και τα τοποθετεί στον φάκελο που έχουμε ορίσει ως φάκελο εργασίας.

Η εντολή `elaborate` εξετάζει το ενδιάμεσο αρχείο που έχει δημιουργηθεί από την εκτέλεση της προηγούμενης εντολής και καθορίζει ποια από τα στοιχεία της σχεδίασης πρέπει να αντικατασταθούν από συνθετικά `modules` της βιβλιοθήκης που έχει επιλεγεί.

Η ακόλουθη εντολή, `set_fix_multiple_port_nets` χρησιμοποιείται στην υποπερίπτωση που η είσοδος στον DC είναι μια κυκλωματική περιγραφή σε Verilog, και αποθαρρύνει την εισαγωγή εντολών `assign` στο παραγόμενο netlist.

Στις εντολές που εισάγονται, στη συνέχεια, καθορίζονται οι περιορισμοί και οι παράμετροι χρονισμού της σχεδίασης. Πιο συγκεκριμένα, δίνονται οι περιορισμοί για τον συνολικό χρόνο απόκρισης του κυκλώματος και το `max transition time` για όλα τα `inputs` του κυκλώματος. Ο καθορισμός αυτών των δεδομένων επηρεάζει άμεσα την σύνθεση της σχεδίασης και τα αποτελέσματα των βελτιστοποιήσεων που εφαρμόζει ο DC.

Η εντολή `compile` δημιουργεί το netlist βάσει των επιλογών που έχουν παρθεί σε προηγούμενα βήματα και η `write` αποθηκεύει το προκείμενο netlist σε ένα αρχείο μορφοποιημένο σύμφωνα με τη γλώσσα περιγραφής υλικού που έχει οριστεί.

Στην συνέχεια με την εντολή `write_parasitics` αποθηκεύονται πληροφορίες για τα παρασιτικά που παρατηρούνται στο κύκλωμα, σε ένα αρχείο `.spcf`, έτσι ώστε να είναι δυνατός σε επόμενα στάδια του σχεδιασμού ο ορθός υπολογισμός της καταναλισκόμενης ισχύος.

Τέλος, η εντολή `write_sdc` δημιουργεί ένα αρχείο που εμπεριέχει όλους τους σχεδιαστικούς περιορισμούς του κυκλώματος και δύναται να χρησιμοποιηθεί κατά το στάδιο του placement για την περαιτέρω βελτίωση του τελικού αποτελέσματος.

DesignWare Libraries:

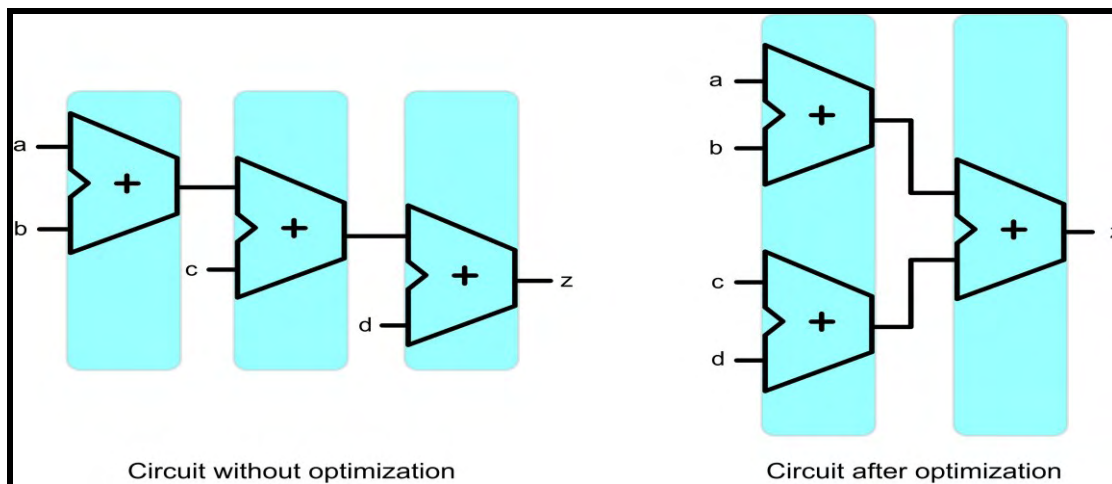
Οι DesignWare βιβλιοθήκες περιέχουν components τα οποία είναι τεχνολογικά ανεξάρτητα, αποτελούν δηλαδή, δομικά blocks σε επίπεδο μικροαρχιτεκτονικής τα οποία είναι πλήρως ενοποιημένα με το περιβάλλον σύνθεσης της Synopsys. Οι σημαντικότερες βιβλιοθήκες της προκειμένης κατηγορίας είναι η Foundation Library και η Digital Signal Processing Library (DSP). Στην παρούσα διατριβή χρησιμοποιήθηκαν κατά κόρον στοιχεία που είναι διαθέσιμα στην πρώτη από αυτές γι' αυτό και θα γίνει εκτενής αναφορά στις ιδιότητές της.

Η Foundation Library παρέχει την δυνατότητα εκτέλεσης βελτιστοποιήσεων υψηλού επιπέδου, μέσω της χρήσης κατάλληλων εργαλείων σύνθεσης. Ως παράδειγμα, μπορούμε να αναφέρουμε την περίπτωση κατά την οποία μέσα σε ένα αρχείο περιγραφής ενός κυκλώματος εντοπιστεί ο τελεστής πρόσθεσης «+». Στην περίπτωση, λοιπόν, αυτή ο HDL compiler, ο οποίος αποτελεί ένα υποεργαλείο του Design Compiler και καλείται σε κάθε εκτέλεση της εντολής `compile`, «αποφασίζει» πως ο τελεστής που διαβάστηκε περιγράφει κατ' ουσίαν έναν αθροιστή. Παραθέτει, ακολούθως, μια περιληπτική αναπαράσταση της πράξης της πρόσθεσης στο netlist του κυκλώματος, χρησιμοποιώντας ένα synthetic operator, τον οποίο και επεξεργάζεται, ανακαλώντας παράλληλα τους υψηλού επιπέδου αλγορίθμους βελτιστοποίησης που εμπεριέχει.

Η συνεισφορά των προκειμένων αλγορίθμων έγκειται, κυρίως, στους τομείς της αριθμητικής βελτιστοποίησης και στην βελτιστοποίηση του καταμερισμού πόρων (resource sharing). Στις περιπτώσεις αριθμητικής βελτιστοποίησης χρησιμοποιούνται κανόνες της άλγεβρας, η εφαρμογή των οποίων, οδηγεί στην ελάττωση του μεγέθους της

σχεδίασης και την αναθεώρηση της απόδοσής της, βάσει του επαναπροσδιορισμού της θέσης των πράξεων.

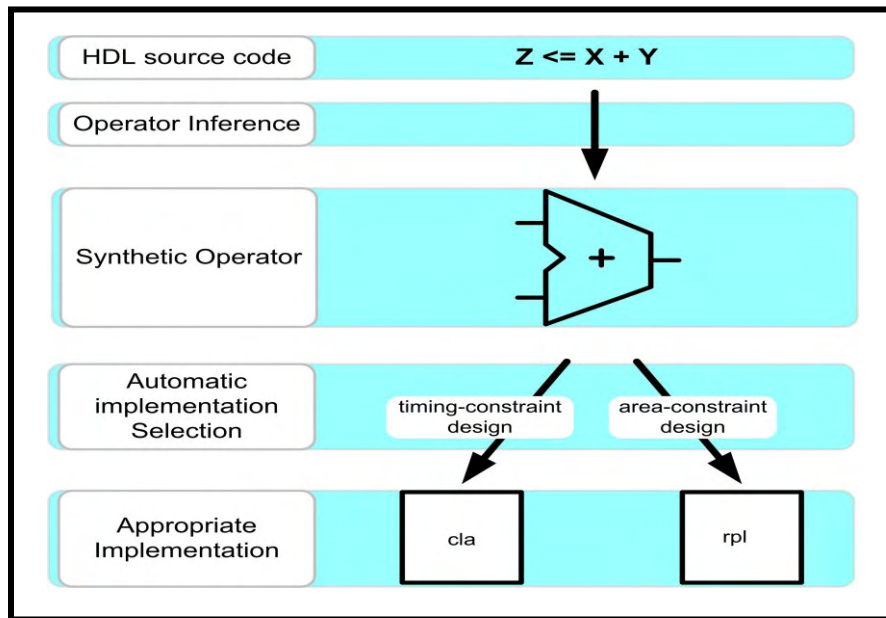
Στην εικόνα που ακολουθεί δίνεται ένα χαρακτηριστικό παράδειγμα. Η μαθηματική πράξη $a+b+c+d$ περιγράφει τρία επίπεδα, διαδοχικών πράξεων πρόσθεσης, όπου οι μεταβλητές προστίθενται σε κάθε στάδιο ανά ζεύγη. Η χρήση της Foundation Library οδηγεί στην αναδιάταξη των παραπάνω πράξεων. Ο νέος τρόπος αναπαράστασης του συνολικού αθροίσματος ενδέχεται να προσφέρει μεγαλύτερη ταχύτητα απόκρισης στην σχεδίασή μας λόγω της ελάττωσης των χρησιμοποιούμενων επιπέδων λογικής. Επιπρόσθετα, ο καταμερισμός πόρων επιτρέπει σε όμοιες λειτουργίες, οι οποίες δεν επικαλύπτονται χρονικά, να εκτελούνται από το ίδιο hardware.



ΕΙΚΟΝΑ 3.4: Foundation Library Arithmetic Optimization

Με την χρήση της Foundation Library, μια πράξη που δίνεται από τον χρήστη, μέσω κάποιου αρχείου εισόδου στον DC, δύναται να υλοποιηθεί με πολλούς τρόπους. Στο σημείο αυτό επεμβαίνει ο Design Compiler επιλέγοντας την καταλληλότερη, ανά περίπτωση, υλοποίηση. Ένα χαρακτηριστικό παράδειγμα είναι η

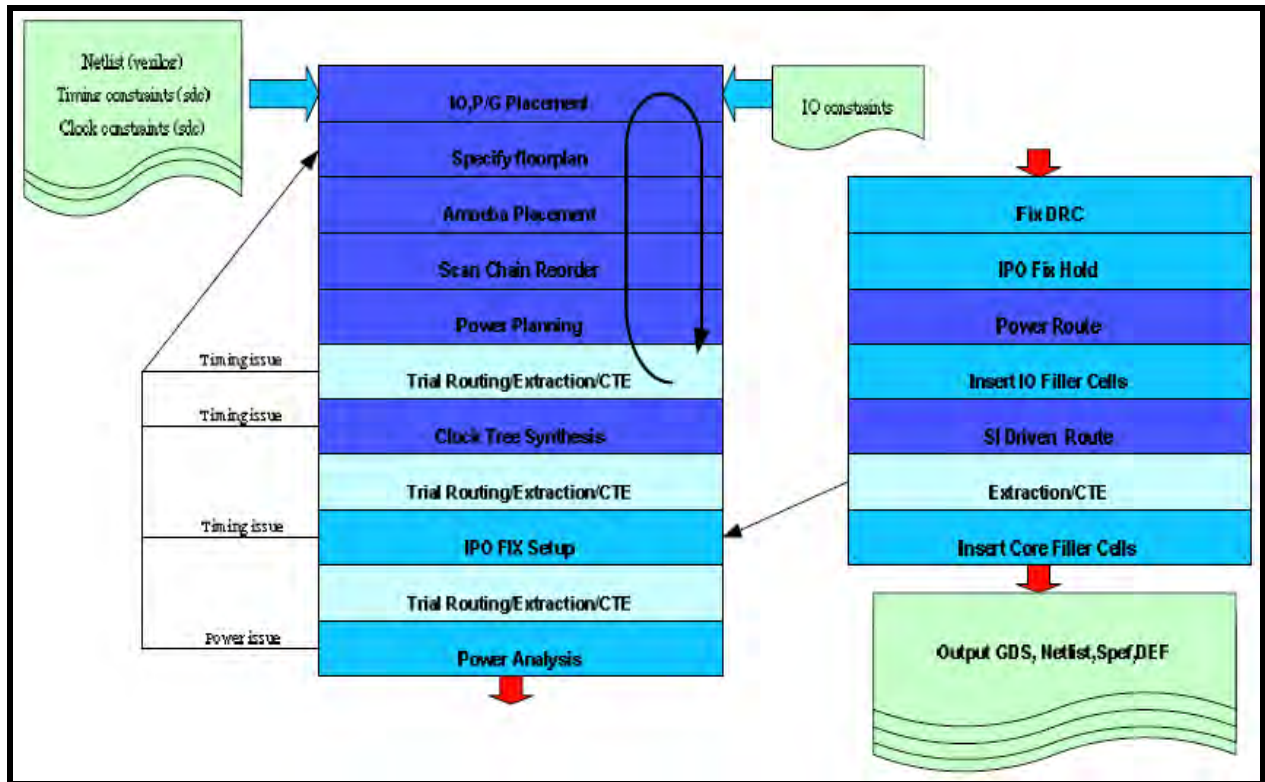
μη προσημασμένη πρόσθεση, η οποία υλοποιείται είτε με την τοπολογία ενός carry lookahead adder είτε με την τοπολογία ενός carry ripple adder, και παρουσιάζεται στην εικόνα που ακολουθεί. Η επιλογή ανάμεσα στις δύο αυτές δυνατές περιπτώσεις θα γίνει βάσει των παραμέτρων που έχουν δοθεί σε προηγούμενα στάδια της ροής σχεδιασμού.



EIKONA 3.5: Foundation Library Adder Implementation

3.1.2.2. Placement CAD Tools

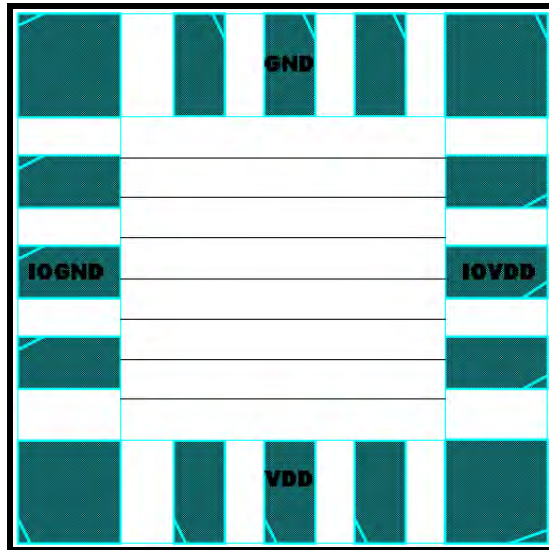
Στην προκείμενη διατριβή χρησιμοποιήθηκαν διάφορα placement tools τόσο για την σύγκριση των αποτελεσμάτων τους με τα αποτελέσματα του δικού μας αλγορίθμου, όσο και για την παροχή μιας pre-placed σχεδίασης, ως είσοδο, στο αλγοριθμικό κομμάτι του ir-drop based re-placement που υλοποιήσαμε. Στην παρακάτω εικόνα παρουσιάζεται μια εκτενής ροή p&r ενός ολοκληρωμένου κυκλώματος, θεμελιώδη βήματα της οποίας αναλύονται στη συνέχεια.



