



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Επισκόπηση αλγορίθμων τοποθέτησης σε Field Programmable Arrays (FPGAs)
Όνοματεπώνυμο Φοιτητή	Γεώργιος Βάσιλας
Πατρώνυμο	Εμμανουήλ
Αριθμός Μητρώου	ΜΠΣΠ / 11029
Επιβλέπων	Μιχαήλ Ψαράκης, Επίκουρος Καθηγητής

Τριμελής Εξεταστική Επιτροπή

Μιχαήλ Ψαράκης
Επίκουρος Καθηγητής

Παναγιώτης Κοτζανικολάου
Επίκουρος Καθηγητής

Άγγελος Πικράκης
Επίκουρος Καθηγητής

Περίληψη

Σκοπός της παρούσης εργασίας είναι η επισκόπηση διαφόρων αλγορίθμων τοποθέτησης σε FPGA. Το FPGA είναι ένα προγραμματιζόμενο τσιπ που μπορεί να υλοποιήσει σχεδόν οποιοδήποτε ψηφιακό κύκλωμα. Η τοποθέτηση είναι ένα σημαντικό βήμα στην σχεδίαση του FPGA που καθορίζει τις φυσικές διευθύνσεις και διασυνδέσεις των λογικών μπλοκ στο κύκλωμα και η ποιότητα του είναι ρυθμιστής για την απόδοση του κυκλώματος. Στην συγκεκριμένη εργασία θα ασχοληθούμε με τις κλασσικές μεθόδους τοποθέτησης που χρησιμοποιούνται τα τελευταία χρόνια, κάποιες καινούργιες τεχνικές τοποθέτησης, μαζί με κάποιες υβριδικές, όπως και επίσης με εξελίξεις αυτών των μεθόδων, όπως και στην εφαρμογή παραλληλίας σε αυτούς.

Abstract

The purpose of this thesis is to provide a survey of placement algorithms of FPGA. FPGA is a programmable chip that can be used to quickly implement almost any digital circuit. Placement is an important part of FPGA design step which determines physical locations and inter connections of each logic block in the circuit design, which the quality of placement becomes the bottleneck of overall circuit performance. In this thesis, we shall review the standard placement methods that have been used for the last years, along with some modern placement techniques, along with hybrids and new approaches of them, as well as with the implementation of parallelism on them.

Περιεχόμενα

Περίληψη	3
Abstract	3
Περιεχόμενα	4
Λίστα Εικόνων	6
Λίστα Πινάκων	7
1. ΕΙΣΑΓΩΓΗ	8
2. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΣΤΑ FPGAS	11
2.1 Ορισμός FPGA	11
2.2 Αρχιτεκτονική FPGA	11
2.3 CAD FPGAs	15
2.4 Κατασκευάστριες Εταιρείες FPGAs	18
2.5 Το πρόβλημα της Τοποθέτησης	29
3. ΑΛΓΟΡΙΘΜΟΣ ΠΡΟΣΟΜΟΙΩΜΕΝΗΣ ΑΝΟΠΤΗΣΗΣ	33
3.1 Γενικά Στοιχεία	33
3.2 Υλοποίηση Αλγορίθμου	34
3.3 Πλεονεκτήματα και Μειονεκτήματα	41
3.4 Παράλληλη Τοποθέτηση	42
3.4.1 Προσέγγιση Παράλληλης Κίνησης	42
3.4.2 Προσέγγιση που βασίζεται στην Περιοχή	44
3.4.3 Ντετερμινιστική Παράλληλη προσέγγιση	45
3.5 Βελτίωση Αλγορίθμου	47
4. ΤΕΤΡΑΓΩΝΙΚΗ ΤΟΠΟΘΕΤΗΣΗ	53
4.1 Γενικά Στοιχεία	53
4.2 Υλοποίηση Αλγορίθμου	53
4.3 Πλεονεκτήματα και Μειονεκτήματα	56
5. MIN-CUT ΤΟΠΟΘΕΤΗΣΗ	57
	4

Μεταπτυχιακή Διατριβή	Γεώργιος Βάσιλας
5.1 Γενικά Στοιχεία	57
5.2 Υλοποίηση Αλγορίθμου	58
5.3 Πλεονεκτήματα και Μειονεκτήματα	58
5.4 Βελτίωση Αλγορίθμου	59
6. ΓΕΝΕΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΜΕ ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ	61
6.1 Γενικά Στοιχεία	61
6.2 Επισκόπηση GASA	61
6.3 Βελτιστοποίηση GASA	61
6.4 Γενετικός Αλγόριθμος	62
6.5 Τοπική βελτιστοποίηση – Προσομοιωμένη ανόπτηση	64
6.6 Πλεονεκτήματα και Μειονεκτήματα	65
7. ΣΥΜΠΕΡΑΣΜΑΤΑ	66
8. ΒΙΒΛΙΟΓΡΑΦΙΑ	68

Λίστα Εικόνων

- Εικόνα 1 : Τυπική Αρχιτεκτονική FPGA
- Εικόνα 2 : Παράδειγμα ενός λογικού μπλοκ που περιέχει δύο LUTs
- Εικόνα 3 : Ροή Σχεδίασης
- Εικόνα 4 : Σχεδιαστικά βήματα με FPGA
- Εικόνα 5 : Λογότυπα των βασικότερων κατασκευαστριών εταιριών FPGA
- Εικόνα 6 : Μεριδίο αγοράς του 2010. (Τα έσοδα σε εκατομμύρια δολάρια (USD))
- Εικόνα 7 : Εύρεση ολικού ελάχιστου της συνάρτησης κόστους $f(T,W,C)$ του Vivado Design Suite
- Εικόνα 8 : Τυπική σχεδιαστική ροή με χρήση VTR
- Εικόνα 9 : Μπλοκ διάγραμμα για το που το XDL εφαρμόζεται στην σχεδιαστική ροή
- Εικόνα 10 : Μπλοκ διάγραμμα του TORC
- Εικόνα 11 : Διάγραμμα ροής ενός κανονικού αλγορίθμου προσομοιωμένης απόπτωσης
- Εικόνα 12 : Παραδοσιακός αλγόριθμος προσομοιωμένης απόπτωσης
- Εικόνα 13 : Συνάρτηση κόστους πλαισίου οριοθέτησης
- Εικόνα 14 : Πλαίσιο οριοθέτησης ενός δικτύου
- Εικόνα 15 : Όριο απόστασης σε υψηλή και χαμηλότερη θερμοκρασία
- Εικόνα 16 : Αρχικά, σε υψηλές θερμοκρασίες, τετράγωνα μακριά από το άλλο μπορούν να ανταλλαχθούν εύκολα. Τέλος, σε χαμηλές θερμοκρασίες, μόνο μπλοκ ένα κοντά στο άλλο μπορούν να ανταλλαχθούν
- Εικόνα 17 : Προσέγγιση της παράλληλης κίνησης
- Εικόνα 18 : Κόστος παρεμπόδισης δικτύου
- Εικόνα 19 : Σύγκρουση σε κατατετμημένη περιοχή
- Εικόνα 20 : Ντετερμινιστική παράλληλη προσέγγιση
- Εικόνα 21 : Αναζήτηση στην γειτονιά
- Εικόνα 22 : Στάδια υλοποίησης του αλγόριθμου κατάτμησης (διαμερισμού)
- Εικόνα 23 : Επιλογή της γραμμής κατάτμησης στο κέντρο της τρέχουσα περιοχής
- Εικόνα 24 : Παράδειγμα παραδοσιακού min-cut τοποθετητή
- Εικόνα 25 : Προτεινόμενη αλληλουχία για εφαρμογή γραμμών κατάτμησης
- Εικόνα 26 : Παράδειγμα τοποθέτησης σε FPGA και του αντίστοιχου χρωμοσώματος
- Εικόνα 27 : Διαδικασία διασταύρωσης

Λίστα Πινάκων

Πίνακας 1 : Λίστα κατασκευαστριών εταιρειών FPGA

Πίνακας 2 : Ψευδοκώδικας από γενικό τοποθετητή βασισμένο σε προσομοιωμένη απόπτηση

Πίνακας 3 : Πίνακας ανανέωσης για το α

Πίνακας 4 : TRY_SWAP συνάρτηση

Πίνακας 5 : Τροποποιημένη TRY_SWAP συνάρτηση

Πίνακας 6 : Εύρος του FACTOR

Πίνακας 7 : Επισκόπηση γενετικού αλγόριθμου με προσομοιωμένη απόπτηση

Πίνακας 8 : Επισκόπηση αλγορίθμων

Εισαγωγή

Από την πρώτη εμφάνιση της το 1984, η «Επιτόπου Προγραμματιζόμενη Διάταξη Πυλών» (Field Programmable Gate Array, εν συντομία FPGA), ένα επαναπρογραμματιζόμενο λογικό τσιπ που μπορεί να ρυθμιστεί να εφαρμόσει πληθώρα ψηφιακών κυκλωμάτων, κέρδισε αρκετή δημοτικότητα εξαιτίας του εξαιρετικά χαμηλού του κόστους, του γρήγορου χρόνου ολοκλήρωσης κατασκευής, των χαμηλών αρχικών εξόδων και την ευκολία σχεδίασης που συνεπάγεται και λιγότερων χρηματοοικονομικών κινδύνων. Συνδυάζοντας τις υψηλές επιδώσεις των «Ενσωματωμένων Κυκλωμάτων Ειδικού Σκοπού» (Application Specific Integrated Circuits, εν συντομία ASIC) με την ευελιξία των επεξεργαστών γενικού σκοπού (CPU) έχουν καταστήσει δυνατή την υλοποίηση εντελώς καινοτόμων εφαρμογών. Εν τούτοις, νέες προκλήσεις διαφανήκανε, από την στιγμή που το μέγεθος των FPGA έχει φθάσει τις εκατομμύρια πύλες. Η σχεδίαση με χρήση FPGA υποφέρει από τους μεγάλους χρόνους τοποθέτησης και ο χρόνος ολοκλήρωσης είναι πολύ σημαντικός. Το πρόβλημα της τοποθέτησης έγινε σημαντικός ρυθμιστής της απόδοσης του κυκλώματος στο FPGA. Για τις επόμενες γενιές των εργαλείων CAD (Computer-Aided-Design) (Σχεδιασμός με βοήθεια Υπολογιστή) των FPGA, γρήγοροι και ποιοτικοί μέθοδοι τοποθέτησης είναι αναγκαίοι. Στην μελέτη αυτή χρησιμοποιούμε το island style (αρχιτεκτονική νησίδας) μοντέλο του FPGA. Όπως θα δούμε και αργότερα η γενική κατασκευή της αρχιτεκτονικής νησίδας (island style) FPGA περιλαμβάνει τέσσερα βασικά μέρη: Configurable Logic Blocks (CLB Προγραμματιζόμενα Λογικά Μπλοκ), που τυπικά περιέχουν είτε συνδυαστικά είτε ακολουθιακά λογικά κυκλώματα. Μπλοκ εισόδου/εξόδου (IOB) που είναι οι συνδέσεις του FPGA και εξωτερικών συσκευών. Το μπλοκ σύνδεσης χρησιμοποιείται για να συνδεθεί ένα CLB στα κανάλια δρομολόγησης ενώ το μπλοκ διακόπτη χρησιμοποιείται για να συνδέσει τα κανάλια δρομολόγησης [6]. Στο στάδιο τοποθέτησης, η συνδεσμολογία του κυκλώματος (netlist) των λογικών μπλοκ τοποθετείται σε κύκλωμα FPGA [6]. Ο στόχος της βελτιστοποίησης της τοποθέτησης είναι να τοποθετηθούν τα μπλοκ σε μια κατάλληλη θέση, έτσι ώστε η αντικειμενική συνάρτηση να ελαχιστοποιείται. Υπάρχουν τρία κοινά κριτήρια βελτιστοποίησης για την τοποθέτηση, την τοποθέτηση με γνώμονα τον χρονοισμό, την τοποθέτηση

με γνώμονα το μήκος της καλωδίωσης και την τοποθέτηση με γνώμονα την δυνατότητα δρομολόγησης. Οι τρεις βασικές κλάσεις από τοποθετητές που χρησιμοποιούνται αυτόν τον καιρό είναι οι min-cut (αλγόριθμοι που βασίζονται στην κατάτμηση), οι αναλυτικοί (αλγόριθμοι που αντιπροσωπεύουν το πρόβλημα της τοποθέτησης με ένα σύστημα εξισώσεων, το οποίο στην συνέχεια επιλύεται αριθμητικά) και αυτοί που βασίζονται στην προσομοιωμένη ανόπτηση. Τα περισσότερα εμπορικά εργαλεία για FPGA εφαρμόζουν αλγόριθμους βασισμένους στην προσομοιωμένη ανόπτηση, εξαιτίας της ευελιξίας του αλγορίθμου να εφαρμόζει ένα μεγαλύτερο φάσμα κριτηρίων βελτιστοποίησης. Υπάρχουν βέβαια αρκετές έρευνες (κυρίως ακαδημαϊκές) που είτε χρησιμοποιούνται συνδυασμοί από αλγόριθμους των παραπάνω κλάσεων, είτε και κάποιοι άλλοι αλγόριθμοι που βασίζονται σε μεθευρετικούς, εξελεκτικούς και γενετικούς αλγόριθμους. Η εξεύρεση της βέλτιστης τοποθέτησης είναι υπολογιστικά ανέφικτη αφού είναι ένα πρόβλημα NP-πολυπλοκότητας, οπότε μπορεί μόνο να προσεγγιστεί από ευρετικές μεθόδους όπως αυτές που αναφέραμε νωρίτερα. Η μελέτη εκπονήθηκε στα πλαίσια της διπλωματικής εργασίας που δόθηκε από τον καθηγητή κύριο Ψαράκη Μιχαήλ στο πλαίσιο του προγράμματος μεταπτυχιακών σπουδών «Προηγμένα Συστήματα Πληροφορικής» με κατεύθυνση «Τεχνολογία Ενσωματωμένων Υπολογιστικών συστημάτων». Η δομή της διπλωματικής αυτής εργασίας ξεκινάει με την εισαγωγή. Στο δεύτερο κεφάλαιο αναλύουμε βασικές έννοιες στα FPGAs και γίνεται αναφορά στο τι είναι FPGA, την αρχιτεκτονική του, τα εργαλεία CAD, τις κατασκευάστριες εταιρίες των FPGA, τα εργαλεία CAD του εμπορείου των εταιριών, όπως και των κυριότερων open-source εργαλείων που υπάρχουν σήμερα και στο τέλος αναφερόμαστε στην τοποθέτηση που είναι και πρακτικά το ζητούμενο της εργασίας. Στα επόμενα κεφάλαια αναλύονται οι κυριότεροι αλγόριθμοι που χρησιμοποιούνται σήμερα. Ξεκινάμε με τον αλγόριθμο προσομοιωμένης ανόπτησης (Simulated Annealing Algorithm) που όπως αναφέρθηκε νωρίτερα είναι ο πιο δημοφιλής αλγόριθμος που χρησιμοποιείται. Μιμείται την διαδικασία ανόπτησης των μετάλλων. Η ανόπτηση είναι μια φυσική διαδικασία στην οποία θερμαίνουμε το στερεό και ύστερα το ψύχουμε αργά μέχρι να κρυσταλλοποιηθεί. Αρχικά γίνεται μία τυχαία αρχική τοποθέτηση για όλα τα λογικά μπλοκ. Στην συνέχεια, τυχαία επιλέγονται ζευγάρια λογικών μπλοκ ως υποψήφια για να ανταλλαχθούν ώστε να βελτιωθεί η συνάρτηση κόστους. Αν η ανταλλαγή επιδρά στην μείωση της συνάρτησης κόστους, αυτόματα επιτρέπεται, αλλιώς επιτρέπεται μόνο με την πιθανότητα να μειωθεί όπως ο αλγόριθμος προχωράει, επιτρέποντας έτσι λιγότερες «επώδυνες» ανταλλαγές μετά από κάθε επανάληψη. Στο επόμενο κεφάλαιο βλέπουμε την τετραγωνική τοποθέτηση. Η μέθοδος τετραγωνικής τοποθέτησης χρησιμοποιεί το τετράγωνο του μήκους

καλωδίωσης ως στοιχείο ελέγχου στην συνάρτηση που θέτει και προσπαθεί να ελαχιστοποιήσει το κόστος με την επίλυση γραμμικών εξισώσεων. Στο επόμενο κεφάλαιο βλέπουμε τον αλγόριθμο κατάτμησης min-cut. Πρακτικά ένα κύκλωμα κατατέμνεται αναδρομικά στην μέση. Η τεχνική ελαχιστοποιεί τον αριθμό των κατατμήσεων των δικτύων ενώ εν τω μεταξύ, αφήνει τα υψηλά συνδεόμενα λογικά μπλοκ σε ένα διαμέρισμα. Στο επόμενο κεφάλαιο βλέπουμε έναν υβριδικό αλγόριθμο που χρησιμοποιεί γενετικό αλγόριθμο και προσομοιωμένη απόπτωση. Αρχικά χρησιμοποιούμε τον γενετικό αλγόριθμο για ολική βέλτιστη λύση και στην συνέχεια την προσομοιωμένη απόπτωση για βελτίωση τοπικά. Ο γενετικός αλγόριθμος είναι μια τεχνική αναζήτησης με βάση τους μηχανισμούς της φυσικής γενετικής και της φυσικής επιλογής, η οποία ενσωματώνει μια προσομοίωση της εξέλιξης ως μια ευρετική αναζήτηση, ψάχνοντας μια καλή λύση. Τέλος γίνεται μία επισκόπηση όλων των αλγορίθμων που αναφέρθηκαν.

Βασικές έννοιες στα FPGAs

2.1 Ορισμός

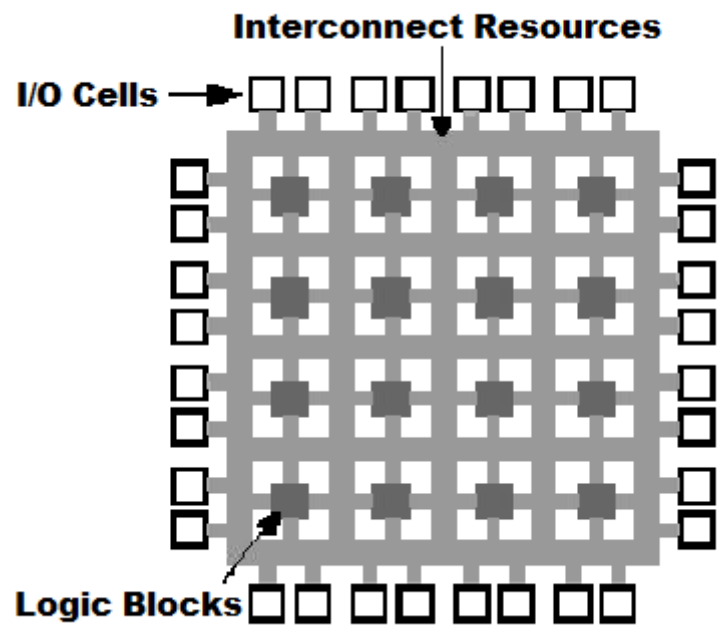
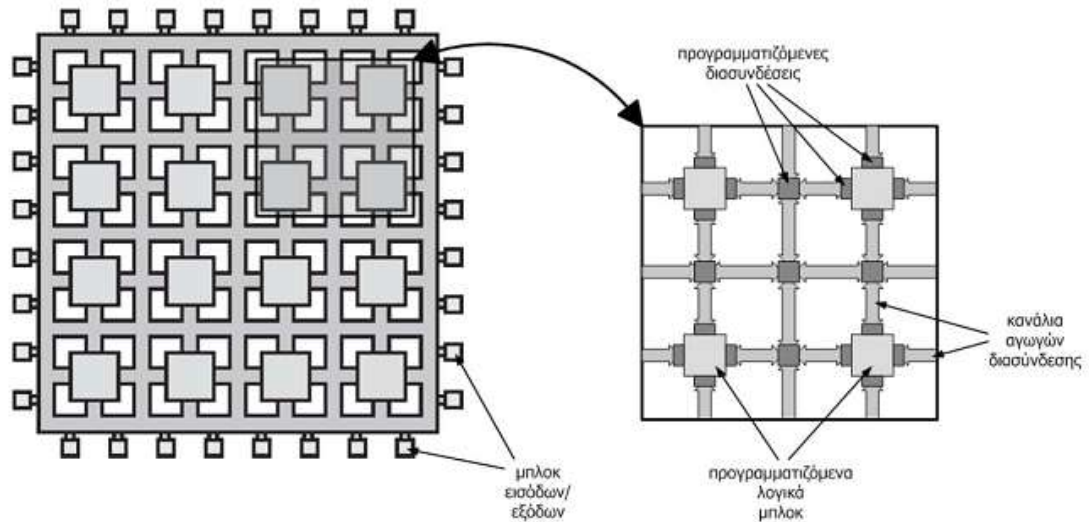
Σύμφωνα με τον Tessier [34] τα Field-Programmable Gate Arrays (FPGAs) ορίζονται ως

«Κυκλώματα ημιαγωγών (ημιαγώγιμη συσκευή) που περιέχουν προγραμματιζόμενη λογική (logic blocks) και προγραμματιζόμενες διασυνδέσεις».

Με σκοπό να παρουσιάζουν τη λειτουργικότητα απλών λογικών πυλών (“and”, “or”, “xor”, “not”) ή τη λειτουργικότητα περισσότερο σύνθετων συνδυαστικών συναρτήσεων τα στοιχεία προγραμματιζόμενης λογικής μπορούν να προγραμματιστούν με την αποκωδικοποίηση (αποκωδικοποιητές), την άθροιση (αθροιστές) και άλλες μαθηματικές συναρτήσεις. Επίσης, στα στοιχεία αυτά περιλαμβάνονται πολλές φορές και στοιχεία μνήμης τα οποία είτε είναι απλά “flip-flops”, είτε είναι πλήρη τμήματα (μπλοκ - blocks) μνήμης. Όλα αυτά τα στοιχεία μπορούν να διασυνδεθούν σύμφωνα με τις απαιτήσεις της εφαρμογής την οποία καλούνται να υλοποιήσουν με τη βοήθεια της υπάρχουσας ιεραρχίας των προγραμματιζόμενων διασυνδέσεων. Αυτή η λειτουργικότητα θα μπορούσε να παρομοιασθεί με εκείνη ενός προγραμματιζόμενου “breadboard” [6].

2.2 Αρχιτεκτονική FPGA

Ένα σύγχρονο FPGA, δομείται από τη μία με έναν δισδιάστατο πίνακα και από την άλλη με προγραμματιζόμενα λογικά μπλοκ (CLBs) , από μπλοκ σταθερής λειτουργίας και πηγές δρομολόγησης υλοποιημένες στη τεχνολογία CMOS. Κατά μήκος της περιμέτρου του FPGA υπάρχουν ειδικά λογικά μπλοκ συνδεδεμένα με εξωτερικές συσκευασίες ακροδεκτών I/O. Τα λογικά μπλοκ συναποτελούνται από πολλαπλά λογικά κελιά, ενώ τα λογικά κελιά εμπεριέχουν γεννήτριες συναρτήσεων και αποθηκευτικά στοιχεία.



Εικόνα 1 Τυπική Αρχιτεκτονική FPGA

Γεννήτριες Συναρτήσεων

Γεννήτριες συναρτήσεων χρησιμοποιούν οι συσκευές FPGA προς υλοποίηση της Boolean λογικής και όχι φυσικές πύλες σε αντίθεση με τις υπόλοιπες συσκευές προγραμματιζόμενης λογικής (PLAs, PALs, CPLDs). Όσον αφορά τις εισόδους μιας Boolean συνάρτησης αρχικά δομείται ένας πίνακας αλήθειας. Για κάθε είσοδο ο πίνακας αλήθειας περιγράφει την τιμή της εξόδου της συνάρτησης. Επιπλέον, κάθε μπιτ (bit) της εξόδου της συνάρτησης αποθηκεύεται σε ξεχωριστό κελί μιας στατικής μνήμης. Η σύνδεση των κελιών αυτών ως είσοδοι σε έναν πολυπλέκτη επιλέγεται ανάλογα με την είσοδο της κατάλληλης εξόδου. Τα αποτελέσματα είναι γνωστά ως Look-Up Table (LUT) (πίνακες αναζήτησης). Σε αντίθεση με τα ψηφιακά κυκλώματα που υλοποιούνται με λογικές πύλες, η καθυστέρηση διάδοσης στα LUT παραμένει σταθερή. Αυτό σημαίνει ότι ανεξάρτητα από την πολυπλοκότητα του Boolean κυκλώματος, εφόσον “χωράει” σε ένα LUT, η καθυστέρηση διάδοσης είναι η ίδια. Αυτό επίσης ισχύει και για κυκλώματα που εκτείνονται σε περισσότερα LUT, αλλά εδώ η καθυστέρηση διάδοσης εξαρτάται και από τον αριθμό των LUT που χρησιμοποιούνται.

Στοιχεία Αποθήκευσης

Αρχικώς, ενώ οι γεννήτριες συναρτήσεων αποτελούν το βασικό μπλοκ για την υλοποίηση συνδυαστικών κυκλωμάτων, τα FPGA διαθέτουν πρόσθετα στοιχεία παρέχοντας έτσι ένα πλούτο λειτουργιών. Όπως και στα μπλοκ των PLA, D Flip-Flop ενσωματώνονται έτσι ανάλογα ενσωματώνονται και στα FPGA. Τα Flip-Flops μπορούν να χρησιμοποιηθούν με διάφορους τρόπους. Ο πιο απλός γίνεται με την αποθήκευση δεδομένων. Συνήθως, μια έξοδος του LUT συνδέεται με την είσοδο ενός Flip-Flop. Επίσης, το Flip-Flop μπορεί να χρησιμοποιηθεί και ως μανδαλωτής που λειτουργεί είτε σε θετική είτε σε αρνητική λογική.

