

Konzeption und Evaluierung eines innovativen Platzierungs- und Verdrahtungsverfahrens

Diplomarbeit

von

Ole Blaurock

November 2001

Betreuer:

Dr. habil. R. Rauscher

Prof. Dr. R. Lang

Universität Hamburg

Fachbereich Informatik

Arbeitsbereich Technische Grundlagen der Informatik (TECH)

Inhalt

1. EINLEITUNG UND MOTIVATION	I
2. PROBLEMANALYSE	3
2.1 Physikalischer VLSI-Entwurf “heute”	3
2.2 State Of The Art	6
2.3 Technologische Entwicklung	7
2.4 Verdrahten vor dem Plazieren	8
2.5 Annahmen, Randbedingungen	10
2.6 Zielsetzung	11
3. DIE EVALUATIONSUMGEBUNG: ESPRO	13
3.1 Modell und Entwurfsstil von ESPRO	13
3.1.1 Der Entwurfsstil in ESPRO	13
3.1.2 Die Zellbibliothek in ESPRO	14
3.1.3 Modell für die Verdrahtung	15
3.2 Aufbau des Systems	15
3.2.1 Systemüberblick und Modulbeschreibungen	16
3.2.2 Verwendete Modulschnittstellen	18
3.3 Änderungen der Aufgabe und Randbedingungen durch ESPRO	18
4. DIE ALGORITHMISCHE UMSETZUNG	21
4.1 Modellierung und Überblick	21
4.1.1 Die Netzliste als Hypergraph	21
4.1.2 Überblick über das in dieser Arbeit verwendete Vorgehen	29
4.2 Details zu Teilproblemen von der Netzliste bis zur fertigen Plazierung	34
4.2.1 Positionierung der Netze	35
4.2.1.1 Der verwendete Plazierungsalgorithmus	37
4.2.1.2 Die Bewertung der Bipartitionen	38
4.2.1.3 Die für die Bipartitionen verwendete Heuristik	39
4.2.1.4 Ein Beispiel: Die Netzliste T4	40
4.2.2 Einfügen der Zellen	42
4.2.2.1 Konsequenzen der Repräsentation der Zellen als Hyperkanten	42
4.2.2.2 Approximation der Zellpositionen	46
4.2.2.3 Ein Beispiel: Die Netzliste T4	46
4.2.3 Verbesserung der Zellposition durch kräftebasiertes Plazieren	47
4.2.3.1 Iterative Verbesserungen	47
4.2.3.2 Die Abbruchbedingung	49
4.2.3.3 Ein Beispiel: Die Netzliste T4	49
4.2.4 Anpassen der Plazierung an den Standardzellentwurf	51
4.2.4.1 Erzeugen der Zellreihen und Abschätzen der Chipgröße	51
4.2.4.2 Abschätzen von Anzahl und Lage der benötigten Feedthroughs	55
4.2.4.3 Plazieren der Padzellen	56
4.2.4.4 Ein Beispiel: Die Netzliste T4	58
4.3 Details zu Teilproblemen von der Plazierung zum fertigen Layout	61
4.3.1 Globalverdrahtung	61

Inhalt

4.3.1.1	Auswahl von Start- und Zielpin des Zweipunktnetzes	63
4.3.1.2	Verschieben von Start- und / oder Zielzelle	63
4.3.1.3	Erstellen des Routinggraphen	65
4.3.1.4	Auswahl des verwendeten Pfades	67
4.3.1.5	Verschieben der Zielzelle und Einfügen der Feedthroughzellen	68
4.3.1.6	Ein Beispiel: Die Netzliste T4	69
4.3.2	Überprüfung des Netzzusammenhangs	71
4.3.3	Globalverdrahtung der Padzellen	72
4.3.4	Die Kanalverdrahtung	73
4.3.5	Abschluß des Layouts	74
5.	EVALUATIONSVERFAHREN	77
5.1	Implementierung eines Prototypen: das Programm „prob“	77
5.2	Eigenschaften der verwendeten Evaluierungsdaten	77
5.3	Durchgeführte Untersuchungen	78
5.3.1	Untersuchung des Verdrahters	79
5.3.1.1	Bestimmung der Pfadbewertungsparameter	79
5.3.1.2	Bestimmung der Parameter zur Plazierungsmodifikation	80
5.3.1.3	Vergleich mit dem dkrouter	80
5.3.2	Untersuchung des Plazierers	81
5.3.2.1	Parametrisierung der Standardzellplatzierung	81
5.3.2.2	Parametrisierung der hierarchischen Partitionierung	83
5.3.2.3	Parametrisierung der kräftegesteuerten Platzierung	83
5.3.2.4	Vergleich mit existierenden Plazierern	84
5.3.3	Untersuchung beider Komponenten zusammen	85
6.	ERGEBNISSE	87
6.1	Untersuchungsergebnisse für den Verdrahter	87
6.1.1	Ergebnisse der Verdrahterparametrisierung	87
6.1.1.1	Ermittlung der Pfadbewertungsparameter anhand von test500	87
6.1.1.2	Parametrisierung der Plazierungsmodifikation anhand von test500	88
6.1.1.3	Ermittlung der Pfadbewertungsparameter anhand von pk690	89
6.1.1.4	Parametrisierung der Plazierungsmodifikation anhand von pk690	91
6.1.2	Vergleich der Verdrahtungsergebnisse für verschiedene Platzierungen	92
6.2	Untersuchungsergebnisse für den Plazierer	97
6.2.1	Ergebnisse zur Parametrisierung der Standardzellplatzierungserstellung	98
6.2.1.1	Ergebnisse der Parameterevaluation anhand der Netzliste test500	98
6.2.1.2	Ergebnisse der Parameterevaluation anhand der Netzliste pk690	99
6.2.2	Ergebnisse zur Parametrisierung der hierarchischen Partitionierung	100
6.2.2.1	Ergebnisse der Parameterevaluation anhand der Netzliste test500	100
6.2.2.2	Ergebnisse der Parameterevaluation anhand der Netzliste pk690	101
6.2.3	Ergebnisse zur Parametrisierung der kräftegesteuerten Platzierung	102
6.2.3.1	Ergebnisse der Parameterevaluation anhand der Netzliste test500	102
6.2.3.2	Ergebnisse der Parameterevaluation anhand der Netzliste pk690	103
6.2.4	Vergleich mit anderen ESPRO-Plazierern	104
6.3	Untersuchungsergebnisse für Plazierer und Verdrahter gemeinsam	108
6.3.1	Vergleich der Parametrisierungen anhand der Netzliste test500	108
6.3.2	Vergleich der Parametrisierungen anhand der Netzliste pk690	108
6.3.3	Vergleich mit existierenden Plazierern und Verdrahtern	109
6.4	Zusammenfassende Diskussion der Ergebnisse	111
7.	AUSBLICK	115
7.1	Weitere Evaluationsmöglichkeiten innerhalb des bestehenden Systems	115
7.2	Alternative Algorithmen für Teilschritte des Verfahrens	116

Inhalt

7.2.1	Anpassen der Datenstrukturen	116
7.2.2	Andere Platzierung der Netze	117
7.2.2.1	Änderungen an der hierarchischen Partitionierung	117
7.2.2.2	Änderung des Bipartitionierungsalgorithmus	118
7.2.2.3	Methode des Zelleneinfügens und die kräftegesteuerte Platzierung	119
7.2.2.4	Standardzellplatzierung erstellen	119
7.2.3	Globalverdrahtung und Platzierungsmodifikationen	121
7.2.3.1	Zellweise sequentielle Verdrahtungsreihenfolge	121
7.2.3.2	Zweipunktnetzbetrachtung	122
7.2.3.3	Platzierungsmodifikation durch den Verdrahter	122
7.2.4	Kompaktifizierung und Pad-Behandlung	123
7.2.5	Kanalverdrahtung	124
7.3	Änderung des Modells	124
7.3.1	Erweiterung von ESPRO	124
7.3.1.1	Verkleinerung der Zellen	125
7.3.1.2	Feedthroughs als Zellen	125
7.3.1.3	Zusätzliche Erweiterungsmöglichkeiten für ESPRO	125
7.3.2	Modellierung außerhalb von ESPRO	126
7.4	Projektion des Konzeptes in die Zukunft	127
8.	ZUSAMMENFASSUNG	129
A.	VERWENDUNG DES PROGRAMMES PROB	131
A.1	Bedienung des Programmes	131
A.1.1	Allgemeiner Programmablauf	131
A.1.2	Programmaufruf	131
A.1.3	Erläuterung der Kommandozeilenparameter	132
A.2	Algorithmusparameter	133
A.2.1	Die Parameter der Teilalgorithmen	133
A.2.1.1	Eigenschaften der Zellbibliothek	134
A.2.1.2	Hierarchische Netzplatzierung	134
A.2.1.3	Kräftebasierte Platzierung der Zellen	135
A.2.1.4	Erzeugen der Standardzellreihen	136
A.2.1.5	Modifikationen des Verdrahters an der Platzierung	137
A.2.1.6	Globalverdrahtung und Beurteilung der platzierten Netze	139
A.2.1.7	Interne Speicherverwaltung	140
A.2.2	Formen der Parametereingabe	140
A.2.2.1	Interaktive Parametereingabe	140
A.2.2.2	Parameterdatei	140
A.3	Beispiel einer Parameterdatei	142
A.4	Eine Beispielsitzung im interaktiven Modus	143
A.5	Batch-Betrieb	147
A.6	Beispiel einer Batch- bzw. Logdatei	147
B.	PARAMETERWERTE DER EVALUIERUNG	149
C.	ERGEBNISSE DER PARAMETRISIERUNG	151
C.1	Parametrisierung des Verdrahters	151
C.2	Parametrisierung des Platzierers	162
C.2.1	Parametrisierung der Generierung einer Standardzellplatzierung	162
C.2.2	Parametrisierung der hierarchischen Partitionierung	167
C.2.3	Parametrisierung der kräftegesteuerten Platzierung	172

Inhalt

C.3	Parametrisierung der Kombination von Plazierer und Verdrahter	182
D.	PROB ERGEBNISSE OHNE KOMPAKTIFIZIERUNGSBUG	187
D.1	Verdrahterergebnisse	187
D.2	Ergebnisse des Gesamtverfahrens	189
E.	INHALT DER BEIGEFÜGTEN CD	191
F.	ANHANG: LITERATURVERZEICHNIS	193

I. EINLEITUNG UND MOTIVATION

Integrierte Schaltkreise (IC: Integrated Circuit) finden sich heute in fast jedem Gerät des täglichen Lebens. Sie sind nicht nur in Geräten der Telekommunikation und Informationsverarbeitung, wie Mobiltelefonen, elektronischen Terminplanern (PDA: Personal Digital Assistant) oder Personalcomputern, sondern auch in Waschmaschinen und Toastern allgegenwärtig. Als eine der bedeutendsten Errungenschaften der Menschheit haben sie in den 40 Jahren seit ihrer Entstehung unser Leben nachhaltig beeinflusst. Ebenso, wie die integrierten Schaltungen unseren Alltag verändert haben, hat sich die IC-Technologie selbst gewandelt [She95].

Die technologische Entwicklung führt zu immer kleineren Strukturen, was heute die Hochintegration vieler Funktionen auf einem Chip ermöglicht. Zusammen mit der Zunahme an Komplexität, der zu integrierenden Funktionalität, steigen auch die Anforderungen an die Entwerfer dieser integrierten Schaltungen. Die heutigen Anforderungen des VLSI-Entwurfes (VLSI: Very Large Scale Integration) lassen sich nur mit der Unterstützung von CAD-Werkzeugen bewältigen [Tri87]. Der Entwurf integrierter Schaltungen erfolgt dabei auf einer stark abstrahierenden Ebene, durch Verhaltens- oder Strukturbeschreibungen des Systems. Die Werkzeuge zur Logiksynthese und -minimierung ermöglichen es, eine rein algorithmische oder funktionale Verhaltensbeschreibung automatisch auf eine Strukturbeschreibung geringerer Abstraktion mit konkreten Basisbausteinen abzubilden. Die Fortschritte der Werkzeuge auf allen Ebenen der verschiedenen Abstraktionsniveaus verkürzen die benötigte Zeit zur Entwicklung einer funktionsfähigen Schaltung dramatisch. Dabei werden vermehrt Methoden aus dem Softwareentwurf verwendet [Hag95].

Auf niedrigeren Abstraktionsniveaus, als den bisher angesprochenen, sogenannten „höheren Entwurfsebenen“, ist der physikalische VLSI-Entwurf angesiedelt. Auf der Ebene des physikalischen Entwurfs resultiert die technologische Entwicklung der Produktionsprozesse und die steigende Komplexität der zu fertigenden integrierten Schaltungen in sich ändernden Anforderungen für die verwendeten Werkzeuge.

Der gesamte Entwurfsprozeß zerfällt in eine Reihe von Bearbeitungsschritten, die, mehr oder weniger getrennt voneinander, sequentiell bearbeitet werden. Die Abgrenzung der zu bewältigenden Teilprobleme ist nicht starr, sondern, ebenso wie die Teilprobleme selbst, einem Wandel unterworfen ([Len90], [She95]).

In dieser Arbeit sollen die höheren Entwurfsebenen nicht weiter betrachtet und der Fokus auf den physikalischen VLSI-Entwurf gelegt werden.

Die Schnittstelle zwischen dem physikalischen VLSI-Entwurf und den auf höheren Abstraktionsebenen angesiedelten Entwurfsschritten stellt die Netzliste dar. Eine Netzliste enthält eine vollständige Beschreibung der zu fertigenden Schaltung. In ihr sind alle Instanzen der verwendeten Bausteine einer zugrundeliegenden Bauteilebibliothek aufgeführt, sowie beschrieben, wie diese miteinander verbunden sind. Als Netz bezeichnet man die Leitungen, welche eine Menge von Anschlußpunkten (Pins) der Instanzen von Bibliotheksbausteinen (Zellen) elektrisch verbindet.

Aufgabe des physikalischen VLSI-Entwurfes ist es, die Spezifikation der Schaltung in eine geometrische Beschreibung, das Chiplayout, zu transferieren. Dabei wird ausgehend von

der Netzliste eine Anordnung der Zellen auf der verfügbaren Chipfläche ermittelt, sowie die elektrischen Verbindungen der enthaltenen Elemente untereinander vorgenommen.

Die meisten Aufgaben des physikalischen VLSI-Entwurfes fallen in die Klasse der NP-harten Probleme. Um die hohe Problemkomplexität handhaben zu können, werden meist heuristische Algorithmen verwendet. Dennoch ist es notwendig, den physikalischen VLSI-Entwurf in mehrere Teilprobleme aufzuteilen. Bei der Partitionierung wird die Netzliste in mehrere Teile von besser bearbeitbarer Größe zerlegt, welche dann separat weiter verarbeitet werden können. Die beiden Hauptschritte des physikalischen VLSI-Entwurfes bilden das Plazieren der Basiselemente (Placement) und das Erstellen der elektrischen Verbindungen, das Verdrahten (Routing). In anschließenden Schritten, wie der Kompaktifizierung (Compaction) und der Nachoptimierung (Post-Layout-Optimization) wird das gewonnene Layout weiter verbessert [She95].

Durch die Verringerung der Strukturgrößen der Produktionsprozesse, ändern sich die Anforderungen an die für Plazieren und Verdrahten verwendeten Algorithmen. Bei der Verdrahtung gewinnt z.B. die maximal zulässige Netzlänge und die Wechselwirkungen benachbarter Leitungssegmente immer mehr an Bedeutung. Die Größe der zu platzierenden Basiselemente nimmt dagegen immer mehr ab, so daß es beim Finden einer Platzierung mittlerweile weniger um das Unterbringen der Bausteine auf der Chipfläche, als um eine für die Verdrahtung günstige Anordnung dieser relativ zueinander geht. Bereits heute ist die von aktiven Zellen belegte Fläche deutlich kleiner, als die für die Verdrahtung verwendete. Die Bedeutung der Teilaufgabe des Verdrahtens steigt dabei mit kleiner werden Strukturen beständig (siehe Kapitel 2.3).

Diesen sich wandelnden Anforderungen wird durch eine stetige Weiterentwicklung der Algorithmen für Plazieren und Verdrahten begegnet. Die Konzentration der Entwicklungsaktivitäten liegt dabei auf Detailverbesserungen und Anpassungen an spezielle Gegebenheiten eines verwendeten Fertigungsprozesses. Trotz der steigenden Bedeutung der Verdrahtungsaufgabe bleibt die Qualität der Verdrahtung immer von der zuvor erstellten Platzierung abhängig (siehe Kapitel 2.2).

Um diese Benachteiligung des Verdrahtungsalgorithmus zu beseitigen, wäre es wünschenswert, die Reihenfolge von Plazieren und Verdrahten zu vertauschen. Eben dieser innovative Ansatz ist Grundlage des in dieser Arbeit entwickelten Verfahrens. Durch eine geeignete Modellierung, wird während des Plazierens und Verdrahtens nicht von den Zellen als den zu platzierenden Elementen ausgegangen, sondern die Netze in dieser Weise betrachtet. Das hier präsentierte Verfahren stellt die Netze in den Mittelpunkt der Betrachtung und wird damit ihrer, durch die technologische Entwicklung immer mehr an Bedeutung gewinnenden Rolle im physikalischen VLSI-Entwurf, gerecht.

2. PROBLEMANALYSE

In diesem Kapitel werden zuerst die Grundlagen des physikalischen VLSI Entwurfes dargelegt und der Einfluß der aktuellen Entwicklung im Bereich der Produktionstechniken für integrierte Schaltungen auf den Entwurfsprozeß aufgezeigt. Es wird eine Möglichkeit skizziert, die einige der zu erwartenden Schwierigkeiten aus diesem technologischen Wandel auf innovative Art bewältigt. Ausgehend von dieser Idee werden dann die Randbedingungen und Annahmen, unter denen dieses neue Vorgehen betrachtet werden soll, dargelegt. Den Abschluß dieses Kapitels bildet die Darlegung der in dieser Arbeit verfolgten Zielsetzung und die Definition der Aufgabenstellung.

2.1 Physikalischer VLSI-Entwurf “heute”

Aufgabe des physikalischen VLSI-Entwurfes ist es, die in Form einer Netzliste gegebene Schaltung für die Produktion geeignet umzusetzen. Dabei wird eine Beschreibung der Anordnung der verwendeten Grundelemente und ihrer elektrischen Verbindungen auf dem Chip erstellt. Das Ergebnis ist eine vollständige geometrische Beschreibung des Chips, das Chiplayout. Das im folgenden Dargelegte ist, sofern nicht anders erwähnt, aus [She95] und [Ger99] entnommen.

Für die Herstellung integrierter Schaltungen gibt es verschiedene Entwurststile mit zum Teil erheblich unterschiedlichen Charakteristiken. Der Full-Custom Entwurf verwendet als Grundelemente die bei der Herstellung verwendeten geometrischen Formen, also Rechtecke mit wählbarer Länge und Breite. Der Entwurfsaufwand ist sehr hoch, aber kein anderer Entwurststil erreicht die mit dem Full-Custom Entwurf mögliche Effizienz. Daher lohnt sich dieser Entwurststil für Großserien und Entwürfe, für die extrem schwierige Bedingungen, etwa sehr hohen Taktfrequenzen, gefordert werden. Andere Entwurststile verwenden mehr oder weniger weit vorgefertigte Grundbausteine und kombinieren diese in der Art, daß das gewünschte Verhalten erzielt wird. Der Entwurfsaufwand sinkt erheblich, die Resultate sind jedoch immer ein Kompromiß. Ein wesentlicher Grund für den deutlich reduzierten Entwurfsaufwand ist die gute Automatisierbarkeit des physikalischen Entwurfs für diese auf hoch regulären Strukturen basierenden Layoutstile. So bestehen Gate Arrays und FPGAs (FPGA: Field Programmable Gate Array) aus einer Vielzahl von auf einem Raster angeordneten gleichartigen Zellen. Das Verhalten eines Gate Arrays oder FPGAs wird durch die Verschaltung dieser Zellen miteinander sowie der Festlegung der Funktionalität der einzelnen Zellen definiert. Beim Standardzellentwurf werden verschiedenartige Zellen verwendet, die in parallelen Reihen angeordnet werden. In Abbildung 1 sind Beispiellayouts zu diesen vier Entwurststilen dargestellt.

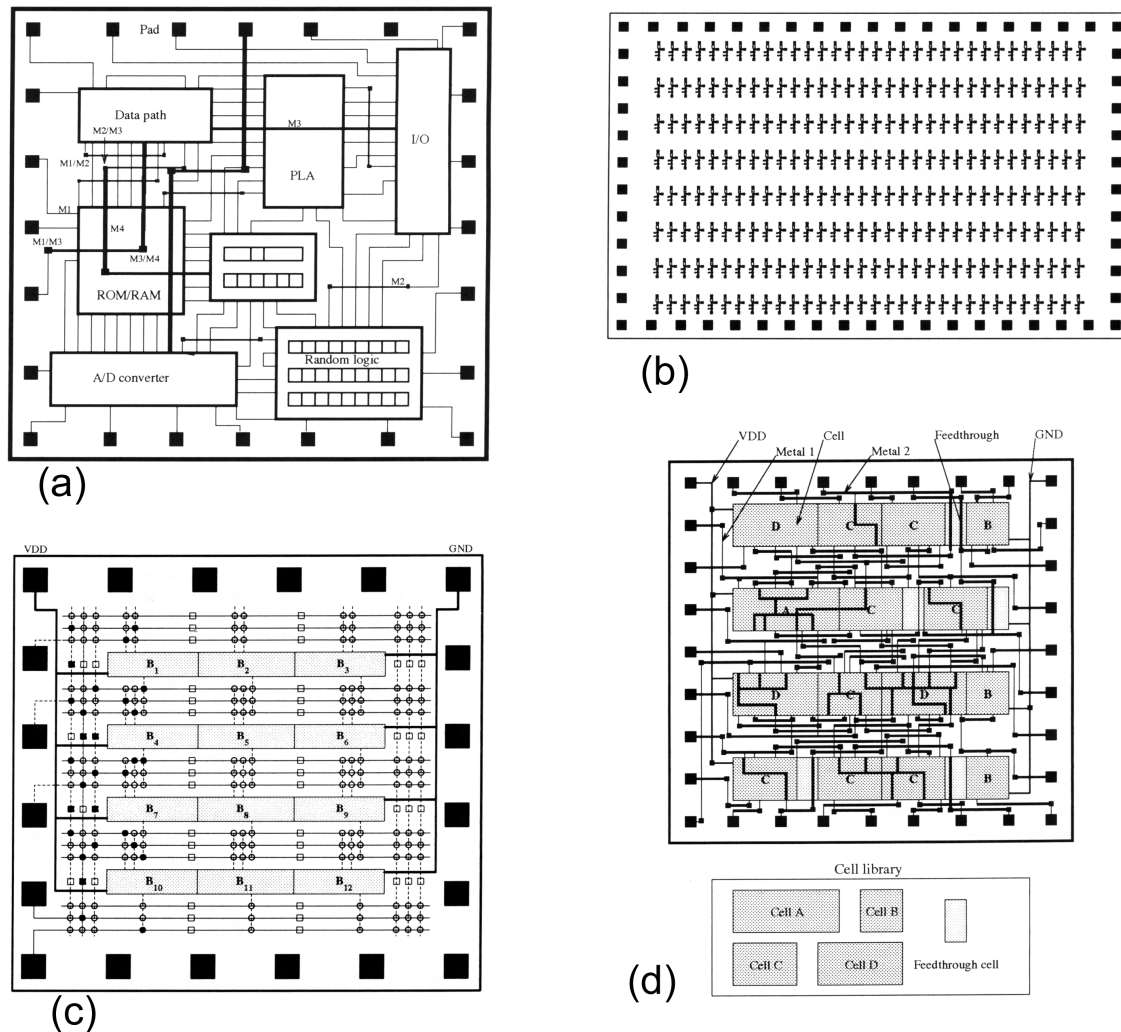


Abbildung 1: Verschiedene Entwurststile: Full Custom (a), Gate Array (b), FPGA (c) und Standardzellentwurf (d). Abbildungen aus [She95]

Die Entscheidung für einen Entwurststil ist von der Stückzahl des entworfenen Chips, sowie dessen Komplexität und der zur Verfügung stehenden Entwicklungszeit abhängig. Ein Überblick über Entwurststile findet sich zum Beispiel in [She95].

Allen Entwurststilen ist gemein, daß im Rahmen des physikalischen Entwurfs die durch die Netzliste gegebene Schaltung auf die verfügbaren Basiselemente für die aktiven Elemente und die Verdrahtung abzubilden ist. Dabei sind, unter Berücksichtigung der technologischen Randbedingungen, die für den fertigen Chip gegebenen Ziele (z.B. die Taktfrequenz) zu erfüllen. Primäre Ziele des physikalischen VLSI-Entwurfs sind z.B. die Reduktion der benötigten Chipfläche, die Verkürzung der maximalen sowie der durchschnittlichen Netzlänge und die Erhöhung des Prozentsatzes an funktionsfähigen Chips (Yield) durch Reduktion der Anzahl produktionsfehlerträchtiger Strukturen wie z.B. Kontaktlöcher. Die genannten Kriterien für das Layout widersprechen einander teilweise. So kann die Verringerung der Anzahl von Kontaktlöchern leicht zu einer Vergrößerung des Flächenbedarfs führen.

Der physikalische VLSI-Entwurf gehört zur Gruppe der NP-harten Probleme. Eine analytische Lösung der, auf Grundlage obiger Anforderungen aufgestellten, Zielfunktion

kommt daher nur für sehr kleine Netzlisten in Betracht. Im allgemeinen handelt es sich beim VLSI-Entwurf jedoch um Problemstellungen, die aus mehreren tausend Grundelementen (Zellen) sowie deren elektrischen Verbindungen (Netze) bestehen. Um die Problemgröße in einem bewältigbaren Rahmen zu halten, ist es das übliche Vorgehen, den Schritt des physikalischen VLSI-Entwurfs in die beiden Teilprobleme Plazieren und Verdrahten zu zerlegen. Während des Plazierens werden die Zellen aus der Netzliste gemäß dem verwendeten Entwurfsstils auf der zur Verfügung stehenden Chipfläche verteilt. Das anschließende Verdrahten fügt die benötigten elektrischen Verbindungen ein. Diese Zweiteilung ist nicht unproblematisch und die Teilprobleme lassen sich nicht völlig separat betrachten. Die Aufteilung ist jedoch aufgrund der Komplexität notwendig, denn in aktuellen Entwürfen sind mehrere hunderttausend Zellen zu positionieren und zu verbinden. Die Zerlegung in Plazieren und Verdrahten reduziert allerdings nur die Größe der zu betrachtenden Teilprobleme, sie bringt zwei NP-harte Teilprobleme hervor. Diese beiden Teilprobleme (Plazieren und Verdrahten) werden jeweils mit heuristischen Verfahren gelöst.

Man unterscheidet dabei zwei Klassen von Algorithmen, die konstruktiven und die iterativen. Die konstruktiven Algorithmen erstellen eine Lösung und sobald alle zu berücksichtigenden Komponenten enthalten sind, ist dies das Resultat. Iterative Verfahren beginnen bei einer zuvor erzeugten suboptimalen (gültigen) Lösung und führen eine Folge von Transformationen aus, die auf eine bessere Lösung hinzielen. Die Startlösung kann auf einfachste Weise erzeugt werden, für die Plazierung etwa durch zufällige Anordnung der Komponenten. Meist wird für die Plazierung ein iteratives Verfahren, z.B. Simulated Annealing, und für die Verdrahtung ein konstruktives, z.B. der Lee-Algorithmus, verwendet.

Die Aufgabe des Verdrahtens kann in die Globalverdrahtung und die Detailverdrahtung aufgeteilt werden. Während der Globalverdrahtung wird für jedes Netz bestimmt, durch welche Verdrahtungsregionen es verläuft und an welchen Stellen es in die einzelnen Regionen eintritt bzw. sie wieder verläßt. Dabei wird der Weg des Netzes innerhalb der Regionen nicht exakt festgelegt. Die Detailverdrahtung betrachtet die Verdrahtungsregionen einzeln und bestimmt für jedes Netz den exakten Verlauf innerhalb der Region. Diese Aufteilung ist nicht zwingend und weniger stark ausgeprägt als die Trennung zwischen Plazieren und Verdrahten. Es gibt Verdrahtungsalgorithmen, die diese Trennung vornehmen, ebenso wie solche Algorithmen, die in nur einem Schritt direkt eine vollständige Verdrahtung erzeugen.

Obwohl die beiden Teile Plazieren und Verdrahten voneinander getrennt betrachtet und einzeln bearbeitet werden, sind sie so eng gekoppelt, daß dieses Vorgehen neue Schwierigkeiten aufwirft. Die Qualität einer Plazierung läßt sich erst nach der vollständigen Verdrahtung beurteilen. Zur Erstellung einer guten Plazierung müßte folglich im Idealfall nach jedem Hinzufügen von Zellen oder Verändern ihrer Positionen eine vollständige Verdrahtung vorgenommen werden. Da dies nicht effektiv möglich ist, werden Plazierungen und Teilplatzierungen vermöge mehr oder weniger einfach zu berechnender Bewertungsfunktionen beurteilt. Diese Funktionen ermitteln die Qualität der Plazierung im Hinblick auf eine günstige Anordnung für das Verdrahten ab. Das von ihnen gelieferte Qualitätsmaß weicht jedoch im allgemeinen in seiner Bewertung von der nach dem Verdrahten vorgenommenen Bewertung einer Plazierung ab.

Das Verdrahten wiederum setzt auf einer bestehenden Platzierung auf. Diese Platzierung kann von dem Verdrahtungsalgorithmus im allgemeinen nicht modifiziert werden. Eine ungünstige Zellanordnung kann zu einer ineffizienten Verdrahtung oder gar einem vollständig nicht verdrahtbaren Entwurf führen. Ist dies der Fall, so muß der gesamte Vorgang, beginnend mit dem Platzieren, neu durchlaufen werden.

Meist läßt sich ein schlechtes Layout nicht durch alleiniges Neuverdrahten ausreichend verbessern, so daß im allgemeinen zur Verbesserung des Ergebnisses ebenfalls der gesamte Zyklus, beginnend mit der Platzierung, erneut durchlaufen werden muß.

Das Ergebnis eines Verdrahtungsalgorithmus hängt also auch stark von der Qualität der zuvor erstellten Platzierung ab. Die Platzierung definiert die Aufgabe für den Verdrahter genauso wie es die Netzliste und die weiteren Randbedingungen tun. Damit befindet sich der Verdrahter gegenüber dem Platzierer in einer benachteiligten Situation.

2.2 State Of The Art

Heutige Ansätze zum physikalischen VLSI Entwurf halten überwiegend an der gängigen Trennung in Platzieren und Verdrahten fest. Aktuelle Algorithmen und Entwicklungen konzentrieren sich meist auf die Verbesserung existierender Teillösungen oder auf spezielle Problemfälle, die durch neue Randbedingungen veränderter Herstellungsprozesse gegeben sind.

In [LLL96] wird ein Platzierungsansatz vorgestellt, der durch Vergrößerung (Bottom-up Clustering) des für die Platzierung verwendeten Graphen die Problemgröße reduziert. Für das modifizierte Problem können dann Algorithmen verwendet werden, die für das ursprüngliche Problem, aufgrund seiner Größe, nicht anwendbar wären. Dabei kommt eine Bewertungsfunktion für die Platzierung zur Anwendung, obwohl bekannt ist, daß die Qualität der Verdrahtung nicht immer mit der Qualität aus dieser Bewertungsfunktion korreliert. Der Fokus ist nur auf die Verbesserung der Platzierung gerichtet, das Verdrahtungsproblem bleibt völlig unbetrachtet.

Der in [HuL99] vorgestellte Platzierungsalgorithmus verwendet verschiedene Granularitäten bei der Betrachtung des Graphen, der das Platzierungsproblem modelliert. Durch die dynamische Bildung und Auflösung von Clustern aus einigen zig Knoten wird erreicht, daß der auf lokalen Veränderungen der Platzierung basierende Algorithmus nicht in einem lokalen Minimum der Bewertungsfunktion gefangen bleibt. Durch die Clusterung gelingt es, das lokale Minimum zu verlassen und eine insgesamt bessere Platzierung zu erreichen. Die Beurteilung der Platzierungsqualität geschieht durch eine Schätzfunktion, das Verdrahtungsproblem wird wiederum nicht betrachtet.

Eine Anpassung bewährter Algorithmen zur Suche in großen Gitterstrukturen wird in [Het98] präsentiert. Hierbei wird lediglich die Art der Suche nach möglichen Verdrahtungswegen diskutiert und die Platzierung als unverrückbar akzeptiert. Notwendig sind solche Erweiterungen aufgrund der zunehmenden Größe der Suchräume.

Weitere aktuelle Projekte zum Verdrahtungsproblem betrachten das Einfügen von benötigten Leitungstreibern in Netze als Teil der Aufgaben des Verdrahtungsalgorithmus. So wird in [LaW00] und [ZWL99] das Verdrahten unter dem Aspekt der kürzesten Verzögerung nicht nur auf die Wegewahl für die Leitungssegmente beschränkt, sondern das Einfügen von Leitungstreibern in dieses integriert. Es ist Aufgabe des Verdrahtungsalgorithmus durch Kombination der Positionierung von Leitungstreibern

(Buffer Insertion) und der Wegewahl eine Verdrahtung mit minimaler Verzögerung zu finden. [LaW00] betrachtet als zusätzlichen Aspekt die Breite der Leitungssegmente (Wiresizing). In [ZWL99] wird die Wahl der Ebene in der die Leitungssegmente verlaufen sollen (Layer Assignment) als weiterer Parameter zur Beeinflussung der Verzögerung berücksichtigt. Diese Ansätze lassen somit einen Teil der Platzierung durch den Verdrahter ausführen, jedoch wird die Lage der zu verdrahtenden Zellen (oder Blöcke) nicht angetastet. Die Grenze zwischen Platzieren und Verdrahten wird dadurch unscharf, jedoch ist der Einfluß des Verdrahters auf die Platzierung eng begrenzt, nämlich auf das Einfügen und Positionieren reiner Leitungstreiber beschränkt. Die Wahl der möglichen Positionen für diese Leitungstreiber wird durch die Platzierung der zu verbindenden Zellen beeinflusst, da in den betrachteten Modellen die Verdrahtung über die platzierten Zellen hinweg (OTC-Routing: Over-the-Cell Routing) meist erlaubt ist, die Regionen von Zellen oder Blöcken jedoch keine weiteren aktiven Elemente, wie eben die Leitungstreiber, aufnehmen können. Durch die technologische Notwendigkeit vermehrt Leitungstreiber in Netze einzufügen, wird das Verdrahtungsproblem mittlerweile kaum noch getrennt vom Einfügen von Leitungstreibern betrachtet [ZWL99], es kann also als neues Verständnis des Verdrahtens gesehen werden. Die Unveränderbarkeit der Platzierung, mit Ausnahme der Leitungstreiber, bleibt dabei bestehen.

2.3 Technologische Entwicklung

Zu Beginn des CAD-gestützten VLSI-Entwurfes lag die Anzahl der zu berücksichtigen Elemente in der Größenordnung mehrerer Tausend, heute ist sie um mehr als das einhundert bis tausendfache größer. Als Beispiel soll hier die Anzahl der Transistoren eines Mikroprozessors dienen. Der im Jahre 1971 am Markt eingeführte 4004 Prozessor von Intel bestand aus 2250 Transistoren, der im Jahre 2001 von Intel präsentierte Pentium4 aus ca. 42 Millionen Transistoren [Int01].

Die technologische Entwicklung führte und führt zu immer kleineren Strukturen auf den Chips und mit der einhergehenden höheren Integration zu sich beständig ändernden Anforderungen an die Algorithmen für den physikalischen VLSI-Entwurf. Vor 20 Jahren dominierte die von den aktiven Zellen benötigte Fläche die Größe des Entwurfs. Die Verdrahtung der Netze erfolgte auf einer oder zwei Ebenen in explizit ausgewiesenen Regionen zwischen den aktiven Blöcken. Die Größe der von den Verdrahtungsregionen beanspruchten Fläche wurde durch die Qualität der Platzierung beeinflusst, machte aber weniger als 40% der Chipfläche aus [She95].

Die massive Verkleinerung der Strukturgrößen auf einem Chip führte zum einen zur Integration von immer mehr Elementen auf einem Chip und zum anderen zu einer Erhöhung der Taktfrequenz. Während der DEC Alpha Prozessor 1992 mit Strukturgrößen von $0,75\text{ }\mu\text{m}$ gefertigt wurde [Gil93], beginnt Intel im Jahr 2001 mit der Produktion in einem $0,13\text{ }\mu\text{m}$ Prozeß [Int00]. Garant für die effektive Nutzung der Verkleinerung der Strukturgrößen ist die hohe Positionierungsgenauigkeit für die bei der Fertigung verwendeten Masken. Die erhöhte Präzision des Lithographieprozesses ermöglicht mittlerweile mehr als zwei Verdrahtungsebenen, vier bis fünf Ebenen sind heute für Mikroprozessoren in der Anwendung [She95], sechs Ebenen sind in den neuesten Fertigungsprozessen realisierbar geworden [TAB00].

Durch die Verkleinerung der Strukturgrößen steigt bei gleichbleibender Chipgröße die Netzlänge relativ zur Strukturgröße, die gleichzeitig erfolgende Vergrößerung der

Chipfläche erhöht die absoluten Netzlängen. Beide Effekte verstärken sich in ihren Auswirkungen, so daß die Netzlänge immer stärkere Bedeutung für den Entwurfsprozeß erhält. Die Länge der Netze bestimmt das Ausmaß der zum Teil unerwünschten Eigenschaften der Leitungen wie parasitäre Kapazitäten und Antenneneffekte. Je länger die Netze sind, um so mehr wirken sich diese Effekte aus. Das Unterschreiten der maximal zulässigen Netzlänge ist mittlerweile im allgemeinen nur durch das Einfügen von Leitungstreibern in die Netze zu gewährleisten, wie die oben erwähnten Forschungsaktivitäten belegen.

Die Verkleinerung der Strukturgrößen wirkt sich auf die aktiven Elemente stärker aus als auf die Dichte der Leitungen in den Verdrahtungsregionen. Der Platzbedarf der Zellen im Vergleich zum Platzbedarf der Leitungen nimmt somit technologisch bedingt ab. Durch die größer werdende Komplexität der Netzlisten nimmt der Anteil der Verdrahtungsfläche an der Gesamtchipfläche im Vergleich zum Anteil der Fläche aktiver Zellen an der Gesamtchipfläche (bei gleichbleibender Strukturgröße) zu. Diese beiden Effekte haben dazu geführt, daß heute der Flächenbedarf für die Verdrahtung beträchtlich höher ist als in der Vergangenheit. Die Verwendung von mehr Verdrahtungsebenen als früher mildert diese Entwicklung ab, jedoch läßt sich der Bedarf an Verdrahtungsfläche nicht beliebig durch Hinzunahme weiterer Verdrahtungsebenen reduzieren. Die Wechselwirkungen zwischen benachbarten Leitungssegmenten nehmen durch Verkleinerung der Strukturgrößen und die Erhöhung der Taktfrequenz stark zu. Die Packungsdichte von Leitungssegmenten ergibt sich dadurch weniger aus den produktionstechnisch machbaren Strukturgrößen als vielmehr aus den erforderlichen Mindestabständen zur Reduktion der störenden Wechselwirkungen, wie parasitären Kapazitäten, Übersprechen von Signalen und Antenneneffekten. Der verwendete Abstand zweier benachbarter Leitungselemente ist heute um das 5- bis 15-fache größer als die Größe eines Transistors [TAB00]. Die sich immer stärker auswirkenden Wechselwirkungen zwischen benachbarten Netzen beschränkt die maximal zulässige Netzlänge erheblich. Dies und die große Komplexität der einzelnen Netze haben das Verdrahtungsproblem in den letzten Jahren zum dominierenden Problem des physikalischen VLSI-Entwurfs werden lassen ([Law00], [Het98]).

2.4 Verdrahten vor dem Plazieren

Trotz der wachsenden Dominanz des Verdrahtungsproblems müssen die Verdrahtungsalgorithmen in gängigen Lösungsansätzen mit der vorgegebenen Platzierung zurechtkommen und sind somit durch die Reihenfolge, in der Plazieren und Verdrahten erfolgt, den Plazierern gegenüber benachteiligt. Diese Überlegung läßt es wünschenswert erscheinen, die schwierigere Aufgabe des Verdrahtens als Erstes zu lösen und die Zellen erst in einem zweiten Schritt einzufügen. Dabei muß allerdings die für die Zellen benötigte Fläche in der Verdrahtung problemlos zu finden sein.

Die Idee zur Umkehr der Bearbeitungsreihenfolge der Teilprobleme Plazieren und Verdrahten ist die Basis des in dieser Arbeit dargelegten Lösungsansatzes für den physikalischen VLSI-Entwurf. Sie geht auf den ehemaligen Leiter des Arbeitsbereiches „Technische Grundlagen der Informatik“ (TECH) des Fachbereichs Informatik der Universität Hamburg, Prof. Dr.-Ing. Klaus Lagemann, zurück. Sie entstand, als ein Chipdesign wiederholt neu platziert und verdrahtet werden mußte. Die wiederholte Unverdrahtbarkeit der Platzierung lag an der fehlerhaften Abschätzung des Flächenbedarfes

für das Verdrahten durch den verwendeten Platzierer und nicht am verwendeten Verdrahtungsalgorithmus.

Die Annahme, daß die für die Zellen benötigte Fläche sich ohne wesentliche Veränderung der zuvor (ohne Platzierung) erstellten Verdrahtung finden lasse, ist jedoch bisher nicht gegeben. Erst mit dem heutigen Stand der fertigungstechnischen Entwicklung ist die Erfüllung dieser Forderung in absehbarer Zeit im Bereich des Möglichen.

Extrapoliert man die ungebrochene Entwicklung zur Verkleinerung der aktiven Elemente, so sind Zellen in der Größenordnung der Breite der Verdrahtungselemente denkbar. An diesem Punkt angekommen wird der Abstand zweier benachbarter Leitungselemente zum alleinigen limitierenden Kriterium und der Platzbedarf der Zellen selbst tritt beim Platzieren fast vollständig in den Hintergrund. Was bleibt, ist die Zielsetzung, diese als punktförmig annehmbaren Zellen möglichst platzsparend zu verbinden. Schon heute läßt sich dies beobachten, wenn in einem Produktionsprozeß ausreichend viele Verdrahtungsebenen verfügbar sind, so daß die Verdrahtung unabhängig von explizit ausgewiesenen Verdrahtungskanälen und dabei über die platzierten Zellen hinweg erfolgen kann (OTC-Routing).

Soll das Verdrahtungsproblem zuerst betrachtet werden, ist der intuitive erste Ansatz, die Verdrahtung ohne die Zellen als störende Hindernisse durchzuführen. Dies ergibt offensichtlich das Problem, Leitungen zu verlegen, ohne zu wissen, wo die zu verbindenden Elemente positioniert sind. Führt man als Abhilfe eine vorübergehende Positionierung der Zellen ein, die etwa iterativ verändert werden kann oder rekursiv detaillierter wird, so ist dies ein paralleles Lösen von Platzierungs- und Verdrahtungsproblem. Für nicht allzu komplexe Schaltungen ist das ein bei der heutigen Verarbeitungsgeschwindigkeit der CAD-Systeme durchaus gangbarer Weg. Dieses Vorgehen spiegelt jedoch nicht die obige Idee wider, das Verdrahtungsproblem vor dem Platzieren der Zellen durchzuführen.

Das Festhalten an einer Sicht auf die Zellen und Verbindungen eines Layouts, die sich an ihrer physikalischen Manifestation orientiert, muß zwangsläufig immer an das Problem gelangen, daß die Leitungen immer erst festgelegt sind, sobald ihre Endpunkte bekannt sind. Damit ist ein Lösen des Verdrahtungsproblems vor dem Platzieren, also nicht klassisch danach oder gleichzeitig mit diesem, ausgeschlossen.

Dieses prinzipielle Problem läßt sich angehen, indem die Granularität der Betrachtung angepaßt wird und zunächst nur die Globalverdrahtung betrachtet wird. Auf einer Ebene größerer Granularität ist ein Netz als atomare Einheit repräsentierbar und nicht als ein aus mehreren Leitungselementen zusammengesetzter Baum. Ziel ist es nun, die Netze so auf der Chipfläche anzuordnen, daß sich Netze, die mit einer gemeinsamen Zelle verbunden sind, an mindestens einem Ort nahe kommen, so daß die Zelle hier platziert werden kann. Dabei dürfen sich die Bereiche, die die Netze überdecken, überschneiden. In Abbildung 2 ist gezeigt, in welchem Gebiet sich die Position einer an drei Netze angeschlossenen Zelle bei diesem Vorgehen befinden soll.

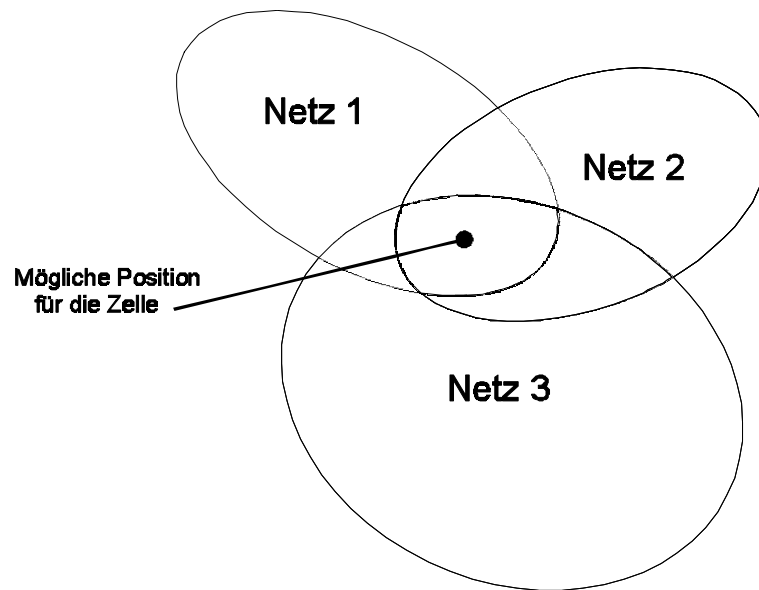


Abbildung 2: Von Netzen überdeckte Gebiete auf dem Chip und günstige Lage einer angeschlossenen Zelle

Die Globalverdrahtung läßt sich somit als ein Platzierungsproblem für Netze auffassen. Bevor eine endgültige Verdrahtung (einschließlich der Detailverdrahtung) erfolgen kann, muß eine Platzierung der Zellen existieren, so daß die Lage der durch die Netze zu verbindenden Punkte bekannt ist. Auch wenn dieser Ansatz es nicht erlaubt, die Verdrahtung komplett vor der Platzierung durchzuführen, bietet er doch die Möglichkeit die Verdrahtungsproblematik vor dem Erzeugen einer Platzierung zu berücksichtigen. Damit erscheint es ein vielversprechendes Vorgehen zu sein, welches die Idee, das Verdrahten vor dem Plazieren durchzuführen, geeignet mit dem Realisierbaren verbindet.

Die Anordnung der Netze bestimmt den lokalen Bedarf an Verdrahtungsressourcen. In Regionen mit vielen einander überlappenden Netzen wird mehr Platz für die Unterbringung der Leitungssegmente benötigt, als in Regionen mit wenigen Netzen. Eine gleichmäßige Verteilung der Netze kann nun direkt durch den Algorithmus erfolgen, der die Netze anordnet. Es ist nicht nötig dies indirekt durch die Anordnung der Zellen beim Plazieren vermöge einer Bewertungsfunktion zu steuern.

In dieser Arbeit wird genau dieser Ansatz verfolgt. Zunächst werden die Netze in einer geeignet zu wählenden Repräsentation in der Ebene angeordnet. Anschließend werden die Zellen zwischen ihnen plaziert (siehe Abbildung 2). Als abschließender Schritt werden noch die Leitungssegmente der Netze eingefügt und das Layout somit vervollständigt.

2.5 Annahmen, Randbedingungen

Die im letzten Abschnitt dargelegte Idee für eine alternative Lösung der Aufgabe des physikalischen VLSI-Entwurfs wird durch die aktuelle und sich sehr wahrscheinlich in Zukunft fortsetzende Entwicklung der Fertigungstechnologie interessant. Mittlerweile sind die Strukturgrößen und das Größenverhältnis der Verdrahtungsrasterabstände zur Größe der aktiven Elemente eine akzeptable Näherung für den angenommenen Grenzzustand, in dem die Zellen verschwindend klein gegenüber der Leitungsbreite werden.

Unter den im folgenden aufgeführten Annahmen scheint diese Methode besonders erfolgversprechend. Die Zellen sind als klein gegenüber dem verwendeten Verdrahtungsraster anzusehen. Idealisiert lassen sie sich durch Punkte repräsentieren,

werden aber in ihrer Ausdehnung in Realität auch so klein sein, daß der von ihnen benötigte Platz vernachlässigbar ist gegenüber dem Platzbedarf für Verdrahtungsressourcen. Insbesondere verursacht das nachträgliche Einfügen von Zellen in eine existierende Verdrahtung keine wesentlichen Veränderungen an dieser. Die Verdrahtung ist das dominierende Problem, gegeben durch Forderungen nach kurzen Netzen, wenigen kritischen Abschnitten und der steigenden Komplexität der Netzlisten. Für die Verdrahtung werden keine Flächenrestriktionen angenommen, es werden also keine Verdrahtungskanäle ausgewiesen, außerhalb derer keine Leitungssegmente verlaufen dürfen. Vielmehr werden die Verbindungen, ähnlich dem beim Platinenentwurf heute üblichen Vorgehen, in „Wildwestverdrahtung“ angeordnet.

2.6 Zielsetzung

In dieser Arbeit soll die in Kapitel 2.4 präsentierten Idee ausgearbeitet werden und daraus ein kompletter Platzierungs- und Verdrahtungsalgorithmus entwickelt werden. Aus den dargelegten Gründen wird es sich dabei um Algorithmen in der klassischen Reihenfolge Platzieren und Verdrahten handeln, jedoch wird die Platzierung der Zellen komplett aus der zuvor vorgenommenen Anordnung der Netze abgeleitet. Außerdem wird dem Verdrahtungsalgorithmus ermöglicht, die Platzierung durch lokale Änderungen an seine Bedürfnisse anzupassen, wodurch die oben beschriebene harte Trennung der beiden Teilprobleme aufgeweicht wird.

Zur Evaluierung des so entwickelten Algorithmus wird ein Prototyp implementiert, dessen Ergebnisse mit existierenden Platzierern und Verdrahtern verglichen werden sollen. Zur Realisierung der Teilschritte des entstehenden Algorithmus sollen, so weit möglich erprobte, einfache Algorithmen zur Anwendung kommen. Dies hat zum einen den Vorteil, daß die Implementierungsarbeit möglichst gering gehalten werden kann und daß zum anderen die Ergebnisse sich leichter bewerten lassen. Würden für die Teilschritte des neuen Algorithmus wiederum innovative neue Methoden verwendet, wo es wohl bekannte Alternativen gibt, so fiel es schwer zu beurteilen, ob das Endergebnis bestimmte Eigenschaften wegen der guten Teillösungen oder des neuen Gesamtansatzes aufweist.

Um die Ergebnisse des zu entwickelnden Algorithmus mit Layouts anderer Platzierer und Verdrahter vergleichen zu können, wird auf das offene System ESPRO (siehe Kapitel 3) zurückgegriffen. Durch die Verwendung von ESPRO ist es möglich auf einfache Weise einen Vergleich mit anderen Algorithmen herstellen zu können. Die durch ESPRO zusätzlich entstehenden Randbedingungen und die daraus resultierenden Folgen für den hier erstellten Algorithmus sind in Kapitel 3.3 dargelegt.

3. DIE EVALUATIONSUMGEBUNG: ESPRO

In diesem Kapitel wird die verwendete Evaluationsumgebung vorgestellt und die Konsequenzen, die sich aus ihrer Verwendung für den zu implementierenden Algorithmus ergeben, dargelegt.

Das „Evaluation System for Placement and Routing“ (ESPRO) ist ein offenes System, das dazu konzipiert wurde, Algorithmen für den physikalischen VLSI-Entwurf möglichst unabhängig von den Details des Herstellungsprozesses untersuchen zu können. Es ist im Rahmen mehrerer Studien- und Diplomarbeiten am Arbeitsbereich TECH des Fachbereichs Informatik der Universität Hamburg entstanden.

ESPRO verwendet ein stark abstrahierendes Modell, welches sich auf die für den physikalischen Entwurf relevanten Basiseigenschaften des Standardzellentwurfes konzentriert. Es ist eine Sammlung von Programmen, die Module für die Erstellung von Beispielschaltungen, den Import von Netzlisten, das Plazieren, Verdrahten und die Nachoptimierung sowie die statistische Analyse der entstandenen Layouts zur Verfügung stellt. Ein Systemüberblick ist der Abbildung 4 zu entnehmen, die einzelnen Komponenten werden in Abschnitt 3.2.1 näher vorgestellt.

3.1 Modell und Entwurfsstil von ESPRO

Das in ESPRO verwendete Modell hat zum Ziel, die für den physikalischen VLSI-Entwurf relevanten Aspekte eines Schaltkreises darzustellen. Von Eigenschaften, die nur durch den verwendeten Herstellungsprozeß definiert werden, und damit dem schnellen Wandel der technologischen Entwicklung in diesem Sektor unterworfen sind, wird abstrahiert. So kann dieses Modell über einen längeren Zeitraum Verwendung finden. Ein weiterer Vorteil der starken Abstraktion ist die Reduktion des Aufwandes für die Implementierung zu evaluierender Algorithmen.

Obwohl auf längerfristige Verwendung ausgelegt, orientiert sich das Modell von ESPRO an den zum Zeitpunkt der Modellbildung gegebenen Produktionsparametern. Der in ESPRO verwendete Entwurfsstil ist der Standardzellentwurf für ASICs (ASIC: Application Specific Integrated Circuit). Die im Modell verwendete Maßeinheit ist der kleinste zulässige Abstand zwischen zwei benachbarten Leitungssegmenten. Welchen Wert diese Distanz auf einem realen Chip haben würde, ist abhängig vom Herstellungsprozeß und wurde in den ersten Implementierungen mit $5.59 \cdot 10^{-3}$ mm gleichgesetzt. Dieser Wert wird in allen bisherigen Implementierungen verwendet und bildet die Grundlage für alle Längen- und Flächenangaben, die bezüglich Strukturen eines ESPRO Layouts gemacht werden.

3.1.1 Der Entwurfsstil in ESPRO

In ESPRO werden Standardzellentwürfe modelliert. Der Kernbereich (Core Area) des Chips enthält die Standardzellreihen, die externen Anschlüsse des Chips, die Padzellen oder kurz Pads, sind in einem Ring am Rand der Chipfläche angeordnet, dem Padkranz. Die Anschlußpunkte der Pads befinden sich an ihrer inneren Kante. Die Padzellen können nicht an beliebigen Stellen einer Kante liegen, sondern werden an äquidistant über die Kante verteilten Positionen, im weiteren auch Slots genannt, angeordnet.

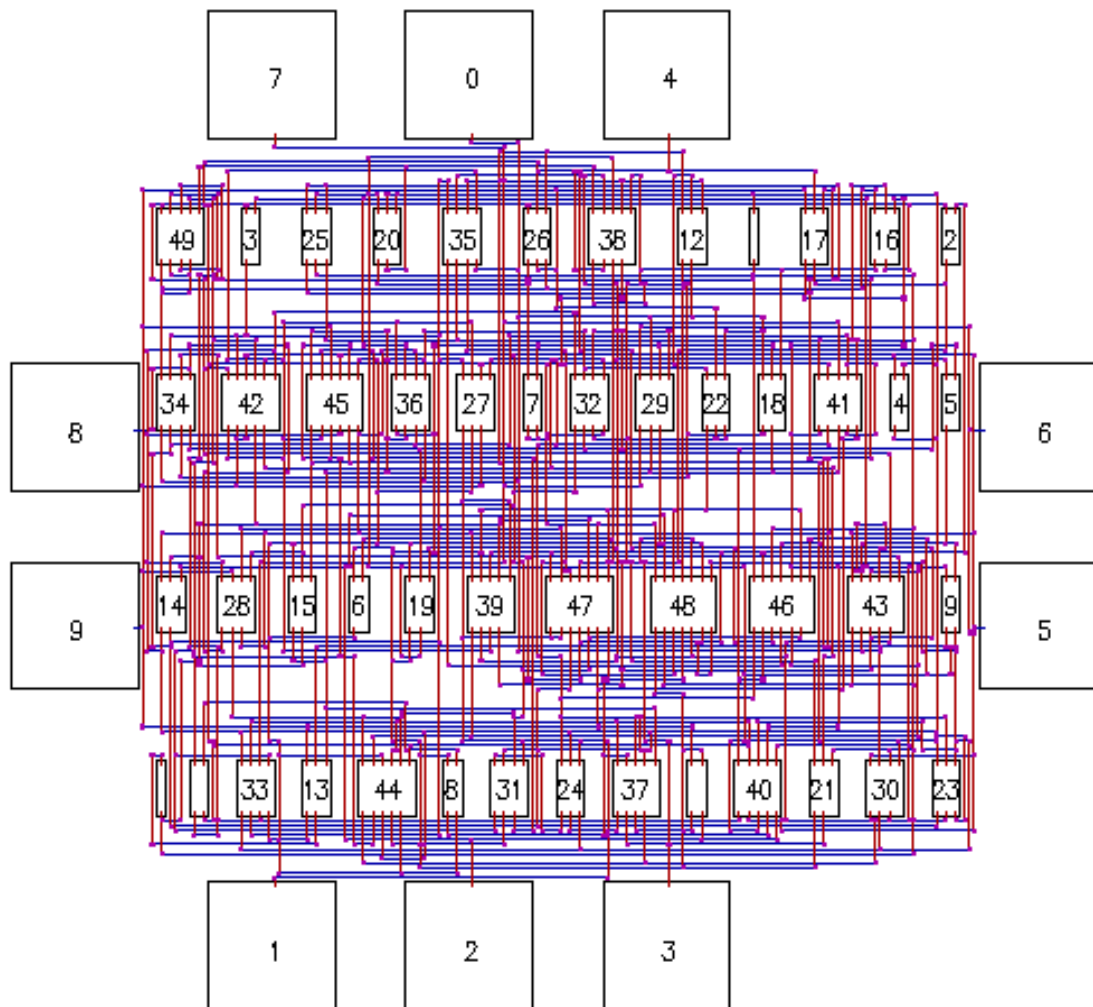


Abbildung 3: Beispiel eines mit ESPRO erzeugten Layouts, wie es vom Visualisierungswerkzeug in ESPRO dargestellt wird.