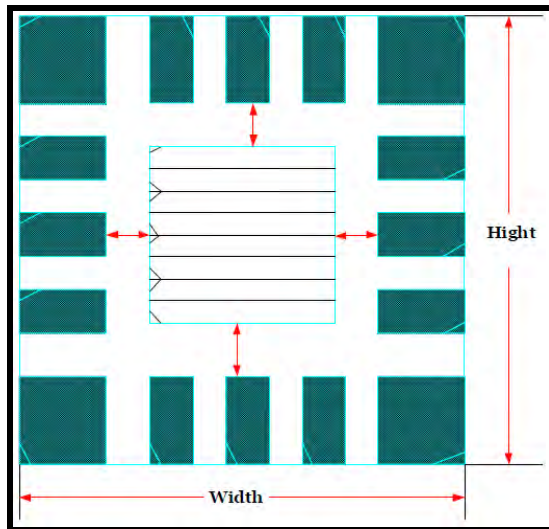
ΕΙΚΟΝΑ 3.6: Analytical P&R Flow

Τα βιομηχανικά placement tools δέχονται ως είσοδο ένα netlist σε επίπεδο κελιών και ένα σύνολο αρχείων τα οποία περιγράφουν τους χρονικούς περιορισμούς βάσει των οποίων πρέπει να εκτελεστεί η χωροθέτηση του κυκλώματος. Ακολούθως, ο χρήστης ορίζει τις θέσεις των pads, τα οποία χωρίζονται σε τρεις κατηγορίες: I/O pads, PWR/GND pads και Corner pads, με τα τελευταία να χρησιμοποιούνται αποκλειστικά για την διασύνδεση με τα power rings.

Στο επόμενο βήμα καθορίζεται το floorplan, παρέχεται δηλαδή η χωρική αναλογία μεταξύ του πυρήνα της σχεδίασης και της κενής περιοχής μεταξύ των pads και αυτού. Επιπρόσθετα, καθορίζονται οι «σχετικές» θέσεις τυχών hard blocks που μπορεί να εμπεριέχονται στη σχεδίαση.



ΕΙΚΟΝΑ 3.7: I/O, PWR/GND, Corner PAD placement

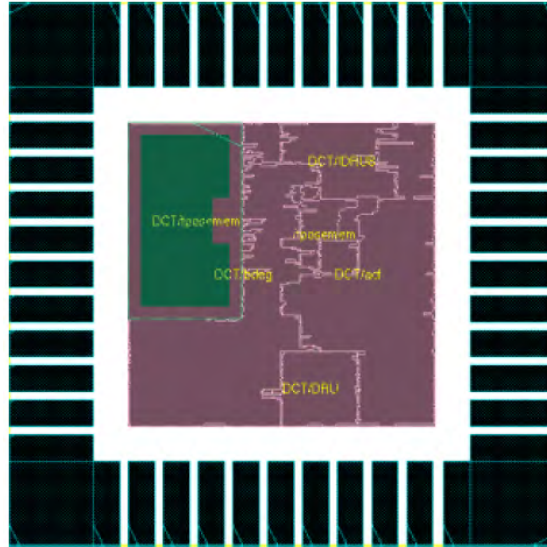


ΕΙΚΟΝΑ 3.8: Floorplanning

Ακολουθεί το Amoeba placement όπου πραγματοποιείται μια πρώτη χωροθέτηση όλων των modules της σχεδίασης.

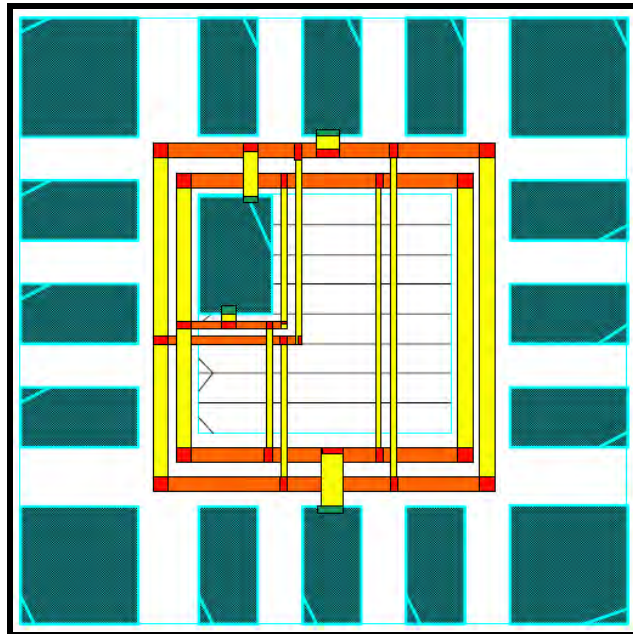
Το πέρας της παραπάνω διαδικασίας σηματοδοτεί την έναρξη του power planning. Κατά την εκτέλεση αυτό του βήματος, χωροθετούνται τα power rings και τα

power stripes λαμβάνοντας υπόψη τα φαινόμενα πτώσης τάσης που θα εμφανιστούν και μπορούν να αλλοιώσουν την απόδοση του κυκλώματος.

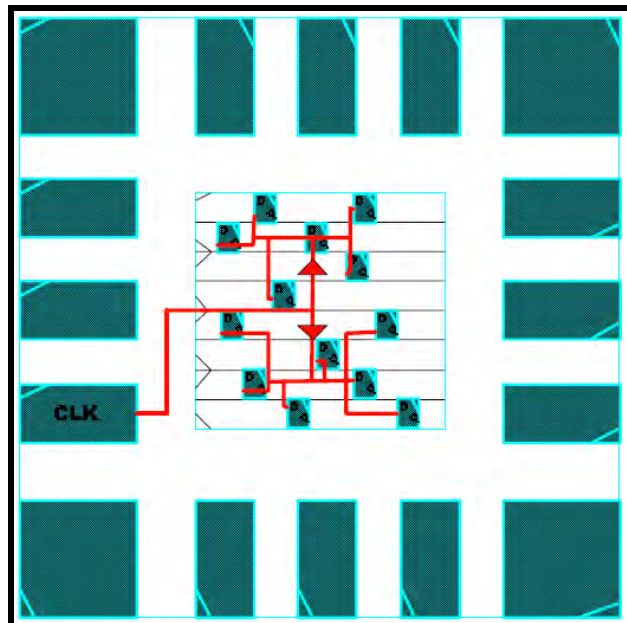


ΕΙΚΟΝΑ 3.9: Amoeba placement

Το στάδιο του power planning διαδέχεται η διαδικασία του Clock Tree Synthesis (CTS). Στόχος του CTS είναι η ελαχιστοποίηση του skew της σχεδίασης και του insertion delay.

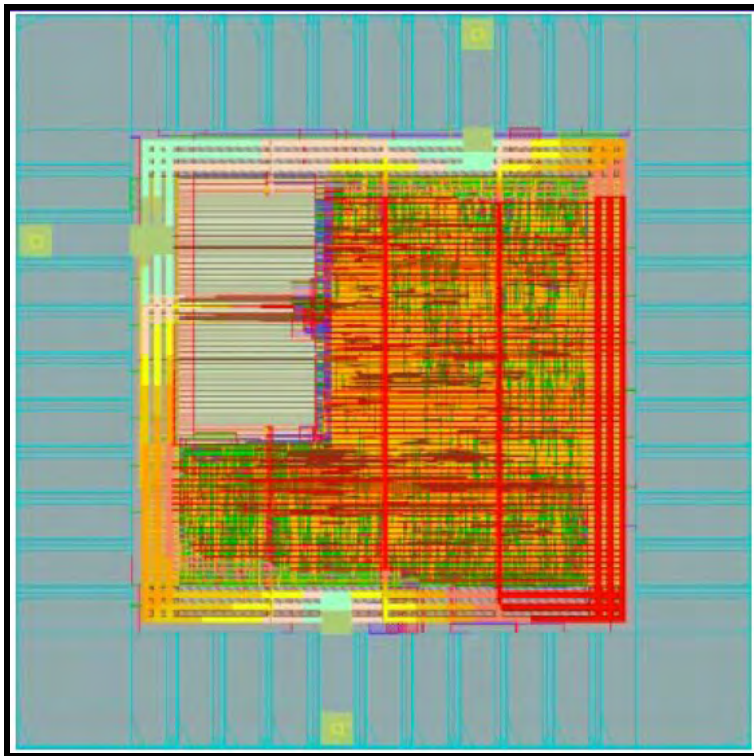


EIKONA 3.10: Power Planning



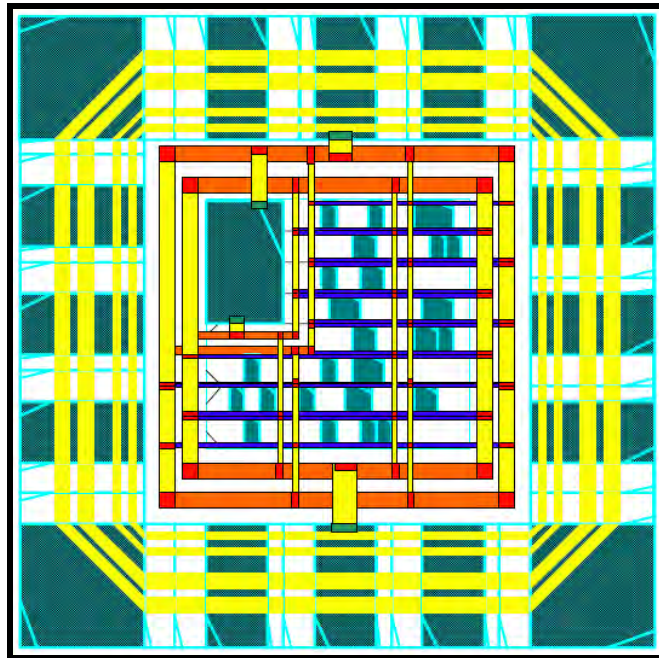
EIKONA 3.11: Clock Tree Sythesis

Τα δύο επόμενα βήματα του p&r flow είναι το power analysis και το power routing. Στο πρώτο από αυτά, πραγματοποιούνται οι υπολογισμοί για την αριθμητική απεικόνιση των αποτελεσμάτων των φαινομένων πτώσης τάσης και μετανάστευσης ηλεκτρονίων, ενώ στο δεύτερο υλοποιούνται οι συνδέσεις των power pins των κελιών του κυκλώματος με τις global power lines που έχουν καθοριστεί σε προηγούμενο βήμα.

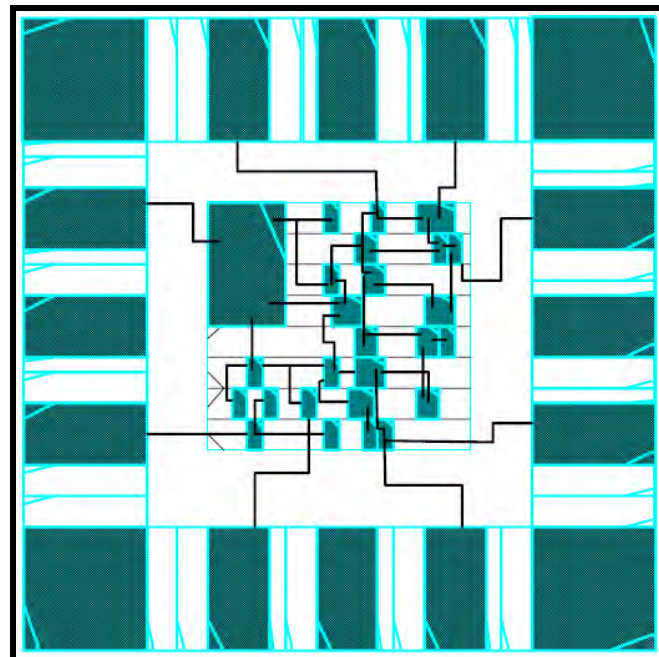


ΕΙΚΟΝΑ 3.12: Power Analysis

Τέλος, στο στάδιο του routing υλοποιούνται οι τελικές διασυνδέσεις μεταξύ των επιμέρους τμημάτων της σχεδίασης έτσι ώστε να απαιτείται το μικρότερο δυνατό μήκος καλωδίου.



EIKONA 3.13: Power Routing



EIKONA 3.14: Final Routing

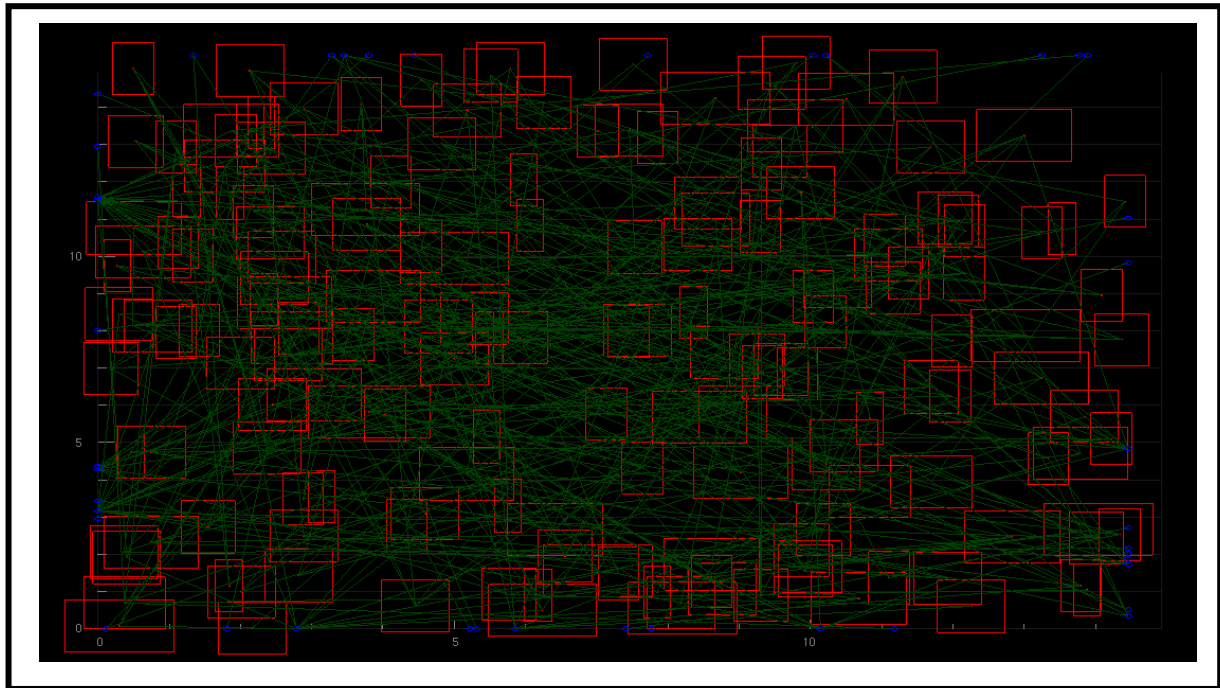
Το τελικό αποτέλεσμα της διαδικασίας είναι ένα σύνολο αρχείων περιγραφής του κυκλώματος, με σημαντικότερο από αυτά ένα αρχείο `.def` το οποίο περιέχει αναλυτικά τις θέσεις και τον τρόπο διασυνδέσεις όλων των `modules` του κυκλώματος.