Λογικά Κελιά

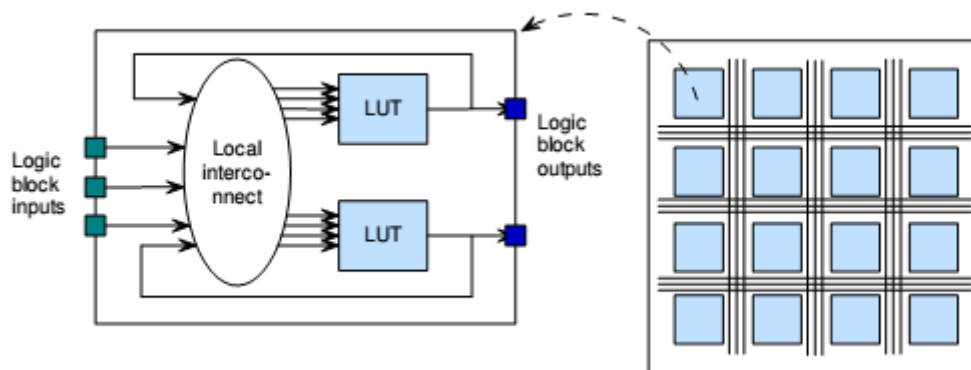
Συνδυάζοντας ένα LUT και ένα D Flip-Flop αυτό που προκύπτει ως αποτέλεσμα, αναφέρεται ως λογικό κελί. Τα λογικά κελιά αποτελούν το χαμηλότερο επίπεδο του δομικού μπλοκ σε ένα FPGA. Τόσο η συνδυαστική όσο και η ακολουθιακή λογική μπορούν να υλοποιηθούν από ένα λογικό κελί ή μια συλλογή από αυτά.

Λογικά Μπλοκ

Τα λογικά κελιά θεωρούνται ως τα βασικά δομικά μπλοκ των εφαρμογών FPGA, που στην πραγματικότητα είναι σύνηθες να ομαδοποιούμε μερικά λογικά κελιά σε μπλοκ και να προσθέτουμε κυκλώματα ειδικού σκοπού με σκοπό να δημιουργήσουμε ένα λογικό μπλοκ. Το παραπάνω επιτρέπει σε μια ομάδα από λογικά κελιά, τα οποία βρίσκονται κοντά μεταξύ τους,

να έχουν γρήγορο δίαυλο επικοινωνίας, μειώνοντας έτσι την καθυστέρηση διάδοσης και βελτιώνοντας σημαντικά την υλοποίηση του σχεδίου. Για παράδειγμα, η οικογένεια Virtex-5 της Xilinx ομαδοποιεί τέσσερα λογικά κελιά σε ένα slice. Δυο slices και ένα carry-logic σχηματίζουν ένα προγραμματιζόμενο λογικό μπλοκ (Configurable Logic Block, CLB).

Τα λογικά κελιά, συνδέονται με ένα δίκτυο διασυνδέσεων ώστε να παρέχουν υποστήριξη για πιο πολύπλοκες υλοποιήσεις κυκλωμάτων. Αυτό το δίκτυο διασυνδέσεων αποτελείται από switch boxes. Ένα switch box χρησιμοποιείται για τη δρομολόγηση μεταξύ των εισόδων/εξόδων ενός λογικού κελιού με το γενικό δίκτυο δρομολόγησης του τσιπ. Επίσης θεωρείται υπεύθυνο για τη διέλευση σημάτων από ένα τμήμα καλωδίωσης σε ένα άλλο και έτσι τα τμήματα καλωδίωσης μπορεί να είναι είτε μικρά, συνδέοντας μόνο δυο λογικά κελιά είτε μεγάλα, διατρέχοντας όλη την επιφάνεια του τσιπ.



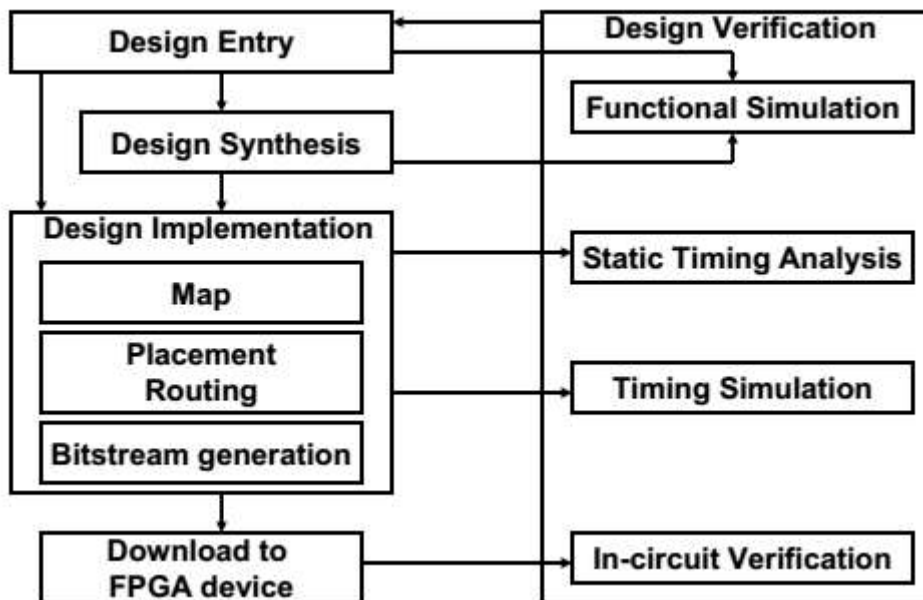
Εικόνα 2 Παράδειγμα ενός λογικού μπλοκ που περιέχει δύο LUTs

Μπλοκ Εισόδου/Εξόδου

Οι εξωτερικοί ακροδέκτες της συσκευής συνδέουν τον πίνακα των λογικών μπλοκ και τις πηγές δρομολογήσεων με τα μπλοκ εισόδου/εξόδου (Input/Output Blocks – IOBs), τα οποία υφίστανται περιμετρικά του τσιπ.

2.3 CAD

Τα FPGA υλοποιούνται χρησιμοποιώντας ένα μεγάλο αριθμό από προγραμματιζόμενες διακόπτες που χρησιμοποιούνται για να υλοποιήσουν μία συγκεκριμένη λογική συνάρτηση. Τα εργαλεία CAD μετατρέπουν την σχεδίαση, που εισέρχεται είτε με την μορφή σχηματικού (δεν είναι τόσο δημοφιλής όσο τα κυκλώματα γίνονται πιο περίπλοκα και τα εργαλεία των γλωσσών βελτιώνονται), είτε με χρήση γλώσσας περιγραφής υλικού HDL (συνήθως VHDL ή Verilog), σε μία σειρά από άσσους και μηδενικά που προγραμματίζουν το FPGA κατά την διάρκεια του χρόνου διαμόρφωσης. Η όλη διαδικασία ανάπτυξης υλικού δίνετε στο παρακάτω σχήμα.



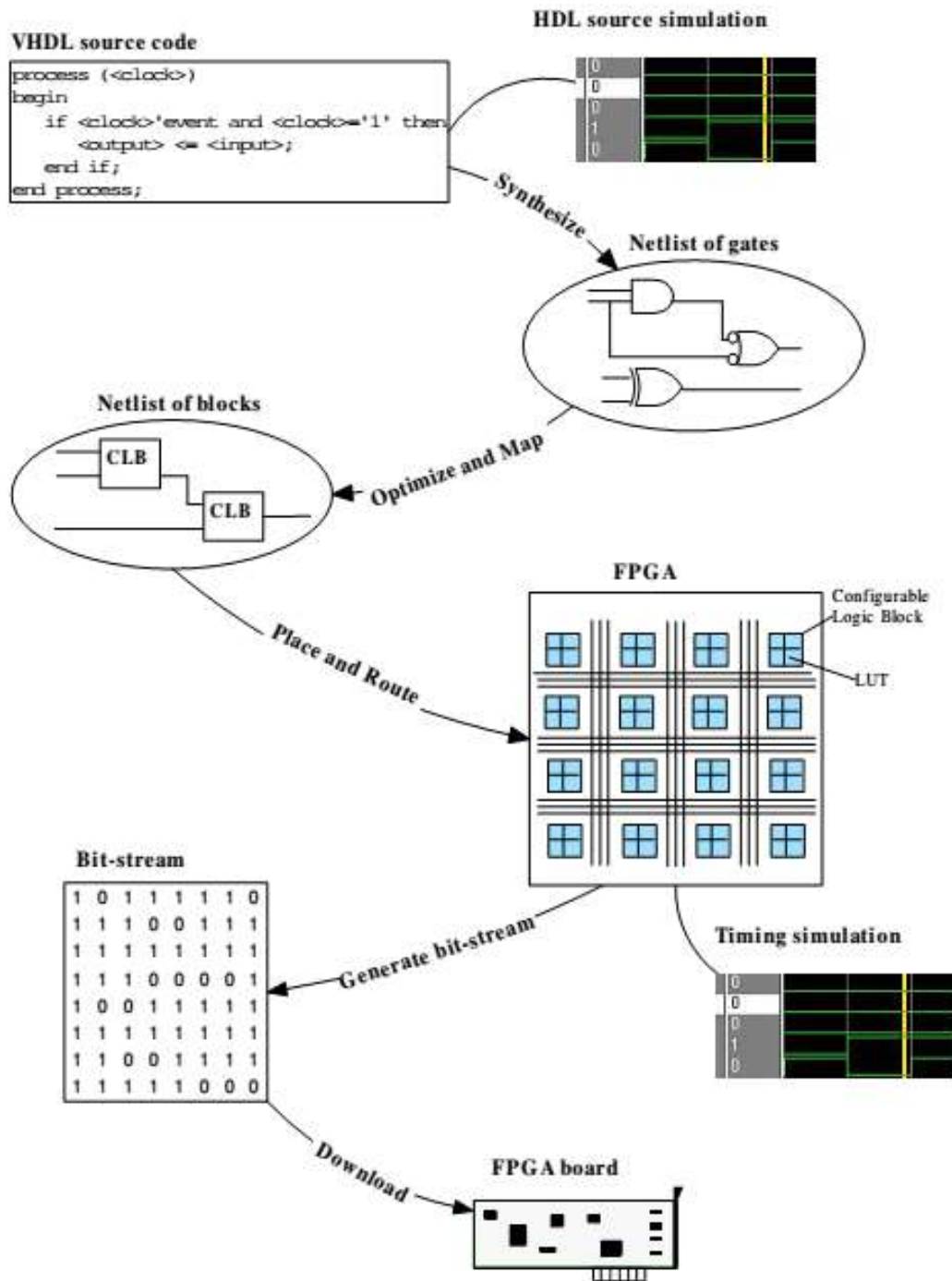
Εικόνα 3 Ροή Σχεδίασης [29]

Ίσως η πιο χτυπητή διαφορά μεταξύ σχεδιασμού υλικού (hardware) και λογισμικού (software) είναι ο τρόπος που πρέπει να σκέφτεται ο υπεύθυνος ανάπτυξης για ένα πρόβλημα. Οι σχεδιαστές λογισμικού τείνουν να σκέφτονται ακολουθιακά, ακόμα και όταν αναπτύσσουν μια πολύπλοκη εφαρμογή. Οι γραμμές του κώδικα που γράφουν εκτελούνται κατά αυτό τον τρόπο πάντα. Κατά την διάρκεια της εισόδου της σχεδίασης, οι σχεδιαστές υλικού πρέπει να σκέφτονται και να προγραμματίζουν παράλληλα. Όλα τα σήματα εισόδου επεξεργάζονται παράλληλα, όπως ταξιδεύουν μέσα από ένα σύνολο μηχανών επεξεργασίας (κάθε μία μια σειρά από καταχωρητές και διασυνδέσεις) προς τον προορισμό τους, τα σήματα εξόδου. Γι' αυτό το λόγο, οι αναφορές των γλωσσών περιγραφής υλικού δημιουργούν δομές, οι οποίες εκτελούνται ταυτόχρονα.

Τυπικά, η βαθμίδα της εισαγωγής της σχεδίασης ακολουθείται από περίοδο προσομοίωσης. Εδώ χρησιμοποιείται ένας προσομοιωτής για να εκτελέσει την σχεδίαση και να επιβεβαιώσει ότι παράγονται οι σωστές έξοδοι για το δοσμένο σύνολο των δοκιμαστικών εισόδων. Αν και προβλήματα με το μέγεθος και τον χρονισμό του υλικού μπορεί να προκύψουν αργότερα, ο σχεδιαστής μπορεί τουλάχιστον να είναι σίγουρος ότι η λογική του είναι συναρτησιακά ορθή πριν κάνει το επόμενο βήμα της υλοποίησης.

Η προσομοίωση αρχίζει μόνο αφού υπάρξει μια λειτουργική παρουσίαση του υλικού. Η προσομοίωση του υλικού αποτελείται από δύο διακριτά μέρη. Πρώτα παράγεται μια ενδιάμεση παρουσίαση του υλικού σχεδίου. Αυτό το βήμα ονομάζεται σύνθεση (synthesis) και το αποτέλεσμα της είναι η λεγόμενη συνδεσμολογία του κυκλώματος (netlist). Η συνδεσμολογία του κυκλώματος είναι ανεξάρτητη της συσκευής, έτσι τα περιεχόμενά της δεν βασίζονται των FPGA και των CPLD. Συνήθως αποθηκεύεται σε μια κοινή μορφή που ονομάζεται EDIF (Electronic Design Interchange Format).

Το δεύτερο βήμα είναι η διαδικασία της μετάφρασης που ονομάζεται τοποθέτηση και δρομολόγηση (place and route). Αυτό το βήμα περιέχει την αντιστοίχιση των λογικών δομών που περιγράφηκαν στην συνδεσμολογία του κυκλώματος σε πραγματικούς καταχωρητές, αλληλοσυνδέσεις και ακίδες εισόδου και εξόδου. Αυτή η διαδικασία είναι παρόμοια του ισότιμου βήματος της δημιουργίας τυπωμένων κυκλωμάτων. Το αποτέλεσμα της διαδικασίας τοποθέτησης και δρομολόγησης (place and route) είναι μία αλληλουχία μπιτ (bitstream). Το όνομα αυτό χρησιμοποιείται γενικά, παρά το γεγονός ότι κάθε CPLD ή FPGA έχει τη δική του μοναδική δομή αλληλουχίας μπιτ (bitstream).



Εικόνα 4 Σχεδιαστικά βήματα με FPGA

Τα εργαλεία CAD διαδραματίζουν σημαντικό ρόλο στη συμπεριφορά μίας σχεδίασης FPGA. Η μεγαλύτερη πρόκληση που αντιμετωπίζουν τα σημερινά εργαλεία CAD είναι η ανάγκη για την παραγωγή υψηλής ποιότητας τοποθετήσεων και δρομολογήσεων για όλο και μεγαλύτερα κυκλώματα. Η χωρητικότητα των FPGA διπλασιάζεται κάθε δύο με τρία χρόνια, διπλασιάζοντας το μέγεθος των προβλημάτων αυτών με τον ίδιο ρυθμό. Επιπλέον, η ταχύτητα μονοεπεξεργαστικά δεν αυξάνεται πλέον τόσο γρήγορα όπως γινόταν στο παρελθόν, πράγμα που σημαίνει ότι και μόνο η ταχύτητα του επεξεργαστή θα αυξηθεί κατά λιγότερο από δύο φορές κατά την ίδια περίοδο. Προκειμένου να διατηρηθεί ο γρήγορος χρόνος διάθεσης στην αγορά, και την ευκολία χρήσης που παρέχουν ιστορικά τα FPGAs, οι αλγόριθμοι τοποθέτησης και δρομολόγησης δεν μπορούν να επιτρέπουν να πάρουν όλο και περισσότερο χρόνο της CPU. Υπάρχει επομένως επιτακτική ανάγκη για αλγόριθμους που να είναι πολύ επεκτάσιμοι αλλά να εξακολουθούν να παράγουν αποτελέσματα υψηλής ποιότητας. Ο χάρτης πορείας για τους μελλοντικούς μικροεπεξεργαστές δείχνει ότι ο αριθμός των ανεξάρτητων πυρήνων ή νημάτων σε ένα μόνο τσιπ θα αυξηθεί ραγδαία τα επόμενα χρόνια. Κατά συνέπεια, οι περισσότεροι μηχανικοί θα έχουν παράλληλους επεξεργαστές στους επιτραπέζιους υπολογιστές τους. Μέρος της λύσης του προβλήματος της να κρατήσουμε τον χρόνο τοποθέτησης FPGA σε λογικό χρόνο μπορεί να είναι να βρούμε τεχνικές και αλγόριθμους για την εκμετάλλευση της παράλληλης επεξεργασίας χωρίς να θυσιάζεται η ποιότητα του αποτελέσματος. Επιπλέον, νέες αρχιτεκτονικές FPGA (όπως η *three-dimensional chip stacking* (τρισδιάστατο στοίβαγμα τσιπ)) μπορούν να προσφέρουν μεγαλύτερη ευελιξία και να απελευθερωθούμε από τους σημερινούς περιορισμούς.

2.4 Κατασκευάστριες Εταιρείες FPGAs

Τα τελευταία χρόνια υπάρχουν πέντε μεγάλες εταιρείες παγκοσμίως που κατασκευάζουν FPGAs. Οι δυο κυριότερες που επικρατούν στην αγορά είναι η Xilinx και η Altera

Πίνακας 1 Λίστα κατασκευαστριών εταιρειών FPGA

Xilinx	Η Xilinx παραδοσιακά είναι ο οδηγός στον τεχνολογικό τομέα πυριπίου (<i>silicon technology</i>). Η φιλοσοφία της Xilinx είναι η παραγωγή όλων των πιθανών τύπων ασχέτως του βαθμού πολυπλοκότητας.
Altera	Η φιλοσοφία της Altera είναι η παραγωγή των τύπων συσκευών με την μεγαλύτερη ζήτηση, αλλά και με ευκολία στη χρήση .Συσκευές με αποδοτικές και μικρού πάχους αρχιτεκτονικές .
Lattice	Γνωστή για τα CPLDs και έχει ακόμη την σειρά των "instant-on" FPGA

Microsemi Προϊόντα προγραμματισμού μιας χρήσεως

(πρώην Actel)

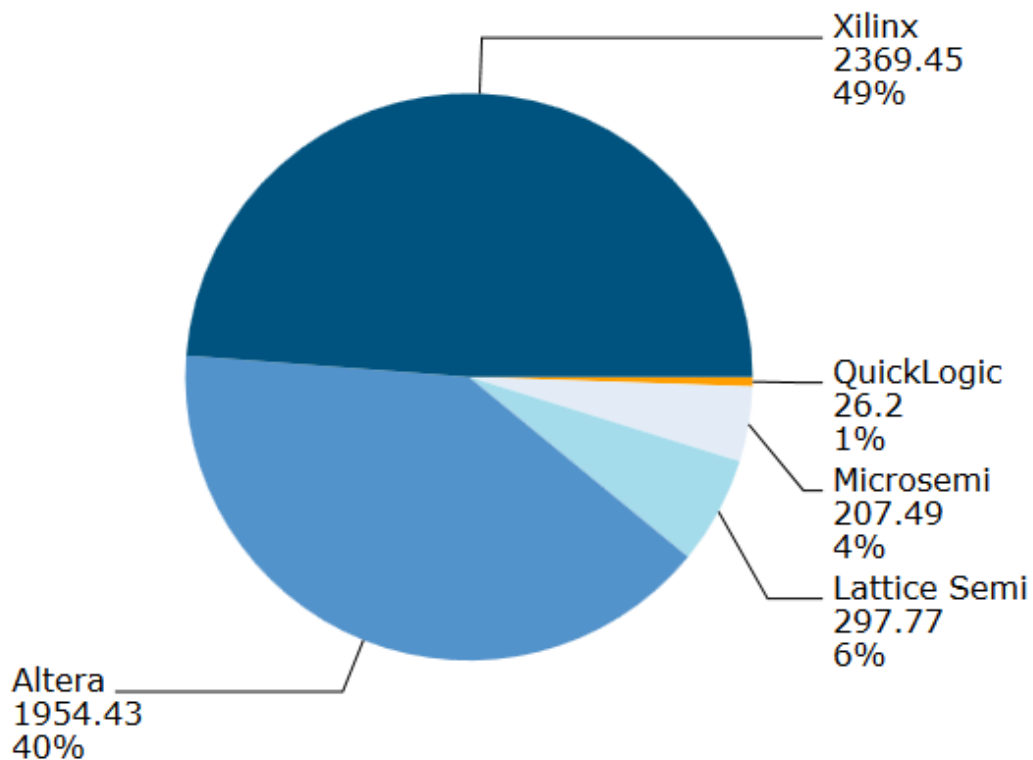
SiliconBlue Technologies Πολύ χαμηλής κατανάλωσης FPGA με προαιρετική ενσωματωμένη μη πτητική μνήμη. Αποκτήθηκε από την Lattice το 2011

Achronix Προϊόντα προγραμματισμού μιας χρήσεως

Quicklogic Προϊόντα προγραμματισμού μιας χρήσεως



Εικόνα 5 Λογότυπα των βασικότερων κατασκευαστριών εταιριών FPGA

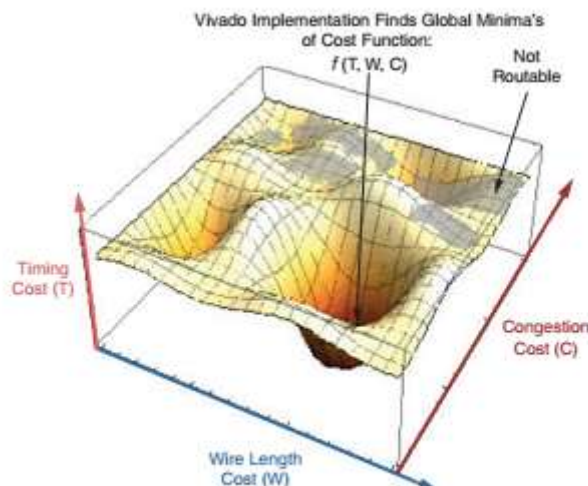


Εικόνα 6 Μεριδίο αγοράς του 2010. (Τα έσοδα σε εκατομμύρια δολάρια (USD))

Όλες οι εταιρείες έχουν τα δικά τους εμπορικά εργαλεία σχεδίασης CAD. Η αντίθεση των εμπορικών προγραμμάτων των εταιριών, από αυτά που θα δούμε παρακάτω του ανοιχτού κώδικα είναι ότι ο οποιοσδήποτε δεν μπορεί να μοιραστεί το πρόγραμμα με τους άλλους, δεν μπορεί να κάνει αλλαγές σε αυτό, ακόμα και κάποιο ελαττώματα να παρουσιαστεί πρέπει να διορθωθεί από την κατασκευάστρια εταιρία. Επίσης δεν μπορούμε να κάνουμε «reverse engineering» για να δούμε ποιες τεχνικές χρησιμοποιούνται, γιατί κάτι τέτοιο είναι παράνομο και γενικότερα έχουμε ένα πρόγραμμα που κάνει το ζητούμενο, χωρίς όμως να γνωρίζουμε το πώς και το γιατί. Οι δυο μεγάλες εταιρίες που δραστηριοποιούνται στα FPGA έχουν τα δικά τους γνωστά εργαλεία. Η Altera χρησιμοποιεί Altera Quartus II σχεδιαστικό λογισμικό που ως τοποθετητή χρησιμοποιεί τον αλγόριθμο του VPR (προσομοιωμένη ανόπτηση). Στην τελευταία έκδοση του χρησιμοποιείται μία πιο παράλληλη μορφή του αλγορίθμου, ίδια περίπτωση με αυτή που περιγράφουμε παρακάτω, την ντετερμινιστική παράλληλη εκδοχή.

Η Xilinx που έχει το πακέτο σχεδίασης VIVADO (συνεχιστής του γνωστού ISE) χρησιμοποιεί έναν αναλυτικό αλγόριθμο για τοποθέτηση. Σύμφωνα με το εγχειρίδιο του πακέτου VIVADO[13], οι αλγόριθμοι προσομοιωμένης ανόπτησης λόγω της τυχαίας φύσης της αρχικής λύσης και των ακολουθιακών κινήσεων δεν κλιμακώνεται σε σχεδιασμούς εκατομμύριων LUT. Επειδή λειτουργεί με τις τοπικές κινήσεις, αυτή η παραδοσιακή μέθοδος βελτιστοποίησης είναι τυφλή στις μετρήσεις ολικού σχεδιασμού, όπως το επίπεδο της συμφόρησης της λύσης ή του συνολικού μήκος της καλωδίωσης.