Die Parameter für die Zellbibliothek und den Fertigungsprozeß wurden aus zum Zeitpunkt der Modellbildung verbreiteten Prozessen abgeleitet und abstrahiert.

Das verwendete Modell ist bewußt stark abstrahiert, dies hält den Implementationsaufwand in bewältigbaren Dimensionen und filtert Effekte heraus, die nur für eine bestimmte, nämlich eben die betrachtete, Zellbibliothek gelten, indem nur allgemeingültige Eigenschaften in das Modell aufgenommen werden.

3.1.2 Die Zellbibliothek in ESPRO

Eine Zelle in ESPRO ist ein Rechteck mit fester Höhe und einer Breite, die von der Anzahl ihrer Anschlußpunkte (Pins) abhängt. Alle Zellen haben die gleiche Höhe, aber eine individuelle, von der Anzahl ihrer Pins abhängende, Breite. Die Pins befinden sich an der oberen bzw. unteren Kante der Zelle, eine Unterscheidung nach Eingangs- und Ausgangspins wird nicht vorgenommen. Es gibt keine funktional äquivalenten Pins an einer Zelle, d.h. jeder Signalanschluß einer Zelle ist genau einem Pin zugeordnet. Die Zellen sind nicht, wie in modernen Prozessen meist zulässig, entlang einer vertikalen oder horizontalen Linie spiegelbar. Von der Funktion der Zelle wird vollständig abstrahiert, da sie für das Platzierungs- und Verdrahtungsproblem keine Rolle spielt. Alle Zellen sind

opak, d.h. die von ihnen belegte Fläche ist für das Verdrahten gesperrt. Ein Verdrahten über die Zellen hinweg (OTC-Routing) ist nicht zulässig.

Heutige Zellbibliotheken enthalten meist mehrere Implementierungen für eine bestimmte Funktionalität [DKS98]. Diese auf der Logikebene äquivalenten Zellen unterscheiden sich in der Leistung ihrer Ausgangstreiber. Damit können für kleine Netze Zellen mit geringer Treiberleistung verwendet werden, was den Platzbedarf für die Zellen und die Leistungsaufnahme des Chips reduziert. Diese Auswahl der Treiberstärke von Zellen wird in der momentanen Form des Modells nicht unterstützt, die dafür nötige Information, welche Pins eines Netzes Treiber und welche Eingänge sind, ist in der Zellbeschreibung nicht enthalten.

3.1.3 Modell für die Verdrahtung

Es werden nur die Netze der Logikleitungen betrachtet. Die Power-, Ground-, Reset- und Taktnetze stellen völlig andere Anforderungen an den Verdrahtungsalgorithmus und werden üblicherweise in separaten Verarbeitungsschritten behandelt. Es wird eine zur Logikverdrahtung vollkommen konfliktfreie Lösung dieser im Modell unberücksichtigten Teile des Layouts angenommen, etwa durch Verwendung einer weiteren Verdrahtungsebene. Für die Verdrahtung der Logiknetze stehen zwei Ebenen zur Verfügung, je eine für vertikale und eine für horizontale Leitungselemente, ein Ebenenwechsel erfolgt durch ein Kontaktloch (Via). Alle Leitungen der Logikverdrahtung haben die gleiche Breite und der Abstand zweier direkt benachbarter Leitungen ist auf allen Verdrahtungsebenen identisch. Die möglichen Positionen für Leitungssegmente bilden ein diskretes äquidistantes Gitter. Das verwendete Gitterraster ist zugleich die Basiseinheit für Längenmessungen auf dem Chip. Vereinfachend wird angenommen, daß die Größe von Kontaktlöchern ebenfalls in dieses Raster paßt, so daß Kontaktlöcher auf direkt benachbarten Rasterpunkten möglich sind. Diese Definition der Verdrahtungsebenen und des Chipkoordinatensystems bedingt, daß nur Algorithmen, die mit diskreten ganzzahligen Koordinaten arbeiten können, verwendbar sind.

Die Verdrahtung erfolgt in ausgewiesenen horizontalen Kanälen zwischen den Zellreihen, sowie vier Kanälen entlang der Innenseite des Padkranzes. Over-the-Cell Routing ist nicht vorgesehen. Um Leitungen von einem horizontalen Kanal zum benachbarten legen zu können, bedarf es sogenannter Reihentransits oder Feedthroughs, die in die Zellreihen eingefügt werden und das Durchqueren der sonst für Leitungen gesperrten Zellreihen ermöglicht. Im einfachsten Falle sind diese Feedthroughs Lücken in den Zellreihen, in denen ein vertikales Leitungselement verläuft. Dies ist der in ESPRO gewählte Fall.

Die in ESPRO definierten Schnittstellen erlauben die Verwendung weiterer Verdrahtungsebenen und lassen das OTC-Routing prinzipiell zu, keiner der bisher implementierten Verdrahtungsalgorithmen hat davon jedoch Gebrauch gemacht (vergleiche Kapitel 7.3).

3.2 Aufbau des Systems

Das System ist im Rahmen mehrerer Studien- und Diplomarbeiten am Arbeitsbereich TECH des Fachbereichs Informatik der Universität Hamburg entstanden. Den Anfang machte eine Diplomarbeit ([Kla94]), die in Kooperation mit dem Fachbereich Mathematik entstand. Es folgten weitere Studien- und Diplomarbeiten ([Ger95], [Kah95], [Kas95],

[Jen97], [Kah97]), die weitere der vorgesehenen Module implementierten oder Alternativen für existierende Module darstellten.

Dieser Abschnitt stellt den Aufbau des Gesamtsystems vor und erläutert die Module so weit sie für diese Arbeit von Relevanz sind.

3.2.1 Systemüberblick und Modulbeschreibungen

Das Gesamtsystem besteht aus verschiedenen Modulen, die, wie in Abbildung 4 dargestellt, in Beziehung zu einander stehen. Die Schnittstellen zwischen den Modulen werden durch Dateien realisiert, die für diese Arbeit relevanten Schnittstellen sind in Abschnitt 3.2.2 erläutert. Die bisher implementierten Module sind für MS-DOS oder Linux geschrieben, durch die Wahl von Dateien als Schnittstelle ist auch jedes andere Betriebssystem als Plattform für neue Entwicklungen geeignet.

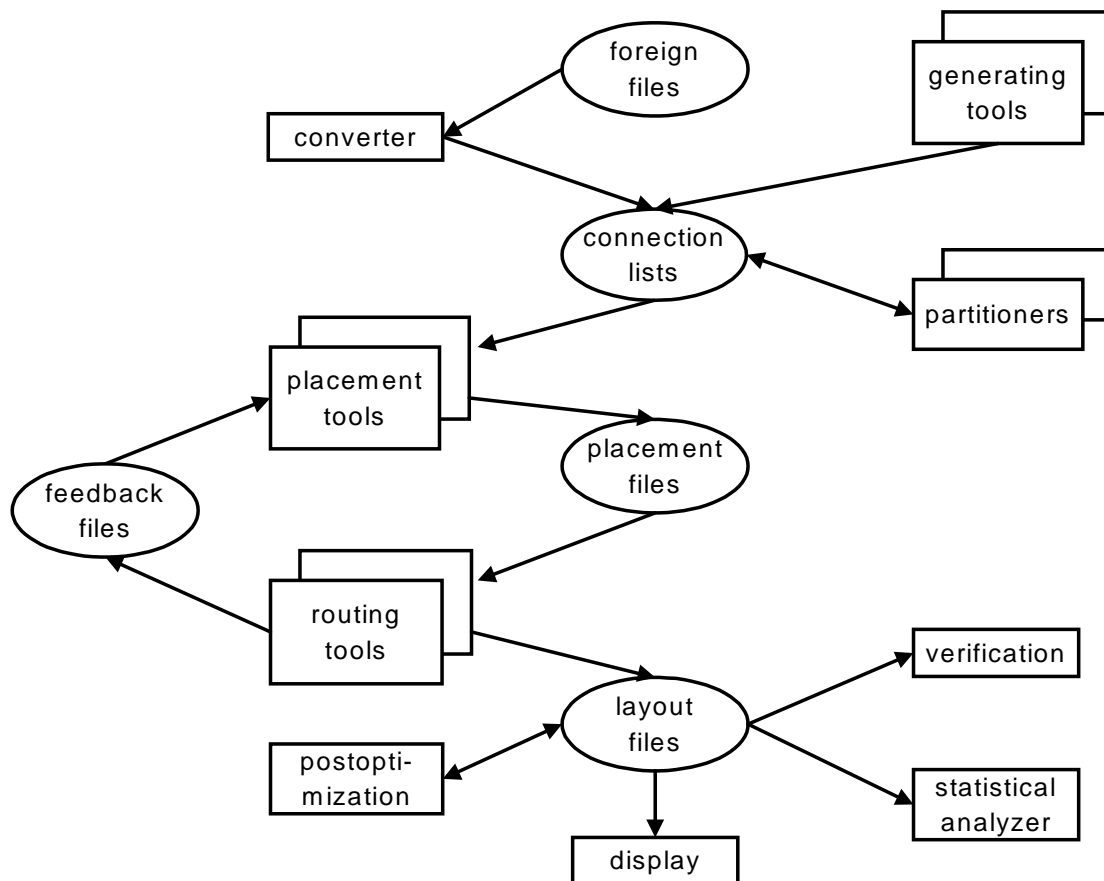


Abbildung 4: Die Module von ESPRO und ihre Beziehungen untereinander

Entsprechend der in Abbildung 4 dargestellten Struktur existieren die folgenden Arten von Modulen. Die in Klammern angegebenen Namen sind die Dateinamen der Programme für Linux (Kleinbuchstaben) bzw. DOS (Großbuchstaben).

Es existieren bisher zwei Module zur Erzeugung von Netzlisten im ESPRO-Format (Basisdateien, siehe Abschnitt 3.2.2). Das erste (key2chp) ermöglicht die manuelle Eingabe der gesamten Netzliste, damit ist es nur für extrem kleine Netzlisten mit weniger als 10 Zellen geeignet. Das zweite Modul (rnd2chp) erzeugt aus gegebenen Charakteristika eine zufällige Netzliste. Der Benutzer kann z.B. die maximale Pinanzahl

der Zellen, die Anzahl der Zellen und Netze etc. angeben. Beschreibungen zu beiden Programmen finden sich in [Kla94].

Zum Import existierender Netzlisten existiert ein Modul (EXL2CHP), welches Dateien aus dem Format für SOLO 1400 umwandelt. Dieses Format ist eine proprietäre Repräsentation von Netzlisten von European Silicon Structures Ltd. (siehe [ESS91]). Dieses Modul existiert nur für die MS-DOS Plattform und wird hier nur der Vollständigkeit halber erwähnt. Der Import standardisierter Netzlistenbeschreibungen, etwa das EDIF-Format wird noch nicht unterstützt.

Es existieren zwei Partitionierer, die in [Ger95] beschrieben sind und ebenso wie die Implementierung zweier Nachoptimierungsalgorithmen aus [Kas95] hier keine Anwendung finden.

Für die Auswertung und Beurteilung der erstellten Layouts stehen mehrere Module zu Verfügung. Ein Visualisierungsmodul (DISPLAY) stellt das Layout graphisch dar. Es ermöglicht Detailvergrößerungen und das Hervorheben bestimmter Netze. Dieses Modul existiert nur für MS-DOS, eine Beschreibung findet sich in [Kla94]. Ein Verifikationsmodul (extract) überprüft ein gegebenes Layout auf Korrektheit. Es werden z.B. Kurzschlüsse zwischen Netzen und unverbundene Leitungssegmente detektiert (siehe [Kah97]). Zwei Module (laystat und auswertung) werten die Layoutdateien statistisch aus, sie liefern Werte wie die durchschnittliche oder die maximale Netzlänge (siehe [Kas95] und [Kla94]).

Im Zentrum des Systems liegen die Platzierungs- und Verdrahtungsmodule. Von diesen sind die meisten unterschiedlichen Implementierungen vorhanden. Es existiert ein Programm (rsplacer), welches zufällige, aber korrekte, Platzierungen zu einer gegebenen Netzliste produziert (siehe [Kla94]). Diese können als Startplatzierungen für iterativ verbessernd arbeitende Platzierungsalgorithmen dienen. Zu den Platzierungsalgorithmen zählen bekannte Implementierungen wie „Timber Wolf“ (twplacer) und ein auf dem Min-Cut Verfahren basierender Plazierer (flplacer). Als weitere Plazierer existieren eine Implementierung des „Great Deluge“ Algorithmus (gdplacer, siehe [Kah95]), eine auf Fuzzy Logik basierende Heuristik (flplacer) und ein genetischer Algorithmus (gpplacer, siehe [Jen97]).

An Verdrahtungsalgorithmen existieren ein in [Kla94] vorgestelltes Verfahren (dkrouter), welches Global- und Detailverdrahtung in einem Schritt durchführt, sowie ein Programmpaket aus drei Teilen, welche die Global- und Detailverdrahtung getrennt durchführt [Kah95]: Der Globalverdrahter arbeitet auf Basis des Lee-Algorithmus (lee), die Reihenfolge, in der die Netze bearbeitet werden sollen, kann separat bestimmt werden (twrouter). Für die Kanalverdrahtung existiert eine Implementierung des Greedy Algorithmus (greedy).

Konzeptioniert als ein modulares offenes System können alle Programme unabhängig von einander verwendet werden. Dadurch ist z.B. eine Untersuchung der Einflüsse der Platzierung auf die Verdrahtung möglich, indem bei Verwendung des immer gleichen Verdrahters verschiedene Platzierungsmodule verwendet werden. Eine ausführliche Untersuchung hierzu ist in [RKB96] und [Rau00] erfolgt.

Die Modularisierung von ESPRO bringt es mit sich, daß Plazierer und Verdrahter nicht zwingend aufeinander abgestimmt sind. Dies ist im allgemeinen keine Schwäche, da dies durch geeignete Parametrisierung der Algorithmen leicht erfolgen kann. Unbeeinflußbar bleibt dagegen die vorgegebene Schnittstelle zwischen Plazierer und Verdrahter, und daß

der Verdrahter das Ergebnis des Plazierers als Startbedingung hat. Keiner der bisher in ESPRO implementierten Verdrahter hat direkten Einfluß auf die Platzierung. Die Parametrisierung der Algorithmen erfolgt bei Programmstart durch den Benutzer.

3.2.2 Verwendete Modulschnittstellen

ESPRO verwendet für die Kommunikation der Module untereinander verschiedene Dateien. Im Kontext dieser Arbeit sind davon drei Dateiformate relevant und sollen daher kurz erläutert werden. Es handelt sich dabei um die Basisdatei, die Platzierungsdatei und die Layoutdatei. Alle drei sind ASCII Dateien und somit, zumindest für kleine Beispiele, menschenlesbar.

Basisdatei

Die Basisdatei (Standardendung `.chp`) enthält eine Liste aller Zellen und Padzellen sowie die Liste der Netze. Sie definiert damit das zu lösende Platzierungs- und Verdrahtungsproblem.

Platzierungsdatei

Die Platzierungsdatei (Standardendung `.pla` oder `.plr`) enthält die Platzierungsinformationen. Dies beinhaltet die Größe des Chips, die Lage von Pads und Zellen sowie die Verteilung der Zellen auf die Zellreihen. Es sind keine Daten zur Netzliste enthalten, somit ist eine Platzierungsdatei nur mit der zugehörigen Basisdatei für einen Verdrahtungsalgorithmus verwendbar.

Layoutdatei

Die Layoutdatei (Standardendung `.lay`) enthält alle Informationen über die Lage sämtlicher Objekte auf dem Chip. Das sind Zellen, Pads, Leitungssegmente und Kontaktlöcher (Vias). Die Positionen der Zellen und Pads sind als Lage und Größe von Rechtecken auf dem Chip und nicht als Informationen über die Zellreihen und den Padkranz, wie in der Platzierungsdatei, enthalten. Die Liste sämtlicher Leitungselemente enthält für jedes Segment die Position, die Länge, den Layer und das Netz, zu dem es gehört. Die Aufzählung der Lage der Kontaktlöcher vervollständigt die Layoutbeschreibung. An statistischen Daten enthält die Layoutdatei zusätzlich die Länge jedes Netzes.

Über diese drei Dateitypen hinaus gibt es weitere von ESPRO-Modulen erzeugte Dateien. Diese sind entweder Ausgabedateien oder dienen der Kommunikation zwischen ESPRO-Modulen, die jedoch für diese Arbeit nicht relevant sind. Die Ausgabedateien sind der in ESPRO gewählte Weg zur Ausgabe von Ergebnissen. Sie enthalten in menschenlesbarer Form z.B. statistische Analysen, Fehlermeldungen oder Protokolle zum Programmablauf. Da diese Dateien nicht als Eingabe für weitere Module konzeptioniert sind, existiert für diese kein festes Format.

3.3 Änderungen der Aufgabe und Randbedingungen durch ESPRO

Die im vorigen Kapitel dargelegten Eigenschaften von ESPRO weichen merklich von den in Teilkapitel 2.5 angenommenen Eigenschaften eines Layouts ab. So sind z.B. die Zellen in ESPRO nicht als klein anzusehen und damit ist die Annahme, daß sich die Zellen in eine existierende (Global-)Verdrahtung ohne wesentliche Veränderungen an dieser einfügen

lassen, nicht gegeben. Da alle von bisherigen für ESPRO implementierten Algorithmen erzeugten Layouts mit nur zwei Verdrahtungsebenen ausgestattet sind und hinzukommend kein OTC-Routing verwendet wurde, stellen die Zellen durchaus Hindernisse für die Verdrahtung dar. Obwohl man sich durch weitere Verdrahtungsebenen und OTC-Routing den weiter oben genannten Voraussetzungen annähern könnte, ist dies für diese Arbeit keine Option, da damit die Vergleichbarkeit mit den existierenden Algorithmen nicht mehr gegeben wäre.

Trotzdem soll an ESPRO als Evaluationsumgebung festgehalten werden. Ein Grund dafür ist, daß es die einzige dem Autor verfügbare Plattform darstellt, die ohne wesentliche Vergrößerung des Implementierungsaufwandes einen Vergleich mit existierenden Algorithmen erlaubt. Hinzu kommt, daß die bereits in ESPRO implementierten Algorithmen unter denselben Bedingungen, als studentische Arbeit mit begrenzter Bearbeitungszeit, entstanden und damit in der Implementierungsqualität ebenbürtig sind. Ein Vergleich mit einer Implementierung, die von einem Team von Entwicklern durchgeführt wurde, wäre wesentlich weniger aussagekräftig. Es wird daher von einer Neudefinition des zugrundeliegenden Modells, was auch die Aufhebung der Beschränkung auf den Standardzellentwurf beinhalten könnte, abgesehen. Es werden nur die bereits in anderen ESPRO Implementationen eingesetzten Layoutelemente verwendet.

Diese, zugunsten der Vergleichbarkeit akzeptierten Einschränkungen, erstrecken sich auch auf die für die Bewertungen von Platzierung und Layout herangezogenen Kriterien. Es werden die bisher in ESPRO verwendeten Kriterien weiterverwendet. Dies sind der Flächenbedarf, die durchschnittliche und die maximale Netzlänge eines Layouts. Auf Basis dieser drei Eigenschaften wurde in [Rau00] ein Vergleich existierender ESPRO-Module durchgeführt. Die Ergebnisse aus [Rau00] dienen als Grundlage für die Bewertung des in dieser Arbeit erstellten Prototyps.

Die Kontaktlochanzahl und die Größe der kritischen Bereiche werden vor allem von der Detailverdrahtung bestimmt. Da die Detailverdrahtung in dieser Arbeit nicht näher betrachtet werden soll (vergleiche Abschnitt 4.1.2 und Abschnitt 4.3.4), werden diese beiden Merkmale nicht als Bewertungskriterien herangezogen.

Um sich bei dem Entwurf eines Lösungsweges nicht zu sehr auf den Standardzellentwurf oder ESPRO im Speziellen festlegen zu müssen, soll die hier vorgeschlagene Umsetzung so lange wie möglich nur solche Algorithmen verwenden und solche Modelle zugrunde legen, die nicht speziell auf den Standardzellentwurf zugeschnitten sind. Dadurch läßt sich eine Adaption des Vorgehens auf andere Entwurfsstile leichter bewerkstelligen.

Zusammenfassend bleibt zu bemerken, daß die in Kapitel 2.5 dargestellten Annahmen bezüglich der Layouteigenschaften von ESPRO nur teilweise erfüllt werden und daher viele der Eigenschaften der umzusetzenden Idee nicht in vollem Maße zum Tragen kommen werden. Die Resultate des Prototyps müssen immer auch unter diesem Gesichtspunkt bewertet werden. In Kapitel 7 werden Möglichkeiten aufgezeigt, wie in weiterführenden Arbeiten die in diesem Abschnitt dargelegten Restriktionen vermindert werden könnten.

4. DIE ALGORITHMISCHE UMSETZUNG

In diesem Kapitel wird der Algorithmus des in Kapitel 2.4 beschriebenen Konzeptes vorgestellt. Zunächst wird die Modellierung mittels Hypergraphen motiviert und beschrieben (Abschnitt 4.1.1), anschließend werden die sich ergebenden Teilprobleme in einem Überblick dargelegt (Abschnitt 4.1.2). Nach dieser Übersicht über die Gesamtproblemstellung widmet sich der Hauptteil dieses Kapitels den Teilschritten des hier verwendeten Vorgehens im Detail. Das hier dargelegte Vorgehen ist auf die in Kapitel 2.5 und Kapitel 3.3 präsentierten Randbedingungen und Anforderungen ausgelegt und spiegelt damit direkt den in dem erstellten Prototyp implementierten Ablauf wider, ohne auf Details der verwendeten Entwicklungs- und Laufzeitumgebungen einzugehen. Das allgemeine Vorgehen bei der Auswahl von Algorithmen für die Lösung der Teilprobleme erfolgte vor allem nach zwei Aspekten: Erstens sollen die Algorithmen leicht nachvollziehbar und die Charakteristik der von ihnen gelieferten Ergebnisse bekannt sein. Zweitens darf der Implementierungsaufwand nicht zu groß sein, um den Rahmen einer Prototypenimplementierung nicht zu sprengen.

Es ist daher nicht verwunderlich, daß für die Lösung der sich darbietenden Teilprobleme meist geringfügig modifizierte Verfahren zur Anwendung kommen, die aufgrund ihres Alters und Einfachheit als „nicht mehr dem Stand der Technik entsprechend“ eingestuft werden können. Für die hier gestellten Anforderungen, nämlich einen ersten Prototyp als Machbarkeitsbeleg zu implementieren, ist eben diese Einfachheit von Vorteil und ermöglicht es, den Fokus auf dem Gesamtverfahren zu behalten.

4.1 Modellierung und Überblick

Dieses Unterkapitel stellt zunächst eine erprobte Modellierung für den physikalischen VLSI-Entwurf mittels Graphen vor und entwickelt aus diesem Modell eine geeignete Darstellungsmethode für die Realisierung der umzusetzenden Idee. Im zweiten Teil dieses Abschnittes wird dann ein Überblick über den gesamten hier verwendeten Lösungsansatz gegeben.

4.1.1 Die Netzliste als Hypergraph

Die Aufgabe des physikalischen VLSI-Entwurfes besteht darin, eine günstige Platzierung und Verdrahtung für eine gegebene Netzliste zu finden. Günstig heißt, daß eine gegebene Zielfunktion zu minimieren (oder zu maximieren) ist. Die Zielfunktion enthält z.B. Komponenten zur Bewertung der durchschnittlichen sowie der maximalen Netzlänge und der benötigten Chipfläche. Im folgenden wird davon ausgegangen, daß die verwendeten Bewertungsfunktionen die Kosten repräsentieren und ein gutes Ergebnis daher durch eine Minimierung der Bewertungsfunktionen erreicht wird.

Die Netzliste eines VLSI-Entwurfes läßt sich als Multi-Hypergraph repräsentieren. Hier wird dieses Modellierungsmittel ebenfalls verwendet und soll daher kurz definiert werden:

Definition 1: Ein *Hypergraph* $G(V,E)$ besteht aus einer (endlichen) Knotenmenge $V \neq \emptyset$ und einer Kantenmenge $E \subseteq 2^V$, die aus Teilmengen von V besteht. Für jedes $e \in E$ gilt $|e| \geq 2$.

Definition 2: Ein *Multi-Hypergraph* $G(V,E)$ ist ein Hypergraph, in dem Mehrfachkanten zugelassen sind.

Im weiteren soll keine explizite Unterscheidung zwischen Hypergraphen und Multi-Hypergraphen getroffen werden. Alle folgenden Betrachtungen gelten für beide Typen von Hypergraphen gleichermaßen. Um ein möglichst allgemeingültiges Modell zu erhalten, ist die Verwendung von Multi-Hypergraphen angezeigt. Aus Gründen der sprachlichen Einfachheit, wird hier nur von Hypergraphen gesprochen werden.

Den Hyperkanten eines Hypergraphen kann, wie den Kanten eines Graphen auch, ein Kantengewicht zugeordnet sein. Dies geschieht durch eine Abbildung $w: E \rightarrow K$, wobei E die Kantenmenge von $G(V,E)$ und K die Menge der natürlichen, ganzen oder reellen Zahlen ist. Es sind auch Abbildungen auf Tupel, z.B. $w: E \rightarrow \mathbb{R} \times \mathbb{R}$ möglich. Die Abbildung auf Tupel ermöglicht es, die in der Zielfunktion eines den Hypergraphen verwendenden Algorithmus berücksichtigten Komponenten den Hyperkanten zuzuordnen. Die Komponenten des Tupels könnten z.B. für die Entfernung der Knoten voneinander und die Priorität des Netzes stehen.

Bei der Interpretation einer Netzliste als Hypergraph werden die Zellen als Knoten und die Netze als Hyperkanten repräsentiert. Der Vorteil der Modellierung durch Hypergraphen gegenüber der Modellierung mit Graphen ist, daß Hyperkanten, so wie die zu modellierenden Netze, mehr als genau zwei Knoten (Zellen) verbinden können. Der so entstehende Multi-Hypergraph ist im allgemeinen zusammenhängend und kann für die Betrachtung der Platzierungs- und Verdrahtungsprobleme als ungerichtet angesehen werden. Soll ein Einfügen von Leitungstreibern während der Verdrahtung erfolgen, so muß die Signalrichtung ebenfalls modelliert werden, dies kann separat erfolgen und soll hier nicht weiter betrachtet werden.

Netzliste für Schaltkreis: T4

4 Zellen: Z1: 3 Pins, Z2: 2 Pins, Z3: 4 Pins, Z4: 2 Pins
2 Pads: Pad1, Pad2
5 Netze: N1: Z1(0), Z4(0)
N2: Z1(2), Z2(0), Z4(1)
N3: Z2(1), Z3(3)
N4: Z1(1), Z2(0), Z3(1), Pad2
N5: Z3(2), Pad1

Abbildung 5: Die Netzliste T4 einer einfachen Schaltung mit vier Zellen, zwei Pads und fünf Netzen (vergleiche auch Abbildung 6).

Die Abbildung 5 zeigt das Beispiel einer Netzliste in einer der in ESPRO verwendeten Repräsentation ähnlichen Darstellung. Nach der Angabe des Namens der Netzliste (hier T4), folgen Listen mit den Zellen und Pads, einschließlich der Anzahl der Pins dieser Zellen. Die Liste der Netze gibt die zu erstellenden Verbindungen an. Für jedes Netz ist eine Liste der Pins aufgeführt, die es verbindet. Jeder Eintrag in der Liste eines Netzes spezifiziert einen Pin über seine Nummer und die zugehörige Zelle.

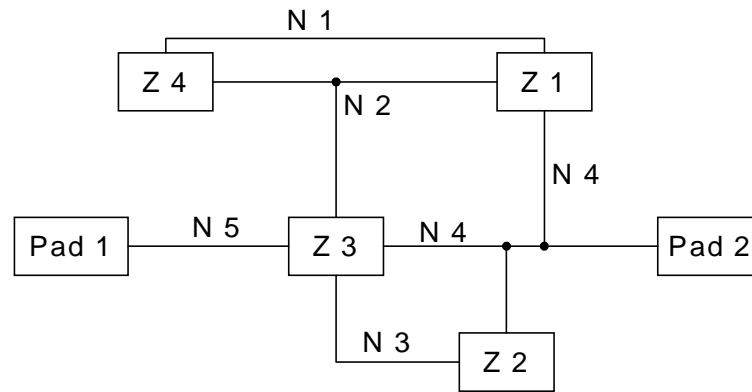


Abbildung 6: Die Netzliste T4 als Multi-Hypergraph $G_{T4}(V,E)$ dargestellt. Die Zellen bilden die Knoten, verbunden durch die Netze, repräsentiert durch die Hyperkanten

Abbildung 6 und Abbildung 7 zeigen als Beispiel Darstellungen für den Hypergraphen, der in Abbildung 5 gezeigten Netzliste T4 mit vier internen Zellen Z1 bis Z4 sowie zwei Padzellen Pad1 und Pad2. Diese insgesamt sechs Zellen sind durch die fünf Netze N1 bis N5 miteinander verbunden. Die Netze N1, N3, und N5 sind Zweipunktnetze, sie verbinden genau zwei Pins (verschiedener) Zellen miteinander. Das Netz N2 verbindet Pins der Zellen Z1, Z3 und Z4 und stellt ein Beispiel für ein Dreipunktnetz dar. Das Vierpunktnetz N4 verbindet Pins der Zellen Z1, Z2, Z3 und Pad2. Da die Netze N2 und N4 mehr als zwei Knoten verbinden, lassen sie sich nicht direkt durch die Kanten eines Graphen darstellen. Ein Hypergraph $G_{T4}(V,E)$ zur Modellierung der Netzliste T4 hat die Knotenmenge $V=\{Z1, Z2, Z3, Z4, Pad1, Pad2\}$ und die Kantenmenge $E=\{N1, N2, N3, N4, N5\}$ mit $N1=\{Z1, Z4\}$, $N2=\{Z1, Z3, Z4\}$, $N3=\{Z2, Z3\}$, $N4=\{Z1, Z2, Z3, Pad2\}$ und $N5=\{Z3, Pad1\}$.

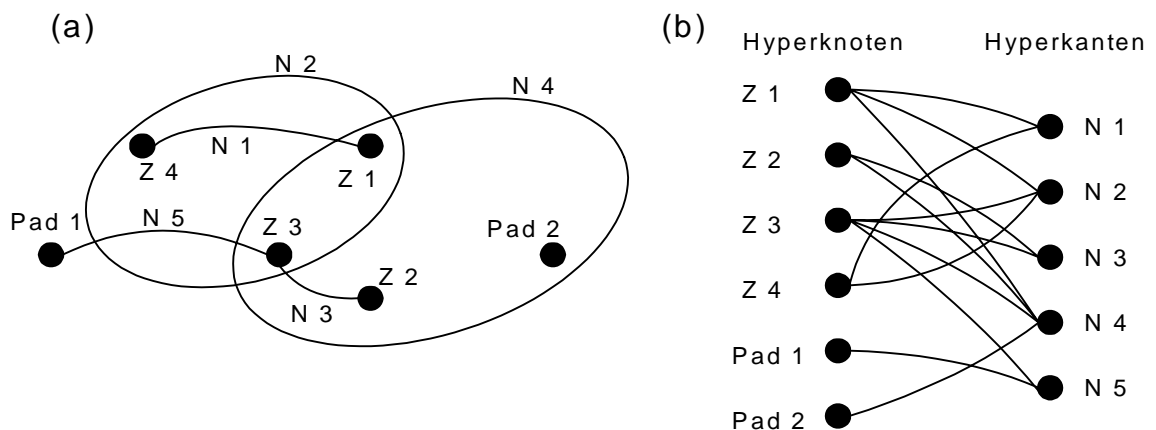


Abbildung 7: Zwei abstrakte Darstellungsformen für den Hypergraphen $G_{T4}(V,E)$. In Abbildung (a) sind die Kanten als Verbindungslinien zwischen den verbundenen Knoten oder als Ellipsen, die die verbundenen Knoten umschließen, repräsentiert. In Abbildung (b) wird der Hypergraph durch einen bipartiten Graphen beschrieben.

Die Abbildung 7 zeigt den gleichen Hypergraphen wie Abbildung 6, jedoch in zwei wesentlich abstrakteren Darstellungsformen. Während in Abbildung 6 eine dem Anwendungsgebiet entlehnte Symbolik verwendet wird, verzichten die abstrakten Darstellungen in Abbildung 7 auf jeglichen Anwendungsbezug.

Das Erstellen eines Layouts zu einer gegebenen Netzliste entspricht der Einbettung des Multi-Hypergraphen in die Ebene. Da der Multi-Hypergraph im allgemeinen nicht planar ist, sind hierbei Überschneidungen von Kanten zugelassen, ein Überlappen von Knoten (den Zellen) ist dagegen für eine korrekte Platzierung nicht zulässig. Heutige Algorithmen für den physikalischen VLSI-Entwurf führen diese Einbettung durch Anordnen der Knoten in der Ebene (Plazieren) und anschließendes Einfügen der Multi-Hyperkanten dazwischen (Verdrahten) durch. Überschneidungen von Kanten sind hierbei unkritisch, da die Verdrahtung auf mehreren Ebenen erfolgt.

Ausgehend von dieser im physikalischen VLSI-Entwurf etablierten Modellierung läßt sich ein Modell ableiten, welches die Darstellung der Netze als Knoten eines Hypergraphen zuläßt. Betrachtet man das Einbettungsproblem für Graphen, so existiert dort der duale Graph für planare Graphen [Len90]. Bei einem solchen planaren dualen Graphen werden die von den Kanten des in die Ebene eingebetteten planaren Graphen umschlossenen Gebiete durch Knoten repräsentiert, die durch Kanten mit den Knoten der Nachbargebiete verbunden sind. Abbildung 8 zeigt einen planaren Graphen und seinen dualen Graphen.

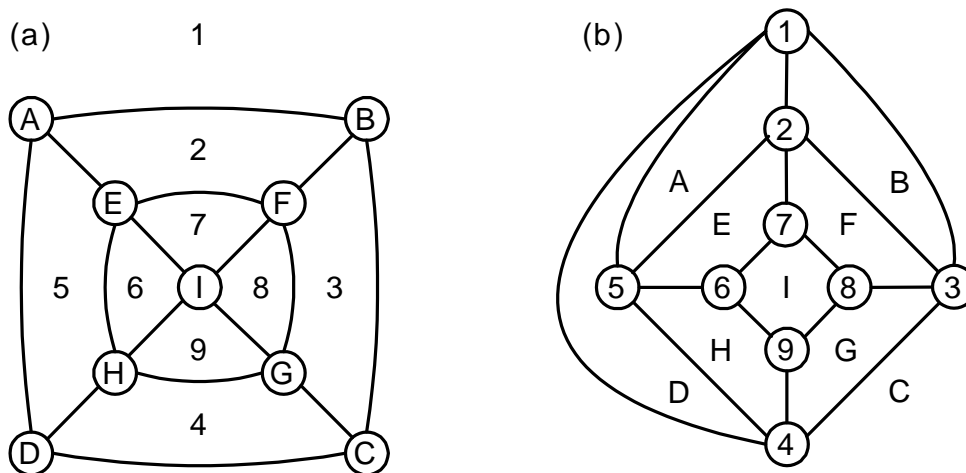


Abbildung 8: Ein planarer Graph (a) und der zugehörige planare duale Graph (b).

In dem dualen Graphen werden somit Flächen durch die Knoten dargestellt. Dieses Vorgehen zeigt damit die für die Modellierung der Netze als Knoten benötigte Eigenschaft. Das Konzept der dualen Graphen ist ein in der Graphentheorie bekanntes Konzept und soll hier ebenfalls verwendet werden. Der Begriff des dualen Hypergraphen ist etwas anders definiert, als der des dualen Graphen zu einem planaren Graphen und soll im folgenden dargelegt werden.

Bevor der duale Hypergraph eingeführt wird, ist es sinnvoll, eine wichtige Knoteneigenschaft zu erklären:

Definition 3: Der *Grad* eines Knoten v (oder *Knotengrad*) $\text{grad}(v)$ ist gegeben durch die Kardinalität der Teilmenge E_v der Knotenmenge E , die aus allen Hyperkanten besteht, die v enthalten: $\text{grad}(v) = |\{e \in E : v \in e\}|$.

Zunächst werden nur Hypergraphen $G(V,E)$ betrachtet, deren Knoten alle mindestens den Grad zwei haben ($\text{grad}(v) \geq 2 \forall v \in V$). Da diese Bedingung im allgemeinen von Hypergraphen, die eine Netzliste modellieren, nicht erfüllt ist, wird der Begriff des dualen

Hypergraphen anschließend so erweitert, daß auch Knoten mit Grad eins enthalten sein können.

Definition 4: Zu einem gegebenen Hypergraph $G(V, E)$ mit $\text{grad}(v) \geq 2 \forall v \in V$ heißt der Hypergraph $Y'(U_E, H_V)$ *dualer Hypergraph* genau dann, wenn $U_E = \text{node}(E)$ und $H_V = \text{edge}(V)$. Hierbei sind $\text{node}()$ und $\text{edge}()$ Bijektionen mit folgenden Eigenschaften: Die Abbildung $\text{node}: E \rightarrow U_E$ stellt eine Umbenennung dar. Die Abbildung $\text{edge}: V \rightarrow H_V$ liefert für jeden Knoten $v \in V$ die Teilmenge $E_v = \{e \in E : v \in e\}$ aller Hyperkanten, die v enthalten: $\text{edge}(v) = \{e \in E : v \in e\}$.

Informell gesprochen wird der duale Hypergraph dadurch gebildet, daß die Bedeutung von Knoten und Hyperkanten vertauscht wird. Die Kanten des Hypergraphen $G(V, E)$ werden zu den Knoten des dualen Hypergraphen $Y'(U_E, H_V)$. Analog werden die Knoten von $G(V, E)$ zu den Kanten von $Y'(U_E, H_V)$. Diese Neuinterpretation der beiden Mengen V und E ist möglich aufgrund der starken Ähnlichkeit der Kanten und Knoten in einem Hypergraphen, was ihre Beziehung zu Objekten der jeweils anderen Menge betrifft und zählt zu den grundlegenden Konzepten bei der Betrachtung von Hypergraphen [Ber89].

Wird der Hypergraph zur Darstellung einer Netzliste verwendet, so repräsentieren die Knoten des dualen Graphen nun die Netze und die Kanten die Zellen.

Abbildung 9 und Abbildung 10 zeigen einen Hypergraphen $G_1(V, E)$ zu einer Netzliste und dessen dualen Hypergraphen $Y'_1(U_E, H_V)$. Die Netzliste ist aus der Netzliste T4 (vergleiche Abbildung 5 und Abbildung 6) hervorgegangen, indem alle Knoten mit einem Grad von weniger als zwei entfernt und die Kantenmenge entsprechend angepaßt wurde.

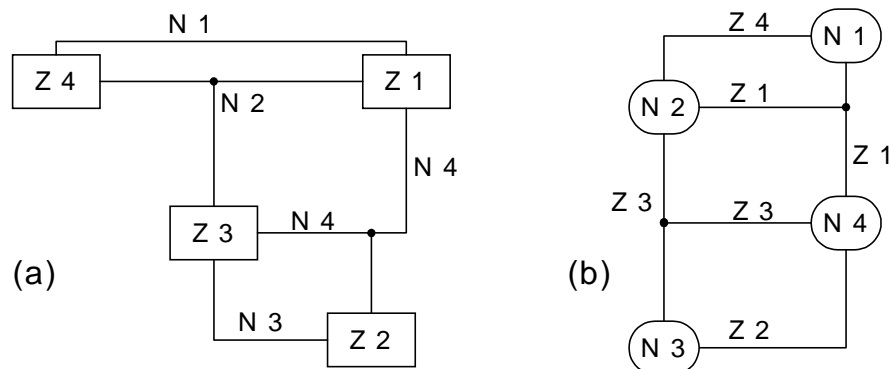


Abbildung 9: Die linke Darstellung (a) zeigt den Hypergraph $G_1(V, E)$ zu einer gegebenen Netzliste. Rechts (b) ist der nach Definition 4 zugehörige duale Hypergraph $Y'_1(U_E, H_V)$ dargestellt.

Abbildung 9 und Abbildung 10 verdeutlichen, daß die Nachbarschaftsbeziehungen im dualen Hypergraphen dieselben wie im originalen Hypergraphen sind. Die Zelle Z3 ist in $G_1(V, E)$ durch einen Knoten repräsentiert und durch die Kanten (Netze) N2, N3 und N4 mit ihren Nachbarzellen verbunden (vergleiche Abbildung 9.a). Im dualen Hypergraphen ist die Zelle Z3 durch eine Hyperkante repräsentiert, welche die Knoten N2, N3, und N4 verbindet, welche nun die entsprechenden Netze repräsentieren (vergleiche Abbildung 9.b).

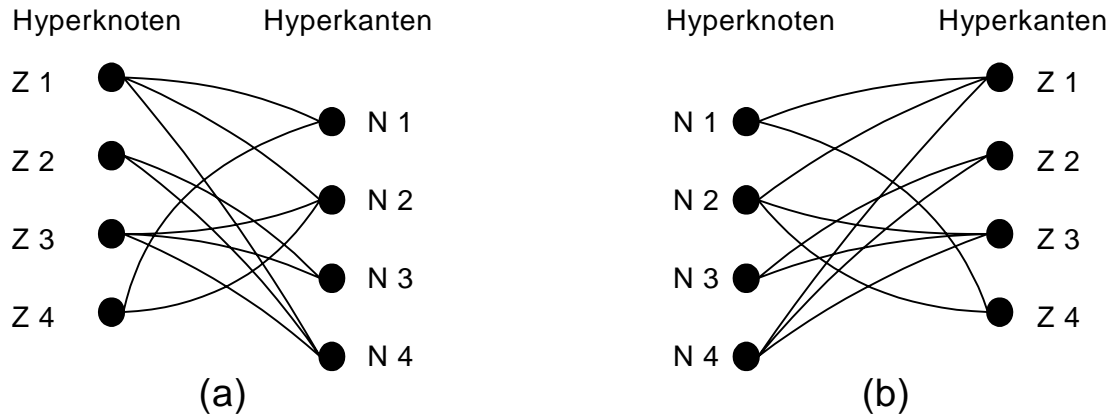


Abbildung 10: Die abstrakte Darstellung des Hypergraphen $G_1(V,E)$, wie in Abbildung 7.b, zeigt die Analogie von Hypergraph $G_1(V,E)$ und dualen Hypergraphen $Y'_1(U_E, H_V)$.

Der in Abbildung 9 und Abbildung 10 dargestellte Hypergraph $G_1(V,E)$ hat die Knotenmenge $V=\{Z1, Z2, Z3, Z4\}$ und die Kantenmenge $E=\{N1, N2, N3, N4\}$ mit $N1=\{Z1, Z4\}$, $N2=\{Z1, Z3, Z4\}$, $N3=\{Z2, Z3\}$ und $N4=\{Z1, Z2, Z3\}$. Die Kanten und Knoten des dualen Hypergraphen $Y'_1(U_E, H_V)$ werden ebenfalls durch die Symbole Z_i und N_j dargestellt, um die modellierten Elemente der zugrundeliegenden Netzliste leicht identifizieren zu können. Die Knotenmenge von $Y'_1(U_E, H_V)$ ist gegeben durch $U_E=\{N1, N2, N3, N4\}$, was einer Abbildung $\text{node}: E \rightarrow U_E$ mit $\text{node}(N_i)=N_i$ entspricht. Die Kantenmenge ist gegeben durch $H_V=\{Z1, Z2, Z3, Z4\}$ mit $Z1=\text{edge}(Z1)=\{N1, N2, N4\}$, $Z2=\text{edge}(Z2)=\{N3, N4\}$, $Z3=\text{edge}(Z3)=\{N2, N3, N4\}$ und $Z4=\text{edge}(Z4)=\{N1, N2\}$.

Wie durch einen Vergleich der Hypergraphen aus Abbildung 6 und Abbildung 9 ersichtlich wird, wurden genau die Knoten entfernt, welche die Padzellen der Netzliste repräsentieren. Padzellen sind genau die Zellen, die in einer Hypergraphrepräsentation einen Knotengrad von eins erhalten. Um Netzlisten komplett durch einen dualen Hypergraphen darstellen zu können, darf die Einschränkung auf Hypergraphen, deren Knoten mindestens einen Knotengrad von zwei haben, nicht bestehenbleiben. Für Hypergraphen mit Knoten, die alle mindestens den Grad eins haben ($\text{grad}(v) \geq 1 \forall v \in V$), wird daher die folgende Erweiterung durchgeführt, welche die Definition 4 ersetzt:

Definition 5: Der Hypergraph $Y(U,H)$ heißt *dualer Hypergraph* des Hypergraphen $G(V,E)$ mit $V = V_1 \cup V_P$, wobei $V_1 = \{v \in V : \text{grad}(v_i) \geq 2\}$ und $V_P = \{p \in V : \text{grad}(p)=1\}$ genau dann, wenn $U = U_E \cup U_P$ und $H = H_V \cup H_P$. Dabei sind $U_E = \text{node}(E)$ und $H_V = \text{edge}(V_1)$ wie in Definition 4 gegeben. Die Mengen U_P und H_P sind wie folgt definiert: $U_P = \{u_{pj} : p_j \in V_P, j=1..m\}$ und $H_P = \{ \{p_j, u_{pj}\} : p_j \in V_P, u_{pj} \in U_P, j=1..m\}$.

Auch diese Definition soll kurz informell beschrieben werden: Zu jedem Knoten p mit Grad 1 des Ausgangshypergraphen wird im dualen Hypergraphen ein neuer Knoten u_p erzeugt. Die Hyperkante zwischen diesem neuen Knoten u_p und dem Knoten des Netzes, an welches die Zelle des Ausgangsknoten p angeschlossen ist, repräsentiert im dualen Hypergraphen nun die zuvor durch den Knoten p dargestellte Zelle. In Hypergraphen, die Netzlisten repräsentieren, sind diese Knoten, wie oben erwähnt, die Knoten, welche Padzellen repräsentieren. Die neuen Knoten der Menge U_P im dualen Hypergraphen repräsentieren keine Zellen oder Netze aus der gegebenen Netzliste, sondern dienen zur

Darstellung der Padzellen als Kanten zwischen zwei Knoten. Dies entspricht der Aufgabe der Padzellen, ein Netz des Chips mit der Außenwelt zu verbinden. Diese zusätzlich erzeugten Knoten werden im folgenden als Repräsentanten externer Netze bezeichnet.

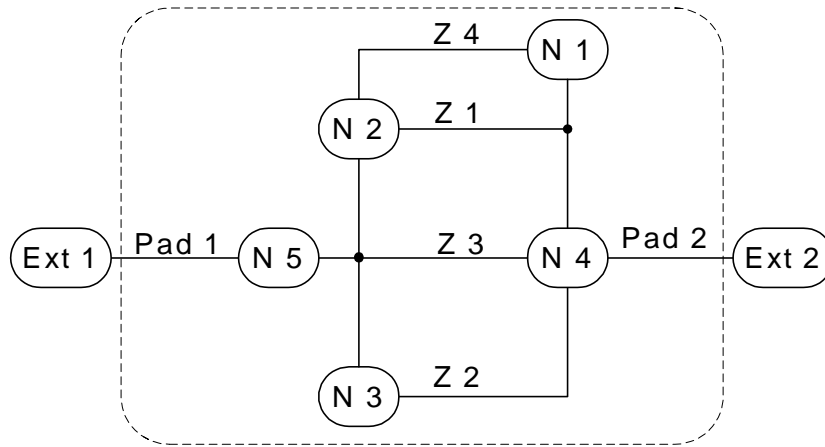


Abbildung 11: Der duale Hypergraph $Y_{T4}(U,H)$ des Hypergraphen $G_{T4}(V,E)$ aus Abbildung 6. Die Knoten repräsentieren die Netze, die Hyperkanten die Zellen. Innerhalb der gestrichelten Linie befinden sich die Knoten und Kanten, die eine Repräsentation in $G_{T4}(V,E)$ (und in der Netzliste T4) besitzen.

Die Abbildung 11 zeigt den dualen Hypergraphen $Y_{T4}(U,H)$ zu dem in Abbildung 6 gezeigten Hypergraphen $G_{T4}(V,E)$ zur Netzliste T4. Die Padzellen sind nun jeweils durch eine Hyperkante zwischen dem Knoten des Netzes, an das sie angeschlossen sind, und dem neu eingeführten Knoten (dem Repräsentanten des externen Netzes) dargestellt. Ein Vergleich von Abbildung 9 und Abbildung 11 zeigt deutlich, wie die in Definition 5 gegebene Erweiterung die Darstellung der Padzellen (welche jeweils den Knotengrad eins haben) ermöglicht. Die durch N1 mit den Zellen Z1, Z2 und Z3 verbundene Padzelle Pad2 wird in $Y_{T4}(U,H)$ durch die Hyperkante zwischen den Knoten N4 und dem neuen Knoten Ext2, der das an Pad2 angeschlossene externe Netz darstellt, repräsentiert.

Die Beschreibung des Hypergraphen $G_{T4}(V,E)$ aus Abbildung 6 wurde oben bereits gegeben, sie soll nun mit der Symbolik aus Definition 5 erneut gegeben werden. Die Knotenmenge $V = V_I \cup V_P$ besteht aus der Menge der Knoten mit Grad zwei oder mehr $V_I = \{Z1, Z2, Z3, Z4\}$ und der Menge der Knoten mit Grad eins $V_P = \{Pad1, Pad2\}$, welche die Padzellen repräsentieren. Die Kantenmenge bleibt unverändert dargestellt und ist gegeben durch $E = \{N1, N2, N3, N4, N5\}$ mit $N1 = \{Z1, Z4\}$, $N2 = \{Z1, Z3, Z4\}$, $N3 = \{Z2, Z3\}$, $N4 = \{Z1, Z2, Z3, Pad2\}$ und $N5 = \{Z3, Pad1\}$.

Der in Abbildung 11 gezeigte duale Hypergraph $Y_{T4}(U,H)$ besitzt die Knotenmenge $U = U_E \cup U_P$ und die Kantenmenge $H = H_V \cup H_P$. Die Knoten aus $U_E = \{N1, N2, N3, N4, N5\}$, welche je ein Netz repräsentieren, ergeben sich durch die Abbildung $node: E \rightarrow U_E$ mit $node(N_i) = N_i$. Die Hyperkanten zwischen diesen Knoten aus U_E sind gegeben durch die Menge $H_V = \{Z1, Z2, Z3, Z4\}$ mit $Z1 = edge(Z1) = \{N1, N2, N4\}$, $Z2 = edge(Z2) = \{N3, N4\}$, $Z3 = edge(Z3) = \{N2, N3, N4, N5\}$ und $Z4 = edge(Z4) = \{N1, N2\}$. Die Menge der neu eingeführten externen Netzrepräsentanten U_P ist gegeben durch $U_P = \{Ext1, Ext2\}$. Die Knoten der Menge V_P des ursprünglichen Hypergraphen $G_{T4}(V,E)$ werden nun durch die Hyperkanten der Menge $H_P = \{Pad1, Pad2\}$ repräsentiert, die jeweils den zu ihnen gehörenden externen Netzrepräsentanten und den Knoten des Netzes, an das die Padzelle in $G_{T4}(V,E)$ angeschlossen ist, verbinden.

Die Zelle Z3, welche in $G_{T4}(V,E)$ durch einen Knoten mit Grad vier modelliert wird, ist im dualen Hypergraphen durch eine Hyperkante die vier Knoten (N2, N3, N4 und N5) verbindet repräsentiert. Die Padzelle Pad1 ist in $Y_{T4}(U,H)$ durch eine Hyperkante zwischen dem Knoten Ext1 des externen Netzrepräsentanten und dem Knoten N5 dargestellt. Im ursprünglichen Hypergraphen $G_{T4}(V,E)$ hat der Knoten Pad1 nur eine Hyperkante N5, die ihn mit Z3 verbindet. Das Zweipunktnetz N5 selbst ist in $G_{T4}(V,E)$ durch die Hyperkante zwischen Pad1 und Z3 repräsentiert, in $Y_{T4}(U,H)$ durch den Knoten N5 mit dem Knotengrad zwei.

Analog zu $G_{T4}(V,E)$ kann auch der duale Multi-Hypergraph $Y_{T4}(U,H)$ in die Ebene eingebettet werden. Dies entspricht einem Plazieren der Netze und dem Verbinden der Netze durch die Zellen. Genau dieses Vorgehen bildet die Basis des im nächsten Abschnitt erläuterten Vorgehen zur Durchführung von Plazieren und Verdrahten.

Da es sich bei den in Definition 4 eingeführten Abbildungen $edge()$ und $node()$ um Bijektionen handelt, wird im folgenden auf eine explizite Unterscheidung von $u_i=node(e_i)$ und e_i (bzw. $h_i=edge(v_j)$ und v_j) verzichtet und die Mengen U_E und E (bzw. H und V) gleichgesetzt. Da im weiteren nur der in Definition 5 beschriebene duale Hypergraph $Y(U,H)$ verwendet werden wird, kann sogar die explizite Unterscheidung von $U=U_P \cup U_E$ und $U_E=node(E)$ entfallen, sofern die Bedeutung aus dem Kontext klar ist. Mit diesen sprachlichen Vereinfachungen wird U als die Knotenmenge des dualen Hypergraphen $Y(U,H)$ bezeichnet, die aus der Kantenmenge E des Hypergraphen $G(V,E)$ hervorgeht, ohne auf U_P und U_E näher eingehen zu müssen.

Das Modell des dualen Hypergraphen, wie er in Definition 5 eingeführt wurde, soll verwendet werden, um die in Kapitel 2.4 erwähnte Idee umzusetzen. Die resultierende Methode ist im folgenden Abschnitt beschrieben.

4.1.2 Überblick über das in dieser Arbeit verwendete Vorgehen

Die Aufgabe des Plazierens und Verdrahtens wird auch in dieser Arbeit in eben diese Teilprobleme aufgespalten und in dieser Reihenfolge bearbeitet. Der Plazierer jedoch ermittelt die Anordnung der Zellen aus der Lage der Netze. Hierfür wird eine Transformation der Betrachtung der Netzliste durchgeführt: Ausgehend von der Repräsentation der Netzliste als Hypergraph $G(V,E)$, wird der duale Hypergraph $Y(U,H)$ erstellt und bearbeitet. Die Knoten von $Y(U,H)$ werden nun in die Ebene eingebettet und damit die Netze plazierte. Unter Betrachtung beider Sichtweisen $G(V,E)$ und $Y(U,H)$ werden die Zellen eingefügt und ihre Platzierung verbessert. Die endgültige Platzierung der Zellen in Standardzellreihen wird dann wieder unter Betrachtung des ursprünglichen Hypergraphen $G(V,E)$ erstellt. Die Festlegung auf den Entwurfsstil des Standardzellentwurfes erfolgt erst nach Abschluß der Betrachtung des dualen Hypergraphen. Dadurch läßt sich das hier vorgestellte Verfahren leicht an andere Entwurfsstile adaptieren. Die Verdrahtung erfolgt nach der Platzierung, dabei ist es dem Verdrahter möglich, lokale Änderungen an der Platzierung vorzunehmen. Der Globalverdrahter verwendet Informationen aus beiden Repräsentationen $G(V,E)$ und $Y(U,H)$ der Netzliste. Die abschließende Kanalverdrahtung komplettiert das Layout. In Abbildung 12 ist der Ablauf dieses Vorgehens dargestellt.

0. Einlesen der Netzliste aus einer ESPRO Basisdatei und Repräsentieren als Hypergraph $G(V,E)$ und dualer Hypergraph $Y(U,H)$
1. Plazieren:
 - a. Plazieren der Netze unter Betrachtung des dualen Problems $Y(U,H)$.
 - b. Einfügen der Zellen (betrachtet $Y(U,H)$ und $G(V,E)$)
 - c. Kräftegesteuerte Verbesserung der Platzierung (betrachtet $Y(U,H)$ und $G(V,E)$)
 - d. Erzeugen einer gültigen Standardzellplatzierung
 - e. Speichern der Platzierung in einer ESPRO Platzierungsdatei
2. Verdrahten:
 - a. Zellweise Globalverdrahtung mit lokalen Modifikationen der Platzierung
 - b. Überprüfung des Zusammenhangs aller Netze
 - c. Kanalverdrahtung und Adaption der Kanalbreiten
 - d. Speichern der modifizierten Platzierung in einer neuen ESPRO Platzierungsdatei
 - e. Speichern des erzeugten Layouts in einer ESPRO Layoutdatei

Abbildung 12: Ablauf des als Prototyp realisierten Verfahrens.