3.1.3. ΕΞΟΔΟΣ

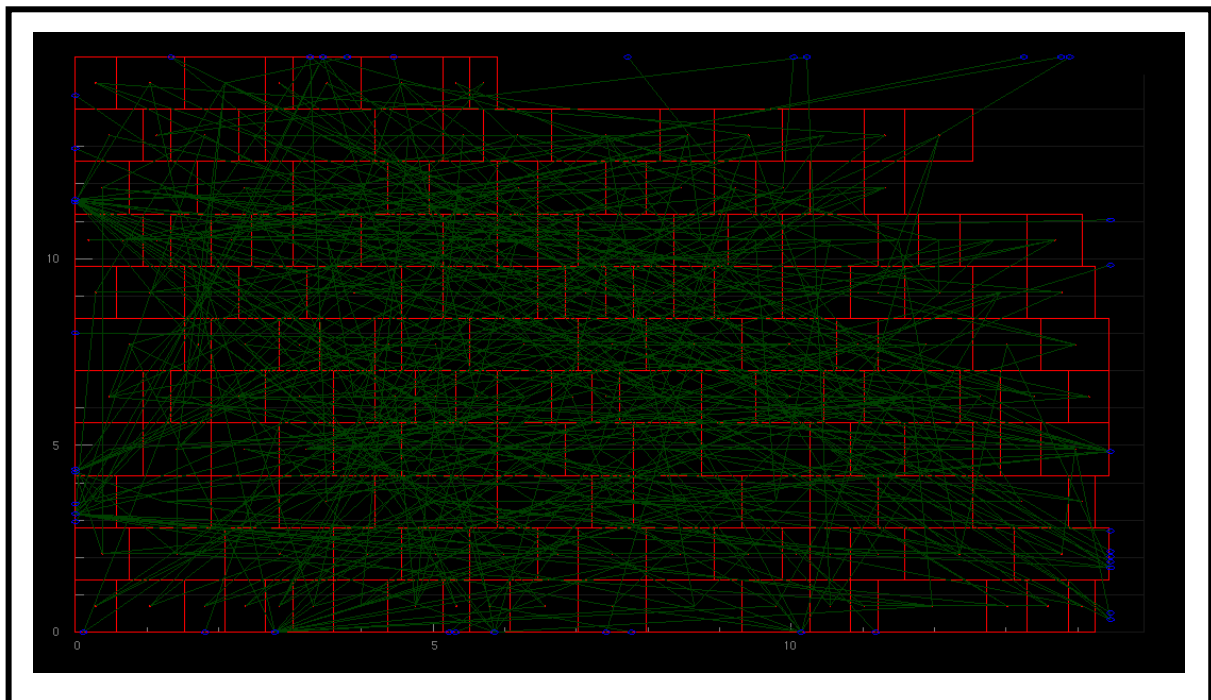
Η πλατφόρμα που υλοποιήσαμε παράγει στην έξοδό της μια ομάδα από `reports` τα οποία αναφέρονται σε όλες τις επιμέρους διεργασίες που επιτελέστηκαν, ενώ παρουσιάζονται επιπρόσθετες πληροφορίες για την διάταξη του κυκλώματος.

Παράλληλα, δημιουργείται ένα `.def` αρχείο, το οποίο προσδιορίζει, πλήρως, όλες τις αλλαγές που η εκτέλεση του αλγορίθμου έχει επιφέρει στο κύκλωμα. Η προκείμενη επιλογή αυτού του τύπου αρχείου έγκειται στο γεγονός πως θέλουμε τα αποτελέσματά μας να συνάδουν με όλα τα σύγχρονα σχεδιαστικά `standards` για την ευκολότερη εισαγωγή τους στις σύγχρονες ροές σχεδιάσεις που ακολουθούνται στη βιομηχανία.

Τέλος, δίνεται η δυνατότητα στον χρήστη να παρακολουθήσει, μέσα από μια σειρά γραφικών αναπαραστάσεων, όλες τις αλλαγές που επιτελούνται στο προς εξέταση κύκλωμα, σε κάθε ένα από τα στάδια εκτέλεσης του αλγορίθμου. Τα γραφικά έχουν υλοποιηθεί κάνοντας χρήση του πακέτου `GLUT` που προσφέρει η `C`, εξαλείφοντας την απαίτηση εγκατάστασης νέου λογισμικό. Στις εικόνες που ακολουθούν παρουσιάζονται δύο στάδια εκτέλεσης του αλγορίθμου, ένα αρχικό τυχαίο `placement` που παρουσιάζει φαινόμενα επικάλυψης και η τελική χωροθέτηση του κυκλώματος μετά το πέρας της διαδικασίας «νομιμοποίησης» των τελικών θέσεων των κελιών (`legalization`).



EIKONA 3.15: Random Initial Placement



EIKONA 3.16: Final Placement

3.2. WIRELENGTH-DRIVEN PLACEMENT

Σε αυτό το κεφάλαιο θα παρουσιαστεί η προσέγγιση που ακολουθήσαμε για την υλοποίηση ενός wirelength-driven placer, ενός εργαλείου, δηλαδή, υπεύθυνου για την κατάλληλη χωροθέτηση ενός κυκλώματος κατά τέτοιο τρόπο ώστε το συνολικό απαιτούμενο μήκος καλωδίου για την διασύνδεση των στοιχείων του να είναι το ελάχιστο δυνατό.

Ο αλγόριθμος δέχεται στην είσοδό του ένα netlist σε επίπεδο κελιών και την τεχνολογική βιβλιοθήκη που χρησιμοποιήθηκε για την σύνθεσή του. Στο πρώτο βήμα εκτέλεσης του αλγορίθμου πραγματοποιείται το parsing των παραπάνω αρχείων και η αποθήκευση των χρήσιμων πληροφοριών που αυτά περιέχουν σε κατάλληλες δομές δεδομένων. Για την ευκολότερη και γρηγορότερη πρόσβαση στα δεδομένα έχει υλοποιηθεί ένα hashtable κατάλληλου μεγέθους, του οποίου η χρησιμότητα γίνεται προφανής όσο αυξάνεται το μέγεθος, των προς εξέταση κυκλωμάτων.

Ακολούθως, υπολογίζεται το μέγεθος της συνολικής επιφάνειας πάνω στην οποία θα γίνει η τοποθέτηση των δομικών στοιχείων του κυκλώματος. Η επιλογή που έχει παρθεί είναι να θεωρούμε τα input και output pins ως οντότητες χωρίς πραγματικές διαστάσεις τα οποία τοποθετούνται στην περιφέρεια του προκείμενου χώρου.

Η συνολική επιφάνεια χωροθέτησης μπορεί είτε να καθοριστεί από τον χρήστη, είτε στην εξ'ορισμού περίπτωση να υπολογιστεί βάσει του ακόλουθου τύπου:

$$\text{placement_area} = 110\% \times \text{total_cell_area}$$

Βάσει του εμβαδού του διαθέσιμου χώρου, γίνεται ο υπολογισμός των πλευρών του παραλληλογράμμου που θα χρησιμοποιηθεί. Τα προς χωροθέτηση κελιά είναι standard cells, έχουν, δηλαδή, ίδιο ύψος και διαφέρουν μόνο ως προς το μήκος τους. Το ύψος, λοιπόν, του παραλληλογράμμου πρέπει να είναι πολλαπλάσιο του ύψους αυτού.

Η επιλογή που έχει γίνει στο σημείο αυτό είναι να δημιουργείται ένα παραλληλόγραμμο, όσο πιο κοντά γίνεται στην μορφή ενός τετραγώνου. Παρόλαυτα, παρέχεται η δυνατότητα χειρονακτικού καθορισμού των διαστάσεων.

Στο επόμενο στάδιο, πραγματοποιείται η αρχικοποίηση των θέσεων των κελιών του κυκλώματος. Η αρχικοποίηση δύναται να εκτελεστεί με δύο τρόπους, είτε ψευδοτυχαία βάσει μιας ακέραιης τιμής η οποία παρέχεται από το χρήστη, είτε μέσω ενός .def file από το οποίο, με το κατάλληλο parsing, εξάγονται οι συντεταγμένες των κελιών ενός placed κυκλώματος που έχει προκύψει από κάποιο βιομηχανικό εργαλείο placement.

Ακολουθεί η αρχικοποίηση των συντεταγμένων των pins εισόδου και εξόδου, της σχεδίασης κατ'αναλογία των παραπάνω περιγραφόμενων τρόπων (ψευδοτυχαία ή μέσω .def file).

Από το σημείο αυτό και έπειτα, ακολουθούνται ορισμένα βήματα εκτέλεσης υπολογισμών τα οποία προσομοιάζουν στον ευρύτερο τρόπο λειτουργίας του Gordian, όπως αυτός έχει περιγραφεί σε παραπάνω ενότητα. Υπολογίζεται, λοιπόν, ο adjacency matrix ο οποίος «απεικονίζει» την διασυνδεσιμότητα μεταξύ των κελιών και ακολούθως ο pin connection matrix ο οποίος περιγράφει τον τρόπο σύνδεσης μεταξύ των κελιών και των pins εισόδου και εξόδου της σχεδίασης. Με αφετηρία τους προκείμενους πίνακες υπολογίζονται οι degree matrix και laplacian matrix, όπως και τα fixed pin vectors, έτσι ώστε το προς επίλυση πρόβλημα να μορφοποιηθεί ως ένα quadratic problem.

Υπολογίζεται το αρχικό συνολικό wirelength του κυκλώματος, το οποίο μοντελοποιείται με δύο διαφορετικούς τρόπους. Ο πρώτος είναι σύμφωνα με την ευκλείδεια απόσταση μεταξύ των συνδεδεμένων κελιών, ενώ ο δεύτερος βασίζεται στο μήκος της ημιπεριμέτρου του νοητού παραλληλογράμμου που περιβάλλει (εσωκλείει)

κάθε ένα από τα nets της σχεδίασης. Ο διπλός προσδιορισμός του απαιτούμενου μήκους καλωδίου, βοηθά στην εξαγωγή περαιτέρω συμπερασμάτων για τον κατάλληλο τρόπο αναπαράστασης των συνδέσεων των δομικών μονάδων του ολοκληρωμένου μετά το πέρας της εκτέλεσης του αλγορίθμου και την εξαγωγή των τελικών αποτελεσμάτων.

Στη συνέχεια, ορίζεται ένα καθολικό partition το οποίο καταλαμβάνει όλο τον διαθέσιμο χώρο και εμπερικλείει όλα τα κελιά. Το global partition πρόκειται να διαχωριστεί αναδρομικά για την επιμέρους εξέταση και εύρεση των ιδανικών θέσεων των δομικών μονάδων της σχεδίασης.

Στο σημείο αυτό, παρεμβάλλεται ένα ακόμα βήμα επιλογής. Ο χρήστης καλείται να επιλέξει εάν θα αφήσει τις αρχικές θέσεις των κελιών ως έχουν ή θα επιλέξει την δημιουργία ενός legalized initial placement ως αρχική λύση του προβλήματος πάνω στην οποία θα βασιστούν οι μετέπειτα βελτιστοποιήσεις. Επιπρόσθετα, ορίζεται το αριθμητικό όριο, της ομάδας των κελιών που εσωκλείονται σε ένα partition, κάτω από το οποίο ο αλγόριθμος κρίνει άσκοπη την ενασχόλησή του με το προκείμενο χωρικό τμήμα.

Ακολούθως, εκκινείται η επαναληπτική διαδικασία επίλυσης του προβλήματος και κατ'επέκταση βελτιστοποίησης των μετρικών ποσοτήτων που έχουν επιλεγεί. Ο αριθμός των επαναλήψεων καθορίζεται από την ύπαρξη subpartitions της σχεδίασης τα οποία πληρούν όλες τις προδιαγραφές προκειμένου να ασχοληθεί ο αλγόριθμος μαζί τους. Οι επαναλήψεις τερματίζονται, επομένως, όταν ο διαδοχικός διαχωρισμός της επιφάνειας σε επιμέρους τμήματα σταματήσει να παράγει χώρους με μέγεθος και αριθμό κελιών ικανούς προς ανάλυση.

Ο τρόπος που γίνεται η επιλογή του cut line έχει κρατηθεί απλός. Η τομή του partition γίνεται στην μεγαλύτερη πλευρά του και το χωρίζει σε δύο τμήματα. Παρολαυτά, έχει υλοποιηθεί και μια δεύτερη μέθοδος, κατ'επιλογή του χρήστη,

σύμφωνα με την οποία ελέγχεται και επιλέγεται το καταλληλότερο σημείο τομής το οποίο βρίσκεται εντός του διαστήματος ($0.5 \times \text{side}$, $0.75 \times \text{side}$). Η σύμβαση που ακολουθήθηκε στην περίπτωση εύρεσης του κατάλληλου cut line, είναι η ακόλουθη: ο αλγόριθμος εξετάζει αρχικά την περιοχή που βρίσκεται αριστερά του cut line στην περίπτωση τομής στον άξονα Y και την κάτω περιοχή στην περίπτωση τομής στον άξονα X.