Το Vivado Design Suite μοντελοποιεί τη λύση τοποθέτησης χώρου σε μία μεγάλη μαθηματική εξίσωση και χρησιμοποιεί μια αναλυτική λύση για να βρούμε μια λύση που ελαχιστοποιεί μια δεδομένη συνάρτηση κόστους. Μια βέλτιστη λύση τοποθέτησης εξαρτάται από πολλαπλές διαστάσεις, όπως χρονισμού (T), το μήκος της καλωδίωσης (W), και μετρήσεις συμφόρησης (C). Το Vivado Design Suite χρησιμοποιεί μια συνάρτηση κόστους πολλών μεταβλητών για να βρει τη βέλτιστη τοποθέτηση, επιτρέποντας στο σχεδιαστή να βρεθεί γρήγορα μια λύση με δυνατότητα δρομολόγησης που μεγιστοποιεί την απόδοση ελαχιστοποιώντας ταυτόχρονα το μήκος της καλωδίωσης, και ως εκ τούτου την δυναμική ισχύ (Δυναμική ή ενεργή ισχύς είναι το ποσό της ενέργειας που καταναλώνει η συσκευή όταν είναι σε ενεργή λειτουργία). Αυτό απεικονίζεται στην παρακάτω εικόνα



Εικόνα 7 Εύρεση ολικού ελάχιστου της συνάρτησης κόστους : $f(T, W, C)$ του Vivado Design Suite [13]

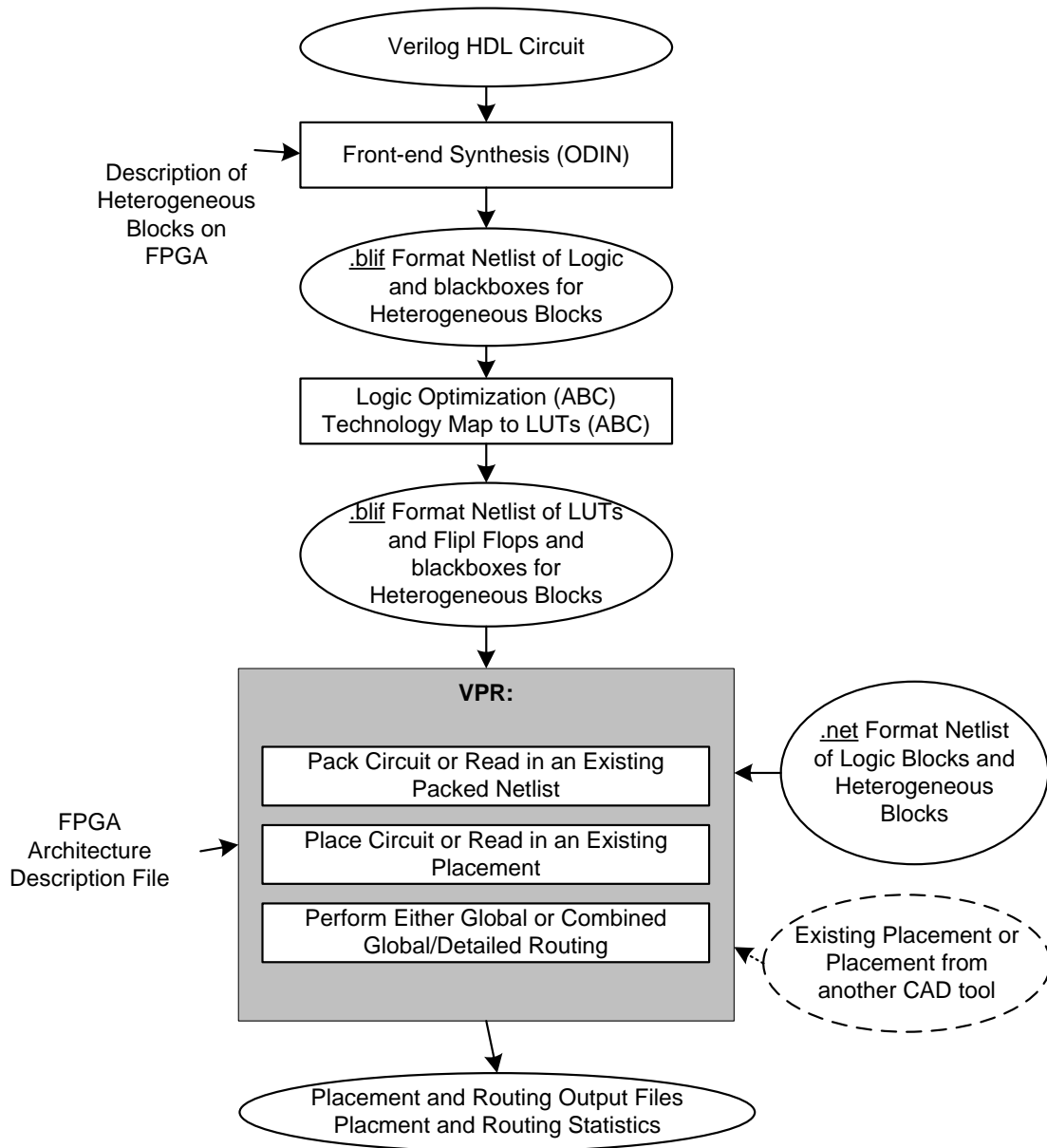
Ο John Wawrzynek, καθηγητής του Πανεπιστημίου Berkeley των Ηνωμένων Πολιτειών της Αμερικής στο 19^ο διεθνές συμπόσιο για τα FPGAs της ACM/SIGDA έφτιαξε μία επιτροπή με την ερώτηση «μήπως θα έπρεπε η ακαδημαϊκή κοινότητα να ξεκινήσει την προσπάθεια μίας ανοιχτού κώδικα FPGA συσκευής και εργαλείων;» [35]. Αναφέρει ότι για χρόνια, πολλοί ακαδημαϊκοί ερευνητές στην επαναπρογραμματιζόμενη υπολογιστική έχουν απογοητευτεί από την εξάρτησή τους από τα εμπορικά FPGAs και τα εργαλεία τους. Τα εμπορικά FPGAs έχουν εξαιρετικά πολύπλοκες μικρο-αρχιτεκτονικές, έρχονται με ατεκμηρίωτες binary interfaces, δεν έχουν συμβατότητα μεταξύ των γενεών, και έρχονται με δύσκολα στην χρησιμοποίησή τους ιδιόκτητα εργαλεία τοποθέτησης και δρομολόγησης (place and route tools). Οι εταιρίες FPGA κάνουν τις σωστές κινήσεις για την εξυπηρέτηση της εμπορικής πελατειακής τους βάσης, αλλά φαίνεται ότι μερικές φορές οι κινήσεις αυτές έρχονται σε αντίθεση με τις ανάγκες της ακαδημαϊκής ερευνητικής κοινότητας. Τα προβλήματα αυτά καθιστούν δύσκολο για τους επιστήμονες να διδάξουν σχεδιασμό FPGA και να συμμετάσχουν σε σχετικές έρευνες που σχετίζονται με τα FPGAs. Πολύπλοκες κατασκευές FPGA και ατεκμηρίωτες ιδιόκτητες διεπαφές καθιστούν σχεδόν αδύνατο να οικοδομήσουμε νέα εργαλεία με στόχο τις υπάρχουσες εμπορικές συσκευές. Ερευνητές του πανεπιστημίου είναι πλέον διστακτικοί στο να συμμετάσχουν σε FPGA αρχιτεκτονική έρευνα, επειδή το επίπεδο της πολυπλοκότητας των FPGAs είναι πέρα από ό, τι είναι δυνατόν να εφαρμοστεί σε ένα ακαδημαϊκό περιβάλλον. Προγραμματιστές επαναπρογραμματιζόμενων υπολογιστικών εφαρμογών είναι στο έλεος των σύνθετων συσκευών FPGA και του κλειστού κώδικα των εργαλείων τοποθέτησης και δρομολόγησης (place and route tools) – που δεν έχει βελτιστοποιηθεί για υπολογιστικές εφαρμογές. Ορισμένοι υποστηρίζουν ότι η στιγμή είναι

κατάλληλη για την ακαδημαϊκή κοινότητα να ξεφύγει και να δημιουργήσει μια ανοικτή κώδικα συσκευή FPGA. Μια τέτοια προσπάθεια θα επιτρέψει στους πανεπιστημιακούς ερευνητές να αναλάβουν ενεργό ρόλο στο σχεδιασμό της αρχιτεκτονικής, και να προωθήσουν μια προσπάθεια για μία κοινότητα ανοιχτού κώδικα, για την ανάπτυξη και την προώθηση της κατάστασης της τεχνολογίας των εργαλείων. Μια παρόμοια κατάσταση υπήρχε στις αρχές της δεκαετίας του 80 γύρω από το σχεδιασμό του επεξεργαστή. Ως αποτέλεσμα του ανταγωνισμού μεταξύ των μεγάλων εταιρειών πληροφορικής, τα σχέδια μικροεπεξεργαστών είχαν εξελιχθεί από πολύ απλές αρχές μέχρι πολύπλοκες αρχιτεκτονικές και εξίσου πολύπλοκους μεταγλωττιστές. Τα πανεπιστήμια ήταν σε μεγάλο βαθμό στο περιθώριο, καθώς δεν μπορούσαν να ανταγωνιστούν το επίπεδο των βιομηχανικών σχεδίων τότε. Το "RISC" άλλαξε όλα αυτά. Με την υιοθέτηση απλών αρχιτεκτονικών, τα πανεπιστήμια ήταν για άλλη μια φορά σε θέση πραγματικά να σχεδιάσουν και να εφαρμόσουν τις δικούς τους επεξεργαστές και να καινοτομήσουν με μικρο-αρχιτεκτονική και μεταγλωττιστές. Μήπως ήρθε η ώρα για μια "RISC" επανάσταση για τα FPGAs; Οι Εταιρείες των FPGAs θα μπορούσαν να υποστηρίξουν ότι οι FPGA πλατφόρμες είναι κατ'ανάγκη πολύπλοκες και έξω από τη σφαίρα του τι είναι δυνατόν για την εφαρμογή μέσα σε ένα ακαδημαϊκό περιβάλλον, και θα έλεγαν ότι μία προσπάθεια ανοικτού κώδικα δεν είναι απαραίτητη, δεδομένου ότι συμβάλλουν στην προώθηση της ακαδημαϊκής έρευνας με συνεργασίες τα με πανεπιστήμια, και με δωρεά των εργαλείων και τις πλατφόρμες υλικού. Αυτή η επιτροπή θα συζητάγε για τα πλεονεκτήματα και τα μειονεκτήματα μίας ακαδημαϊκή κοινότητα που θα οδηγούσε σε μία προσπάθεια κατασκευής μίας ανοικτή συσκευή FPGA και εργαλείων ανάπτυξης. Δυστυχώς δεν βρήκα πρακτικά του τι ειπώθηκε κατά την διάρκεια αυτής της επιτροπής στα πλαίσια του συμποσίου, αλλά μόνο και μόνο η ερώτηση και τα επιχειρήματα καταδεικνύουν το κενό που υπήρχε στα σχεδιαστικά εργαλεία και που έχει ως αποτέλεσμα να αρχίσουν να εμφανίζονται μετά από αυτή επιτροπή κάποιες προσπάθειες για εργαλεία ανάπτυξης, όπως θα δούμε και παρακάτω.

Πέρα από τα εργαλεία των εταιριών που είδαμε παραπάνω υπάρχουν εργαλεία σχεδίασης για την ολοκληρωμένη ροή της σχεδίασης του FPGA ανοιχτού κώδικα που εμφανίστηκαν μετά την επιτροπή που αναφέραμε νωρίτερα. Όλα τα προγράμματα που ανήκουν στην κατηγορία αυτή συνοδεύονται από τον κώδικα τους επιτρέποντας την αλλαγή του, για τη βελτίωση του ίδιου του προγράμματος, ή ακόμη και την χρησιμοποίηση του κώδικα αυτούσιου (ή με αλλαγές) για την εξυπηρέτηση των εκάστοτε αναγκών. Στο τομέα των εργαλείων CAD, οι περισσότερες προσπάθειες είναι από ακαδημαϊκές έρευνες.

Το σχεδιαστικό εργαλείο VPR [5] που δημιουργήθηκε από το Πανεπιστήμιο του Τορόντο χρησιμοποιείται ευρέως για τη τοποθέτηση και τη διασύνδεση των λογικών μπλοκ σε αρχιτεκτονικές FPGA για ερευνητικούς και ακαδημαϊκούς σκοπούς. Χρησιμοποιείται σε συνδυασμό με το T-VPack, που αποτελεί το εργαλείο για τη δημιουργία των λογικών μπλοκ και των συστοιχιών από αυτά, ελαχιστοποιώντας το πλήθος των συνδέσεων μεταξύ των συστοιχιών στο κρίσιμο μονοπάτι. Το VPR αποτελείται από δύο τμήματα, ένα εργαλείο που

κάνει την τοποθέτηση των λογικών μπλοκ στη διάταξη του FPGA και ένα άλλο για τη διασύνδεση, έτσι ώστε τελικά να απεικονιστεί το σχεδιασμένο κύκλωμα στην διάταξη του FPGA. Στο παρακάτω σχήμα βλέπουμε την σχεδιαστική ροή που συνήθως ακολουθείτε.



Εικόνα 8 Τυπική σχεδιαστική ροή με χρήση VTR [26]

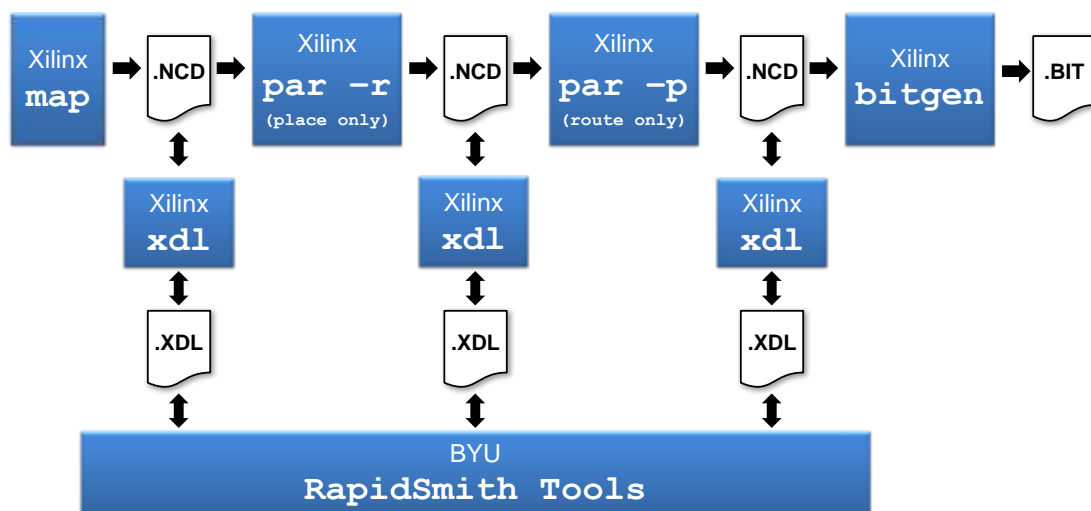
Το Πανεπιστήμιο του Τορόντο που είχε υλοποιήσει το VPR, μετά το συμπόσιο που αναφέραμε ξεκίνησε ένα project που είναι μια παγκόσμια προσπάθεια συνεργασίας πολλών

ερευνητικών ομάδων για να παρέχουν μία ολοκληρωμένη, σε πλαίσιο ανοιχτού κώδικα, για την διεξαγωγή έρευνας και ανάπτυξης σε αρχιτεκτονική FPGA και CAD. Το project αυτό ονομάζεται Verilog-to-Routing (VTR) [30],[42]. Αυτή η ροή λογισμικού ξεκινάει με μία Verilog περιγραφή υλικού ψηφιακών κυκλωμάτων και ένα αρχείο που περιγράφει την στοχευόμενη υποθετική αρχιτεκτονική και επεξεργάζεται, συνθέτει, πακετάρει, τοποθετεί και δρομολογεί το κύκλωμα και εκτελεί ανάλυση χρονισμού στο αποτέλεσμα.

Πρώτον, με την χρήση του ODIN II [16] μετατρέπει μία Verilog Hardware Description Language (HDL – Γλώσσα Περιγραφής Υλικού Verilog) σχεδίαση σε μία συνδεσμολογία του κυκλώματος (netlist) που αποτελείται από λογικές πύλες και blackboxes που αντιπροσωπεύουν ετερογενή τμήματα. Στη συνέχεια, το πακέτο σύνθεσης ABC [7] χρησιμοποιείται για την εκτέλεση της τεχνολογικά-ανεξάρτητης λογική βελτιστοποίησης κάθε κυκλώματος και μετά κάθε κύκλωμα είναι η τεχνολογικά-αντιστοιχισμένο σε LUTs και flip-flops. Η έξοδος του ABC είναι μίας .blif μορφής συνδεσμολογίας κυκλώματος (netlist) από LUTs, flip flops, και blackboxes. Μετά το VPR συσκευάζει αυτή τη συνδεσμολογία κυκλώματος (netlist) σε περισσότερο μεγάλων γκρουπ λογικών μπλοκ, τοποθετεί το κύκλωμα, και το δρομολογεί. Η έξοδος του VPR αποτελείται από διάφορα αρχεία. Ένα αρχείο για να περιγράψει τη συσκευασία του κυκλώματος, ένα άλλο αρχείο που περιγράφει την τοποθέτηση του κυκλώματος, ένα άλλο αρχείο που περιγράφει τη δρομολόγηση του κυκλώματος, καθώς και διάφορα αρχεία που περιγράφουν τα στατιστικά στοιχεία σχετικά με τον ελάχιστο αριθμό των κομματιών ανά κανάλι που απαιτούνται για την επιτυχή διαδρομή, το συνολικό μήκος καλωδίωσης, κλπ. Για να βρεθεί ο ελάχιστος αριθμός των κομματιών που απαιτούνται για την επιτυχή δρομολόγηση, το VPR επιχειρεί στην πραγματικότητα να δρομολογήσει το κύκλωμα πολλές φορές με διαφορετικούς αριθμούς των κομματιών που επιτρέπονται ανά κανάλι σε κάθε απόπειρα δρομολόγησης.

Επίσης υπάρχει και το RapidSmith [20], ένα εργαλείο που δημιουργήθηκε από το Πανεπιστήμιο Brigham Young (Brigham Young University – BYU), που πρακτικά είναι ένα σετ εργαλείων και API γραμμένο σε Java που στοχεύει να παρέχει ακαδημαϊκά, με μία εύκολη στην χρήση πλατφόρμα για να δοκιμάζονται πειραματικές ιδέες και αλγόριθμοι στα σύγχρονα FPGA της Xilinx. Το RapidSmith βασίζεται στην Γλώσσα Σχεδιασμού της Xilinx (Xilinx Design Language XDL), η οποία παρέχει μία αναγνώσιμη από τον άνθρωπο μορφή αρχείου που ισοδυναμεί με το ιδιόκτητο της Xilinx Netlist Circuit Description (NCD). Με το RapidSmith, οι ερευνητές είναι σε θέση να εισάγουν XDL / NCD, χειριστούν, τοποθετήσουν, δρομολογήσουν και να εξάγουν σχέδια ανάμεσα σε μία ποικιλία μετασχηματισμών σχεδιασμού. Το project του RapidSmith κάνει ένα εξαιρετική πλατφόρμα δοκιμών για να δοκιμαστούν νέες ιδέες και αλγόριθμοι για έρευνα FPGA CAD αφού ο κώδικας μπορεί γρήγορα να γραφτεί για να επωφεληθούν από τα διαθέσιμα APIs.του Το RapidSmith περιέχει επίσης πακέτα που μπορούν να αναλύσουν/εξάγουν (αλληλουχίες μπιτ) bitstreams (σε επίπεδο πακέτου) και να αντιπροσωπεύσουν τα πλαίσια και τα διαμορφώσιμα μπλοκ στις παρεχόμενες δομές

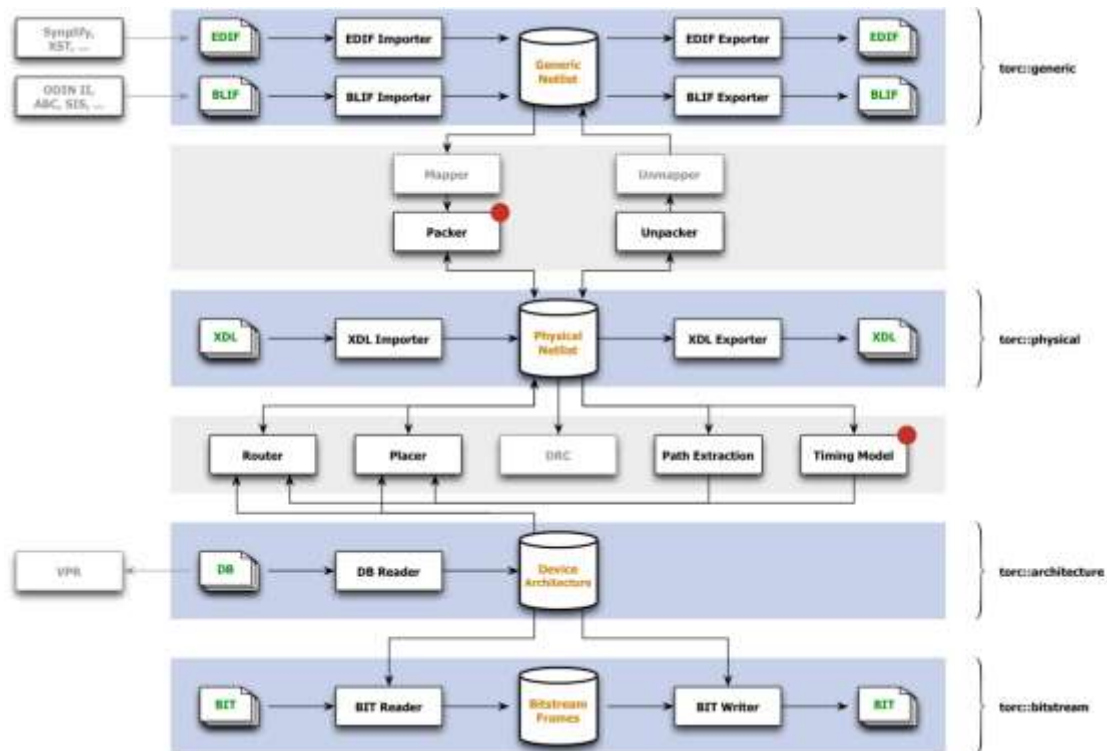
δεδομένων. Μπορεί να αναλύσει, χειριστεί και εξάγει αλληλουχίες μπιτ (bitstreams), σύμφωνα με τις τεκμηριωμένες μεθόδους της Xilinx. Το RapidSmith δεν περιλαμβάνει ιδιόκτητες πληροφορίες σχετικά με τα Xilinx FPGAs που να μην είναι διαθέσιμες στο κοινό. Η διαφορά του με το VPR σύμφωνα με το manual του RapidSmith είναι ότι μπορεί να δίνει την δυνατότητα να στοχεύονται τα εμπορικά Xilinx FPGAs, ενώ με το VPR μπορεί να στοχεύει όσο μόνο FPGA χαρακτηριστικά, που μπορεί να περιγραφούν από το δυνατότητα αρχιτεκτονικών περιγραφής του VPR. Το εργαλείο τοποθέτησης του RapidSmith, έχει μέσα έναν αλγόριθμο τοποθέτησης που είναι όμως τυχαίος όπως λέει και το όνομα του (RandomPlacer). Στην αναζήτηση μας είδαμε ότι υπάρχει εφαρμογή της προσομοιωμένης ανόπτησης σε τοποθετητή του RapidSmith που έχει προέλθει από τα «Εργαστήρια Ενσωματωμένων Συστημάτων» του Πανεπιστημίου Πειραιώς.



Εικόνα 9 Μπλοκ διάγραμμα για το που το XDL εφαρμόζεται στην σχεδιαστική ροή [20].

Το TORC (Tools for Open Reconfigurable Computing) [33] είναι εργαλείο CAD ανοιχτού κώδικα που αναπτύχθηκε από Ινστιτούτο υπηρεσιών Πληροφορικής του Πανεπιστημίου της Νότιας Καλιφόρνια (Information Services Insitute at the University of Southern California (USC-ISI). Όπως και το RapidSmith, το TORC χρησιμοποιεί XDL και XDLRC για να συμπληρώσει τις δικές του δομές δεδομένων και παρέχει ένα API που επιτρέπει στους χρήστες να χειριστούν σχέδια. Σε αντίθεση με το RapidSmith, το TORC προσφέρει μία πιο ολοκληρωμένη σειρά εργαλείων που παρέχονται για την τεχνολογική αντιστοίχιση και την συσκευασία. Πιο συγκεκριμένα, το TORC είναι σε θέση να ανοίξει XDL, σε αναπαράσταση στο επίπεδο BEL. Αυτό τον καιρό φτιάχνετε ένα χαρακτηριστικό που θα συσκευάζει αυτή την αναπαράσταση πάλι σε XDL. Το RapidSmith από την άλλη παρέχει μόνο υποστήριξη στο

επίπεδο του XDL και αναμένει από τους χρήστες να χειριστούν όλες τις υποπεριοχές με χαρακτηριστικά που περάστηκαν στο XDL. Δυστυχώς τα εργαλεία συσκευασίας και τοποθέτησης (αυτά που μας ενδιαφέρουν και στην παρούσα εργασία) είναι υπό κατασκευή. Βλέποντας την επίσημη σελίδα του TORC [41], βλέπουμε ότι υπάρχει μία υλοποίηση τοποθετητή (placer) που χρησιμοποιεί τον κλασικό αλγόριθμο της προσομοιωμένης ανόπτησης.



Εικόνα 10 Μπλοκ διάγραμμα του TORC [33].

Το πιο σημαντικό αρχιτεκτονικό στοιχείο του FPGA είναι αναμφισβήτητα η δομή διασύνδεσης. Δεδομένου ότι κάθε FPGA έχει έναν πεπερασμένο αριθμό διακριτών πόρων δρομολόγησης, ένα μεγάλο μέρος της αρχιτεκτονικής ερευνητικής προσπάθειας είναι αφιερωμένο στον καθορισμό της σύνθεσης της δομής διασύνδεσης ενός FPGA. Κατά την ανάπτυξη της αρχιτεκτονικής.

Τα παραπάνω εργαλεία είναι χτισμένα πάνω στην XDL, η οποία διακόπηκε με την εισαγωγή του νέου εργαλείου της Xilinx το Vivado Design Suite. Το καλοκαίρι του 2012 η Xilinx κυκλοφόρησε το Vivado, την επόμενη σουίτα του για τις μελλοντικές αρχιτεκτονικές FPGA. Με την αντικατάσταση του ISE, το Vivado φέρνει μια σειρά από αλλαγές, συμπεριλαμβανομένης της διακοπής της XDL. Αντ' αυτού το Vivado παρέχει άμεση πρόσβαση στις δομές δεδομένων μέσω μιας διεπαφής Tcl, καθώς και EDIF και Xilinx Design

Constrain (XDC) αρχεία. Αυτό το κενό προσπαθεί να καλύψει το Tincr [37], μια σουίτα από δύο βιβλιοθήκες με βάση Tcl, η οποίες, κάθε μία ενσωματώνουν μια ξεχωριστή μέθοδο για την εφαρμογή ενός τέτοιου πλαισίου. Η πρώτη είναι η βιβλιοθήκη TincrCAD, ένας υψηλού επιπέδου εργαλείο CAD χτισμένο μέσα στο Tcl περιβάλλον του Vivado. Το δεύτερο είναι TincrIO, ένα σύνολο εντολών Tcl που περιλαμβάνει μια διεπαφή που βασίζεται σε αρχεία σε Vivado, παρόμοια με XDL. Αυτές οι βιβλιοθήκες προσφέρονται ως απόδειξη ότι το Vivado Design Suite μπορεί να αποτελέσει τη βάση για την εφαρμογή διαμορφώμενων εργαλείων CAD που λειτουργούν σε Xilinx FPGAs για το άμεσο μέλλον. Την δεύτερη βιβλιοθήκη μπορούμε να την χρησιμοποιήσουμε, ώστε να μπορέσουμε να εφαρμόσουμε τα εργαλεία που αναφέρθηκαν νωρίτερα όπως το RapidSmith και το TORC. Δημιουργεί μία διεπαφή του Vivado με εξωτερικά εργαλεία CAD. Αυτό μας δίνει τρόπο ώστε τα υπάρχοντα εργαλεία CAD βασισμένα σε XDL μπορούν να χρησιμοποιηθούν και σε επόμενες γενιές των FPGAs της Xilinx. Το TincrCAD, ένα βασισμένο στην Tcl εργαλείο CAD για το Vivado. Χρησιμοποιώντας τον ενσωματωμένο στο Vivado διεργασμένο Tool Command Language (tcl), το TincrCAD είναι ένα σύνολο συναρτήσεων φτιαγμένο στην κορυφή των σετ εντολών του Vivado, για να παρέχει πρόσβαση στις εσωτερικές δομές δεδομένων του Vivado σε ένα υψηλότερο επίπεδο αφάιρησης. Μπορεί να χρησιμοποιηθεί για την εφαρμογή προσαρμοσμένων εργαλείων όπως τοποθετητές και δρομολογητές. Το TincrCAD σχεδιάστηκε για τρεις βασικούς σκοπούς:

1. Παροχή ενός υψηλού επιπέδου, εργαλείο CAD με γραφικά σε Tcl διεπαφή του Vivado.
2. Επέκταση της λειτουργικότητας του Vivado προσθέτοντας χειροκίνητα πληροφορίες που λείπουν.
3. Να διαχειρίζεται απρόσκοπτα πολύπλοκους μηχανισμούς, όπως sites και BELs.