Es folgt ein Überblick über die zu lösenden Teilprobleme (vergleiche Abbildung 12), um von der in einer ESPRO-Basisdatei gegebenen Netzliste zu einem gültigen Standardzellenlayout für ESPRO zu gelangen. Anhand des Beispiels der in Abschnitt 4.1.1 vorgestellten Netzliste T4 werden im folgenden die Stadien bis zum vollständigen Layout illustriert. Die durchzuführenden Teilschritte sind jeweils in einem eigenen Abschnitt der folgenden Teilkapitel 4.2 und 4.3 näher dargelegt.

Kapitel 4 Die algorithmische Umsetzung

Zuerst wird die Netzliste eingelesen und intern derart repräsentiert, daß die Interpretation als Hypergraph $G(V,E)$ und als dualer Hypergraph $Y(U,H)$ möglich ist. Für die Netzliste T4 (vergleiche Abbildung 5) ergeben sich die in Abbildung 13 gezeigten Hypergraphen $G_{T4}(V,E)$ und $Y_{T4}(U,H)$.

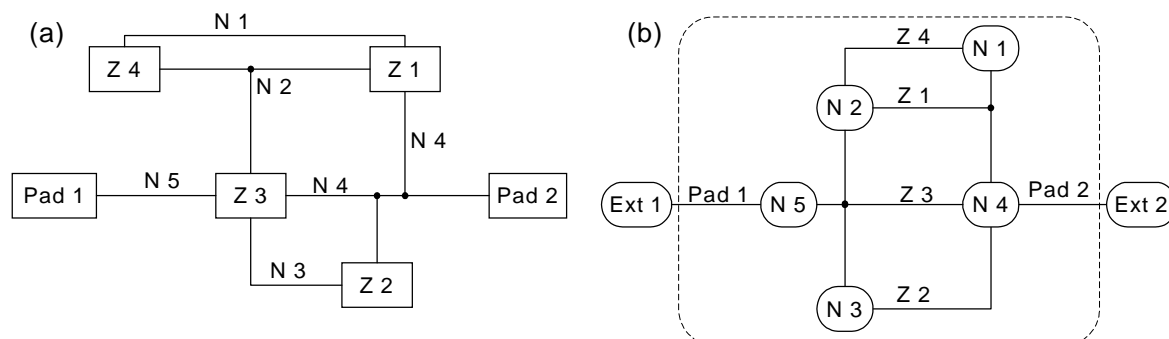


Abbildung 13: Die Netzliste T4 in (a) dargestellt als Hypergraph $G_{T4}(V,E)$ und in (b) als dualer Hypergraph $Y_{T4}(U,H)$.

Für die Erstellung der Platzierung werden zunächst, den dualen Graphen $Y(U,H)$ betrachtend, die Netze als punktförmige Objekte in der Ebene platziert. Dabei repräsentiert ein Knoten die Fläche, über den sich das zugehörige Netz erstreckt. Diese Fläche ist gegeben durch das n -Eck, welches die konvexe Hülle der verbundenen Pins umschließt (vergleiche Abbildung 15). Die Lage dieses, im folgenden als Netzrepräsentant bezeichneten Knoten, in der Ebene ist gegeben durch den Ort, an dem sich der Massenmittelpunkt eines Systems punktförmiger Objekte (gleicher Masse) befinden würde, welche an den Positionen der Pins liegen. Im weiteren wird dieser Ort als Netzmittelpunkt bezeichnet. Das Platzieren der Netze kann unter Verwendung bekannter Algorithmen geschehen, als Bewertungskriterium für die Güte der Lage eines Netzes kann z.B. der Abstand verbundener Netze in der Euklidischen oder der Manhattan-Metrik dienen. In dieser Arbeit wird, sofern nicht explizit anders erwähnt, mit der Manhattan-Metrik gearbeitet. Die Abbildung 14 zeigt eine mögliche Lage der Netze von T4, wie sie durch das in Abschnitt 4.2.1 näher beschriebene Platzieren der Netze entsteht.

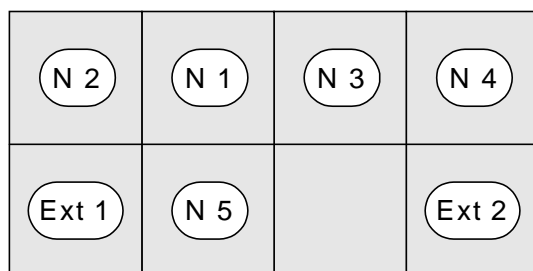


Abbildung 14: Die Netze von T4 werden in der Ebene platziert.

Anschließend werden die Zellen eingefügt. Hierbei bestimmen die Koordinaten der Netzrepräsentanten die empfohlenen Positionen für die einzelnen Zellen. Eine Zelle kommt zwischen den Netzrepräsentanten der mit ihr verbundenen Netze zu liegen (vergleiche Abbildung 16).

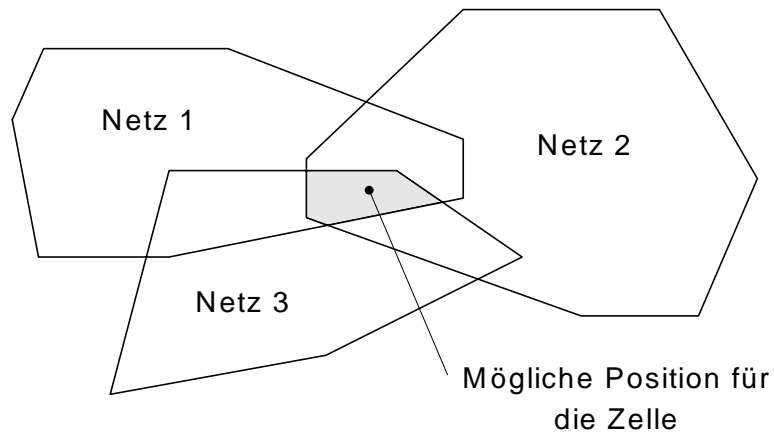


Abbildung 15: Das grau unterlegte Gebiet gibt die Region an, welche die, für eine an die gezeigten Netze angeschlossene Zelle, günstigen Positionen enthält.

Wie in Abbildung 15 dargestellt, befindet sich die ideale Position für eine Zelle in der von allen angeschlossenen Netzen überdeckten Fläche. In Abschnitt 4.2.2 ist das Vorgehen des Zelleinfügens näher beschrieben. Für die Netzliste T4 ergibt sich die in Abbildung 16 gezeigte Lage der eingefügten Zellen.

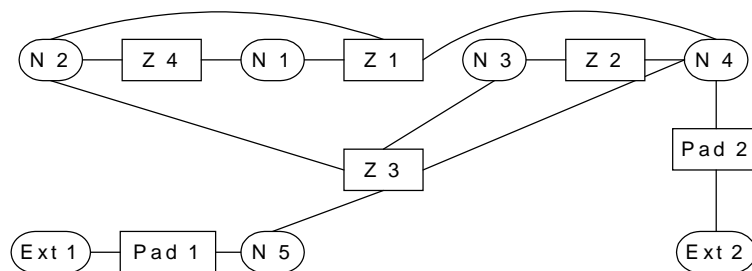


Abbildung 16: Die Zellen aus T4 werden zwischen den an sie angeschlossenen Netzen eingefügt.

Unter Verwendung beider Repräsentationen $G_{T4}(V,E)$ und $Y_{T4}(U,H)$ wird anschließend die Anordnung der Zellen in einem iterativen Verfahren verbessert, welches in jeder Iteration zunächst die Lage der Netzrepräsentanten und anschließend die Positionen der Zellen ändert. Die Abbildung 17 zeigt Anordnung von Zellen und Netzen nach der ersten Iteration des in Abschnitt 4.2.3 näher beschriebenen Verfahrens zur Verbesserung der Zellpositionen am Beispiel der Netzliste T4.

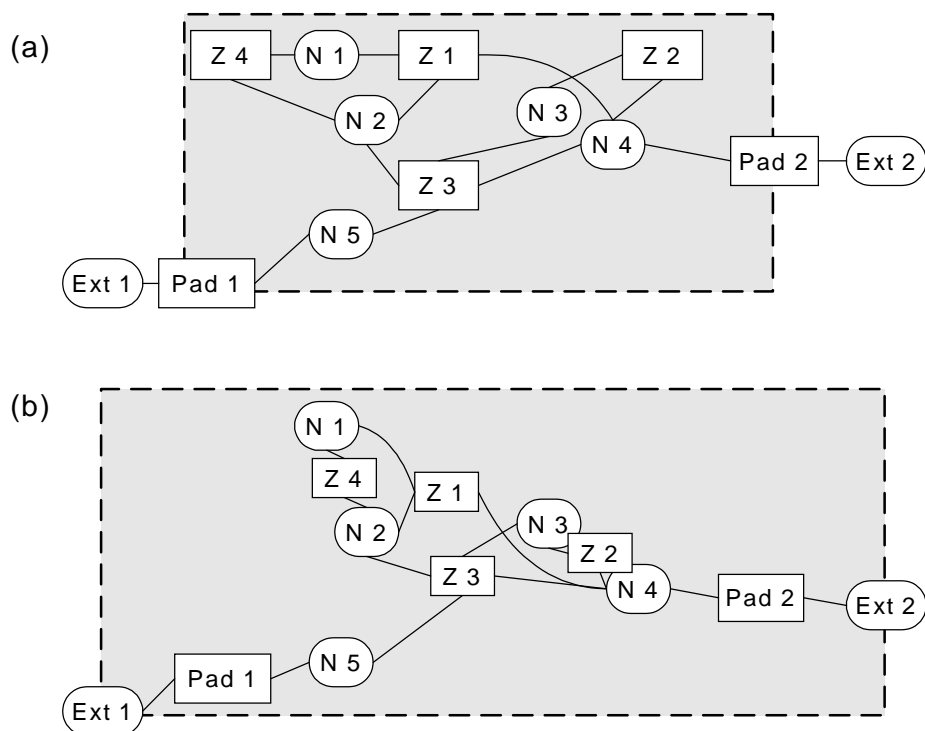


Abbildung 17: Eine Iteration zur Verbesserung der Zellanordnung für die Netzliste T4. In (a) wird die Position der Netzrepräsentanten verändert, in (b) die Position der Zellen auf Basis der in (a) generierten neuen Anordnung der Netzrepräsentanten.

Die so gewonnene Platzierung (vergleiche Abbildung 17) von dimensionslosen Zellen und Netzrepräsentanten wird im weiteren als Vorplatzierung bezeichnet und ist nun an den gewünschten Entwurfstil anzupassen. Im Falle des hier verwendeten Standardzellentwurfs sind spätestens an dieser Stelle für die Verdrahtung erforderlichen Feedthroughs Freiräume in den Zellreihen in den Zellreihen vorzusehen. Die bis zu diesem Punkt erstellte Platzierung wird als Zwischenergebnis in eine ESPRO Platzierungsdatei ausgegeben. In Abschnitt 4.2.4 ist das Verfahren zum Erstellen einer Standardzellplatzierung aus der Anordnung der Zellen beschrieben. Die Abbildung 18 zeigt die Platzierung zur Netzliste T4, wie sie aus der in Abbildung 17.b gezeigten Zellanordnung abgeleitet wird.

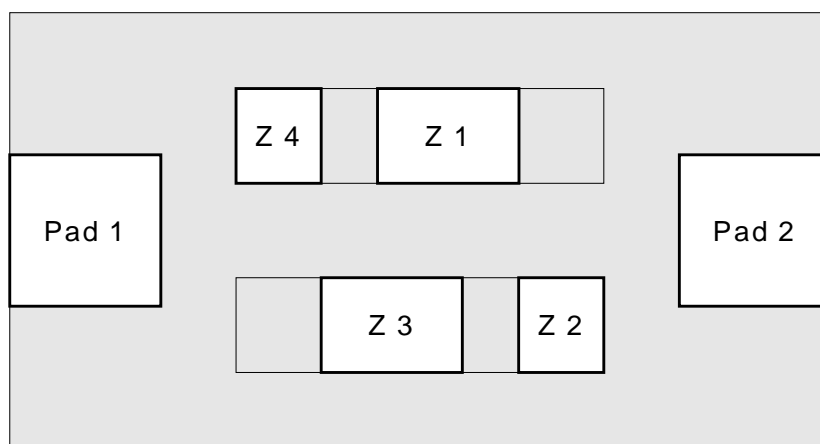


Abbildung 18: Die zur Netzliste T4 erzeugte Standardzellplatzierung.

Ausgehend von der gewonnenen Standardzellplatzierung (vergleiche Abbildung 18) beginnt die Verdrahtung. Es ist auch möglich, eine von einem anderen Algorithmus

generierte Platzierung aus einer ESPRO Platzierungsdatei einzulesen und den Verdrahter so unabhängig vom hier präsentierten Platzierer zu verwenden.

Die Verdrahtung ist in die Schritte Globalverdrahtung und Kanalverdrahtung aufgeteilt. Die Globalverdrahtung erfolgt, im Gegensatz zum üblichen Vorgehen, zellweise sequentiell. Dies spiegelt das Verdrahtungsproblem aus der Perspektive des dualen Graphen $Y(U,H)$ wider. So wie im ursprünglichen Graphen $G(V,E)$ die Netze die Kantenmenge bilden, und in bisherigen Algorithmen sukzessive verdrahtet werden, so bilden in $Y(U,H)$ die Zellen die Kantenmenge und damit die zu bearbeitenden Objekte. Dies ist ein Ansatz, um die bei der netzweisen Verdrahtung auftretende Benachteiligung von spät verdrahteten Netzen zu mildern (vergleiche [She95] und [She89]). Werden die Netze als Ganzes verdrahtet, so sind spät betrachtete Netze benachteiligt, weil weniger Verdrahtungsressourcen zur Verfügung stehen und deshalb mit höherer Wahrscheinlichkeit für einen möglichen Verdrahtungsweg Umwege in Kauf genommen werden müssen. Beim zellweisen Verdrahten werden immer nur Teilnetze verdrahtet, die spät verdrahteten Zellen werden ebenso benachteiligt wie zuvor die spät verdrahteten Netze, jedoch verteilt sich der Effekt der begrenzten Verdrahtungsressourcen auf mehrere Netze, eben diejenigen, an die die Zelle angeschlossen ist.

Während des Globalverdrahtens werden die Feedthroughzellen eingefügt und so die in der Platzierung vorgesehenen anonymen Lücken in den Zellreihen jeweils fest einem Netz zugeordnet. Durch das zellweise Vorgehen kann es nicht ausgeschlossen werden, daß einige Netze zwischenzeitlich in zwei oder mehr unverbundene Teilnetze zerfallen. Eine Überprüfung der Netze und gegebenenfalls das Einfügen weiterer benötigter Feedthroughs vor der Kanalverdrahtung garantiert den Netzzusammenhang. Die Globalverdrahtung (mit Ausnahme der Padzellen), sowie das Allokieren der benötigten Feedthroughs ist in Abschnitt 4.3.1 dargelegt. Die Abbildung 19 zeigt die Platzierung zur Netzliste T4 mit allokierten Feedthroughs und der Zuordnung der Netze zu den Verdrahtungskanälen.

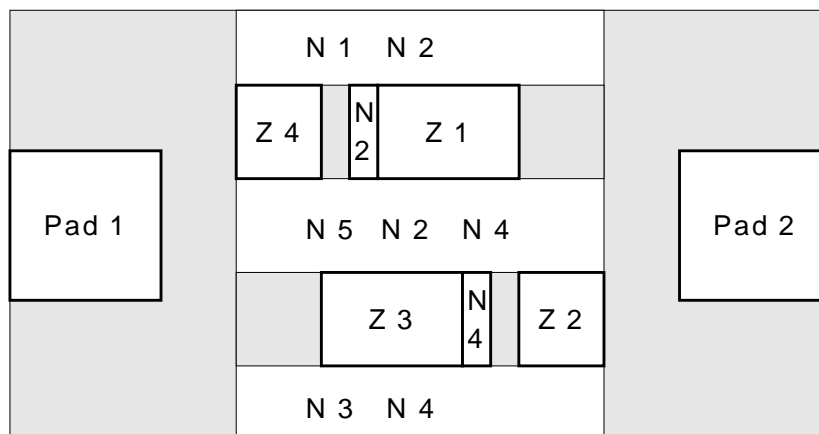


Abbildung 19: Nach der Globalverdrahtung ist die Zuordnung der Feedthroughs zu Netzen erfolgt und festgelegt, durch welche Verdrahtungskanäle die Netze verlaufen.

Die Globalverdrahtung der Padzellen erfolgt nach Abschluß der Überprüfung des Netzzusammenhangs (vergleiche Abschnitt 4.3.1.6). Hierbei werden die gleichen Kriterien wie bei der Globalverdrahtung verwendet. Das verwendete Vorgehen hierfür ist in Abschnitt 4.3.3 dargelegt. Die Abbildung 20 zeigt die komplette Globalverdrahtung, einschließlich der Teilnetze zu den Padzellen für die Netzliste T4.

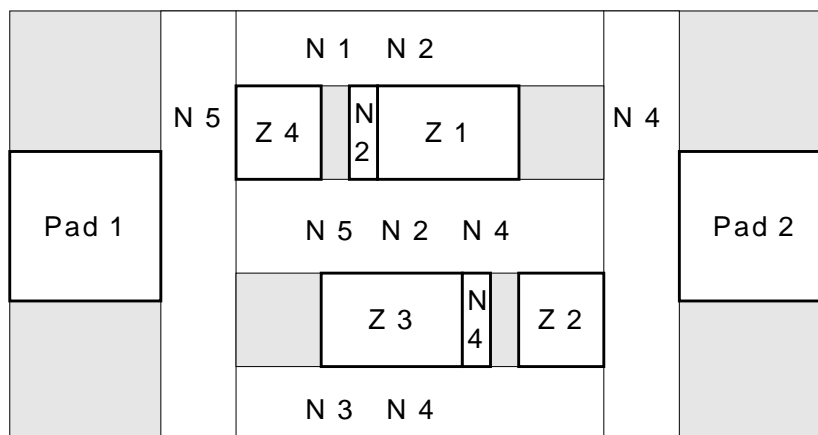


Abbildung 20: Die Globalverdrahtung der Teilnetze zur Verbindung der Padzellen erfolgt in einem separaten Schritt und schließt die Globalverdrahtung ab.

Der letzte Schritt besteht aus der Kanalverdrahtung. Das Problem der Kanalverdrahtung ist wohlverstanden und wird als gelöst betrachtet [She95]. Es ist daher nicht Teil der Untersuchung dieser Arbeit und wird mit einem erprobten Kanalverdrahtungsalgorithmus gelöst (vergleiche Abschnitt 4.3.4). Die Abbildung 21 zeigt das fertige Layout für die Netzliste T4 nach abgeschlossener Kanalverdrahtung.

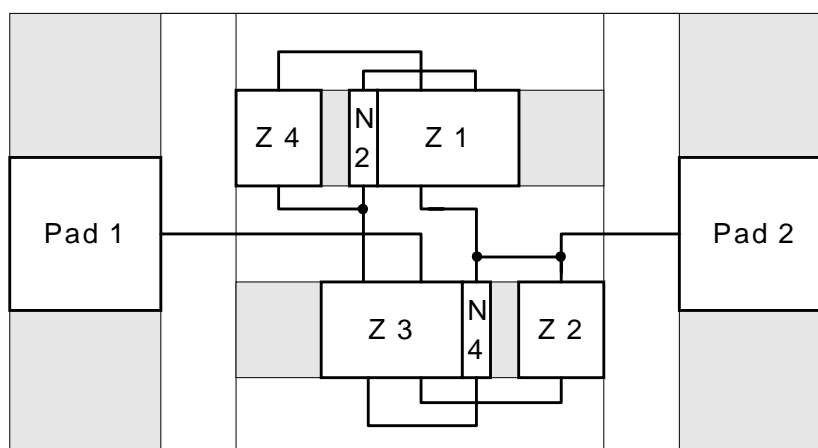


Abbildung 21: Das fertige Layout für die Netzliste T4 nach Abschluß der Kanalverdrahtung.

Das entstehende Layout wird in einer ESPRO Layoutdatei ausgegeben, außerdem werden statistische Werte des Layouts ermittelt sowie Informationen über während der Programmausführung vorgenommene Aktionen (z.B. die Anzahl der vom Plazierer vorgesehenen Feedthroughs) ausgegeben.

Es ergibt sich der in Abbildung 12 gezeigte Ablauf für die Erstellung eines ESPRO Layouts aus einer gegebenen Netzliste. Die folgenden Teilkapitel 4.2 und 4.3 legen die Teilaufgaben im Detail dar.

4.2 Details zu Teilproblemen von der Netzliste bis zur fertigen Plazierung

Dieses Teilkapitel widmet sich den durchzuführenden Ablaufschritten, um ausgehend von einer gegebenen Netzliste eine gültige Standardzellplazierung zu erhalten. Ohne auf Implementationsdetails einzugehen, sollen die verwendeten Algorithmen beschrieben und

die jeweils getroffene Auswahl motiviert werden. Die folgenden Abschnitte beschreiben je einen der vier Schritte zur Erzeugung einer gültigen Standardzellplazierung, die in Abschnitt 4.1.2 und Abbildung 12 skizziert wurden.

4.2.1 Positionierung der Netze

In diesem ersten Schritt wird das durch den dualen Hypüergraphen $Y(U,H)$ gegebene Einbettungsproblem betrachtet. Die Knoten von $Y(U,H)$ sind geeignet so in der Ebene anzuordnen, daß das Einfügen der Kanten aus H mit möglichst geringen Kosten erfolgen kann. Die Ebene ist dabei in ihrer Ausdehnung nicht beschränkt, insbesondere ist die Größe des resultierenden Chips hiervon vollkommen unabhängig. Als Kostenfunktion kann die Distanz zwischen den verbundenen Knoten dienen, oder die Anzahl der in einem bestimmten Flächensegment verlaufenden Kanten und damit die Auslastung der verfügbaren Verdrahtungsregionen. Für die Kostenfunktion können alle Kriterien herangezogen werden, die sonst bei der Plazierung von Zellen, also bei der Betrachtung des durch $G(V,E)$ gegebenen Problems, Anwendung finden.

Beide Plazierungsprobleme $G(V,E)$ und $Y(U,H)$ stellen vergleichbare Anforderungen an den zu ihrer Lösung verwendeten Algorithmus. Als Beispiel sei hier die Netzlänge genannt. Um für das durch $G(V,E)$ gegebene Problem möglichst kurze Netze zu erhalten, sollten alle Zellen, die an das gleiche Netz angeschlossen sind, möglichst dicht benachbart sein. Da jede Zelle an mehrere Netze angeschlossen ist, muß immer eine Position der Zellen gefunden werden, die keines der beteiligten Netze zu stark benachteiligt.

Aus der Sicht des dualen Problems $Y(U,H)$ läßt sich die Zielsetzung einer geringen Netzlänge wie folgt darlegen: Die Länge eines Netzes wird von der Größe der von ihm überdeckten Fläche bestimmt. Je größer die Fläche des Gebiets ist, über das sich ein Netz erstreckt, um so länger sind auch die Leitungen, die es bilden. Der Rand des von einem Netz belegten Gebietes wird durch die Lage der Pins definiert, die durch das Netz verbunden werden (vergleiche Abschnitt 4.1.2). Die vom Netz überdeckte Fläche ist folglich um so kleiner, je dichter die Pins und damit die betroffenen Zellen beieinander liegen. Durch eine Anordnung der Knoten von $Y(U,H)$ in der Ebene, welche die durch eine Hyperkante verbundenen Knoten möglichst dicht nebeneinander plaziert, wird genau dies erreicht.

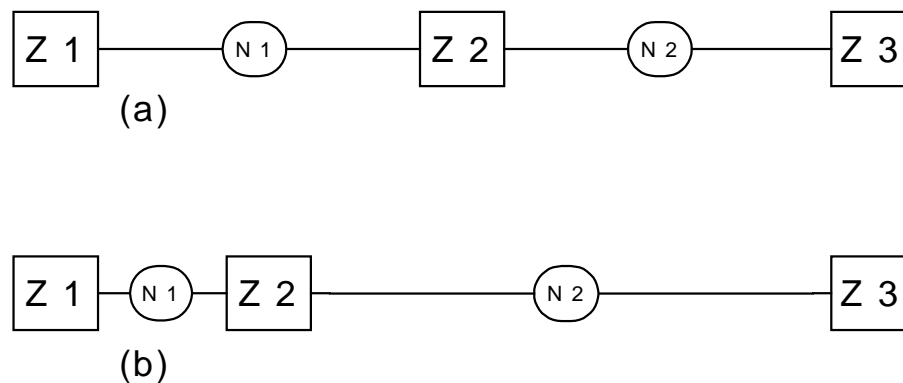


Abbildung 22: Die Lage der Zellen definiert die Position der Netzgrenze. Zelle Z2 trennt die Netze N1 und N2, einmal etwa gleichmäßig (a) und einmal wird N1 auf Kosten von N2 bevorzugt (b).

Obiger Zusammenhang soll an einer Worst-Case Betrachtung für die Netzlänge eines Netzes verdeutlicht werden: Die Grenze eines Netzes ist zunächst entlang der Kanten von $Y(U,H)$ beweglich und wird erst durch die noch zu bestimmende Position der verbindenden Zelle bestimmt (siehe Abbildung 22). Im ungünstigsten Fall für ein betrachtetes Netz liegen, wie in Abbildung 23.a für N1 zu sehen ist, alle Netzgrenzen direkt bei den Knoten der benachbarten Netzrepräsentanten. Die von einem Netz maximal eingenommene Fläche ist somit durch die Position der benachbarten Netzrepräsentanten begrenzt. Eine dicht benachbarte Anordnung von durch eine Hyperkante verbundenen Netzen reduziert folglich die Fläche, die diese Netze (selbst im ungünstigsten Fall) beanspruchen können. Da die Zellen nur in Ausnahmefällen so angeordnet werden, daß ein Netz deutlich bevorzugt (und damit mindestens ein anderes ebenso deutlich benachteiligt) wird, werden sich die Netzgrenzen im allgemeinen nicht direkt am Nachbarknoten, sondern mehr oder weniger in der Mitte der Kante befinden.

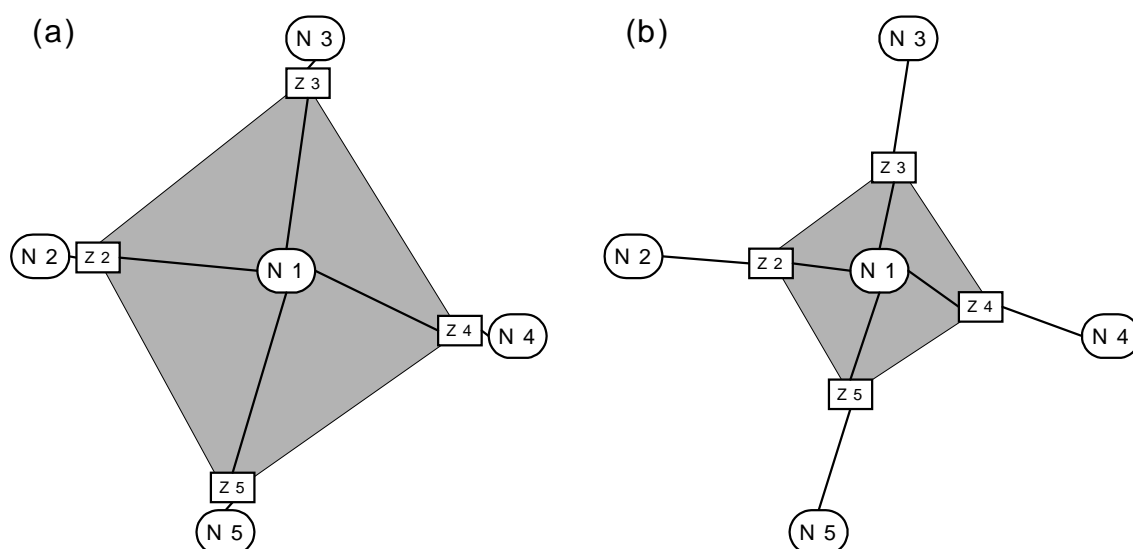


Abbildung 23: Die Größe des von Netz N1 überdeckten Gebietes wird durch die Lage der begrenzenden Pins (und damit der Zellen) bestimmt. In (a) ist der Worst-Case für die Netzgröße und in (b) ein Beispiel für eine wesentlich wahrscheinlicher auftretende Ausdehnung des Netzes gezeigt.

Die Netzlängen aller Netze werden somit durch enge Nachbarschaft verbundener Netze reduziert. Wird weiter davon ausgegangen, daß die Chipgröße wesentlich von dem Bedarf an Verdrahtungsressourcen bestimmt wird, so reduziert eine günstige Anordnung der Knoten von $Y(U,H)$ ebenfalls die benötigte Chipfläche.

Eine günstige Anordnung der Knoten von $Y(U,H)$ läßt sich durch Betrachtung analoger Kriterien, wie sie bei $G(V,E)$ zur Anwendung kommen, erreichen. Es lassen sich prinzipiell die gleichen Platzierungsalgorithmen anwenden, wie sie für die Platzierung von Zellen benutzt werden. Die einzige Einschränkung ist, daß sich die Bewertungskriterien geeignet an den veränderten Blickwinkel von $Y(U,H)$ anpassen lassen müssen.

Für die Entscheidung, welches Platzierungsverfahren zu verwenden ist, um die Repräsentanten der Netze zu positionieren, wurden die folgenden drei Gesichtspunkte berücksichtigt: Es sollte sich um einen verbreiteten Algorithmus handeln, dessen Stärken und Schwächen wohlverstandenen sind, die Kostenfunktion sollte sich leicht auf das duale Problem übertragen lassen und der Implementierungsaufwand sollte möglichst gering sein.

4.2.1.1 Der verwendete Plazierungsalgorithmus

Es kommt ein auf hierarchischer Partitionierung basierender Plazierungsalgorithmus zum Einsatz. Dabei wird der Chip schrittweise in immer kleinere Gebiete (Kacheln) zerlegt und die Knoten von $Y(U,H)$ (die Netze) diesen Kacheln zugeordnet. Ausgehend von der gesamten zur Verfügung stehenden Fläche, werden bei jeder Iteration die bisher entstandenen Gebiete (Kacheln) durch eine waagerechte oder senkrechte Linie in zwei (hier gleich große) Teile geteilt. Jeder Knoten der geschnittenen Gebiete wird einer der beiden neu entstandenen Teilflächen zugeteilt. Dieses wird durchgeführt, bis jeder Kachel höchstens eine zuvor festgelegte Anzahl von Knoten zugeteilt worden ist [She95]. Die Zuteilung der Knoten zu den Teilflächen wird so vorgenommen, daß die Anzahl der von der Trennlinie (Cut Line) geschnittenen Kanten des Graphen $Y(U,H)$ möglichst klein ist.

In der aktuellen Implementierung erfolgt die Aufteilung in immer kleinere Gebiete bis jede der entstandenen Kacheln höchstens einen Knoten (ein Netz) enthält. Kacheln mit mehr als einem Netz können wünschenswert sein, um innerhalb der Kacheln andere Plazierungsverfahren, etwa kombinatorisches Plazieren (vergleiche [She89], [Vyg98], [Ger99]) durchzuführen.

In dem hier vorgestellten Prototyp wird Breuers Algorithmus ([Bre77a], [Bre77b]) eingesetzt. Von den möglichen Varianten kommt für die Wahl der Lage der Schnittlinie eine modifizierte Form der „Quadrature Placement Procedure“ zur Anwendung. Die Chipfläche wird dabei wiederholt durch alternierend horizontal und vertikal gerichtete Schnittlinien in zwei Teile gleicher Größe aufgeteilt. Abbildung 24 zeigt die ersten vier Schnitte eines Plazierungsablaufs. Diese Wahl der Schnittlinien reduziert die Kantendichte im Zentrum und ist die am häufigsten verwendete Schnittfolge [She95].

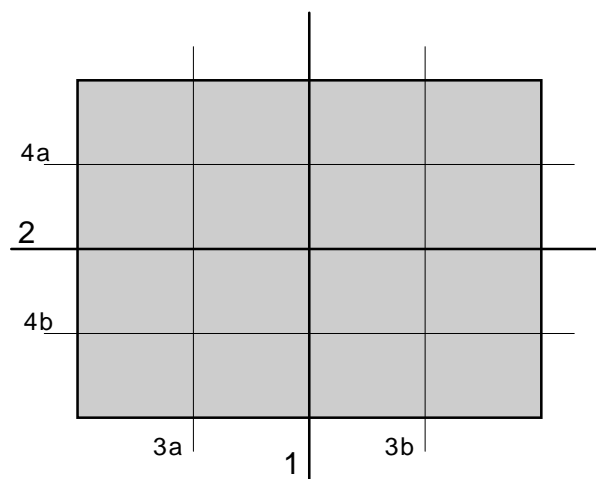
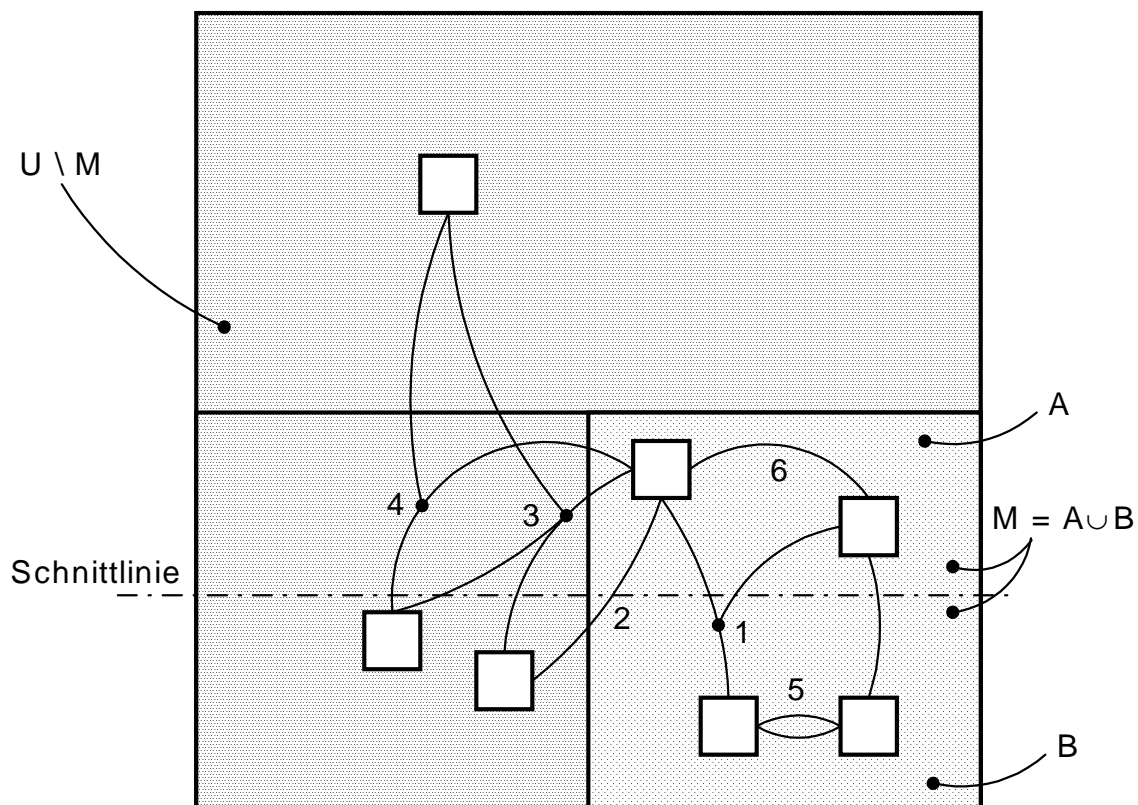


Abbildung 24: Lage und Reihenfolge der Schnittlinien für die ersten vier Iterationen bei der Partitionierung nach Breuers Algorithmus

Für die Verteilung der Knoten auf die durch die Schnittlinien gebildeten Gebiete offeriert Breuers Algorithmus mehrere Varianten zur Minimierung der Anzahl der geschnittenen Kanten. Die in dieser Arbeit verwendete Bewertungsfunktion (siehe Abschnitt 4.2.1.2) für die vorgenommenen Bipartitionen berücksichtigt zusätzlich zu der Anzahl der geschnittenen Kanten die Ausgewogenheit der Kardinalität der beiden Teile.

Die hier verwendete und im folgenden dargelegte Bewertungsfunktion für die vorgenommenen Bipartitionen ist eine Mischung der „total net-cut objective function“ und der „sequential cut line objective function“ [She95]. Als zusätzliches Kriterium wird die Ausgewogenheit der Kardinalität der beiden Teile berücksichtigt.

Eine Verbindung heißt interne Verbindung, wenn sie zwei Elemente aus M miteinander verbindet. Eine Verbindung heißt externe Verbindung wenn sie ein Element aus M mit zu einem Element aus $U \setminus M$ verbindet. Eine Hyperkante kann sowohl externe als auch interne Verbindungen herstellen.



Universität Hamburg
Fachbereich Informatik

Es seien die Elemente aus A oberhalb und die Elemente von B unterhalb der Schnittlinie angeordnet. Eine Hyperkante zwischen einem Element aus B und einem Element aus $U \setminus M$, das oberhalb der verlängerten Schnittlinie (in Abbildung 25 strichpunktiert dargestellt) liegt, ist eine geschnittene externe Verbindung. Stellt eine Hyperkante externe Verbindungen zu mehreren Pins her, so wird sie bei der Betrachtung geschnittener Kanten nicht berücksichtigt, wenn diese Pins zu beiden Seiten der Schnittlinie liegen. Die externen Verbindungen der Hyperkante 2 aus Abbildung 25 wird als geschnitten betrachtet, die Hyperkanten 3 und 4 dagegen nicht. Eine interne Verbindung wird geschnitten, wenn mindestens eine der von der Hyperkante verbundenen Zellen auf der anderen Seite der Schnittlinie liegt. Hyperkante 1 zeigt eine geschnittene interne Verbindung. Die Hyperkanten 5 und 6 sind nicht geschnittene interne Verbindungen.

Die Einbeziehung der Kardinalitätsdifferenz $||A| - |B||$ ist nur dann von Bedeutung, wenn einige der Zellen aufgrund von weiteren Bedingungen nicht beliebig A oder B zugeordnet werden dürfen. Wie in Abschnitt 4.2.1.1 dargelegt, betrifft dies nur diejenigen Knoten, welche externe Netze repräsentieren.

Für die Beurteilung einer Bipartition ergibt sich die folgende Bewertungsfunktion:

$$\text{Rating}(Y, M, A, B) = \begin{aligned} & c1 * \text{Anz. der geschnittenen internen Verbindungen} \\ & + c2 * \text{Anz. der geschnittenen externen Verbindungen} \\ & + c3 * ||A| - |B|| \end{aligned}$$

Gleichung 1: Bewertungsfunktion für eine Bipartition, die Faktoren $c1$, $c2$, und $c3$ sind empirisch ermittelte Gewichte.

Die Bestimmung der internen geschnittenen Verbindungen als Hauptkomponente der Bewertungsfunktion ist effektiv möglich. Die Einbeziehung der externen Verbindungen stellt die vorgenommenen Bipartition in den Kontext der betroffenen Zellen aus $U \setminus M$ (vergleiche [She89]). Die Wahl der Gewichtungsfaktoren $c1$, $c2$, $c3$ hat empirisch zu erfolgen. Die Möglichkeit Werte aus analogen Implementationen (vergleiche [She89]) für das durch $G(V, E)$ gegebene Plazierungsproblem übernehmen zu können, erscheint fraglich, da die Qualität der abschließend resultierenden Plazierung von dem nachfolgenden Schritt des Zelleinfügens (siehe Abschnitt 4.2.4) abhängt und daher die hier vorgenommene hierarchische Partitionierung nicht unabhängig davon bewertet werden kann. Die Wertebelegung für $c1$, $c2$, $c3$ und die maximale Anzahl von Knoten wird im Rahmen der Parameterevaluation in Kapitel 6.2 diskutiert.

4.2.1.3 Die für die Bipartitionen verwendete Heuristik

Für die Durchführung der Bipartition entlang der oben dargestellten Schnittlinien nach der in Abschnitt 4.2.1.2 beschriebenen Bewertungsfunktion ist jeder Algorithmus geeignet, der das in Gleichung 1 gegebene Qualitätsmaß verwenden kann. Hier wird eine Feduccia-Mattheyes (FM) Implementierung [She95] verwendet. Diese Weiterentwicklung des Kernighan-Lin (KL) Algorithmus ermöglicht es, die in Gleichung 1 gegebene Bewertung direkt anzuwenden. Die FM-Heuristik ist wohlverstanden und die Qualität ihrer Ergebnisse im Vergleich zu aktuellen und wesentlich komplexeren Heuristiken bekannt (vergleiche z.B. [Alp98]).

Ausgehend von einer Startpartitionierung wird durch sukzessives Verschieben einzelner Knoten von einer Seite der Schnittlinie auf die andere versucht, die Anzahl der

geschnittenen Kanten zu reduzieren. Es wird jeweils der Knoten bewegt, dessen Verschieben die Zahl der geschnittenen Kanten am stärksten reduziert.

Da die FM Heuristik einzelne Knoten bewegt und nicht wie der KL Algorithmus paarweise Vertauschungen vornimmt, ist sie flexibler, was die Kardinalität der beiden Partitionen angeht. Durch die geschickte Wahl der Datenstrukturen ist die FM Heuristik von wesentlich geringerer Laufzeitkomplexität als der KL Algorithmus, welcher mit $O(n^3)$ als eher langsam einzustufen ist, ohne dabei den Implementierungsaufwand erheblich zu vergrößern [She95].

4.2.1.4 Ein Beispiel: Die Netzliste T4

Wie bereits in Abschnitt 4.1.2 soll nun der bisher dargelegte Ablauf am Beispiel der Netzliste T4 (vergleiche Abbildung 5) veranschaulicht werden.

Zunächst wird die Netzliste T4, wie in Abschnitt 4.1.1 beschrieben, als Hypergraph $G_{T4}(V,E)$ und dualer Hypergraph $Y_{T4}(U,H)$ dargestellt. Es ergeben sich die in der Abbildung 26 dargestellten beiden Hypergraphenrepräsentationen.

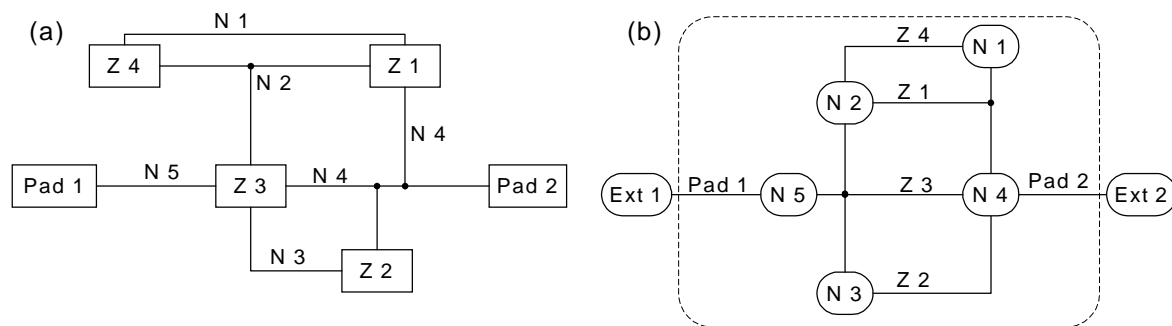


Abbildung 26: Die Netzliste T4 in (a) dargestellt als Hypergraph $G_{T4}(V,E)$ und in (b) als dualer Hypergraph $Y_{T4}(U,H)$.

Ausgehend von der dualen Hypergraphrepräsentation (vergleiche Abbildung 26.b) wird die hierarchische Partitionierung durchgeführt. Zunächst befinden sich, wie Abbildung 28.a zeigt, alle Knoten auf der noch nicht unterteilten Fläche. Die erste Bipartition entlang einer vertikalen Schnittlinie (Cut Line) ordnet die Netze N1, N2, N5 und Ext1 der linken Teilfläche zu, die Netze N3, N4 und Ext2 werden der rechten Teilfläche zugeordnet. Diese in Abbildung 27.a durch eine strichpunktierte Schnittlinie dargestellte Aufteilung erfordert lediglich das Schneiden zweier Hyperkanten (Z1 und Z3).

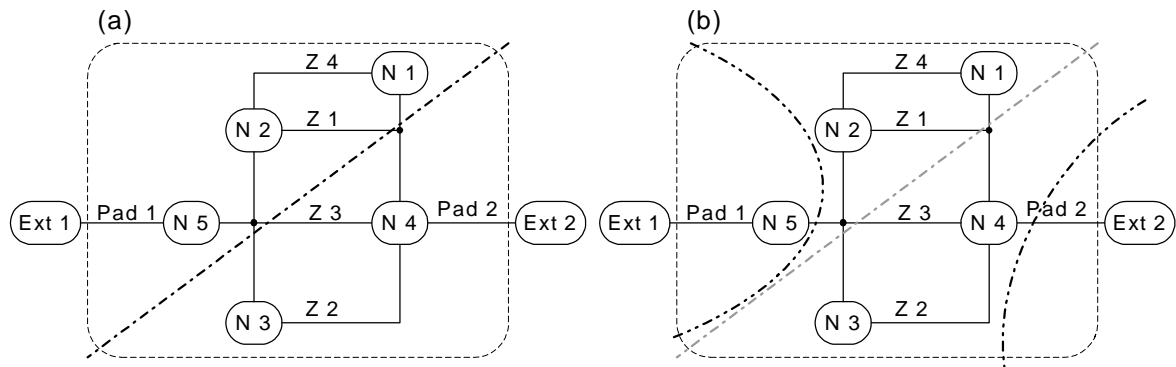


Abbildung 27: Die erste Bipartition (a) schneidet zwei Hyperkanten entlang der strichpunktirt dargestellten Linie, die zweite (b) teilt den Hypergraphen entlang der doppelt strichpunktirten Linien.

Die beiden entstandenen Teilflächen (Kacheln) sind in Abbildung 28.b dargestellt und werden nun durch eine horizontale Schnittlinie in jeweils zwei Kacheln unterteilt, die dadurch entstehende Zuordnung der Netze zu den neuen Teilgebieten ist in Abbildung 28.c gezeigt. Die bei dieser zweiten Teilung durchgeführten Schnitte im Hypergraphen sind in Abbildung 27.b durch eine doppelt strichpunktirte Linie kenntlich gemacht. Durch die nun folgende Unterteilung der entstandenen vier Kachel entlang zweier wiederum vertikaler Schnittlinien wird jeder entstehenden Kachel höchstens ein Knoten zugewiesen (siehe Abbildung 28.d).

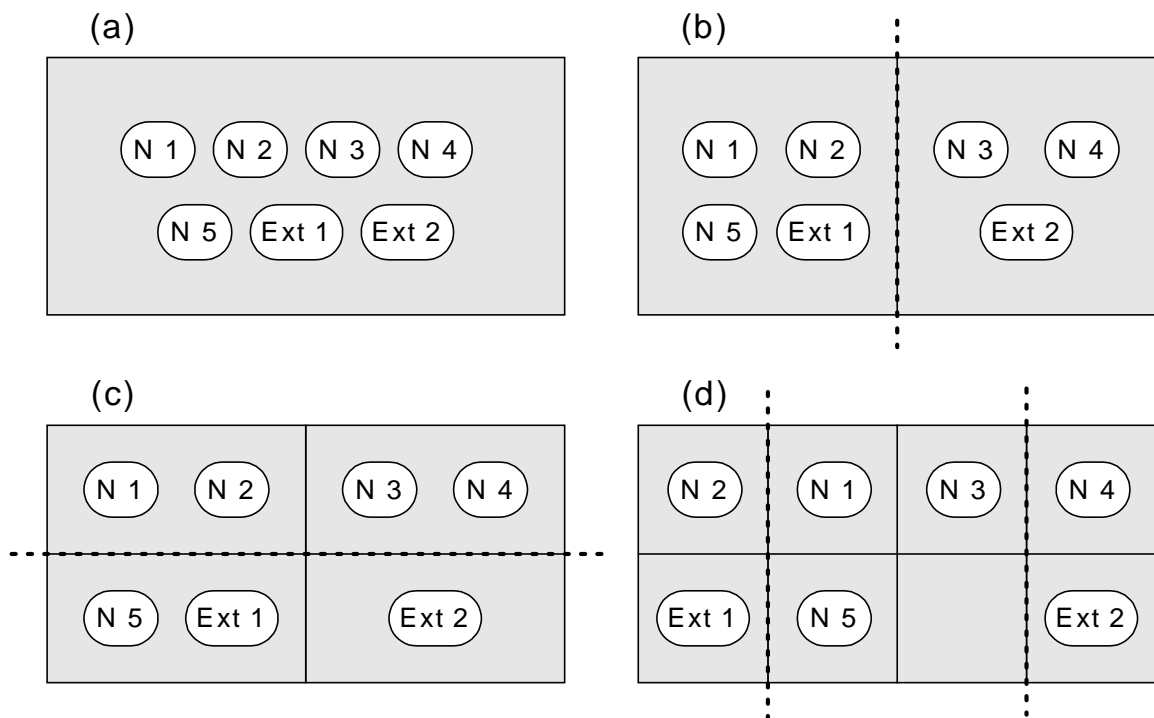


Abbildung 28: Die Zwischenschritte der mittels hierarchischer Partitionierung generierten Platzierung der Netzrepräsentanten für die Netzliste T4.

Eine weitere Unterteilung der Fläche ist nicht erforderlich und das Endergebnis der Netzplatzierung für die Netzliste T4 ist in Abbildung 29 wiedergegeben.

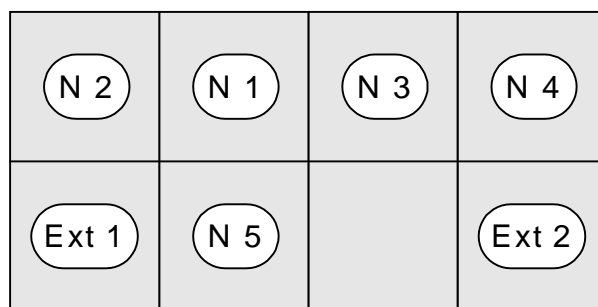


Abbildung 29: Die vollständige Platzierung der Netzrepräsentanten von Netzliste T4 unterteilt die verfügbare Fläche in acht Kacheln.

Durch das in diesem Abschnitt 4.2.1 beschriebene Vorgehen ist zu der gegebenen Netzliste nun eine Anordnung der Knoten des dualen Hypergraphen $Y(U,H)$ in der Ebene erfolgt. Diese stellt eine Platzierung der Netzrepräsentanten dar und dient als Grundlage der im folgenden geschilderten Anordnung der Zellen.

4.2.2 Einfügen der Zellen

In diesem Schritt sind die Zellen geeignet zwischen den Netzrepräsentanten zu positionieren. Ziel ist es, wiederum den durch eine geeignete Zielfunktion geschätzten Verdrahtungsaufwand so gering wie möglich zu halten. Um die Lösung möglichst allgemein zu halten, soll der gewählte Entwurstil so spät wie möglich berücksichtigt werden. Dieser Schritt wird, ebenso wie der vorhergehende, auf einer in ihrer Ausdehnung unbeschränkten diskret gerasterten Ebene durchgeführt und die betrachteten Objekte, die Netze und Zellen, als dimensionslos angesehen.

4.2.2.1 Konsequenzen der Repräsentation der Zellen als Hyperkanten

Der Pin einer Zelle ist der Punkt, an dem ein Netz endet. Da jede Zelle, mit Ausnahme von automatisch eingefügten Leitungstreibern oder Feedthroughs, mindestens an zwei Netze angeschlossen ist, lässt sich die Ausdehnung und damit die Länge eines Netzes, durch Verschieben der Zelle nur mit Rückwirkung auf die Ausdehnung der anderen angeschlossenen Netze beeinflussen.

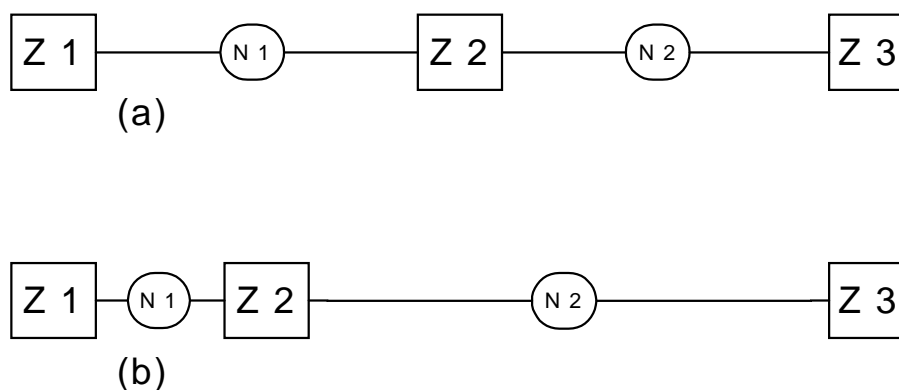


Abbildung 30: Das Verschieben der Zellen entlang der Verbindungslinien zwischen den Netzen beeinflusst die Größe der angeschlossenen Netze.

Die Abbildung 30 zeigt, wie die Lage der Zelle Z2 die Ausdehnung der Netze N1 und N2 beeinflusst. In Bild (a) sind die Netzlängen gleich, in Bild (b) ist N2 zugunsten eines kurzen Netzes N1 erheblich länger. Im allgemeinen ist eine Zelle an mehr als zwei Netze angeschlossen, wodurch die Wahl einer günstigen Position entsprechend komplexer wird.

Betrachtet man zunächst weiter das duale Problem $Y(U,H)$, so ist der nächste Schritt die Verdrahtung. Nun sind die Hyperkanten des betrachteten Graphen $Y(U,H)$ so in die Ebene einzubetten, daß keine technologischen Randbedingungen (z.B. die maximal zulässige Netzlänge) verletzt werden und die gewählte Zielfunktion nach Möglichkeit minimiert wird.

Mag bei der Betrachtung des dualen Problems die Repräsentation der Netze als Punkte, gegeben durch den Massenmittelpunkt der von ihnen verbundenen Pins, noch praktikabel sein, führt die konsequent fortgeführte duale Sichtweise zu einem Problem bei der Repräsentation der nahezu punktförmigen Zellen durch ausgedehnte Flächen. Diese flächige Repräsentation ergibt sich, wenn die Kanten aus H des Graphen $Y(U,H)$ in die Ebene eingebettet werden. Eine Kante von $Y(U,H)$ überdeckt nun die von dem unregelmäßigen n -Eck, das durch die verbundenen Knoten gebildet wird, umschlossene Fläche.

An einem noch festzulegenden Punkt des Lösungsverfahrens muß von der abstrakten dualen Problemsicht auf die ursprüngliche Aufgabe, reale Zellen und reale Leitungssegmente auf der Chipfläche anzuordnen, zurückgekehrt werden. Zu diesem Zeitpunkt muß die flächige Repräsentation der Zellen als Kanten aus H spätestens in eine eindeutige Position der Zelle auf dem Chip übersetzt werden. Um durch diese Rücktransformation nicht zuvor geleistete Arbeit zu verwerfen, ist es angebracht, die punktförmige Eigenschaft der Zellen möglichst früh zu berücksichtigen.

Führt man die duale Betrachtungsweise fort, so ist die Platzierung der Netze als nächstes zu verdrahten und anschließend die Zellen auf den Leitungen einzufügen, um dadurch die Grenzen der Netze zu definieren (vergleiche Abbildung 30). Bei diesem Vorgehen stellen die Leitungen die möglichen Positionen da, an denen die Zellen angeordnet werden können. Für Zellen mit zwei oder drei Pins ist das Finden einer geeigneten Position entlang den Leitungen unproblematisch. Abbildung 31 zeigt einen Ausschnitt aus einer möglichen Platzierung und Verdrahtung eines dualen Problems (Abbildung 31.a) und deren Rücktransformation auf das ursprüngliche Platzierungs- und Verdrahtungsproblem (Abbildung 31. b).

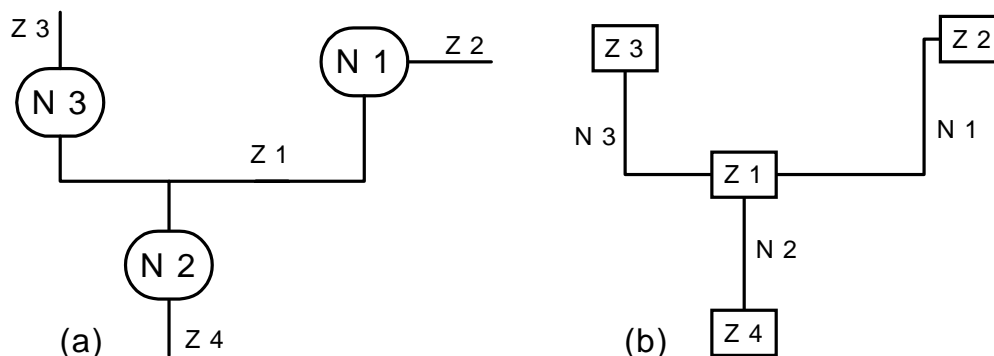


Abbildung 31: Ein Ausschnitt einer verdrahteten Platzierung des dualen Problems $Y(U,H)$ (a) und die Rücktransformation (b) für eine Zelle vom Grad drei.