Έπεται, ο έλεγχος καταλληλότητας του partition που δημιουργήθηκε, έτσι ώστε να αποφασιστεί εάν το προκειμένο τμήμα θα εξεταστεί από το κυρίως τμήμα του αλγορίθμου. Τα κριτήρια βάσει των οποίων γίνεται αποδεκτός ή μη ο τρόπος διαχωρισμού είναι τα ακόλουθα:

1. Το προς εξέταση partition πρέπει να εμπεριέχει συνολικό αριθμό κελιών μεγαλύτερο ή ίσο από αυτόν που έχει οριστεί σε προηγούμενο βήμα.
2. Το μήκος ή το πλάτος του νέου partition, ανάλογα με τη φορά της τομής, πρέπει να υπερβαίνει το τετραπλάσιο του μήκους του μεγαλύτερου κελιού, που περιβάλλεται από τα όριά του.
3. Το νέο cut point επιβάλλεται να είναι αρκούντως διαφορετικό από τα επιλεγμένα cut points προηγούμενων επαναλήψεων του αλγορίθμου.
4. Τα κελιά που περιέχονται στο νέο partition πρέπει να καλύπτουν στο σύνολό τους χώρο μικρότερο του νέου τμήματος που δημιουργήθηκε.

Με την εφαρμογή των παραπάνω κριτηρίων υποβοηθείται ο αλγόριθμος στην επιλογή και την ελάττωση των προς εξέταση περιοχών (και επομένως και στην ταχύτητα εκτέλεσής του), χωρίς παράλληλα να αλλοιώνεται η ποιότητα της τελικής λύσης.

Ο έλεγχος καταλληλότητας, μπορεί να παραβλεφθεί, ανάλογα με τις επιλογές του χρήστη, έτσι ώστε να εξεταστούν οι πιθανές διαφορές στις τελικές λύσεις που παράγονται.

Αφού ολοκληρωθεί ο παραπάνω έλεγχος, καλείται ο Kernighan-Lin partitioner (KL) προκειμένου να βρεθεί ο βέλτιστος τρόπος διαμοιρασμού των κελιών στα δύο νέα partitions που έχουν δημιουργηθεί. Ο KL δέχεται στην είσοδό του έναν μη κατευθυνόμενο γράφο αποτελούμενο από τα standard cells του προς εξέταση χώρου. Το βάρος των ακμών καθορίζεται από το μοντέλο που έχει επιλεγεί για την συμπλήρωση των θέσεων του adjacency matrix. Στην συνέχεια εξετάζει, διεξοδικά, όλες τις πιθανές ανταλλαγές θέσεων που δύναται να πραγματοποιηθούν και τις κοστολογεί αναλόγως. Οι τελικές επιλογές είναι αυτές που παρουσιάζουν το το μικρότερο κόστος μετακίνησης.

Στο σημείο αυτό έχει οριστεί πλήρως το νέο partition το οποίο θα μας απασχολήσει. Εντός του partition, λοιπόν, εκτελείται μια παραλλαγή της διαδικασίας που περιγράψαμε κατά την αναφορά μας στον αλγόριθμο Gordian. Υπολογίζονται εκ νέου όλοι οι τύποι πινάκων και διανυσμάτων που έχουν προαναφερθεί και επιλύεται το γραμμικό σύστημα που δημιουργείται.

Για την ορθή συμπλήρωση των πινάκων, ακολουθείται η εξής σύμβαση: Όλες οι συνδέσεις μεταξύ κελιών ή μεταξύ κελιών και pins οι οποίες υπερβαίνουν τα όρια του partition μετατρέπονται σε pseudo-pins πάνω στην περιφέρεια της περιοχής, γεγονός που οδηγεί στον επαναπροσδιορισμό των βαρών των νέων nets που δημιουργούνται.

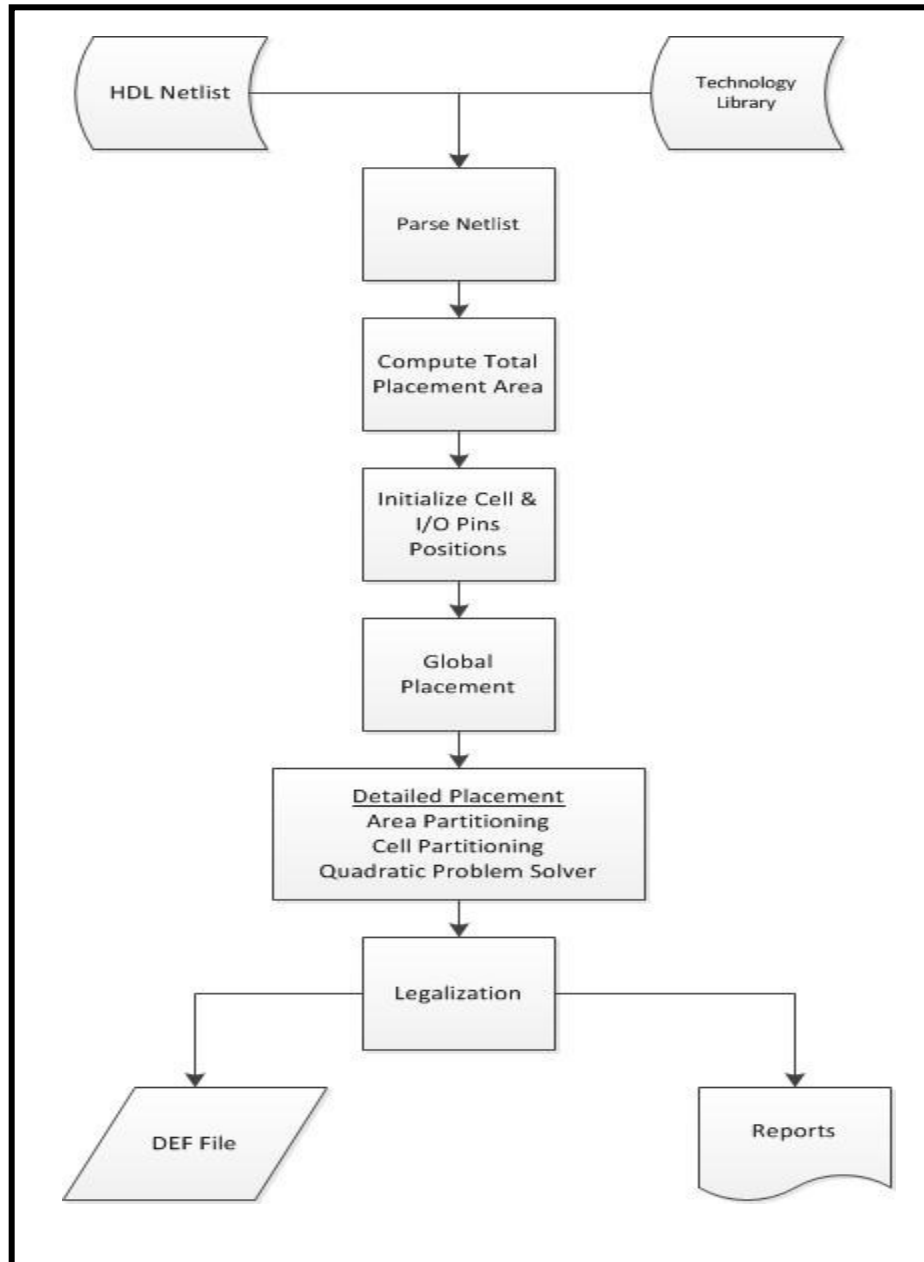
Όταν ολοκληρωθεί η επαναληπτική διαδικασία, όταν δηλαδή έχουν εξετασθεί όλοι οι υποχώροι που έχουν δημιουργηθεί και έχουν υπολογιστεί οι ιδανικές θέσεις των κελιών, καλείται ο αλγόριθμος Abacus προκειμένου να τελεστεί το κατάλληλο legalization και να εξαλειφθούν τα φαινόμενα επικάλυψης που παρουσιάζονται. Ο τρόπος λειτουργίας παρουσιάζεται στον ψευδοκώδικα στην εικόνα που ακολουθεί:

```
Sort cells according to x-position;  
foreach cell i do  
    best_cost ← ∞;  
    foreach row r do  
        Insert cell i into row r;  
        PlaceRow r (trial);  
        Determine cost c;  
        if c < best_cost then  
            best_cost = c;  
            rbest = r;  
        Remove cell i from row r;  
    end  
    Insert Cell i to row rbest ;  
    PlaceRow rbest (final);  
end
```

EIKONA 3.17: Abacus legalization approach

Για κάθε ένα από τα κελιά της σχεδίασης πραγματοποιείται μια πειραματική τοποθέτησή του σε κάθε μία από τις προκαθορισμένες σειρές. Η εκάστοτε κίνηση προσδιορίζεται από κάποιο κόστος. Μετά το πέρας όλων των επαναλήψεων το κελί τοποθετείται σε εκείνη την σειρά, όπου παρουσιάζεται το μικρότερο κόστος. Όταν όλα τα κελιά ανατεθούν στις κατάλληλες σειρές, εξαλείφονται φαινόμενα επικάλυψης (ως προς τον άξονα x) εντός των ορίων τους.

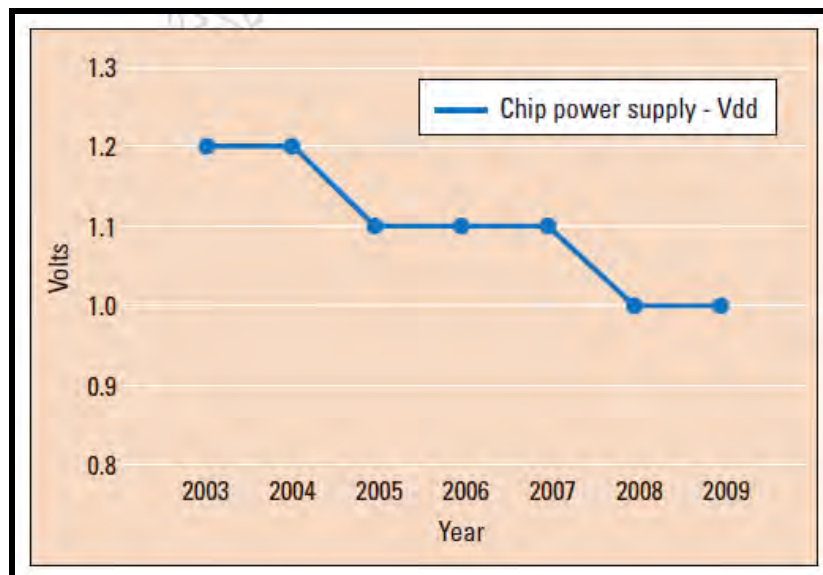
Στο τελευταίο στάδιο εκτέλεσης του αλγορίθμου υπολογίζεται εκ νέου το συνολικό wirelength της σχεδίαση και αποθηκεύονται οι θέσεις των κελιών στα κατάλληλα αρχεία εξόδου. Στην εικόνα που ακολουθεί παρουσιάζεται η συνολική ροή εκτέλεσης της προτεινομένης προσέγγισης.



EIKONA 3.18: Wirelength-Driven placement

3.3. IR-DROP BASED RE-PLACEMENT

Η αύξηση της πολυπλοκότητας, όπως και της ταχύτητας λειτουργίας των σύγχρονων ολοκληρωμένων κυκλωμάτων, έχει επιφέρει δραματική αύξηση και της καταναλισκόμενης ισχύος. Επιπρόσθετα, παρατηρείται αισθητή ελάττωση της απαιτούμενης τάσης τροφοδοσίας.



ΕΙΚΟΝΑ 3.19: IC Supply Voltages by year

Η μείωση της τάσης τροφοδοσίας έχει ως επακόλουθο τον περιορισμό των περιθωρίων θορύβου (noise margins) εντός των οποίων μπορεί να λειτουργήσει ορθά μία σχεδίαση, καθώς τα στοιχεία που την αποτελούν καθίστανται ευάλωτα στο power supply noise. Η πτώση τάσης που παρατηρείται κατά την λειτουργία ενός κυκλώματος επηρεάζει και το χρονισμό της σχεδίασης, καθώς στα κελιά δεν παρέχεται η κατάλληλη τάση με αποτέλεσμα να μην μπορούν να φτάσουν στο επιθυμητό επίπεδο εντός του rise time που τα χαρακτηρίζει.

Βασιζόμενη σε αυτή την παρατήρηση, υλοποιήσαμε έναν αλγόριθμο επαναχωροθέτησης ενός κυκλώματος βασιζόμενοι στην ανάλυση των φαινομένων *ir-drop* που παρουσιάζει.

Ο αλγόριθμος δέχεται στη είσοδό του ένα αρχείο `.def` μιας *pre-placed legalized* σχεδίασης, ένα `.lib` αρχείο το οποίο περιγράφει την τεχνολογική βιβλιοθήκη που χρησιμοποιείται και ένα αρχείο εξόδου του *Design Compiler* το οποίο παρουσιάζει δεδομένα που αφορούν το χρονοισμό της σχεδίασης, και πιο συγκεκριμένα, το σύνολο των κελιών που αποτελούν το *critical path* του προκειμένου κυκλώματος.

Επιπρόσθετα, τροφοδοτούμε τον αλγόριθμο με ένα αρχείο το οποίο προκύπτει από την *ir-drop analysis* του κυκλώματος και παρουσιάζει τις πτώσεις τάσης που παρατηρούνται σε κάθε κελί. Η *ir-drop analysis* πραγματοποιείται χρησιμοποιώντας *Extreme Value Theory* προκειμένου να λάβουμε μια πολύ καλή εκτίμηση της πραγματικής τιμής του *ir-drop* σε κάθε κελί εντός ενός μικρού περιθωρίου λάθους. Για να ολοκληρωθεί επιτυχώς η διαδικασία εκτίμησης απαιτούνται 3000 δείγματα κατ'ελάχιστη ποσότητα τα οποία προκύπτουν από την εφαρμογή των κατάλληλων *input vectors* στην είσοδο.

Ο αλγόριθμος εντοπίζει το *critical path* και το αποθηκεύει σε μια κατάλληλη δομή προκειμένου να είναι εύκολη η πρόσβασή μας στα συγκεκριμένα μόνο κελιά. Ακολούθως, δημιουργείται ένα *ir-drop mapping* του κυκλώματος βάσει των δεδομένων που έχουν αντληθεί από την διαδικασία εκτίμησης που προσδιορίστηκε παραπάνω. Στο σημείο αυτό, ο αλγόριθμος καλείται να εντοπίσει τα κατάλληλα κελιά, βάσει συγκεκριμένων κριτηρίων τα οποία είναι υποψήφια προς ανταλλαγή με τα κελιά που απαρτίζουν το κρίσιμο μονοπάτι.