Το TincrCAD πληροί κάθε έναν από τους σκοπούς αυτούς, μέσα από τα διάφορα πακέτα που προσφέρει. Συλλογικά, αυτά περιλαμβάνουν ένα πλήρες βασισμένο στην Tcl εργαλείο CAD πάνω στο οποίο μπορεί να υλοποιηθεί ένας αριθμός από εφαρμογές. Επίσης υπάρχει παράδειγμα πώς κάποιος θα εφαρμόσει ένα εργαλείο τοποθέτησης με σετ εντολών TincrCAD. Και εδώ, όπως στο RapidSmith η εφαρμογή του αλγορίθμου είναι ενός RandomPlacer.

Τέλος η ίδια η Altera έχει μία εργαλειοθήκη την Quartus II University Interface Program (QUIP) [40] που παρέχει τεκμηρίωση, σεμινάρια, αρχεία δεδομένων, και δείγμα κώδικα για να καταστεί δυνατή η πρόσβαση στη σουίτα CAD Quartus II σε διαφορετικά στάδια της ροής CAD. Οι ακαδημαϊκοί ερευνητές έχουν συχνά εξαιρετικές ιδέες για νέους αλγορίθμους λογισμικού ή εντελώς νέες φάσεις της ροής σχεδίασης λογισμικού, αλλά δεν διαθέτουν τους πόρους για να δημιουργήσουν μια ολοκληρωμένη ροή σχεδίασης λογισμικού ώστε να δοκιμάσουν τις ιδέες τους. Χρησιμοποιώντας το QUIP για να προσθέσει ένα νέο αλγόριθμο στο λογισμικό Quartus II, ένας ακαδημαϊκός ερευνητής μπορεί τώρα γρήγορα να ελέγξει το πόσο καλά λειτουργεί ένας νέος αλγόριθμος σε FPGA σε μια πλήρη ροή σχεδίασης λογισμικού. Αυτό θα έχει ως αποτέλεσμα την αύξηση τόσο σε ποσότητα όσο και σε ποιότητα της έρευνας σε FPGA σχεδίαση λογισμικού. Οι εταιρείες που προσπαθούν να

εμπορευματοποιήσουν την έρευνα μπορούν επίσης να χρησιμοποιήσουν το QUIP για τη διασύνδεση στο λογισμικό Quartus II, δίνοντάς τους τη δυνατότητα να οδηγήσουν τα προϊόντα τους στην αγορά χωρίς την κατασκευή μίας πλήρους ροής σχεδίασης λογισμικού από το μηδέν. «Υπάρχουν πολλά μεγάλα μυαλά στα Πανεπιστήμια», δήλωσε ο Dr. Vaughn Betz, διευθυντής του τμήματος του Software Engineering στην Altera. «Η Altera, και το σύνολο της βιομηχανίας FPGA, έχουν ωφεληθεί τα μέγιστα από τις εργασίες τους κατά το παρελθόν. Αλλά, όσο τα FPGAs γίνονται πιο περίπλοκα, γίνεται εξαιρετικά δύσκολό, αν όχι αδύνατο, για οποιοδήποτε πανεπιστήμιο να αναπτύξει μία ολοκληρωμένη ροή σχεδίασης λογισμικού, που να μπορεί να χειριστεί όλα τα χαρακτηριστικά και την πολυπλοκότητα του σύγχρονου FPGA. Το αποτέλεσμα είναι ότι η περισσότερη έρευνα πραγματοποιήθηκε σε πολύ απλουστευμένα μοντέλα FPGA, χρησιμοποιώντας απλοποιημένη ροή σχεδίασης λογισμικού. Με το QUIP, ελπίζουμε να επιτρέψουμε στους ερευνητές να εργαστούν σε όλο το φάσμα των προβλημάτων που τα σύγχρονα FPGAs παρουσιάζουν και για να μπορέσει να υπάρξουν ανακαλύψεις σε οποιαδήποτε ή όλα αυτά τα προβλήματα ».Με το λογισμικό Quartus II και τις πληροφορίες που παρέχονται σε αυτό το kit, οι προγραμματιστές CAD εργαλείων μπορούν να ενσωματώσουν τα δικά τους εργαλεία CAD και τις ιδέες τους σε μια πλήρη ροή FPGA CAD, από το επίπεδο μεταφοράς καταχωρητή (RTL) (καθώς και υψηλότερου επιπέδου) περιγραφές των κυκλωμάτων σε αρχεία προγραμματισμού για FPGAs. Η εργαλειοθήκη QUIP έχει σχεδιαστεί για να επιτρέψει πανεπιστήμιο (ή άλλους) ερευνητές να βάλουν νέα εργαλεία CAD και ιδέες στη ροή λογισμικό CAD Altera Quartus II. Το QUIP περιγράφει τις συσκευές Altera, τις διασυνδέσεις με τις οποίες τα δεδομένα μπορούν να σταλούν στο λογισμικό Quartus II σε διάφορα σημεία της ροής CAD και μορφές δεδομένων στην οποία τα δεδομένα μπορούν να απορρίπτονται από το λογισμικό Quartus II. Αυτή η εργαλειοθήκη επιτρέπει στους ερευνητές να γράψουν εργαλεία CAD που εκτελούν μία βελτιστοποίηση CAD σε ένα νέο ή καλύτερο τρόπο, και να ενσωματώσουν το νέο εργαλείο CAD τους σε μια πλήρη ροή CAD έτσι μπορούν να πάρουν ρεαλιστικά αποτελέσματα σχετικά με το πώς αυτή η νέα ιδέα βελτιώνει κύκλωμα χρονισμού, η δυνατότητα δρομολόγησης, η αξιοποίηση της συσκευής, τη διάρκεια της μεταγλώττισης, ή άλλες μετρήσεις. Το site του QUIP [40] αναφέρει μερικές ροές CAD που θα μπορούσαν να αξιοποιηθούν με το QUIP και αναφέρω μία που μας ενδιαφέρει και για τη συγκεκριμένη έρευνα:

- **Αντικαταστήστε τον αλγόριθμο τοποθέτησης του λογισμικού Quartus II με έναν δικό σας.** Η είσοδος σας είναι η τεχνολογικά αντιστοιχισμένη συνδεσμολογία κυκλώματος (netlist) από το λογισμικό Quartus II, και η έξοδος σας είναι μια τοποθέτηση για να πάει πίσω στο λογισμικό Quartus II. Μπορείτε ακόμη να εξάγετε μερικές τοποθετήσεις – τοποθετήστε τα μέρη του κυκλώματος που το εργαλείο CAD σας καταλαβαίνει (π.χ., λογική κύτταρα και I / O ακροδέκτες) και αφήνετε τις πολύπλοκες λειτουργίες (π.χ., μνήμη RAM και μπλοκ DSP) για το λογισμικό Quartus II να τα τοποθετήσει.

2.5 Το πρόβλημα της τοποθέτησης (Placement)

Το πιο σημαντικό αρχιτεκτονικό στοιχείο του FPGA είναι αναμφισβήτητα η δομή διασύνδεσης. Δεδομένου ότι κάθε FPGA έχει έναν πεπερασμένο αριθμό διακριτών πόρων δρομολόγησης, ένα μεγάλο μέρος της αρχιτεκτονικής ερευνητικής προσπάθειας είναι αφιερωμένο στον καθορισμό της σύνθεσης της δομής διασύνδεσης ενός FPGA. Κατά την ανάπτυξη της αρχιτεκτονικής, η αποτελεσματικότητα της δομής διασύνδεσης ενός FPGA της αξιολογείται, με τα εργαλεία τοποθέτησης και δρομολόγησης (συλλογικά ονομάζονται εργαλεία τοποθέτησης και δρομολόγησης (place-and-route tools)). Το εργαλείο τοποθέτησης και δρομολόγησης (place-and-route) είναι υπεύθυνο για την παραγωγή μιας φυσικής υλοποίησης μίας συνδεσμολογίας κυκλώματος (netlist) εφαρμογής για το προκατασκευασμένο υλικό του FPGA. Συγκεκριμένα, ο τοποθετητής προσδιορίζει την πραγματική φυσική θέση του κάθε λογικού μπλοκ της συνδεσμολογίας του κυκλώματος (netlist) στη διάταξη FPGA, και ο δρομολογητής εκχωρεί τα σήματα που συνδέουν τα τοποθετημένα λογικά μπλοκ για τη δρομολόγηση των πόρων στη δομή διασύνδεσης του FPGA. Λόγω του πεπερασμένου χαρακτήρα της δομής διασύνδεσης ενός FPGA του, η επιτυχία του δρομολογητή εξαρτάται σε μεγάλο βαθμό από την ποιότητα των λύσεων που παράγονται από τον τοποθετητή. Δεν αποτελεί έκπληξη, ότι από τους πρωταρχικούς στόχους του τοποθετητή είναι να παράγει μια τοποθέτηση που να μπορεί πραγματικά να δρομολογηθεί από τον δρομολογητή.

Οι αλγόριθμοι τοποθέτησης καταδεικνύουν ποιο λογικό μπλοκ μέσα στο FPGA, πρέπει να υλοποιήσει κάθε λογικό μπλοκ που απαιτεί το κύκλωμα. Υπάρχουν τρεις βασικοί στόχοι βελτιστοποίησης. Ο πρώτος είναι τοποθετηθούν τα συνδεδεμένα λογικά μπλοκ, αρκετά κοντά ώστε να μειωθεί η απαιτούμενη καλωδίωση (wire-length-driven placement – τοποθέτηση σύμφωνα με το μήκος καλωδίωσης), μερικές φορές να τοποθετηθούν τα μπλοκ ώστε να εξισορροπείται η πυκνότητα καλωδίωσης σε όλο το FPGA (routability-driven placement – τοποθέτηση σύμφωνα με την δρομολόγηση) ή να μεγιστοποιήσουμε την ταχύτητα του κυκλώματος (timing-driven placement – τοποθέτηση σύμφωνα με τον χρονισμό). Η δρομολόγηση που απαιτείται για να συνδεθούν δύο μπλοκ είναι μία συνάρτηση όχι μόνο της απόστασης μεταξύ αυτών, αλλά και της αρχιτεκτονικής του FPGA. Στην εργασία μας βλέπουμε την αρχιτεκτονική νησίδας (island-style FPGA), το μέγεθος της καλωδίωσης που απαιτείται για την σύνδεση των δύο μπλοκ είναι περίπου ανάλογη ως προς την Manhattan απόσταση μεταξύ τους. Αλλά για ιεραρχικές αρχιτεκτονικές (hierarchical architectures) τα

πράγματα είναι αρκετά διαφορετικά. Εκεί η ποσότητα καλωδίωσης που απαιτείται για την σύνδεση των δύο μπλοκ είναι ανάλογη με τον αριθμό των επιπέδων της ιεραρχίας δρομολόγησης που πρέπει να μετατοπιστεί για την σύνδεση τους. Σαφώς και οι αλγόριθμοι τοποθέτησης σε FPGA πρέπει να έχουν ένα μοντέλο της αρχιτεκτονικής δρομολόγησης που στοχεύουν προκειμένου να επιτύχει καλά αποτελέσματα. Επίσης, κάποιοι πρόσφατοι αλγόριθμοι τοποθέτησης σε FPGA, προσπαθούν επίσης να μειώσουν την κατανάλωση ενέργειας (τοποθέτηση με βάση την κατανάλωση – power-driven placement).

Ο αλγόριθμος τοποθέτησης είναι ένας κρίσιμος παράγοντας που επηρεάζει την τελική απόδοση του FPGA και ένας καλός αλγόριθμος τοποθέτησης μπορεί να βελτιώσει σημαντικά την αποτελεσματικότητα της εφαρμογής της σύνθεσης.

Οι τρεις μεγαλύτερες ομάδες τοποθετητών που χρησιμοποιούνται σήμερα.

- Οι τοποθετητές με βάση την κατάτμηση (partitioning-based) με κυρίαρχο τον min-cut
- Οι αναλυτικοί τοποθετητές (analytic) με κύριο εκφραστή τον τετραγωνικό (quadratic)
- Οι τοποθετητές με βάση την προσομοιωμένη απόπτωση (simulated annealing)

Οι τοποθετητές με βάση την κατάτμηση λειτουργούν διαχωρίζοντας αναδρομικά την συνδεσμολογία του κυκλώματος (circuit netlist) και αναθέτοντας κάθε περιοχή σε μία διαφορετική φυσική περιοχή στο FPGA. Συνήθως σε κάθε βήμα διαχωρισμού (κατάτμησης) χωρίζεται ένα προηγούμενο (μεγαλύτερο) διαμέρισμα σε δύο κομμάτια («κόβοντας στην μέση» το περιεχόμενο), αν και ορισμένοι αλγόριθμοι εκτελούν πολυεπίπεδο διαχωρισμό για να παράγουν ένα μεγαλύτερο αριθμό κατατμήσεων του κυκλώματος σε κάθε βήμα. Οι αλγόριθμοι κατάτμησης, προσπαθούν να ελαχιστοποιήσουν το αριθμό των δικτύων που κόβονται ή διασταυρώνονται μεταξύ των κατατμήσεων. Δεδομένου ότι κάθε κατάτμηση του κυκλώματος θα ανατεθεί σε διαφορετική περιοχή του FPGA, η τοποθέτηση με βάση την κατάτμηση ελαχιστοποιεί τον αριθμό των δικτύων αφήνοντας κάθε περιοχή και ως εκ τούτου βελτιώνει εμμέσως και το συνολικό μέγεθος καλωδίωσης που απαιτείται από τη σχεδίαση. Η τοποθέτηση με βάση την κατάτμηση μπορεί να αξιοποιήσει την τη διαθεσιμότητα από υψηλής ποιότητας, αποδοτικούς αλγορίθμους κατάτμησης, κάνοντας αυτή την προσέγγιση επεκτάσιμη σε μεγάλα προβλήματα.

Μία άλλη προσέγγιση της τοποθέτησης είναι η αναλυτική τοποθέτηση. Έχει για βασική ιδέα να εκφράσει την συνάρτηση κόστους και τους περιορισμούς ως αναλυτική συνάρτηση των συντεταγμένων των κόμβων. Στην συνέχεια το πρόβλημα της τοποθέτησης αναγάγεται σε μαθηματικό πρόβλημα. Αποδοτικές αριθμητικές τεχνικές χρησιμοποιούνται για να βρουν το ολικό ελάχιστο αυτής της συνάρτησης. Εάν η συνάρτηση προσεγγίζει το μήκος της καλωδίωσης καλά, αυτή η λύση είναι μία τοποθέτηση με καλό μήκος καλωδίωσης. Αυτή η μέθοδος χρησιμοποιείται ευρέως στην τρέχουσα γενιά φυσικής σχεδίασης των ASIC, λόγω του μικρότερου χρόνου ολοκλήρωσης, καθώς και της απόδοσής τους. Στα FPGA υπάρχουν λίγες αναλυτικές προσεγγίσεις τοποθέτησης, πιθανότητα εξαιτίας της μεγαλύτερης δυσκολίας στην τοποθέτηση λόγω φυσικών περιορισμών.

Η προσομοιωμένη ανόπτηση είναι ο πιο ευρέως χρησιμοποιούμενος αλγόριθμος για την τοποθέτηση σε FPGAs. Μιμείται τη διαδικασία ανόπτησης με τον οποίο δημιουργούνται ισχυρά κράματα μετάλλων. Αρχικά τα μπλοκ μπορούν να κινηθούν αρκετά ελεύθερα, αλλά καθώς η θερμοκρασία πέφτει σταδιακά παγώνει σε μια τοποθέτηση υψηλής ποιότητας [18]. Η βασική ροή της προσομοιωμένης ανόπτησης για την τοποθέτηση έχει ως εξής: Πρώτα μια αρχική τοποθέτηση δημιουργείται. Αυτή η αρχική τοποθέτηση είναι συνήθως χαμηλής ποιότητας και συχνά δημιουργείται απλά με την ανάθεση σε κάθε μπλοκ στην πρώτη επιτρεπτή θέση που βρέθηκε. Η τοποθέτηση στη συνέχεια βελτιώνεται επαναληπτικά με την πρόταση και την αξιολόγηση των ανταλλαγών μπλοκ. Μια συνάρτηση κόστους χρησιμοποιείται για την αξιολόγηση των επιπτώσεων της κάθε προτεινόμενης κίνησης. Κινήσεις που μειώνουν το κόστος είναι πάντα δεκτές, ενώ εκείνες που αυξάνουν το κόστος γίνονται δεκτές με πιθανότητα $e^{-\Delta Cost/T}$, όπου T είναι η τρέχουσα θερμοκρασία. Η συνάρτηση αυτή εξασφαλίζει ότι οι κινήσεις που αυξάνουν το κόστος κατά ένα ποσό το οποίο είναι μικρό σε σύγκριση με την τρέχουσα θερμοκρασία είναι πιθανό να γίνουν αποδεκτές, ενώ κινήσεις που αυξάνουν το κόστος κατά ένα ποσό πολύ μεγαλύτερο από την τρέχουσα θερμοκρασία δεν είναι πιθανό να γίνουν αποδεκτές. Αποδεχόμενοι κάποιες κινήσεις που αυξάνουν το κόστος βοηθά να ξεφύγουν τα τοπικά ελάχιστα και παράγεται υψηλότερης ποιότητας τελική τοποθέτηση. Κατά την έναρξη της ανόπτησης η θερμοκρασία είναι υψηλή. Στη συνέχεια μειώνεται σταδιακά σύμφωνα με το χρονοδιάγραμμα της ανόπτησης. Αυτό το χρονοδιάγραμμα ελέγχει επίσης πόσες κινήσεις πραγματοποιούνται μεταξύ των ενημερώσεων της θερμοκρασίας και όταν η τοποθέτηση αυτή θεωρείται ότι είναι επαρκώς βελτιστοποιημένη ότι η ανόπτηση πρέπει να λήξει. Το VPR [5] είναι ίσως το πιο δημοφιλές εργαλείο τοποθέτησης βασισμένο στην προσομοιωμένη ανόπτηση.

Επίσης οι γενετικοί αλγόριθμοι επίσης έχουν προταθεί για να εφαρμοστούν σε τοποθέτηση σε FPGA. Ο γενετικός αλγόριθμος (GA), ο οποίος εφευρέθηκε από τον Holland (1975), είναι μια κατευθυνόμενη στοχαστική τεχνική αναζήτησης που μπορεί να βρει την συνολική βέλτιστη λύση σε πολυσύνθετους χώρους αναζήτησης με πολλές διαστάσεις. Ένας γενετικός αλγόριθμος (GA) μοντελοποιείται πάνω στη φυσική εξέλιξη δεδομένου ότι οι τελεστές που χρησιμοποιεί επηρεάζονται από την διαδικασία εξέλιξης. Αυτοί οι τελεστές, γνωστοί σαν γενετικοί τελεστές, εφαρμόζονται σε άτομα ενός πληθυσμού μέσω μερικών γενεών για να βελτιώσουν σταδιακά την καταλληλότητά τους. Τα άτομα σε έναν πληθυσμό παρομοιάζονται με τα χρωμοσώματα και συνήθως αναπαριστάνονται σαν ακολουθίες από δυαδικούς αριθμούς. Ο τρόπος λειτουργίας των γενετικών αλγορίθμων (GA) είναι εμπνευσμένος από την βιολογία. Χρησιμοποιεί την ιδέα της εξέλιξης μέσω γενετικής μετάλλαξης, φυσικής επιλογής και διασταύρωσης. Οι γενετικοί αλγόριθμοι διατηρούν έναν πληθυσμό πιθανών λύσεων, του προβλήματος που μας ενδιαφέρει, πάνω στον οποίο δουλεύουν, σε αντίθεση με άλλες μεθόδους αναζήτησης που επεξεργάζονται ένα μόνο σημείο του διαστήματος αναζήτησης. Έτσι ένας γενετικός αλγόριθμος πραγματοποιεί αναζήτηση σε πολλές κατευθύνσεις και υποστηρίζει καταγραφή και ανταλλαγή πληροφοριών μεταξύ αυτών των κατευθύνσεων. Ο

πληθυσμός υφίσταται μια προσομοιωμένη γενετική εξέλιξη χρησιμοποιώντας διάφορους γενετικούς τελεστές όπως η επιλογή, η διασταύρωση και η μετάλλαξη. Στην πράξη ο αλγόριθμος ξεκινά μ' ένα σύνολο λύσεων - ονομάζονται γονιδιώματα, δανειζόμενες το όνομά τους από τη βιολογία, οι οποίες συνιστούν τον "πληθυσμό". Κατόπιν ζητείται από τον υπολογιστή να δημιουργήσει μια σειρά τυχαίων ανασυνδυασμών και μεταλλάξεων των "γονιδιωμάτων". Οι πιο ικανές λύσεις για ένα συγκεκριμένο πρόβλημα συνεχίζουν να εξελίσσονται και ανασυνδυάζονται τυχαία, μέχρις ότου "επιβιώσουν" οι καλύτερες. Συνήθως, όσο περισσότερες γενιές περνούν τόσο καλύτερες λύσεις βρίσκονται, μπορεί όμως ο αλγόριθμος να βρεθεί σε σημείο του πεδίου των λύσεων από όπου και δεν μπορεί να προχωρήσει λόγο του ότι βρίσκεται σε τοπικό μέγιστο. Για το λόγο αυτό έχουν υπάρχουν διαφορετικές εκδοχές του αλγόριθμου ανάλογα με τη μορφή του προβλήματος.

Στην τελική φάση του γενετικού αλγόριθμου (GA) υπάρχουν μικρές βελτιώσεις και πολύ αργά. Η προσομοιωμένη απόπτωση (SA) είναι ικανή να επιτύχει βελτιώσεις γρηγορότερα από το γενετικό αλγόριθμο (GA) στην τελική φάση της διαδικασίας. Ως εκ τούτου, μετά από ένα ορισμένο αριθμό γενεών η προσομοιωμένη απόπτωση χρησιμοποιείται με σκοπό τη βελτιστοποίηση του συνολικού πληθυσμού σε χαμηλή θερμοκρασία .

Είναι πολύ πιο εύκολο να θέσεις νέους στόχους βελτιστοποίησης ή περιορισμούς σε έναν τοποθετητή με βάση την προσομοιωμένη απόπτωση σε σχέση με τους min-cut και αναλυτικούς, εξαιτίας της ευελιξίας του να εφαρμόζει σε μία ευρεία ποικιλία στόχων βελτιστοποίησης.

Στα παρακάτω κεφάλαια θα δούμε αναλυτικά τους αλγόριθμους που χρησιμοποιούνται ή έχουν προταθεί.

Simulated Annealing Algorithm (Αλγόριθμος προσομοιωμένης ανόπτησης)

3.1 Γενικά Στοιχεία

Ο αλγόριθμος προσομοιωμένης ανόπτησης δημιουργήθηκε από στατιστικούς μηχανικούς¹ και εξελίχθηκε στα πλαίσια του προβλεπτικού ελέγχου. Ο προβλεπτικός έλεγχος (model predictive control) είναι μια μορφή βέλτιστου ελέγχου στον οποίο η μεταβλητή ελέγχου προκύπτει μέσω της επίλυσης ενός προβλήματος βελτιστοποίησης ανά δειγματοληψία μετατιθέμενου ορίζοντα (receding horizon control) εξελισσόμενου χρόνου (on line) ανοιχτού βρόχου, χρησιμοποιώντας την τρέχουσα τιμή του διανύσματος κατάστασης ως αρχική συνθήκη. Η βελτιστοποίηση επιτυγχάνεται με την επιλογή της πρώτης μεταβλητής ελέγχου από ένα σύνολο τιμών μεταβλητών ελέγχου, η οποία ακολούθως αποδίδεται στο σύστημα που μελετάται. Αυτή είναι κυρίως, η διαφορά με τους συμβατικούς ελέγχους, οι οποίοι χρησιμοποιούν έναν προ-υπολογισμένο νόμο ελέγχου.

Η ικανότητα του να διαχειρίζεται πολύπλοκα μη γραμμικά συστήματα τα οποία υπόκεινται σε λειτουργικούς περιορισμούς, ενισχύει την δυναμικότητα του προβλεπτικού ελεγκτή. Το τελευταίο δε, αποτελεί το σημαντικότερο πλεονέκτημα του έναντι των υπολοίπων βέλτιστων ελεγκτών οι οποίοι αδυνατούν να ενσωματώσουν περιοριστικούς όρους ή και να εφαρμοστούν σε μη γραμμικά συστήματα.

Κατά τη λειτουργία τους, σχεδόν όλα τα συστήματα υπόκεινται σε περιορισμούς όπως λόγου χάρη, στα συστήματα ελέγχου θερμοκρασίας, στα συστήματα ελέγχου ταχύτητας. Κάλιστα τα παραπάνω μπορούν να τεθούν σε όρια διακύμανσης και ασφάλειας. Το πλεονέκτημα αυτό, οδηγεί στην ανάπτυξη αλγορίθμων προβλεπτικού ελέγχου στην πετρελαϊκή και χημική βιομηχανία, όπου και ο προβλεπτικός έλεγχος παρουσιάζεται για πρώτη φορά. Ο αλγόριθμος τοποθέτησης της προσομοιωμένης ανόπτησης μιμείται την διαδικασία ανόπτησης που χρησιμοποιείται για τη σταδιακή ψύξη λιωμένου μετάλλου σε υψηλής ποιότητας μεταλλικά αντικείμενα. Με την επιβράδυνση της ψύξης του υλικού, κάθε μόριο κινείται σε μια θέση που ταιριάζει καλύτερα σε ένα σύστημα. Σε υψηλή θερμοκρασία, τα μόρια έχουν περισσότερο χώρο για να κινηθούν γύρω, ως εκ τούτου, η πιθανότητα της μετατόπισης τους είναι μεγαλύτερη. Καθώς η θερμοκρασία μειώνεται σιγά-σιγά, τα μόρια δεν είναι σε θέση να μετακινούνται ελεύθερα, πράγμα που μειώνει την πιθανότητα της

¹ Kickpatrich et al. (1983)

μετακίνησης. Σε χαμηλή θερμοκρασία, το μόριο είναι σε θέση να κινηθεί σε μια σταθερή θέση με αποτέλεσμα σε στερεά κατάσταση. Μια αρχική τοποθέτηση δημιουργείται με τυχαία τοποθέτηση των μπλοκ λογικής στο κύκλωμα. Μία μεγάλη εναλλαγή των μπλοκ γίνεται για να μειώσει σταδιακά το κόστος. Εδώ, το γνωστό εργαλείο VPR (Versatile Place and Route) που χρησιμοποιεί προσομοιωμένη ανόπτηση περιγράφεται [5].