Für Zellen mit vier oder mehr Pins ist das Finden einer Position entlang den Leitungen zwischen den Netzen für das Einfügen einer Zelle problematisch. Es existiert nicht zwingend ein Ort, an dem sich alle Netze treffen, und welcher damit prädestiniert für das Einfügen der Zelle wäre. In Abbildung 32.a ist der Ausschnitt einer Lösung eines dualen Problems dargestellt, die Zelle Z1 hat vier Pins. Die in Abbildung 32.b dargestellte Rücktransformation analog dem obigen Beispiel (vergleiche Abbildung 31) führt zu einer ungültigen Lösung, da das markierte Leitungssegment sowohl zu Netz N3 als auch zu Netz N4 gehören würde. Die in Abbildung 32.c dargestellte Ersetzung dieses Segmentes durch zwei parallele Segmente löst die Unvereinbarkeit zwar auf, zeigt aber einen deutlichen Schwachpunkt des Ansatzes. Selbst wenn ausreichend Platz für das zusätzliche Leitungssegment existiert, erzeugt diese Lösung viele kritische Bereiche wodurch die Gefahren des Übersprechens und die parasitären Kapazitäten zunehmen.

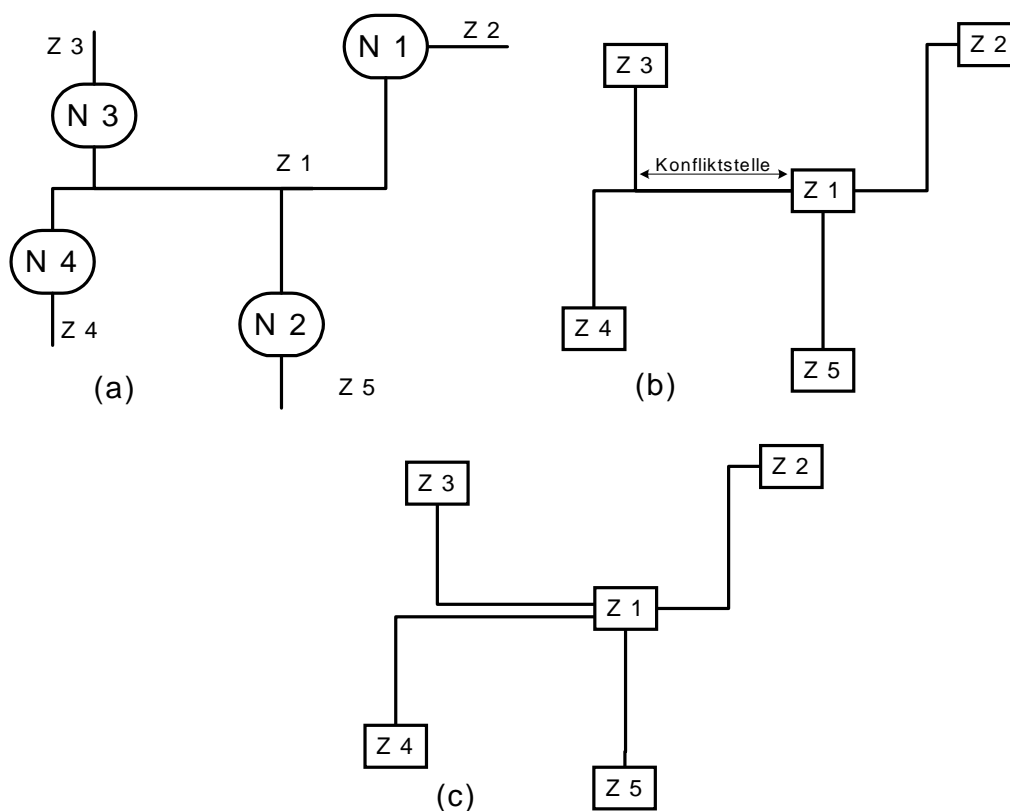


Abbildung 32: Rücktransformation einer Zelle vom Grad vier. Wird die Verdrahtung des dualen Problem (a) direkt auf das Ursprungsproblem abgebildet (b) entsteht eine unzulässige Verdrahtung. Das markierte Leitungselement wird von zwei Netzen verwendet. In (c) ist eine mögliche Korrektur dargestellt.

Die Probleme des Findens einer gültigen Rücktransformation verstärken sich mit der Anzahl der Pins der betrachteten Zelle. Es muß daher die Punktförmigkeit der Zelle bereits beim Verdrahten des dualen Problems berücksichtigt werden. Hierzu kann eine sternförmige Verdrahtung des dualen Problems vorgenommen werden, die sich problemlos auf die ursprüngliche Problemstellung zurücktransformieren läßt. Der Knotenpunkt der Verdrahtung wird hierbei bei der Rücktransformation zur Position der Zelle (siehe Abbildung 33).

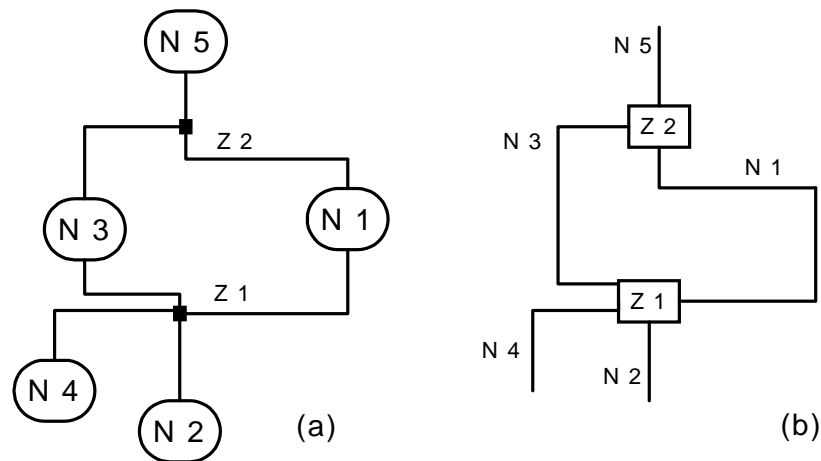


Abbildung 33: Die direkte Weiterverwendung der sternförmigen Verdrahtung des dualen Problems (a) führt zu einer ungünstigen Lösung der Verdrahtung des Ursprungsproblems (b).

Der in Abbildung 33 dargestellte Ausschnitt einer sternförmigen Verdrahtung eines dualen Problems verdeutlicht, daß die durch eine einfache Rücktransformation der Lösung des dualen Problems erhaltene Verdrahtung noch suboptimal ist. Zur Verbesserung sind entweder eine aufwendige Nachbesserung oder das Verwerfen der rücktransformierten Verdrahtung des dualen Problems und das Durchführen einer kompletten Neuverdrahtung, wobei lediglich die gewonnenen Zellpositionen erhalten bleiben, möglich. Da es sich bei der Nachoptimierung eines bestehenden Layouts um ein weiteres eigenständiges Aufgabenfeld im Kontext des physikalischen VLSI-Entwurfes handelt, wird dieser Weg nicht verfolgt, sondern der Weg der kompletten Neuverdrahtung gewählt.

Wird nun jedoch die sternförmige Verdrahtung des dualen Problems nur verwendet, um die Zelle auf dem zentralen Knotenpunkt zu positionieren, bietet es sich an, die Lage dieses Knotens zu ermitteln, ohne die Verdrahtung überhaupt durchzuführen. Die Verwendung der Steinerpunkte für die Zellposition liegt nahe und würde annähernd gleiche Ergebnisse, wie die oben beschriebene Verdrahtung, erzielen. Das Steinerproblem ist NP-hart ([GaJ79], [She95]), die existierenden effektiven Heuristiken (z.B. in [Mar93], [WWW84], [She95]) sind die Basis vieler Verdrahtungsalgorithmen, so daß der Aufwand zum Finden der Steinerpunkte dem der vollständigen Verdrahtung nahezu entspricht. Da die Position der Zellen sich im anschließenden Schritt der Anpassung an den Entwurfstils (hier den Standardzellentwurf) noch verändern kann (vergleiche Abschnitt 4.2.4), ist die Verwendung einer geeigneten Näherung der Zellposition ausreichend.

4.2.2.2 Approximation der Zellpositionen

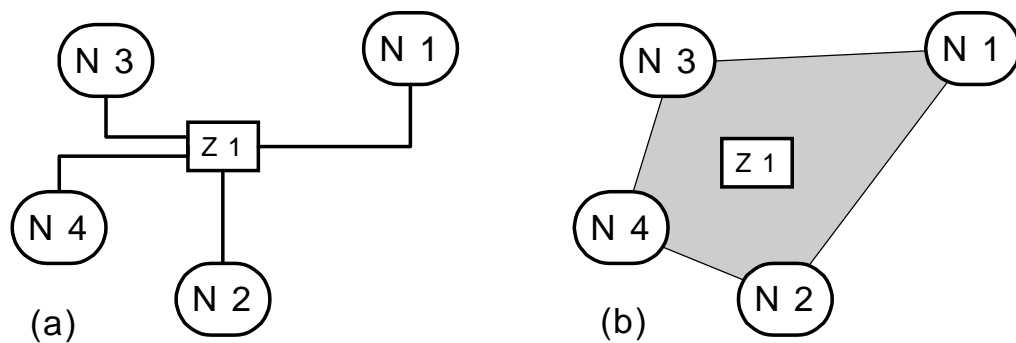


Abbildung 34: Vergleich der approximierten Zellposition (b) mit der durch sternförmiges Verdrahten ermittelten (a).

Für die Wahl der Position der Zelle wird, analog der Repräsentation der Netze im dualen Problem, der Massenmittelpunkt der Netzrepräsentanten aller Netze gewählt, an die die Zelle angeschlossen ist (siehe Abbildung 34). Dieses Vorgehen verwendet zum einen die gleiche Methode für die Rücktransformation, die schon für die Erzeugung des dualen Problems zur Anwendung kam und ist zum anderen durch die kräftegesteuerte Platzierung motiviert (siehe Abschnitt 4.2.3).

4.2.2.3 Ein Beispiel: Die Netzliste T4

Die in Abbildung 29 dargestellte Platzierung der Netzrepräsentanten der Netzliste T4 bildet die Grundlage für das Einfügen der Zellen und Pads. In Abbildung 35 ist gezeigt, wie die Zelle Z3 zwischen den Repräsentanten der Netze N2, N3, N4 und N5, an die sie angeschlossen ist, positioniert wird.

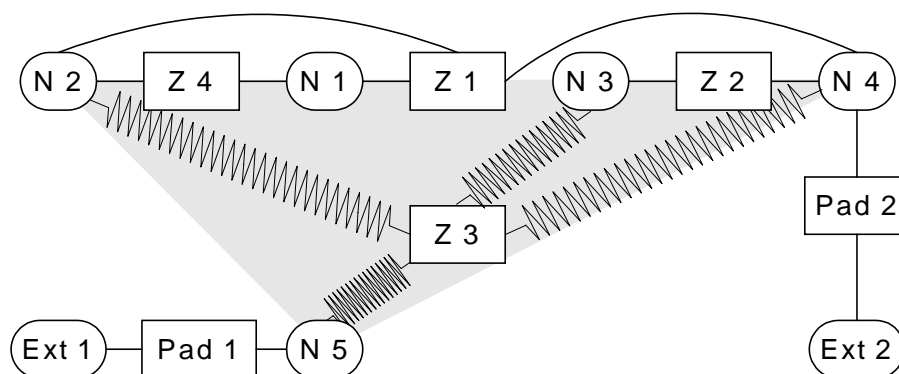


Abbildung 35: Die Zelle Z3 wird zwischen den Netzrepräsentanten der an sie angeschlossenen Netze N2, N3, N4 und N5 angeordnet.

Die Abbildung 36 zeigt über welches Gebiet sich das Netz N4 nach der vollständigen Anordnung der Zellen zwischen den Netzrepräsentanten erstreckt. Der Netzrepräsentant von N4 liegt in dieser ersten Anordnung noch außerhalb des Netzgebietes, daher wird vor einer Verwendung der Lage der Netzrepräsentanten in einem folgenden Schritt deren Lage neu ermittelt.

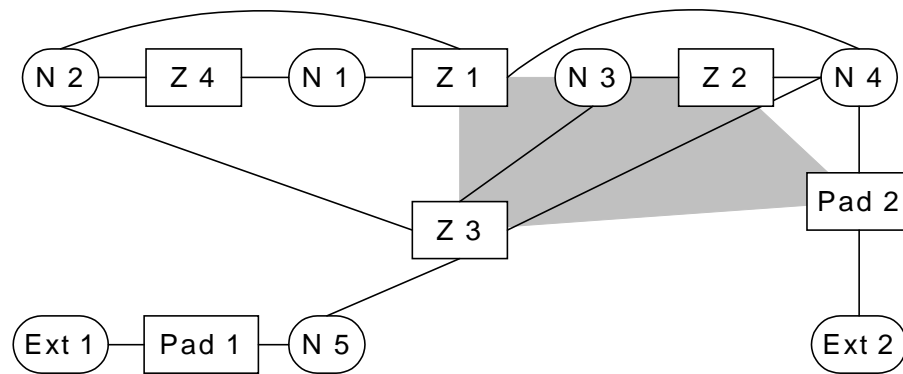


Abbildung 36: Die grau unterlegt dargestellte, vom Netz N4 überdeckte Fläche, wird durch die Lage der verbundenen Zellen Z1, Z2, Z3 und Pad2 bestimmt.

Das Endergebnis des Einfügens der Zellen zwischen den Netzrepräsentanten für die Netzliste T4 ist in Abbildung 37 gezeigt. Die Zellen sind als rechteckige Knoten zwischen den durch abgerundete Knoten dargestellten Netzrepräsentanten eingefügt. Die Kanten zwischen den Knoten für Zellen und Netze zeigen, wie die Netze die Zellen verbinden.

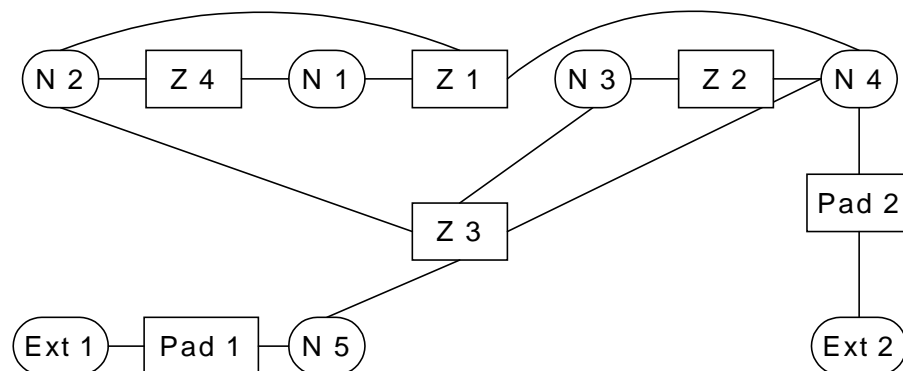


Abbildung 37: Die Zellen der Netzliste T4 sind zwischen den Netzrepräsentanten angeordnet.

Das Resultat der in diesem Abschnitt 4.2.2 beschriebenen Verarbeitungsschritte ist eine Platzierung von als punktförmig angenommenen Zellen (und Netzrepräsentanten) auf einer in ihrer Ausdehnung unbeschränkt angenommenen diskreten Ebene.

4.2.3 Verbesserung der Zellposition durch kräftebasiertes Plazieren

Durch die bei der Interpretation des dualen Problems und der Rücktransformation verwendete Repräsentationsmethode für die Kanten der Hypergraphen (vergleiche Abschnitte 4.1.2 und 4.2.2.2), bietet sich eine einfache Methode zur iterativen Verbesserung der Platzierung der Zellen an, die in diesem Teilkapitel dargelegt werden soll.

4.2.3.1 Iterative Verbesserungen

Die Verbesserung der Zellpositionen wird unter Verwendung beider Repräsentationen $G(V,E)$ und $Y(U,H)$ iterativ durchgeführt. Jede Iteration besteht aus zwei Schritten. Im ersten Schritt werden die Positionen der Netzrepräsentanten verändert und im zweiten Schritt die Positionen der Zellen. Das Vorgehen ist in beiden Schritten identisch: Es werden jeweils die punktförmigen Repräsentanten der Hyperkanten zwischen den

festgehaltenen Knoten neu angeordnet. Die Lage eines Kantenrepräsentanten ist dabei gegeben durch den Massenmittelpunkt der Knoten, die durch die von dem betrachteten Repräsentanten dargestellte Hyperkante verbunden sind.

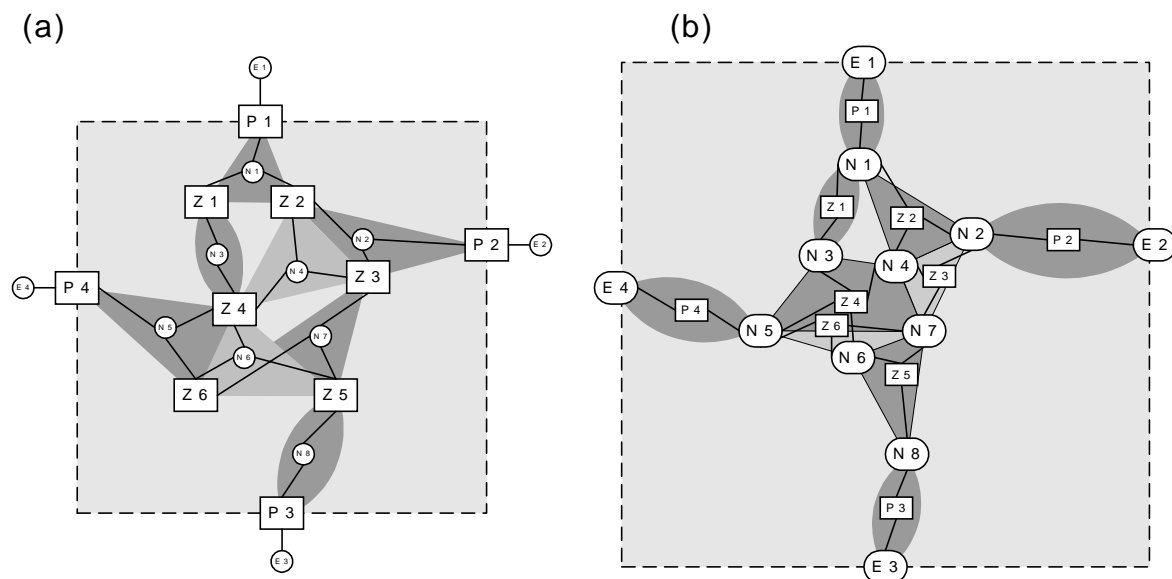


Abbildung 38: Abwechselndes Verschieben der Netzrepräsentanten (a) und Zellen (b) zur Verbesserung der Platzierung.

Im ersten Schritt einer Iteration wird zunächst der ursprüngliche Hypergraph $G(V,E)$ betrachtet. Es werden die Netzrepräsentanten zwischen den Zellen neu angeordnet. In Abbildung 38.a sind die Zellen unbeweglich und die kleiner dargestellten Netzrepräsentanten werden zwischen ihnen positioniert. Die grau unterlegten Flächen und Verbindungslinien zwischen den Netzen und Knoten veranschaulichen die durch die Netze hergestellten Verbindungen. Die externen Netze werden entlang der Chipkante, der sie zugeordnet sind, verschoben (vergleiche Abschnitt 4.2.3.2), so daß sie zu der jeweils zugehörigen Padzelle einen möglichst kleinen Abstand aufweisen.

Im zweiten Schritt einer Iteration vertauschen Netzrepräsentanten und Zellen die Rollen. Nun sind die in Abbildung 38.b größer dargestellten Netzrepräsentanten unbeweglich und die klein dargestellten Zellen werden dazwischen neu positioniert.

Die Iterationen werden beendet, wenn die Bewegungen der Zellen einen durch einen Parameter vorgegebenen Schwellwert unterschreiten (vergleiche Abschnitt 4.2.3.2).

Das obige Vorgehen ist motiviert aus der kräftegesteuerten Platzierung (Force Directed Placement, vergleiche [She95], [Brü93]). Betrachtet man die Knoten aus V des Graphen $G(V,E)$ als fest und platziert die punktförmigen Kantenrepräsentanten jeweils im Massenmittelpunkt des Mehrkörpersystems, welches durch die über das betrachtete Netz verbundenen Zellen gebildet wird, so fällt dieser Punkt mit dem Ort des Kräftegleichgewichts für n gleiche verbundene (lineare) Federn zusammen (siehe Abbildung 39).

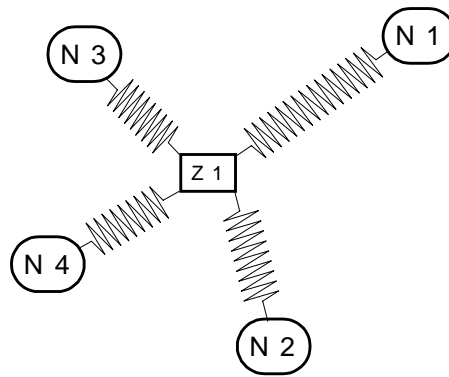


Abbildung 39: Als Modell für die kräftegesteuerte Platzierung dient ein System aus idealen Federn, die die zu platzierenden Objekte verbinden. Hier wird die neue Lage der Zelle Z1 zwischen den Netzen N1 bis N4 ermittelt.

4.2.3.2 Die Abbruchbedingung

Ein, wie im letzten Abschnitt beschriebenes, physikalisches Mehrkörpersystem, strebt, wenn es sich selbst überlassen wird, dem Zustand geringster innerer Energie zu. Das System sich selbst zu überlassen, bedeutet in diesem Fall, den verbundenen Knoten zu erlauben, sich in der Ebene beliebig zu bewegen. Das Verhalten eines solchen Systems läßt sich durch iterative Berechnung der Positionen der Knoten simulieren. Nach ausreichend vielen Iterationen stellt sich in diesem System der Zustand minimaler innerer Energie ein. Existieren in einem solchen System keine Fixpunkte (und keine Hindernisse), werden schließlich alle beteiligten Knoten in der Mitte zusammenklumpen. Bei der Annahme dimensionsloser Knoten und Federn wird das gesamte System in einem Punkt kollabieren. Die Fixpunkte werden durch die Knoten der im dualen Graphen eingeführten externen Netze gebildet. Diese werden am Rande der verfügbaren Fläche verankert und können sich nur entlang der Kanten bewegen. Durch diese Fixpunkte wird dem Kollaps entgegengewirkt und das System wird, wenn es sich selbst überlassen wird, dem Minimum der inneren Energie zustreben und in diesem schließlich verweilen.

Sobald die inneren Zustandsänderungen einen geeigneten Schwellwert unterschreiten, werden die Iterationen beendet und der erreichte Zustand stellt das Ergebnis des entwurfstilunabhängigen Platzierens dar.

Zur Beurteilung der Veränderungen in dem betrachteten System, werden die Entfernungen betrachtet, um die sich die Knoten, welche Zellen repräsentieren, je Iteration bewegen. Dabei wird zum einen der Mittelwert der Distanz über alle Zellen betrachtet (D_{Avg}) und zum anderen die Entfernung, um die sich die Position der am weitesten bewegten Zelle verändert hat (D_{Max}). Die Iterationen werden beendet, wenn entweder die Distanz D_{Avg} oder die Distanz D_{Max} außerhalb der für die beiden Werte getrennt parametrisierbaren Intervalle liegen.

Die verwendeten Abbruchkriterien und das Verhalten dieses Systems in der realisierten Implementierung sind in Abschnitt 5.3.2.3 näher dargelegt.

4.2.3.3 Ein Beispiel: Die Netzliste T4

Für die Netzliste T4 wird in der Abbildung 40 die Anordnung von Zellen und Netzrepräsentanten bei der ersten Iteration des Verfahrens zur Verbesserung der Zellposition gezeigt.

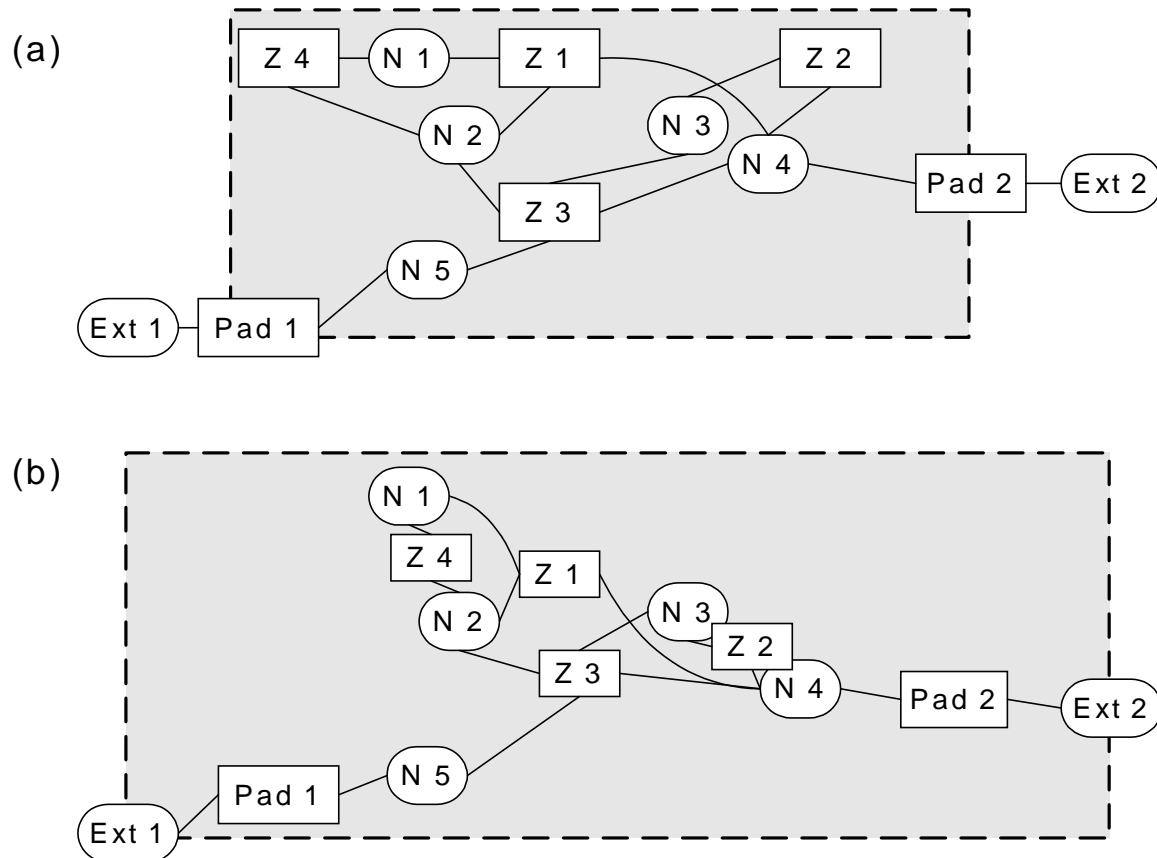


Abbildung 40: Die Lage der Netzrepräsentanten und Zellen der Netzliste T4 bei der ersten Iteration zur Verbesserung der Zellpositionen. In (a) sind die Netzrepräsentanten bewegt worden, in (b) anschließend die Zellen.

In Abbildung 40.a ist die Situation dargestellt, nachdem die Lage der Netzrepräsentanten verändert wurde. Ein Vergleich mit Abbildung 37 zeigt, daß nach dem ersten Teilschritt der ersten Iteration nun die Netzrepräsentanten innerhalb der von dem durch sie dargestellten Netz überdeckten Fläche liegen. Die Position der Zellen wurde in diesem ersten Schritt nicht verändert.

Das Resultat des zweiten Teils der ersten Iteration ist in Abbildung 40.b gezeigt. Hier wurden die Zellen bewegt und zwischen den Repräsentanten der an sie angeschlossenen Netze positioniert. In diesem Teilschritt wurde die Lage der Netzrepräsentanten im Vergleich zur Situation in Abbildung 40.a nicht verändert.

Ein Vergleich der Abbildung 37 mit Abbildung 40.b zeigt, daß sich bereits nach einem Iterationsschritt die Neigung der Zellen und Netzrepräsentanten, sich in der Mitte der verfügbaren Fläche zu sammeln (vergleiche Abschnitt 4.2.3.2), deutlich abzeichnet.

Nach Abschluß der hier in Abschnitt 4.2.3 beschriebenen Verarbeitungsschritte sind die punktförmig angenommenen Zellen nun in der diskret gerasterten Ebene angeordnet. Die Zellkoordinaten in der Ebene sind kein Maß für die benötigte Chipfläche, sondern nur durch die bei der Implementierung vorgenommene Wahl der Ebenengröße bestimmt. Dieses Zwischenergebnis, im folgenden als die Vorplazierung bezeichnet, ist nun an den gewünschten Entwurstil, hier den Standardzellentwurf, anzupassen.

4.2.4 Anpassen der Platzierung an den Standardzellentwurf

Ziel dieses Schrittes ist es, die Vorplatzierung in eine Standardzellplatzierung zu überführen. Beim Standardzellentwurf sind die Zellen in parallelen Reihen angeordnet (vergleiche Abbildung 1.d). Zwischen den Zellreihen liegen die Verdrahtungskanäle (Routing Channels). Die Zellen werden nun nicht mehr als dimensionslose Objekte aufgefaßt, sondern durch Rechtecke mit der in der Zellbibliothek gegebenen Größe, d.h. einheitlicher Höhe und individueller Breite, repräsentiert. Die Anschlußpunkte (Pins) der Zellen befinden sich an der Ober- bzw. Unterkante des Rechteckes und sind gemäß der Netzliste einem Netz zugeordnet.

Während der Zuordnung der Zellen zu geeigneten Zellreihen wird der Bedarf an Feedthroughs geschätzt. Dies geht im Gegensatz zu den bisher in ESPRO realisierten Platzierern über eine grobe Abschätzung der insgesamt benötigten Anzahl an Feedthroughs hinaus. Der hier verwendete Ansatz fügt die für jedes Netz benötigten Feedthroughs gezielt dort ein, wo sie nach einer Abschätzung, basierend auf den Positionen der vom Netz verbundenen Zellen und des Netzrepräsentanten, wahrscheinlich benötigt werden. Dies ist ein effektiver Weg zu kompakteren Platzierungen, da nur der wirkliche Bedarf an Feedthroughs vorgesehen wird. Bei der gleichmäßigen Verteilung von Feedthroughs, wie bisher geschehen ([Kla94], [Jen97]), müssen so viele Feedthroughs eingefügt werden, daß auch an Orten hoher Verdrahtungsdichte ausreichend Möglichkeiten zum Durchqueren der Zellenreihen existieren. Dies führt zu einer großen Anzahl ungenutzter möglicher Feedthroughs und damit einer unnötig geringen Zelldichte in Regionen mit geringerer Verdrahtungsdichte.

Der folgende Abschnitt 4.2.4.1 beschreibt den Algorithmus zum Anordnen der Zellen in Standardzellreihen. Das Schätzen von Anzahl und Lage der benötigten Feedthroughs macht einen großen Teil des Anpassens an den Standardzellentwurf aus und wird aufgrund seines Umfanges in einem eigenen Abschnitt 4.2.4.2 behandelt. Die Padplatzierung ist ein eigener Bearbeitungsschritt und wird ebenfalls in einem eigenen Abschnitt 4.2.4.3 behandelt. Den Abschluß dieses Teilkapitels bildet wieder das Beispiel der Netzliste T4.

4.2.4.1 Erzeugen der Zellreihen und Abschätzen der Chipgröße

Die Gesamtchipfläche eines Standardzellentwurfes setzt sich aus der Kernfläche (Core Area) und dem Padkranz zusammen. Es wird, je nach dem welche der beiden Teilflächen die Gesamtgröße des Chips bestimmt, von einem pad-dominierten oder core-dominierten Layout gesprochen. Beim core-dominierten Layout (vergleiche Abbildung 41.a) bestimmt der Platzbedarf der Zellen und die benötigte Fläche für die Verdrahtung die Chipgröße. Die resultierende Kantenlänge des Chips bietet ausreichend Platz zur Positionierung der Padzellen. Bei einem pad-dominierten Layout (vergleiche Abbildung 41.b) wird die Chipfläche von der Anzahl der Padzellen und dem von ihnen benötigten Platz bestimmt. Die für die Standardzellreihen verbleibende Kernfläche ist mehr als ausreichend groß. Für diesen Fall wird die Ausdehnung der Kernfläche (Core Extent) und die der Gesamtfläche separat bestimmt. Für die Anordnung der Zellreihen ist nur die Kernfläche von Relevanz. Meist wird für die Chipfläche ein Seitenverhältnis von 1:1 angestrebt, da ein Quadrat aufgrund seiner Eigenschaften der Minimierung der Netzlängen förderlich ist [She89], aber auch jedes andere Seitenlängenverhältnis ist wählbar.

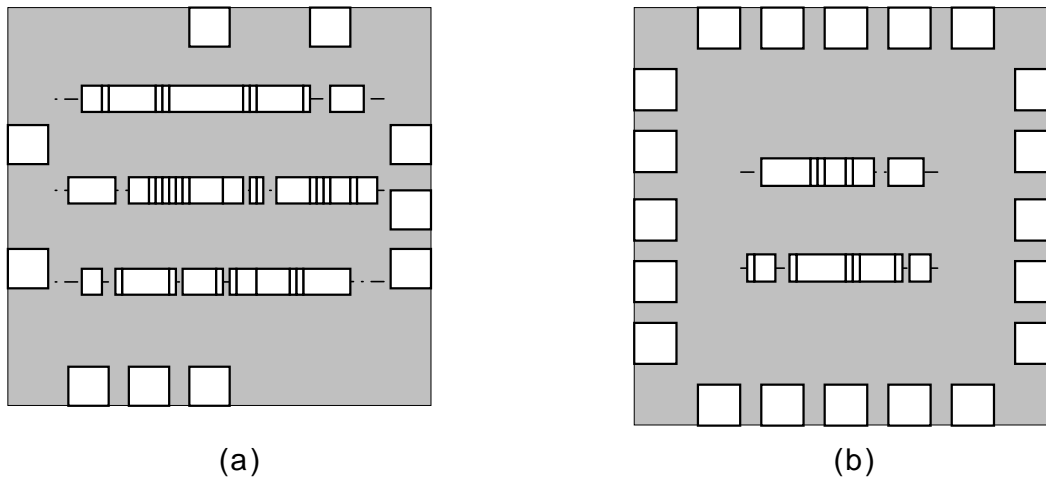


Abbildung 41: Vergleich von core-dominiertem (a) und pad-dominiertem (b) Layout

Für das Abschätzen der Corefläche vor dem Aufbau der Standardzellreihen wird eine einfach zu berechnende Schätzfunktion verwendet. Die Anzahl und Länge der verwendeten Zellreihen hängt von dem für Zellen und Feedthroughs benötigten Platz ab. Der Platzbedarf der Zellen ist durch die gegebenen Zellbreiten bekannt, der von den Feedthroughs belegte Raum muß geeignet geschätzt werden, hier dient die Anzahl der Netze in den Kanälen als Anhaltspunkt. Die benötigte Corefläche läßt sich nur bei Kenntnis der Kanalbreiten, also der Anzahl der benötigten Tracks im Kanal, ermitteln. Für die Kanalbreiten kommt eine Schätzfunktion zum Einsatz, die auf Anzahl der Netze im Kanal und der Packungsdichte der zu verdrahtenden Pins basiert. Je dichter die Pins beieinander liegen, um so wahrscheinlicher ist die Notwendigkeit, den Kanal um weitere Tracks zu verbreitern, um Verdrahtbarkeit zu gewährleisten.

Für die Zuordnung der Zellen zu Reihen wird ein sogenannter Sweep-Line Algorithmus eingesetzt, ein dem in der Computergraphik verbreiteten Scan-Line Algorithmus [Str01] verwandtes Verfahren. Eine Gerade, die parallel zur gewünschten Zellreihenorientierung verläuft (in ESPRO horizontal) durchläuft die Vorplazierung der Zellen von oben nach unten (vergleiche Abbildung 42). Dabei werden alle Zellen, die sie dabei überstreicht, einer Zellreihe zugeordnet, bis diese Reihe eine gewünschte Länge erreicht hat. Der von den Zellen benutzte Platz entspricht dabei der Zellbreite. Ist eine vollständige Zellreihe entstanden, wird eine neue Zellreihe gefüllt und kommt später unterhalb der vorigen zu liegen. Nach dem Einfügen einer Zelle in die aktuelle Reihe wird für alle an diese Zelle angeschlossenen Netze der Bedarf an Feedthroughs in dieser Zellreihe ermittelt und an den geschätzten Positionen je ein Platzhalter für ein Feedthrough vorgesehen (siehe Abschnitt 4.2.4.2).

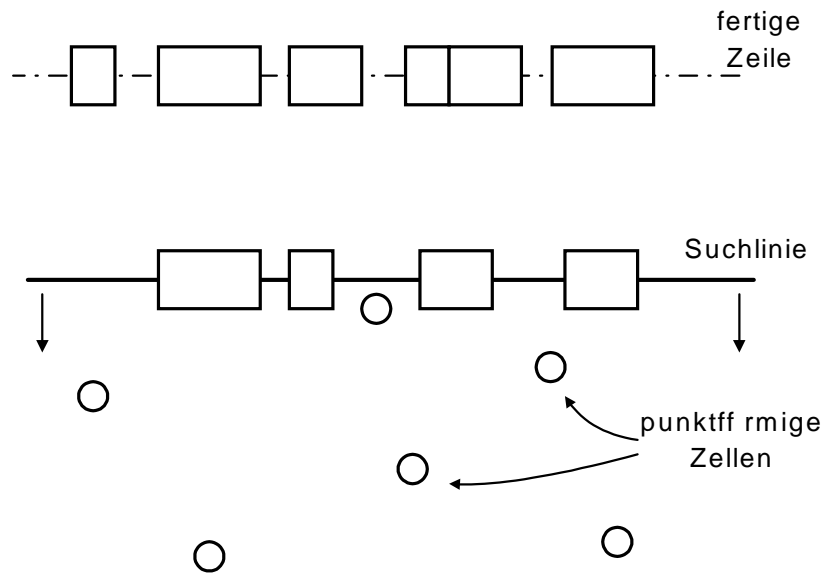


Abbildung 42: Anordnen der Zellen in Reihen. Die Sweep-Line überstreicht die Vorplazierung und ordnet die Zellen in Reihen an. Dabei wird von der punktförmigen auf eine flächige Zellrepräsentation gewechselt.

Bei der Plazierung einer von der Sweep-Line überstrichenen Zelle, wird die Position der Zelle innerhalb der Reihe aus der x-Koordinate der Zelle in der Vorplazierung ermittelt. Befindet sich an der so ermittelten Stelle in der Reihe keine ausreichend große Lücke, um die Zelle dort zu plazieren, wird in der unmittelbaren Nachbarschaft nach einer passenden Lücke gesucht. Wird innerhalb eines vorgegeben Suchradius kein Ort für die Zelle gefunden, so wird an der ursprünglich vorgesehenen Zelle die Lücke vergrößert, indem die Nachbarzellen nach links und rechts verschoben werden. Analog wird für die einzufügenden Feedthroughzellen verfahren.

Als Kriterium, ob eine Zellreihe vollständig ist, dient der in Gleichung 2 angegebene Quotient aus der Summe der bisher in der Reihe befindlichen Zellen (einschließlich vorgesehener Feedthroughs) und der maximal zulässigen Zellreihenlänge, die sich aus der geschätzten Chipgröße ergibt.

$$\text{RowUtilization} = \frac{\text{Zellbreitensumme}}{\text{maximaleZellreihenlänge}}$$

Gleichung 2: Der Quotient aus der Summe der Zellbreiten einer Reihe und der zulässigen Zellreihenlänge gibt den Füllungsgrad der Zellreihe an.

Übersteigt dieser Quotient den Wert eines vorgegeben Parameters (siehe Anhang A), so wird die Zellreihe als komplett angesehen und es wird mit der Füllung der nächsten begonnen.

Sind alle Zellen einer Zellreihe zugeordnet, oder sind alle vorgesehenen Zellreihen gefüllt, so wird geprüft, ob das gefundene Layout zulässig und akzeptabel ist. Zulässig ist es, sofern alle Zellen untergebracht sind. Als Akzeptanzkriterium dient der Füllungsgrad der letzten verwendeten Zellreihe sowie das erreichte Seitenlängenverhältnis der Core Area. Beide Kriterien müssen innerhalb der durch Parameter des Algorithmus (siehe Abschnitt 5.3.2 und Anhang A) vorgegebenen Grenzen liegen. Wird die erreichte Plazierung

zurückgewiesen, so wird nach einer Neuschätzung der benötigten Kernfläche und Zellreihenanzahl das Verfahren wiederholt.

Für die Abschätzung der Kanalbreiten kann nach einem Durchlauf der Sweep-Line durch die Vorplazierung eine andere, genauere Schätzfunktion verwendet werden, als bei der Ermittlung der initialen Chipgröße. Diesem Schätzer stehen detaillierte Informationen über die Lage der Zellen und damit der Ausdehnung der Netze und ihre Verteilung auf die Verdrahtungskanäle zur Verfügung. Diese Schätzfunktion bestimmt für jedes Netz das umschließende Rechteck. Die Verdrahtung des Netzes wird mit hoher Wahrscheinlichkeit möglich sein ohne Leitungssegmente außerhalb dieses Rechtecks verlegen zu müssen. Es wird angenommen, daß jeder Kanal innerhalb dieses Rechtecks mit der gleichen Wahrscheinlichkeit für die Verdrahtung verwendet wird. Damit wird das Netz zu der Füllung jedes Kanals innerhalb des umschließenden Rechtecks zu gleichen Teilen beitragen. Für ein Rechteck, das k Kanäle umschließt, wird in dem betroffenen Abschnitt der k Kanäle an jeder Stelle insgesamt ein Track in irgendeinem der Kanäle für dieses Netz benötigt. In jedem Kanal wird also jeweils $\frac{1}{k}$ Track benötigt. Die Kanalbreiten ergeben sich durch Aufsummieren dieser Anteile von allen Netzen (vergleiche [Die89]). Ein Skalierungsfaktor (siehe Abschnitt 5.3.2 und Anhang A) ermöglicht es, diese Abschätzung an empirisch gewonnene Ergebnisse anzupassen.

In Abbildung 43 ist der Ablaufplan des Algorithmus ohne die Details des Feedthrough Einfügens dargestellt.

Sweep-Line – Algorithmus

- 1) Schätzen der benötigten Chipfläche und der Anzahl an benötigten Zellreihen (rowCount) sowie deren maximal zulässige Länge.
- 2) Erzeugen eines leeren Layouts mit rowCount Zellreihen und den zugehörigen (rowCount+1) Verdrahtungskanälen. Wähle die oberste leere Zellreihe als aktuelle (rowNo = 0).
- 3) Wähle die Zelle Z mit kleinster y-Koordinate.
- 4) Füge Z in die aktuelle Zellreihe ein. Die Position innerhalb der Zellreihe ergibt sich aus der x-Position der Zelle in der Vorplatzierung.
- 5) Einfügen von Platzhaltern für Feedthroughs für alle von der Zelle berührten Netze, soweit erforderlich.
- 6) Test, ob die aktuelle Zellreihe den gewünschten Füllungsgrad erreicht hat, wenn ja, wähle die nächste leere Zellreihe als aktuell (rowNo \leftarrow rowNo+1).
- 7) Wenn alle Zellen untergebracht sind oder rowNo > rowCount gehe zu 8), sonst weiter bei 3).
- 8) Schätze die Kanalbreiten neu mit einer platzierungsbasierten Schätzfunktion,
- 9) Wenn nicht alle Zellen platziert wurden, zerstöre die bisher gewonnene Platzierung, erhöhe die Anzahl an Zellreihen oder die zulässige Zellenlänge. Weiter bei 2).
- 10) Test, ob die letzte verwendete Zellreihe die geforderte Mindestfüllung aufweist, wenn nein, verwirfe die bisher gewonnene Platzierung, reduziere die zulässige Zellreihenlänge oder die Anzahl der Zellreihen und weiter bei 2).
- 11) Test, ob die gewonnene Standardzellenplatzierung das geforderte Seitenlängenverhältnis aufweist. Wenn nein, zerstöre die gefundene Platzierung, ändere Zellreihenanzahl und Zellreihenlänge, um dem geforderten Seitenlängenverhältnis näher zu kommen und weiter bei 2). Wenn ja, ENDE.

ENDE

Abbildung 43: Ablauf des Sweep-Line Algorithmus zur Anordnung der Zellen in Reihen

4.2.4.2 Abschätzen von Anzahl und Lage der benötigten Feedthroughs

Feedthroughs sind erforderlich, um ein Netz von einem Kanal in einen anderen fortzusetzen, wenn diese durch eine Zellreihe getrennt werden. Um die Feedthroughs gezielt einzufügen, wird für jedes an eine aufgereihete Zelle angeschlossene Netz überprüft, ob in der Zellreihe Platz für ein Feedthrough vorgesehen werden muß. Dabei sind die Position der Pins, die Lage des Netzmittelpunktes sowie die Zuordnung des Netzes zum Kanal oberhalb bzw. unterhalb der aktuellen Zellreihe verwendete Anhaltspunkte.

Es wird für jeden Kanal protokolliert, welche Netze dort verlaufen. Ein Netz wird einem Kanal zugeordnet, wenn mindestens ein von ihm verbundener Pin sich in diesem Kanal befindet. Die Zuordnung der Netze zu Kanälen geschieht zusammen mit dem Zuordnen der Zellen zu Reihen. Dabei werden jeweils nur die beiden Kanäle, die an die aktuell betrachtete Zellreihe grenzen, behandelt.

Feedthroughs sind nötig, wenn bekannt ist, daß ein Netz in den beiden von der aktuellen Zellreihe getrennten Kanälen präsent ist. Sind in beiden benachbarten Kanälen Pins des Netzes, ist dies offensichtlich der Fall und sofern noch keine Feedthroughzelle für dieses Netz existiert, wird eine vorgesehen. Da der Sweep-Line Algorithmus die Zellreihen von oben nach unten aufbaut, sind die Informationen über Netze in Kanälen unterhalb der aktuell betrachteten Zellreihe unvollständig. Um möglichst früh die benötigten

Feedthroughs einfügen zu können, wird die Lage des Netzmittelpunktes ausgewertet. Ist ein Netz im Kanal oberhalb der aktuellen Zellreihe präsent und hat es im Kanal unterhalb der aktuellen Zellreihe keinen Anschlußpunkt, aber sein Netzmittelpunkt liegt unterhalb der aktuellen Sweep-Line Position, so muß noch mindestens ein weiterer Pin unterhalb der aktuellen Reihe folgen, daher wird in diesem Fall ein Feedthrough vorgesehen. Entsprechend wird ein Feedthrough vorgesehen und das Netz dem oberen Kanal zugeordnet, sofern dieses Netz im Kanal unterhalb der aktuellen Zellreihe vorkommt und bekannt ist, daß das Netz ebenfalls in einem weiter oben gelegenen Kanal präsent ist.

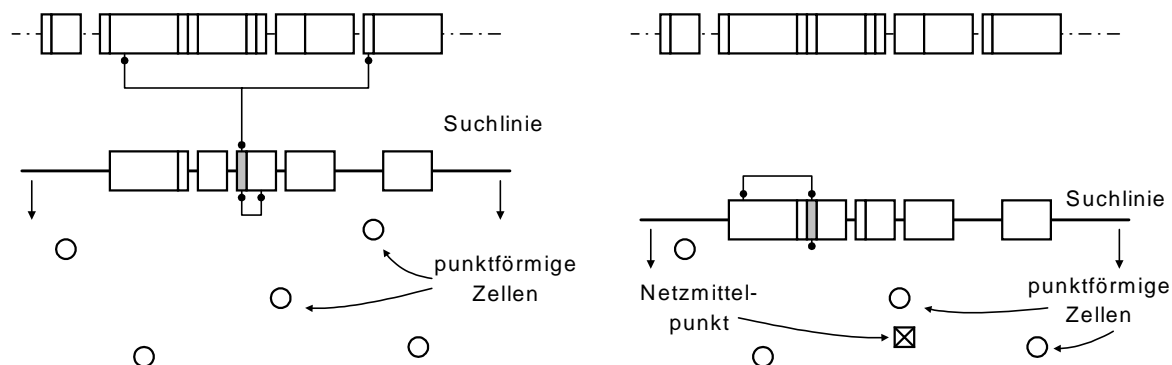


Abbildung 44: Verschiedene Situationen, in denen eine Feedthroughzelle vorgesehen wird. Die neu eingefügte Feedthroughzelle ist jeweils grau dargestellt.

Die Lage der Feedthroughs innerhalb einer Zellreihe wird zwischen der Zelle, deren Netze momentan untersucht werden und dem Pin, der für die Erzeugung dieses Feedthroughs verantwortlich ist, gewählt. Das Einfügen der Feedthroughzelle erfolgt analog dem Einfügen der aktiven Zellen wie oben beschrieben.

Auch wenn die Feedthroughs für die Dauer des Sweep-Line Algorithmus einem Netz zugeordnet werden, sind die entstehenden Freiräume in den Zellreihen in der erzeugten Platzierung anonym, also keinem bestimmten Netz zugeordnet. Die endgültige Verwendung der entstehenden Lücken für ein spezielles Netz wird durch die Globalverdrahtung vorgenommen.

Nach Abschluß des Sweep-Line Algorithmus werden die Orte für Feedthroughs nur mehr durch Lücken in der Zellreihe repräsentiert. Es ist möglich, die Zellen in den Zellreihen gleichmäßig zu verteilen. Damit wird die zuvor getroffene Abschätzung des lokal zu erwartenden Feedthroughbedarfes wieder über die gesamte Zellreihe gemittelt und der Platzierungsalgorithmus verhält sich ähnlich wie frühere Implementierungen innerhalb des ESPRO Umfeldes. Diese Gleichverteilung der Zellen ist optional und kann erfolgen, um die Platzierung für ein nachgeschaltetes Verdrahtungsmodul anzupassen.

4.2.4.3 Platzieren der Padzellen

Die Padzellen nehmen eine Sonderstellung unter den zu platzierenden Zellen ein. Im Gegensatz zu den in der Core Area platzierten Zellen, muß die Anordnung der Padzellen, entsprechend ihrer Schnittstellenfunktion, Bedingungen aus zwei Kontexten erfüllen. Einerseits die auch für die Zellen der Kernfläche geltenden Anforderungen bezüglich ihrer Verbindungsnetze und andererseits die Bedingungen für das Pinout des mit einem Gehäuse versehenen Chips. Wegen dieser Sonderstellung werden Pads oft nicht automatisch platziert, sondern ihre Anordnung vom Benutzer vorgegeben und z.B. aus einer Datei

eingelezen. In dieser Arbeit wird eine automatische Padplatzierung vorgenommen, um die Zahl der notwendigen Benutzerabfragen zu reduzieren.

In ESPRO werden die Padzellen nicht an beliebigen Positionen entlang der Chipkanten angeordnet, sondern auf äquidistant entlang der Kante angeordneten Stellen. Die Lage dieser Positionen ergibt sich aus der Größe der Chipfläche. Im weiteren wird eine mögliche Position für eine Padzelle als Slot bezeichnet. Aufgabe der Padplatzierung ist es, die Padzellen den verfügbaren Slots auf dem Padkranz zuzuordnen. Die Anzahl der Pads ist relativ zur Anzahl der Zellen klein, ihre Platzierung beeinflusst folglich nur eine kleine Anzahl von Netzen. Wegen dieses geringen Einflusses auf die Qualität des gesamten Layouts und da die Pads in realen Layouts oft nicht nach Kriterien des Chiplayouts, sondern vielmehr nach Richtlinien für die Anschlußbelegung des fertigen Chips platziert werden, wird ihnen hier eine geringere Bedeutung zugemessen und auf die Verwendung einer aufwendigen Heuristik verzichtet.

Der hier verwendete Platzierungsalgorithmus ordnet jede Padzelle der Chipkante zu, der auch das während der hierarchischen Partitionierung eingeführte externe Netz der Padzelle angehörte. Die Reihenfolge und Position der Padzellen entlang der Chipkante richtet sich nach der Lage der nächsten Nachbarzelle des Verbindungsnetzes der Padzelle. Durch diese Platzierung möglichst dicht am nächsten Nachbarn ihres Verbindungsnetzes, soll erreicht werden, daß die Netzlänge möglichst wenig beeinflusst wird.

Die Platzierung der Padzellen erfolgt im Anschluß an die Platzierung der Zellen in zwei Schritten. Zunächst wird jede Padzelle, wie soeben dargelegt, einer Kante des Chips zugeordnet und dort angeordnet. Hierbei wird die Lage und Anzahl der verfügbaren Slots unberücksichtigt gelassen. Im zweiten Schritt werden die Padzellen unter Beibehaltung der Reihenfolge den zur Verfügung stehenden Slots zugeordnet. Sind einer Chipkante mehr Pads zugeordnet als freie Slots verfügbar, werden die äußeren Pads auf die angrenzende Nachbarkante geschoben. Werden nicht alle verfügbaren Slots von Padzellen belegt, so werden die Pads auf den mittleren Slots der Kante angeordnet. In Abbildung 45 sind die Zwischenergebnisse einer Padplatzierung dargestellt.

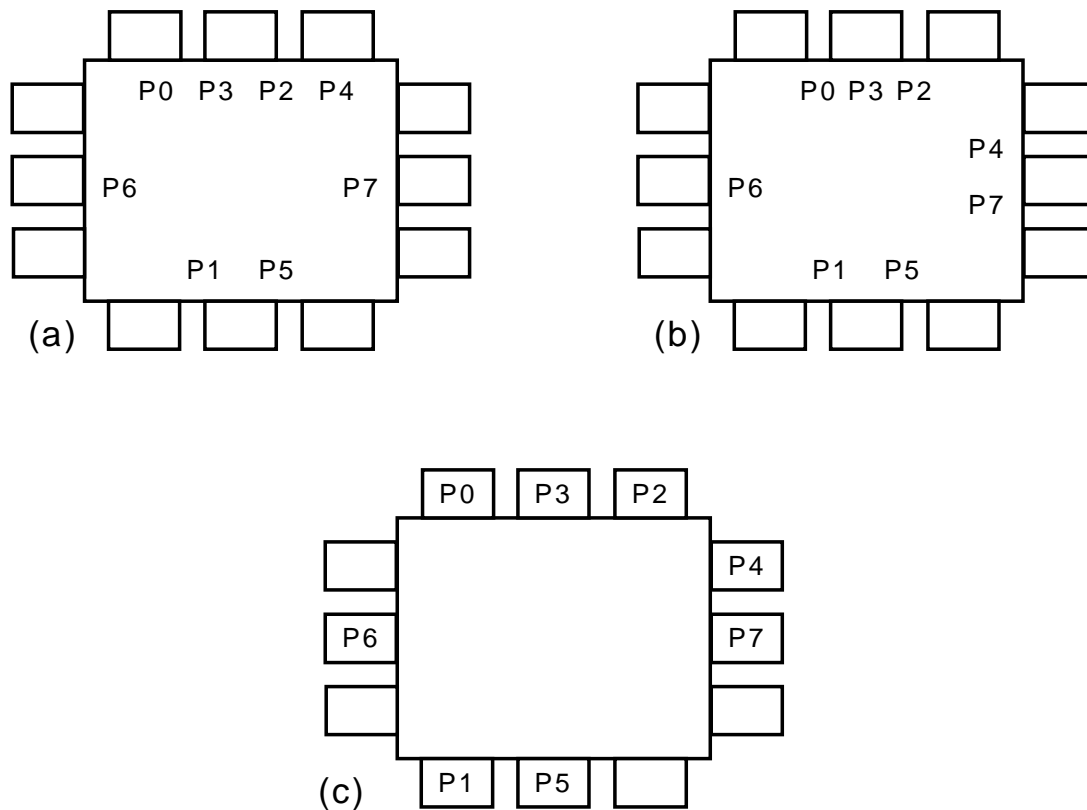


Abbildung 45: Zwischenergebnisse bei der Padplazierung. In (a) sind die Padzellen den Chipkanten zugeordnet und geordnet, in (b) wurden Pads so auf Kanten mit verfügbaren Slots verschoben, daß keine Kante überlastet ist. In (c) schließlich ist die Zuordnung der Pad zu den Slots abgeschlossen.

4.2.4.4 Ein Beispiel: Die Netzliste T4

Die in Abbildung 40.b gezeigte Anordnung von Zellen und Netzrepräsentanten soll im folgenden als Vorplazierung verwendet werden, um die Anpassung an den Standardzellentwurf für die Netzliste T4 darzustellen. Da hierfür nur die Positionen der Zellen benötigt werden, sind in den folgenden Abbildungen die Netzrepräsentanten und Padzellen nicht mit dargestellt.