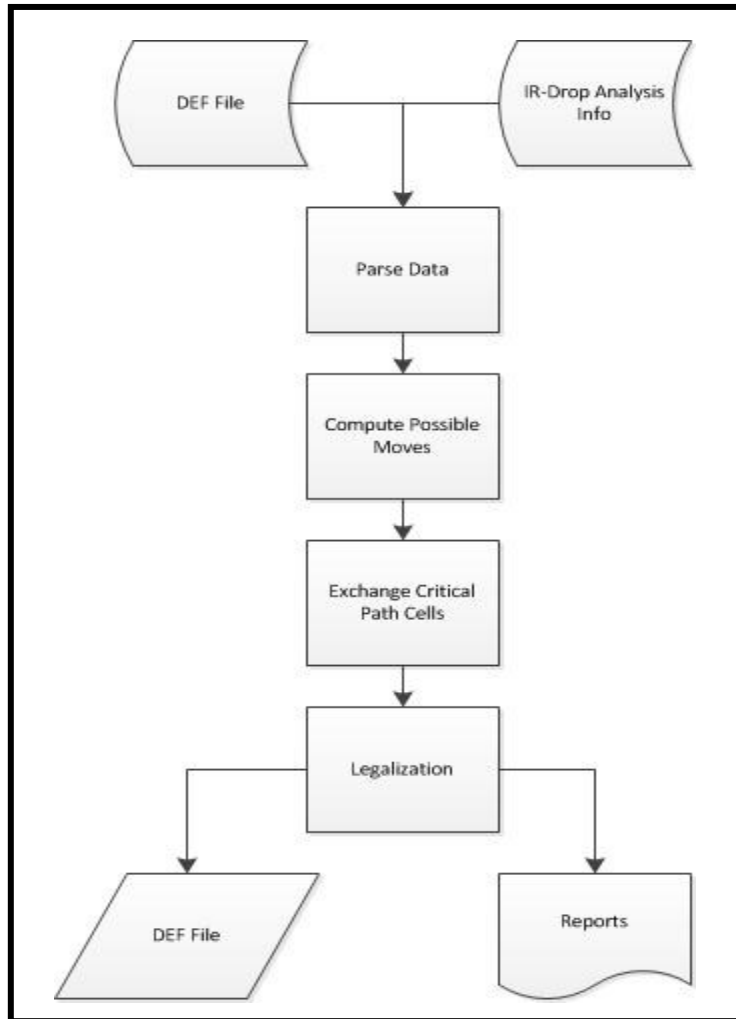
Για να διευκολυνθεί η επιλογή των κελιών αλλά και χάριν πιθανής επεκτασιμότητας των κριτηρίων επιλογής, τα προς επιλογή κελιά έχουν ομαδοποιηθεί

ως προς το ir-drop που παρουσιάζουν. Αυτό δίνει την δυνατότητα στο σχεδιαστή να «προεπιλέξει» την ομάδα κελιών στην οποία θα πραγματοποιηθεί η έρευνα. Ένα επιπλέον πλεονέκτημα του άτυπου αυτού clustering, είναι η ευκολία αποκλεισμού ολόκληρων ομάδων κελιών από τα αποτελέσματα της έρευνας, γεγονός το οποίο με τη σειρά του οδηγεί στην βελτίωση της ταχύτητας εκτέλεσης του αλγορίθμου.

Η προσέγγιση που ακολουθήθηκε βασίζεται στην ανταλλαγή μεταξύ του κελιού με το μεγαλύτερο ir-drop και του κελιού που βρίσκεται εγγύτερα σε αυτό και παρουσιάζει μικρότερη πτώση τάσης. Αφού εκτελεστούν οι ανταλλαγές η σχεδίαση γίνεται εκ νέου legalized προκειμένου να εξαλειφθούν τυχών φαινόμενα επικάλυψης που ενδέχεται να εμφανιστούν μετά την πάροδο των ανταλλαγών.

Η «νομιμοποίηση» του τελικού αποτελέσματος πραγματοποιείται εξετάζοντας κάθε μία από τις σειρές τοποθέτησης των κελιών, και μετακινώντας κατ'ελάχιστη απόσταση, προς εκείνη την κατεύθυνση όπου υπάρχει περιθώριο τα κελιά, διαφορώντας για την μαθηματική διατύπωση κάποιας συνάρτησης κόστους, έτσι ώστε να μην παρουσιάζεται υπέρ του δέοντος αύξηση στο συνολικό μήκος καλωδίου που απαιτείται.

Η ροή εκτέλεσης της προτεινόμενης προσέγγισης παρουσιάζεται στην εικόνα που ακολουθεί.



ΕΙΚΟΝΑ 3.20: IR-Drop Based Re-Placement

Η τελική καθυστέρηση που παρατηρείται στο critical path υπολογίζεται ως το άθροισμα των καθυστερήσεων των επιμέρους κελιών που το αποτελούν. Η νέα καθυστέρηση του κελιού υπολογίζεται βάσει του ακόλουθου τύπου:

$$\text{new_cell_delay} = \text{old_cell_delay} * (V_{dd_old} - V_t) / (V_{dd_new} - V_t)$$

όπου V_t είναι το $V_{\text{threshold}}$ και $V_{dd} = V_{\text{nom}} - \text{IR-Drop}$.

ΚΕΦΑΛΑΙΟ 4

BENCHMARK CIRCUITS

Ο ορθός προσδιορισμός της απόδοσης ενός εργαλείου EDA είναι ένα από τα σημαντικότερα ζητήματα που έχει να αντιμετωπίσει ένας σχεδιαστής. Ένας, ευρέως διαδεδομένος, τρόπος ελέγχου του τελικού αποτελέσματος είναι η επιλογή των κατάλληλων benchmark circuits τα οποία θα δοθούν ως είσοδος στο λογισμικό. Στην προκειμένη διατριβή χρησιμοποιήθηκαν τα ISCAS '89 και τα ITC '99 benchmark circuits. Στο υπολειπόμενο κομμάτι του κεφαλαίου θα παρουσιαστούν βασικά χαρακτηριστικά των κυκλωματικών αυτών σχεδιάσεων και στατιστικά στοιχεία για τον αριθμό και το είδος των modules που τα αποτελούν.

ISCAS '89: Οι κυκλωματικές περιγραφές των ISCAS '89 κυκλωμάτων παρέχονται τόσο σε structural όσο και σε behavioral μορφή. Στο σύνολό τους, αυτά τα υψηλού επιπέδου μοντέλα σχεδίασης έχουν αποδειχθεί, ιδιαιτέρως, χρήσιμα ως εργαλεία έρευνας σε πολλούς τομείς της ψηφιακής σχεδίασης κυκλωμάτων με σημαντικότερους το test generation, την διαδικασία του timing analysis και το technology mapping. Στο επίσημο documentation των προκειμένων benchmarks παρατίθεται ένα σύνολο επιπλέον πληροφοριών, πέρα από τις περιγραφές τους σε γλώσσα υλικού. Εμείς θα αρκεστούμε στο να παρουσιάσουμε, στην εικόνα που ακολουθεί, έναν πίνακα με τον

αριθμό των primary inputs/outputs και κελιών που περιέχονται σε κάθε ένα από τα κυκλώματα αυτής της ομάδας.

Circuit	FF's	PI's	PO's
S298	14	3	6
S349	15	9	11
S382	21	3	6
S386	6	7	7
S400	21	3	6
S420	16	19	2
S444	21	3	6
S510	6	19	7
S526	21	3	6
S641	19	35	24
S820	5	18	19
S832	5	18	19
S838	32	35	2
S953	29	16	23
S1196	18	14	14
S1238	18	14	14
S1423	74	17	5
S1488	6	8	19
S1494	6	8	19
S5378	179	35	49
S9234	211	19	22
S35932	1728	35	320

EIKONA 4.1: ISCAS '89 benchmark circuits

ITC '99: Η ανάγκη ύπαρξης benchmark circuits τα οποία δεν θα είναι αρκετά μικρά ή αρκετά απλά, οδήγησε στην ομαδοποίηση ενός συνόλου ψηφιακών σχεδιάσεων, οι οποίες συλλέχτηκαν από εταιρείες των οποίων το αντικείμενο έρευνας άπτεται του τομέα της ψηφιακής σχεδίασης ολοκληρωμένων κυκλωμάτων. Τα βασικά χαρακτηριστικά των ITC '99 benchmark circuits είναι τα ακόλουθα:

- Είναι πλήρως συνθέσιμα (synthesizable) με χρήση του Synopsys Design Compiler.
- Δεν εμπεριέχουν compiler specific οδηγίες.
- Όλα τα πακέτα που απαιτούνται από τις περιγραφές των κυκλωμάτων σε HDL γλώσσα είναι είτε arithmetic packages είτε IEEE standard logic packages.
- Το global reset signal είναι πάντα διαθέσιμο.

Οι κυκλωματικές περιγραφές που περιέχονται ποικίλουν, από απλά «μονολιθικά» κυκλώματα (1 entity, 1 process) έως κυκλώματα πολλαπλών entity και διεργασιών. Πιο συγκεκριμένα, ένα από τα κυκλώματα που περιέχονται σε αυτή την ομάδα είναι τρεις φορές μεγαλύτερο από το μεγαλύτερο ISCAS '89 (37 inputs, 69.917 gates, 3.320 flip-flops).

Ακολουθεί ένας πίνακας με τον αριθμό των πολών, των primary inputs/outputs και flip-flops που εμπεριέχονται σε κάθε ένα από τα κυκλώματα.

Circuit	Gates	PI's	PO's	FF's
B01	49	2	2	5
B02	28	1	1	4
B03	160	4	4	30
B04	737	8	11	66
B05	998	1	36	34
B06	56	2	6	9
B07	441	1	8	49
B08	183	9	4	21
B09	170	1	1	28
B10	206	11	6	17
B11	770	7	6	31
B12	1.076	5	6	121
B13	362	10	10	53
B14	10.098	32	54	245
B15	8.992	36	70	449
B17	32.326	37	97	1.415
B18	114.621	36	23	3.320
B19	231.320	21	30	6.642
B20	20.226	32	22	490
B21	20.571	32	22	490
B22	29.951	32	22	735

EIKONA 4.2: ITCC '99 benchmark circuits

ΚΕΦΑΛΑΙΟ 5

ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ & ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο κεφάλαιο αυτό θα παρουσιαστούν τα πειραματικά αποτελέσματα από την εφαρμογή του αλγορίθμου πάνω στο σύνολο των benchmark circuits που προαναφέραμε.

Στην περίπτωση του wirelength-driven placement το μετρικό που χρησιμοποιήθηκε ήταν το συνολικό απαιτούμενο μήκος καλωδίου διασύνδεσης των στοιχείων του κυκλώματος, ενώ στην περίπτωση του ir-drop based re-placement εξετάσαμε την επίδραση του αλγορίθμου πάνω στον χρονισμό του critical path.

5.1. Fixed Pin Positions, Random Initial Cell Placement, Tool-Calculated Area

Στην πρώτη εξεταζόμενη περίπτωση, υπολογίσαμε το μέγεθος και τις διαστάσεις της επιφάνειας πάνω στην οποία θα γίνει το placement βάσει του εμπειρικού τύπου που παρουσιάσαμε σε προηγούμενο κεφάλαιο. Επιλέχθηκε, τυχαία, μια θέση για κάθε ένα από τα I/O pins του κυκλώματος (στην περιφέρεια του παραλληλογράμμου) και εκτελέσαμε επαναληπτικά τον αλγόριθμο για ένα σύνολο τυχαίων αρχικών χωροθετήσεων των κελιών.

Επιπρόσθετα, θεωρήσαμε πως ο ελάχιστος αριθμός κελιών που στοιχειοθετούν ένα partition είναι ίσος με τέσσερα.

Στον παρακάτω πίνακα παρουσιάζονται οι μέγιστες και οι ελάχιστες τιμές ανάμεσα στις οποίες κυμαίνεται η βελτιστοποίηση του wirelength για κάθε ένα από τα χρησιμοποιούμενα net models όπως και ο μέσος όρος βελτίωσης που παρατηρείται στο σύνολο των επαναλήψεων.

exp.1	max.opt.eucl.	max.opt.hp.	min.opt.eucl.	min.opt.hp.	eucl.avg.	hp.avg.
S27	21.66506347	27.17410255	-27.63435627	-39.3270377	0.698072789	1.280903425
S208	23.48890575	26.72290007	-12.53209464	-16.64214134	7.303163675	9.184266371
S298	22.45345625	25.57499982	-8.883305797	-9.125780931	8.589262686	9.2611862
S344	19.97369022	22.46515337	-6.01206355	-9.223546858	6.543607588	7.357392786
S349	19.56080974	21.69304528	-7.175212594	-9.994692153	6.651535155	7.654431673
S382	19.85853271	24.49416655	-9.236229312	-7.937283269	7.278945596	8.439843747
S386	21.97712807	26.0255992	-12.34125944	-8.438724605	6.028576884	8.281044304
S400	22.43193097	27.07522857	-7.495531654	-11.16454687	8.425727679	9.052868008
S420	17.54884534	21.9980845	-7.690204153	-9.244196851	5.061529681	6.61130864
S444	19.79478086	21.70309551	-6.03485912	-8.469837521	7.324559169	7.477720879
S510	16.65094994	20.64898895	-6.115077779	-7.176703831	5.713188348	7.478917242
S526	20.97998675	23.09807111	-10.02968698	-14.26536916	7.356435417	8.689040555
S641	17.66709992	19.77215614	-3.962102793	-2.52391868	7.951799585	9.485287624
S713	16.79820754	18.24879333	-2.652055389	-4.128972744	7.592942456	8.87910134
S820	17.67660838	23.97873058	-7.699853697	-8.799537338	7.116062594	8.847953601
S832	17.21767818	19.68781279	-4.502343337	-4.491997554	6.165192333	7.516858676
S838	14.33303528	15.59821639	-5.630463144	-6.694883129	4.725137608	6.681563338
S953	16.37082108	18.40807549	-0.310833479	-2.146579291	7.834604452	8.679310763
S1196	16.85468074	18.47285411	-0.854452126	-3.747938362	7.604774101	8.854048715
S1238	11.42206036	15.06542788	-2.700737107	-2.265858424	5.108406793	6.762831444
S1423	14.03240043	16.37857851	0.595803401	1.651840657	7.02410773	9.43015034
S1488	13.95594656	15.64579457	-2.788068349	-4.298775475	6.518929974	6.738579197
S1494	11.21691044	14.93182774	-2.333701166	-4.350051615	3.774701142	4.558035062
S5378	9.329390232	11.34860053	-1.069781256	-0.193010795	4.096296361	5.657477607
S9234	9.928515312	11.7299332	0.302138957	1.195898929	4.771624757	6.565257627