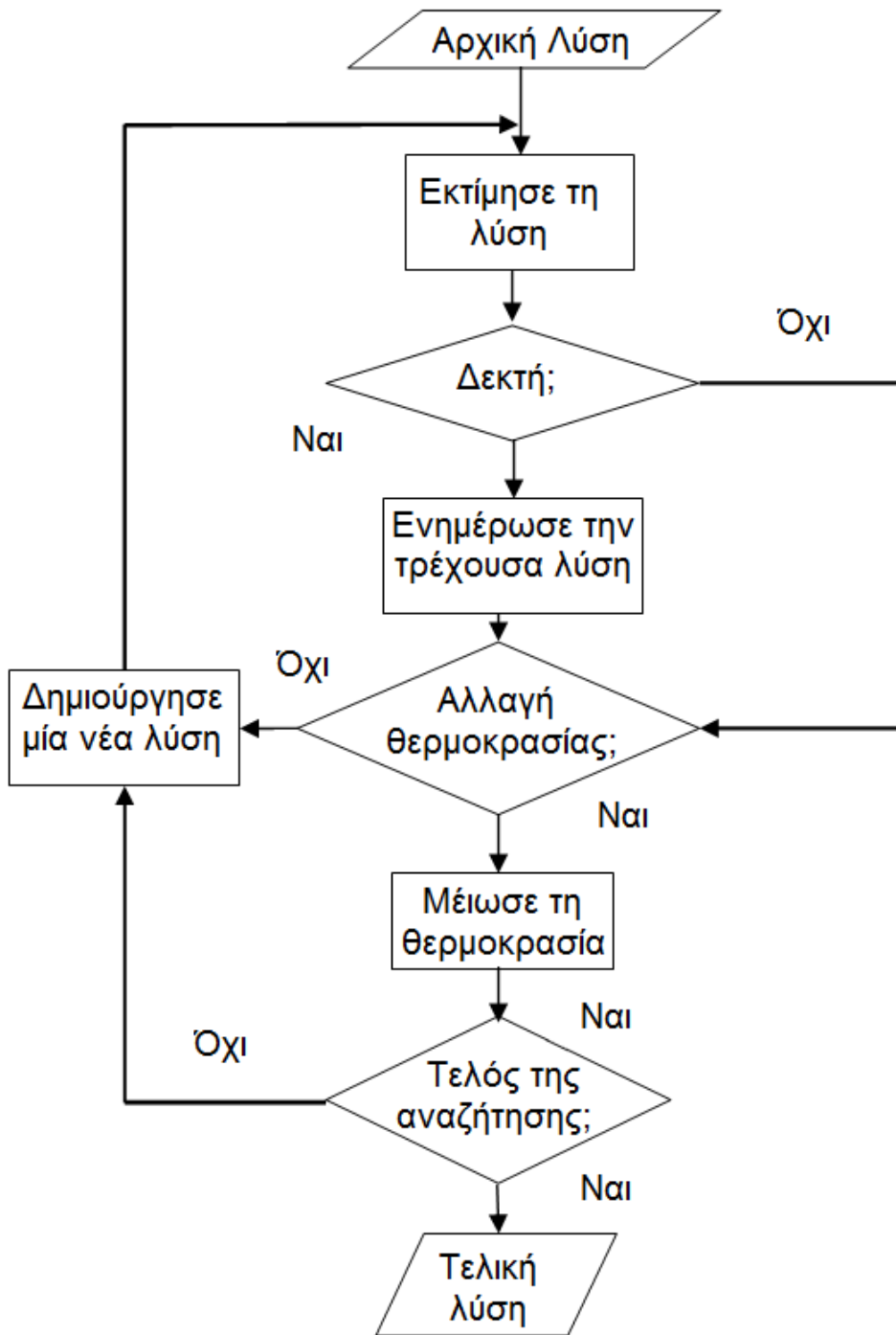
3.2 Υλοποίηση Αλγόριθμου

Αρχικά, ο αλγόριθμος αποτελείται από μία ακολουθία επαναλήψεων και κάθε επανάληψη αποτελείται από μια τυχαία αλλαγή της τρέχουσας λύσης ώστε να δημιουργηθεί μια άμεση λύση όσον αφορά την προηγούμενη τρέχουσα. Η νέα λύση που δημιουργήθηκε αντιπροσωπεύει μια αλλαγή στη συνάρτηση κόστους που υπολογίστηκε για να αποφασιστεί το κατά πόσον η νέα παραγόμενη μπορεί να γίνει αποδεκτή ως ισχύουσα. Εάν η αλλαγή στη συνάρτηση κόστους είναι αρνητική η νέα παραγόμενη λύση θεωρείται άμεσα ως η τρέχουσα λύση. Εναλλακτικά, γίνεται δεκτή σύμφωνα με το Metropolis's criterion πάνω στην πιθανότητα του Boltzmann [10].

Το διάγραμμα ροής ενός γενικού αλγορίθμου προσομοιωμένης ανόπτησης παρουσιάζεται στην εικόνα της επόμενης σελίδας. Για να υλοποιήσουμε τον αλγόριθμο όσον αφορά ένα πρόβλημα υφίστανται τέσσερις θεμελιώδεις επιλογές και αυτές είναι:

- Αναπαράσταση των λύσεων
- Καθορισμός της συνάρτησης κόστους
- Προσδιορισμός του γενετικού μηχανισμού για τους γείτονες
- Σχεδίαση της κρυσταλλικής δομής

Η αναπαράσταση των λύσεων και ο ορισμός της συνάρτησης κόστους λειτουργούν όπως οι γενετικοί αλγόριθμοι. Αρκετοί γενετικοί μηχανισμοί θα μπορούσαν να αναπτυχθούν όπως για παράδειγμα οι μηχανισμοί μετάλλαξης και αντιστροφής [6].

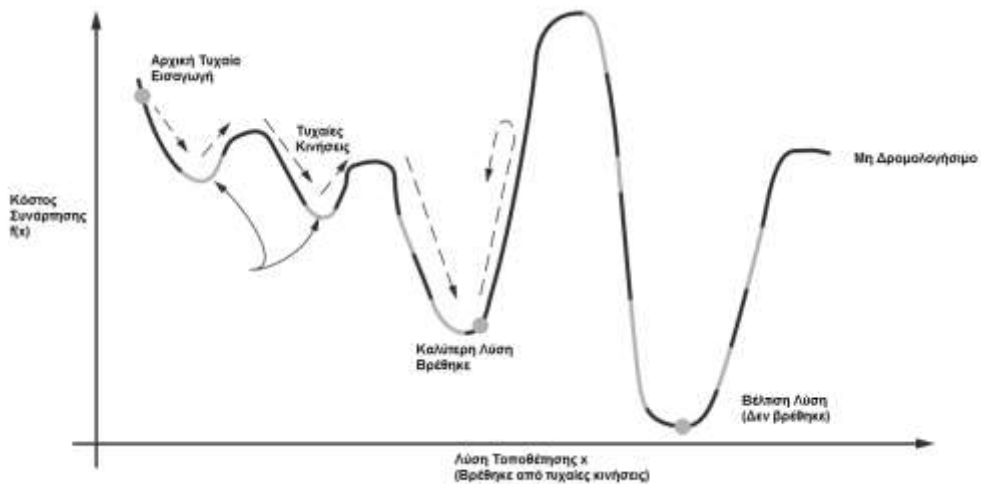


Εικόνα 11 Διάγραμμα ροής ενός κανονικού αλγορίθμου προσομοιωμένης ανόπτωσης

Πίνακας 2 Ψευδοκώδικας από γενικό τοποθετητή βασισμένο σε προσομοιωμένη απόπτωση

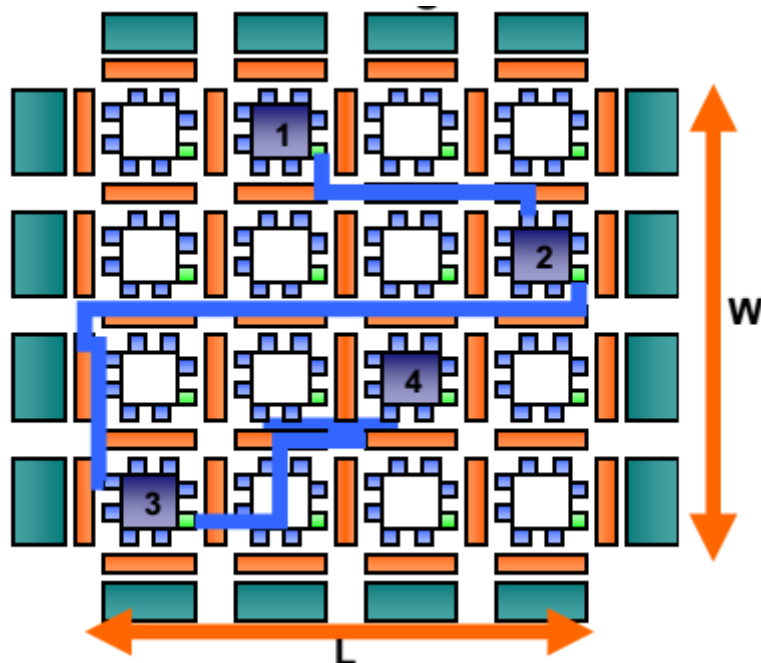
```
S = RandomPlacement ();
T = InitialTemperature ();
Rlimit = InitialRlimit ();

while (ExitCriterion () == False) {           // Εξωτερικός Βρόχος
  while (InnerLoopCriterion () == False) {    // Εσωτερικός Βρόχος
    Snew = GenerateViaMove (S,Rlimit);
    ΔC = Cost (Snew) – Cost (S);
    r = random (0,1);
    if (r < e-ΔC/T) {
      S = Snew;
    }
  }                                           // Τέλος Εσωτερικού Βρόχου
  T = UpdateTemp ();
  Rlimit = UpdateRlimit ();
}                                           // Τέλος Εξωτερικού Βρόχου
```



Εικόνα 12 Παραδοσιακός αλγόριθμος προσομοιωμένης απόπτωσης

Η διαδικασία της προσομοιωμένης απόπτωσης εκκινείται με έναν τυχαίο διαχωρισμό του κυκλώματος. Στη συνέχεια, 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.



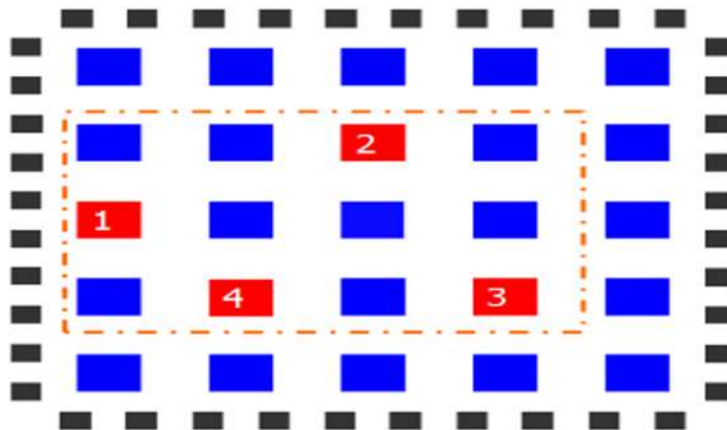
Εικόνα 13 Συνάρτηση κόστους πλαισίου οριοθέτησης

Εφόσον το κόστος της κίνησης των κελιών είναι αρνητικό τότε η προκείμενη κίνηση γίνεται δεκτή, ενώ υπό προϋποθέσεις, εάν είναι ίση με το μηδέν ή θετική. Αυτή η διαφοροποίηση

είναι και ο βασικός λόγος που η προσομοιωμένη ανόπτηση μπορεί να υπερβεί τοπικά ελάχιστα στην διαδικασία εύρεσης του καθολικού ελάχιστου. Μια διαδικασία ανόπτησης είναι να επιτρέψει στα μόρια να κρυώσουν με ελεγχόμενο τρόπο από τη θερμοκρασία για να βρουν την καλύτερη προσαρμογή τους στο σύστημα. Ο αλγόριθμος προσομοιωμένης ανόπτησης βασίζεται στην τυχαία μετακίνηση των μπλοκ λογικής, η οποία ονομάζεται "κίνηση" [9]. Η συνάρτηση κόστους ορίζεται για την αξιολόγηση της ποιότητας της τοποθέτησης και μία γραμμική συνάρτηση κόστους συμφόρησης ως εξής παρέχει τα καλύτερα αποτελέσματα σε ένα εύλογο χρονικό διάστημα υπολογισμού [10].

$$\text{Cost} = \sum_{n=1}^{N_{nets}} q(n) \left[\frac{bb_x(n)}{C_{av,x}(n)} + \frac{bb_y(n)}{C_{av,y}(n)} \right]$$

Για κάθε δίκτυο, η οριζόντια έκταση του πλαισίου οριοθέτησης, $bb_x(n)$, και κάθετη έκταση του πλαισίου οριοθέτησης, $bb_y(n)$, αθροίζονται με την εξέταση της μέσης χωρητικότητας του καναλιού στις x και y κατευθύνσεις αντίστοιχα, $C_{av,x}(n)$ και $C_{av,y}(n)$. Η μέση χωρητικότητα του καναλιού είναι σταθερή για την αρχιτεκτονική νησίδας FPGA [10]. Επίσης, ένας παράγοντας αντιστάθμισης, $q(n)$ θεωρείται για το μήκος της καλωδίωσης υπό εκτίμηση με την εισαγωγή πλαισίου οριοθέτησης ως εκτίμηση καλωδίωσης. Η προσομοιωμένη ανόπτηση ξεκινάει με μια αρχική τοποθέτηση που λαμβάνεται με την εκχώρηση CLBs τυχαία. Ο αλγόριθμος επιτρέπει ένα μεγάλο αριθμό κινήσεων για να βελτιώσει σταδιακά την τοποθέτηση.



Εικόνα 14 Πλαίσιο οριοθέτησης ενός δικτύου (με διακεκομμένη γραμμή είναι το πλαίσιο οριοθέτησης ενός δικτύου που αποτελείται από τέσσερα μπλοκ)

Μια κίνηση αποτελείται από τυχαία επιλογή ενός λογικού μπλοκ και τυχαία τοποθέτηση σε άλλη θέση. Η περιοχή της κίνησης του λογικού μπλοκ περιορίζεται από το όριο απόστασης, D_{limit} . Το όριο απόστασης ενημερώνεται ως εξής [10]:

$$D_{limit}^{new} = D_{limit}^{old} \cdot (1 - 0.44 + R_{accept}^{old})$$

Όπου $R_{oldaccept}$ είναι το ποσοστό των κινήσεων που έχει γίνει αποδεκτό στην παλιά θερμοκρασία. Η πιθανότητα της αποδοχής μια «κακής» κίνησης ελέγχεται από τη θερμοκρασία ανόπτωσης. Η αρχική θερμοκρασία προσδιορίζεται ως ακολούθως. Από μια αρχική τυχαία τοποθέτηση, επιλέγεται μια τυχαία κίνηση για κάθε λογικό μπλοκ και η τυπική απόκλιση του κόστους υπολογίζεται. Η αρχική θερμοκρασία είναι ρυθμισμένη 20 φορές την τυπική απόκλιση. Αυτή η αρχική θερμοκρασία επιτρέπει όλες τις επόμενες κινήσεις να γίνουν αποδεκτές. Η θερμοκρασία ανόπτωσης ενημερώνεται ως εξής [5]:

$$T_{new} = \alpha * T_{old}$$

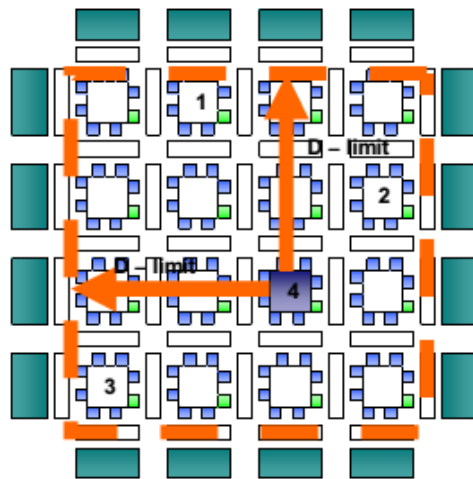
Η νέα θερμοκρασία εξαρτάται από το α , το οποίο καθορίζεται από το κλάσμα της απόπειρας κινήσεων που έγιναν δεκτές στην παλιά θερμοκρασία. Για το VPR, το α ενημερώνεται ως εξής [10]:

Πίνακας 3 Πίνακας ανανέωσης για το α

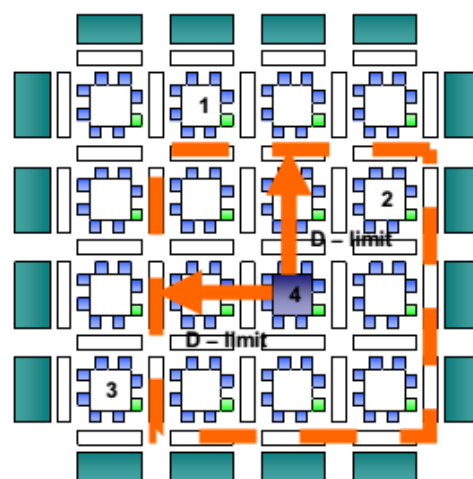
Κλάσμα των αποδεχόμενων κινήσεων (R_{accept})	α
$R_{accept} > 0.96$	0.5
$0.8 < R_{accept} \leq 0.96$	0.9
$0.15 < R_{accept} \leq 0.8$	0.95
$R_{accept} \leq 0.15$	0.8

Σε υψηλή θερμοκρασία, το όριο απόστασης, D_{limit} , είναι μεγάλο, όπως φαίνεται στην εικόνα 15 (αριστερά). Αυτό επιτρέπει μια υψηλή πιθανότητα της κίνησης να γίνει αποδεκτή στη θερμοκρασία, η οποία οδηγεί σε υψηλή τιμή R_{accept} . Καθώς η θερμοκρασία μειώνεται, το όριο απόστασης είναι μικραίνει με αποτελέσματα μικρότερη έκταση για τυχαία κίνηση όπως φαίνεται στην εικόνα 15 (δεξιά).

Μεταπτυχιακή Διατριβή



Γεώργιος Βάσιλας

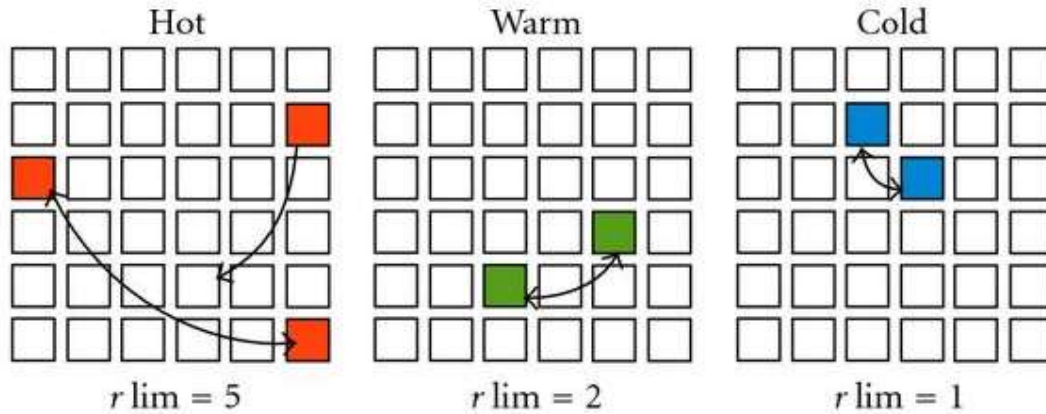


Εικόνα 15 Όριο απόστασης σε υψηλή και χαμηλότερη θερμοκρασία

Σε κάθε κίνηση, τυχαία επιλεγμένη από τον αλγόριθμο, η συνάρτηση κόστους χρησιμοποιείται για να καθορίσει είτε να αποδεχθεί, είτε να απορρίψει τη κίνηση. Αν το κόστος μειώνεται, τότε η κίνηση είναι πάντα δεκτή. Ωστόσο, εάν το κόστος αυξάνεται, υπάρχει μια πιθανότητα της κίνησης να γίνει δεκτή. Αυτό επιτρέπει στον αλγόριθμο να αποφευχθεί η πρόωρη σύγκλιση σε τοπικά ελάχιστα και επιτρέπει την κίνηση αναρρίχησης λόφων που επιτρέπει στην διαδικασία προσομοιωμένης απόπτωσης, την επίτευξη ολικού ελάχιστου [10]. Για το VPR, η προσομοιωμένη απόπτωση τερματίζεται όταν η θερμοκρασία είναι μικρότερη από 0,5% του μέσου κόστους ανά δίκτυο [5].

$$T < 0.005 * \text{Cost} / N_{\text{nets}}$$

Όταν η θερμοκρασία είναι μικρότερη από ένα κλάσμα του μέσου κόστους ενός δικτύου, είναι απίθανο ότι όποια κίνηση μπορεί να αποφέρει σε μια αύξηση του κόστους που θα γίνονται δεκτές. Ως αποτέλεσμα, ο αλγόριθμος θα τερματίσει όταν η εξίσωση $T < 0.005 * \text{Cost} / N_{\text{nets}}$ ικανοποιείται.



Εικόνα 16 Αρχικά, σε υψηλές θερμοκρασίες, τετράγωνα μακριά από το άλλο μπορούν να ανταλλαχθούν εύκολα. Τέλος, σε χαμηλές θερμοκρασίες, μόνο μπλοκ ένα κοντά στο άλλο μπορούν να ανταλλαχθούν

3.3. Πλεονεκτήματα και μειονεκτήματα

Υπάρχουν πολλά πλεονεκτήματα της τεχνικής τοποθέτησης της προσομοιωμένης ανόπτησης. Υπερτερεί των άλλων τοποθετητών όσο μπορούν να γίνουν άμεσες συγκρίσεις. Το εργαλείο VPR που είναι το πιο ευρέως διαδεδομένο εργαλείο στον τομέα χρησιμοποιεί τον προσομοιωμένο ανοπτητή. Έχει μια ανοιχτή συνάρτηση κόστους, η οποία μπορεί να χρησιμοποιηθεί είτε με γνώμονα το μήκος καλωδίωσης ή συνάρτηση κόστους με γνώμονα τον χρονισμό ή με γνώμονα την δρομολόγηση. Η συνάρτηση κόστους μπορεί επίσης να είναι ο γραμμικός συνδυασμός των παραπάνω τύπων αν και είναι δύσκολο να αποφασιστούν τα βάρη. Επίσης, ο αλγόριθμος της προσομοιωμένης ανόπτησης είναι σε θέση να επιτύχει την ολική βέλτιστη λύση. Ωστόσο, ο αλγόριθμος της προσομοιωμένης ανόπτησης είναι πολύ αργός εξαιτίας της υπολογιστικά δαπανηρής και χρονοβόρας αξιολόγησης κάθε κίνησης. Εκτός αυτού, λόγω της εγγενούς διαδοχικής φύσης της προσομοιωμένης ανόπτησης, είναι πολύ δύσκολο να παραλληλιστεί με τη χρήση πολυπύρηνων επεξεργαστών. [10].

3.4. Παράλληλη Τοποθέτηση (Parallel Placement)

Καθώς η κλίμακα των σύγχρονων FPGAs έχει φτάσει τα εκατομμύρια των λογικών μπλοκ, πιο αποτελεσματικοί και επεκτάσιμοι αλγόριθμοι τοποθέτησης FPGA χρειάζονται. Η παραλληλία είναι μια ελκυστική λύση για την παροχή γρήγορων τοποθετήσεων λόγω της ταχείας ανάπτυξης των πολυπύρηνων επεξεργαστών κατά τα τελευταία έτη. Οι παράλληλες προσεγγίσεις που πρόκειται να εξετάσουμε με βάση την προσομοιωμένη απόδοση, δεδομένου ότι θεωρείται η κορυφαία, ενώ το κύριο μειονέκτημα της είναι η χρονοβόρα κίνησή. Χωρίζουμε τους σύγχρονους παράλληλους τοποθετητές προσομοιωμένης απόδοσης σε FPGA σε τρεις κατηγορίες: Προσέγγιση παράλληλης κίνησης [28], Προσέγγιση που βασίζεται περιοχή [28] και ντετερμινιστική παράλληλη προσέγγιση[22].

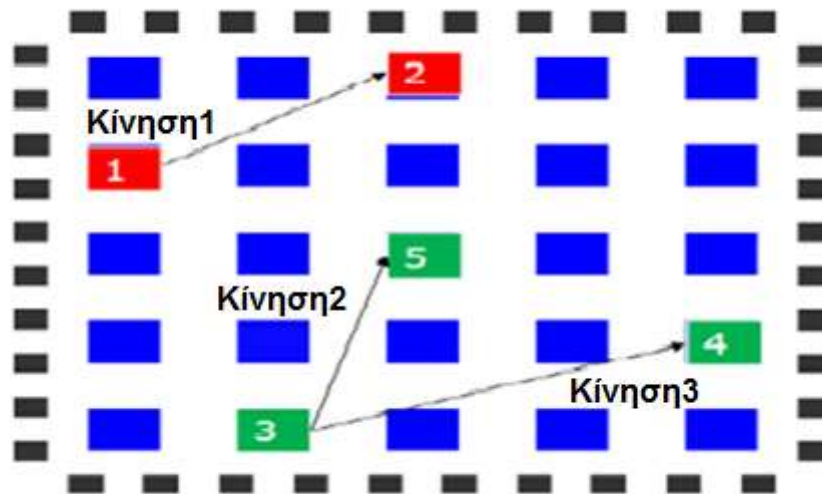
3.4.1. Προσέγγιση Παράλληλης Κίνησης (Parallel Move Approach)

Δεδομένου ότι υπάρχει ένας αρκετά μεγάλος αριθμός κινήσεων σε κάθε θερμοκρασία, το κίνητρο της παράλληλης προσέγγισης όσον αφορά την κίνηση είναι να προσπαθεί να επιταχύνει τη διαδικασία προσομοιωμένης απόδοσης με το να εκτελεί πολλές κινήσεις ταυτόχρονα. Ως εκ τούτου λοιπόν υπάρχουν μερικές πιθανές περιπτώσεις μετά από κάθε κίνηση. Αυτές λοιπόν οι κινήσεις είναι οι εξής:

- i. δύο τετράγωνα να ανταλλαχθούν
- ii. ένα μπλοκ να μετακινηθεί σε μια κενή θέση
- iii. η κίνηση να απορριφθεί

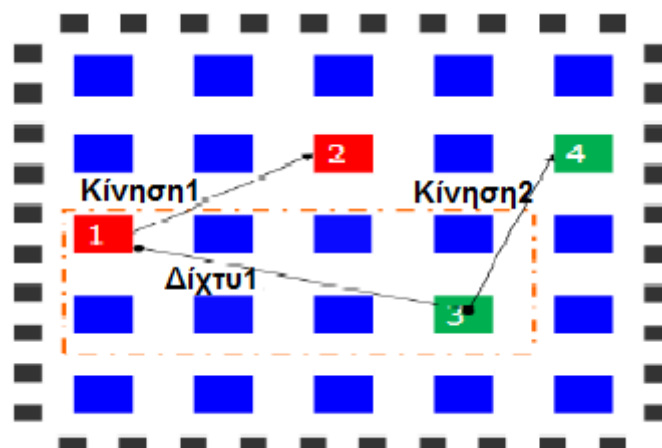
Οι κινήσεις μπορούν να επιτευχθούν παράλληλα μόνον εφόσον δεν μετακινούν το ίδιο μπλοκ ή μετακινούνται προς την ίδια κατεύθυνση.