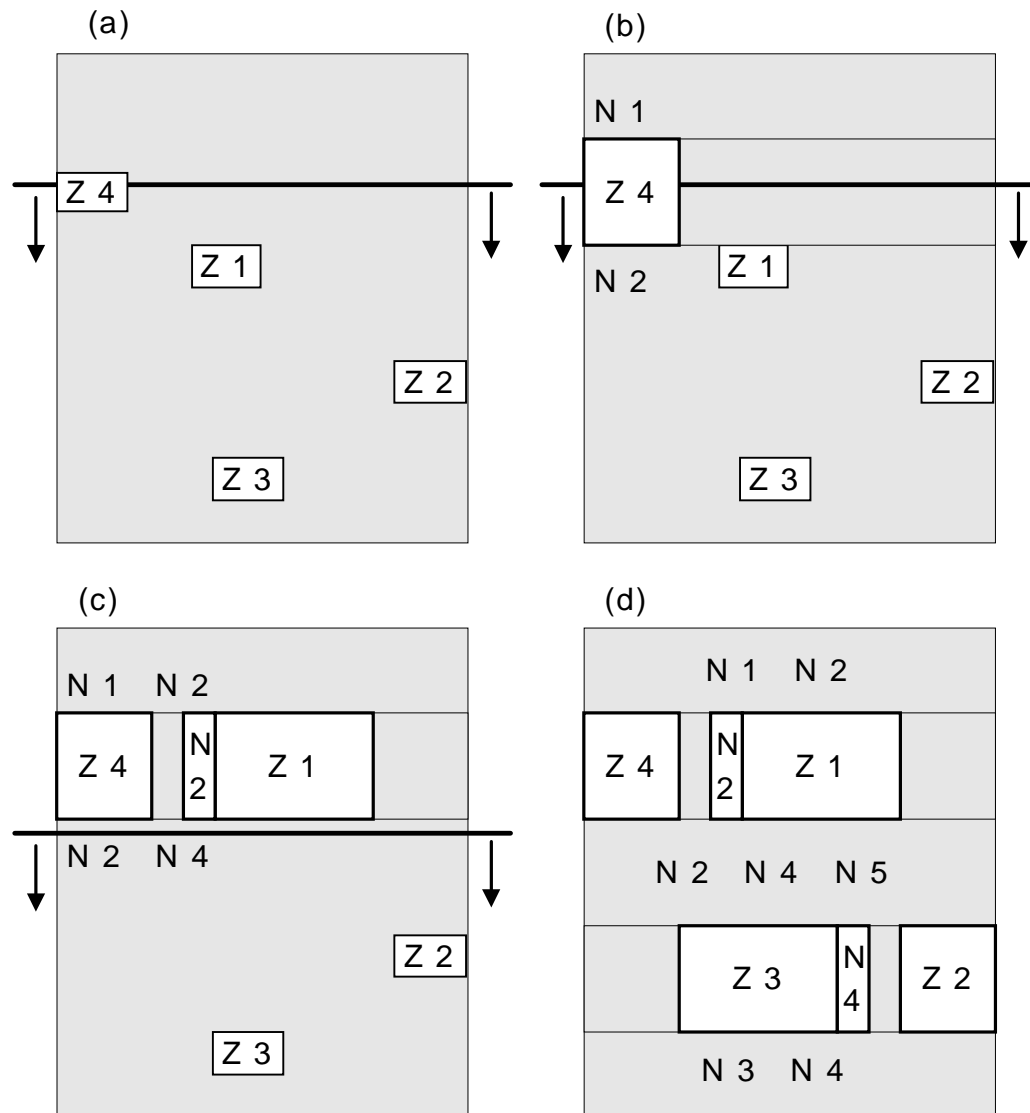


Abbildung 46: Die Schritte der Erststellung einer Standardzellplazierung für die Netzliste T4. Die Sweep-Line läuft von oben nach unten durch die Vorplazierung und reiht die Zellen auf.

Nach der Abschätzung der benötigten Chipfläche durchläuft die Sweep-Line von oben her die Vorplazierung. Die erste Zelle, die von der Sweep-Line überstrichen wird, ist, wie in Abbildung 46.a gezeigt, die Zelle Z 4. Diese Zelle wird nun zu ihrer richtigen Größe expandiert und in der aktuellen Zellreihe positioniert. Nachdem die Zelle Z 4 in die Zellreihe eingefügt wurde, beginnt die Bearbeitung der angeschlossenen Netze. Für Z 4 sind dies die Netze N 1 und N 2 (vergleiche Abbildung 5). Die Abbildung 46.b zeigt, wie Z 4 in die aktuelle Zellreihe eingefügt wurde sowie die im Kanal oberhalb und unterhalb der aktuellen Zellreihe präsenten Netze. Die nächste von der Sweep-Line erreichte Zelle ist Z 1, sie wird rechts von Z 4 in die aktuelle Zellreihe eingefügt. Eine Untersuchung der an Z 1 angeschlossenen Netze (das sind N 2, N 3, N 4 und N 5) ergibt, daß N 2 die aktuelle Zellreihe durchqueren muß, da es sowohl oberhalb als auch unterhalb der Zellreihe präsent ist. In Abbildung 46.c ist der Zustand nach dem Einfügen von Z 1 und der für Netz N 2 benötigten Feedthroughzelle dargestellt.

Nach dem Einfügen von Z 1 ist die erste Zellreihe komplett und es wird eine neue Reihe begonnen. Die beiden Zellen Z 2 und Z 3 werden in dieser Reihenfolge in die neue Zellreihe

eingefügt und nach dem Hinzufügen von Z3 wird eine Feedthroughzelle für Netz N4 vorgesehen. Die erzeugten Zellreihen und die Verteilung der Netze auf die Kanäle ist in Abbildung 46.d gezeigt.

Für eine vollständige Standardzellplazierung fehlen noch die Padzellen. Die beiden Pads werden jeweils der Chipkante zugeordnet, zu welcher der Repräsentant ihres externen Netzes gehört. Wie aus Abbildung 40.b zu entnehmen ist, wird Pad1 der linken und Pad2 der rechten Chipkante zugeordnet.

Die Abbildung 47 zeigt die Standardzellplazierung für die Netzliste T4. Die Information über die Zuordnung von Netzen zu Kanälen wurde nur für die Ermittlung des Bedarfes an Feedthroughs benötigt und ist nicht Teil der Plazierung. Auch die Lage der vorgesehenen Feedthroughs ist nicht Teil der Plazierung, es werden statt der Feedthroughs nur Freiräume in den Zellreihen gelassen, die von einem Verdrahtungsalgorithmus wiederum für Feedthroughs verwendet werden können.

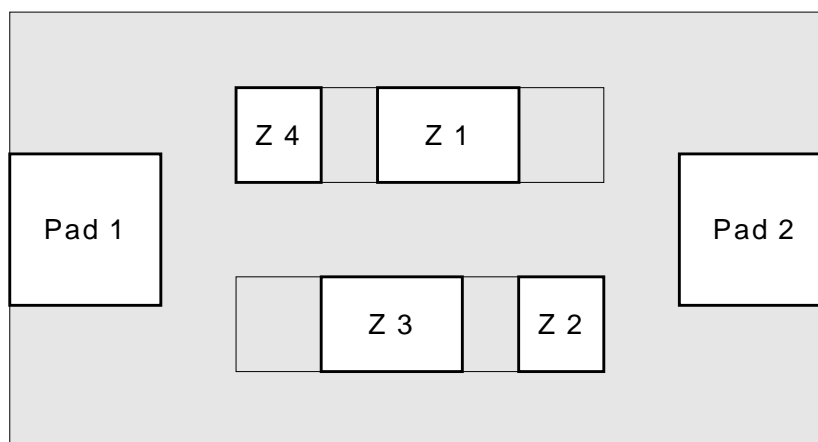


Abbildung 47: Die Standardzellplazierung für die Netzliste T4

Nach Abschluß der in diesem Abschnitt 4.2.4 beschriebenen Verarbeitungsschritte ist eine vollständige Standardzellenplazierung für ESPRO entstanden. Diese kann direkt weiterverwendet werden und als Ausgangssituation für den Verdrahter dienen und/oder in einer Datei abgelegt werden.

4.3 Details zu Teilproblemen von der Platzierung zum fertigen Layout

Dieses Teilkapitel widmet sich den durchzuführenden Ablaufschritten, um ausgehend von einer gegebenen Platzierung ein gültiges Standardzelllayout zu erhalten. Wie in Kapitel 4.2 sollen die verwendeten Algorithmen beschrieben und die getroffene Auswahl motiviert werden, ohne auf Implementationsdetails einzugehen. Eine Übersicht der für die Verdrahtung durchzuführenden Teilschritte ist in Abschnitt 4.1.2 und in Abbildung 12 gegeben.

Die Aufgabe der Verdrahtung kann in zwei Teile zerlegt werden, die Global- und die Detailverdrahtung. Bei der Globalverdrahtung wird für jedes Netz bestimmt, durch welche Verdrahtungsregion es verläuft, sowie die Orte definiert, an denen es in die einzelnen Regionen eintritt bzw. sie verläßt. Die Detailverdrahtung bestimmt den genauen Verlauf der Leitungssegmente innerhalb der vorab definierten Verdrahtungsregionen. Die Verdrahtungsregionen sind beim Standardzellentwurf die Kanäle zwischen den Zellreihen, sowie zwischen dem Padkranz und dem Kerngebiet (Core Area). Für die Detailverdrahtung ist nur der Spezialfall der Kanalverdrahtung zu betrachten. Die Kanalverdrahtung ist ein wohlverstandenes Teilproblem des physikalischen VLSI-Entwurfs, für das effektive Algorithmen existieren ([She95], [Het98]). Daher wird dieses Teilproblem hier nicht näher betrachtet, sondern ein erprobter Algorithmus, der Greedy Kanalverdrahter, verwendet.

Im weiteren Verlauf dieses Teilkapitels wird zunächst das allgemeine Vorgehen der Globalverdrahtung genauer dargelegt und das zellweise Vorgehen motiviert. In den folgenden Abschnitten werden die Teilschritte der Globalverdrahtung näher erläutert. Nach Schilderung der Globalverdrahtung werden die Probleme der Überprüfung des Netzzusammenhangs und der Globalverdrahtung der Padzellen erläutert. Das Ende des Kapitels bilden die Abschnitte über die Kanalverdrahtung und den Abschluß des Layouts.

4.3.1 Globalverdrahtung

In bisherigen sequentiellen Verdrahtern wird, ausgehend von dem durch den Graphen $G(V,E)$ gegebenen Problem, die Globalverdrahtung für ein Netz nach dem anderen durchgeführt. Dies entspricht einem Bearbeiten aller Elemente der Kantenmenge E des gegebenen Graphen $G(V,E)$. In dieser Arbeit kommt die in vorherigen Schritten bereits verwendete Darstellung durch den dualen Graphen $Y(U,H)$ auch in diesem Schritt zur Anwendung. Die Globalverdrahtung erfolgt wiederum sequentiell durch die Betrachtung aller Elemente der Kantenmenge, hier jedoch der Menge H des dualen Graphen $Y(U,H)$. Übertragen auf das ursprüngliche Problem bedeutet dies, daß die Verdrahtung zellweise und nicht netzweise erfolgt.

Durch die Bearbeitungsreihenfolge bei der netzweisen Verdrahtung werden generell die spät verdrahteten Netze gegenüber den früh verdrahteten Netzen benachteiligt. Das Verdrahtungsproblem ist damit nicht reihenfolgeninvariant. Die Wahl der Reihenfolge, in der die Netze bearbeitet werden, ist daher wichtiger Bestandteil der Verdrahtungsheuristik. Bei der zellweisen Verarbeitung wird jedes einzelne Netz in mehreren Etappen bearbeitet. Dadurch wird die Diskriminierung der spät verdrahteten Leitungssegmente gleichmäßiger auf die Netze verteilt. Die unvermeidbare entstehende Benachteiligung der spät

verdrahteten Zellen verteilt sich auf die an die Zelle angeschlossenen Netze und wirkt sich zudem nur auf das zur Zelle führende Teilnetz aus. Es wird erwartet, daß durch dieses Vorgehen die zuletzt bearbeiteten Netze weniger benachteiligt werden, was sich in kürzeren maximalen Netzlängen zeigen müßte.

Die netzweise Verdrahtung erfordert bei Mehrpunktnetzen das Erstellen komplexer Verdrahtungsgraphen (Routinggraphen). Zum Finden günstiger Verdrahtungswege sind darin die Steiner Bäume der Routinggraphen (STOGP: Steiner Tree On Graph Problem) zu bestimmen, meist wird der Spezialfall der rechtwinkligen Steinerbäume betrachtet [She95]. Da das STOGP NP-hart ist [GaJ79], werden Heuristiken zu seiner Lösung verwendet. Ein gängiges Vorgehen ist es, die Suche nach dem optimalen Steiner Baum durch die Suche nach anderen, leichter bestimmbareren Graphen, z.B. dem minimalen aufspannenden Baum (Minimal Spanning Tree, MST), zu ersetzen. Die gefundenen Ergebnisse sind in ihrer Qualität bekannt und im Falle des MST in polynomieller Zeit zu bestimmen [Ger99].

Das hier verwendete zellweise Verdrahten führt die Verdrahtung der Netze auf die Verdrahtung mehrerer Zweipunktnetze zurück. Dabei wird auf die explizite Suche von Steiner Bäumen gänzlich verzichtet und die einzufügenden Feedthroughs als Steinerpunkte verwendet. Die ausschließliche Betrachtung von Zweipunktnetzen reduziert die Problemkomplexität der Verdrahtung und ermöglicht es, verhältnismäßig einfache Algorithmen zur Suche des kürzesten Weges (Shortest Path) zu verwenden.

Im Standardzellentwurf ist es, zusätzlich zum Zuordnen der Netze zu den Verdrahtungsregionen, eine der Aufgaben des Globalverdrahtens, die Feedthroughs einzufügen oder, wenn sie bereits existieren, diese den Netzen zuzuordnen. Durch die Auswahlmethode für die zwei zu verbindenden Pins (vergleiche Abschnitt 4.3.1.1), ergibt sich, daß der hier verwendete Verdrahter die Anzahl der Feedthroughs minimiert.

Es ist dem Verdrahtungsalgorithmus möglich lokale Veränderungen an der Platzierung vornehmen. Diese sind hier auf Zellverschiebungen innerhalb einer Zellreihe beschränkt und auch nur dann möglich, wenn es in der Zellreihe noch Lücken gibt, so daß Verschiebungen möglich sind. Eine Vergrößerung der Chipfläche in Richtung der Zellreihenorientierung ist dem Verdrahter nicht gestattet. Zellen dürfen verschoben werden, um Zwischenräume für Feedthroughzellen zu schaffen oder um Start- und Zielzelle des aktuell betrachteten Zweipunktnetzes näher bei einander zu platzieren. Das Maß an erlaubten Veränderungen kann vorab durch Parameter gesteuert werden (siehe Abschnitt 5.3.2.1). Folglich ergibt sich die endgültige Platzierung erst nach Abschluß aller Globalverdrahtungsschritte.

Die zellweise Betrachtung des Verdrahtungsproblems erzeugt nach der Bearbeitung aller Zellen nicht zwangsläufig eine vollständige Globalverdrahtung. Es ist möglich, daß ein Netz zwischenzeitlich in zwei oder mehr unverbundene Teile geteilt ist. Eine abschließende Prüfung aller Netze (siehe Abschnitt 4.3.1.6) garantiert die vollständige Verbindung der Netze.

Die folgenden Abschnitte beschreiben das Vorgehen des Verdrahters für ein Zweipunktnetz. Dies beinhaltet neben dem eigentlichen Erstellen der Verbindung die Auswahl der beiden zu verbindenden Pins, sowie gegebenenfalls das Durchführen von Veränderungen an der Platzierung. Die Globalverdrahtung betrachtet zunächst nur die Zellen der Core Area, keine Padzellen. Padzellen können keine Startzellen für eine

Zweipunktnetzverdrahtung sein, wohl aber können Padzellen der oberen oder unteren Chipkante Zielzellen sein. Die Globalverdrahtung der Pads erfolgt in einem separaten Schritt (vergleiche Abschnitt 4.3.3)

4.3.1.1 Auswahl von Start- und Zielpin des Zweipunktnetzes

Die Auswahl von Start- und Zielpin des zu verdrahtenden Zweipunktnetzes bestimmt die Reihenfolge, in der die Teilnetze bearbeitet werden, und damit hat sie, wegen der Reihenfolgeninvarianz des sequentiellen Verdrahtens, Einfluß auf das entstehende Layout.

Die Auswahl des Startpins erfolgt hier nach aufsteigender Indexnummer des Pins an der zugehörigen Zelle. Es bleibt die Reihenfolge der Zellen zu bestimmen. Bei netzweise sequentiellen Verdrahtungen wird die Reihenfolge der Netze durch eine Bewertungsfunktion, die abhängig von der geschätzten Netzlänge und der Anzahl der verbundenen Pins sein kann (vergleiche [She89], [Die89]), bestimmt. Die Bewertung soll dabei Netze mit kritischer Länge bevorzugen, um diese möglichst ohne Umwege zu verdrahten. Netze, die unproblematisch erscheinen, werden dadurch spät verarbeitet, so daß für diese Netze Umwege in Kauf genommen werden. Da beim zellweise sequentiellen Ansatz jeweils nur die von einer Zelle ausgehenden Teilnetze betrachtet werden können, ist das, auf der Bewertung der kompletten Netze basierende, Vorgehen aus der netzweise sequentiellen Verdrahtung nicht direkt auf die zellweise sequentielle Bearbeitung übertragbar.

In dem hier gewählten Algorithmus wird eine Liste der zu verdrahtenden Zellen aufgebaut. Zu Beginn der Verdrahtung enthält die Liste diejenigen Zellen, die mit einer der Padzellen verbunden sind. Beim Verdrahten der Pins einer Zelle wird nun jeweils die Zelle, die zu den ausgewählten Zielpins der Zweipunktnetze gehört, an diese Liste angehängt. Dieses Vorgehen fügt jede Zelle mindestens einmal in die Liste ein, sofern die Netzliste zusammenhängend ist, wovon ausgegangen werden kann. Als nächste zu verdrahtende Zelle wird jeweils das erste Element der Liste genommen und nach vollständiger Verdrahtung der Zelle wird sie vom Kopf der Liste entfernt.

Die Auswahl des Zielpins erfolgt unter allen dem Netz zugehörigen Pins, dies beinhaltet auch die Pins der bereits vom Verdrahter eingefügten und damit fest an ein Netz gebundenen Feedthroughs, so vorhanden. Es wird der Pin als Ziel verwendet, zu dessen Erreichen die wenigsten Kanalwechsel nötig sind. Liegen mehrere Pins im gewählten Zielkanal, so werden alle als mögliche Ziele gewählt und die endgültige Bestimmung des Zielpins erst bei der Bewertung des komplett aufgebauten Routinggraphen (vergleiche Abschnitt 4.3.1.4) durchgeführt.

Diese Auswahlmethode erzeugt keine Feedthroughs in Zellreihen, in der für das betrachtete Netz bereits ein Feedthrough vorhanden ist, und minimiert somit die Anzahl der verwendeten Feedthroughs.

4.3.1.2 Verschieben von Start- und / oder Zielzelle

Ist die Wahl von Start- bzw. Zielpin(s) getroffen, erfolgt die Überprüfung, ob ein Verschieben der beiden betroffenen Zellen möglich und vorteilhaft ist. Die Position der Zielzelle wird nach der Wahl des Verdrahtungsweges überprüft. Da das Verfahren für Start- und Zielzelle identisch ist, wird hier beides beschrieben.

Die Beurteilung der aktuellen Position im Vergleich zu alternativen Plätzen muß unter Berücksichtigung aller an die Zelle angeschlossenen Netze erfolgen. Als Bewertungsmaßstab bieten sich die gleichen Kriterien an, die für den gesamten Verdrahtungsprozeß als Maßstab dienen, also die bei der Beurteilung des Routinggraphen (vergleiche Abschnitt 4.3.1.4) verwendeten Bewertungsfunktionen. Wird diese Methode angewandt, sind die folgenden Betrachtungen notwendig:

Für bereits verdrahtete Pins der Zellen muß für eine mögliche Alternativposition die Veränderung der Netzlänge, unter Beibehaltung des gewählten Verdrahtungspfades, betrachtet werden. Für noch nicht verdrahtete Pins ist eine Abschätzung des Einflusses auf die zu erwartende Netzlänge nötig. Zusätzlich zu den Netzlängenänderungen der bereits verdrahteten Netze entlang des Kanals, kann die Verschiebung einer Zelle die Kanalauslastung ändern [Die89]. Diese Betrachtungen zur Kanalauslastung werden hier nicht weiter behandelt.

Während die Ermittlung der Netzlängenänderung für bereits festgelegte Verdrahtungspfade einfach zu bestimmen ist, erfordert die Beurteilung des Einflusses einer Zellbewegung auf noch frei wählbare Verdrahtungsrouten eine Bewertung aller möglichen Alternativrouten. Die Alternativrouten beinhalten alle Routen zu allen Pins des betroffenen Netzes. Bei einer so detaillierten Betrachtung der noch unverdrahteten Netze sollte jedoch die Wahl des Verdrahtungspfades der bereits verdrahteten Pins ebenfalls neu zur Disposition gestellt werden. Für die Wahl und Beurteilung der möglichen Verdrahtungswege wäre ein Vorgehen entsprechend dem in 4.3.1.3 und 4.3.1.4 beschriebenen Routinggraphen und dessen Auswertung nötig.

Wegen des daraus resultierenden erheblichen Aufwandes, ist es angebracht, eine exakte Bestimmung des Einflusses der Zellverschiebung durch eine Approximation zu ersetzen und im Zweifelsfall auf eine Verschiebung zu verzichten. Für diese Approximation wird die Lage der Netzmittelpunkte aus $Y(U,H)$ verwendet. Für die approximative Beurteilung der möglichen Positionen wird ihre Lage relativ zum Netzmittelpunkt betrachtet. Eine Bewegung der Zelle weg vom Mittelpunkt eines angeschlossenen Netzes wird als ungünstig angesehen, eine Bewegung auf den Netzmittelpunkt zu als vorteilhaft. Da die Wahl des Verdrahtungsweges für die noch nicht verdrahteten Pins die neue Position der Zellen berücksichtigt, dies aber nicht für die bereits verdrahteten Pins erfolgt, trifft die Approximation über den Abstand zum Netzmittelpunkt für bereits verdrahtete Teilnetze nicht zu. Die Beurteilung der Bewegung bereits teilweise verdrahteter Zellen bedarf daher weiterer Betrachtungen, wie sie oben beschrieben und als zu aufwendig eingestuft wurden. Es wird hier deshalb, zu Gunsten einer einfachen Bewertungsfunktion, auf die Verschiebung von Zellen mit bereits verdrahteten Pins völlig verzichtet.

Der Aufwand zur Bewertung der möglichen Alternativpositionen ist ebenfalls von der Zahl der betrachteten Positionen abhängig. Um dies in einem angemessenen Rahmen zu halten, darf der Suchradius nach geeigneten Zellpositionen nicht zu groß gewählt werden. Insbesondere ein Zellreihenwechsel erhöht den Bewertungsaufwand, da die einfache Approximation über die Lage relativ zum Netzmittelpunkt hier ungeeignet ist. Ein Verschieben einer Zelle in eine andere Zellreihe hat Konsequenzen für die Feedthroughs aller angeschlossenen Netze, es werden dadurch erheblich veränderte Verdrahtungswege verwendet, wobei sich die Kanalauslastung verändert. Daher wird die Suche nach möglichen Alternativpositionen auf die aktuelle Zellreihe beschränkt. Die Suchradien für

die Startzelle und die Zielzelle können mit entsprechenden Parametern getrennt eingestellt werden.

4.3.1.3 Erstellen des Routinggraphen

Der verwendete Routinggraph ist ein gerichteter azyklischer Graph (DAG: Directed Acyclic Graph). Die Knoten repräsentieren die zu verbindenden Pins. Die Knotenmenge wird, abgesehen vom Startpin und den Zielpins, während des Aufbaus des Graphen ermittelt. Die Kanten beinhalten als Gewicht ein Tupel, das aus für die Verdrahtung relevanten Werten besteht. Dies sind hier die Länge der Kante entlang der Ausbreitungsrichtung des Kanals, die durchschnittliche und die maximale Kanalauslastung im von der Kante belegten Intervall. Die Knoten erhalten ebenfalls Gewichte, die den Abstand des Knotens vom Netzmittelpunkt wiedergeben.

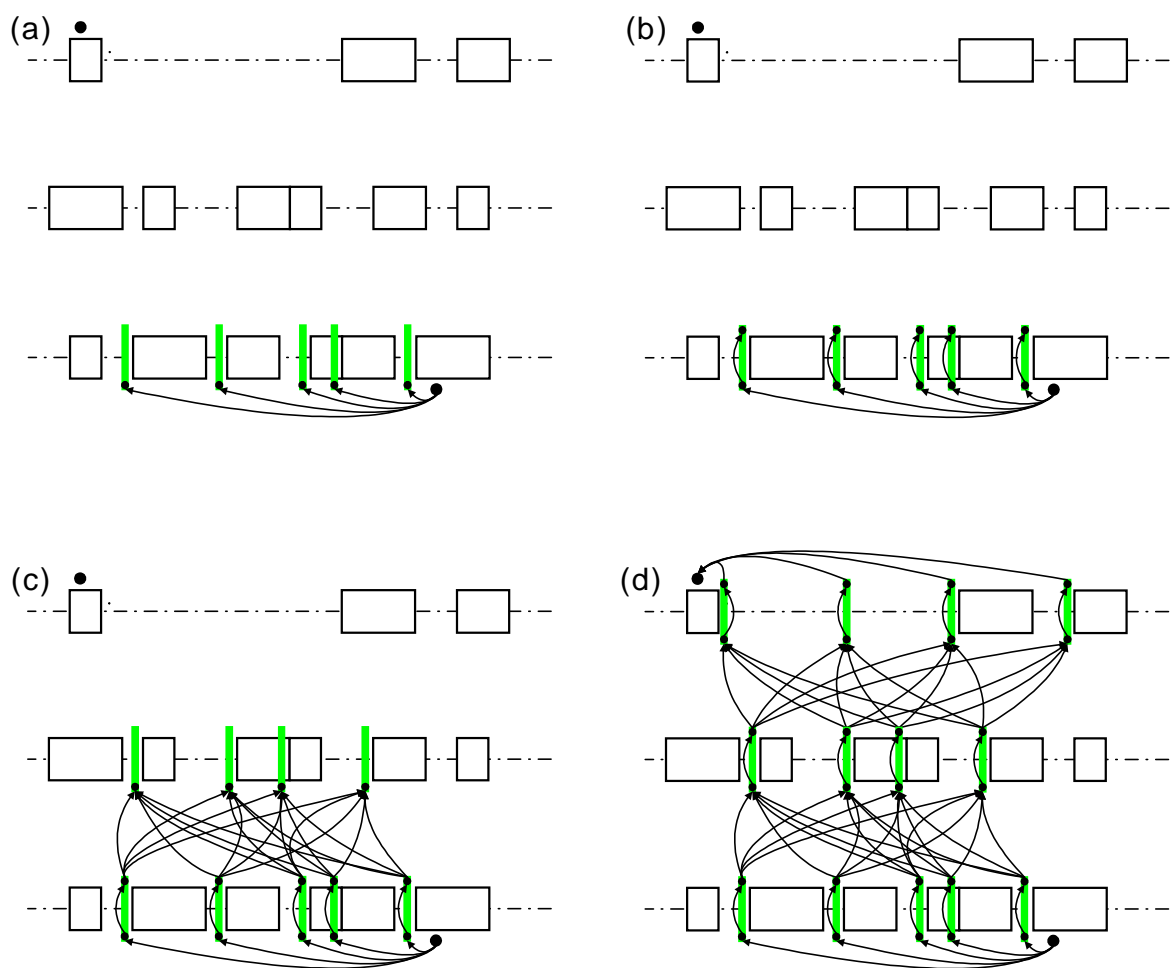


Abbildung 48: Aufbau eines Routinggraphen für einen gegebenen Startpin und einen Zielpin. Beginnend beim Startpin wird die erste Reihenebene zu den Feedthroughkandidaten gebildet (a) und die unterste Zellreihe mit einer Reihenebene von Kanten durchquert (b). Die Bilder (c) und (d) zeigen weitere Stadien der Graphenkonstruktion: In (c) ist die zweite Reihenebene erzeugt worden, in (d) ist der vollständige Routinggraph dargestellt. Die dargestellten Zellreihendurchquerungen sind *mögliche* Feedthroughpositionen, die gegebenenfalls noch durch Zellverschiebungen geschaffen werden müssen.

Der Aufbau des Graphen erfolgt, beginnend beim Startpin, in Form einer Breitensuche. Dabei bilden jeweils die Knoten des Graphen, die zu Pins eines Kanals (des aktuellen

Kanals) gehören, die Ausbreitungsfront, von der aus der Graph weiterentwickelt wird. Die Knoten der Ausbreitungsfront werden untereinander nicht verbunden. Die erste Ausbreitungsfront wird durch den Startpin gebildet.

Ist der aktuelle Kanal nicht der Kanal, in dem sich die Zielpins befinden (der Zielkanal), so werden in der Zellreihe, die sich in Ausbreitungsrichtung des Graphen befindet, sogenannte Feedthroughkandidaten (Candidate Feedthrough) vorgesehen. Die Positionen dieser Feedthroughkandidaten ergeben sich aus den vom Plazierer vorgesehenen Lücken in den Zellreihen, sowie solchen Positionen, an denen durch Verschieben von Zellen eine Lücke geschaffen werden kann, sofern dies zugelassen ist. Die durch die hierfür nötigen Verschiebungen von Zellen möglicherweise hervorgerufenen Änderungen der Kanalauslastungen [Die89] werden nicht betrachtet. Von diesen Feedthroughkandidaten werden zunächst nur die Pins im aktuellen Kanal betrachtet, die zur Repräsentation dieser Pins entstehenden Knoten des Graphen erhalten als Gewicht ihre Entfernung zum Netzmittelpunkt.

Die Pins der Feedthroughkandidaten im aktuellen Kanal werden nun mit den Pins der Ausbreitungsfront verbunden (vergleiche Abbildung 48.a). Die so entstandene Ebene von Kanten verläuft durch einen Kanal und enthält alle Möglichkeiten den Kanal in Richtung auf die Zielpins zu durchqueren, im folgenden auch als Kanalebene (Channel Level) des Graphen bezeichnet. Die Kanten der Kanalebene erhalten die oben erwähnten Gewichtstupel, bestehend aus ihrer Länge und mittlerer sowie maximaler Auslastung des Kanals auf dem von ihnen verwendeten Abschnitt.

Die Feedthroughs werden durch Kanten zwischen ihren beiden (in verschiedenen Kanälen liegenden) Pins repräsentiert. Es werden nun die Knoten für die Pins der Feedthroughkandidaten im nächsten Kanal erzeugt. Von den beiden Knoten, die einen Feedthroughkandidaten repräsentieren, erhält einer als Gewicht seine Entfernung zum Netzmittelpunkt, der andere das Gewicht Null. Die durch Verbinden der beiden jeweils zum selben Feedthroughkandidaten gehörenden Knoten für alle Feedthroughkandidaten einer Zellreihe entstehende Kantengruppe wird im folgenden als Reihenebene (Row Level) des Routinggraphen bezeichnet (vergleiche Abbildung 48.b). Die Existenz der Reihenebene impliziert, daß die Verwendung eines Feedthroughs unabdingbar ist. Somit erhielten alle Kanten einer Reihenebene das gleiche Gewicht (alle Feedthroughs sind gleich hoch), wodurch sie keinen Einfluß auf die Rangfolge der Wegequalitäten hätten. Kanten der Reihenebene erhalten daher als Gewicht Null.

Die Komplettierung der Reihenebene erzeugt eine neue Ausbreitungsfront des Graphen, deren Pins sämtlich zum nächsten in Ausbreitungsrichtung liegenden Kanal gehören.

Ist der Zielkanal erreicht, wird die letzte Kanalebene eingeführt, die alle Knoten der Ausbreitungsfront mit den Zielpins verbindet, ist der Zielkanal nicht erreicht, wiederholt sich das Vorgehen.

Der Algorithmus zur Erstellung eines Routinggraphen für ein Zweipunktnetz ist in Abbildung 49 wiedergegeben, ein Beispiel für den Ablauf ist in Abbildung 48 dargestellt.

Aufbau des Routinggraph $R(P,K)$

- 0) Gegeben: Startpin S im Startkanal C_S , Zielpinmenge $Z = \{Z_1, \dots, Z_n\}$ im Zielkanal C_Z
 - 1) Setze Knotenmenge $P = \{S\}$ und Kantenmenge $K = \emptyset$, aktueller Kanal C ist Kanal des Startpins $C := C_S$, Frontmenge $F := \{S\}$
 - 2) Wenn $C = C_Z$ weiter bei 9)
 - 3) Ermittle Menge von Feedthroughkandidaten CFT der Zellreihe, die den aktuellen Kanal C in Richtung auf den Zielkanal begrenzt
 - 4) Erzeuge Menge CFT^A von Knoten für die Pins der Feedthroughkandidaten aus CFT , die im aktuellen Kanal liegen
 - 5) Erzeuge Kanten der Kanalebene $K_C = \{ (a, w, b) \mid \forall a \in F, \forall b \in CFT^A, w: \text{Kantengewicht} \}$, Erweitere Knotenmenge $P := P \cup CFT^A$, Erweitere Kantenmenge $K := K \cup K_C$. Setze neue Frontmenge $F := CFT^A$
 - 6) Erzeuge Menge CFT^B von Knoten für die Pins der Feedthroughkandidaten aus CFT , die im nächsten Kanal in Richtung auf den Zielkanal liegen
 - 7) Erzeuge Kanten der Reihenebene $K_R = \{ (a, w, b) \mid \forall a \in CFT^A, \forall b \in CFT^B, \text{mit } a \text{ und } b \text{ gehören zum selben Feedthroughkandidaten, } w: \text{Kantengewicht} \}$, Erweitere Knotenmenge $P := P \cup CFT^B$, Erweitere Kantenmenge $K := K \cup K_R$. Setze neue Frontmenge $F := CFT^B$. Setze aktuellen Kanal C auf den Kanal, zu dem die Knoten aus CFT^B gehören
 - 8) Wenn $C \neq C_Z$ weiter bei 3)
 - 9) Erzeuge Kanten der letzten Kanalebene $K_C = \{ (a, w, b) \mid \forall a \in F, \forall b \in Z, w: \text{Kantengewicht} \}$, Erweitere Knotenmenge $P := P \cup Z$, Erweitere Kantenmenge $K := K \cup K_C$
- ENDE

Abbildung 49: Algorithmus für die Erstellung eines Routinggraphen für ein Zweipunktnetz.

4.3.1.4 Auswahl des verwendeten Pfades

Zur Auswahl des günstigsten Verdrahtungsweges wird im wie oben dargelegt erzeugten Routinggraph der Pfad vom Startpin S zu einem der Zielpins $Z_i \in Z$ mit den geringsten Kosten gesucht. Die Kosten eines Pfades ergeben sich aus der Summe der Kosten der verwendeten Kanten und passierten Knoten. Die verwendeten Kostenfunktionen für die Beurteilung von Kanten und Knoten bestimmen maßgeblich das Verhalten des Verdrahters.

Die Bewertungsfunktion für die Kanten ist die gewichtete Summe der drei Komponenten des Kantengewichtstupels, sowie dem Produkt aus mittlerer Kanalauslastung und Kantenlänge. Diese Summe wird dann mit einem Skalierungsfaktor multipliziert, bevor sie zur Pfadbewertung herangezogen wird (vergleiche Gleichung 3). Die Bewertungsfunktion ist eine Erweiterung der in [Die89] vorgeschlagenen Gewichtsfunktion, hier werden weitere Komponenten hinzugenommen, die durch die Wahl der Parameter mehr oder weniger Einfluß erhalten können. Die Bewertung der Knoten ergibt sich aus ihrem Gewicht, multipliziert mit einem Skalierungsfaktor (vergleiche Gleichung 4). Die Parameter für die Bewertungsfunktionen können beim Programmstart eingestellt werden (siehe Anhang).

$$\begin{aligned} \text{EdgeCost} = & (\text{Length} * (\text{EdgeRatingLengthWeight} \\ & + \text{AvgChannelUtilization} * \text{EdgeRatingLengthUtilizationWeight}) \\ & + \text{AvgChannelUtilization} * \text{EdgeRatingAvgUtilWeight} \\ & + \text{MaxChannelUtilization} * \text{EdgeRatingMaxUtilWeight}) \\ & * \text{parEdgeWeight} \end{aligned}$$

Gleichung 3: Bewertungsfunktion für eine Kante des Routinggraphen.

$$\begin{aligned} \text{NodeCost} = & (\text{XDistanceFormNetCentre} + \text{YDistanceFormNetCentre}) \\ & * \text{parEdgeWeight} \end{aligned}$$

Gleichung 4: Bewertungsfunktion für einen Knoten des Routinggraphen.

Für die Suche nach dem kürzesten Pfad in einem Graphen sind viele Algorithmen verfügbar (vergleiche z.B. [Des87], [Jun94]). Hier kommt ein Verfahren ähnlich dem von Moore vorgestellten Algorithmus zum Einsatz (siehe [Die89]). Die Kosten der Pfade im Routinggraph werden rekursiv durch Traversierung des kompletten DAG ermittelt. Nach Abschluß der Markierung des Graphen enthält jede Kante und jeder Knoten die Kosten, die der günstigste Pfad zu einem Zielpin über diesen Knoten oder diese Kante verursacht. Für die weitere Bearbeitung wird der kürzeste Pfad außerdem markiert.

4.3.1.5 Verschieben der Zielzelle und Einfügen der Feedthroughzellen

Ist der Pfad mit minimalen Kosten gefunden, ist auch bekannt, welche der möglichen Zielzellen verwendet wurde. Die Lage dieser Zelle wird nun auf Verbesserungsmöglichkeiten überprüft. Dabei wird wie unter 4.3.1.2 beschrieben vorgegangen.

An den Stellen der für den gewählten Pfad verwendeten Feedthroughkandidaten werden die endgültigen Feedthroughs für das Netz eingefügt. Die dafür gegebenenfalls nötigen Zellverschiebungen und Änderungen der Zellreihen werden durchgeführt. Diese Feedthroughs können nun bei der Bearbeitung eines weiteren Zweipunktnetzes dieses Netzes als Zielpins gewählt werden. Abschließend werden die Eigenschaften des soeben teilweise verdrahteten Netzes aktualisiert, dies beinhaltet unter anderem die Lage des Netzmittelpunktes, sowie die Statistik, welches Netz sich in welchem Kanal befindet. Dabei haben Feedthroughs keinen Einfluß auf die Lage des Netzmittelpunktes. In Abbildung 50 ist ein kompletter Routinggraph zu sehen, in dem bereits feste Feedthroughzellen für den besten Weg eingefügt sind.

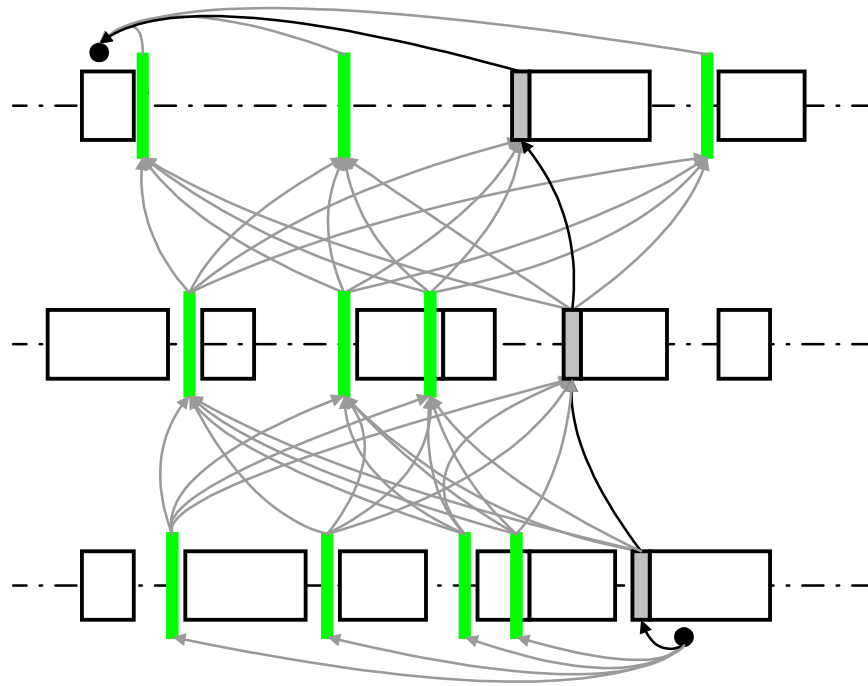


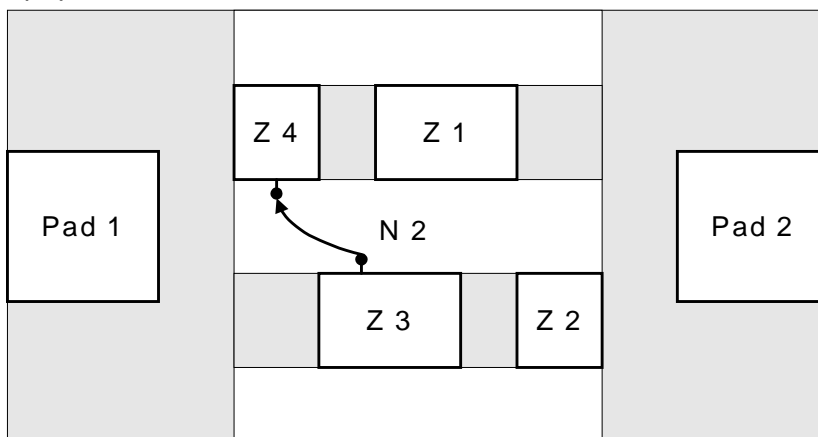
Abbildung 50: Der Routinggraph eines Zweipunktnetzes, die Feedthroughs des ausgewählten Pfades sind als Zellen eingefügt und grau ausgefüllt dargestellt. Die hellgrauen Pfeile zeigen die nicht gewählten Verdrahtungsalternativen. Die Kanten der Kanalebenen sind nicht dargestellt, nur die Feedthroughkandidaten.

4.3.1.6 Ein Beispiel: Die Netzliste T4

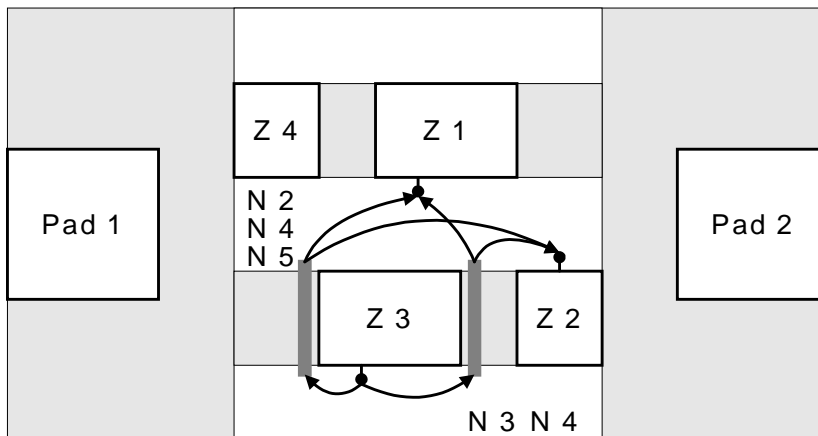
Anhand der in Abbildung 51 gezeigten Stadien soll hier nun die Entstehung der Globalverdrahtung für die Netzliste T4 illustriert werden.

Die beim zellweisen Vorgehen zuerst betrachteten Zellen sind diejenigen, die über ein Netz mit einer Padzelle verbunden sind. Die Zelle Z3 ist über das Netz N5 mit Pad1 verbunden. Die Globalverdrahtung für die Netzliste T4 beginnt daher mit der Zelle Z3. Pin0 von Z3 im mittleren Kanal wird mit Pin1 von Zelle Z4 auf der gegenüberliegenden Kanalseite verbunden, damit ist N2 im mittleren Kanal präsent (vergleiche Abbildung 51.a). Für Pin1 von Z3 ist eine Zellreihendurchquerung nötig, der sich ergebende Routinggraph ist in Abbildung 51.b gezeigt, die Reihenebenen sind, wie in Abbildung 50, nur als Feedthroughkandidaten dargestellt. Die Festlegung der verwendeten Feedthroughzelle und die Komplettierung der Globalverdrahtung der an Z3 angeschlossenen Teilnetze ist in Abbildung 51.c gezeigt.

(a)



(b)



(c)

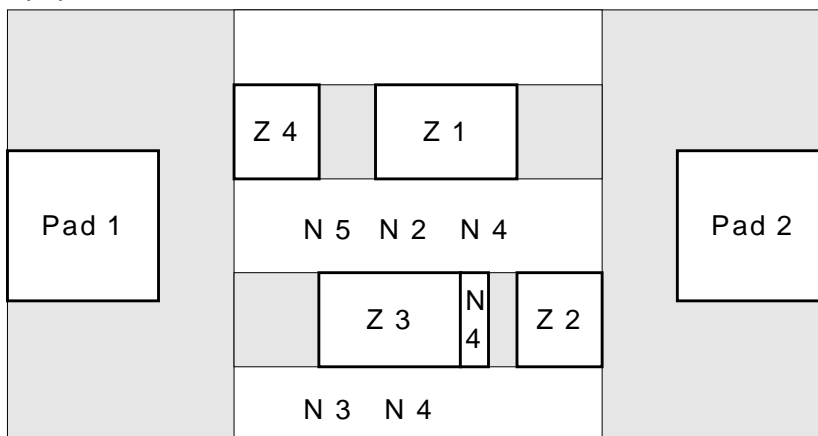


Abbildung 51: Die Globalverdrahtung beginnt mit der Betrachtung der an Z3 angeschlossenen Netze. In (a) ist Pin0 von Z3 verdrahtet, in (b) ist der Routinggraph für die Globalverdrahtung des an Pin1 angeschlossenen Netzes N4 dargestellt und (c) zeigt die Situation nach vollständiger Betrachtung von Z3.

Die weitere Globalverdrahtung erfolgt in der Reihenfolge Z2, Z1 und abschließend Z4. Die nach der Bearbeitung von Z4 entstehende Verteilung der Netze auf die Kanäle sowie die eingefügten Feedthroughs sind in Abbildung 52 zu sehen.

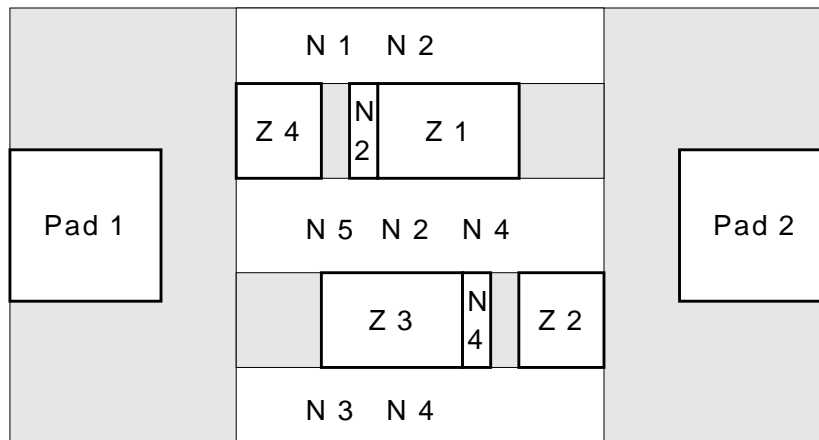


Abbildung 52: Die Globalverdrahtung und die durch eingefügte Feedthroughzellen modifizierte Platzierung zu der Netzliste T4.

Nach Abschluß der in diesem Abschnitt 4.3.1 dargelegten Verarbeitungsschritte ist nun eine Globalverdrahtung für die Kernfläche, also alle Zellen außer den Padzellen, für die gegebene Netzliste entstanden.

4.3.2 Überprüfung des Netzzusammenhanges

Durch die zellweise Verdrahtungsreihenfolge und verwendete Methode zur Auswahl der verbundenen Pins kann nicht zu jedem Zeitpunkt garantiert werden, daß alle Netze zusammenhängend sind. Da immer der jeweils nächste Pin eines Netzes als Ziel gewählt wird, ist es für große Netze wahrscheinlich, daß sie zwischenzeitlich in zwei oder sogar noch mehr Teile zerfallen. Der in diesem Abschnitt vorgestellte Prüfalgorithmus fügt Feedthroughs in die Zellreihen ein, die ein Netz in zwei oder mehr Teile zertrennen.

Während des zellweisen Verdrahtens wurde für alle Netze bestimmt, in welchen Kanälen sie vorkommen, d.h. sich mindestens ein von diesem Netz verbundener Pin befindet. Ausgehend von diesen Informationen wird für jedes Netz geprüft, ob sich in allen Zellreihen zwischen den vom Netz durchlaufenen Kanälen jeweils ein diesem Netz zugeordneter Feedthrough befindet. Existiert eine Zellreihe ohne eine solche Feedthroughzelle, teilt diese Zellreihe das Netz in zwei unverbundene Teile. Die in der Zeile benötigte Feedthroughzelle wird so eingefügt, daß sie nach Möglichkeit zwischen den Zellreihenpositionen der Pins in den getrennten Kanälen liegt (siehe Abbildung 53). Dadurch ist gewährleistet, daß das Netz nicht unnötig verlängert wird. Ist diese Überprüfung für alle Netze erfolgt, ist die Globalverdrahtung, mit Ausnahme der Padzellen, abgeschlossen.

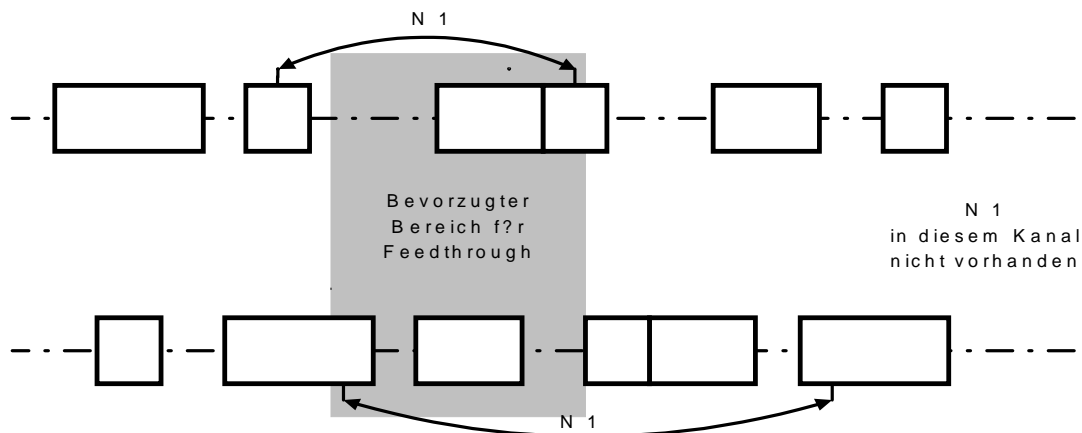


Abbildung 53: Das Einfügen eines zusätzlichen Feedthroughs garantiert den Zusammenhang des Netzes. Der Bereich, in dem es eingefügt wird, sofern dort ausreichend Platz ist, ist grau unterlegt.

4.3.3 Globalverdrahtung der Padzellen

Der letzte Teil der Globalverdrahtung ist der Anschluß der Padzellen. Soweit dies nicht schon durch die Globalverdrahtung oder bei der Überprüfung des Netzzusammenhanges geschehen ist, werden in diesem Schritt Feedthroughzellen eingefügt, die garantieren, daß die Padzellen mit ihrem jeweiligen nächsten Nachbarn desselben Netzes verbunden sind. Dabei wird ein Routinggraph analog dem für das Globalverdrahten der Zellen in der Core Area erstellt und die benötigten Feedthroughs eingefügt. In diesem Schritt wird die Position der Padzelle nicht neu evaluiert, ebensowenig die der ausgewählten Zielzelle.

Für die Beispielnetzliste T4 ist die nach der Berücksichtigung der Padzellen entstehende Globalverdrahtung in der Abbildung 54 dargestellt. Die zuvor durchgeführte Überprüfung des Netzzusammenhanges hat für die Netzliste T4 zu keinen Ergänzungen der Globalverdrahtung oder neuen Feedthroughs geführt.

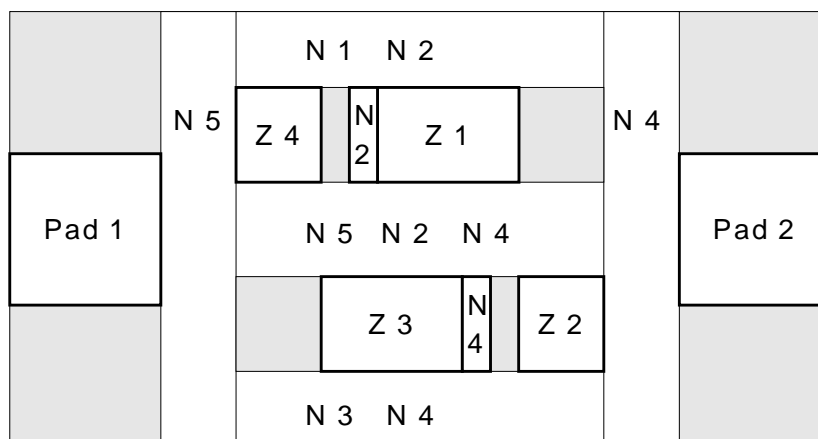


Abbildung 54: Die Globalverdrahtung für die Netzliste T4 nach der Überprüfung des Netzzusammenhanges und der Berücksichtigung der Padzellen.

Nach Abschluß der in den Abschnitten 4.3.2 und 4.3.3 dargelegten Verarbeitungsschritte ist nun eine vollständige Globalverdrahtung für die gegebene Netzliste entstanden.

4.3.4 Die Kanalverdrahtung

Das Ergebnis der Globalverdrahtung liefert die Ausgangsposition für die Detailverdrahtung, hier die Kanalverdrahtung. In dem von ESPRO verwendeten Modell ist die Kanalbreite nicht starr, sondern variabel. Es ist dem Kanalverdrahtungsalgorithmus somit möglich, weitere Tracks in einen Kanal einzufügen, falls die Kanalbreite vorab zu klein geschätzt wurde. Die zu verdrahtenden horizontalen Kanäle sind durch die Lage der Pins in den angrenzenden Zellreihen bzw. dem Padkranz definiert. Die beiden vertikalen Kanäle rechts und links der Kernfläche sind, wegen der Variabilität der Kanalbreite, erst nach dem Beenden der Kanalverdrahtung der horizontalen Kanäle vollständig definiert. Dies bestimmt die Reihenfolge, in der die Kanäle verdrahtet werden. Zunächst werden die horizontalen Kanäle nach steigender y-Koordinate bearbeitet, anschließend zunächst der linke und abschließend der rechte vertikale Kanal.

Das Kanalverdrahtungsproblem ist nicht Teil des in dieser Arbeit vorgestellten Vorgehens für Platzierung und Verdrahtung, da nach erfolgter Globalverdrahtung allein aber kein gültiges und damit vergleichbares Ergebnis vorliegt, wird hier der Greedy Kanalverdrahter ohne weitere Betrachtung übernommen.

Für die Kanalverdrahtung wird jeder Kanal, ungeachtet seiner Orientierung (horizontal oder vertikal), in einem eigenen Koordinatensystem betrachtet. Dabei wird die Koordinate entlang der Kanalausdehnung als Spalte und die Koordinate orthogonal dazu als Reihe oder Track bezeichnet. Dies ermöglicht es, das Kanalverdrahtungsproblem losgelöst von der Umgebung zu betrachten. Für die folgende Beschreibung wird angenommen, daß der betrachtete Kanal horizontal verläuft.

Die Bearbeitung jedes Kanals teilt sich in drei Schritte auf: Zunächst wird die erforderliche Kanalbreite geschätzt, dann erfolgt die Kanalverdrahtung und abschließend werden alle unbenutzten Tracks aus dem Kanal gelöscht.

Die Kanalbreitenschätzung erfolgt aufgrund der durch den Globalverdrahter erstellten Kanalauslastungen, die mit einem empirisch bestimmten Skalierungsfaktor gewichtet werden. Für die Kanalverdrahtung wird ein Greedy-Router verwendet. Dieser von Rivest und Feduccia entwickelte Algorithmus betrachtet den Kanal spaltenweise von links nach rechts und weist den Netzen, die die Spalte durchqueren, geeignete Tracks zu. Der Greedy-Router ist robust und flexibel einsetzbar, hat aber, da er viele sogenannte Doglegs einführt, den Nachteil viele Kontaktlöcher (Vias) zu verwenden. Unter einem Dogleg versteht man eine L-förmige Verbindung aus zwei orthogonalen Leitungssegmenten. Der hier implementierte Greedy Kanalverdrahter basiert auf den Beschreibungen aus [She95] und [Zob94].

In die erste Spalte treten von links die Netze herein, die den Kanal an der linken Kante betreten und werden den entsprechenden Tracks zugewiesen. Anschließend werden die folgenden Schritte für jede Spalte durchgeführt:

Zunächst wird überprüft, ob die beiden Pins an der Ober- und Unterkante des Kanals zum selben Netz gehören, ist dies der Fall, so werden sie mit einem senkrechten Leitungssegment verbunden. Gehören die Pins zu verschiedenen Netzen, so wird der Pin an der Oberkante (Unterkante) mit dem am weitesten oben (unten) liegenden von seinem Netz belegten Track verbunden. Existiert noch kein dem Netz zugewiesener Track, so wird der am weitesten oben (unten) gelegene freie Track ausgewählt, dem Netz des Pins zugewiesen und der Pin mit diesem verbunden. Ist ein Verbinden der beiden Netze von

oberem und unterem Pin mit einem von dem jeweiligen Netz schon verwendeten Track unmöglich, weil der Track des oberen Netzes unterhalb des Tracks des unteren Netzes liegt, so wird einer der Pins mit dem Track seines Netzes verbunden und für den anderen ein leerer Track verwendet. Wird ein leerer Track benötigt, es ist aber keiner mehr verfügbar, so wird an der gewünschten Stelle ein neuer Track in den Kanal eingefügt.

Belegt ein Netz mehr als einen Track, so werden die beiden Tracks mit einem vertikalen Leitungssegment verbunden und der Track freigegeben, der weiter vom nächsten anzuschließenden Pin des Netzes entfernt liegt. Ist die Verbindung eines so getrennten Netzes nicht möglich, werden Doglegs eingefügt, so daß zwei dichter beieinander liegende Tracks verwendet werden. Freie Tracks werden verwendet, um Netze dichter an den nächsten zu verbindenden Pin heranzubringen, soweit dies möglich ist.