EIKONA 5.1: Fixed Pin Positions, Random Initial Cell Placement, Tool-Calculated Area ISCAS results

exp.2	max.opt.eucl.	max.opt.hp.	min.opt.eucl.	min.opt.hp.	eucl.avg.	hp.avg.
b01	14.61624324	15.58662634	-16.16218535	-20.07541589	-0.46021191	0.111307971
b02	21.02995479	24.09736216	-22.66306811	-32.72024952	-0.62361884	1.721037316
b03	7.048343443	7.78380461	-14.62169743	-18.39599905	-1.48931107	-1.13387021
b04	15.30199705	19.70033264	-1.797288931	1.119700258	8.432314385	11.2164444
b05	18.87733551	20.98496028	3.401353983	6.596140626	11.36244392	13.79872559
b06	12.57536638	14.05666566	-17.1813337	-17.77119424	-0.60834303	0.012630465
b07	19.58600135	21.49366378	-2.283878947	-3.016851641	9.323857023	12.61127367
b08	8.778776045	7.374875356	-10.64930334	-9.476893774	-0.3840942	-0.40001631
b09	8.696287204	6.76308108	-16.8721597	-15.66113042	-1.93640351	-1.33569968
b10	6.875543347	8.490075652	-11.65284918	-15.65828182	-0.70507776	-0.70478783
b11	20.23303207	21.40650664	-1.246579603	3.866338362	11.01158589	14.22495248
b12	14.71959199	19.01536626	1.832472742	5.065396312	9.318591776	11.95621003
b13	19.16737062	22.507429	2.492935054	2.163261386	10.05797391	14.65916354
b14	12.14986356	15.4665288	3.342992933	7.781480021	8.227183016	11.10763434
b15	11.12819891	19.11697796	5.743732189	8.206988986	8.592084527	11.69964005

EIKONA 5.2: Fixed Pin Positions, Random Initial Cell Placement, Tool-Calculated Area ITC results

Βάσει των παραγόμενων αποτελεσμάτων, καθίσταται εμφανής η επιρροή που έχει στο τελικό αποτέλεσμα το αρχικό placement των κελιών. Η επαναληπτική διαδικασία που εκτελείται βελτιώνει σταδιακά την θέση του εκάστοτε κελιού βάσει ενός συνόλου κριτηρίων. Ένα net του οποίου τα κελιά έχουν διασκορπιστεί σε μεγάλη απόσταση μεταξύ τους, ή σε μεγάλη απόσταση από τα pins εισόδου και εξόδου που αναλογούν στα κελιά που το απαρτίζουν, θα χωροθετηθεί «ακατάλληλα» επιβαρύνοντας έτσι το συνολικό μήκος καλωδίου που απαιτείται για την διασύνδεση των στοιχείων του.

Ο καταλληλότερος τρόπος αρχικής τοποθέτησης των κελιών εντός της υφιστάμενης περιοχής είναι η ψευδοτυχαία τοποθέτησή τους σε άμεση ισορροπία μεταξύ της απόστασης τους από τα έτερα κελιά που απαρτίζουν το net και τα I/O pins που τους αναλογούν.

5.2. Fixed Initial Cell Positions, Random Pin Positions, Tool-Calculated Area

Στην δεύτερη περίπτωση που εξετάσαμε, θεωρήσαμε μια σταθερή αρχική τυχαία χωροθέτηση των κελιών, και εκτελέσαμε κατ'επανάληψη την διαδικασία χωροθέτησης που υλοποιήθηκε αλλάζοντας κατά τυχαίο τρόπο τις θέσεις των pins εισόδου και εξόδου.

Το συνολικό εμβαδό της περιοχής που έχουμε στη διάθεσή μας υπολογίστηκε με τον τρόπο που αναφέραμε παραπάνω, και ο ελάχιστος αριθμός κελιών βάσει των οποίων στοιχειοθετείται ένα partition κρατήθηκα ίσος με 4.

Στους πίνακες που ακολουθούν παρουσιάζονται αναλυτικά τα αποτελέσματα που προέκυψαν από την εφαρμογή του αλγορίθμου.

Βάσει των αποτελεσμάτων καθίσταται σαφής η επιρροή των θέσεων των pins στο τελικό αποτέλεσμα. Η τοποθέτηση των pins σε αντιδιαμετρικά αντίθετες πλευρές του παραλληλογράμμου, «εγκλωβίζει» το κελί και περιορίζει την μετακίνησή του.

Η εύρεση του καταλληλότερου τρόπου τοποθέτησης των I/O pins στα πλαίσια μιας πλήρως αυτοματοποιημένης διαδικασίας χωροθέτησης είναι ένα σημαντικό ζήτημα στο οποίο δεν έχει δοθεί η δέουσα σημασία.

Τέλος, βάσει των αποτελεσμάτων παρατηρούμε πως τα πιθανά τελικά αποτελέσματα της εφαρμογής του αλγορίθμου, κυμαίνονται μεταξύ ενός μικρότερου περιθωρίου σε σχέση με τα αποτελέσματα που παρουσιάστηκαν στην προηγούμενη ενότητα. Το γεγονός αυτό, μας οδηγεί στο συμπέρασμα πως οι θέσεις των pins επηρεάζουν σε μικρότερο βαθμό το τελικό αποτέλεσμα από ότι η αρχική τοποθέτηση των κελιών.

exp.3	max.opt.eucl.	max.opt.hp.	min.opt.eucl.	min.opt.hp.	eucl.avg.	hp.avg.
S27	3.702351408	5.285176134	-6.176998462	-2.31971	-0.16755	2.355642
S208	14.07936146	21.31194851	-0.360554262	5.919922	7.295084	14.0058
S298	15.38207493	21.58353397	-7.814778622	-2.44975	6.613484	12.53938
S344	14.52046891	16.0322137	-1.467392776	-0.41757	6.789391	7.692707
S349	19.52926672	20.17659111	4.823245329	1.899562	12.31397	12.97138
S382	18.12813706	18.98603311	5.52512102	3.111286	11.32592	11.26453
S386	10.33770681	20.07372413	-3.30945408	0.809953	3.247952	9.86592
S400	19.934546	21.09830651	5.686254026	8.443784	13.35342	15.00286
S420	15.13306395	17.27860061	-0.175181754	1.30804	7.97938	8.847009
S444	17.50639743	20.35932011	1.879154018	3.158019	9.806969	12.21078
S510	14.3362191	20.84199304	1.502257435	2.586214	7.58221	11.21175
S526	8.974808742	13.62340969	-2.221189495	1.105275	3.563103	7.362096
S641	14.04367364	12.70445668	3.306256045	-1.09782	8.556546	6.568918
S713	12.13196234	15.96674449	0.535620443	5.300343	6.96126	10.87032
S820	17.0143546	21.17444179	4.849401874	9.075553	11.26021	15.37342
S832	9.046033196	11.43377771	-1.036091227	-0.62154	4.306152	4.897135
S838	13.28597893	16.01318054	3.013667208	3.115637	8.015185	9.3111
S953	10.0361752	11.15729796	1.172835778	1.909006	5.42978	6.363436
S1196	9.83395528	12.69758355	0.894120479	0.717295	4.744567	6.33923
S1238	7.610041817	10.1785187	0.419535826	-0.2287	3.80463	5.043169
S1423	12.05363693	14.43061088	5.250215387	7.368031	8.675241	10.5612
S1488	8.405011468	8.022473564	0.29250961	-2.545	4.762008	2.876706
S1494	8.430406725	11.88836107	1.164889079	1.675913	4.892481	5.595778
S5378	8.265731304	8.786047191	2.63407067	2.998773	5.4707	6.010282
S9234	8.131379265	9.733341263	2.836611999	3.633168	5.319224	6.778202

EIKONA 5.3: Fixed Initial Cell Positions, Random Pin Positions, Tool-Calculated Area ISCAS results

exp.4	max.opt.eucl.	max.opt.hp.	min.opt.eucl.	min.opt.hp.	eucl.avg.	hp.avg.
b01	-6.571520647	-5.96308034	-11.81409983	-5.96308	-9.12363	-5.96308
b02	6.938434029	6.485926162	3.908573268	6.485926	5.805523	6.485926
b03	-4.131584885	-1.00363229	-5.24099149	-1.00363	-4.70428	-1.00363
b04	9.773021021	11.94327213	4.449374651	6.441961	6.886882	9.389903
b05	12.82778016	12.68295716	7.3064547	8.804104	9.812369	10.59313
b06	0.676909542	-0.29631915	-3.597592215	-0.29632	-1.70579	-0.29632
b07	15.23296024	22.69271664	6.760475002	16.77805	11.11546	19.23827
b08	4.386668371	4.621784185	3.64499733	4.621784	4.060661	4.621784
b09	2.660192956	-0.2565579	2.313465053	-0.25656	2.520013	-0.25656
b10	2.910844473	4.479532705	1.407359506	4.479533	2.215038	4.479533
b11	13.35766893	17.85423589	8.612106761	13.16348	11.31611	15.50937
b12	16.41457544	18.96249131	8.564572065	11.22389	12.98073	16.05326
b13	18.26114488	19.55271442	10.99728152	14.00892	14.45192	16.90522
b14	9.764196904	13.83424841	4.912352355	8.063356	7.615559	10.87882
b15	9.523499093	14.11710422	6.664496194	11.4379	8.063526	12.83217

EIKONA 5.4: Fixed Initial Cell Positions, Random Pin Positions, Self-Calculated Area ITC results

5.3. Fixed Initial Cell Positions, Random Pin Positions, CAD Tool-Calculated Area

Στην επόμενη περίπτωση που εξετάσαμε, κρατήσαμε σταθερό το initial placement των κελιών και τρέξαμε τον αλγόριθμο για μια σειρά τυχαίων τοποθετήσεων των I/O pins σε μια επιφάνεια της οποίας το εμβαδό προϋπολογίστηκε από ένα βιομηχανικό εργαλείο χωροθέτησης.

exp.5	max.opt.eucl.	max.opt.hp.	min.opt.eucl.	min.opt.hp.	eucl.avg.	hp.avg.
S27	1.208503937	0.551514449	-6.279134402	-4.759549398	-2.10358195	-1.52166345
S208	17.23353566	22.54079028	-8.7319296	-3.307040103	5.604406888	8.930599741
S298	14.96456431	20.32830966	-3.495327099	-1.657381803	6.503613091	9.110597455
S344	17.78167597	22.45606456	2.184830887	5.899045289	10.06395602	13.87654042
S349	15.9951486	14.94527574	-2.117787197	-5.485372147	7.116980415	5.724221894
S382	18.83139706	20.76437323	0.639594945	1.096453309	9.665210114	10.72275995
S386	13.77126805	20.76391674	-2.341106652	-0.356157465	6.303820166	10.01462784
S400	13.46927011	16.95866837	-5.311318218	-1.464660472	5.145815756	7.130869134
S420	9.702990534	11.12442415	-3.159676424	-5.43779537	3.871629466	2.806115777
S444	14.75955249	17.58369299	-0.808514943	0.437082739	7.189391177	9.108207969
S510	11.2536266	16.55375272	-1.924026444	0.955147655	4.768141118	8.918840119
S526	17.01184319	26.38998929	0.467342821	9.525652905	9.715102247	17.4563974
S641	17.66046545	17.7253814	0.657986538	-2.237176431	9.814179632	7.640017865
S713	12.06530937	17.39771521	-2.317382143	1.251082621	5.113622357	9.521010304
S820	12.95126713	18.86469088	2.118698562	5.878236822	7.54435964	12.5837186
S832	13.35293348	17.66373849	0.168800288	0.423071636	7.140480439	9.360877635
S838	11.49158716	14.31150995	-1.955495907	-0.19341656	5.32201889	7.1379282
S953	13.86713829	15.989384	2.787607198	2.872605203	8.078856031	9.256010665
S1196	12.06473414	11.53284503	0.854915596	-1.899069369	6.131560287	5.197609228
S1238	13.32113932	15.43715267	2.618153642	2.006646616	7.5018401	8.844891901
S1423	14.5875253	15.69056102	4.24590753	6.713024246	10.83365138	11.31551197
S1488	13.25013308	14.80119211	2.808852759	0.868964652	8.09541439	7.802486867
S1494	12.21372337	13.97884373	0.881596365	1.280429998	5.969811168	7.483201536
S5378	10.5146851	11.58920077	3.529439148	3.70915322	6.888325726	7.469569592
S9234	11.57399673	13.49167388	4.786744362	6.481078431	8.397832574	9.873433659

EIKONA 5.5: Fixed Initial Cell Positions, Random Pin Positions, CAD Tool-Calculated Area ISCAS results

exp.6	max.opt.eucl.	max.opt.hp.	min.opt.eucl.	min.opt.hp.	eucl.avg.	hp.avg.
b01	3.056248354	9.006000594	-3.59410308	9.006000594	-0.39131553	9.006000594
b02	6.443832838	0.21501563	0.226199599	0.21501563	3.680527574	0.21501563
b03	8.55629431	22.4089948	-9.698946531	5.987042713	-0.87349707	13.74882189
b04	17.77701941	22.4938356	6.000576007	9.343265827	10.85236414	13.99426104
b05	20.15204932	22.25937432	14.43500239	15.55653148	17.23090248	18.25966523
b06	2.106715589	-0.66519504	-2.64523215	-0.665195037	0.140451881	-0.66519504
b07	17.78227823	26.20609856	5.315020486	16.64849637	10.64081743	21.36209799
b08	3.550249575	4.527562925	2.250679007	4.527562925	3.007364557	4.527562925
b09	12.00046114	12.59526247	3.588737082	2.093784874	8.168105535	7.701672527
b10	0.443772758	1.462361681	-1.975779317	1.462361681	-0.71540574	1.462361681
b11	16.17344346	18.58644719	8.81157049	11.67213627	12.23446169	15.4899711
b12	16.03730789	16.52804599	7.268385395	7.264130745	12.36299442	12.26660741
b13	24.28344769	27.30744678	16.87119206	18.45504856	20.92585565	22.58852114
b14	19.38226429	19.77434696	14.58574217	15.04862811	16.8937231	17.72485256
b15	14.55609025	15.97921595	8.656860054	13.08229303	11.92084591	14.566925

EIKONA 5.6: Fixed Initial Cell Positions, Random Pin Positions, CAD Tool-Calculated Area ITC results

Όπως και στην προηγούμενη ενότητα παρατηρούμε πως ο τρόπος τοποθέτησης των I/O pins επηρεάζει άμεσα το τελικό αποτέλεσμα του placement. Η βασική διαφορά με το προηγούμενο πείραμα είναι πως ο απαιτούμενος χώρος για την χωροθέτηση είναι διαφορετικός. Πιο συγκεκριμένα, τα κριτήρια χώρου που επιλέξαμε είναι αρκετά αυστηρά. Τα βιομηχανικά εργαλεία απαιτούν μεγαλύτερο χώρο για την κατάλληλη χωροθέτηση ενός κυκλώματος. Βάσει αυτής της παρατήρησης δικαιολογείται το γεγονός πως τα μεγαλύτερα κυκλώματα που εξετάσαμε παρουσιάζουν καλύτερα αποτελέσματα αφού υπάρχει μεγαλύτερο περιθώριο μετακίνησης και καταλληλότερης τοποθέτησης των στοιχείων τους.