Στην παρακάτω εικόνα παρουσιάζεται ένα απλό παράδειγμα της παράλληλης κίνησης. Ας σημειωθεί ότι η Κίνηση 1 και Κίνηση 2 μπορούν να γίνουν παράλληλα, δεδομένου ότι είναι εντελώς ανεξάρτητες, ενώ η Κίνηση 2 και Κίνηση 3 δεν μπορούν να επιτευχθούν επειδή προσπαθούν να μετακινήσουν το Μπλοκ 3 σε διαφορετικές τοποθεσίες ταυτόχρονα.



Εικόνα 17 Προσέγγιση της παράλληλης κίνησης

Ωστόσο, η εξασφάλιση των παραπάνω μπορεί να εγγυηθεί μόνο και εάν δεν υπάρχουν συγκρούσεις στην διαδικασία της κίνησης ενώ συγκρούσεις του κόστους δικτύου θα μπορούσαν ακόμη να υφίστανται. Όπως φαίνεται στην Εικόνα 18 τα Μπλοκ 1 και 3 ανήκουν στο ίδιο δίκτυο (Δίκτυο1). Ενώ η Κίνηση 1 και η Κίνηση 2 γίνονται παράλληλα, το προκύπτον πλαίσιο οριοθέτησης της Κίνησης 1 είναι το πλαίσιο οριοθέτησης του μπλοκ 2 και 3, ενώ το προκύπτον πλαίσιο οριοθέτησης της Κίνησης 2 είναι το πλαίσιο οριοθέτησης των μπλοκ 1 και 4. Δύο κινήσεις που μετακινούν μπλοκ του ίδιου δικτύου, μπορούν να αξιολογήσουν το πλαίσιο οριοθέτησης εσφαλμένα ως κάθε μία από τις κινήσεις δεν μπορούν να λάβουν υπόψη το γεγονός ότι η άλλη κίνηση αλλάζει το πλαίσιο οριοθέτησης.



Εικόνα 18 Κόστος παρεμπόδισης δικτύου

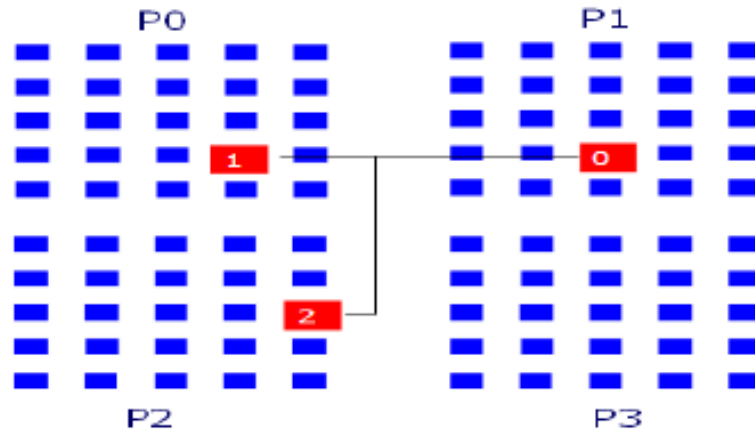
Υπάρχουν δύο τρόποι για να ασχοληθεί κανείς με τη σύγκρουση της κίνησης και το κόστος σύγκρουσης δικτύου.

- Να αγνοήσουμε τα σφάλματα στη συνάρτηση κόστους είναι ο ευκολότερος τρόπος για να αντιμετωπίσει κανείς με τις συγκρούσεις. Αυτό βέβαια έχει αρνητικές επιπτώσεις όσον αφορά την ακρίβεια του κόστους. Αυτό με την σειρά του επηρεάζει την αποδοχή των κινήσεων. Όλα τα παραπάνω έχουν ως αποτέλεσμα να επηρεάζουν αρνητικά τα αποτελέσματα.
- Βρίσκοντας τις ασύνδετες κινήσεις που όχι μόνο κινούν διαφορετικά μπλοκ, αλλά και ανήκουν σε διαφορετικά δίκτυα. Οι υπερβολικά περιορισμένες κινήσεις έχουν αποτέλεσμα σε μικρότερο χώρο ανταλλαγής και η επιβάρυνση συγχρονισμού τείνουν να συντρίψουν το κέρδος σε παραλληλισμό.

Και οι δύο αυτές μέθοδοι παρουσιάζουν αρνητική επιτάχυνση. Ο λόγος αυτός οφείλεται στην επιβάρυνση του συγχρονισμού που αντισταθμίζει τα πλεονεκτήματα του παραλληλισμού. Αλλά η σκέψη του να προσπαθήσουμε να παραλληλίσουμε τις κινήσεις, εμπνέει πολλές άλλες παράλληλες μεθόδους τοποθέτησης FPGA.

3.4.2. Area Based Approach (Προσέγγιση που βασίζεται στην περιοχή)

Η προσέγγιση που βασίζεται στην περιοχή υποκινείται από την επίλυση της σύγκρουσης η οποία απεικονίζεται στην παράλληλη προσέγγιση κίνησης. Το πρόβλημα σε εκείνη την προσέγγιση ήταν ότι οι κινήσεις ήταν πολύ περιορισμένες. Σε αυτή την προσέγγιση προσπαθούμε να «ελαφρύνουμε» αυτό το πρόβλημα με το να διαμερισματοποιήσουμε την περιοχή του FPGA και να εκχωρήσουμε τα διαμερίσματα σε διαφορετικούς επεξεργαστές. Όπως φαίνεται στην παρακάτω εικόνα, όλο κύκλωμα χωρίζεται σε τέσσερα τμήματα (διαμερίσματα) και κάθε επεξεργαστής είναι υπεύθυνος για ένα διαμέρισμα. Ο κάθε επεξεργαστής είναι ελεύθερος να κινήσει μπλοκ στην περιοχή του. Οι κινήσεις που αξιολογούνται είναι αρκετά λιγότερο περιορισμένες από ό, τι στην προσέγγιση παράλληλης κίνησης. Ωστόσο, οι συγκρούσεις θα μπορούσαν ακόμα να πραγματοποιηθούν, διότι οι πολλαπλοί επεξεργαστές μπορούν να μετακινήσουν μπλοκ που ανήκουν στο ίδιο δίκτυο όπως παρουσιάζεται στην εικόνα.



Εικόνα 19 Σύγκρουση σε κατατεμημένη περιοχή

Για παράδειγμα, το πλαίσιο οριοθέτησης του μπλοκ 1, 2 και 3 δεν μπορεί να υπολογιστεί, δεδομένου ότι ανήκουν σε διαφορετικά διαμερίσματα. Αυτά τα λάθη μπορούν να γίνουν ανεκτά, διότι δεν περιμένουμε το δίκτυο, το οποίο εκτείνεται σε πάνω από δύο ή περισσότερα διαμερίσματα να συμβαίνει πολύ συχνά. Επιπλέον, με τη θερμοκρασία να μειώνεται, οι ανταλλαγές τείνουν γίνονται μεταξύ των κοντινών μπλοκ. Δεδομένου ότι κάθε επεξεργαστής μπορεί να κινηθεί μόνο μπλοκ εντός της δικιάς του κατατεμημένης περιοχής, για να επιτραπεί στην τοποθέτηση να φθάσει στο ολικό ελάχιστο, το διαμέρισμα θα πρέπει να δημιουργηθεί προσεκτικά ώστε κάθε μπλοκ να έχει την ελευθερία να κινηθεί προς οποιαδήποτε αυθαίρετη περιοχή στο FPGA. Η προσέγγιση βασισμένη στην περιοχή χρησιμοποιεί τόσο οριζόντια όσο και κάθετη κατάτμηση για να διασφαλιστεί το ολικό ελάχιστο μπορεί να επιτευχθεί [28].

Τα πειραματικά αποτελέσματα δείχνουν ότι μια μη γραμμική επιτάχυνση έχει καταφερθεί εν συγκρίσει με τον ακολουθιακό τοποθετητή και το κόστος δεν υποβαθμίζεται με την αύξηση των επεξεργαστών. Αυτό οφείλεται στις λιγότερες απαιτήσεις συγχρονισμού [23].

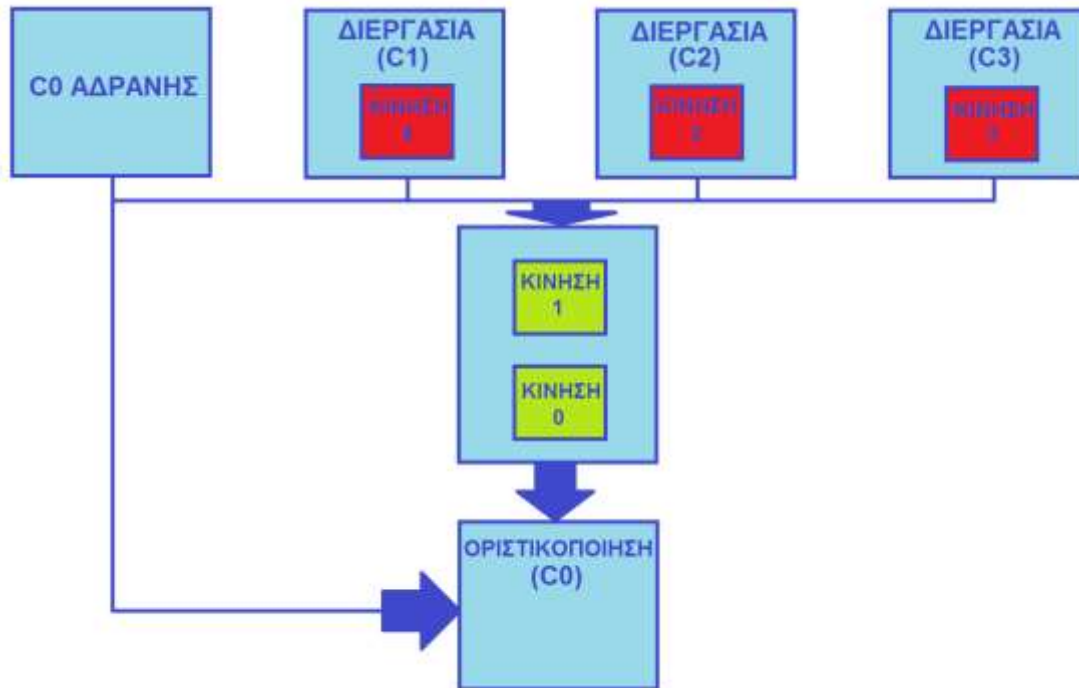
3.4.3. Deterministic Parallel Approach (Ντετερμινιστική Παράλληλη Προσέγγιση)

Ένας από τους περιορισμούς του παραλληλισμού είναι ο μη-ντετερμινισμός των αποτελεσμάτων, δηλαδή η μη τυποποιημένη παράλληλη προσέγγιση. Αυτό έχει προσδιοριστεί σπάνια στις παρελθοντικές εργασίες (εξάιρεση βρίσκουμε στο paper [9]), αλλά είναι ζωτικής σημασίας σε ένα εμπορικό πλαίσιο για τους ακόλουθους δύο λόγους [22]:

- ❖ Όταν ο χρήστης χρησιμοποιεί ένα εμπορικό εργαλείο τοποθέτησης FPGA, πρέπει να είναι σε θέση να αναπαράγει το πρόβλημα, όταν ένα λάθος του εργαλείου είναι να αναφερθεί. Ο μη – ντετερμινισμός κάνει αυτό εξαιρετικό δύσκολο, γιατί τα αποτελέσματα είναι διαφορετικά σε κάθε εκτέλεση.
- ❖ Στο στάδιο των δοκιμών κατασκευής ενός τοποθετητή, θα ήταν τρομερά δύσκολο να εξεταστούν οι αποτυχημένες δοκιμές, ελλείψει δεδομένων καθώς τα αποτελέσματα δύναται να αλλάξουν τυχαία.

Επιπροσθέτως στο ντετερμινισμό, υπάρχει ένας ισχυρότερος περιορισμός που μπορούμε να εφαρμόσουμε στον αλγόριθμο, γνωστός ως σειριακή ισοδυναμία. Αυτή είναι η ιδιότητα που ο αλγόριθμος πρέπει να δώσει την ίδια ακριβώς απάντηση, ανεξαρτήτως από το πόσοι πολλοί επεξεργαστικές πυρήνες χρησιμοποιούνται. Ένας σειριακός ισοδύναμος αλγόριθμος είναι επίσης και καθαρά ντετερμινιστικός.

Ο αλγόριθμος που προτείνεται στο [22], παραλληλίζει την τοποθέτηση, ενώ την ίδια στιγμή, διατηρεί τα αποτελέσματα ντετερμινιστικά. Η ντετερμινιστική παράλληλη προσέγγιση διαχωρίζει μια κίνηση σε δύο στάδια : την επεξεργασία και την οριστικοποίηση . Όπως παρουσιάζεται στην εικόνα 20, κατά το στάδιο της επεξεργασίας, κάθε επεξεργαστής προσδιορίζει μια κίνηση και την αξιολογεί. Αυτό δαπανά ένα πολύ μεγάλο κομμάτι του χρόνου και ως εκ τούτου εμφανίζεται παράλληλα. Προκειμένου να αποφευχθεί η σύγκρουση και να διατηρηθεί η ντετερμινιστικότητα, οι υπολογιζόμενες κινήσεις μπαίνουν σε σειρά και μία σχέση εξάρτησης που απαιτείται για να εξασφαλιστεί ότι δεν υπάρχουν συγκρούσεις και δεν επαναπροτείνονται κινήσεις που έχουν ήδη συγκρουστεί. Ας σημειωθεί ότι το κομμάτι της οριστικοποίησης μπορεί να γίνει από οποιοδήποτε επεξεργαστή σε αδράνεια. Στο παράδειγμα μας, ο C0 είναι σε αδράνεια όταν όλες οι κινήσεις στην ουρά έχουν ελεγχθεί, οπότε ο C0 κάνει την δουλειά της οριστικοποίησης.



Εικόνα 20 Ντετερμινιστική παράλληλη προσέγγιση

Επιπροσθέτως, υπάρχουν πολλά πλεονεκτήματα της ντετερμινιστική παράλληλης προσέγγισης. Πρώτον, η επιτάχυνση μπορεί να είναι γραμμική, με δεδομένη την υπόθεση ότι ο χρόνος ολοκλήρωσης είναι αμελητέος. Έπειτα και δεύτερο πλεονέκτημα αποτελεί ότι μία κίνηση βρίσκεται σε επεξεργασία εξ ολοκλήρου από έναν επεξεργαστή, η οποία βελτιώνει την τοποθεσία μνήμης (memory locality). Τρίτον, τα αποτελέσματα είναι ντετερμινιστικά και σειριακά ισοδύναμα.

3.5. Βελτίωση του αλγορίθμου

Σε αυτό το κεφάλαιο θα δούμε μία βελτίωση που έχει γίνει στον αλγόριθμο της προσομοιωμένης ανόπτησης που χρησιμοποιεί το εργαλείο VPR, με επιτάχυνση τουλάχιστον διπλάσια (2x) χωρίς να χάνεται η ποιότητα των αποτελεσμάτων. Όπως είδαμε στην ανάλυση και περιγραφή του αρχικού αλγόριθμου τοποθέτησης VPR, βασίζεται στην προσομοιωμένη

ανόπτηση και χρησιμοποιεί μια τυχαία ρουτίνα για την επιλογή των μπλοκ που θα ανταλλάγουν όταν προσπαθεί να βελτιώσει το κόστος τοποθέτησης. Ούτε η επιλογή του μπλοκ προέλευσης για να ανταλλαγεί ούτε η επιλογή της θέσης για να τοποθετηθεί απαιτεί οποιαδήποτε σύνθετη ρουτίνα. Η μόνη απαιτούμενη ρουτίνα είναι να δημιουργήσει τυχαίες συντεταγμένες. Η ρουτίνα αυτή φαίνεται παρακάτω. Παρά τα καλά αποτελέσματα, ο αναγκαίος αριθμός των κινήσεων για την επίτευξη καλής ποιότητας αποτελεσμάτων είναι εξαιρετικά υψηλός και αυξάνεται εκθετικά με το ύψος των λογικών μπλοκ που πρέπει να τοποθετηθούν. Το πιο σημαντικό μέρος της ρουτίνας τοποθέτησης VPR είναι η συνάρτηση που εμφανίζεται παρακάτω, που ονομάζεται TRY_SWAP () και είναι υπεύθυνη για την επιλογή των μπλοκ και τις θέσεις αυτών που προσπαθεί να προσπαθήσει να ανταλλάξει. Παρουσιάζομαι την συνάρτηση TRY_SWAP

Πίνακας 4 TRY_SWAP συνάρτηση

```

Procedure TRY_SWAP
  b_from, b_to
  x_to, y_to
  T
  choice b_from
  choice x_to e y_to
  if position(x_to,y_to) empty then
    move b_from to position( x_to,y_to)
  else
    b_to = block(x_to,y_to)
    swap (b_from , b_to)
  endif
  calculate cost variation ( $\Delta C$ )
  Take  $x \in [0, 1]$ ;
  if( $x < e^{-\Delta C/T}$ ) then
    return move accept;
  else
    move is rejected (undo move);
    return move rejected;
  endif
end TRY_SWAP

```


Για κάθε τιμή της θερμοκρασίας, και ανάλογα με την τιμή του `inner_num`, σε κάθε επανάληψη η ρουτίνα `try_swap ()` καλείται, ένα μπλοκ προέλευσης και η θέση προορισμού επιλέγεται τυχαία, είτε μέχρι το ποσό των κινήσεων να έχει φθάσει στα όριά του ή η θερμοκρασία δικτύου να έχει φτάσει στο ελάχιστο όριο της.

Γίνεται τροποποίηση τριών σημείων στο εργαλείο VPR.

- Να αντικαταστήσει την τυχαία αρχική τοποθέτηση από μια εποικοδομητική νέα ευρετική μέθοδο για την αρχική τοποθέτηση
- Να φτιάξει μία νέα συνάρτηση για τον υπολογισμό της αρχικής θερμοκρασίας ανόπτωσης
- Να τροποποιήσει τη ρουτίνα `TRY_SWAP`, ώστε να ελαχιστοποιηθεί η τυχαία επιλογή στη θέση προορισμού για να τοποθετεί ένα επιλεγμένο μπλοκ.

Όπως παρουσιάζει στο `paper [2]`, σε ένα `kxk` πίνακα `L`, αντί για τυχαία τοποθέτηση όλης της συνδεσμολογίας του κυκλώματος (`netlist`) των `CLBs` και των `IOBs`, να τοποθετούνται μόνο οι βασικές έξοδοι τυχαία κατά μήκος της περιφέρειας του `L`. Για το δοθέν κύκλωμα καθορισμένο ως συνδεσμολογία κυκλώματος (`netlist`), ας ορίσουμε έναν κατευθυνόμενο γράφο $D = \langle V, E \rangle$, όπου $V = \{ v \mid v \text{ είναι είτε ένα CLB, είτε ένα IOB} \}$ και $E = \{ \langle v_i, v_j \rangle \mid v_i \in \text{fanin}(v_j) \text{ και } v_j \in \text{fanout}(v_i) \}$.

Για κάθε μία από τα βασικές εξόδους που παρουσιάζεται στη συνδεσμολογία κυκλώματος (`netlist`), ένας κώνος ορίζεται. Ένας κώνος f_i του O_i είναι το σύνολο αποτελούμενο από το O_i και όλους τους προγόνους του [18]. Δηλαδή $f_i = \text{cone}(O_i) = \{ u \mid \exists \text{ ένα απλό κατευθυνόμενο μονοπάτι από το } u \text{ στο } O_i \}$. Η κορυφή του κώνου είναι η βασική έξοδος O_i και θέτει το επίπεδο της από 1 να γίνει 0. Διασχίζουμε τα fan-in μπλοκ σε κάθε μπλοκ στο προηγούμενο επίπεδο με κατά πλάτος τρόπο, μέχρι βρούμε καθόλου νέα `CLBs` ή βασικές εισόδους για τον κώνο f_i . Η διάσχιση κατά πλάτους του κώνου οδηγεί σε μία δομή δέντρου με το O_i σαν ρίζα.

Έτσι, όταν τοποθετούνται όλα τα `CLBs` και τα μπλοκ εισόδου σε O_i ρίζα γείτονα. Αυτό δίνει την αρχική διαμόρφωση τοποθέτησης για τεχνολογικά-αντιστοιχισμένη συνδεσμολογία κυκλώματα (`netlist`) καθορισμένη, ως είσοδος σε μια επαναληπτική διαδικασία για την περαιτέρω βελτίωση της διαμόρφωσης της τοποθέτησης. Σε αντίθεση με το VPR, το αρχικό κόστος καλωδίωσης βελτιώθηκε κατά μέσο όρο πενήντα τοις εκατό. Ως εκ τούτου, για την περαιτέρω βελτίωση της τοποθέτησης, μια προσομοιωμένη ανόπτωση εξαιρετικά χαμηλής θερμοκρασίας εκτελείται σε αυτή την αρχική τοποθέτηση για να ληφθεί η τελική διαμόρφωση τοποθέτησης.

Στη συνέχεια, είναι αναγκαίος ένα νέος τρόπος για να υπολογιστεί η αρχική θερμοκρασία για την ανόπτωση. Επειδή, στην παλιά μορφή, η εκτέλεση των N ($N = \# \text{CLBs} + \# \text{IOBs}$) κινήσεων σε υψηλή θερμοκρασία, η καλή αρχική τοποθέτηση είναι πολύ διαταραγμένη και επιδεινώθηκε. Παρακάτω παρουσιάζεται η τροποποιημένη `TRY_SWAP` συνάρτηση.

Πίνακας 5 Τροποποιημένη TRY_SWAP συνάρτηση

```

Procedure Modified_TRY_SWAP
  define b_from;
  define, b_to;
  define x_to, y_to
  define T
  define FACTOR
  choice b_from
  choice x_to and y_to
  do
    if position(x_to, y_to) is empty then
      move b_from to position(x_to,y_to)
    else
      b_to = block at position(x_to,y_to)
      swap( b_from, b_to)
    endif
  calculate cost variable ( $\Delta$ )
  Take  $x \in [0,1]$ ;
  if ( $x < e^{-\Delta T}$ ) then
    return accept
  else
    undo move
    if (b_from == CLB and T < 10 and total moves < FACTOR)
      b_to = search neighbor to b_to
    endif
  endif
  enddo ( move not accept and total moves < FATOR and b_from=CLB)
  return rejected
end Modified_TRY_SWAP

```

Μια νέα μορφή για τον υπολογισμό της θερμοκρασίας προσπαθεί να κινήσει όλα τα μπλοκ, ένα προς ένα, με μηδενική θερμοκρασία. Η αρχική θερμοκρασία στη συνέχεια ορίζεται από $t_{init} = 0.035\phi$ στην τοποθέτηση με γνώμονα το μήκος της καλωδίωσης, καθώς και $t_{init} = 54.05 \phi / \text{num_blocks}$ για τοποθέτηση με γνώμονα τον χρονισμό (ϕ = τυπική απόκλιση του κόστους σε σχέση με N ανταλλαγές και $\text{num_blocks} = \# \text{CLBs} + \# \text{IOBs}$, τις τιμές 0.035 και 54.05 ορίστηκαν για να αναγκάσουμε ίδιες συνθήκες σε VPR).

Στην ρουτίνα TRY_SWAP, ένα μεταβλητό στοιχείο FACTOR περιορίζει μια σειρά από προσπάθειες για ανταλλαγές χρησιμοποιώντας κάποιο μπλοκ προέλευσης, της οποίας για κάθε εύρος θερμοκρασιών υπάρχουν ειδικές τιμές που ορίζονται εμπειρικά.

Πίνακας 6 Εύρος του FACTOR

Θερμοκρασία (t)	Τιμή FACTOR
$t > 50.0$	2
$2.0 \leq t \leq 50.0$	1
$0.005 \leq t \leq 2.0$	4
$0.0025 \leq t \leq 0.005$	2
$t \leq 0.0025$	1

Ορίζεται ότι από μια δεδομένη θερμοκρασία, σε κάθε επανάληψη, αφού το μπλοκ προέλευσης και η θέση προορισμού επιλέγονται, εάν μια κίνηση δεν είναι αποδεκτή, μια νέα κίνηση επιχειρείται, διατηρώντας το μπλοκ προέλευσης και επιλέγοντας μια νέα θέση προορισμού. Στη συνέχεια, μια άλλη κίνηση μπορεί να ελεγχθεί χωρίς την ανάγκη της επιλογής ενός νέου μπλοκ προέλευσης. Εάν η αρχική κίνηση δεν γίνεται δεκτή και το μπλοκ προέλευσης είναι ένα CLB, τότε μια κενή θέση στο μπλοκ προορισμού άμεση γειτνίασης αναζητείται. Σε περίπτωση επιτυχίας, μια προσπάθεια να ανταλλάξουν το μπλοκ προέλευσης στη θέση αυτή γίνεται, αντιστρόφως, όποια γειτονικού μπλοκ θέση μπορεί να χρησιμοποιηθεί για να εκτελέσει την ίδια προσπάθεια. Αυτή η διαδικασία επαναλαμβάνεται έως ότου η κίνηση γίνει αποδεκτή ή να επιτευχθεί το όριο των προσπαθειών που ορίζονται μέσω της μεταβλητής FACTOR.