Sind alle diese Aktionen für eine Spalte abgeschlossen, werden alle Tracks in die nächste Spalte propagiert und die Schritte wiederholt. In der letzten Spalte noch verwendete Tracks beinhalten Netze, die den Kanal an der rechten Kante verlassen. Existieren in der letzten Spalte noch Netze, die mehr als einen Track belegen, müssen diese in der benachbarten Verdrahtungsregion verbunden werden. In Abbildung 55 ist der Greedy-Routing Algorithmus dargestellt.

Greedy-Router:

- 1) Abschätzen der Kanalbreite, Kanal dieser Breite erzeugen
- 2) Links eintretende Netze in erste Spalte aufnehmen
- 3) Verbinde oben und unten eintretende Netze mit Tracks, die diesen Netzen schon zugeordnet sind. Ist das Netz neu im Kanal, belege einen freien Track. Existiert kein freier Track, verbreitere den Kanal
- 4) Vereinige Netze, die mehrere Tracks belegen, soweit möglich
- 5) Bringe Tracks von Netzen, die mehrere Tracks belegen, dichter zusammen, sofern möglich
- 6) Bringe Netze dichter an den nächsten anzuschließenden Pin (oben/unten) heran
- 7) Wenn die rechte Kante des Kanals erreicht ist, ENDE
- 8) Propagiere alle belegten Tracks in die nächste Spalte und weiter bei 3)

ENDE

Abbildung 55: Algorithmus des Kanalverdrahters: Greedy-Router

Ist die eigentliche Kanalverdrahtung abgeschlossen, erfolgt eine einfache Form der Kompaktifizierung (Compaction): Es werden alle nicht verwendeten Tracks aus dem Kanal gelöscht und dessen Breite entsprechend angepaßt.

4.3.5 Abschluß des Layouts

Nach erfolgreicher Kanalverdrahtung ist das endgültige Layout des von der Netzliste definierten physikalischen VLSI-Entwurfs entstanden. Das Ergebnis kann nun durch einen Nachoptimierer verbessert, sowie durch eine Kompaktifizierung die ungenutzten Freiräume im Layout eliminiert werden und so die benötigte Chipfläche reduziert werden.

Diese Schritte sind von dem Erzeugen eines gültigen Layouts unabhängig und nicht Teil dieser Arbeit.

Die durch den Verdrahter modifizierte Platzierung sowie das entstandene Layout werden in einer ESPRO-Platzierungs- bzw. Layoutdatei abgelegt. Abschließend werden statistische Werte zum Programmmlauf ausgegeben. Diese umfassen unter anderem die Anzahl der eingefügten Tracks bei der Kanalverdrahtung und die Anzahl verwendeter Feedthroughzellen.

Das entgültige Layout für die Netzliste T4, wie es sich durch die in den Abschnitten 4.3.4 und 4.3.5 beschriebenen Verarbeitungsschritte aus der in Abbildung 54 gegebenen Globalverdrahtung ergibt, ist in Abbildung 56 dargestellt.

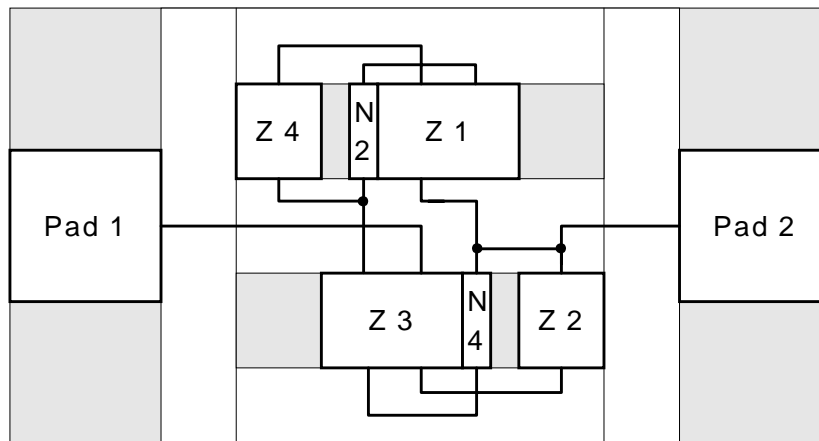


Abbildung 56: Das entgültige Layout für die Netzliste T4.

5. EVALUATIONSVERFAHREN

Dieses Kapitel beschreibt das zur Evaluation des im vorigen Kapitel beschriebenen Verfahrens angewendete Vorgehen. Das erste Teilkapitel 5.1 legt Grundlegendes zum implementierten Prototypen dar. Das Kapitel 5.2 beschreibt die verwendeten Beispielnetzlisten und die Vergleichsdaten. Den Hauptteil macht das letzte Teilkapitel 5.3 aus, es beschreibt das Vorgehen der Evaluierung, mit Schwerpunkt auf dem Prozedere zur Ermittlung der Algorithmusparameter.

5.1 Implementierung eines Prototypen: das Programm „prob“

Zur Evaluation des in Kapitel 4 dargelegten Verfahrens wurde ein Prototyp implementiert und die mit diesem Programm erstellten Ergebnisse mit von anderen ESPRO-Modulen generierten Layouts verglichen.

Das Programm prob (Placement- and Routing-Module by Ole Blaurock) wurde als Kommandozeilenapplikation für Linux entwickelt. Als Programmiersprache ist ANSI-C verwendet worden, als Compiler kam der gcc zum Einsatz. Die Benutzereingaben können über Kommandozeilenparameter interaktiv in textbasierten Abfragemasken oder über Dateien erfolgen. Bei der Gestaltung der Datenübergabe wurde besonderer Wert auf Verwendbarkeit des Programms in Shellskripten gelegt, auf eine graphische Benutzerführung wurde dagegen vollends verzichtet.

Der Prototyp ist, wie alle ESPRO-Module, beliebig mit anderen ESPRO-Modulen kombinierbar. Das Programm prob enthält zwei Module, den in Kapitel 4.2 erläuterten Platzierungsalgorithmus und den in Kapitel 4.3 dargelegten Verdrahter. Es ist möglich, diese beiden Algorithmen einzeln zu verwenden oder Platzieren und Verdrahten vollständig mit prob durchzuführen.

Ziel der Prototypenimplementierung ist es, wie in Kapitel 2.6 dargelegt, eine erste Evaluation des prinzipiellen Verfahrens vorzunehmen. Es ist daher auf eine Untersuchung zu möglichst effizient einsetzbaren Datenstrukturen verzichtet worden. Die vorwiegend verwendete Datenstruktur ist die doppelt verkettete Liste. Sie bietet den Vorteil einer dynamischen Größe, sowie einfacher Such-, Einfüge- und Löschoptionen. Der Nachteil der linear von der Anzahl der Listenelemente abhängigen Zugriffszeit macht sich nur für große Layouts wesentlich bemerkbar und wurde zu Gunsten der im Vergleich zu effizienteren Strukturen, wie AVL-Bäumen oder B-Bäumen, einfachen Handhabung und dem wesentlich geringeren Implementierungsaufwand in Kauf genommen.

Eine Beschreibung des Programmes prob und Erläuterungen zu seiner Bedienung finden sich in Anhang A. Dort sind auch alle Parameter, mit denen das Verhalten der implementierten Algorithmen beeinflusst werden können, dargelegt.

5.2 Eigenschaften der verwendeten Evaluierungsdaten

Als Umgebung für die Evaluation dient das in Kapitel 3 beschriebene offene System ESPRO, es kommen jedoch nicht alle der dort vorgestellten Module zum Einsatz.

Die mit den in dieser Arbeit entwickelten ESPRO-Modulen erstellten Ergebnisse werden mit den in [Rau00] dargelegten Werten verglichen. Hierfür werden die mit dem dkrouter verdrahteten Layouts, die jeweils aus Platzierung, die mit dem Genetischen Platzierer, dem Fuzzylogik Platzierer bzw. dem Great Deluge Platzierer erstellt wurden,

betrachtet. Zum Vergleich wird eine Auswahl der in [Rau00] verwendeten Netzlisten herangezogen. Im Detail sind dies die synthetisch generierten Netzlisten `test100`, `test200`, `test500` und `test1000` sowie die aus am Arbeitsbereich TECH des Fachbereichs Informatik der Universität Hamburg entworfenen Layouts gewonnenen Netzlisten `pk246`, `pk276` und `pk690`.

Als Netzliste mit den meisten Zellen wird zusätzlich die ebenfalls synthetisch generierte Netzliste `test3000` betrachtet, für welche mit existierenden ESPRO-Modulen jedoch nur unvollständige Layouts erzeugt werden konnten, deren statistische Werte nicht verwendet werden.

Die beiden Gruppen von Testnetzlisten, zum einen die mit dem ESPRO-Modul `rnd2chp` synthetisch generierten Netzlisten und zum anderen die importierten Netzlisten, weisen stark unterschiedliche Charakteristika auf. Die synthetisch generierten Netzlisten werden im folgenden auch als theoretische oder konstruierte Beispiele, die aus realen Entwürfen importierten Netzlisten als praktische Beispiele bezeichnet.

Die praktischen Netzlisten bestehen überwiegend aus Zweipunktnetzen (ca. 69 % aller Netze sind Zweipunktnetze [Kla94]). Die Anzahl der Padzellen ist bei allen drei praktischen Netzlisten hoch, so daß die entstehenden Layouts pad-dominiert sind. Da die Chipfläche somit nicht durch den Platzbedarf der Kernfläche, sondern die benötigte Anzahl der Padzellen bestimmt wird, läßt sich die Chipgröße nur eingeschränkt als Bewertungskriterium einsetzen. Auf diesen Aspekt wird in den folgenden Abschnitten und bei der Diskussion der Ergebnisse (vergleiche Kapitel 6) bei Bedarf näher eingegangen.

Die künstlich generierten Netzlisten weisen einen großen Anteil an Mehrpunktnetzen auf, von denen einige extrem viele Pins (von 10% bis zu 25% der vorhandenen Zellen) umfassen. Die Zweipunktnetze stellen dagegen nur einen Anteil von 7% der Netze. Diese Eigenschaft der konstruierten Netzlisten resultiert in einer, im Verhältnis zur Zellenanzahl, schwierigen Verdrahtungsaufgabe.

Die generierten Netzlisten sind, anders als die praktischen, core-dominiert. Die benötigte Chipfläche läßt sich folglich vollwertig als Bewertungskriterium für die erstellten Layouts heranziehen.

5.3 Durchgeführte Untersuchungen

Wie in Kapitel 3.3 dargelegt, erfolgt die Bewertung der Layouts in dieser Arbeit nach den gleichen Kriterien wie in [Rau00]. Dies sind die Chipgröße sowie die maximale und die durchschnittliche Netzlänge.

Die durchgeführte Evaluierung gliedert sich in zwei Phasen. Für jeden Algorithmusschritt werden zunächst an je einem konstruierten, der Netzliste `test500`, und einem praktischen Beispiel, der Netzliste `pk690`, günstige Parametereinstellungen ermittelt. In der zweiten Phase werden, wie in [Rau00], alle hier betrachteten Testschaltungen (vergleiche Kapitel 5.2) mit diesen Einstellungen bearbeitet.

Die beiden in `prob` implementierten Module werden sowohl einzeln als auch in ihrem Zusammenwirken untersucht. Der Vergleich der in `prob` implementierten Module mit den in [Rau00] verwendeten Algorithmen erfolgt mit den im ersten Evaluierungsschritt ermittelten Parametrisierungen für das jeweils betrachtete Modul.

Der Rest dieses Kapitels widmet sich der Beschreibung der Evaluierungsmethode. Die drei folgenden Abschnitte legen die Abläufe jeweils für den Verdrahter (Abschnitt 5.3.1), den Plazierer (Abschnitt 5.3.2) und die Kombination aus Plazierer und Verdrahter von prob (Abschnitt 5.3.3) dar.

Die ermittelten Parametrisierungen sowie die Ergebnisse mit existierenden ESPRO-Modulen werden in Kapitel 6 diskutiert. Dort werden auch Details zu Auswahlentscheidungen für bzw. gegen mögliche Parametrisierungen behandelt.

5.3.1 Untersuchung des Verdrahters

Zur Bestimmung der Parameter für den Verdrahter im Alleinbetrieb werden die vom Genetischen Plazierer erzeugten Plazierungen für die Netzlisten `test500` und `pk690` verwendet.

Die das Verhalten des Verdrahters bestimmenden Parameter lassen sich in die folgenden zwei Gruppen teilen: Die erste Gruppe enthält die Parameter zur Pfadbewertung und bestimmt damit die eigentliche Globalverdrahtung. Die zweite Parametergruppe enthält die Parameter zur Veränderung der Plazierung.

Die Suche nach geeigneten Parametern für den Verdrahter erfolgt entsprechend in zwei Schritten, je einer für jede der beiden Parametergruppen und wird in den beiden folgenden Abschnitten erläutert.

In Abschnitt 5.3.1.1 wird zunächst die Pfadbewertung betrachtet. Eine Veränderung der Plazierung durch den Verdrahter ist dabei zunächst nicht zugelassen. Der Abschnitt 5.3.1.2 beschreibt die Evaluierung der Parametergruppe für die Plazierungsveränderung. In Abschnitt 5.3.1.3 wird das Vorgehen für den Vergleich mit dem `dkrouter` beschrieben. Die gewonnenen Parametrisierungen, sowie die daraus ableitbaren Charakteristika des Verdrahtungsalgorithmus, sind in Abschnitt 6.1.1 diskutiert.

5.3.1.1 Bestimmung der Pfadbewertungsparameter

Zunächst werden bei unveränderbarer Plazierung die Bewertungskriterien für die Pfadauswahl untersucht. Die Pfadbeurteilung geschieht nach den in Gleichung 3 und Gleichung 4 dargelegten Bewertungsfunktionen, deren Parametrisierung zu bestimmen ist. Wie aus Gleichung 3 und Gleichung 4 ersichtlich ist, sind sechs Parameter wählbar. Um die aufwendige Bearbeitung eines sechsdimensionalen Suchraumes zu vermeiden, wird die Parametrisierung in zwei Schritten sequentiell bestimmt.

Zunächst wird ein dreidimensionaler Suchraum betrachtet. Er wird durch Variation der Parameter zur Gewichtung der mit der Kanalauslastung gewichteten Leitungslänge, der maximalen Kanalauslastung und des Abstandes der verwendeten Feedthroughs zum Netzmittelpunkt aufgespannt. Die Tabelle B- 1 gibt die sechs Parameter namentlich wieder und zeigt die Variationen der Parameterwerte für den ersten Suchabschnitt.

Aus diesem dreidimensionalen Suchraum werden die Unterräume mit gleichem Wert für die Gewichtung der Lage des Pfades relativ zum Netzmittelpunkt einzeln betrachtet. Aus jedem dieser Unterräume wird die jeweils beste Parametrisierung bezüglich maximaler Netzlänge bzw. Chipgröße (oder durchschnittlicher Netzlänge, vergleiche Abschnitt 6.1.1.3) ausgewählt.

Für die so getroffene Auswahl von 16 Parametrisierungen wird in einem zweiten Schritt der Parameter zur Gewichtung der Netzlänge variiert, die hierfür verwendeten Parametrisierungen sind in Tabelle B- 2 wiedergegeben.

Das Variieren der verbleibenden zwei Parameter ist nicht erforderlich, da sich ihr Einfluß auf die Pfadbewertung bereits durch die betrachteten Parametrisierungen ändert.

Die als günstig ermittelten Parametereinstellungen für die Pfadbewertung sind in Abschnitt 6.1.1.1 und Abschnitt 6.1.1.3 diskutiert. Sie bilden die Grundlage der weiteren Parametrisierung des Verdrahtungsalgorithmus.

5.3.1.2 Bestimmung der Parameter zur Platzierungsmodifikation

Wie in 4.3.1 dargelegt, ist es dem Verdrahtungsalgorithmus möglich, Änderungen an der Platzierung vorzunehmen. Diese Modifikationen beinhalten das Bewegen noch vollständig unverdrahteter Zellen zur Verbesserung der Platzierung sowie das Verschieben von Zellen und Feedthroughs, um Raum für neue mögliche Feedthroughpositionen zu schaffen.

Die Suche nach einer günstigen Parametrisierung dieser Algorithmeigenschaften erfolgt in einem Schritt unter der Betrachtung eines dreidimensionalen Suchraumes.

Es wird der Suchradius für Alternativpositionen von noch unverdrahteten Zellen und der Suchradius für betrachtete Feedthroughpositionen variiert, sowie die Zellverschiebung für das Schaffen neuer möglicher Feedthroughpositionen zugelassen bzw. unterlassen. Die Namen und untersuchten Werte der hierfür variierten Algorithmusparameter sind in Tabelle B- 3 aufgeführt. Für jede der, wie in Abschnitt 5.3.1.1 beschrieben ermittelten, Parametrisierungen der Pfadbewertung, wird ein solcher Suchraum aufgespannt und evaluiert. Jeder aufgespannte Suchraum wird dabei bezüglich desselben Kriteriums bewertet, welches bereits für die Auswahl der zugrundeliegenden Pfadbewertungsparameter verwendet wurde.

Als Ergebnis der Untersuchung der Parameter zur Platzierungsmodifikation ergeben sich Konfigurationen für den Verdrahter, die, jeweils abgestimmt auf die Charakteristika der konstruierten bzw. der praktischen Netzlisten, eine bestimmte Layouteigenschaft unterstützen, z.B. die des möglichst geringen Flächenbedarfes.

Die Ergebnisse dieses Evaluierungsschrittes sind in Abschnitt 6.1.1.2 und Abschnitt 6.1.1.4 dargelegt.

5.3.1.3 Vergleich mit dem dkrouter

Auf Basis der in [Rau00] ermittelten Werte wird das in dieser Arbeit erstellte Verdrahtermodul mit dem `dkrouter` verglichen. Für diesen Vergleich werden nicht alle ermittelten Parametrisierungen verwendet. Da in [Rau00] keine unterschiedlichen Parametrisierungen für verschiedene Bewertungskriterien der in `dkrouter` implementierten Heuristik betrachtet werden, soll hier ebenfalls nur eine Gruppe von Parametrisierungen untersucht werden. Für die praktischen Netzlisten entfällt das Kriterium der Chipgröße, somit bleibt die Betrachtung der auf kurze maximale Netzlängen ausgelegten Parametrisierungen.

Die mit drei verschiedenen Platzierungsmodulen erzeugten Platzierungen der Beispielnetzlisten werden, mit den unterschiedlichen Parametrisierungen, die unter

Betrachtung der maximalen Netzlänge ermittelt wurden, mit dem Verdrahter von `prob` verdrahtet.

Die Qualität der entstehenden Layouts wird bezüglich der benötigten Chipfläche, der durchschnittlichen und der maximalen Netzlänge mit den durch den `dkrouter` erzeugten Layouts verglichen. Die Ergebnisse dieses Vergleiches sind in Abschnitt 6.1.2 präsentiert.

5.3.2 Untersuchung des Plazierers

Die Untersuchung des Plazierers teilt sich, genau wie die Evaluation des Verdrahters, in zwei Teile. Zunächst wird untersucht, welche Parametrisierungen geeignet sind, um Layouts mit den gewünschten Eigenschaften zu erhalten. Abschließend wird der geeignet parametrisierte Algorithmus mit existierenden ESPRO-Plazierern verglichen. Um bewertbare Layouts zu erhalten, werden die Plazierungen mit dem `dkrouter` verdrahtet.

Die Parameter des Plazierungsalgorithmus lassen sich in zwei Gruppen einteilen. Die erste Gruppe bestimmt die Eigenschaften der entwurfstilunabhängigen Vorplatzierung, die zweite die Charakteristika der Überführung in eine Standardzellplatzierung. Die Gruppe der die Vorplatzierung bestimmenden Parameter zerfällt wiederum in zwei Teile: Eine Teilgruppe bestimmt die Eigenschaften der hierarchischen Partitionierung, die andere Teilgruppe definiert das Verhalten des Algorithmus der kräftegesteuerten Platzierung.

Der Rest dieses Kapitels gliedert sich entsprechend in vier Abschnitte. In Abschnitt 5.3.2.1 wird zunächst die Konvertierung der Platzierung in eine Standardzellplatzierung betrachtet. Der Abschnitt 5.3.2.2 legt das Vorgehen zur Parameterbestimmung für die hierarchische Partitionierung dar und in Abschnitt 5.3.2.3 wird die Evaluierung einer geeigneten Parametrisierung für die kräftegesteuerte Platzierung behandelt. Der Abschnitt 5.3.2.4 legt das Vorgehen zum Vergleich mit drei existierenden ESPRO-Plazierern dar.

5.3.2.1 Parametrisierung der Standardzellplatzierung

Die Parametrisierung der Algorithmusschritte zum Erstellen der Vorplatzierung wird zunächst auf geeignet erscheinende Art vorgenommen und bleibt für die Evaluation der Erzeugung einer Standardzellplatzierung unverändert.

Die Parameter für die hierarchische Partitionierung werden dabei so gewählt, daß die Gewichtung der internen geschnittenen Kanten und der externen geschnittenen Kanten im Verhältnis 2:1 erfolgt. Die Kardinalitätsdifferenz erhält als Gewicht die Summe der beiden Gewichte der geschnittenen Kanten. Die genaue Parametrisierung für Gleichung 1 ist im Begleittext zu Tabelle B- 4 aus Anhang B gegeben. Die kräftegesteuerte Platzierung wird in diesem Evaluierungsschritt zunächst ohne Berücksichtigung weiterer möglicher Abbruchbedingungen über zehn Iterationen ausgeführt.

Für den Algorithmus zur Überführung der Vorplatzierung in eine Standardzellplatzierung teilen sich die Parameter wiederum in zwei Gruppen. Die erste Gruppe bestimmt das Gesamterscheinungsbild des Chips, also die Breite der Kanäle, die Füllungsdichte der Zellreihen und das Seitenlängenverhältnis etc. Die zweite Gruppe bestimmt das Algorithmusverhalten bei der Betrachtung einer Zelle. So wird in dieser Gruppe die maximale Entfernung gewählt, um die sich die Einfügeposition einer Zelle oder eines

Feedthroughs von der berechneten Sollposition innerhalb einer Zellreihe unterscheiden darf.

Als Gesamterscheinungsbild des fertigen Chiplayouts ist ein Quadrat möglichst kleiner Fläche und mit hoher Packungsdichte der aktiven Zellen erwünscht. Da die Parameter der ersten genannten Gruppe hauptsächlich die Vorgaben an das Erscheinungsbild definieren, sind hier kaum Evaluierungen der Parameter nötig.

Die nicht evaluierten Parameter und ihr Einfluß auf das Algorithmusverhalten bzw. Erscheinungsbild des Layouts werden hier kurz dargelegt.

Die Güte der Schätzfunktion zur Ermittlung der initial verwendeten Chipgröße bestimmt lediglich die Anzahl der nötigen Iterationen, jedoch nicht die Eigenschaften der Standardzellplatzierung. Da die Kanalbreiten variabel sind, können ungenaue Kanalbreitenschätzungen des Plazierers in späteren Layoutschritten, z.B. der Globalverdrahtung, verbessert werden. Aus diesen Gründen wird auf eine ausführliche Evaluierung der Parameter für die Kanalbreitenschätzer verzichtet und die Werte, wie sie in Anhang B gegeben sind, verwendet.

Es ist eine möglichst hohe Zellreihenauslastung anzustreben, da diese maßgeblich den Flächenbedarf des Layouts bestimmt. Es werden Auslastungen von 95% und 90% untersucht. Sollte dieses zu einzelnen unverdrahtbaren Platzierungen führen, kann dieser Wert herabgesetzt werden, ohne die Charakteristik der erstellten Platzierung wesentlich zu verändern. Da eine quadratische Chipfläche erstrebenswert ist (vergleiche Abschnitt 4.2.4.1), wird der Parameter für das Seitenlängenverhältnis ebenfalls nicht betrachtet.

Somit bleiben die Parameter, die für die Betrachtung einer einzelnen Zelle das Algorithmusverhalten bestimmen. Es ergibt sich ein dreidimensionaler Suchraum für die Parametrisierung der Erzeugung einer Standardplatzierung.

Der Suchraum wird aufgespannt durch die Parameter für den Suchradius für Zellpositionen, den Suchradius für Feedthroughpositionen und die geforderte bzw. nicht geforderte äquidistante Anordnung der Zellen innerhalb der Zellreihe. Die evaluierten Parameterwerte sind in Tabelle B- 4 angegeben.

Die Möglichkeit, eine äquidistante Zellteilung innerhalb der Zellreihe zu fordern, ist implementiert worden, um die für andere ESPRO-Plazierer typische gleichmäßige Verteilung der Feedthroughs erzwingen zu können. Es werden jeweils beide Varianten der gewählten Parametrisierungen, sowohl mit, als auch ohne äquidistante Zellverteilung zur weiteren Verwendung ausgewählt.

Die Auswahl der für die weitere Arbeit mit `prob` zu verwendenden Parameter erfolgt wiederum anhand der Bewertung des mit dem `dkrouter` erstellten Layouts. Es werden die Parametrisierungen verwendet, die sowohl bezüglich der maximalen Netzlänge als auch bezüglich der Chipgröße gute Ergebnisse liefern. Die Qualität der Layouts mit 95% Zeilenauslastung wird dabei stärker berücksichtigt, als die Qualität der Layouts mit 90%iger Reihenauslastung.

Es werden drei Parametrisierungen ausgewählt, zunächst die Parametrisierung, die bezüglich aller betrachteten Kriterien die günstigsten Resultate hervorbringt. Diese Parametrisierung wird für die weitere Evaluierung des Plazierers verwendet.

Zusätzlich wird je eine Parametrisierung ausgewählt, die entweder sehr kleine Suchradien verwendet oder sehr große Suchradien zuläßt.

5.3.2.2 Parametrisierung der hierarchischen Partitionierung

Ausgehend von der gewählten Parametrisierung zur Überführung der Vorplazierung in eine Standardzellplazierung (vergleiche Abschnitt 5.3.2.1), folgt die Evaluierung der Parametrisierung der Vorplazierung. In diesem Abschnitt wird zunächst die Untersuchung der hierarchischen Partitionierung beschrieben.

Der zweite Schritt der Vorplazierung, die kräftegesteuerte Plazierung, nimmt wesentliche Änderungen an der Zell- und Netzanordnung vor. Daher wird zur besseren Bewertbarkeit der hierarchischen Partitionierung die kräftegesteuerte Plazierung vorerst komplett deaktiviert.

Die Untersuchung der Parametrisierung erfolgt in einem Schritt. Die Parameter für die Gewichtung von internen und externen geschnittenen Verbindungen sowie dem Kardinalitätsunterschied der entstehenden Partitionen (vergleiche Gleichung 1 und Abschnitt 4.2.1.2) spannen einen dreidimensionalen Suchraum auf. Die verwendeten Parameterbelegungen sind unter Tabelle B- 5 wiedergegeben.

Für die folgende Evaluierung der Parametrisierung der kräftegesteuerten Plazierung wird eine Stichprobe aus den jeweils besten sieben Parametrisierungen jedes Bewertungskriteriums ausgewählt.

5.3.2.3 Parametrisierung der kräftegesteuerten Plazierung

Die Parametrisierungsmöglichkeit der kräftegesteuerten Plazierung beschränkt sich auf die Vorgabe verschiedener Abbruchbedingungen. Die innerhalb einer Iteration ablaufenden Algorithmusschritte sind nicht parametrisierbar.

Für die Beurteilung der durch eine Iteration hervorgerufenen Änderungen an der Vorplazierung wird die Entfernung herangezogen, um welche die Zellen bewegt werden. Es lassen sich Abbruchbedingungen bezüglich der mittleren und der maximalen auftretenden Verschiebungsdistanz definieren.

Die Abbildung 57 zeigt den ungefähren Verlauf der Verschiebungsdistanz über die Iterationen, sowie, als gestrichelte Linien, die parametrisierbaren Abbruchbedingungen.

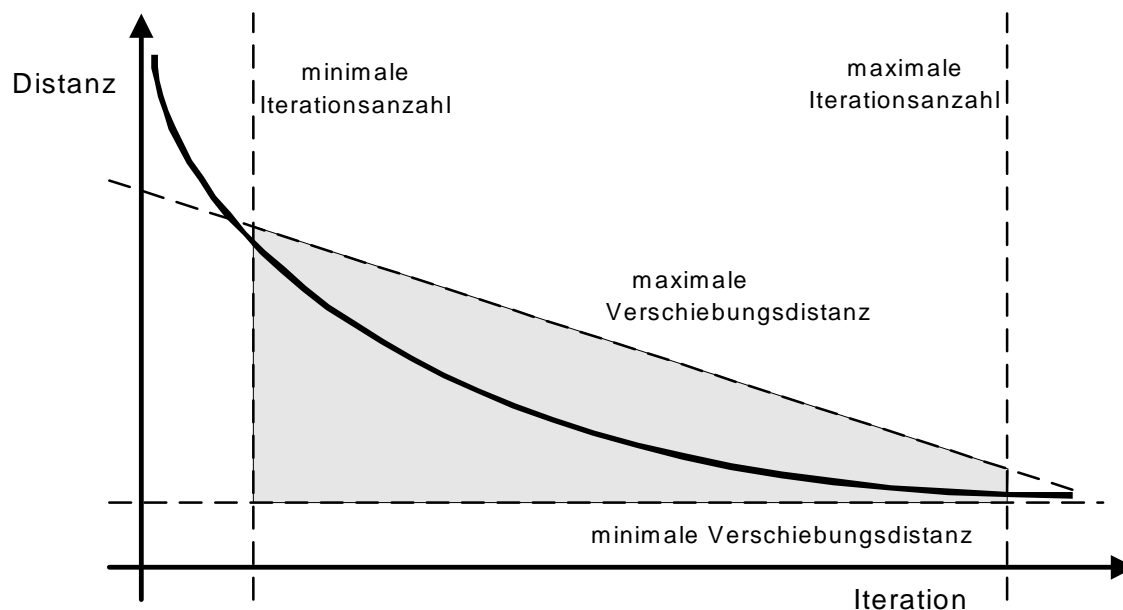


Abbildung 57: Verlauf der Zellverschiebungsdistanz während der kräftegesteuerten Platzierung. Die gestrichelt dargestellten Geraden zeigen die Abbruchbedingungen an.

Wie der Abbildung 57 leicht zu entnehmen ist, sind bei dem sich zeigenden Kurvenverlauf für die Verschiebungsdistanzen, die durch Geraden gegebenen Abbruchbedingungen, nicht sehr geeignet. Liegt die untere Schranke der mittleren Verschiebungsdistanz in derselben Größenordnung wie die Ausdehnung der verwendeten Basiselemente, so bewirken die Zellverschiebungen keine merklichen Platzierungsverbesserungen. Die Anpassung der Vorplatzierung an den Standardzellentwurf führt zu Zellbewegungen, die ebenfalls in der Größenordnung der Ausdehnung der Basiselemente liegen und diese sogar wesentlich überschreiten können. Analoges gilt für die maximale auftretende Verschiebungsdistanz, für die ein ähnlicher Wert als Abbruchkriterium gewählt wird.

Die hier durchgeführte Evaluierung setzt daher die Anzahl der durchzuführenden Iterationen fest und deaktiviert alle übrigen Abbruchkriterien. Unter Verwendung der wie im vorigen Abschnitt ermittelten Zellanordnungen der hierarchischen Partitionierung, wird eine geeignete Anzahl von Iterationen für die kräftegesteuerte Platzierung ermittelt. Die verwendeten Werte sind in Tabelle B- 6 angegeben.

Zunächst wird anhand der exemplarischen Betrachtung einer Parametrisierung der hierarchischen Partitionierung aus der gewählten Stichprobe mit verschiedenen Iterationsanzahlen der Einfluß der kräftegesteuerten Platzierung auf die Eigenschaften der Vorplatzierung bewertet. Anschließend wird ein Vergleich aller im vorigen Abschnitt ausgewählten Parametrisierungen mit den in Tabelle B- 6 gegebenen Iterationsanzahlen durchgeführt und anhand der entstehenden Layouts die endgültig verwendeten Parametrisierungen bestimmt.

5.3.2.4 Vergleich mit existierenden Platzierern

Die beiden möglichen Platzierungsmethoden des prob-Plazierers werden mit drei der in [Rau00] präsentierten Platzierungsalgorithmen verglichen. Als Referenz werden der Great Deluge, der Fuzzylogik- und der genetische Platzierungsalgorithmus herangezogen. Es wird mit jeder der in Tabelle C- 32 bzw. Tabelle C- 33 angegebenen Parametrisierung eine prob-Platzierung zu jeder betrachteten Netzliste erstellt. Die Verdrahtung dieser

Plazierungen erfolgt mit dem `dkrouter`. Die entstehenden Layouts werden in Kapitel 6.2.4 mit den Ergebnissen aus [Rau00] verglichen.

5.3.3 Untersuchung beider Komponenten zusammen

Für die Beurteilung des gesamten in dieser Arbeit entwickelten und in `prob` implementierten Verfahrens zum Plazieren und Verdrahten, werden beide in `prob` enthaltenen Module zur Layouterzeugung verwendet. Dabei wird in einem Programmstart sowohl das Plazieren als auch das Verdrahten einer gegebenen Netzliste durchgeführt. Die entstehenden Plazierungsdateien werden nicht benötigt.

Im Gegensatz zur Evaluierung der beiden Module wird nun nicht erneut die gesamte Parametrisierung untersucht. Ausgehend von den zuvor gewonnenen einzelnen Wertebelegungen der Parametergruppen für Teilaufgaben des Plazierens und Verdrahtens wird nur ein Vergleich für verschiedene Belegungen dieser Parametergruppen durchgeführt.

Es werden für drei Parametergruppen verschiedene Wertebelegungen betrachtet. Durch Bildung aller möglichen Kombinationen der Parametrisierungen der Gruppen wird ein dreidimensionaler Suchraum mit 36 Parametrisierungen aufgespannt. Die Parametergruppen sind wie folgt variiert worden:

- Die Vorplatzierung wird entweder nur durch die hierarchische Partitionierung oder durch zusätzliche Anwendung der kräftegesteuerten Methode erzeugt (vergleiche Tabelle C- 32 und Tabelle C- 33).
- Die Überführung der Vorplatzierung wird mit jeder der sechs Parametrisierungen aus Tabelle C- 17 bzw. Tabelle C- 18 durchgeführt.
- Die Verdrahtung erfolgt mit Bevorzugung des Kriteriums der maximalen Netzlänge. Dabei werden drei verschiedene Grade der Platzierungsmodifikation durch den Verdrahter betrachtet: Zum ersten die Verwendung der komplett unmodifizierten Platzierung, zum zweiten das Zulassen von Zellverschiebungen zur Schaffung weiterer möglicher Feedthroughpositionen und zum dritten das zusätzliche Verschieben von Standardzellen durch den Verdrahter (vergleiche Tabelle C- 11 und Tabelle C- 12).

Die Tabelle 1 zeigt die Parametergruppen mit ihren möglichen Wertebelegungen.

Kapitel 5 Evaluationsverfahren

Tabelle 1: Zusammenfassung der untersuchten Kombinationen von Parameterwerten zu den Teilschritten bei der Evaluation des Gesamtkonzeptes.

Plazierer					Verdrahter	
Art der Erstellung der Vorplatzierung		Überführung in eine Standardzellplatzierung			Art der Platzierungsmodifikation durch den Verdrahter	
	Bezeichnung	Suchradius beim Einfügen	Zellverteilung in der Reihe	Bezeichnung		Bezeichnung
nur hierarchische Partitionierung	part	bester Suchradius	äquidistant	row1	keine	no
			variabel	row2		
		kleiner Suchradius	äquidistant	row3	Zellverschiebung für neue Feedthrough-Positionen	yes
			variabel	row4		
hierarchische Partitionierung und kräftegesteuerte Methode	force	großer Suchradius	äquidistant	row5	Zellbewegungen zur Netzverkürzung	move
			variabel	row6		

Die Beurteilung der für das Gesamtverfahren günstigen Parametrisierungen erfolgt zunächst wiederum anhand der beiden Netzlisten test500 und pk690. Für ausgewählte Parametrisierungen erfolgt ein Vergleich für alle Netzlisten. Die Ergebnisse der Evaluierung des Gesamtkonzeptes sind in Kapitel 6.3 dargelegt.

6. ERGEBNISSE

In diesem Kapitel werden die Ergebnisse der Evaluierung des in dieser Arbeit entwickelten Verfahrens dargelegt. Der Aufbau dieses Kapitels ergibt sich aus dem in Kapitel 5 beschriebenen Vorgehen.

Das Kapitel 6.1 präsentiert die Evaluierungsergebnisse für den Verdrahtungsalgorithmus, in Kapitel 6.2 wird die Plaziererevaluierung dargelegt und Kapitel 6.3 legt die Resultate der Kombination der beiden neu implementierten ESPRO-Module dar. Den Abschluß bildet das Kapitel 6.4, in dem die Ergebnisse gemeinsam diskutiert und das in dieser Arbeit entwickelte Verfahren beurteilt werden.

6.1 Untersuchungsergebnisse für den Verdrahter

Dieses Teilkapitel legt die Ergebnisse der Verdrahteruntersuchung dar. Zunächst werden in Abschnitt 6.1.1 die Resultate der Parameterevaluierung präsentiert, anschließend wird in Abschnitt 6.1.2 der in dieser Arbeit entwickelte Verdrahtungsalgorithmus mit dem `dkrouter` verglichen, wobei Plazierungen von verschiedenen ESPRO-Modulen verwendet werden.

6.1.1 Ergebnisse der Verdrahterparametrisierung

Dieser Abschnitt zeigt die Ergebnisse der Suche nach günstigen Parametrisierungen für den in `prob` implementierten Verdrahter anhand der Netzlisten `test500` und `pk690`. Jeder der vier folgenden Unterabschnitte legt die Ergebnisse der in Abschnitt 5.3.1.1 bzw. Abschnitt 5.3.1.2 beschriebenen Evaluierungsschritte für jeweils eine der beiden Netzlisten dar.

6.1.1.1 Ermittlung der Pfadbewertungsparameter anhand von `test500`

Wie in Abschnitt 5.3.1.1 beschrieben, wurde zunächst in einem dreidimensionalen Suchraum eine günstige Pfadbewertung ermittelt.

Der dreidimensionale Suchraum wurde in acht zweidimensionale Suchräume zerlegt, je ein Suchraum für eines der in Tabelle B- 1 gegebenen Knotengewichte zur Bewertung eines Pfades im Routinggraphen. In jedem Suchraum wurde die jeweils beste Parametrisierung für die Verdrahtung bezüglich der maximalen Netzliste bzw. der kleinsten Chipfläche ermittelt. Die so ausgewählten zwei Parametrisierungen zu jedem der acht Suchräume sind in Tabelle C- 1 wiedergegeben. Diese 16 ausgewählten Parametrisierungen der insgesamt 810 betrachteten Konfigurationen bilden die Grundlage der weiteren Untersuchung zur Parametrisierung der Pfadbewertung.

Die durchschnittliche Netzlänge zeigt keine signifikante Abhängigkeit von der Parametrisierung und wird als Auswahlkriterium zunächst nicht weiter beachtet.

Es zeigt sich, daß mit steigendem Knotengewicht auch eine Tendenz zu steigenden Gewichten in der Kantenbewertung zu verzeichnen ist. Ein zu starkes Ungleichgewicht in der Gewichtung der durch das Knotengewicht berücksichtigten Lage des gewählten Pfades und dessen positionsunabhängigen Eigenschaften wie Länge und Kanalauslastung längs des Weges, führt offenbar zu ungünstigeren Verdrahtungen (vergleiche Tabelle C- 1).

Für jede der 16 Parametrisierungen aus Tabelle C- 1 wurde anschließend die Gewichtung der Netzlänge gemäß Tabelle B- 2 variiert. Die Tabelle C- 2 zeigt eine Auswahl der insgesamt 160 Parametrisierungen, sortiert nach der Qualität des resultierenden Layouts bezüglich der maximalen Netzlänge. Der Tabelle C- 3 liegen dieselben Daten zugrunde, sie sind hier jedoch nach der Güte des Layouts bezüglich der benötigten Chipfläche sortiert.

Die zur Reduktion des Flächenbedarfes günstigen Parametereinstellungen weisen fast alle einen großen Gewichtungsfaktor für die Netzlänge auf. Die im Kanal entlang eines Pfades auftretende maximale Auslastung dagegen hat bei den bezüglich der Chipgröße günstigsten Parametrisierungen aus Tabelle C- 3 das Gewicht Null. Die mit der Berücksichtigung der maximalen Kanalauslastung ausgedrückte Benachteiligung des aktuell betrachteten Zweipunktnetzes zu Gunsten der später betrachteten Netze, führt offenbar zu schlechteren Wegewahlen durch den Globalverdrahtungsalgorithmus. Dies wirkt sich auf die Chipgröße stärker aus, als auf die maximale Netzlänge, wie ein Vergleich mit den Parametrisierungen aus Tabelle C- 2 zeigt.

Die Gewichtung der Lage eines Pfades relativ zum Mittelpunkt des betrachteten Netzes, ausgedrückt durch den Wert des Parameters für das Knotengewicht, liegt bei der Mehrheit der günstigen Verdrahtungsparametrisierungen in der gleichen Größenordnung wie die Gewichtung der lageunabhängigen Kriterien für die Pfadbewertung, wie Länge, Kanalauslastung etc.

Die jeweils besten Parametrisierungen für die Pfadbewertung aus Tabelle C- 2 bzw. Tabelle C- 3 bilden die Grundlage für die im nächsten Abschnitt dargelegte Untersuchung der Parametrisierung der Platzierungsmodifikation durch den Verdrahter.

6.1.1.2 Parametrisierung der Platzierungsmodifikation anhand von test500

Für die beiden, wie oben beschrieben gefundenen Parametrisierungen, wurde die in Abschnitt 5.3.1.2 beschriebene Suche nach Parameterkombinationen für die vom Verdrahter ausführbaren Platzierungsmodifikationen durchgeführt. Für jeden der beiden durch die Parametrisierung der Pfadbewertung aufgespannten Suchräume wurde das gleiche Bewertungskriterium herangezogen, wie bereits für die Parametrisierung zur Pfadbewertung, auf welcher er basiert. So wird zu der Pfadbewertung, die kurze maximale Netzlängen hervorbringt, die Platzierungsmodifikation so parametrisiert, daß sie diese Eigenschaft verstärkt. Analoges gilt für die Parametrisierungen zur Reduktion des Flächenbedarfes. Da diese Suchräume jeweils 192 Parametrisierungen umfassen, werden die Ergebnisse hier wiederum nur auszugsweise wiedergegeben.

Tabelle C- 4 und Tabelle C- 5 zeigen jeweils die bezüglich Chipgröße bzw. maximaler Netzlänge besten Parametereinstellungen für die Platzierungsmodifikationen durch den Verdrahter. Dabei basieren die in Tabelle C- 4 gezeigten Ergebnisse auf der in Tabelle C- 2 als beste ausgewiesene Parametrisierung und die in Tabelle C- 5 entsprechend auf der besten Parametrisierung aus Tabelle C- 3.

Trotz der eindeutigen Betonung der Parametrisierung auf die Verbesserung eines der beiden Kriterien maximale Netzlänge oder Flächenbedarf, zeigen sich Korrelationen. So ist eine Tendenz zu steigendem Flächenbedarf für die Parametrisierungen aus Tabelle C- 4, die lange Netzlängen hervorbringen, festzustellen. Umgekehrt steigen die maximalen Netzlängen der Parametrisierungen aus Tabelle C- 5, welche flächenintensivere Layouts

hervorbringen. Dies ist kaum verwunderlich, da bei steigender Chipfläche die zurückzulegenden Entfernungen und damit die Netzlängen zwangsläufig steigen.

Tabelle C- 4 und Tabelle C- 5 zeigen, daß die Layoutqualität durch die Veränderung der Platzierung verschlechtert werden. Dies ist auf die Tatsache zurückzuführen, daß die Umpositionierung einer Zelle vom Verdrahtungsalgorithmus nur dann erwogen wird, sofern noch keiner ihrer Pins verdrahtet ist. Es wird vom Verdrahter folglich die für den ersten bearbeiteten Pin einer Zelle günstiger erscheinende Position gewählt. Die resultierende Beeinträchtigung der erzielbaren Verdrahtungsqualität für weitere Pins der Zelle bleibt unberücksichtigt. Die verwendete Schätzfunktion zur Berücksichtigung des Einflusses einer Zellbewegung auf die anderen an die Zelle angeschlossenen Netze erreicht die angestrebte Wirkung, eben diese Beeinflussung anderer Netze zu bewerten, offenbar nicht. Die vom Platzierer ermittelte Zellposition ergibt sich dagegen aus dem Einfluß aller angeschlossenen Netze und ist daher der hier vom Verdrahter gewählten Alternative überlegen.

Es ist zu erkennen, daß das Zulassen bzw. Unterbinden des Verschiebens von Zellen zur Erzeugung weiter möglicher Feedthroughpositionen keinen signifikanten Einfluß auf die Verdrahtungsqualität hat. Dies gilt zumindest für Platzierungen, die mit bisherigen ESPRO-Plazierern erstellt wurden.

Obwohl das Zulassen von Zellbewegungen zur Erzeugung weiterer möglicher Feedthroughpositionen keine Verbesserung der Verdrahtung durch Zellverschiebungen erkennen läßt, wird eine Parametrisierung gewählt, die diese Modifikation der Platzierung zuläßt. Dies stellt einen wesentlichen Unterschied zu bisherigen ESPRO-Verdrahtern dar und soll daher in weiteren Betrachtungen zur Algorithmusparametrisierung aktiviert bleiben.

Für den in Abschnitt 6.1.2 beschriebenen Vergleich des in dieser Arbeit entwickelten Verdrahters mit dem `dkrouter`, werden beide Möglichkeiten berücksichtigt.

Für das Zulassen von Platzierungsmodifikationen, die über das Verschieben von Zellen für weitere Feedthroughpositionen hinausgehen, gilt ähnliches. Es ist in keinem anderen ESPRO-Modul vorgesehen und soll deshalb nicht völlig unberücksichtigt bleiben, auch wenn die aktuelle Umsetzung dieser Idee noch zu einer Ergebnisverschlechterung führt. Daher wird die beste Parametrisierung, welche die Modifikation der Platzierung durch den Verdrahter zuläßt und Zellverschiebungen zugunsten neuer Feedthroughpositionen vornimmt, ebenfalls für einen Vergleich mit dem `dkrouter` verwendet.

Es ergeben sich insgesamt sechs Parameterkonstellationen für den Verdrahter, welche in der Tabelle C- 11 aufgeführt sind.

6.1.1.3 Ermittlung der Pfadbewertungsparameter anhand von pk690

Die Wegewahlparameter für die der Praxis entnommenen Netzlisten werden analog dem Vorgehen für die künstlich generierten Netzlisten, wie in Abschnitt 5.3.1.1 beschrieben, ermittelt.

Der Unterschied besteht in dem Wert des Parameters für die Gewichtung des während der Globalverdrahtung ermittelten Bedarfs an Tracks in den Kanälen. Da in praktischen

Netzlisten der Anteil an Zweipunktnetzen höher als in den künstlich generierten ist, entstehen wesentlich weniger Gebiete, in denen der Globalverdrahtungsalgorithmus für ein Netz mehrere Tracks innerhalb eines Kanals reserviert. Diese Situation spiegelt sich für die praktischen Netzlisten in einem geringeren Gewicht des bei der Globalverdrahtung ermittelten Trackbedarfes wider.

Wie bereits in Abschnitt 6.1.1.1 geschehen, wird auch hier der dreidimensionale Suchraum (vergleiche Abschnitt 5.3.1.1) in acht Unterräume aufgeteilt betrachtet.

Da die betrachtete Platzierung der Netzliste `pk690`, ebenso wie die anderen der Praxis entnommenen betrachteten Netzlisten, `pad`-dominiert ist, läßt sich über den Einfluß der Parametrisierung auf den Flächenbedarf keine Aussage treffen. Das Kriterium der Chipfläche wird daher hier für diese Netzlisten nicht weiter betrachtet. Als zweites Kriterium zur Bewertung der Verdrahtungsqualität bleibt die durchschnittliche Netzlänge. Um, analog der Auswahl für Tabelle C- 1, insgesamt 16 Konfigurationen als Ausgangspunkt für die weitere Parameteruntersuchung zur Verfügung zu haben wird für Tabelle C- 6 daher die durchschnittliche Netzlänge zur Bewertung herangezogen,. Sollten zwei der aufgrund der maximalen bzw. der durchschnittlichen Netzlänge ausgewählten Parametrisierungen identisch sein, so wird zusätzlich die zweitbeste Parametrisierung bezüglich der maximalen Netzliste ausgewählt.

In Tabelle C- 6 ist für jedes Knotengewicht jeweils die bezüglich maximaler und durchschnittlicher Netzlänge günstigste Parametrisierung angegeben.

Die bei den künstlich generierten Netzlisten erkennbare Korrelation von zunehmender Kantengewichtung bei steigendem Knotengewicht (vergleiche Tabelle C- 1) ist in Tabelle C- 6 nicht erkennbar.

Anschließend wurde der Parameter zur Gewichtung der Pfadlänge für jede der Parametrisierungen aus Tabelle C- 6 gemäß Tabelle B- 2 variiert. In Tabelle C- 7 und Tabelle C- 8 ist eine Auswahl der entstehenden Layouts verglichen. Die beiden Tabellen basieren auf denselben Daten, lediglich die Sortierung erfolgt nach unterschiedlichen Bewertungskriterien.

Wie in Tabelle C- 8 zu sehen ist, bringt die Nichtbeachtung der Lage der Knoten in der Pfadbewertung (`NodeWeight=0`) die kürzesten durchschnittlichen Netzlängen hervor und fördert mit fünf der bezüglich der maximalen Netzlänge besten zehn Konfigurationen offenbar auch die Reduktion der maximalen Netzlänge.

Die Auswahl der beiden im weiteren verwendeten Konfigurationen soll, im Gegensatz zum Vorgehen für die künstlich erzeugten Netzlisten, nicht durch die jeweils beste Parametrisierung bezüglich eines der zwei Bewertungskriterien gegeben sein. Da die zur Reduktion der maximalen Netzlänge beste Parametrisierung aus Tabelle C- 7 starke Ähnlichkeit mit den neun der bezüglich der durchschnittlichen Netzlänge günstigsten Parametrisierungen hat, wird statt der besten Konfiguration aus Tabelle C- 8 die zweite aus Tabelle C- 7 für die weiteren Betrachtungen gewählt. Diese Konfiguration berücksichtigt durch das von Null verschiedene Knotengewicht `NodeWeight`, im Gegensatz zur ersten Parametrisierung aus Tabelle C- 7 auch die Lage der Pfade relativ zum Netzmittelpunkt. Dieser in dieser Arbeit in ESPRO neu eingeführte Bewertungsaspekt soll nicht bereits in einem frühen Stadium der Untersuchungen außer Acht gelassen werden.

Als Grundlage der weiteren Parameterwahl ergeben sich damit die beiden obersten, grau hervorgehobenen, Konfigurationen aus Tabelle C- 7.

6.1.1.4 Parametrisierung der Plazierungsmodifikation anhand von pk690

Basierend auf den beiden ausgewählten Parametrisierungen (vergleiche Abschnitt 6.1.1.3) wurde, ebenso wie für die in Abschnitt 6.1.1.2 dargelegten Ergebnisse, die in 5.3.1.2 beschriebene Suche nach geeigneten Parametrisierungen für die Plazierungsveränderung durchgeführt. Da die Auswahl der zugrundeliegenden Pfadbewertungsparametrisierungen nicht auf der Chipgröße, sondern alle auf den Netzlängen beruhen, werden die Parametrisierungen der Plazierungsmodifikation ebenfalls nur bezüglich der Netzlängen betrachtet, wobei die maximale Netzlänge wiederum das wichtigere der beiden Kriterien darstellt. Tabelle C- 9 und Tabelle C- 10 zeigen eine Auswahl der Resultate. Die Tabelle C- 9 ist nach maximaler Netzlänge sortiert, die Tabelle C- 10 nach der durchschnittlichen Netzlänge. In beiden Tabellen sind jeweils die Parametrisierungen basierend auf beiden im vorigen Abschnitt beschriebenen Pfadbewertungsparametrisierungen verglichen. Die Parametrisierungen basierend auf der besten Konfiguration aus Tabelle C- 7 sind in der Spalte Pfadbewertung mit „edge“ gekennzeichnet, da hier nur die Eigenschaften entlang des Pfades, also der Kanten im Routinggraph (vergleiche Abschnitt 4.3.1.4) betrachtet werden. Entsprechend sind die auf der zweitbesten Parametrisierung aus Tabelle C- 7 basierenden Konfigurationen in Tabelle C- 9 und Tabelle C- 10 mit „node“ gekennzeichnet, da hier die Lage der verwendeten Feedthroughs, also die Eigenschaften der Knoten im Routinggraph, bei der Pfadbewertung mit berücksichtigt werden.

Entgegen dem anhand der theoretischen Netzliste `test500` ermittelten Ergebnis bewirkt für die praktische Netzliste `pk690` ein Zulassen von Plazierungsmodifikationen durch den Verdrahter nicht zwingend eine Verschlechterung des Layouts. Mit kleinen Suchradien für die Zellverschiebung werden ebenso gute Ergebnisse erzielt, wie mit Parametrisierungen, die keine Plazierungsmodifikationen zulassen. Für größere Suchradien ergibt sich jedoch auch für die praktischen Netzlisten eine Verschlechterung des Ergebnisses.

Wie bereits für die Netzliste `test500` beobachtet, hat die Verschiebung von Zellen zur Schaffung weiterer möglicher Feedthroughpositionen keinen signifikanten Einfluß auf das Ergebnis. Selbiges gilt auch für den zugelassenen Suchradius für ein Feedthrough, welcher für `pk690`, genau wie für `test500`, keinen wesentlichen Einfluß zeigt. Dies läßt auf eine ausreichende Anzahl an Lücken für Feedthroughs in den Zellreihen schließen, was sich durch in Augenscheinnahme der zugrundeliegenden Plazierung bestätigen läßt.

Den Ergebnissen in Tabelle C- 9 und Tabelle C- 10 ist des weiteren zu entnehmen, daß die Einbeziehung der Lage der Verbindungen relativ zum Netzmittelpunkt keine Verbesserung der Verdrahtung erzielt. Die besten Verdrahtungen wurden unabhängig vom Grad der Plazierungsmodifikationen mit solchen Parametrisierungen der Pfadbewertung erzielt, welche nur die Eigenschaften längs des Pfades betrachten und die Lage der Verbindung relativ zum Netzmittelpunkt unberücksichtigt lassen. So finden sich die Verdrahtungen mit Berücksichtigung der Pfadlage erst ab dem vierten bzw. elften Rang in der Bewertung bezüglich maximaler bzw. durchschnittlicher Netzlänge.

Die Auswahl der Parametrisierungen für den Verdrahter erfolgt analog der Betrachtungen in Abschnitt 6.1.1.2 sowohl nach der Qualität des erzeugten Layouts, als auch unter dem Aspekt die in `prob` erstmals betrachteten Aspekte nicht unberücksichtigt zu lassen.

Es handelt sich dabei um die Parametrisierung, welche die besten Resultate erzeugte, die beste Parametrisierung ohne Modifikation der Plazierung und die beste Parametrisierung mit Betrachtung der absoluten Lage des Verdrahtungspfades. Für jede dieser Parametrisierungen wurde einmal die Verschiebung von Zellen zur Erzeugung weiterer möglicher Feedthroughpositionen zugelassen und einmal unterdrückt, so daß sich insgesamt sechs Parametrisierungen ergeben. Die gewählten sechs Parametrisierungen sind in Tabelle C- 12 zusammengefaßt.

Im weiteren werden von diesen sechs Parametrisierungen drei ausgewählt, die denen für die konstruierten Netzlisten möglichst gut entsprechen. Dies sind die Parametrisierung mit Modifikation der Plazierung und Zellverschiebung für neue Feedthroughpositionen und die besten Parametrisierungen mit Berücksichtigung der Pfadposition, sowohl mit als auch ohne Zellverschiebung für neue Feedthroughpositionen. Diese drei Parametrisierungen des Verdrahters werden im folgenden Vergleich für die praktischen Netzlisten verwendet. Sie ähneln in ihrer Charakteristik den Parametrisierungen für theoretische Netzlisten mit dem Ziel der Reduktion der maximalen Netzlänge.

6.1.2 Vergleich der Verdrahtungsergebnisse für verschiedene Plazierungen

In diesem Abschnitt werden die drei im vorangehenden Abschnitt beschriebenen verschiedenen Parametrisierungen des in `prob` implementierten Verdrahtungsalgorithmus miteinander und mit den Ergebnissen des `dkrouters` verglichen. Dafür werden mehrere Netzlisten mit drei verschiedenen Plazierern plziert und anschließend mit `prob` bzw. dem `dkrouter` verdrahtet. Für die theoretischen Netzlisten wurden die Parametrisierungen zur Reduktion der maximalen Netzlänge verwendet. Die drei folgenden Tabellen zeigen die Resultate für die drei verschiedenen Parametrisierungen. Als Vergleichsdaten dienen die in Tabelle 5 wiedergegebenen Ergebnisse des `dkrouters`, die aus [Rau00] entnommen wurden.

Tabelleneinträge, die keine Zahlenwerte, sondern ein „-“ enthalten zeigen an, daß keine Verdrahtung erstellt werden konnte. Für den `dkrouter` liegt dies an der Größe der zu verwaltenden Datenstrukturen der jeweiligen Netzliste. Die für `prob` unverdrahtbaren Plazierungen erforderten eine Verbreiterung eines vertikalen Kanals. Diese Möglichkeit ist in der verwendeten Version von `prob` nicht implementiert.