5.4. Fixed Pin Positions, Random Initial Cell Placement, CAD Tool-Calculated Area

Στην τελευταία περίπτωση wirelength-driven placement που εξετάσαμε οι διαστάσεις του χώρου υπολογίστηκαν βάσει βιομηχανικών εργαλείων χωροθέτησης και τα pins εισόδου και εξόδου τοποθετήθηκαν σε συγκεκριμένες θέσεις οι οποίες παρέμειναν σταθερές σε όλες τις επαναλήψεις.

exp.7	max.opt.eucl.	max.opt.hp.	min.opt.eucl.	min.opt.hp.	eucl.avg.	hp.avg.
S208	24.47025558	29.00248268	-8.511908067	-13.3542	9.10201	10.01769
S298	20.36693751	23.5956977	-7.17423353	-10.1997	6.629723	7.948524
S344	20.26311571	23.56651416	-2.366259449	-5.79135	6.919214	7.741345
S349	19.15511487	22.5440831	-8.896076266	-6.66567	7.805646	8.599306
S382	17.2033831	22.49177581	-10.6733836	-7.03994	6.37316	7.338066
S386	18.37612195	21.17527424	-9.931433902	-8.44584	5.799353	7.3431
S400	19.77098198	20.22951128	-9.102571599	-10.3776	6.321283	7.104904
S420	14.83582036	18.92924286	-5.915235977	-5.47801	3.646894	5.835189
S444	18.9265214	22.22032761	-4.57837795	-4.10994	7.679237	8.392367
S510	14.93215994	20.26279938	-5.72476766	-7.88766	5.102898	7.413193
S526	22.00381951	27.4023157	-10.0604987	-9.16405	8.105576	9.498831
S641	17.90663456	19.21835442	-1.726951899	-1.69883	6.892892	8.149883
S713	15.65874287	19.65991893	-2.583778488	-3.03017	6.699895	7.643628
S820	15.4908063	20.5606271	-3.691526582	-4.24458	6.509826	7.984524
S832	15.45398878	20.99676158	-3.993685219	-4.56312	6.348916	7.510379
S838	14.26824172	16.44174527	-3.297522965	-2.61409	5.610221	6.872001
S953	15.63036683	17.06291033	-1.816455198	-1.13586	7.68935	7.804829
S1196	14.97188244	17.13229127	0.009942861	-0.69147	7.425947	7.975361
S1238	13.35634171	16.61523177	-1.941681199	-1.85086	7.149366	7.881045
S1423	13.98095415	17.09307397	1.644008962	2.70087	7.777475	9.436264
S1488	14.11042042	16.30476938	0.000287688	-1.52715	7.328887	7.859375
S1494	12.07823222	17.82075414	-1.434761716	-2.38308	6.070555	6.993382
S5378	11.95125647	13.66045468	0.60647727	0.658808	5.608788	6.72293
S9234	11.22990994	13.08073273	2.444210153	2.363647	6.678582	7.971984

EIKONA 5.7: Fixed Pin Positions, Random Initial Cell Placement, CAD Tool-Calculated Area ISCAS results

exp.8	max.opt.eucl.	max.opt.hp.	min.opt.eucl.	min.opt.hp.	eucl.avg.	hp.avg.
b01	23.25268502	27.99315775	-29.53292853	-22.653	1.196817	2.899375
b02	24.60846365	29.39264631	-39.50025802	-47.1466	1.159777	2.908743
b03	21.83827011	29.51244344	-15.74931286	-7.92925	5.303708	9.803202
b04	16.72275801	18.3333509	-0.373289916	0.232917	8.252364	10.26006
b05	24.89603208	23.56814352	4.11349236	3.09021	12.77396	13.77352
b06	16.53387181	17.10069884	-22.55748629	-16.502	0.741128	1.784721
b07	23.9544966	23.92979046	-1.535305017	-0.88852	9.173003	11.49004
b08	13.53926948	12.42931463	-13.03517	-12.4055	1.346728	1.539709
b09	18.61346272	27.39086032	-12.62736948	-12.3725	4.428253	7.280453
b10	13.13838513	12.1158734	-10.59627733	-11.623	1.192539	1.727219
b11	22.34920093	22.40462728	1.431808172	2.331438	11.03578	12.44445
b12	20.81949981	23.05438175	1.870847628	1.765015	12.27471	13.51131
b13	25.96239408	26.5332919	-2.505939653	-0.22572	10.63432	12.5017
b14	20.62775993	21.84134767	7.561824153	9.515734	15.39603	15.74247
b15	15.07655859	18.21697835	7.063284246	11.02643	10.92905	13.95506

EIKONA 5.8: Fixed Pin Positions, Random Initial Cell Placement, CAD Tool-Calculated Area ITC results

Βάσει των παραγόμενων αποτελεσμάτων, καθίσταται εμφανής η επιρροή που έχει στο τελικό αποτέλεσμα το αρχικό placement των κελιών. Ο καταλληλότερος τρόπος αρχικής τοποθέτησης των κελιών εντός της υφιστάμενης περιοχής είναι η ψευδοτυχαία τοποθέτησή τους σε άμεση ισορροπία μεταξύ της απόστασης τους από τα έτερα κελιά που απαρτίζουν το net και τα I/O pins που τους αναλογούν.

Επιπρόσθετα, λόγω της μεγαλύτερης διαθέσιμης επιφάνειας ο αλγόριθμος δύναται να πραγματοποιήσει μεγαλύτερες βελτιστοποιήσεις όσον αφορά το συνολικό μέγεθος καλωδίου διασύνδεσης που απαιτείται.

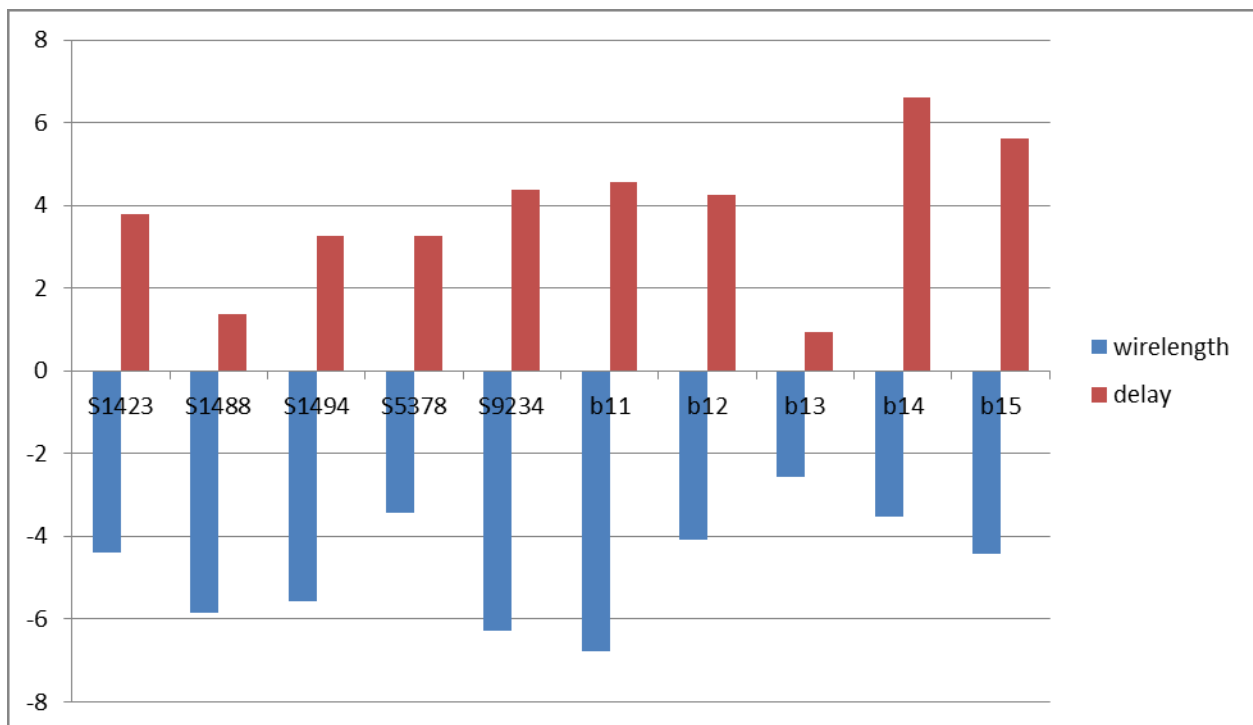
Στο σημείο αυτό θα διατυπώσουμε και μια σειρά ευρύτερων παρατηρήσεων βάσει των συνολικών αποτελεσμάτων των παραπάνω στιγμιότυπων εκτέλεσης του αλγορίθμου.

1. Η τελική χωροθέτηση του κυκλώματος επηρεάζεται σε μεγαλύτερο βαθμό από την τυχαιότητα του αρχικού cell placement παρά από την τυχαία τοποθέτηση των I/O pins.
2. Στις περιπτώσεις όπου ο χώρος καθορίστηκε βάσει βιομηχανικών εργαλείων παρουσιάστηκε μεγαλύτερη βελτίωση, λόγω της διευρυμένης περιοχής τοποθέτησης με την οποία έπρεπε να δουλέψει ο αλγόριθμος.
3. Η βελτίωση που παρατηρείται κάνοντας χρήση του clique model είναι στην πλειοψηφία των περιπτώσεων μεγαλύτερη από την αντίστοιχη της χρήσης του ευκλείδειου μοντέλου, γεγονός το οποίο οφείλεται κατά μεγάλο βαθμό στο γεγονός πως το μοντέλο κλικας λαμβάνει υπόψη του την ισορροπία μεταξύ όλων των στοιχείων που απαρτίζουν το net.
4. Όσο μεγαλύτερος είναι ο αριθμός των επαναλήψεων του αλγορίθμου τόσο πιο εύκολη καθίσταται η εφαρμογή του Extreme Value Theory στο σύνολο των αποτελεσμάτων για τον υπολογισμό μιας καλής εκτίμησης του καλύτερου δυνατού ποσοστού βελτιστοποίησης που μπορούμε να επιτύχουμε.

5.5. IR-Drop Based Re-Placement

Στην εικόνα που ακολουθεί παρουσιάζονται τα αποτελέσματα της εφαρμογής του IR-Drop Based Re-Placement αλγορίθμου που υλοποιήθηκε, κατά την εφαρμογή του στα μεγαλύτερα από άποψη συνόλου κελιών benchmark circuits που παρουσιάσαμε σε προηγούμενη ενότητα.

Παρατηρούμε πως οι υπό προϋποθέσεις ανταλλαγές κελιών που ανήκουν στο critical path με άλλα κελιά της σχεδίασης μπορούν να οδηγήσουν σε σημαντική ελάττωση του χρονισμού της σχεδίασης επηρεάζοντας σε μικρό βαθμό το συνολικό μήκος καλωδίου.



EIKONA 5.9: IR-Drop Based Re-Placement, ISCAS & ITC results

Το tradeoff μεταξύ της βελτίωσης του χρονισμού της σχεδίασης και της αύξησης του συνολικού μήκους καλωδίου κρίνεται αποδεκτό, εάν αναλογιστούμε πως οποιαδήποτε βελτίωση μεγαλύτερη του 1% θεωρείται πρακτικά συμφέρουσα, ασχέτως των αλλαγών που αναγκάζεται να επιτελέσει ο σχεδιαστής στο κύκλωμα.

ΚΕΦΑΛΑΙΟ 6

ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Η πλατφόρμα που παρουσιάστηκε μπορεί να αποτελέσει την βάση ενός βιομηχανικού εργαλείου χωροθέτησης ολοκληρωμένων κυκλωμάτων. Οι δυνατές επεκτάσεις και παραλλαγές των αλγορίθμων παρουσιάζονται συνοπτικά στο προκείμενο κεφάλαιο.

- Υλοποίηση επιπλέον μεθόδων αναπαράστασης των δικτύων κελιών (net modeling)
- Χρήση πολλαπλών net models κατά την επεξεργασία του κυκλώματος.
- Υλοποίηση νέων αλγορίθμων cell partitioning.
- Επιλογή του κατάλληλου αλγορίθμου cell partitioning βάσει παραμέτρων που δίνονται από το χρήστη.
- Υπολογισμός των «ιδανικών» τιμών επιπρόσθετων παραμέτρων βάσει της εκάστοτε εξεταζόμενης σχεδίασης (αριθμός κελιών που στοιχειοθετούν ένα υποχώρο, περιορισμοί στον προσδιορισμό του κατάλληλου cutline).
- Υλοποίηση επιπρόσθετων αλγορίθμων area partitioning.
- Χωροθέτηση εντός πλαισίου με μη διαθέσιμους προς επεξεργασία χώρους.
- Υλοποίηση επιπρόσθετων μεθόδων «νομιμοποίησης».
- Υλοποίηση επιπλέον τρόπων επιλογής των προς ανταλλαγή κελιών στο IR-Drop Based Re-Placement.