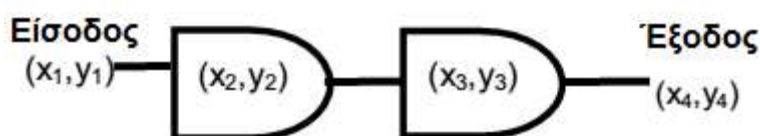
Εικόνα 21 Αναζήτηση στην γειτονία

Τέλος, ένα νέος προεπιλεγμένος αριθμός των κινήσεων που αξιολογούνται σε κάθε θερμοκρασία ορίζεται από $\text{mon_lim} = \eta \cdot 10 \cdot (N_{\text{blocks}})^{1.33}$. Σε περίπτωση τοποθέτησης με γνώμονα το μήκος της καλωδίωσης, $\eta = 1/3$ και στην περίπτωση της τοποθέτησης με γνώμονα τον χρονισμό, $\eta = 3/4$, μειώνοντας τον αριθμό των κινήσεων ανά θερμοκρασία και επιταχύνει την τοποθέτηση χωρίς να προκαταλαμβάνει το τελικό αποτέλεσμα. Η απώλεια στην ποιότητα του αποτελέσματος αποζημιώνεται από το κέρδος που παρέχονται από νέες συναρτήσεις που αναδείχθηκαν παραπάνω. Στα αποτελέσματα της έρευνας βλέπουμε τις συγκρίσεις με το VPR βλέπουμε ότι ο χρόνος διεργασίας μειώθηκε έως 56% χωρίς σημαντικές αλλαγές στην ποιότητα των τελικών αποτελεσμάτων, ίσως και βελτίωση σε κάποιες περιπτώσεις

Quadratic Placement (Τετραγωνική τοποθέτηση)

4.1 Γενικά Στοιχεία

Η μέθοδος της τετραγωνικής τοποθέτησης χρησιμοποιεί το τετράγωνο του μήκους καλωδίωσης ως στοιχείο ελέγχου στην συνάρτηση που θέτει και προσπαθεί να ελαχιστοποιήσει το κόστος με την επίλυση των γραμμικών εξισώσεων [38]. Αν και η μέθοδος αυτή λαμβάνει υπόψη μόνο το τετραγωνικό μήκος της καλωδίωσης μπορεί να τελειώσει αποτελεσματικά τη διαδικασία τοποθέτησης με σχεδόν καμία μείωση της ποιότητας. Ο τετραγωνικός αλγόριθμος προσπαθεί να ελαχιστοποιεί το συνολικό τετραγωνικό μήκος από την επίλυση γραμμικής εξίσωσης [12]. Ένα παράδειγμα κυκλώματος φαίνεται στο παρακάτω σχήμα.



Η συνάρτηση κόστους είναι το τετραγωνικό άθροισμα της απόστασης από την πηγή στον προορισμό του κάθε σημείου στο μονοπάτι. Μια συνάρτηση κόστους για το κύκλωμα του παραδείγματος, έχει ως εξής:

$$C=(x_1-x_2)^2+(y_1-y_2)^2+(x_2-x_3)^2+(y_2-y_3)^2+(x_3-x_4)^2+(y_3-y_4)^2$$

4.2 Υλοποίηση Αλγορίθμου

Αυτή η συνάρτηση κόστους επεκτείνεται για ολόκληρο το κύκλωμα στο FPGA. Το αρχείο εισόδου στη περίπτωση της τετραγωνικής τοποθέτησης είναι ένα υπερ-γράφημα συνδεσμολογίας του κυκλώματος (netlist) και η διαδικασία αυτή προσπαθεί να ελαχιστοποιήσει το σύνολο των τετραγώνων των αποστάσεων μεταξύ κάθε δύο κόμβων. Το κόστος υπολογίζεται σύμφωνα με τον τύπο:

$$\Phi(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \sum_{i=1, j=1}^m W_{i,j} [(x_i - x_j)^2 + (y_i - y_j)^2]$$

Όπου \mathbf{x} και \mathbf{y} είναι οι συντεταγμένες του λογικού μπλοκ στην συνδεσμολογία κυκλώματος (netlist). Το $W_{i,j}$ είναι το βάρος της ακμής που συνδέει του κόμβους $(\mathbf{x}_i, \mathbf{y}_i)$ και $(\mathbf{x}_j, \mathbf{y}_j)$. Δεδομένου ότι η είσοδος της τετραγωνικής τοποθέτησης συνήθως παρίσταται με ένα υπερ-γράφημα, οι δύο κόμβοι μπορούν να συνδεθούν με πολλά περισσότερα από ένα δίκτυο. Υπάρχουν δύο μοντέλα για να μετατραπεί το υπερ-γράφημα σε γράφημα. Η συνάρτηση κόστους μπορεί να ξαναγραφεί σε μορφή πινάκων:

$$\Phi(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{d}_x^T \mathbf{x} + \frac{1}{2} \mathbf{y}^T \mathbf{Q} \mathbf{y} + \mathbf{d}_y^T \mathbf{y} + const$$

,όπου \mathbf{Q} είναι ένας $\mathbf{n} \times \mathbf{n}$ συμμετρικός πίνακας και $\mathbf{d}_x, \mathbf{d}_y$ n-διαστάσεων διανύσματα. Καθώς η συνάρτηση κόστους μπορεί να διαχωριστεί σε x διάσταση και y διάσταση αντίστοιχα, κάθε διάσταση μπορεί να εξεταστεί ξεχωριστά και να επιλυθεί ως γραμμικό σύστημα. Στη συνέχεια, η συνάρτηση γράφεται ως εξής με υπολογιζόμενη μόνο μία διάσταση:

$$\Phi(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{d}_x^T \mathbf{x} + const$$

Στα πλαίσια εύρεσης της ελάχιστης τιμής, κάνουμε $\nabla \Phi(\mathbf{x}) = 0$ που έχει σαν αποτέλεσμα την ακόλουθη εξίσωση πινάκων:

$$\mathbf{Q}_x + \mathbf{d}_x = 0$$

Αυτή η τετραγωνική εξίσωση ελαχιστοποιεί το συνολικό τετραγωνικό μήκος της καλωδίωσης και μπορεί να λυθεί με τη χρήση μη στάσιμων επαναληπτικών μεθόδων. Ο συγκεκριμένος αλγόριθμος που προτείνεται στο [38] μπορεί να χωριστεί σε τρία στάδια. Ο στόχος του πρώτου σταδίου είναι να επιτύχει μία καλή αρχική τοποθέτηση. Σε αυτό το στάδιο, επανειλημμένα δημιουργούμε, μορφοποιούμε και λύνουμε γραμμικές εξισώσεις για να φθάσουμε προοδευτικά σε μία καλύτερη τοποθέτηση. Σε κάθε επανάληψη, πρώτα δημιουργούμε γραμμικές εξισώσεις και τις λύνουμε για να βρούμε τις συντεταγμένες από κάθε κόμβο στο κύκλωμα. Στην συνέχεια αντιστοιχούμε τους κόμβους σε ολόκληρη την περιοχή του τσιπ σύμφωνα με τις τρέχουσες συντεταγμένες. Τελικά χρησιμοποιούμε αυτές τις νέες

τοποθεσίες ως αναφορές και εισάγουμε επιπλέον «χαζούς» κόμβους στο κύκλωμα. Αυτοί οι νέοι κόμβοι χρησιμοποιούνται για να μορφοποιήσουν τον πίνακα των συντελεστών στην επόμενη επανάληψη. Κατά την διάρκεια κάθε επανάληψης, πραγματοποιούμε επίσης γραμμική προσαρμογή για να ελαχιστοποιηθεί το γραμμικό μήκος καλωδίωσης, καθώς και το τετραγωνικό μήκος καλωδίωσης. Αυτό επιτυγχάνεται τροποποιώντας τον πίνακα συντελεστών των τετραγωνικών εξισώσεων. Το πρώτο στάδιο διαρκεί μέχρις ότου δεν μπορεί να επιτευχθεί καμία σημαντική βελτίωση.

Πρώτο Στάδιο

- Δημιουργία και επίλυση γραμμικών εξισώσεων
- Αντιστοίχιση του κυκλώματος στο FPGA
- Εισαγωγή «χαζών» κόμβων και επέκταση της τοποθέτησης
- Βελτιστοποίηση για ελαχιστοποίηση γραμμικού μήκους καλωδίωσης
- Επανάληψη μέχρι να μην υπάρχει σημαντική βελτίωση

Όταν αφήνουμε το πρώτο στάδιο, έχουμε μία αρκετά καλή νόμιμη τοποθέτηση. Στο δεύτερο στάδιο προσπαθούμε να βελτιώσουμε αυτή την τοποθέτηση, χρησιμοποιώντας παρόμοια γραμμική τεχνική μείωσης του μήκους της καλωδίωσης. Δεν δημιουργούμε και επιλύουμε γραμμικές εξισώσεις σε αυτό το στάδιο. Αντ' αυτού μετακινούμε τους κόμβους κατευθείαν για να μειώσουμε το συνολικό γραμμικό μήκος καλωδίωσης. Καθώς δεν χρειάζεται επίλυση εξισώσεων σε αυτό στάδιο, αυτή η διεργασία είναι πολύ ταχύτερη από την γραμμική προσαρμογή στο πρώτο στάδιο και μπορούμε να κάνουμε περισσότερες επαναλήψεις και να πάρουμε μία καλύτερη βελτίωση.

Δεύτερο Στάδιο

- Βελτιστοποίηση για ελαχιστοποίηση γραμμικού μήκους καλωδίωσης βασισμένο στη νόμιμη τοποθέτηση, μέχρις ότου δεν υπάρχει καμία σημαντική βελτίωση
- Επανα-αντιστοίχιση του κυκλώματος στο FPGA

Τελικά, στο τρίτο στάδιο η τοποθέτηση βελτιώνεται από προσομοιωμένη ανόπτηση χαμηλής θερμοκρασίας, για περαιτέρω ελαχιστοποίηση του μήκους καλωδίωσης

4.3 Πλεονεκτήματα και Μειονεκτήματα

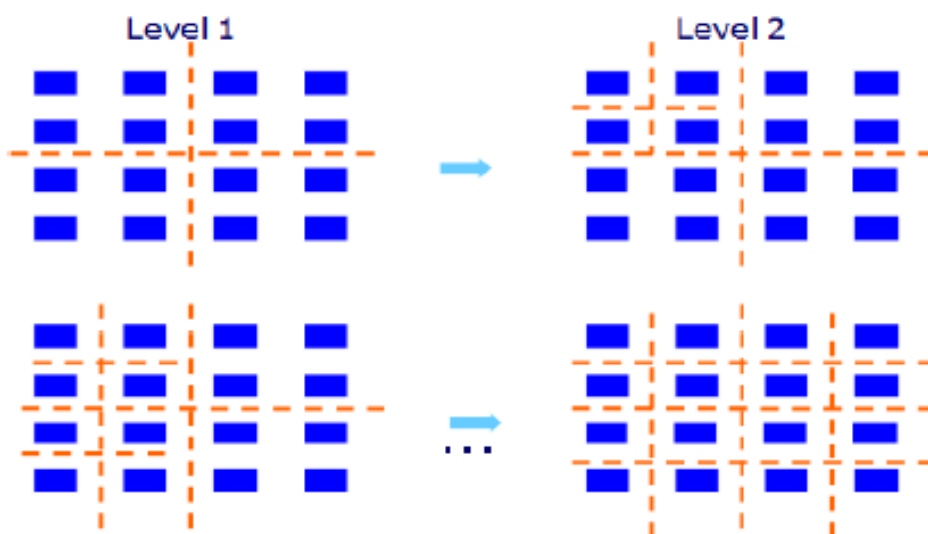
Το κύριο πλεονέκτημα της μεθόδου τετραγωνικής τοποθέτησης είναι ότι βελτιώνει σημαντικά το χρόνο λειτουργίας με σχεδόν καμία απώλεια ποιότητας σε σύγκριση με άλλες τεχνικές βελτιστοποίησης. Επίσης, είναι σε θέση να χειριστεί μεγάλη σχεδίαση χωρίς να επηρεάζει το κόστος του χρόνου υπολογισμού. Σύμφωνα με τα αποτελέσματα της μελέτης του Xu [38] σε όλα τα κυκλώματα αναφοράς 20 MCNC ο αλγόριθμος τετραγωνικής τοποθέτησης τρέχει 5,8 φορές ταχύτερα από ό, τι κατά μέσο όρο ένα VPR, ενώ το μήκος της καλωδίωσης που λαμβάνεται με τον εν λόγω αλγόριθμο είναι μόνο 1,9% σε σχέση με τον VPR [5]. Με την χρήση καλύτερων μεθόδων άλγεβρας για την επίλυση γραμμικών εξισώσεων ο χρόνος τοποθέτησης μπορεί να μειωθεί περαιτέρω. Ωστόσο δεδομένου ότι το τετράγωνο του μήκους καλωδίωσης είναι ο μόνος παράγοντας που λαμβάνεται υπόψη στην συνάρτηση κόστους, το τμήμα χρόνου της τοποθέτησης δεν μπορεί να εμφανιστεί στην τετραγωνική τοποθέτηση.

Min-Cut Placement (min-cut τοποθέτηση)

5.1 Γενικά Στοιχεία

Οι αλγόριθμοι κατάτμησης χαρακτηρίζονται για την ταχύτητα των αποτελεσμάτων τους και την ευκολία επεκτασιμότητας για Application Specific Integrated Circuits (Ολοκληρωμένα Κυκλώματα Ειδικής Εφαρμογής (ASICs)), και η τοποθέτηση μπορεί να εφαρμοστεί και σε κυκλώματα FPGAs. Μία από τις πρόσφατες τεχνικές τοποθέτησης κατάτμησης είναι η min-cut τοποθέτηση που αναδρομικά διαμερισματοποιεί την συνδεσμολογία κυκλώματος (netlist) σε αυξανόμενα μικρότερα υπό-κυκλώματα. Κυρίως διαμερισματοποιεί στην μέση για να αντιστοιχίσει τη συνδεσμολογία κυκλώματος (netlist) στην περιοχή διάταξης του FPGA. Η τεχνική ελαχιστοποιεί τον αριθμό των κοψιμάτων των δικτύων ενώ εν τω μεταξύ, αφήνει τα υψηλά συνδεόμενα λογικά μπλοκ σε ένα διαμέρισμα [23], [25].

Η καθυστέρηση της βελτιστοποίησης είναι ένα πολύ σημαντικό κομμάτι στο σχεδιασμό του κυκλώματος ενώ η αποτελεσματική ελαχιστοποίηση καθυστέρησης σε μεγάλα κυκλώματα είναι δυνατή μόνο αν υπολογιστεί η επίδοση όσο δυνατόν νωρίτερα στη ροή του σχεδιασμού. Η min-cut τοποθέτηση στοχεύει την ελαχιστοποίηση της καθυστέρησης στο στάδιο της τοποθέτησης, το οποίο είναι ένα πρώιμο στάδιο στην διαδικασία σχεδιασμού. Ο min-cut τοποθετητής εφαρμόζει μία βασική μέθοδο του διαίρει και βασίλευε. Ένα κύκλωμα αναδρομικά διαμερισματοποιήθηκε στην μέση σε ένα εύρος πρώτου τρόπου όπως φαίνεται στο παρακάτω σχήμα



Εικόνα 22 Στάδια υλοποίησης του αλγόριθμου κατάτμησης (διαμερισμού)

Η κατεύθυνση κοπής (οριζόντια ή κάθετη) αποφασίζεται με βάση την κρισιμότητα των δικτύων που διαπερνούν τα τέσσερα όρια, ώστε να ελαχιστοποιούνται οι συνολικοί αριθμοί κοπής. Αυτή η αναδρομική διαδικασία επαναλαμβάνεται μέχρις ότου κάθε διαμέρισμα να περιέχει μόνο ομαδοποιημένα μπλοκ (με στόχο την μείωση του κόστους). Ο στόχος του αλγόριθμου min-cut είναι να εντοπίσει το κατάλληλο διαμέρισμα που κόβει λιγότερα καλώδια στο δίκτυο. Όλες οι ακμές στο δίκτυο σταθμίζονται με το χρονοδιάγραμμα κρισιμότητας, καθώς και του τερματικού ευθυγράμμισης των κρίσιμων δικτύων.

5.2 Υλοποίηση Αλγόριθμου

Ο αλγόριθμος μπορεί να διαιρεθεί σε τρία στάδια. Στο πρώτο στάδιο, χρησιμοποιείται ο πολυεπίπεδος διαμέριστης hMetis[1], σαν μηχανή κατάτμησης. Κατά τη διαδικασία του διαχωρισμού η στενή σύνδεση ανάμεσα στο γράφημα του κυκλώματος και την τοποθέτηση διατηρείται, η οποία αντιπροσωπεύει τις συντεταγμένες όλων των μπλοκ στο ιστό του FPGA. Η αναδρομική διαχωρισμού γίνεται μέχρι κάθε διαμέρισμα φύλλο να έχει μόνο λίγα τετράγωνα μακριά, ενώ σε ορισμένες περιπτώσεις, μερικοί κόμβοι φύλλων μπορεί να περιέχουν περισσότερους κόμβους από ό, τι μπορούν να φιλοξενήσουν, ώστε οι επικαλύψεις να αφαιρούνται. Σε δεύτερο στάδιο, οι επικαλύψεις αφαιρούνται χρησιμοποιώντας μια τεχνική, όπου το μπλοκ κινείται στο πλησιέστερο καλύτερο ευθυγραμμισμένο διαμέρισμα. Τέλος, η τοποθέτηση τελειοποιείται με τη χρήση χαμηλής θερμοκρασίας προσομοιωμένης για την περαιτέρω ελαχιστοποίηση της καθυστέρησης.

5.3 Πλεονεκτήματα και Μειονεκτήματα

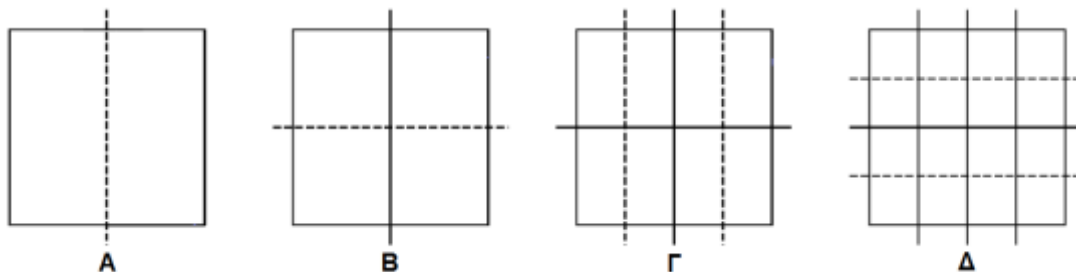
Το πλεονέκτημα της τεχνικής τοποθέτησης min-cut είναι η μείωση της καθυστέρησης στο στάδιο της τοποθέτησης, η οποία καθορίζει το σχεδιασμό ενός αποδοτικού κυκλώματος. Επίσης έχει ανοιχτή συνάρτηση κόστους που μπορεί να εφαρμοστεί είτε συνάρτηση με βάση το μήκος της καλωδίωσης είτε συνάρτηση με βάση τον χρονοισμό.

Εκτός αυτού, ο χρόνος λειτουργίας δείχνει ότι είναι τρεις με τέσσερις φορές πιο γρήγορο σε σχέση με τον αλγόριθμο που χρησιμοποιεί το VPR στις 20 MCNC ως σημείο αναφοράς, με μια ελαφρά υποβάθμιση της ποιότητας [23]. Ωστόσο, ο αλγόριθμος δεν είναι σε θέση να φθάσει σε ολικό ελάχιστο. Τα αποτελέσματα του min-cut αλγορίθμου μπορούν να διαφέρουν και να εξαρτώνται από το πως ο αλγόριθμος εφαρμόζεται. Επίσης, όταν η κατάτμηση γίνεται, η πληροφορία από το προηγούμενο βήμα χάνεται [17], ως εκ τούτου, η λύση μπορεί να μην είναι σε θέση να φθάσει τοπικό ελάχιστο. Επίσης ο min-cut τοποθετητής μπορεί να μην είναι

σε θέση να φθάσει σε ολικό ελάχιστο, λόγω μερικών από των άπληστων στρατηγικών (greedy strategies) που χρησιμοποιεί.

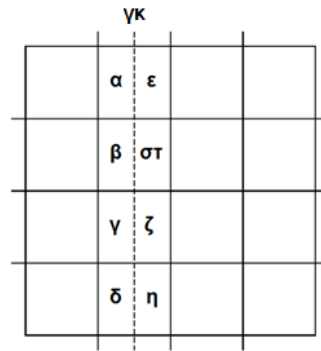
5.4 Βελτίωση Αλγορίθμου

Όπως είδαμε και παραπάνω, η ποιότητα των αποτελεσμάτων του αλγορίθμου min-cut, εξαρτάται πάρα πολύ από το πώς εφαρμόζεται ο αλγόριθμος. Όπως περιγράφηκε νωρίτερα στον παραδοσιακό min-cut αλγόριθμο η αλληλουχία της κατάτμησης γίνεται όπως φαίνεται και στο σχήμα παρακάτω. Με διακεκομμένη γραμμή φαίνονται οι γραμμές κατάτμησης που εφαρμόζονται σε εκείνο το στάδιο και οι συμπαγείς γραμμές δείχνουν της γραμμές κατάτμησης που έχουν εφαρμοστεί σε προηγούμενο στάδιο.



Εικόνα 23 Επιλογή της γραμμής κατάτμησης στο κέντρο της τρέχουσας περιοχής

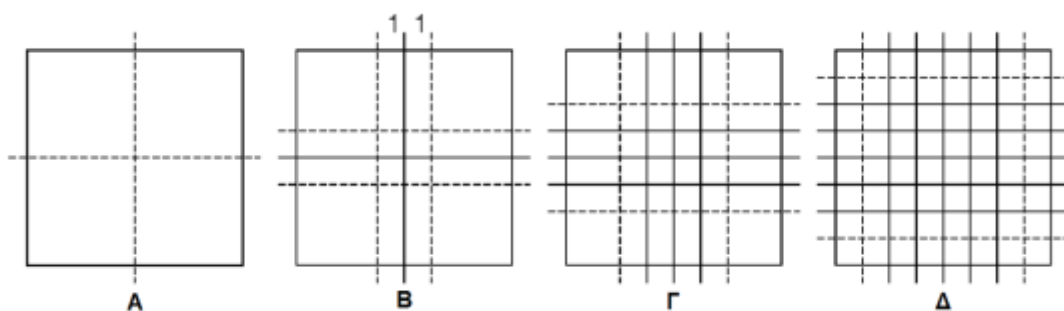
Εάν υποθέσουμε ότι η τρέχουσα γραμμή κατάτμησης γκ, φαίνεται ως διακεκομμένη γραμμή στο παρακάτω σχήμα, είναι ακριβώς δίπλα από την κεντρική γραμμή κατάτμησης. Η γραμμή κατάτμησης γκ θα κόψει σε τέσσερις περιοχές και τα αντίστοιχα υποδίκτυα τους στην μέση. Σε κάθε βήμα του min-cut αλγορίθμου μπορούμε να ανταλλάξουμε ένα ζεύγος κόμβων όπως ο ένας να είναι στην περιοχή α και ο άλλος στην περιοχή ε, ένας στην περιοχή β και ο άλλος στην περιοχή στ και ούτω καθεξής. Ωστόσο, προκειμένου να ληφθούν υπόψη τα αποτελέσματα της κατάτμησης νωρίτερα, δεν επιτρέπεται καμία ανταλλαγή κόμβων από περιοχές που έχουν κατατμηθεί σε προηγούμενα στάδια.



Εικόνα 24 Παράδειγμα παραδοσιακού min-cut τοποθετητή

Ο αριθμός των κόμβων σε μια περιοχή είναι ανάλογος ως προς την έκταση της εν λόγω περιοχής. Δεδομένου ότι η γραμμή τεμαχισμού γκ είναι κοντά στην κεντρική γραμμή τεμαχισμού, οι τέσσερις περιοχές που κόβονται από την γκ είναι μικρές. Ως εκ τούτου, ο αριθμός των κόμβων που τοποθετούνται σε περιοχές α-η είναι μικρός. Κατά συνέπεια, ο αριθμός των πιθανών ζευγών που μπορούμε να επιλέξουμε από μια κίνηση σε αυτή τη διαδικασία είναι περιορισμένος. Αυτό συνήθως οδηγεί σε μια σχετικά μεγάλη τομή μεγέθους για τις γραμμές τεμαχισμού κοντά στο κέντρο, λόγω ενός μικρού αριθμού πιθανών κινήσεων [2].

Με βάση αυτή την παρατήρηση, προτείνεται μια άλλη ακολουθία κατάτμησης γραμμών [3], η οποία εφαρμόζεται για τη μείωση της κυκλοφοριακής συμφόρησης κοντά στο κέντρο. Οι οριζόντιες γραμμές κατάτμησης και κάθετες γραμμές κατάτμησης εφαρμόζονται όπως και πριν. Ωστόσο, για τις γραμμές του κατάτμησης του ίδιου προσανατολισμού, οι γραμμές κατάτμησης που είναι πιο κοντά στο κέντρο εφαρμόζονται νωρίτερα, όπως φαίνεται στο παρακάτω σχήμα.



Εικόνα 25 Προτεινόμενη αλληλουχία για εφαρμογή γραμμών κατάτμησης

Από τα εργαστηριακά αποτελέσματα της έρευνας [3], βλέπουμε ότι η προταθείσα αλληλουχία, σε σχέση με την παραδοσιακή αλληλουχία του αλγορίθμου min-cut, μειώνει το μέγιστο αριθμό κατατμήσεων και την συνολική καλωδίωση του κυκλώματος.