Tabelle 2: Ergebnisse des in prob implementierten Verdrahters für verschiedene Plazierer mit der Parametrisierung ohne jegliche Platzierungsmodifikation. Die Längenangaben verstehen sich in mm, die Flächenangaben in mm².

Plazierer	Great Deluge			Fuzzylogik Plazierer			Genetischer Plazierer		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
test100	4.31	1.84	3.22	3.62	1.61	3.03	4.58	1.78	3.19
test200	7.14	3.76	7.44	7.59	3.41	7.06	6.94	3.33	7.01
test500	17.98	8.65	30.35	17.40	7.59	28.17	18.56	8.50	30.01
test1000	45.44	19.59	69.33	35.30	16.26	60.15	46.89	18.88	66.95
test3000	128.41	50.78	1732.32	116.59	35.63	1302.13	124.78	48.83	1673.54
pk246	2.09	0.46	4.08	2.12	0.22	3.49	2.64	0.55	3.83
pk276	-	-	-	1.41	0.26	4.68	3.28	0.90	6.04
pk690	6.45	2.04	26.39	4.52	0.59	23.54	6.34	1.97	25.01

Tabelle 3: Ergebnisse des in prob implementierten Verdrahters mit Zulassen von Zellverschiebungen um neue Feedthroughpositionen zu erhalten, jedoch ohne weitere Platzierungsmodifikationen für verschiedene Plazierer. Die Längenangaben verstehen sich in mm, die Flächenangaben in mm².

Plazierer	Great Deluge			Fuzzylogik Plazierer			Genetischer Plazierer		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
test100	4.31	1.84	3.22	3.62	1.61	3.03	4.62	1.79	3.19
test200	7.17	3.74	7.43	7.59	3.41	7.06	6.91	3.32	7.00
test500	17.84	8.66	30.37	17.53	7.58	28.14	18.96	8.52	30.01
test1000	45.41	19.61	69.37	35.28	16.26	60.11	46.91	18.88	66.98
test3000	126.23	50.64	1727.78	116.18	35.69	1303.63	129.25	49.00	1677.68
pk246	2.09	0.47	4.08	2.14	0.22	3.49	2.68	0.55	3.87
pk276	-	-	-	1.41	0.26	4.69	3.36	0.90	6.04
pk690	6.46	2.07	26.39	4.50	0.60	23.54	6.52	1.95	25.01

Kapitel 6 Ergebnisse

Tabelle 4: Ergebnisse des in prob implementierten Verdrahters mit Modifikationen an der Platzierung für verschiedene Platzierer. Die Längenangaben verstehen sich in mm, die Flächenangaben in mm².

Platzierer	Great Deluge			Fuzzylogik Platzierer			Genetischer Platzierer		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
test100	4.91	1.94	3.30	3.89	1.82	3.29	4.65	1.95	3.36
test200	7.70	4.00	7.71	7.70	3.70	7.48	7.34	3.64	7.32
test500	17.89	8.85	30.70	16.56	7.74	28.48	20.76	8.90	30.98
test1000	45.08	19.54	68.80	39.25	16.81	61.79	47.63	19.49	68.34
test3000	127.92	50.94	1735.35	114.25	36.23	1321.43	125.21	49.24	1684.46
pk246	2.20	0.49	4.08	2.71	0.27	3.49	2.88	0.59	4.04
pk276	-	-	-	1.38	0.27	4.68	3.85	0.91	6.04
pk690	6.73	2.06	26.39	4.49	0.62	23.54	6.31	1.92	25.01

Tabelle 5: Ergebnisse des dkrouters für verschiedene Platzierer, aus [Rau00]. Die Längenangaben verstehen sich in mm, die Flächenangaben in mm².

Platzierer	Great Deluge			Fuzzylogik Platzierer			Genetischer Platzierer		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
test100	9.06	3.95	3.28	7.35	3.19	2.96	8.42	3.59	3.13
test200	14.89	7.80	7.04	12.87	7.01	6.48	15.59	7.35	6.65
test500	39.45	16.65	26.80	34.52	15.26	23.72	36.88	16.33	25.69
test1000	134.02	52.11	68.13	106.22	41.65	57.00	125.70	50.73	65.65
test3000	-	-	-	-	-	-	-	-	-
pk246	8.26	0.97	3.80	3.28	0.46	3.21	8.01	1.35	4.30
pk276	7.27	1.45	4.83	3.71	0.65	3.80	8.53	1.81	5.57
pk690	23.82	4.10	27.37	10.44	1.37	23.54	22.60	4.38	27.40

Die in Tabelle 2, Tabelle 3 und Tabelle 4 wiedergegebenen Resultate bestätigen die bei der Parameterevaluierung gewonnenen Erkenntnisse. Die im vorigen Abschnitt an nur zwei Beispielen abgeleiteten Charakteristika der mit verschiedenen Parametrisierungen von prob erstellten Layouts zeigen sich für alle betrachteten Netzlisten mehr oder weniger stark ausgeprägt. Die folgende kurze Diskussion der Tabellen legt die wichtigsten Aspekte dar.

Das Zulassen von Veränderungen der Plazierungen durch den Verdrahter führt nicht zu Verbesserungen der Ergebnisse. Zwischen den vollends unmodifizierten Layouts (vergleiche Tabelle 2) und den Layouts, für die ein Verschieben von Zellen zur Gewinnung neuer Feedthroughpositionen zugelassen ist (vergleiche Tabelle 3), bestehen kaum Unterschiede. Für einige Layouts, insbesondere für größere theoretische Netzlisten, ergeben sich geringfügige Verbesserungen durch die neuen Feedthroughpositionen. Für kleine theoretische und die praktischen Netzlisten bringen die neuen Feedthroughpositionen eher eine Verschlechterung des Gesamtergebnisses.

Wesentlich deutlicher ist der Unterschied zu den Layouts, die unter Zulassung einer Plazierungsänderung, die direkt die aktiven Zellen betrifft, erstellt wurden. Diese Layouts weisen überwiegend längere Netze und größeren Flächenbedarf auf, als die ohne Plazierungsmodifikation durch prob erstellten Layouts. Jedoch ist auch hier zu erkennen, daß für größere theoretische Netzlisten die Ergebnisse mit Plazierungsmodifikation besser sind, als für kleinere Netzlisten. Insbesondere fällt aber auf, daß die Plazierung des Fuzzylogik Plazierers der Netzlisten `test1000` und `test3000` mit Verdrahtungen, die Modifikationen an der Plazierung vornehmen, die besten Layouts des prob-Verdrahters hervorbringen. Dies legt den Schluß nahe, daß sich eine Modifikation der Plazierung nur für große Netzlisten lohnt. Dies zu bestätigen, bedarf weiterer Untersuchungen, die nicht Inhalt dieser Arbeit sind.

Die Ergebnisse des Verdrahters weisen alle deutlich kürzere Netzlängen auf, als die mit dem `dkrouter` erstellten Layouts. Die Werte der maximalen Netzlängen betragen zwischen 25% für die Great Deluge Plazierung der Netzliste `pk246` und 65% für die Fuzzylogik Plazierung der Netzliste `pk246` verglichen mit den Resultaten des `dkrouters`. Im Mittel sind die maximalen Netzlängen etwa halb so groß, wie in mit dem `dkrouter` erstellten Layouts.

Die deutliche Verbesserung der maximalen Netzlänge gegenüber den mit dem `dkrouter` erzeugten Layouts wird jedoch mit einem erhöhten Flächenbedarf von etwa 10% erkauft. Eine Ausnahme bilden die mit dem Great Deluge Algorithmus erstellten Plazierungen und die auf dem Genetischen Plazierer basierenden Layouts der praktischen Netzlisten, für die neben der Verkürzung der maximalen Netzlänge teilweise auch eine Reduktion des Platzbedarfes von 2 – 11% erreicht wurde.

Der Mehrbedarf an Fläche liegt zwischen 2% für die genetische Plazierung der Netzliste `test100` und 23% für die Fuzzylogik Plazierung der Netzliste `pk276`.

Der erhöhte Flächenbedarf liegt an der ungleichmäßigen Kanalauslastung. Die gewählte Parametrisierung fördert kurze Netzlängen, dabei wird die Kanalauslastung kaum berücksichtigt. Dies führt zu einer Ballung von Netzen in der Chipmitte, die zu einer starken Kanalauslastung führt. Die Anzahl der Tracks in einem Kanal in der Chipmitte bestimmt die Kanalbreite. Durch die breiten Kanäle steigt die Gesamtfläche des Layouts, wie beobachtet. Es ist zu erwarten, daß dieses Phänomen durch eine andere Parametrisierung, die die Kanalauslastung stärker berücksichtigt, beseitigt werden kann.

Die Ursache der extrem großen Chipflächen für die Netzliste `test3000` ist nicht vollends geklärt. Eine visuelle Inspektion der Layouts scheiterte, da das ESPRO-Modul `display` die Layouts aufgrund ihrer Größe nicht anzeigen konnte.

Der in den bisherigen Betrachtungen völlig unberücksichtigte Aspekt der benötigten Rechenzeit soll anhand der in der folgenden Tabelle 6 präsentierten Werte diskutiert

werden. Die benötigten Rechenzeiten für die Parametrisierung mit und ohne Zellverschiebung für neue mögliche Feedthroughpositionen unterscheiden sich nicht wesentlich, daher ist nur einer der beiden Meßwerte in der Tabelle 6 aufgeführt. Für die Rechenzeit des dkroueters für Plazierungen des Fuzzylogik Plazierers lagen keine Zeitangaben vor. Für den prob-Verdrahter war die Zeitangabe für die Kanalverdrahtung für große Netzlisten zum Teil nicht verfügbar.

Ein Vergleich der Rechenzeiten des prob-Verdrahters für Parametrisierungen mit und ohne Modifikationen der Platzierung zeigen keine wesentlichen Unterschiede. Einzige Ausnahme bildet die mit dem genetischen Algorithmus erstellte Platzierung. Für diese ist der Zeitbedarf für die Parametrisierung mit Platzierungsmodifikationen deutlich niedriger als ohne Modifikationen der Platzierung. Der Grund für dieses Verhalten ist nicht völlig geklärt. Naheliegend ist jedoch, daß die zeitintensive Suche in einem großen Routinggraphen abgekürzt wird, wenn der Verdrahter die beiden Zellen eines betrachteten Zweipunktnetzes dichter beieinander positioniert und so ein kleinerer Routinggraph zu untersuchen ist.

Für den prob-Verdrahter ist zu erkennen, daß ein Großteil des Zeitaufwandes für die Kanalverdrahtung aufgebracht wird. Dies ist auf die Datenstrukturen zur Verwaltung der Kanäle zurückzuführen, die für den Kanalverdrahter eine Laufzeitkomplexität von mindestens $O(k^2)$ erzwingen, wobei k die Anzahl der Tracks im Kanal angibt. Es ist zu erwarten, daß sich durch Verwendung besserer Datenstrukturen der Rechenzeitbedarf für die Kanalverdrahtung deutlich reduzieren läßt.

Tabelle 6: Rechenzeit für dkrouter und prob-Verdrahter in Sekunden für verschiedene Plazierungen. Ergebnisse aus dieser Arbeit und [Rau00]. Für den prob-Verdrahter sind die Zeit für Globalverdrahtung und die gesamte für Global und Detailverdrahtung benötigte Zeit getrennt ausgewiesen, der obere Wert gibt jeweils den Zeitbedarf für die Globalverdrahtung an.

Plazierer	Great Deluge			Fuzzylogik Plazierer			Genetischer Plazierer		
Verdrahter	dkrouter	ohne prob Plazierungsmodifikation	mit prob Plazierungsmodifikation	dkrouter	ohne prob Plazierungsmodifikation	mit prob Plazierungsmodifikation	dkrouter	ohne prob Plazierungsmodifikation	mit prob Plazierungsmodifikation
test100	<1	<1	<1	-	<1	<1	<1	<1	<1
test200	1	<1	<1	-	<1	<1	1	<1	<1
test500	9	1	1	-	2	1	9	2	1
		5	4		5	4		5	5
test1000	18	8	4	-	7	3	34	8	4
		19	16		16	13		19	16
test3000	-	316	339	-	169	173	-	1638	342
		-	-		1782	1858		-	-
pk246	<1	<1	<1	-	<1	<1	<1	<1	<1
pk276	<1	<1	<1	-	<1	<1	<1	1	1
								1	1
pk690	2	5	4	-	1	1	2	5	3
		6	5		2	1		7	5

6.2 Untersuchungsergebnisse für den Plazierer

Dieses Teilkapitel enthält die Ergebnisse der Plaziererevaluierung. Es werden, analog dem bereits für den Verdrahter angewendeten Vorgehen, die Ergebnisse der Suche nach günstigen Plazierungen des in prob implementierten Plazierers anhand der Netzlisten test500 und pk690 präsentiert und ein Vergleich mit bereits existierenden ESPRO-Modulen dargelegt.

Die folgenden Unterabschnitte legen die Ergebnisse der in den Abschnitten 5.3.2.1 bis 5.3.2.4 beschriebenen Evaluierungsschritte für je eine der beiden Netzlisten dar. In den Abschnitten 6.2.1 bis 6.2.3 wird die Parametrisierung dargelegt und in Abschnitt 6.2.4 ein Vergleich des in prob implementierten Plazierers mit bestehenden ESPRO-Plazierern durchgeführt.

6.2.1 Ergebnisse zur Parametrisierung der Standardzellplatzierungserstellung

Dieser Abschnitt legt die Ergebnisse der Parameterevaluierung für die Überführung der Vorplatzierung in eine Standardzellplatzierung, wie in Abschnitt 5.3.1.1 beschrieben, dar. Die beiden Unterabschnitte diskutieren jeweils die Resultate der theoretischen Netzliste `test500` bzw. der praktischen Netzliste `pk690`.

Die Auswahl der weiterzuverwendenden Parametrisierungen erfolgt nicht durch das Betrachten eines einzelnen Charakteristikums der entstehenden Layouts, sondern unter Betrachtung der Qualität des zugehörigen Layouts bezüglich maximaler Netzlänge einerseits und benötigtem Flächenbedarf andererseits. Die Auswahl fällt so nicht zwingend auf eine Parametrisierung, die bezüglich eines der Kriterien das beste Resultat hervorbringt, jedoch verspricht ein besseres Gesamtergebnis zu erzeugen.

6.2.1.1 Ergebnisse der Parameterevaluation anhand der Netzliste `test500`

Wie in Abschnitt 5.3.2.1 beschrieben, wurde zunächst in einem dreidimensionalen Suchraum nach geeigneten Parameterwerten für das Einfügen einer Zelle der Vorplatzierung in eine Standardzellreihe gesucht. Die Ergebnisse für die Netzliste `test500` mit 95%iger Reihenauslastung sind in Tabelle C- 13 und Tabelle C- 14 wiedergegeben.

Es zeigt sich, daß der Radius, in der nach geeigneten Lücken für die Feedthroughs gesucht wird, keinen signifikanten Einfluß auf die Qualität der Platzierung hat. Offenbar bewirkt die Verschiebung von bereits in der Zellreihe angeordneten Zellen beim Schaffen einer neuen Feedthroughposition keine wesentlichen Veränderungen der Platzierung..

Der Tabelle C- 14 ist zu entnehmen, daß die äquidistante Zellanordnung Platzierungen hervorbringt, die in Kombination mit dem `dkrouter` zu kompakteren Layouts führt, als Platzierungen mit variabler Feedthroughverteilung. Zu erklären ist diese Chipvergrößerung durch die Verbreiterung der Kanäle. Werden für viele Netze lange Verbindungen nötig, so vergrößert sich damit der Bedarf an Verdrahtungsressourcen und damit die Kanalbreite. Offenbar benötigt der `dkrouter` die Feedthroughs an anderen Stellen, als dies vom `prob`-Plazierer angenommen wird, so daß die äquidistante Zellanordnung für den `dkrouter` günstiger liegende Feedthroughpositionen enthält.

Für das Einfügen einer Zelle in die aktuelle Zellreihe erzielen Radian für die Suche nach einer geeignet großen Lücke in der Größenordnung der 5- bis 10-fachen mittleren Zellbreite die besten Resultate. Geeignete Suchentfernungen für einzufügende Feedthroughs liegen in der gleichen Größenordnung. Die genauen Werte sind in Tabelle C- 17 zusammengefaßt.

Es zeigt sich, daß die Überführung der Vorplatzierung in eine gültige Standardzellplatzierung maßgeblich die Qualität des Layouts bestimmt. Als Beispiel seien hier für die Netzliste `test500` das jeweils beste und schlechteste Layout bezüglich maximaler Netzlänge bzw. Chipgröße einander gegenübergestellt.

Die kürzesten auftretenden maximalen Netzlängen weisen eine Länge von nur 66% bezogen auf die längsten auftretenden maximalen Netzlängen auf. Für die Chipgröße gilt ähnliches, wenn auch deutlich weniger stark ausgeprägt. Das kleinste Layout benötigt nur 92% der vom größten Layout verwendeten Fläche.

Die Auswahl der weiterbetrachteten Parametrisierungen erfolgt, wie in Abschnitt 5.3.2.1 beschrieben, nach der Qualität des Layouts, gemessen an der maximalen Netzlänge und der Chipgröße. Zusätzlich zu der Parametrisierung, die das beste Layout hervorbringt, wird jeweils eine Parametrisierung gewählt, die unter Verwendung besonders kleiner bzw. besonders großer Suchradien für mögliche Einfügepositionen in eine Zellreihe gute Plazierungen hervorbringt. Für jede dieser drei Parametrisierungen des Einfügens einer Zelle der Vorplatzierung in die aktuelle Zellreihe, wird einmal die äquidistante Zellverteilung innerhalb der Zellreihe gefordert und einmal die durch den von prob geschätzten Feedthroughbedarf ermittelte Zellverteilung belassen. Die so entstehenden sechs Parametrisierungen sind in Tabelle C- 17 wiedergegeben. Die dort grau hervorgehobene Parametrisierung wurde für die weiteren Betrachtungen des Plazierers verwendet.

6.2.1.2 Ergebnisse der Parameterevaluation anhand der Netzliste pk690

Nach der Betrachtung der theoretischen Netzliste `test500` im vorangegangenen Abschnitt, wird nun das in Abschnitt 5.3.2.1 beschriebene Vorgehen für die Netzliste `pk690` untersucht. Die Ergebnisse für 95%ige Reihenauslastung sind in Tabelle C- 15 und Tabelle C- 16 wiedergegeben.

Es zeigen sich die gleichen Charakteristika für Parametrisierungen, wie für die Netzliste `test500`. Die Forderung nach äquidistanter Zellanordnung bringt auch für die praktische Netzliste kompaktere Layouts hervor. Ebenfalls analog dem zu `test500` gesagten hat die äquidistante Zellanordnung kaum signifikanten Einfluß auf die Layoutqualität bezüglich der maximalen Netzlänge.

Die günstigsten Suchradien für das Einfügen einer Zelle in die aktuelle Zellreihe liegen in der gleiche Größenordnung, wie auch bei `test500`. Dabei ist im Vergleich zur theoretischen Netzliste jedoch die Tendenz zu etwas größeren Suchradien festzustellen. Dies verstärkt sich mit abnehmender Reihenauslastung, vermutlich begründet durch die damit einhergehende Vergrößerung des Layouts.

Ein Vergleich der Layoutqualität von ähnlichen Parametrisierungen zeigt, daß die Prägnanz der Parameter gering ist. So bewirkt die Vergrößerung des Suchradius für die Feedthroughposition bei der drittplatzierten Parametrisierung aus Tabelle C- 16 um 50%, daß das mit der neuen Parametrisierung entstehende Layout das mit dem größten Flächenbedarf ist. Für die theoretischen Netzlisten sind die Parametrisierungen stabiler.

Es werden wiederum sechs Parametrisierungen ausgewählt. Die Parametrisierungen mit großen Suchradien weisen dabei eine große Instabilität auf. Die gleiche Parametrisierung führt, bei Reduktion der Reihenauslastung von 95% auf 90%, zu Layouts von völlig anderer Qualität. Dabei treten sowohl deutliche Verbesserungen als auch deutliche Verschlechterungen auf. Die wie in Abschnitt 5.3.2.1 und Abschnitt 6.2.1.1 beschrieben getroffene Auswahl an Parametrisierungen ist in Tabelle C- 18 wiedergegeben. Die im weiteren für die Betrachtungen des Plazierers verwendete Parametrisierung ist wiederum grau hervorgehoben.

6.2.2 Ergebnisse zur Parametrisierung der hierarchischen Partitionierung

Dieser Abschnitt präsentiert die Ergebnisse der Parametrisierung des Platzierungsschrittes der hierarchischen Partitionierung wie sie in Abschnitt 5.3.2.2 beschrieben wurde. Die beiden Unterabschnitte legen die Ergebnisse jeweils für die theoretische Netzliste `test500` und die praktische Netzliste `pk690` dar.

Die kräftegesteuerte Platzierung wurde für diesen Evaluierungsschritt komplett deaktiviert. Die Überführung der Vorplatzierung in eine Standardzellplatzierung erfolgt mit der in Abschnitt 6.2.1 ermittelten Parametrisierung.

6.2.2.1 Ergebnisse der Parameterevaluation anhand der Netzliste `test500`

Die Untersuchung des wie in Abschnitt 5.3.2.2 beschriebenen definierten Suchraumes zeigte, daß nur 538 der insgesamt 1331 Parametrisierungen durch den `dkrouter` verdrahtbar waren.

Die in Abschnitt 5.3.2.1 zunächst angenommene Parametrisierung der Vorplatzierung stellt sich als wenig geeignet heraus. Wie Tabelle 7 zu entnehmen ist, weisen Parametrisierungen von Layouts mit kurzen maximalen Netzlängen ein größeres Gewicht für die externen geschnittenen Netze als für die internen geschnittenen Netzen auf. Die Parametrisierungen mit größerem Gewicht auf den internen geschnittenen Kanten, genau wie die in Abschnitt 5.3.2.1 angenommene Parametrisierung, bringen deutlich schlechtere Layouts hervor.

Der Einfluß der Gewichtung der Kardinalitätsdifferenz der erzeugten Bipartitionen auf die Qualität des Layouts bezüglich der maximalen Netzlänge ist gering. Dies ist nicht verwunderlich, da nur für Bipartitionen, die eine größere Anzahl der sogenannten externen Netze aufweisen, eine unausgewogene Kardinalität der entstehenden Teilmengen möglich ist.

Tabelle 7: Resultate für Parametrisierungen der hierarchischen Partitionierung gemäß Tabelle B- 5, für die Netzliste `test500`, geordnet nach maximaler Netzlänge. Die Längenangaben verstehen sich in mm, die Flächenangaben in mm².

Rang	Maximale Netzlänge	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	20	75	30	36.704	16.985	25.978	
2	5	5	75	38.202	17.316	26.169	
3	1	2	1	40.019	17.805	27.502	
4	5	10	5	40.019	17.805	27.502	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
535	75	30	5	74.850	19.079	30.438	
536	100	30	5	75.113	21.909	40.622	
537	75	20	50	76.203	22.325	44.461	
538	30	5	75	80.669	23.236	46.608	

Die Bewertung der entstandenen Layouts anhand der beiden anderen Kriterien zeigt, daß diese Parametrisierungscharakteristika auch bezüglich der durchschnittlichen Netzlänge und dem Flächenbedarf auftreten. Die Ergebnisse dazu sind in Tabelle C- 19 und Tabelle C- 20 wiedergegeben.

Die Auswahl der jeweils sieben besten Parametrisierungen aus Tabelle 7, Tabelle C- 19 und Tabelle C- 20 liefert, aufgrund der auftretenden Überschneidungen, 17 Parametrisierungen für die in Abschnitt 5.3.2.3 beschriebene und weiter unten diskutierte Evaluierung der kräftegesteuerten Platzierung.

Die in Tabelle 7, Tabelle C- 19 und Tabelle C- 20 grau unterlegte Parametrisierung wird als Grundlage des ersten Untersuchungsschrittes für die kräftegesteuerte Platzierung gemäß Abschnitt 5.3.2.3 verwendet. Dies ist auch die in Tabelle C- 32 angegebene Parametrisierung der Vorplatzierung unter ausschließlicher Verwendung der hierarchischen Partitionierung.

6.2.2.2 Ergebnisse der Parameterevaluation anhand der Netzliste pk690

Von den untersuchten 1331 Parametrisierungen des dreidimensionalen Suchraumes, führen mit 1168 Wertebelegungen wesentlich mehr zu einer durch den `dkrouter` verdrahtbaren Platzierung, als dies für die Netzliste `test500` der Fall ist.

Die Ergebnisse der Parametrisierungen für die Netzliste `pk690` zeigen, daß es wiederholt Gruppen von Parameterkonstellationen gibt, die identische Platzierungen hervorbringen. Dies liegt daran, daß Parametrisierungen, die unter Beibehaltung des Verhältnisses der Werte der variierten Parameter, vielfache der Gewichte verwenden, eigenständig bewertet werden, obwohl sie die gleiche Wirkung auf den Algorithmus haben.

Für das Kriterium der maximalen Netzlänge liefern Parametrisierungen der gleichen Charakteristik, wie bereits bei `test500`, gute Resultate. Die schlechten Parametrisierungen folgen keiner so klaren Charakteristik wie es für die theoretische Netzliste beobachtet wurde.

Parametrisierungen, die Layouts mit kleinem Flächenbedarf hervorbringen, zeigen mehrheitlich einen hohen Wert des Gewichtungsparameters für die Kardinalitätsdifferenz. Die für `test500` beobachtete charakteristische starke Gewichtung der externen geschnittenen Verbindungen ist nicht zu beobachten. Die Parametrisierungen der schlechten Layouts folgen keinem ausgeprägtem Muster.

Für die Parametrisierungen der Layouts mit kurzen durchschnittlichen Netzlängen sind die Gewichte für interne geschnittene Verbindungen und die Kardinalitätsdifferenz mehrheitlich größer als für die externen geschnittenen Verbindungen. Schlechte Parametrisierungen zeichnen sich durch gleiche Gewichte für interne und externe geschnittene Verbindungen aus.

Im Gegensatz zu den Eigenschaften von `test500` läßt sich für die praktische Netzliste `pk690` keine einzelne Parametrisierung auswählen, die für alle drei Bewertungskriterien sehr gute Resultate hervorbringt. Die gewählte Parametrisierung liegt bezüglich Chipfläche und maximaler Netzlänge jeweils unter den besten 20 Parametrisierungen. Die erzielte durchschnittliche Netzlänge liegt mit 112% der Länge der besten Plazierungen unter den besten 10% der Plazierungen. Die genaue Wertebelegung der Parameter ist in Tabelle C- 33 angegeben.

Es sind wiederum die besten sieben Parametrisierungen jedes Bewertungskriteriums ausgewählt worden, damit ergeben sich 21 Parametrisierungen für die folgende Evaluierung der kräftegesteuerten Platzierung.

Die für die Beurteilung der kräftegesteuerten Platzierung gemäß Abschnitt 5.3.2.3 verwendete Parametrisierung ist in Tabelle C- 21, Tabelle C- 22 und Tabelle C- 23 grau unterlegt dargestellt.

6.2.3 Ergebnisse zur Parametrisierung der kräftegesteuerten Platzierung

Dieser Abschnitt schildert die Ergebnisse der Evaluation der kräftegesteuerten Platzierung, wie sie in Abschnitt 5.3.2.3 beschrieben ist. In den beiden folgenden Abschnitten werden für jeweils eine Netzliste die Resultate für die Beurteilung der Iterationsanzahl an einem Beispiel durchgeführt, sowie ein Vergleich der, wie in Abschnitt 5.3.2.2 dargelegt, ermittelten Stichprobe von Parametrisierungen für verschiedene Iterationsanzahlen angestellt.

6.2.3.1 Ergebnisse der Parameterevaluation anhand der Netzliste `test500`

Ausgehend von der in Abschnitt 6.2.2.1 gewählten Parametrisierung wurde die Netzliste `test500` mit verschiedenen Iterationszahlen der kräftegesteuerten Methode gemäß Tabelle B- 6 platziert.

Die Tabelle C- 24 zeigt die elf mit dem `dkrouter` verdrahtbaren Plazierungen. Es ist zu erkennen, daß die Qualität der Layouts bezüglich aller drei Kriterien mit zunehmender Iterationsanzahl deutlich abnimmt. Des weiteren ist Tabelle C- 24 zu entnehmen, daß die Erhöhung der Anzahl der Iterationen über einen Wert von wenigen Hundert nicht zu einer

deutlich signifikanten Veränderung der Bewertungskriterien führt. Es bringt offenbar keinen Vorteil, sehr viele Iterationen der kräftegesteuerten Platzierung durchzuführen. Des weiteren ist zu erkennen, daß die unverdrahtbaren Platzierungen meist eine mittelgroße Anzahl von Iterationen aufweisen. Dies läßt den Schluß zu, daß die Änderungen an der Platzierung zunächst gering sind. Der Einfluß der Startplatzierung, erzeugt durch die hierarchische Partitionierung, ist nach wenigen Iterationen noch dominierend. Nach weiteren Iterationen setzt sich die durch die kräftegesteuerte Platzierung aufgeprägte Charakteristik stärker durch. Das Resultat sind die schlechten oder unverdrahtbaren Platzierungen nach einigen zig Iterationen. Erst nach einigen Hundert Iterationen tritt der Einfluß der Startplatzierung soweit in den Hintergrund, daß wieder bessere Platzierungen hervorgebracht werden. Dies legt die Vermutung nahe, daß sich die Charakteristika der mit der hierarchischen Partitionierung erstellten Platzierungen deutlich von denen der mit der kräftebasierten Methode weiterbearbeiteten unterscheiden. Die Änderungen an der Platzierung in einer Iteration sind jedoch so gering, daß sehr viele Iterationen nötig sind, um eine gegenüber der Startplatzierung deutlich veränderte Platzierung zu erzeugen. Die gewählte Implementierung der kräftegesteuerten Platzierung erzeugt offenbar auch nach mehreren Hundert Iterationen noch immer Platzierungen, die stark von der durch die hierarchische Partitionierung gegebene Startplatzierung beeinflusst sind.

Tabelle C- 25, Tabelle C- 26 und Tabelle C- 27 zeigen eine Auswahl der Ergebnisse des Vergleiches der 18 Startplatzierungen mit jeweils allen in Tabelle B- 6 aufgeführten Iterationsanzahlen. In Tabelle C- 25 ist zu erkennen, daß die unmodifizierten Startplatzierungen bezüglich der Chipgröße die besten Platzierungen hervorbringen. Die sich in der Tabelle C- 27 anschließenden Platzierungen wurden mit mehreren Hundert Iterationen der kräftegesteuerten Methode bearbeitet. Das Kriterium der maximalen Netzlänge zeigt die gleichen Charakteristika, jedoch deutlich weniger ausgeprägt. So findet sich die schlechteste Startplatzierung bezüglich der maximalen Netzlänge in Tabelle C- 25 auf Rang 96, bezüglich der Chipgröße in Tabelle C- 27 dagegen auf Rang 20 von 122 verdrahteten Platzierungen. Die Betrachtung der Qualitäten bezüglich der durchschnittlichen Netzlänge in Tabelle C- 26 bestätigt das bereits für die Chipgröße Gesagte. Die maximale Netzlänge wird durch ein einziges Netz bestimmt und ist bei der hier vorliegenden Stichprobe offenbar starken Schwankungen unterworfen und so als aussagekräftiges Qualitätsmaß kaum verwendbar. Das zunächst nur am Beispiel beobachtete Verhalten zeigt sich also auch in der Stichprobe von 18 verschiedenen Parametrisierungen der hierarchischen Partitionierung.

Für die weitere Plaziererbeurteilung werden zwei Parametrisierungen ausgewählt, die sich nur in der Anzahl der Iterationen bei der kräftegesteuerten Platzierung unterscheiden. Die gewählten Parametrisierungen sind in Tabelle C- 25, Tabelle C- 26 und Tabelle C- 27 grau unterlegt. Ihre genauen Wertebelegungen sind in Tabelle C- 32 gegeben.

6.2.3.2 Ergebnisse der Parameterevaluation anhand der Netzliste pk690

Die Parameterevaluation für die Netzliste pk690 zeigt für die Chipgröße dieselben Charakteristika, wie sie auch für test500 zu beobachten sind. Wie schon zuvor, ist der Einfluß der Wertebelegungen der Parametrisierungen weniger stark ausgeprägt.

Die in Tabelle C- 28 gezeigte Entwicklung der Bewertungskriterien für verschiedene Iterationsanzahlen entspricht dem in Tabelle C- 24 bereits für die Netzliste test500

beobachteten Verhalten. In Tabelle C- 28 ist für die praktische Netzliste jedoch eine Tendenz zu steigenden durchschnittlichen Netzlängen bei zunehmender Iterationszahl zu erkennen.

Diese Beobachtung bestätigt sich im Vergleich mit der in Tabelle C- 29, Tabelle C- 30 und Tabelle C- 31 präsentierten Stichprobengruppe. Der Vergleich der Plazierungen bezüglich der maximalen Netzlänge zeigt, daß nach wenigen Iterationen der kräftegesteuerten Plazierung, entgegen dem bei den theoretischen Netzlisten beobachteten Verhalten, deutliche Verbesserungen eintreten. Für die anderen Bewertungskriterien gilt dies jedoch nicht. Wie bereits für die Netzliste `test500` beobachtet, bedarf es einer großen Anzahl von Iterationen, um eine Plazierung zu erzeugen, die deutlich von der Startplatzierung abweichende Eigenschaften hat. Dies zeigt, daß die implementierte Methode der kräftegesteuerten Plazierung merklich von der Startplatzierung abhängt.

Analog dem für die theoretische Netzliste verwendeten Vorgehen wird für den Plazierer eine Parametrisierung ohne und eine mit kräftegesteuerter Plazierung weiterverwendet. Die Parametrisierung der ausschließlich mit der Methode der hierarchischen Partitionierung erstellten Plazierung wurde in Abschnitt 6.2.3.1 beschrieben. Für die mit der kräftegesteuerten Plazierungsmethode weiterbearbeitete Plazierung wurde die bezüglich Flächenbedarf und maximaler Netzlänge beste Parametrisierung mit mindestens 250 Iterationen verwendet. Die Resultate dieser Parametrisierung sind in Tabelle C- 29, Tabelle C- 30 und Tabelle C- 31 grau unterlegt. Die komplette Parameterbelegung zur Erzeugung der Vorplatzierung für praktische Netzlisten ist in Tabelle C- 33 angegeben.

6.2.4 Vergleich mit anderen ESPRO-Plazierern

Wie in Abschnitt 5.3.2.4 beschrieben, wurde mit den in Tabelle C- 32 und Tabelle C- 33 ausgegebenen Parametrisierungen für jede Netzliste eine Plazierung erstellt und mit dem `dkrouter` verdrahtet.

Die Plazierungen der Netzliste `test3000` konnten vom `dkrouter` nicht verdrahtet werden, bereits während des Einlesens der Plazierungen wurden interne Datenstrukturgrößen überschritten. Die maximale und durchschnittliche Netzlänge sowie die Chipgröße der übrigen entstandenen Layouts sind in Tabelle 8 und Tabelle 9 wiedergegeben.

Tabelle 8: Ergebnisse des in `prob` implementierten Plazierers, Verdrahtung mit dem `dkrouter`.

Plazierungsmethode	Nur hierarchische Partitionierung			Hierarchische Partitionierung und kräftegesteuerte Methode		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
test100	9.592	4.145	3.714	10.548	4.545	3.679
test200	22.008	9.698	10.465	21.555	11.084	12.804
test500	36.704	16.985	25.978	49.544	18.550	26.854
test1000	91.436	34.771	52.112	107.261	44.006	67.460
test3000	-	-	-	-	-	-
pk246	5.730	0.575	3.714	4.427	0.742	3.856
pk276	5.445	0.879	3.926	7.904	1.718	6.868
pk690	16.099	2.580	27.814	15.048	3.526	26.763

Tabelle 9: Ergebnisse des in `prob` implementierten Plazierers, mit Parametrisierungen, die ein Bewertungskriterium bevorzugt verbessern. Die Verdrahtung erfolgte mit dem `dkrouter`.

Bevorzugtes Kriterium	Maximale Netzlänge			Chipgröße		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
pk246	5.238	0.573	4.110	6.736	0.854	5.248
pk276	6.110	0.806	4.362	4.874	0.840	3.815
pk690	15.115	2.547	29.157	20.554	2.466	24.697

Die in Tabelle 9 aufgeführten Ergebnisse verwenden Parametrisierungen, die darauf ausgelegt sind, eines der Bewertungskriterien bevorzugt zu verbessern. Ein Vergleich der Ergebnisse aus Tabelle 9 mit den Resultaten der nur mit hierarchischer Partitionierung platzierten Layouts aus Tabelle 8 zeigt, daß das Bevorzugen eines Kriteriums für kleine Netzlisten nicht den gewünschten Effekt erzielt und sogar eine deutliche Verschlechterung des betrachteten Kriteriums zur Folge haben kann. So ist die Chipfläche für die Netzliste `pk246` in Tabelle 9 für beide Parametrisierungen größer, als für die in Tabelle 8 gezeigte Parametrisierung.

Für die größte der betrachteten praktischen Netzlisten `pk690` ist in Tabelle 9 jeweils der bevorzugt berücksichtigte Wert merklich, wenn auch nicht wesentlich, kleiner, als in Tabelle 8. Dies wurde jedoch mit einer Verschlechterung in mindestens einem anderen der Bewertungskriterien erkaufte.

Ein Vergleich der beiden in Tabelle 8 gegenübergestellten Parametrisierungen zeigt, daß die unter Verwendung der kräftegesteuerten Methode erstellten Plazierungen, bis auf

einige Ausnahmen, in allen Kriterien schlechtere Werte aufweisen, als die nur mit hierarchischer Partitionierung erstellten Plazierungen.

Für den Vergleich der mit `prob` erstellten Plazierungen mit den in Tabelle 5 dargelegten Resultaten anderer ESPRO-Plazierer ist es sinnvoll, die Resultate für theoretische und praktische Netzlisten getrennt zu betrachten.

Betrachtet man die praktischen Netzlisten, so ist eine Einordnung des `prob`-Plazierers leicht. Die beiden verschiedenen in `prob` verwendeten Plazierungsmethoden liefern deutlich bessere Resultate, als der Great Deluge Plazierer oder der genetische Algorithmus. Die Qualität der vom Fuzzylogik Plazierer erzeugten Plazierungen wird jedoch bei weitem nicht erreicht. Wie weiter oben bereits erwähnt, liefert der `prob`-Plazierer ohne Verwendung der kräftegesteuerten Plazierung bessere Resultate, als die Variante mit Verwendung der kräftegesteuerten Methode.

Für eine Bewertung der Qualität des `prob`-Plazierers für theoretische Netzlisten spielt die Größe der Netzlisten, also die Zahl der enthaltenen Zellen und Netze, eine entscheidende Rolle. Für kleine Netzlisten (`test100` und `test200`) sind die mit `prob` erstellten Plazierungen in allen drei Kriterien deutlich schlechter, als die von einem der Vergleichsplazierer. Für die Netzliste `test500`, die für ESPRO als von mittlerer Größe angenommen werden kann, liegt die Parametrisierung von `prob` mit der kräftegesteuerten Methode noch abgeschlagen auf dem letzten Platz. Die `prob`-Variante, die nur die hierarchische Partitionierung verwendet, erreicht die Qualität des Genetischen Plazierers und erzeugt, mit nur geringfügig schlechteren Plazierungen als dieser, bereits Resultate von akzeptabler Qualität.

Für die Netzliste `test1000` ergeben sich, im Vergleich mit den bestehenden Plazierern die besten Resultate des `prob`-Plazierers. Die mit der Parametrisierungsvariante einschließlich der kräftegesteuerten Methode erstellte Plazierung liegt bezüglich der Chipfläche gleichauf mit dem Genetischen und dem Great Deluge Plazierer, bezüglich der maximalen Netzlänge erreicht sie fast den Wert der mit dem Fuzzylogik Plazierer erstellten Plazierung.

Die Parametrisierung des `prob`-Plazierers, die nur mit der Methode der hierarchischen Partitionierung plaziert, erreicht wiederum die deutlich besseren Ergebnisse. In allen drei herangezogenen Bewertungskriterien wird die Qualität der mit dem Fuzzylogik Plazierer erstellten Plazierung deutlich übertroffen. Das entstandene Layout weist eine um 14% kürzere maximale und eine um 16% kürzere durchschnittliche Netzlänge auf, als das beste bisher mit ESPRO-Modulen erzeugte Layout. Die benötigte Chipfläche liegt 8% unter dem Flächenbedarf des bisher besten Layouts dieser Netzliste.

Der Vergleich der in Tabelle 5 und Tabelle 8 präsentierten Layouts zeigt, daß sich die Qualität der mit dem `prob`-Plazierer erstellten Plazierungen mit zunehmender Netzlistengröße deutlich verbessert. Die Bestätigung dieses Trends scheitert an den Beschränkungen des `dkrouters`, der die Plazierungen der Netzliste `test3000` nicht verdrahten kann.

Die für große Netzlisten erzielte Qualität der Plazierungen ist insbesondere unter dem Aspekt der benötigten Rechenzeit bemerkenswert. Während für den bisher besten ESPRO-Plazierer extrem lange Rechenzeiten in Kauf genommen werden mußten, benötigt der

prob-Plazierer deutlich weniger Zeit. Für den Fuzzylogik Plazierer ist der Zeitbedarf so groß, daß in [Rau00] kein Meßwert mehr angegeben wird. Da die Laufzeitkomplexität für den zugrundeliegenden Algorithmus zwischen $O(n^3)$ und $O(n^4)$ liegt [Jen97], verstärkt sich das Laufzeitproblem für diesen Plazierer mit zunehmender Netzlistengröße deutlich. Der Tabelle 10 ist zu entnehmen, daß das Laufzeitverhalten des prob-Plazierers deutlich besser ist, als das des Fuzzylogik Plazierers. Die beiden übrigen Plazierer benötigen zwar deutlich weniger Zeit, als der in prob implementierte Plazierer, liefern jedoch, wie oben dargelegt, für große Netzlisten auch schlechtere Resultate.

Tabelle 10: Rechenzeit in Sekunden für existierende ESPRO-Plazierer und den prob-Plazierer. Basierend auf Ergebnissen aus dieser Arbeit und [Rau00].

Plazierer	Great Deluge	Fuzzylogik Plazierer	Genetischer Plazierer	prob Plazierer nur hierarchische Partitionierung	prob Plazierer mit kräftege-steuerter Methode
test100	<1	<1	1	<1	<1
test200	<1	6	1	1	1
test500	<1	115	3	7	9
test1000	1	1200*	4	235	234
test3000	-	-	-	384	409
pk246	<1	1	2	<1	<1
pk276	<1	3	2	<1	1
pk690	<1	52	4	1	1

*) Der Wert ist nicht durch die programmeigene Zeitmessung entstanden, sondern mit externer Zeitmessung bestimmt und daher ungenau.

6.3 Untersuchungsergebnisse für Plazierer und Verdrahter gemeinsam

Dieses Teilkapitel enthält die Ergebnisse der Evaluierung des kompletten in dieser Arbeit entwickelten Verfahrens. Es werden, im Gegensatz zu den vorherigen Ergebniskapiteln, keine neuen Parameterwerte untersucht. Die im Abschnitt 5.3.3 dargelegten Kombinationen von Wertebelegungen der Parametergruppen werden zunächst anhand der Netzlisten `test500` und `pk690` betrachtet. Anschließend werden mit ausgewählten Parametrisierungen die Resultate für alle Netzlisten dargelegt.

Die folgenden Unterabschnitte gliedern sich wie folgt: In den Abschnitten 6.3.1 und 6.3.2 wird zunächst anhand je einer Netzliste der Einfluß der Parametrisierungen auf den Algorithmus diskutiert. In Abschnitt 6.3.3 erfolgt ein Vergleich des in `prob` implementierten Verfahrens mit Kombinationen von bestehenden ESPRO-Plazierern und Verdrahtern.

6.3.1 Vergleich der Parametrisierungen anhand der Netzliste `test500`

In diesem Teilkapitel werden die in Tabelle 1 gezeigten Parametrisierungen für das Gesamtverfahren für theoretische Netzlisten betrachtet. Anhand der Netzliste `test500` wird für jede der beiden Platzierungsvarianten (mit und ohne die kräftegesteuerte Platzierung) die sich für alle Kriterien am günstigsten darlegende Parametrisierung ausgewählt. In Tabelle C- 34, Tabelle C- 35 und Tabelle C- 36 ist eine Auswahl der Ergebnisse dargelegt.

Es zeigt sich, daß innerhalb einer Platzierungsvariante nur geringe Unterschiede zwischen den gewählten Parametrisierungen auftreten. Den größten Einfluß, nach der Platzierungsmethode, hat offenbar der Algorithmusschritt zur Überführung der Vorplatzierung in die Standardzellplatzierung.

Die günstigsten Parametrisierungen verwenden für diesen Algorithmusschritt die bereits in der Kombination mit anderen ESPRO-Module erfolgreichen Parameterbelegungen. Einziger auffallender Unterschied für die ausschließlich mit hierarchischer Partitionierung erstellten Platzierungen ist, daß die Zellen innerhalb einer Zellreihe nicht äquidistant angeordnet werden.

Wiederum liefern die Parametrisierungen des Verdrahters, die Modifikationen an der Platzierung zulassen, die schlechtesten Resultate. Das Zulassen von Zellverschiebungen für die Schaffung neuer Feedthroughpositionen zeigt auch hier nur geringen Einfluß auf die Qualität der Layouts.

Auffällig an den Resultaten in Tabelle C- 34, Tabelle C- 35 und Tabelle C- 36 ist die große Chipfläche. Da diese unabhängig von der Parametrisierung ist, wird die Diskussion dieses Phänomens in Abschnitt 6.3.3 erfolgen.

Die zur weiteren Verwendung ausgewählten Parametrisierungen sind in Tabelle C- 34, Tabelle C- 35 und Tabelle C- 36 grau unterlegt dargestellt.

6.3.2 Vergleich der Parametrisierungen anhand der Netzliste `pk690`

In diesem Abschnitt werden, wie im vorangegangenen Abschnitt, die Parametrisierungen des Gesamtverfahrens gemäß Tabelle 1 betrachtet. Als Vertreter für die praktischen Netzlisten wird wiederum die Netzliste `pk690` verwendet. In Tabelle C- 37 und Tabelle

C- 38 sind die Resultate auszugsweise wiedergegeben. Die Layouts der praktischen Netzliste bestätigen das in Abschnitt 6.3.1 für die theoretische Netzliste `test500` Dargelegte. Als wesentlicher Unterschied zu der theoretischen Netzliste, jedoch in Übereinstimmung zu den in Abschnitt 6.1.1.4 präsentierten Ergebnissen, erzielen Parametrisierungen, die Modifikationen an der Platzierung vornehmen, gute Ergebnisse. Auch für die Netzliste `pk690` fällt der große Flächenbedarf der entstehenden Layouts auf. Dieses von der Parametrisierung unabhängige Phänomen wird im nächsten Abschnitt diskutiert.

Die für den Vergleich mit existierenden ESPRO-Modulen ausgewählten Parametrisierungen sind in Tabelle C- 37 und Tabelle C- 38 grau hervorgehoben.

6.3.3 Vergleich mit existierenden Plazierern und Verdrahtern

In diesem Unterkapitel wird das in dieser Arbeit entwickelte Verfahren mit existierenden ESPRO-Modulen verglichen. In der Tabelle 11 sind die Ergebnisse für die beiden Platzierungsvarianten mit und ohne kräftegesteuerte Methode gezeigt. Als Parametrisierungen wurden jeweils die in Tabelle C- 34, Tabelle C- 35 und Tabelle C- 36 bzw. Tabelle C- 37 und Tabelle C- 38 grau hervorgehobenen verwendet.

Die in Tabelle 11 wiedergegebenen Ergebnisse des in `prob` implementierten Verfahrens bestätigen die in den vorangegangenen Abschnitten einzeln abgeleiteten Erkenntnisse zu den beiden implementierten Modulen.

Tabelle 11: Ergebnisse des kompletten in `prob` implementierten Verfahrens mit Platzierung und Verdrahtung in einem Programmaufruf.

Platzierungs- methode	Nur hierarchische Partitionierung			Hierarchische Partitionierung und kräftegesteuerte Methode		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
test100	5.13	1.74	12.68	5.03	1.79	12.44
test200	8.45	3.44	25.67	8.13	3.67	26.03
test500	18.68	7.22	105.46	21.66	8.15	108.36
test1000 ¹⁾	47.77	17.77	149.76	44.66	17.64	148.79
test3000	312.83	40.37	10071.92	211.68	42.54	10776.68
pk246	4.08	0.33	9.36	4.16	0.40	12.30
pk276	4.34	0.35	14.50	7.27	0.55	32.73
pk690	10.82	0.97	65.54	19.55	1.68	143.37

1) Verbreiterung der vertikalen Kanäle war erforderlich.

Die maximalen und durchschnittlichen Netzlängen konnten, wie bereits in Abschnitt 6.1.2 dargelegt, im Vergleich zu anderen Verdrahtern deutlich verkürzt werden. Die bereits beim Verdrahter beobachtete Vergrößerung der Chipfläche ist hier mit etwa 350% gegenüber den mit dem `dkrouter` verdrahteten Layouts (vergleiche Tabelle 5) jedoch nicht mehr

akzeptabel. Wie an dem in Abbildung 58 gezeigten Layout zu erkennen ist, liegt dieser enorme Flächenbedarf jedoch nicht an der Bevorzugung der Netzlängen als Bewertungskriterium, sondern an einem Implementierungsfehler in der Kompaktifizierung des Layouts. Die im Laufe der Verdrahtung verbreiteten Kanäle führen zwischenzeitlich zu sehr großem Flächenbedarf. Das Eliminieren ungenutzter Tracks in den Kanälen nach der Kanalverdrahtung führt, wie in Abbildung 58 zu sehen ist, zu einer Reduktion der benötigten Fläche, jedoch nicht zur Verkleinerung der Abmessungen des Chips.

Diese Vergrößerung der Chipfläche hat Einfluß auf die maximale Netzlänge, wenn die langen Anschlußleitungen zu Pads an der unteren Chipkante Teil des längsten Netzes sind. Die durchschnittliche Netzlänge wird durch die wenigen langen Anschlußleitungen für Padzellen nicht wesentlich beeinflußt.

Das entstehende Layout auf dem genutzten Teil der Chipfläche ist kompakt und läßt die Vermutung zu, daß nach Beheben des Implementationsfehlers die Chipfläche nicht größer ausfällt, als bei der Verwendung anderer ESPRO-Module.

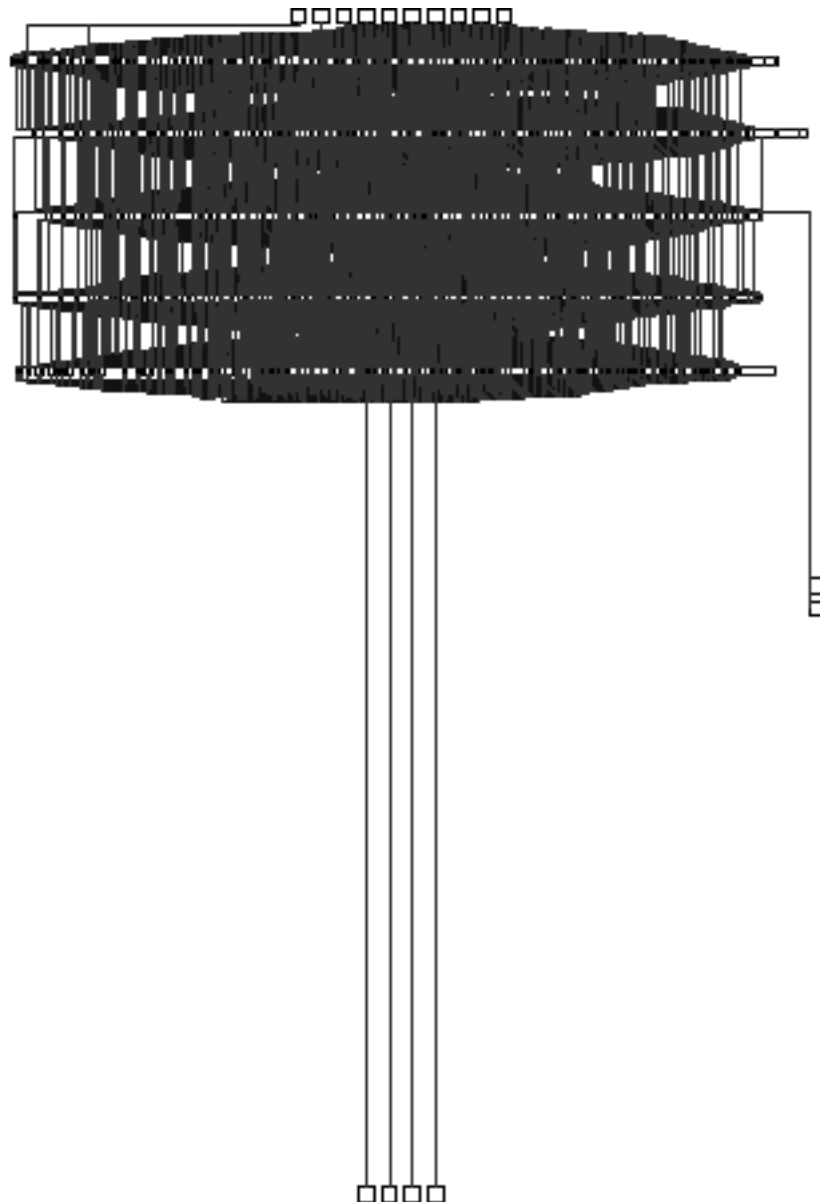


Abbildung 58: Das mit `prob` erstellte Layout für die Netzliste `test500`. Die große ungenutzte Fläche auf dem Chip ist deutlich zu erkennen.

6.4 Zusammenfassende Diskussion der Ergebnisse

Dieses Teilkapitel faßt die in den vorigen Kapiteln 6.1 bis 6.3 dargelegten Ergebnisse zusammen und diskutiert die durch das in dieser Arbeit erarbeitete Verfahren erzielten Resultate.

Der in `prob` implementierte Platzierungsalgorithmus zeigt, daß der in dieser Arbeit verfolgte Ansatz ein konkurrenzfähiges Verfahren darstellt.

Die Qualität der erstellten Platzierungen ist für größere Netzlisten deutlich besser als für kleine. Dies ist darauf zurückzuführen, daß mit zunehmender Chipfläche die Größe der Zellen relativ zum gesamten Layout abnimmt. Damit wird die Annahme, daß die aktiven Komponenten von vernachlässigbarer Fläche seien (vergleiche Abschnitt 2.5) um so besser

erfüllt, je größer der Chip ist. Für kleine Netzlisten ist die Zellgröße relativ zur Chipfläche offenbar zu groß, so daß die entstehenden Plazierungen nicht überzeugen können.

Der aus den Bedürfnissen der Verdrahtung abgeleitete Platzierungsalgorithmus (vergleiche Kapitel 2.4 und Abschnitt 4.1.2) zeigt seine Wirksamkeit für große Netzlisten durch die deutliche Verkürzung der Netzlängen. Betrachtet man die Layouts, so zeigt sich, abgesehen von der für Standardzellentwürfe ohne OTC-Routing üblichen geringeren Kanalauslastung am Rand der Kanäle, eine gleichmäßige dichte Ausnutzung der Verdrahtungsfläche. Die durch den Platzierungsalgorithmus angestrebte gleichmäßige Verteilung der Netze über die zur Verfügung stehende Fläche erzielt offenbar den gewünschten Effekt (vergleiche Abschnitt 4.2.1).

Die Verwendung von zwei verschiedenen Platzierungsalgorithmen nacheinander hat, zumindest für die hier verwendeten Algorithmen, keine Verbesserung der Platzierungsqualität zur Folge. Die Verwendung eines zweiten, iterativ verbessernden, Platzierungsalgorithmus läßt die vom ersten Algorithmus erzeugte Platzierung um so unwichtiger werden, je unabhängiger die Ausgabe des zweiten Algorithmus von seiner Startplatzierung ist. Die hier verwendete Implementierung der kräftegesteuerten Platzierung zeigt eine deutliche Abhängigkeit von der Startplatzierung, erzeugt jedoch keine Verbesserung der Platzierungen, so daß die ausschließliche Verwendung der hierarchischen Partitionierung günstiger ist.

Die dem Bedarf angepaßte Verteilung von Feedthroughs in den Standardzellreihen führt zu einer hohen Auslastung der Zellreihen. Die vom Plazierer ermittelte Lage der Feedthroughs innerhalb der Zellreihe deckt sich nur zum Teil mit dem Bedarf an Feedthroughs für die Verdrahtungsalgorithmen. Die Anzahl der Feedthroughs je Zellreihe ist jedoch für alle betrachteten Verdrahter völlig ausreichend. Diese, hier in ESPRO zum erstenmal eingesetzte Abschätzung des tatsächlichen Feedthroughbedarfes, stellt eine Erweiterung der Platzierungsheuristik für Standardzellentwürfe dar, die auch für andere Platzierungsalgorithmen eine Möglichkeit der Verbesserung bietet.

Die zellweise sequentielle Verdrahtungsreihenfolge bringt offenbar Vorteile für die Netzlänge. Verglichen mit dem `dkrouter` gelingt eine deutliche Reduktion von durchschnittlicher und maximaler Netzlänge. Die bei netzweise sequentieller Verdrahtung auftretende Benachteiligung von spät verdrahteten Netzen, läßt, übertragen auf die zellweise sequentielle Verdrahtung, eine Benachteiligung der spät verdrahteten Zellen erwarten. Die betrachteten Beispiele lassen keinen solchen negativen Einfluß auf das Layout erkennen.