- Υλοποίηση ενός IR-Drop Based Re-Placement αλγορίθμου ο οποίος λαμβάνει υπόψη του ένα σύνολο από critical paths.
- Υλοποίηση Time-Driven Placement.
- Υλοποίηση Thermal Placement.
- Δημιουργία γραφικής διεπαφής.
- Χρήση των υπάρχοντων αλγορίθμων σε συνδιασμό με το EVT για την παραγωγή εκτιμήσεων του ελαχίστου μήκους καλωδίου που απαιτείται σε μια σχεδίαση, και ποιοτικός έλεγχος του τελικού αποτελέσματος και των ευριστικών μεθόδων που χρησιμοποιήθηκαν βάσει της προκειμένης εκτίμησης.
- Παραλληλοποίηση των αλγορίθμων.
- Routing της σχεδίασης.
- Αναπαράσταση των pins εισόδου/ εξόδου βάσει των πραγματικών διαστάσεων, και όχι ως σημεία.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Jun Cheng Chi; Tsung Hui Huang; Mely Chen Chi; , "An IR drop-driven placer for standard cells in a SOC design," *SOC Conference, 2005. Proceedings. IEEE International* , vol., no., pp.29-32, 19-23 Sept. 2005
- [2] Yasar, G.; Chiu, C.; Proctor, R.A.; Libous, J.P.; , "I/O cell placement and electrical checking methodology for ASICs with peripheral I/Os," *Quality Electronic Design, 2001 International Symposium on* , vol., no., pp.71-75, 2001
- [3] Roy, J.A.; Markov, I.L.; , "Seeing the Forest and the Trees: Steiner Wirelength Optimization in Placement," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.26, no.4, pp.632-644, April 2007
- [4] Dutt, S.; Ren, H.; , "Discretized Network Flow Techniques for Timing and Wire-Length Driven Incremental Placement With White-Space Satisfaction," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.19, no.7, pp.1277-1290, July 2011
- [5] Viswanathan, N.; Chu, C.C.-N.; , "FastPlace: efficient analytical placement using cell shifting, iterative local refinement, and a hybrid net model," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.24, no.5, pp. 722- 733, May 2005

- [6] Kleinhans, J.M.; Sigl, G.; Johannes, F.M.; Antreich, K.J.; , "GORDIAN: VLSI placement by quadratic programming and slicing optimization," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.10, no.3, pp.356-365, Mar 1991
- [7] Alpert, C.J.; Jen-Hsin Huang; Kahng, A.B.; , "Multilevel circuit partitioning," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.17, no.8, pp.655-667, Aug 1998
- [8] Dragon, [online] Available: <http://er.cs.ucla.edu/Dragon>
- [9] Kahng, A.B.; Qinke Wang; , "Implementation and extensibility of an analytic placer," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.24, no.5, pp. 734- 747, May 2005
- [10] Cong, J.; Sung Kyu Lim; , "Retiming-based timing analysis with an application to mincut-based global placement," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.23, no.12, pp. 1684- 1692, Dec. 2004
- [11] Kahng, A.B.; Reda, S.; , "Wirelength minimization for min-cut placements via placement feedback," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.25, no.7, pp.1301-1312, July 2006
- [12] Brenner, U.; Rohe, A.; , "An effective congestion-driven placement framework," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.22, no.4, pp. 387- 394, Apr 2003

- [13] Suaris, P.R.; Kedem, G.; , "A quadrisection-based combined place and route scheme for standard cells," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.8, no.3, pp.234-244, Mar 1989
- [14] Dunlop, A.E.; Kernighan, B.W.; , "A Procedure for Placement of Standard-Cell VLSI Circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.4, no.1, pp. 92- 98, January 1985
- [15] Caldwell, A.E.; Kahng, A.B.; Markov, I.L.; , "Optimal partitioners and end-case placers for standard-cell layout ," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.19, no.11, pp.1304-1313, Nov 2000
- [16] Yan, J.Z.; Chu, C.; Wai-Kei Mak; , "SafeChoice: A Novel Approach to Hypergraph Clustering for Wirelength-Driven Placement," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.30, no.7, pp.1020-1033, July 2011
- [17] Yu-Wen Tsay; Youn-Long Lin; , "A row-based cell placement method that utilizes circuit structural properties," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.14, no.3, pp.393-397, Mar 1995
- [18] Madden, P.H.; , "Reporting of standard cell placement results," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.21, no.2, pp.240-247, Feb 2002

- [19] Yeong-Yil Yang; Chong-Min Kyung; , "HALO: an efficient global placement strategy for standard cells," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.11, no.8, pp.1024-1031, Aug 1992
- [20] Kennings, A.; Vorwerk, K.P.; , "Force-Directed Methods for Generic Placement," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.25, no.10, pp.2076-2087, Oct. 2006
- [21] Brenner, U.; Vygen, J.; , "Legalizing a placement with minimum total movement," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.23, no.12, pp. 1597- 1613, Dec. 2004
- [22] Ching-Wei Yeh; Chung-Kuan Cheng; Lin, T.T.Y.; , "A general purpose, multiple-way partitioning algorithm," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.13, no.12, pp.1480-1488, Dec 1994
- [23] Hyunchul Shin; Chunghee Kim; , "A simple yet effective technique for partitioning," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.1, no.3, pp.380-386, Sept. 1993
- [24] Chandy, J.A.; Sungho Kim; Ramkumar, B.; Parkes, S.; Banerjee, P.; , "An evaluation of parallel simulated annealing strategies with application to standard cell placement," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.16, no.4, pp.398-410, Apr 1997

- [25] Wern-Jieh Sun; Sechen, C.; , "A parallel standard cell placement algorithm," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.16, no.11, pp.1342-1357, Nov 1997
- [26] Rose, J.S.; Snelgrove, W.M.; Vranesic, Z.G.; , "Parallel standard cell placement algorithms with quality equivalent to simulated annealing," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.7, no.3, pp.387-396, Mar 1988
- [27] Spindler, P.; Schlichtmann, U.; Johannes, F.M.; , "Kraftwerk2—A Fast Force-Directed Quadratic Placement Approach Using an Accurate Net Model," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.27, no.8, pp.1398-1411, Aug. 2008
- [28] Hwan Gue Cho; Kyung, C.M.; , "A heuristic standard cell placement algorithm using constrained multistage graph model," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.7, no.11, pp.1205-1214, Nov 1988
- [29] ITC '99 Benchmarks, [online] Available: <http://www.cerc.utexas.edu/itc99-benchmarks>
- [30] APlace: A High Quality, Large-Scale Analytical Placer, [online] Available: <http://vlsicad.ucsd.edu/APlace/>
- [31] Yildiz, M.C.; Madden, P.H.; , "Improved cut sequences for partitioning based placement," *Design Automation Conference, 2001. Proceedings* , vol., no., pp. 776- 779, 2001

- [32] Vorwerk, K.; Kennings, A.; Vannelli, A.; , "Engineering details of a stable force-directed placer," *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on* , vol., no., pp. 573- 580, 7-11 Nov. 2004
- [33] Agnihotri, A.; Yildiz, M.C.; Khatkhate, A.; Mathur, A.; Ono, S.; Madden, P.H.; , "Fractional cut: improved recursive bisection placement," *Computer Aided Design, 2003. ICCAD-2003. International Conference on* , vol., no., pp. 307-310, 9-13 Nov. 2003
- [34] Capo, [online] Available: <http://vlsicad.eecs.umich.edu/BK/PDtools/Capo/>
- [35] Forum for Electronics, [online] Available: <http://www.edaboard.com>
- [36] Sangiovanni-Vincentelli, A.; , "The tides of EDA," *Design & Test of Computers, IEEE* , vol.20, no.6, pp. 59- 75, Nov.-Dec. 2003
- [37] Golshan, K. "Physical Design Essentials: An ASIC Design Implementation Perspective". New York: Springer (2007). ISBN 0-387-36642-3
- [38] Lavagno, Martin and Scheffer. "Electronic Design Automation For Integrated Circuits Handbook". (2006). ISBN 0-8493-3096-3
- [39] Jansen, D. "The Electronic Design Automation Handbook". Kluwer Academic Publishers. (2003). ISBN 1-4020-7502-2
- [40] Dutt, S.,, "New faster Kernighan-Lin-type graph-partitioning algorithms," *Computer-Aided Design, 1993. ICCAD-93. Digest of Technical Papers., 1993 IEEE/ACM International Conference on* , vol., no., pp.370-377, 7-11 Nov 1993

- [41] Thang Bui; Jones, C.; Heigham, C.; Leighton, T.; , "Improving the Performance of the Kernighan-Lin and Simulated Annealing Graph Bisection Algorithms," *Design Automation, 1989. 26th Conference on* , vol., no., pp. 775-778, 25-29 June 1989
- [42] Saab, Y.; , "Post-analysis-based clustering dramatically improves the Fiduccia-Mattheyses algorithm," *Design Automation Conference, 1993, with EURO-VHDL '93. Proceedings EURO-DAC '93. European* , vol., no., pp.22-27, 20-24 Sep 1993
- [43] Saab, Y.G.; , "A fast and robust network bisection algorithm," *Computers, IEEE Transactions on* , vol.44, no.7, pp.903-913, Jul 1995
- [44] Wang, T.-C.; Wong, D.F.; , "On Stockmeyer's floorplan optimization technique," *Circuits and Systems, 1992. ISCAS '92. Proceedings., 1992 IEEE International Symposium on* , vol.4, no., pp.1989-1992 vol.4, 3-6 May 1992
- [45] Pei-Ning Guo; Takahashi, T.; Chung-Kuan Cheng; Yoshimura, T.; , "Floorplanning using a tree representation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.20, no.2, pp.281-289, Feb 2001
- [46] ISCAS '89 Benchmarks, [online] Available: <http://www.eecs.umich.edu/~jhayes/iscas/>
- [47] Wakabayashi, K.; Okamoto, T.; , "C-based SoC design flow and EDA tools: an ASIC and system vendor perspective," *Computer-Aided Design of Integrated*

- Circuits and Systems, IEEE Transactions on*, vol.19, no.12, pp.1507-1522, Dec 2000
- [48] Tsung-Yi Ho; Sheng-Hung Liu; , "Fast legalization for standard cell placement with simultaneous wirelength and displacement minimization," *VLSI System on Chip Conference (VLSI-SoC), 2010 18th IEEE/IFIP* , vol.18, pp.369-374, 27-29 Sept. 2010
- [49] Alpert, C.J.; Chan, T.F.; Kahng, A.B.; Markov, I.L.; Mulet, P.; , "Faster minimization of linear wirelength for global placement," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.17, no.1, pp.3-13, Jan 1998
- [50] Caldwell, A.E.; Kahng, A.B.; Mantik, S.; Markov, I.L.; Zelikovsky, A.; , "On wirelength estimations for row-based placement," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.18, no.9, pp.1265-1278, Sep 1999
- [51] Alpert, C.J.; Kahng, A.B.; , "Recent directions in netlist partitioning: a survey, Integration, " *The VLSI Journal*, vol.19, issues 1-2, pp.1-81, August 1995
- [52] Areibi, S.; Thompson, M.; Vanelli, A.; , "A utility-based iterative improvement heuristic for standard cell placement, " *In Proceedings of First International Conference on Engineering and Reconfigurable Systems and Algorithms*, 2001

- [53] Caldwell A. E.; Kahng A. B.; Markov I.L.; , "Design and implementation of move-based heuristics for vlsi hypergraph partitioning," *ACM Journal of Experimental Algorithms*, 2000
- [54] Dasdan, A.; Aykanat, C.; , "Two novel multiway circuit partitioning algorithms using relaxed locking," *IEEE Transactions on Computer-Aided Design*, February 1997
- [55] Doll, K.; Johannes, F.; Sigl.; , "Domino: Deterministic placement with hillclimbing capabilities," *In Proceedings of VLSI*, 1991
- [56] Doll, K.; Johannes, F.; Sigl.; , "Accurate net models for placement improvement by network flow methods," *In Proceedings of the 1992 IEEE/ACM international conference on Computer-aided design*, pp.594-597, 1992
- [57] Dunlop, A.; Agrawal, V., Deutsch, D.; Juki, M.; Kozak, P.; Wiesel, M.; , "Chip layout optimization using critical path weighting," *In Proceedings of ACM/IEEE Design Automation Conference*, 1984
- [58] Faroe, O.; Pisinger, D.; Zachariassen, M.; , "Local search for final placement in VLSI design," *In Proceedings of ICCAD*, pp.565-572, 2001
- [59] Færø, O.; , "Placement of modules in vlsi-layout," *Master's thesis, University of Copenhagen, DIKU*, 2000
- [60] Grover, K.L.; , "A new simulated annealing algorithm for standard cell placement," *In Proceedings of International Conference on Computer-Aided Design*, pp.378-380, 1986

- [61] Grover, K.L.; Mallela, S.; , "Clustering based simulated annealing for standard cell placement," *In Proceedings of 25th Design Automation Conference*, pp.312-317, 1988
- [62] Hagen, L.; Kahng, A.B.; , "A new approach to effective circuit clustering," *In Proceedings of International Conference on Computer-Aided Design*, pp.422-427, 1992
- [63] Hestenes M.R.; Stiefel, E.; , "Methods of conjugate gradient for solving linear Systems," *Journal of Research of the National Bureau of Standards*, pp.409-436, 1952
- [64] Hur, S.; Lillis, J.; , "Mongrel: Hybrid techniques for standard cell placement," *In Proceedings of the International Conference on Computer Aided Design*, pp.165-170, 2000
- [65] Kennings, A.A.; Markov, I.L.; , "Analytic minimization of half perimeter wirelength," *In Proceedings of the 2000 Conference on Asia and South Pacific Design Automation*, pp.179-184, 2000
- [66] Kernighan, B.W.; Lin, S.; , "An efficient heuristic procedure for partitioning Graphs," *Bell Systems Technical Journal*, 1970
- [67] Kirkpatrick, S.; Gelatt, C.; Vecchi, M.; , "Optimization by simulated annealing," *Science*, pp.671-680, 1983

- [68] Kozminski, K.; , "Benchmarks for layout synthesis - evolution and current status," *In proceedings of ACM/IEEE Design Automation Conference*, pp.265-270, 1991
- [69] Krishnamurthy, B.; , "An improved min-cut algorithm for partitioning vlsi networks," *IEEE Transactions on Computing*, pp.438-446, May 1984
- [70] Madden, P.; , "Reporting of standard cell placement results," *IEEE Trans. Computer Aided Design of Integrated Circuits and Systems*, vol.21, pp.240-247, 2002
- [71] Reid, J.K.; , "On the methods of conjugate gradient for the solution of large sparse systems of linear equations," *In Large Sparse Sets of Linear Equations*, pp.231-254, 1971
- [72] Schewchuk, J.; , "An introduction to the conjugate gradient method without the agonizing pain, " *Technical report, CS, Carnegie Mellon University*, <http://www.cs.cmu.edu/jrs/jrspapers.html>, 1994
- [73] Sechen, C.; Lee, K.W.; , "An improved simulated annealing algorithm for row based placement," *In Proceedings of International Conference on Computer-Aided Design*, pp.478-481, 1987
- [74] Sigl, G.; Doll, K.; Johannes, F.M.; , "Analytical placement: A linear or a quadratic objective function," *In Proceedings of 28th ACM/IEEE Design Automation Conference*, pp.427-432, 1991
- [75] Sun, W.; Sechen, C.; , "Efficient and effective placement for very large circuits," *In Proceedings of International Conference on Computer Aided Design*, pp.170-177, 1993

- [76] Vygen, J.; , "Algorithms for large-scale flat placement," *In Proceedings of ACM/IEEE Design Automation Conference*, pp.746-751, 1997
- [77] Vygen, J.; , "Algorithms for detailed placement of standard cells," *In Proceedings of Design, Automation and Test in Europe 1998*, pp.321-324, 1998
- [78] Wang M.; Yang, X.; Sarrafzadeh, M.; , "Dragon2000: Standard-cell placement tool for large industry circuits," *In Proceedings of International Conference on Computer Aided Design*, pp.260-263, 2000
- [79] Sarrafzadeg, M.; Wang, M.; , "Global and detailed placement," *In Proceedings of International conference on Computer-Aided Design*, pp.532-537, 1997