Genetic Algorithm with simulated annealing (Γενετικός αλγόριθμος με προσομοίωση απόψησης)

6.1 Γενικά Στοιχεία

Σε αυτή την ενότητα εξετάζεται μία προσέγγιση υβριδικού αλγόριθμου τοποθέτησης σε FPGA. Αποτελεί μια μικτή εκδοχή των γενετικών αλγορίθμων και της προσομοιωμένης απόψησης τεχνικής βελτιστοποίησης που χρησιμοποιείται για την τοποθέτηση των συμμετρικών FPGA. Αυτό ο αλγόριθμος αποτελείται από δύο στάδια. Πρώτον, χρησιμοποιεί γενετικό αλγόριθμο για τη βελτιστοποίηση της τοποθέτησης σε ολικό επίπεδο, και ο προσομοιωμένος Αλγόριθμος απόψησης χρησιμοποιείται για να βελτιωθεί η λύση τοπικά [39]. Αυτή η υβριδική προσέγγιση τοποθέτησης προτείνεται να χρησιμοποιηθεί γιατί περικλύει το πλεονέκτημα εξεύρεσης ολικής καλύτερης λύσης των γενετικών αλγορίθμων και να ξεπεράσουν την αργή σύγκλιση των Γενετικών αλγορίθμων στην τελική φάση της διαδικασίας, με τη χρήση του αλγορίθμου προσομοιωμένης απόψησης χαμηλής θερμοκρασίας [39].

6.2 Επισκόπηση του Gasa

Όπως παρουσιάστηκε σε προηγούμενο κεφάλαιο, που ο αλγόριθμος προσομοιωμένης απόψησης περιγράφηκε λεπτομερώς για την τοποθέτηση σε συμμετρικού FPGA. Ο Γενετικός αλγόριθμος έχει πλεονέκτημα της ολικής αναζήτησης σε σχέση με την προσομοιωμένη απόψηση, λόγω της ευελιξίας του να αναζητά και να επιλύει ανεξάρτητα οποιοδήποτε πρόβλημα παρουσιάζεται. Ωστόσο, χρειάζεται αρκετό χρόνο ώστε να συγκλίνουν στην όψιμη φάση της διαδικασίας.

6.3 Βελτιστοποίηση με Gasa

Στην αρχή του αλγορίθμου, ο γενετικός αλγόριθμος (GA) λειτουργεί με τη χρήση επιλογής, διασταύρωσης και μετάλλαξης των φορέων. Ξοδεύεται ένας σημαντικός χρόνος στην όψιμη φάση της διαδικασίας του γενετικού αλγορίθμου (GA) κατά την οποία λαμβάνονται μικρές βελτιώσεις και πολύ αργά. Η Προσομοιωμένη απόψηση (SA) είναι ικανή να επιτύχει

βελτιώσεις γρηγορότερα από το γενετικό αλγόριθμο (GA) στην όψιμη φάση της διαδικασίας. Ως εκ τούτου, μετά από ένα ορισμένο αριθμό γενεών η προσομοιωμένη ανόπτηση χρησιμοποιείται με σκοπό τη βελτιστοποίηση του συνολικού πληθυσμού σε χαμηλή θερμοκρασία [4].

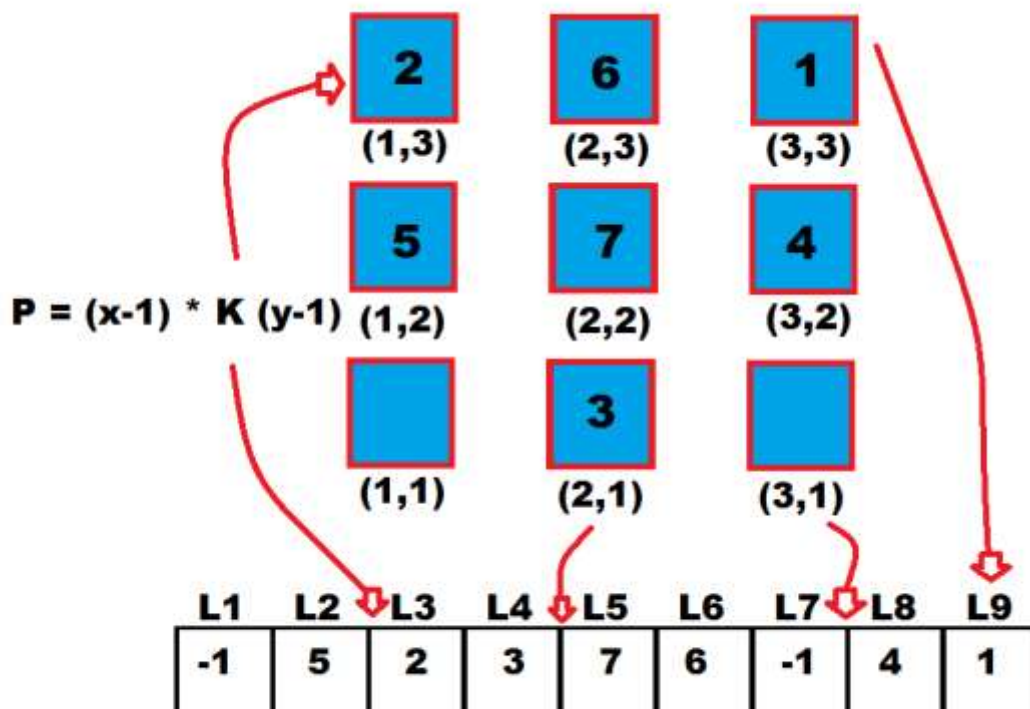
6.4 Γενετικός Αλγόριθμος

Ο Γενετικός αλγόριθμος κατασκευάζεται ως εξής:

- Ο γονότυπος είναι η αναπαράσταση ενός ατόμου στο γενετικό αλγόριθμο.
- Η δομή του χρωμοσώματος είναι κατασκευασμένη ως $L_1, L_2, L_3 \dots L_N$, όπου N εξαρτάται από το K , το μέγεθος ενός FPGA. Κάθε δομή των χρωμοσωμάτων L_i , όπου το i κυμαίνεται από 1 έως N , και αποδίδεται είτε με:
 - i. Ένα θετικό ακέραιο αριθμός: 0 έως M (όπου M είναι ο αριθμός των CLB μπλοκ στο κύκλωμα)
 - ii. -1: Ένα άδειο μπλοκ

Το γονίδιο CLB, έχει μια αντίστοιχη θέση στο χρωμόσωμα. Η θέση του P υπολογίζεται ως:

$$P = (x - 1) \times K(y - 1)$$



Εικόνα 26 Παράδειγμα τοποθέτησης σε FPGA και του αντίστοιχου χρωμοσώματος

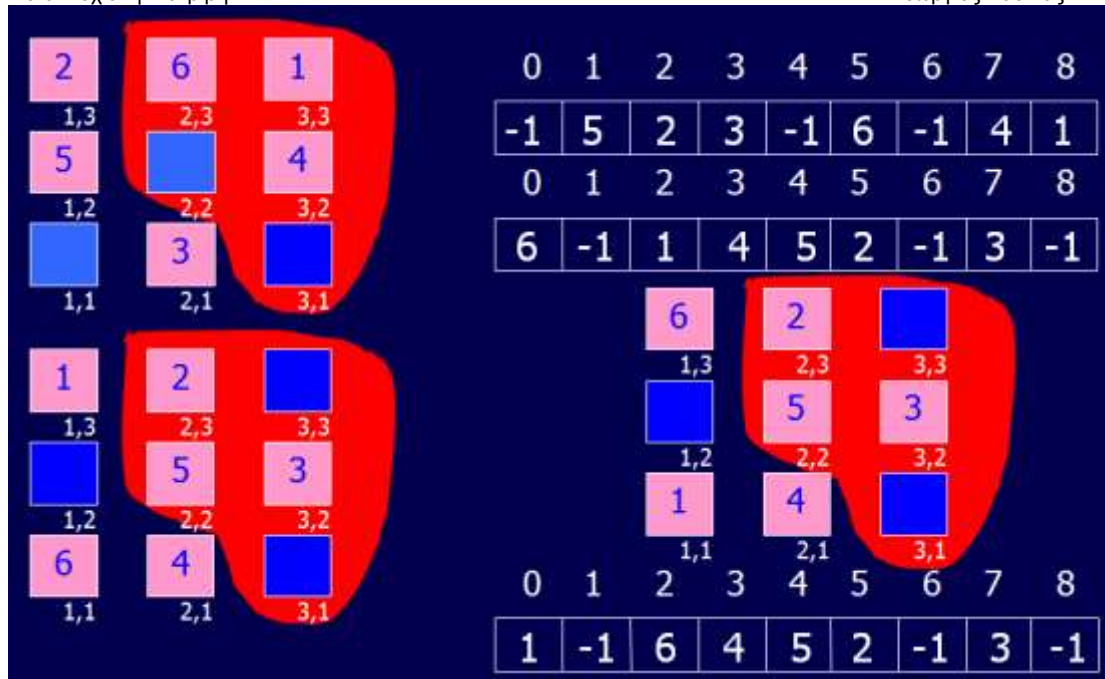
Η συνάρτηση καταλληλότητας χρησιμοποιείται για τη μέτρηση της ποιότητας της τοποθέτησης. Η συνάρτηση καταλληλότητας παρουσιάζεται είναι ως εξής

$$f = \max cost - \sum_{i=1}^{\dots} C(i)[bb_x(i) + bb_y(i)]$$

$$\max cost = no_of_nets \times K \times K$$

Όπου **max cost** παρουσιάζει τη χειρότερη περίπτωση του κόστους για την τοποθέτηση. Το **N** αντιπροσωπεύει τον συνολικό αριθμό των συνδεσμολογιών του κυκλώματος (netlist), το **no_of_nets** αντιπροσωπεύει τον αριθμό των δικτύων για κάθε κύκλωμα και το **K** είναι το μέγεθος του FPGA . Κάθε μία από τις παραμέτρους $bb_x(i)$ και το $bb_y(i)$ είναι οριζόντια και κάθετη έκταση του μοντέλου του πλαισίου οριοθέτησης για μήκος καλωδίωσης. Έχει παρατηρηθεί ότι η συνάρτηση κόστους είναι παρόμοια με τη συνάρτηση κόστους για την τεχνική βελτιστοποίηση της προσομοιωμένης ανόπτησης. Σε κάθε κίνηση, η συνάρτηση καταλληλότητας επαναυπολογίζεται για τον προσδιορισμό της ποιότητας της τοποθέτησης. Όσο υψηλότερο είναι το κόστος, τόσο καλύτερη είναι η τοποθέτηση. Το χαμηλότερο κόστος, μπορεί να επιδεινώσει την τοποθέτηση [39].

- **Επιλογή:** Τα άτομα που επιλέγονται ανάλογα με την αξία καταλληλότητά τους. Οι τιμές καταλληλότητας του πληθυσμού είναι ταξινομημένες κατά αύξουσα σειρά. Ένας μικρός αριθμός των ατόμων του πληθυσμού με την υψηλότερη αξία καταλληλότητας στην τρέχουσα γενιά παραμένουν στο πληθυσμό. Τα N άτομα έχουν επιλεγθεί ταυτόχρονα. Η διαδικασία επιλογής είναι τυχαία, αλλά το πιο ευέλικτο άτομο είναι πιο πιθανό να επιλεγθεί
- **Διασταύρωση :** Η διασταύρωση αποτελεί μια διαδικασία που συνδυάζει τις τιμές λογικού στοιχείου από ένα επιλεγμένο άτομο με το άλλο.
- **Μετάλλαξη:** είναι η διαδικασία μετάλλαξης ανταλλαγών των τιμών όποιον δύο λογικών στοιχείων σε μία επιλεγμένη ατομική τοποθέτηση [36].



Εικόνα 27 Διαδικασία διασταύρωσης

6.5 Τοπική βελτιστοποίηση - προσομοιωμένη ανόπτηση

Μόλις ο γενετικός αλγόριθμος έχει ολοκληρώσει την ολική αναζήτηση στο πρώτο στάδιο, η προσομοιωμένη ανόπτηση εκτελεί την τοπική αναζήτηση. Εάν η βελτίωση αυτή δεν αποκτήθηκε στο γενετικό αλγόριθμο για 5 γενιές ή ο αριθμός της γενιάς είναι μεγαλύτερος από το μέγιστο αριθμό των γενεών, τότε η προσομοίωση ανόπτησης θα αρχίσει να εργάζεται μεμονωμένα αντί ολόκληρου του πληθυσμού.

Η αρχική θερμοκρασία T για την προσομοιωμένη μέθοδο ανόπτησης επιλέγεται σε 1 βαθμό. Παρόμοια με την προσομοίωση διαδικασία ανόπτησης που περιγράφεται σε προηγούμενο κεφάλαιο, η νέα θερμοκρασία υπολογίζεται ως $T_{new} = \beta$ όπου β εξαρτάται α και είναι το ποσοστό των κινήσεων μεταξύ δύο ανταλλάσσόμενων μπλοκ που έχουν γίνει αποδεκτά. (G. Nam, 1999)

Πίνακας 7 Επισκόπηση γενετικού αλγόριθμου με προσομοιωμένη ανόπτηση

```

01 begin
02   initialize_population ();
03   while (generation < MAX_GENS) do
04     evaluate_population_fitness ();
05     reproduce_population (Preserve);

```



```
06   for i = 1 to POP_SIZE/2 do
07       crossover (Pcrossover);
08   for j=1 to NUM_GENES do
09       mutate(Pmutation);
10   for i = 1 to POP_SIZE do
11       local_improvement(Plocal);
12   elitism();
13   end while
14   select_the_best_one();
15   T = set_temperature();
16   R = set_block_movement_range();
17 /* following algorithm is pseudo-code of SA*/
18   while (Exit_criterion() == FALSE) do
19       while (inner_criterion() == FALSE) do
20           Pnew = generate_movement (R, Pold)
21           ΔC = C (Pnew) - C (Pold);
22           RANDOM = generate_number();
23           if (RANDOM < e exp (-ΔC/T))
24               Pold = Pnew;
25       end while
26   end while
27 end algorithm
```

6.6 Πλεονεκτήματα και μειονεκτήματα

Ο γενετικός αλγόριθμος έχει λειτουργία ανοικτού κώδικα, με αποτέλεσμα να μπορεί να βελτιώνεται από πολλούς και ανεξάρτητους φορείς. Ωστόσο, αυτή η τεχνική βελτιστοποίησης είναι πολύπλοκη και μπορεί να είναι δύσκολη να εφαρμοστεί.

Συμπεράσματα

Ως αποτέλεσμα των ως άνω προκύπτουν κάποια συμπερασματικά στοιχεία σχετικά με την αποτελεσματικότητα των αλγόριθμων.

Πίνακας 8 Επισκόπηση αλγορίθμων

	Ποιότητα Τοποθέτησης	Αποδοτικότητα Τοποθέτησης
Προσομοιωμένη Ανόπτηση	Συνολικά δίνει τα καλύτερα αποτελέσματα, όταν χρησιμοποιείται το μήκος καλωδίωσης ως συνάρτηση κόστους. Η τελική τοποθέτηση μπορεί να φθάσει ολική βελτιστοποίηση	Είναι αρκετά αργός εξαιτίας της υπολογιστικά ακριβής αξιολόγησης κάθε νέας κίνησης
Τετραγωνική Τοποθέτηση	Κατά μέσο όρο, απαιτεί 1,9% περισσότερο μήκος καλωδίωσης σε σχέση με τη μέθοδο προσομοιωμένης ανόπτησης. Θεωρεί μόνο παράγοντα το μήκος της καλωδίωσης στη συνάρτηση κόστους, ενώ το κομμάτι του χρονισμού δεν μπορεί να εμφανιστεί.	Συγκριτικά με την προσομοιωμένη ανόπτηση μπορεί να είναι έως και 5,8 φορές γρηγορότερη κατά μέσο όρο
Min-Cut Τοποθέτηση	Το τελικό αποτέλεσμα παρουσιάζει μεγάλη μεταβλητότητα ανάλογα με το πώς έγινε η κατάτμηση. Από όσο γνωρίζουμε, καμία ευθεία σύγκριση με την προσομοιωμένη ανόπτηση δεν έχει γίνει	Κατά μέσο όρο 3-4 φορές επιτάχυνση μπορεί να επιτευχθεί σε σχέση με την προσομοιωμένη ανόπτηση
Βελτίωση Min-Cut Τοποθέτησης	Παρουσιάζεται εργαστηριακά λιγότερη συνολική καλωδίωση, σε σχέση με τον παραδοσιακό min-cut αλγόριθμο	Υπάρχει μία επιτάχυνση, αφού ο αριθμών των κατατμήσεων μειώνεται μέχρι και σε 43,3%
Παράλληλη Τοποθέτηση	Σε γενικές γραμμές, χρησιμοποιώντας την ντετερμινιστική παράλληλη τοποθέτηση, τα αποτελέσματα είναι τόσο καλά όσο με την προσομοιωμένη ανόπτηση	Η καλύτερη επιτάχυνση μπορεί να είναι γραμμική με τον αριθμό των επεξεργαστών, αν ο χρόνος οριστικοποίησης είναι αμελητέος
Βελτίωση προσομοιωμένης ανόπτησης	Στα περισσότερα τεστ δεν υπάρχει σημαντική απώλεια στην ποιότητα, ενώ σε κάποιες περιπτώσεις υπάρχει και βελτίωση	Ο χρόνος διεργασίας μειώνεται έως και 56% σε σχέση με την προσομοιωμένη ανόπτηση

		(VPR)
Γενετικός Αλγόριθμος με προσομοιωμένη ανόπτηση	Τα ίδια με την προσομοιωμένη ανόπτηση	Πιο αποδεκτός χρόνος υπολογισμού σε σχέση με την προσομοιωμένη ανόπτηση

Η τοποθέτηση με **προσομοιωμένη ανόπτηση** σε γενικές γραμμές, υπερτερεί από τους άλλους τοποθετητές σε σχέση με τα τελικά αποτελέσματα. Εκτός αυτού, μπορεί να ξεπεράσει το τοπικό ελάχιστο και να έχει ανοιχτή συνάρτηση κόστους. Ωστόσο, είναι πολύ χρονοβόρα.

Η **τετραγωνική τοποθέτηση** δίνει γρήγορο χρόνο ολοκλήρωσης, ενώ τα αποτελέσματα δεν μπορούν να φτάσουν ολική βελτιστοποίηση λόγω της τετραγωνικής φύσης της συνάρτησης κόστους και ο χρονισμός δεν μπορεί να εμφανίζεται στην συνάρτηση κόστους.

Η **Min-cut μέθοδος** βασισμένη στην κατάτμηση μπορεί επίσης να επιταχύνει το χρόνο ολοκλήρωσης, ενώ η ποιότητα της τοποθέτησης ακόμα δεν μπορεί να είναι εγγυημένη.

Ο **Παράλληλος Αλγόριθμος** μπορεί να δώσει καλές επιταχύνσεις και να μην χάσει σχεδόν σε καμία ποιότητα. Αλλά η επεκτασιμότητα είναι περιορισμένη από την επιβάρυνση που οφείλεται στη μνήμη.

Ο **γενετικός αλγόριθμος με προσομοιωμένη ανόπτηση** είναι σε θέση να επιτύχει την ολική βέλτιστη λύση με αποδεκτό χρόνο υπολογισμού

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **Aggarwal R. Kumar V., Karypis G. and Shekhar S.** Multilevel hypergraph partitioning: Application in vlsi domain. Proc. ACM/IEEE DAC, 1997.
2. **Banerjee, P.**, “Accelerators for fpga placement,” in The 4th Annual Inter Research Institute Student Seminar in Computer Science”, April 2005.
3. **Baruch Z.,Cret O.,Pustztai K.**,”An efficient sequence to apply slicing line in FPGA placement”
4. **Baruch, Z.:** “Contributions to the Computer Aided Design of Digital Systems”, Ph.D. Thesis, Computer Science Department, Technical University of Cluj-Napoca.(1998)
5. **Betz V. and Rose J.V** VPR: A New Packing, Placement and Routing Tool for FPGA Research [Book]. - Toronto : International Workshop on Field Programmable Logic and Applications, 1997.
6. **Betz V. Rose J., Marquardt A. ,** Architecture and Cad for Deep-Submicron FPGAs [Book]. - Toronto : Kluwer Academic Publishers, 2000.
7. **Brayton R., and Mishchenko A.**, “Abc: an academic industrial-strength verification tool,” in Proceedings of the 22nd international conference on Computer Aided Verification, ser. CAV’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 24–40.
8. **Brown S.** An overview of technology, architecture and CAD tools for programmable logic devices [Book]. - [s.l.] : CICC, 1994.
9. **Chandy J. Kim S. Rankumar B., Parkers S.and Banerjee P.**. An evaluation of parallel simulated annealing strategies with application to standard cell placement. TCAD, 16:398–410, 1997.
10. **Clarke P.** Dynachip claims speed breakthrough in its FPGAs [Book]. - [s.l.] : Electronic Engineering Times, 1997.
11. **Dunlop AE, Kernighan BW.** A procedure for placement of standard cell VLSI circuits. IEEE Trans Comput Aided Des. 1985;4(1):92–98.

12. **Edinaldo Mariano Vasconcelos de Lima, Dr. Antonio Carlos Cavalcanti, Dr. Lucidio dos Anjos Formiga Cabral**, A New Approach to VPR Tool's FPGA Placement. Proceedings of the World Congress on Engineering and Computer Science 2007. WCECS 2007, October 24-26, 2007, San Francisco, USA
13. **Feist T.**. Vivado Design Suite. Xilinx WP416 (v1.1) June 22, 2012
14. **Hassan H., Anis M.**, Low-power Design of nanometer FPGAs [Book]. - Morgan Kaufmann, 2010.
15. **Huang DJ-H, Kahng AB**. Partitioning-based standard-cell global placement with an exact objective. In: Proceedings of the ACM International Symposium on Physical Design. 1997:18–25
16. **Jamieson P., Kent K., Gharibian F., and Shannon L.**. Odin II-An Open-Source Verilog HDL Synthesis Tool for CAD Research. In IEEE Annual Int'l Symp. on Field-Programmable Custom Computing Machines, pages 149–156. IEEE, 2010.
17. **Kim C., Shin H.**, "A performance-driven logic emulation system: FPGA network design and performance-driven partitioning," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 15, Issue5, pp.560-568, May 1996.
18. **Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P.**, Optimization by Simulated Annealing, Science, No. 220, pp. 671-680. 1983.
19. **Kleinhans JM, Sigl G, Johannes FM, Antreich KJ**. GORDIAN: VLSI placement by quadratic programming and slicing optimization. IEEE Trans Comput Aided Des. 1991;10(3):356–365
20. **Lavin C., Padilla M., Kanorecht J., Lundrigan P., Nelson B., Hutchings B., Wirthlin M.**, RapidSmith, A Library for Low-level Manipulation of Partially Placed-and-Routed FPGA Designs, Technical Report and Documentation, NSF Center for High Performance Reconfigurable Computing (CHREC) Department of Electrical and Computer Engineering Brigham Young University 2014 (Revised).
21. **Lee S.** FPGA Placement Optimization Methodology Survey Book]. - TORONTO : Department of Electrical and Computer Engineering, 2010.
22. **Ludwin A. Betz V. and Padalia K.**. High-quality, deterministic parallel placement for fpgas on commodity hardware. ACM/Sigda Int. Symp. on FPGAs, pages 14–23, 2008.

23. **Maidee P. Ababei C., and Bazargan K.** , Time-Driven Partitioning-Based Placement for Island Style FPGAs [Book]. - [s.l.] : IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1996.
24. **Mathur A. and Liu C.L.**, “Compression-Relaxation: A New Approach to Timing-Driven Placement for Regular Architectures”, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on Volume 16, Issue 6, pp. 597-608, June 1997.
25. **Mulpuri C. and Hauck S.C** Runtime and quality tradeoffs in FPGA placement and routing [Book]. - [s.l.] : In Proc. ACM International Symp. FPGAs, 2001.
26. **Murray K., Petelin O., Luu J., Betz V., Anderson J., Rose J.** VPR User’s Manual (VPR 7.0)
27. **Nam G. Sakallah K. and Rutenbar R.** , Satisfiability-Based Layout Revisited: Detailed Routing of Complex FPGAs Via Search-Based Boolean SAT [Book]. - [s.l.] : ACM/SIGDA International Symp on FPGA, 1999. - Vols. pp. 167-175.
28. **Nayak A., Choudhary A., Haldar M. and Banerjee P.** Parallel algorithms for FPGA Placement. Proc. of the 10th Great Lakes Symposium on VLSI, pages 86–94,2000.
29. **Psarakis M.**, “Προηγμένη Ψηφιακή Σχεδίαση», Σημειώσεις, (2012).
30. **Rose J., Luu J., Yu C.W., Densmore O., Goeders J., Somerville A., Kent K.B., Jamieson P., Anderson J.**, “The vtr project: architecture and cad for fpgas from verilog to routing,” in Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays ser. FPGA ’12. New York, NY, USA: ACM, 2012, pp. 77–86.
31. **Shi X.** FPGA Placement Methodologies: A Survey [Book]. - EDMONTON : Department of Computing Science,, 2010.
32. **Srinivasan A, Chaudhary K, Kuh ES.** Ritual: a performance driven placement algorithm for small cell ICs. In: Proceedings of the International Conference on Computer Aided Design. 1991:48–51.
33. **Steiner N., Wood A., Shojaei H., Couch J., Athanas P., and French M.**, Torc: Towards an Open-Source Tool Flow, in Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays.

34. **Tessier R.** Fast Place and Route Approaches for FPGAs, [Book]. - [s.l.] :
Massachusetts Institute of Technology PhD thesis, 1999.
35. **Wawrzynek J.** Should the academic community launch an open-source FPGA device and tools effort?: evening panel Proceeding FPGA '11 Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays Pages 3-4
36. **Wing On Fung, Arslan, T., Khawam, S.**, "Genetic Algorithm based Engine for Domain-Specific Reconfigurable Arrays," First NASA/ESA Conference on Adaptive Hardware and Systems, pp. 200-206, June 2006.
37. **White, Brad S.**, "Tincr: Integrating Custom CAD Tool Frameworks with the Xilinx Vivado Design Suite" (2014)
38. **Xu Y. and Khalid.M.A.S.** , Efficient quadratic placement for fpgas. [Book]. - [s.l.] :
International Conference on Field Programmable Logic and Application, 2005.
39. **Yang M., Almaini A.E.A., Wang L., Wang P.**, "FPGA Placement using genetic algorithm with simulated annealing," ASICON 2005 6th International Conf, volume 2, pp.213-222, August 1997.
40. Quartus II University Interface Program. (2015, October 05). Retrieved from <http://wl.altera.com/education/univ/research/quip/unv-quip.html>
41. Tools for Open Reconfigurable Computing (TORC) (2015, October 05). Retrieved from <http://torc-isi.sourceforge.net/index.php>
42. The Verilog-to-Routing (VTR) Project for FPGAs. (2015, October 05). Retrieved from <https://github.com/verilog-to-routing/vtr-verilog-to-routing/wiki>