Die erzielten kurzen Netzlängen resultieren in einer hohen Verdrahtungsdichte in der Mitte der Layouts und damit in sehr ungleichmäßigen Kanalauslastungen. Zum Rand des Chips hin sind die Kanäle nur gering ausgelastet, die große Zahl an Leitungen in der Mitte des Kanals bestimmt seine Breite und führt zu der beobachteten Vergrößerung der Chipfläche. Die für die netzweise Verdrahtung übliche Suche nach minimalen Bäumen zur Ermittlung kurzer Verdrahtungswege konnten bei dem hier verwendeten Verfahren entfallen und durch die reine Betrachtung von Zweipunktnetzen ersetzt werden.

Die Untersuchung des Verdrahters zeigt deutlich, daß das Durchführen lokaler und auf ein Netz fokussierter Modifikationen der Platzierung leicht eine Verschlechterung des gesamten Layouts zur Folge haben kann. Die in dieser Arbeit verwendeten Algorithmen zur Platzierungsmodifikation durch den Verdrahter verursachen durch die durchgeführten

lokalen Veränderungen, insbesondere für Layouts mit hoher Netzkomplexität, überwiegend Verschlechterungen der Layouts.

Die gemeinsame Verwendung des in dieser Arbeit entwickelten Plazierers und Verdrahters zeigt, daß sich keine besonderen Vorteile ergeben, wenn beide Teilprobleme des physikalischen VLSI-Entwurfs mit Algorithmen bearbeitet werden die auf der dualen Sichtweise basieren. Die komplett von prob erstellten Layouts zeigen lediglich die Eigenschaften beider Module, jedoch keine darüber hinausgehenden Charakteristika.

Die aufgrund des Implementierungsfehlers in der Kompaktifizierung unnötig große Chipfläche läßt sich durch Korrigieren des verantwortlichen Programmcodes deutlich verkleinern. In Anhang D sind die Ergebnisse einer korrigierten Programmversion gezeigt. Die Qualität, der mit der hier vorgestellten Implementierung des in dieser Arbeit entwickelten Verfahrens für das Plazieren und Verdrahten eines Standardzellentwurfs, zeigt die prinzipielle Tauglichkeit des verfolgten Ansatzes.

Die verwendeten Heuristiken weisen eine Vielzahl von Parametrisierungsmöglichkeiten auf. Trotz der ausführlichen Untersuchung des Einflusses der verschiedenen Parameter auf das Algorithmusverhalten, bleiben die verwendeten Wertebelegungen ein erster Ansatz. Die in den zum Vergleich herangezogenen Plazierungsheuristiken eingearbeiteten Parameter wurden über mehrere Jahre ermittelt und verbessert, so daß zu erwarten ist, daß sich die Qualität des hier präsentierten Algorithmus durch weitere Untersuchung der Parametrisierung noch verbessern läßt.

7. AUSBLICK

Dieses Kapitel schildert die sich aus den Ergebnissen dieser Arbeit aufzeigenden Perspektiven. Es gibt verschiedene Ansatzpunkte für weiterführende Arbeiten. Zunächst bietet sich die Möglichkeit, den bestehenden Prototyp weiter auszureizen. Als nächstes liegt es nahe, den Prototypen selbst in Details zu verbessern. Durch Änderungen des Verfahrens auf verschiedenen Granularitätsebenen, ergeben sich neue Varianten des Verfahrens, die neu implementiert und untersucht werden können.

Schließlich bleibt die Überlegung, in wieweit die momentane Modellierung noch zeitgemäß ist, so daß nach einer neuen Modellierung des betrachteten Entwurfsstils, was dann auch nicht zwingend der Standardzellentwurf sein muß, eine komplett überarbeitete Implementierung nötig ist, die wiederum vollständig neu zu untersuchen wäre.

Zu diesen verschieden weitreichenden Änderungen in der Problembetrachtung ist in den folgenden Teilkapiteln jeweils ein Überblick möglicher Aktivitäten dargelegt. Kapitel 7.1 zeigt weitere mögliche Evaluierungen am bestehenden Prototypen auf. Das Teilkapitel 7.2 macht den größten Anteil dieses Kapitels aus und schildert mögliche Erweiterungen der Algorithmen mit verschiedenen weitreichenden Änderungen des Verfahrens. Das Kapitel 7.3 hinterfragt die gewählte Modellierung und zeigt Erweiterungsmöglichkeiten und Alternativen. Den Abschluß bildet das Teilkapitel 7.4 mit dem Versuch einer Projektion der Ergebnisse dieser Arbeit in die Zukunft.

7.1 Weitere Evaluationsmöglichkeiten innerhalb des bestehenden Systems

Die Eigenschaften des in dieser Arbeit implementierten Prototyps des vorgestellten Verfahrens ist durch eine Vielzahl von Parametern beeinflussbar. Die Wertebelegung dieser Parameter bestimmt maßgeblich die Qualität der entstehenden Layouts. Die hier durchgeführte Evaluation der Parametrisierung stellt trotz ihres Umfanges nur eine erste Konfiguration dar. Es ist zu erwarten, daß weitere Untersuchungen die Qualität der erzielten Ergebnisse noch verbessern können. So ist der Einfluß der Kanalbreitenschätzung auf die Qualität der Verdrahtung komplett unberücksichtigt geblieben. Das Fortsetzen der Parameterevaluation oder das Verwenden neuer Schätzfunktionen ermöglicht es, den Einfluß der Kanalbreite auf die Eigenschaften der zellweise sequentiellen Verdrahtung weiter zu untersuchen.

Da die Wahl der Parameter, wie in Kapitel 6 ersichtlich ist, von den Eigenschaften der Netzliste abhängt, ist das Finden einer allgemeingültigen, für alle Netzlisten gleich günstigen, Parametrisierung nicht zu erwarten. Eine wünschenswerte Eigenschaft des Algorithmus wäre es, sich selbst an die Anforderungen der Netzliste anzupassen. Dafür wäre eine weitere Komponente zu implementieren, welche die Netzliste und gegebenenfalls die Platzierung analysiert und die Algorithmusparameter entsprechend einstellt. Zu einer solchen Komponente sind in ESPRO bisher noch keine Untersuchungen angestellt worden.

Über die Betrachtung der Parametrisierung des Algorithmus hinausgehend, können durch die Verwendung weiterer Netzlisten die Eigenschaften des Verfahrens für besondere Einsatzgebiete ermittelt werden. So ist es denkbar, durch die Erzeugung entsprechender

Netzlisten die Verwendung als Verdrahter für die Stromversorgung der Zellen (Power-Routing) zu untersuchen. Von Interesse wäre auch die Qualität der Ergebnisse für Netzlisten mit Zellen, die überdurchschnittlich viele Pins haben. Solche Zellen finden Verwendung, wenn auf Standardzellbibliotheken zurückgegriffen wird, die Komponenten hoher Komplexität enthalten, wie Volladdierer hoher Bitbreite.

Von besonderem Interesse sind dabei Netzlisten mit vielen Zellen. Es wäre wünschenswert, einen Vergleich für Netzlisten, die wesentlich mehr Zellen enthalten, als die bisher untersuchten, zu betrachten. Problematisch wird dies weniger wegen der Implementierung von `prob`, als durch die Beschränkungen anderer ESPRO-Module. Die Restriktionen der Implementierungen zum Vergleich herangezogener Algorithmen erfordern zum Teil erhebliche Modifikationen der existierenden ESPRO-Komponenten.

Die in dieser Arbeit verwendeten Kriterien zur Beurteilung der Qualität eines Layouts könnten um die Varianz der Netzlängen erweitert werden. Diese kann herangezogen werden, um das hier eingesetzte zellweise sequentielle Verdrahten mit dem üblichen netzweise sequentiellen Verfahren besser zu vergleichen können. Das Verfahren des hier implementierten Verdrahters, sollte eine merklich kleinere Varianz der Netzlängen zeigen. Für eine solche Untersuchung ist ein erweitertes Statistik-Modul für ESPRO erforderlich, welches die Varianz, zusätzlich zu den bisher ermittelten Werten, ausgibt.

Der in `prob` implementierte Verdrahtungsalgorithmus kann die Platzierung modifizieren. Die dabei entstehenden neuen Platzierungen werden separat in eine ESPRO-Platzierungsdatei ausgegeben. Eine Beurteilung der Platzierungsmodifikationen unabhängig von dem angewendeten Verdrahtungsalgorithmus ist somit möglich. Die vom `prob`-Verdrahter erzeugten Platzierungen müßten hierzu lediglich mit einem anderen Verdrahtungsalgorithmus bearbeitet werden, etwa dem `dkrouter`. Dieses Vorgehen ist insbesondere bei Erweiterung der Platzierungsmodifikation von Interesse.

7.2 Alternative Algorithmen für Teilschritte des Verfahrens

Dieses Teilkapitel schildert mögliche Änderungen oder Erweiterungen der für die Teilschritte des in dieser Arbeit entworfenen Verfahrens eingesetzten Algorithmen. Wie in Kapitel 4 und Teilkapitel 2.6 dargestellt wurde, sind für die einzelnen zu lösenden Teilprobleme des Gesamtverfahrens bewußt einfache Algorithmen verwendet worden. Nach dem abgeschlossenen Machbarkeitsbeweis, ist eine Verbesserung der Algorithmen der einzelnen Teilprobleme angezeigt. Die folgenden Abschnitte zeigen mögliche Ansatzpunkte für Verbesserungen der bestehenden Vorgehensweise.

7.2.1 Anpassen der Datenstrukturen

Die in der aktuellen Implementierung von `prob` verwendeten Datenstrukturen wurden unter dem Gesichtspunkt der Einfachheit ausgewählt. Die für fast alle Strukturen dynamischer Größe gewählte doppelt verkettete Liste ist für große Datenmengen ungeeignet. Wie die Laufzeit des Greedy Kanalverdrahters zeigt, ist die linear mit der Größe wachsende Zugriffszeit in dieser Datenstruktur ein deutliches Geschwindigkeitshemmnis. Insbesondere für den Kanalverdrahter, der durch häufige Suchoperationen in den Datenstrukturen der Kanäle sehr anfällig für die Zugriffszeit in den

Datenstrukturen ist, wäre eine effektiver zugreifbare Struktur wünschenswert, die eine von der Strukturgröße möglichst unabhängige Zugriffszeit aufweist.

Ähnlich zeitkritisch hat sich die Verwaltung der Leitungssegmente innerhalb der Tracks gezeigt. Hier ist durch die Verwendung von Baumstrukturen, z.B. B-Bäumen, eine Bescheunigung ohne Verlust der geforderten Dynamik möglich.

7.2.2 Andere Plazierung der Netze

In dieser Arbeit ist, ausgehend von einer Hypergraphenrepräsentation der Netzliste, der duale Hypergraph gebildet worden. Dieser ist, unabhängig von der veränderten Interpretationsweise, ein Hypergraph. Es ist daher prinzipiell denkbar, andere Plazierungsalgorithmen zu verwenden, als die beiden hier gewählten. Die in Kapitel 4 dargelegten geforderten Eigenschaften der Algorithmen, an die durch die duale Sichtweise veränderten Randbedingungen adaptierbar zu sein, müssen jedoch erfüllt werden. Es eröffnet sich somit eine Vielzahl von Möglichkeiten, die als Knoten des dualen Hypergraphen repräsentierten Netze mit anderen Algorithmen zu plazieren.

Der wachsenden Größe realer Netzlisten kann auch in den hier verwendeten Algorithmen durch Clusterung begegnet werden. Obwohl die in ESPRO bisher betrachteten Netzlisten vergleichsweise klein sind, ist es denkbar, auch wesentlich umfangreichere Netzlisten zu betrachten. Die Verwendung von Hypergraphen als Modellierungsmittel ermöglicht es, Ansätze, die für Plazierung von Zellen konzipiert wurden, auf die hier verwendete duale Sichtweise anzuwenden. So lassen sich die in [LLL96] und [HuL99] präsentierten Ansätze zur Reduktion der Größe des betrachteten Graphen durch Clusterung in das in dieser Arbeit verwendete Vorgehen einbinden und bewerten.

7.2.2.1 Änderungen an der hierarchischen Partitionierung

In der momentanen Implementierung des hierarchischen Partitionierungsalgorithmus erfolgen die Bipartitionen entlang Schnittlinien in der in Abschnitt 4.2.1 dargelegten Art. Dabei wird die Lage von benachbarten Netzen außerhalb der gerade betrachteten Teilmenge, welche die externen Verbindungen bestimmen, jeweils aus der aktuellen Aufteilung der Gesamtfläche ermittelt. Insbesondere werden dabei Positionen, die durch unterschiedlich viele Bipartitionen erzeugt wurden, miteinander verglichen. Es wäre allerdings wünschenswert, daß Bipartitionen der gleichen Hierarchieebene auch von der gleichen Ausgangssituation ausgehend durchgeführt werden.

Als eine Alternative zu dieser Änderung der hierarchischen Partitionierung, läßt sich die bestehende Implementation dahingehend erweitern, die Reihenfolge, in der die Bipartitionen gleicher Tiefe des Partitionierungsbaumes ausgeführt werden, zu modifizieren. Die vorliegende Fassung von `prob` bearbeitet die zu teilenden Mengen zeilenweise von links oben nach rechts unten in der betrachteten Ebene. Andere Reihenfolgen sind durchaus denkbar, etwa von außen nach innen, oder umgekehrt. Der Einfluß dieser Reihenfolge der Bearbeitung einer Hierarchieebene des Partitionierungsbaumes ist bisher unbetrachtet geblieben und bietet Raum für weitere Untersuchungen zu Detailverbesserungen des hier entwickelten Vorgehens.

Es ist nicht zwingend notwendig, die hierarchische Partitionierung durchzuführen, bis jedes Blatt des Partitionierungsbaumes maximal ein Netz enthält. So wird in [She89]

innerhalb der kleinsten erzeugten Kachel des Partitionierungsbaumes ein auf kombinatorischer Vertauschung basierender Algorithmus zum Plazieren verwendet. Es ist denkbar, dieses Verfahren auch in den hier vorgestellten Ansatz zu integrieren. Der hier implementierte Algorithmus zur hierarchischen Partitionierung ist bereits in soweit vorbereitet, als die Größe der kleinsten zu erzeugenden Kachel vorgegeben werden kann. In der hier präsentierten Version, ist diese Kachelgröße jedoch zu eins gewählt. Mögliche Algorithmen zur Plazierung der Netze innerhalb der kleinsten erzeugten Kacheln des Partitionierungsbaumes sind z.B. in [Ger99], [Oht86], [She89] und [Vyg98] enthalten.

Eine Schwäche des Plazierens mit der Methode der hierarchischen Partitionierung ist die feste Zuordnung von Bauteilen zu Regionen, die in späteren Schritten nicht aufgehoben werden kann. Wird ein Baustein in einem Bereich positioniert, der sich in späteren, feineren Iterationsschritten, als ungünstig erweist, so ist eine Verschiebung aus diesem Gebiet in Algorithmen mit reiner hierarchischer Partitionierung nicht möglich. Eine Abhilfe bietet ein zusätzlicher Schritt, der nach Beenden einer jeden Partitionierungsebene eingeschoben wird. In diesem Schritt wird untersucht, ob das Verschieben von Bausteinen in benachbarte Regionen (Reflow) eine Verbesserung der vorliegenden Plazierung bewirkt. Die möglicherweise nachteilhafte strenge Zuordnung von Bauteilen zu Regionen wird damit aufgehoben [DKS98]. Einen ähnlichen Ansatz stellt das Verschmelzen und Repartitionieren (Merge and Repartition) dar. Benachbarte Regionen werden dabei vereinigt und sofort wieder geteilt. Bei der Repartitionierung wird die Zuordnung der Bausteine zu den Regionen neu ermittelt [CKM00].

Diese beiden Erweiterungen der hierarchischen Partitionierung können unabhängig vom verwendeten Bipartitionierungsalgorithmus in das bestehende Verfahren integriert und ihr Einfluß auf die Layoutqualität evaluiert werden.

7.2.2.2 Änderung des Bipartitionierungsalgorithmus

Zu Heuristiken für die Erstellung von Bipartitionen von Mengen gibt es umfangreiche Untersuchungen und eine Vielzahl von Algorithmen [Alp98]. Die hier gewählte FM-Implementierung zeichnet sich durch ihre Einfachheit aus. Des weiteren läßt sich die in dieser Arbeit gewählte Bewertungsfunktion für die Qualität einer Bipartition in dieser Heuristik direkt anwenden. Die Schwächen der FM-Heuristik sind bekannt ([Alp98], [She95]) und in zahlreichen Erweiterungen des Basisalgorithmus abgemildert worden. Die starke Abhängigkeit der erzeugten Bipartition von der gewählten initialen Zuordnung der Netze zu den neuen Teilmengen ist eine der bekannten negativen Eigenschaften, sie läßt sich durch Erweiterung des Algorithmus mildern oder durch die Verwendung einer besseren Ausgangsbipartition in ihrer Bedeutung reduzieren. In [Zob94] und [Ger99] wird zu diesem Vorgehen einer nicht zufällig ermittelten, sonder bewußt konstruierten Startpartitionierung, Stellung genommen.

Die Wahl eines völlig anderen Algorithmus zur Erzeugung der Bipartitionen ist ein anderer gangbarer Weg, zumal, wie in [Alp98] dargelegt, andere Heuristiken dem FM-Algorithmus in Ausführungsgeschwindigkeit und Qualität der Ergebnisse überlegen sind. Insbesondere im Hinblick auf die Eignung des in dieser Arbeit vorgestellten Verfahrens für große Netzlisten ist es angebracht, die für Detaillösungen verwendeten Algorithmen dieser Problemgröße anzupassen. Für alternativ verwendbare Heuristiken sei hier auf die in [Alp98] angegebenen, sowie die unter [Kar01] erwähnten Veröffentlichungen verwiesen.

Je nach Qualität und Eigenschaften der verwendeten Bipartitionierungsheuristik können die im vorigen Abschnitt 7.2.2.1 erwähnten Erweiterungen der hierarchischen Partitionierung stark unterschiedlichen Einfluß haben. Im allgemeinen sind Methoden zur Aufhebung der strikten Bauteilzuordnung zu Regionen, wie „Merge and Repartition“ oder die Reflow-Methode für hochentwickelte Bipartitionierungsheuristiken von geringer Wichtigkeit, da nur wenige Bauteile einer ungünstigen Region zugewiesen werden [CKM00].

7.2.2.3 Methode des Zelleneinfügens und die kräftegesteuerte Platzierung

Nach der Platzierung der Netze in der Ebene, werden die Zellen zwischen diesen eingefügt. In der vorliegenden Fassung des Verfahrens erfolgt dies auf dem Massenmittelpunkt der an die jeweilige Zelle angeschlossenen Netze. Dabei wird von punktförmigen Netzrepräsentanten gleicher Masse ausgegangen. Es ist jedoch denkbar, den Netzrepräsentanten unterschiedliche Prioritäten, in dem verwendeten Modell entspricht dies unterschiedlichen Massen, zuzuweisen. Netzrepräsentanten, die eine große Masse zugewiesen bekommen, ziehen die Zellen stärker an, was zu dichter benachbarten Zellen und damit einem kürzeren Netz führt. Durch diese Modellierung ist es möglich, zeitkritische Netze zu bevorzugen.

Das Einsetzen der kräftegesteuerten Platzierung erfolgte in dem hier vorliegenden Prototyp mit einer einfachen Implementierung. So sind nur anziehende Kräfte berücksichtigt worden, in den meisten Modellen für kräftegesteuerte Platzierung werden dagegen zusätzlich abstoßende Kräfte berücksichtigt (vergleiche [She95], [Zob94]). Durch die abstoßenden Kräfte wird das System aus den bewegten punktförmigen Objekten nicht in einem Punkt kollabieren, sondern ein Kräftegleichgewicht erzielen, bei dem sich die Zellen über eine ausgedehnte Fläche verteilen. Die abstoßenden Kräfte werden meist nichtlinear modelliert, üblich ist die Form $F \sim \frac{1}{|d|^p}$. Dabei ist d der Abstand der beiden Zellen in einer geeigneten Metrik und p eine positive, meist ganze, Zahl [Zob94].

Unabhängig von möglichen Modifikationen des Algorithmus zur kräftegesteuerten Platzierung, wäre es wünschenswert, diesen Ansatz eigenständig, also ohne vorherige hierarchische Partitionierung, anwenden zu können. Es wäre denkbar, statt der hierarchischen Partitionierung eine zufällige Startplatzierung zu wählen, und von dieser ausgehend die Iterationen der kräftegesteuerten Methode zu berechnen.

Die mit der hierarchischen Partitionierung erzielten Ergebnisse (siehe Kapitel 6.3) lassen es jedoch fraglich erscheinen, ob die kräftegesteuerte Methode eine Verbesserung der zuvor erstellten Platzierung der Netze erzielen kann. Dies gilt insbesondere, wenn der für die hierarchische Partitionierung verwendete Algorithmus verbessert wird.

7.2.2.4 Standardzellplatzierung erstellen

Mit dem gewählten Sweep-Line Algorithmus ist ein intuitiver Ansatz verfolgt worden, der mit geringem Implementierungsaufwand akzeptable Ergebnisse liefert. Nach den in Kapitel 6.2 gewonnenen Erkenntnissen des starken Einflusses der Transformation der Vorplatzierung in eine Standardzellplatzierung, ist eine Verfeinerung dieses Schrittes

angezeigt. Die beiden im folgenden genannten Transformationen sind mögliche Ansätze einer Verbesserung des bisher verwendeten Verfahrens.

Das in [Téb93] vorgestellte und dort als „Salami-Slicing-Konzept“ bezeichnete Verfahren plaziert die Zellen der Vorplazierung vom Rand der Chipfläche ausgehend an ihren endgültigen Positionen. Bei einem Standardzellayout werden die jeweils oberste und unterste Zellreihe des noch nicht endgültig plazierten Bereiches erzeugt. Dieses Verfahren wird rekursiv angewendet, bis alle Zellen der Vorplazierung in die Standardzellplazierung integriert sind.

Ein alternatives Vorgehen der Transformation in eine Standardzellplazierung läßt sich aus [Vyg98] ableiten, es verwendet einen rasterbasierten Ansatz. Das von der Vorplazierung verwendete Gebiet $[y_{\min} ; y_{\max}] \times [x_{\min} ; x_{\max}]$ wird in ein $n \times m$ Raster eingeteilt. Jedem Feld dieses Rasters wird eine Kapazität von $\frac{1}{n \times m}$ der für Zellen verfügbaren Fläche zugewiesen. Die Zellen werden den Kacheln gemäß ihrer Lage in der Vorplazierung zugeordnet. In einem anschließenden Schritt werden die Zellen zwischen den Feldern derart verschoben, daß die Kapazitäten der einzelnen Felder nicht von der Summe der Fläche der darin befindlichen Zellen überschritten wird. Wird für den Wert von n die Anzahl der gewünschten Zellreihen gewählt, läßt sich der in [Vyg98] vorgeschlagene Ansatz direkt einsetzen.

Die Plazierung der Padzellen erfolgt in der vorliegenden Fassung von `prob` auf einfachste Weise. Auch wenn die von der Lage der Padzellen beeinflusste Zahl an Netzen gering ist, kann nicht ausgeschlossen werden, daß insbesondere die maximale Netzlänge durch ungünstige Padpositionen vergrößert wird. Eine bessere Integration der Padzellen in die Erstellung der Standardzellplazierung ist daher zu überlegen.

Die Zuordnung der Pads zu einer der Chipkanten erfolgt bereits während der hierarchischen Partitionierung im Verlaufe der Netzplazierung. Diese Zuordnung wird im weiteren Verlauf der Plazierung nicht weiter untersucht. Da sich durch die Verschiebungen der Zellpositionen bei der Transformation der Vorplazierung in eine Standardzellplazierung auch die Voraussetzungen der Padplazierung ändern, sollte die Plazierung der Padzellen zu diesem Zeitpunkt erneut bewertet und gegebenenfalls angepaßt werden.

Der hier beschrittene Weg, die Plazierung über eine entwurfstiltunabhängige Vorplazierung zu erstellen, ist, wie Kapitel 6 schildert, nicht ohne Nachteile. Die Vorplazierung wurde hier verwendet, um eine leichte Adaption des Verfahrens an andere Entwurfstile zu ermöglichen.

Die Qualität der Vorplazierung ist nicht direkt bewertbar, da sie nicht durch Verdrahten in ein gültiges Layout überführt werden kann. Die Qualität der entstehenden Plazierung ist immer auch von der Transformation der Vorplazierung in den gewünschten Entwurfstil anhängig. Es bietet sich an, dieses Problem zu umgehen, indem der Entwurfstil früher als in `prob` in die Plazierungsalgorithmen eingearbeitet wird. Wird der Entwurfstil bereits bei der Plazierung der Netzrepräsentanten berücksichtigt, ist es denkbar, auf eine explizite Vorplazierung völlig zu verzichten.

7.2.3 Globalverdrahtung und Plazierungsmodifikationen

Dieser Abschnitt enthält die Anmerkungen zum implementierten Globalverdrahtungsalgorithmus und dessen Möglichkeiten die Plazierung zu modifizieren.

7.2.3.1 Zellweise sequentielle Verdrahtungsreihenfolge

Die bisher verwendete zellweise sequentielle Verdrahtungsmethode kommt ohne explizite Bestimmung einer Reihenfolge aus, in der die Zellen bearbeitet werden. Die Liste der betrachteten Zellen wird, beginnend bei den Padzellen, jeweils um die an eine neu erreichte Zielzelle angeschlossenen Netze erweitert (vergleiche Abschnitt 4.3.1.1).

Analog der netzweise sequentiellen Verdrahtung ist zu erwarten, daß auch die zellweise sequentielle Methode nicht invariant gegenüber der Reihenfolge, in der die Zellen betrachtet werden, ist.

Eine mögliche Erweiterung des Verdrahtungsalgorithmus ist daher eine Komponente, die zunächst eine sortierte Liste der zu verdrahtenden Zellen bestimmt. Für die in dieser Liste vorgenommenen Sortierung lassen sich analog dem netzweise sequentiellen Vorgehen Ordnungskriterien definieren. In [She89] dient der in Gleichung 5 angegebene Quotient als Ordnungskriterium für die Netze, ein analoges Maß ließe sich, wie in Gleichung 6 gezeigt, für die zu verdrahtenden Zellen definieren.

$$\text{Netzpriorität} = \frac{\text{Fläche des die Pins umspannenden Rechtecks}}{\text{Anzahl der verbundenen Pins}}$$

Gleichung 5: Der Quotient aus dem Flächeninhalt des die Pins eines Netzes umspannenden Rechtecks und der Anzahl der von diesem Netz verbundenen Pins ergibt die Priorität des Netzes. Die Verdrahtung erfolgt gemäß der so ermittelten Priorität.

$$\text{Zellpriorität} = \frac{\text{Fläche des die Netzmittelpunkte umschließenden Rechtecks}}{\text{Anzahl der Pins der Zelle}}$$

Gleichung 6: Der Quotient aus dem Flächeninhalt des die Mittelpunkte der an eine Zelle angeschlossenen Netze umspannenden Rechtecks und der Anzahl der Pins dieser Zelle geben die Priorität einer Zelle bei der zellweise sequentiellen Verdrahtung an.

Die für netzweise sequentiell vorgehende Verdrahtungsalgorithmen häufig angewendete Methode, bereits verdrahtete, aber nachfolgende Netze blockierende Teilnetze wieder zu entfernen und später erneut zu verdrahten (Rip-Up and Reroute) ([Het98], [RaB91]), läßt sich auf das hier verwendete zellweise sequentielle Verfahren übertragen.

Die beiden in diesem Abschnitt geschilderten möglichen Erweiterungen sind nur eine kleine Auswahl. Es ist prinzipiell denkbar, alle für netzweise sequentielle Algorithmen gewonnen Erkenntnisse und entwickelten Erweiterungen unter dem Blickwinkel der Eignung für die zellweise sequentielle Verdrahtung neu zu beurteilen.

7.2.3.2 Zweipunktnetzbetrachtung

Die algorithmische Bearbeitung der Zweipunktnetzverdrahtung erfolgt in einem Routinggraph. Dieser Baum hat eine relativ geringe Tiefe bei einem großen Verzweigungsfaktor. Für Bäume geringer Tiefe, wie sie für die bisher betrachteten vergleichsweise kleinen Layouts betrachtet werden, ist der Rechenzeitaufwand für eine vollständige Suche im Baum noch akzeptabel. Für größere Layouts ist zu erwarten, daß aufgrund des sehr großen Verzweigungsfaktors, schon durch die Hinzunahme weniger Ebenen im Routinggraph, die Suchzeit unakzeptabel groß wird. Daher wäre eine sinnvolle Erweiterung, die bisher vollständige Traversierung des Baumes durch eine Suchheuristik zu ersetzen. In [Téb93] wird für einen kombinatorischen Platzierungsalgorithmus ein Cut-and-Deng Algorithmus verwendet, der dort als Breitenspursuche (BBS) bezeichnet wird. Dieses Verfahren begrenzt den Verzweigungsgrad des betrachteten Teilbaumes im Routinggraph, so daß der Zeitbedarf deutlich reduziert werden kann.

In der aktuellen Fassung des Globalverdrahters geschieht die Reservierung von Tracks in einem Kanal anonym. Obwohl zum Zeitpunkt der Bedarfsbestimmung bekannt ist, für welches Netz ein Track reserviert werden muß, wird diese Information nicht weiter betrachtet. So geschieht es in der momentanen Implementation, daß in einem Kanal für ein Netz in einigen Intervallen zwei oder mehr Tracks reserviert werden. Dies führt zu einer vermeidbar ungenauen Kanalbreitenschätzung. Eine wünschenswerte Erweiterung des Globalverdrahters wäre daher, den Bedarf an Tracks in einem Kanal mit den Identifikationsnummern der Netze zu verknüpfen, so daß doppelte Trackbelegungen durch ein Netz entfallen.

7.2.3.3 Platzierungsmodifikation durch den Verdrahter

Die Ergebnisse des prob-Verdrahters zeigen die Sensibilität der lokal vorgenommenen Veränderungen der Platzierung durch den Verdrahter. Um mit Verschiebungen von Zellen in einer existierenden Platzierung (Circuit Migration) Verbesserungen des gesamten Layouts zu erzielen, sind wesentlich detailliertere Betrachtungen nötig, als sie in dem in dieser Arbeit präsentierten Prototyp zu Grunde gelegt wurden. In [DKS98] werden komplexe Betrachtungen angestellt, auf deren Grundlage mögliche Zellverschiebungen betrachtet werden. Dieses Gebiet bietet, ähnlich wie die Verfeinerung der verwendeten Bipartitionsheuristiken, ein weites Betätigungsfeld für zukünftige Erweiterungen des in dieser Arbeit präsentierten Ansatzes.

Die in der aktuellen Implementierung verwendete Bewertungsfunktion für Platzierungsmodifikationen betrachtet die Veränderung der Kanalauslastungen durch das Verschieben von Zellen nicht. Diese hat jedoch wesentlichen Einfluß auf die Chipgröße und sollte berücksichtigt werden.

Die bisher zugelassenen Veränderungen der Platzierung durch den Verdrahter sind auf wenige Aktionen beschränkt. Eine Erweiterung des Repertoires an zugelassenen Aktionen kann die Bedeutung dieser Komponente des Globalverdrahters deutlich erhöhen. Bisher sind Verschiebungen von Zellen nur auf die aktuelle Zellreihe beschränkt. Eine naheliegende Erweiterung ist das Zulassen der Verschiebung oder Vertauschung einer Zelle auf eine Position in einer benachbarten Zellreihe. In Abbildung 59 ist ein Beispiel

einer solchen Vertauschung gezeigt, durch die der Bedarf an Feedthroughs reduziert werden kann.

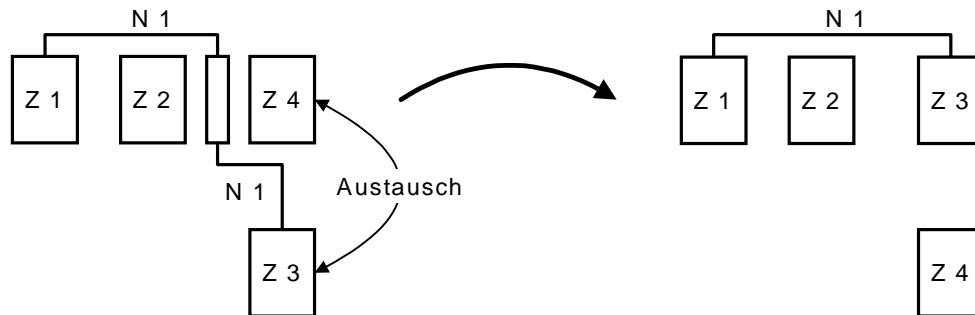


Abbildung 59: Durch Vertauschen der Zellen Z3 und Z4 lässt sich eine Feedthroughzelle für Netz N1 einsparen

7.2.4 Kompaktifizierung und Pad-Behandlung

In der gegenwärtigen Implementierung des vorgestellten Verfahrens wird eine einfache Kompaktifizierung vorgenommen. Dabei werden alle unbenutzten Tracks eines Kanals entfernt und die Kanalbreite so auf die erforderliche Breite reduziert. Die Zellreihen werden entsprechend näher beieinander platziert. Momentan wird das gesamte Layout auf diese Weise dicht an der oberen Kante zusammengeschoben. In einem pad-dominierten Layout führt dies zu einem breiten untersten Kanal und damit möglicherweise zu einer ungünstigen Verteilung der Netzlängen. In pad-dominierten Layouts sollten die nicht benötigten Tracks gleichmäßig auf die Kanäle verteilt werden oder eine möglichst kompakte Kernfläche erzeugt werden, die oben und unten von zwei gleich breiten Kanälen vom Padkranz getrennt wird. Die momentan vorgenommene ungleichmäßige Verteilung der ungenutzten Tracks ist jedoch nicht wünschenswert.

Das Verdrahten der Pads in den vertikalen Kanälen links und rechts der Kernfläche erfolgt momentan vor der Kompaktifizierung. Dies lässt unnötige Umwege in der Kanalverdrahtung entstehen, nachdem die Kompaktifizierung erfolgt ist. Dies ließe sich auf einfache Weise vermeiden, indem die vertikalen Kanäle verdrahtet werden, nachdem die Kompaktifizierung der horizontalen Kanäle erfolgt ist.

Des weiteren werden die Padzellen positioniert, bevor der Chip in vertikaler Richtung verkleinert wurde. Es kann daher geschehen, daß die vertikale Ausdehnung des Chips durch die unterste Padzelle bestimmt wird, obwohl diese ebenso in einem weiter oben gelegenen Slot der Chipkante untergebracht werden könnte. In Abbildung 58 ist dies zu erkennen, würde die Kompaktifizierung die Chipgröße in vertikaler Richtung reduzieren, würden die Padzellen der rechten Kante die vertikale Chipausdehnung bestimmen und nicht der Platzbedarf der Kernfläche. Dies ließe sich durch eine Erweiterung der Kompaktifizierung, die auch die Padpositionen modifiziert, vermeiden.

7.2.5 Kanalverdrahtung

Die Kanalverdrahtung ist in dieser Arbeit vollkommen unbetrachtet geblieben. Die Wahl des Greedy Kanalverdrahters erfolgte, da es sich hierbei um einen erprobten, robusten Algorithmus handelt. Die Wahl eines anderen Kanalverdrahters ist problemlos möglich. Der modulare Aufbau von ESPRO legt es nahe, die Kanalverdrahtung in ein eigenes Modul auszugliedern und die Globalverdrahtungsinformationen über dann zu definierende Schnittstellen zu übergeben.

Unabhängig von der gewählten Art der Umsetzung ließen sich die Kanalbreiten reduzieren, indem direkt vor der Kanalverdrahtung die Positionen der Zellen in den Reihen überprüft werden und durch Verschiebungen, wie eine in Abbildung 60 gezeigt ist, der Trackbedarf reduziert wird [Die89].

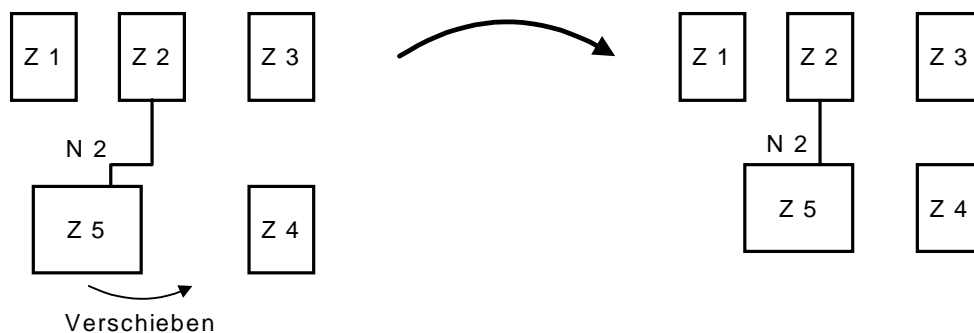


Abbildung 60: Durch das Verschieben von Zellen innerhalb der Zellreihen läßt sich der Bedarf an Tracks innerhalb des Kanals reduzieren, da zwei einander direkt gegenüberliegende Pins zur Verdrahtung keinen Track entlang der Kanalrichtung benötigen [Die89]

7.3 Änderung des Modells

Die bisher präsentierten Ansatzpunkte für weiterführende Tätigkeiten bewegen sich alle innerhalb der in Kapitel 3 dargelegten Grenzen der Evaluierungsumgebung ESPRO. Eine erhebliche Erweiterung der Möglichkeiten für weiterführenden Arbeiten ergibt sich durch die Erweiterung des zugrundeliegenden Modells. Dabei kann entweder ESPRO erweitert werden oder eine völlig andere Umgebung als Grundlage weiterer Untersuchungen dienen.

7.3.1 Erweiterung von ESPRO

Die Grenzen des in ESPRO Modellierbaren werden durch keines der bisher existierenden Module voll ausgeschöpft. Viele der auftretenden Beschränkungen des Modells entstehen durch den Wunsch, bisher gewonnene Ergebnisse weiterhin zu Vergleichszwecken heranziehen zu können. Durch Erweiterung des Modells und Ausnutzen bisher nicht voll ausgeschöpfter Modellaspekte, ist es möglich, ohne den Verlust der in ESPRO geforderten Unabhängigkeit von speziellen Zellbibliotheken eines Herstellers oder eines Prozesses, in der betrachteten Zielfunktion weitere Layouteigenschaften zu berücksichtigen.

Die feststehenden Grenzen des ESPRO-Modelles beschränken sich auf die vollständige Abstraktion von der Funktion der Zellen, den Entwurstil des Standardzellentwurfes und die ausschließliche Betrachtung der Logikverdrahtung.

7.3.1.1 Verkleinerung der Zellen

Die für das in dieser Arbeit präsentierte Verfahren zugrunde gelegte Prämisse, der im Vergleich zu den benötigten Verdrahtungsressourcen kleinen aktiven Elementen, wird in der vorliegenden Form von ESPRO kaum erfüllt (vergleiche Kapitel 3.3). Die Definition der in ESPRO verwendeten Zellen geschieht jedoch auf einfache Weise und kann leicht geändert werden. Das vorliegende Programm `prob` ist hierfür bereits vorbereitet (vergleiche Anhang A).

Eine Evaluation des Verfahrens unter Verwendung kleiner Standardzellen ist so leicht möglich. Die Gewinnung von Vergleichsdaten ist dagegen weniger einfach. Die existierenden ESPRO-Module müßten angepaßt werden, was gegebenenfalls einen erheblichen Implementierungsaufwand bedeuten kann.

7.3.1.2 Feedthroughs als Zellen

Die bisher in ESPRO gewählte Repräsentation der Feedthroughs als vertikale Leitungssegmente, die die Zellreihe in einer Lücke durchqueren, kann erweitert werden. Durch Einführen dedizierter Feedthroughzellen in der Bibliothek von ESPRO, lassen sich die Feedthroughs als Leitungstreiber interpretieren (Buffer Insertion) und die entstehenden Teilnetze beiderseits des Feedthroughs getrennt betrachten. Es entstehen somit durch jedes eingeführte Feedthrough zwei neue Netze, die unabhängig voneinander zu verdrahten und zu bewerten sind.

Der hier implementierte Prototyp `prob` ist bereits auf solche Erweiterungen vorbereitet und erlaubt die getrennte Definition der Breiten von normalen Zellen und Feedthroughs. Die übrigen bereits bestehenden ESPRO-Module sehen diese Erweiterung jedoch nicht vor, so daß diese Erweiterung des Modells einen erheblichen Einschnitt darstellt und, ähnlich wie bei der Zellverkleinerung, eine Überarbeitung der existierenden Implementierung erfordert.

7.3.1.3 Zusätzliche Erweiterungsmöglichkeiten für ESPRO

Durch einfache Erweiterung des Modells um die Signalrichtung der Pins der Zellen, wird eine Betrachtung einfacher Verzögerungsmodelle und die Berücksichtigung der von einer Zelle getriebenen Last möglich. Das Modellieren äquivalenter Zellen mit unterschiedlichen Ausgangsleitungen ist ebenfalls mit vertretbarem Aufwand möglich.

Die Kenntnis der zu treibenden Last ermöglicht es die Auswahl einer Zelle mit geeigneter Treiberleistung vorzusehen, wie es heute in Platzierungsalgorithmen üblicherweise geschieht. Für diese Auswahl von Bibliothekselementen mit passender Treiberleistung gibt es zahlreiche Algorithmen unterschiedlicher Komplexität [DKS98]. Außerdem lassen sich einfache Verzögerungsmodelle anwenden und so Algorithmen untersuchen, die als eines der Zielfunktionskriterien die Verzögerungsminimierung haben. Das Einfügen von Feedthroughs ließe sich, insbesondere mit der im vorigen Abschnitt vorgeschlagenen Modellerweiterung, somit auch unter den Kriterien der zu treibenden Last und der Signallaufzeit betrachten (Delay Driven Buffer Insertion).

Weitergehende Änderungen könnten in der Möglichkeit des Spiegels von Zellen sowie der Modellierung von Leitungsbreiten für exaktere Verzögerungsmodelle bestehen.

Bei allen Erweiterungen des ESPRO zugrundeliegenden Modells für Standardzellentwürfe ist jedoch zu beachten, daß jede Verfeinerung der Modellierung die Gefahr birgt, einen Teil der angestrebten Allgemeingültigkeit zu verlieren. Eine der Stärken des existierenden ESPRO-Modells ist die über große Zeiträume konstante Modellierung des Standardzellentwurfes. Durch zu hohe Detailtreue kann die Lebensdauer des gewählten Modells deutlich herabgesetzt werden.

Eine im derzeitigen Modell von ESPRO bereits vorgesehene, jedoch in keinem bisherigen Modul verwendete Option, ist die Verwendung weiterer Verdrahtungsebenen. Die für die in dieser Arbeit implementierten Module verwendeten Algorithmen, die die Anzahl der verwendeten Feedthroughs minimiert. Diese Zielsetzung ist jedoch nur solange sinnvoll, wie die Feedthroughs Platz in den Zellreihen belegen. Ist das Durchqueren einer Zellreihe möglich, ohne dabei Platz für aktive Zellen zu belegen, besteht keine Notwendigkeit der Minimierung von Reihendurchquerungen. Das Hinzunehmen von weiteren Verdrahtungsebenen erweitert somit die Problemstellung. Es lassen sich nun verschiedene Reihentransits realisieren. Einerseits können Feedthroughzellen als Leitungstreiber in die Zellreihen eingefügt werden und andererseits ist eine Überquerung der Zellreihe in einer weiteren Verdrahtungsebene möglich, ohne dabei Platz für aktive Zellen in der Reihe zu benötigen. Eine Erweiterung der Möglichkeiten der Verdrahtung hat somit auch Auswirkungen auf die Qualität der Platzierung, da das Wechseln des Verdrahtungskanal nicht mehr zwingend Platz benötigt, den der Platzierungsalgorithmus zuvor reservieren muß.

Der Übergang von kanalbasierten Verdrahtungen zu Algorithmen, die OTC-Routing verwenden ist für die netzweise sequentielle Verdrahtungsreihenfolge ausführlich untersucht worden [She95]. Die Auswirkungen von OTC-Routing auf die in dieser Arbeit vorgeschlagene zellweise sequentielle Verdrahtungsreihenfolge, sind noch zu ermitteln. Diese Konsequenzen ließen sich in dem existierenden Rahmenwerk von ESPRO gut evaluieren.

7.3.2 Modellierung außerhalb von ESPRO

Als die sicher weitestreichende Änderung der Modellierung kann die Evaluierungsumgebung von ESPRO verlassen werden. Es wäre denkbar das in dieser Arbeit konzipierte Verfahren für andere Werkzeuge und Evaluierungsumgebungen zu implementieren. Dies bietet die Möglichkeit, ein Modell auszuwählen, in dem die in Kapitel 2.5 dargelegten Annahmen über die Eigenschaften zukünftiger Entwurststile besser erfüllt sind. Darüber hinaus kann die Qualität der Ergebnisse anhand anerkannter Testprobleme mit existierenden Implementierungen weltweit verglichen werden. Diese Vorteile der Vergleichbarkeit der Ergebnisse mit Resultaten von Forschungsaktivitäten weltweit, wird jedoch mit der Aufgabe der sehr abstrakten Modellierung in ESPRO erkaufte.

Als ein mögliches Rahmenwerk sei hier das „VLSI CAD Bookshelf“ der Universität von Los Angeles genannt, welches sich als elektronisches Medium zum Informationsaustausch für die Implementierung und Evaluierung von qualitativ hochwertigen

Optimierungsalgorithmen für den VLSI-Entwurf und angrenzende Themengebiete versteht [CKM00a].

7.4 Projektion des Konzeptes in die Zukunft

Die in 6.2 und 6.3 präsentierten Ergebnisse zeigen, daß das hier vorgestellte Verfahren bereits für die heute auftretenden Produktionsprozesse geeignet ist. Es ist zu erwarten, daß die sich für große Netzlisten in ESPRO abzeichnenden guten Resultate noch verbessern lassen, sofern die in Kapitel 2.5 vorausgesetzten Eigenschaften der Layouts besser erfüllt werden. Bereits mit den heute bewältigbaren Prozessen ([Int00], [Int01]) sind deutlich bessere Resultate zu erwarten, als unter den in ESPRO vorherrschenden Bedingungen. Die fortschreitende Strukturverkleinerung im VLSI-Entwurf ([Law00], [Het98]) gibt Grund zu der Annahme, daß das in dieser Arbeit vorgestellte Konzept der dualen Problembetrachtung durch Weiterentwicklung und Verfeinerung der verwendeten Modelle konkurrenzfähige Algorithmen zur Lösung der Aufgaben des physikalischen VLSI-Entwurfes hervorbringen kann.

Das hier präsentierte Verfahren basiert auf dem Hinterfragen der Abgrenzung der beiden Teilschritte des Plazierens und Verdrahtens des physikalischen VLSI-Entwurfsprozesses. Die erzielten Ergebnisse untermauern die Annahme, daß eine weniger strikte Abgrenzung der Teilprobleme voneinander bessere Resultate hervorbringen kann. In [DKS98] wird eine Grenze zwischen anderen Teilproblemen hinterfragt. Dort wird der Bereich des physikalischen Entwurfes verlassen und ein Ansatz, der Schaltkreissynthese und Plazieren verbindet, vorgestellt. Weitere Untersuchungen zu bisher etablierten Problemabgrenzungen und das Hinterfragen dieser bietet Raum für weitere Aktivitäten, um den steigenden Anforderungen an den automatisierten Schaltungsentwurf gerecht zu werden.

8. ZUSAMMENFASSUNG

Die beiden Aufgaben des Plazierens und Verdrahtens bilden die zentrale Problemstellung des physikalischen VLSI-Entwurfs. Die sich durch die technologische Entwicklung der Fertigungsprozesse wandelnden Anforderungen an die verwendeten Algorithmen fordern eine ständige Weiterentwicklung der Entwurfswerkzeuge. Die Mehrzahl der Verbesserungen konzentriert sich auf die zu lösenden Teilprobleme und auf die speziellen Randbedingungen bestimmter Entwurststile und Herstellungsprozesse. So werden in anderen Arbeiten z.B. Methoden untersucht, um erprobte Heuristiken an die wachsende Komplexität der Layouts anzupassen. Nur wenige Ansätze betrachten das Vorgehen des physikalischen VLSI-Entwurfs als Ganzes. Die übliche Aufteilung der Problematik, eine Netzliste in eine geometrische Beschreibung für den Herstellungsprozeß zu überführen, wird selten hinterfragt.

Durch die beständige Verkleinerung der Strukturgrößen verschärft sich insbesondere das Verdrahtungsproblem. So wird die zulässige Netzlänge immer kürzer, die Wechselwirkungen zwischen benachbarten Leitungssegmenten nehmen zu und die Anzahl der zu berücksichtigenden Netze steigt. Der wachsenden Bedeutung des Verdrahtungsproblems ist in dieser Arbeit durch eine neue Vorgehensweise für das Plazieren und Verdrahten begegnet worden:

Ausgehend von dem Bestreben, das Verdrahten losgelöst von einer vorgegebenen, und eventuell ungünstigen, Plazierung betrachten zu können, wird ein Verfahren verwendet, das die Netze ohne vorheriges Plazieren der Zellen auf der Chipfläche anordnet. Aus dieser „Netzplazierung“ wird die eigentliche Plazierung der Zellen abgeleitet.

Als Modellierungsmittel der Netzliste, und damit des Plazierungs- und Verdrahtungsproblems, wurden Hypergraphen gewählt. Die Verwendung von zwei zueinander dualen Hypergraphen bietet die Möglichkeit, die Zellen und Netze wechselnd als Knoten oder Kanten eines Hypergraphen zu behandeln.

Die Ausdehnung der dualen Betrachtungsweise auf das Verdrahtungsproblem ist nur eingeschränkt erfolgt. In dem hier vorgestellten Verfahren wird die Verdrahtung nicht wie üblich netzweise sequentiell, sondern zellweise sequentiell durchgeführt.

Das in dieser Arbeit präsentierte Verfahren hält an der Einteilung der Layoutgenerierung in Plazieren und Verdrahten und an dieser Bearbeitungsreihenfolge fest. Die Rolle von Netzen und Zellen wurde jedoch, soweit umsetzbar, vertauscht.

Es wurde ein Prototyp des Verfahrens implementiert und unter Verwendung der offenen Evaluierungsumgebung ESPRO (siehe Kapitel 3) mit bestehenden Plazierungs- und Verdrahtungsalgorithmen verglichen. Diese, als eine erste Machbarkeitsstudie ausgelegte, Implementierung verwendet einfache Algorithmen, um die zu bewältigenden Teilprobleme zu lösen.

Die erzeugten Layouts zeigen, daß das Konzept, die Plazierung aus der Verteilung der Netze über die Chipfläche abzuleiten, gute Ergebnisse liefert. Insbesondere für große Netzlisten erzeugt der implementierte Plazierungsalgorithmus Resultate, die zu kompakten Layouts führen.

Die Evaluierung des Verdrahtungsalgorithmus zeigt eine deutliche Verkürzung der benötigten Netzlängen auf etwa 50%, jedoch auf Kosten der benötigten Chipfläche, die

sich um Werte in der Größenordnung von 15-20%, relativ zu Layouts anderer Verdrahter, vergrößert.

Die Qualität der mit dem Prototypen erstellten Plazierungen und Verdrahtungen zeigt, daß das in dieser Arbeit konzeptionierte Vorgehen, die Netze stärker in den Mittelpunkt der Betrachtung für den physikalischen VLSI-Entwurfs, insbesondere des Plazierens, zu rücken, konkurrenzfähig ist. Bei der andauernden Verkleinerung der Basisstrukturen der Herstellungsprozesse ist zu erwarten, daß die Stärken dieser Methode zukünftig noch mehr hervortreten werden.

A. VERWENDUNG DES PROGRAMMES PROB

A.1 Bedienung des Programmes

A.1.1 Allgemeiner Programmablauf

Das in dieser Arbeit entstandene Programm ist eine Kommandozeilenapplikation, die Eingaben durch Benutzerinteraktionen in textbasierten Eingabemasken und durch Kommandozeilenparameter erhält. Dieser Abschnitt erläutert den Programmablauf und die möglichen Kommandozeilenargumente.

Der Programmablauf läßt sich durch Schalter, die als Kommandozeilenargumente übergeben werden, beeinflussen. Es existieren zwei Hauptbetriebsarten, der interaktive Modus und der Batchbetrieb. Der interaktive Modus ist die Vorgabeeinstellung. Der Batchbetrieb wird durch einen Kommandozeilenschalter aktiviert und in Abschnitt A.5 erläutert. Im interaktiven Betrieb werden alle nicht durch Eingaben (auf der Kommandozeile oder aus Dateien) gegebenen Werte vor der eigentlichen Algorithmusbearbeitung vom Benutzer erfragt. Der Hauptunterschied im Batchbetrieb besteht darin, daß alle nicht angegebenen Parameter mit den Defaultwerten belegt werden. Soweit durch Schalter nicht anders gefordert, wird ausgehend von der Basisdatei sowohl die Plazierung als auch das Verdrahten durchgeführt.

Bei jedem Programmlauf werden außer den durch das eigentliche physikalische Design benötigten Dateien zwei weitere erstellt. Dies sind eine Datei mit den aktuell verwendeten Parameterwerten (Parameterdatei) und eine mit den Namen der involvierten Dateien (Logdatei). Die beiden Dateinamen setzen sich aus dem Namen der Basisdatei (ohne Pfadangaben) ohne Endung und dem aktuellen Datum mit Uhrzeit zusammen. Die Parameterdatei hat die Endung `.set` (von „setting“), die Logdatei die Endung `.flg` (von „file log“).

Die Datei mit den verwendeten Parameterwerten wird immer erzeugt, mit zwei Ausnahmen. Erstens im Falle, daß die als Eingabe verwendete Parameterdatei vollständig ist und keine Werte vom Benutzer verändert werden, also der Programmlauf wirklich die angegebenen Werte aus der Datei verwendet. Zweitens, wenn die spezifizierte Parameterdatei nicht existiert, dann dient der angegebene Dateiname als Name der Ausgabedatei für die bei diesem Programmlauf verwendeten Parameter.

Die Logdatei für die Namen der verwendeten Dateien wird nur im Batchmodus nicht erzeugt, wenn die verwendete Eingabedatei alle Dateinamen spezifizierte. Sie kann verwendet werden, um einen Programmlauf später reproduzieren zu können. Die für den Batchbetrieb verwendeten Dateien haben das gleiche Format wie die automatisch generierten Logdateien, so daß diese später direkt als Eingabe verwendet werden können.

A.1.2 Programmaufruf

Das Programm wird von der Kommandozeile aus gestartet. Die Übergabe von Parametern beim Aufruf ist optional. Die möglichen Aufrufmethoden sind im folgenden aufgelistet.

Anhang A Verwendung des Programmes prob

```
prob [-P] [-R] [-B|-I] [-p <placement>] [-r <routedplacement>]  
      [-l <layout>] [-s <parameterfile>] [-b <batchfile>]  
      [-c <basefile>|<basefile>]
```

Abbildung 61: Kommandozeilenparameter des Programmes prob

Die Angabe der Parameter kann in beliebiger Reihenfolge erfolgen. Einzige Ausnahme bildet die Basisdatei, sofern sie ohne den Schalter „-c“ angegeben wird, dann muß es der letzte Parameter sein.

A.1.3 Erläuterung der Kommandozeilenparameter

-c <basefile> oder <basefile>

Dieser Schalter spezifiziert den Namen der Basisdatei (Standardendung .chp). Wird er mit „-c“ angegeben, kann der Parameter an beliebiger Position in der Kommandozeile auftreten. Wird der Dateiname allein genannt, muß es der letzte Parameter der Kommandozeile sein. Wird ein Dateiname ohne Extension angegeben, wird die Endung .chp angehängt.

-p <placement>

Dieser Schalter spezifiziert den Namen der Plazierungsdatei (Standardendung .pla). Die Placementdatei kann Eingabedatei sein, wenn nur der Verdrahter verwendet wird (Schalter -R) oder Ausgabedatei, wenn der Plazierer von prob aktiviert ist. Bei aktiviertem Plazierer wird die Datei mit dem gegebenen Namen <placement> erzeugt, eine eventuell bereits existierende Datei mit diesem Namen wird ohne Rückfrage überschrieben. Wird ein Dateiname ohne Extension angegeben, wird die Endung .pla angehängt.

-r <routedplacement>

Dieser Schalter spezifiziert den Namen der Datei für die vom Verdrahter veränderte Plazierung (Standardendung .plr). Dies ist eine reine Ausgabedatei. Existiert bereits eine Datei mit dem angegebenen Namen, wird diese überschrieben. Wird ein Dateiname ohne Extension angegeben, wird die Endung .plr angehängt.

-l <layout>

Dieser Schalter spezifiziert den Namen der Datei für das fertige Layout (die Layoutdatei, Standardendung .lay). Dies ist eine reine Ausgabedatei. Existiert bereits eine Datei mit dem angegebenen Namen, wird diese überschrieben. Wird ein Dateiname ohne Extension angegeben, wird die Endung .lay angehängt.

-s <parameterfile>

Dieser Schalter spezifiziert den Namen der Parameterdatei (Standardendung .set). Die Parameterdatei kann Eingabe- oder Ausgabedatei sein. Existiert eine Datei mit dem angegebenen Namen, so wird sie eingelesen und die enthaltenen Parameterwerte verwendet. Existiert keine Datei dieses Namens, so werden die bei diesem Programmablauf verwendeten Parameterwerte in diese Datei geschrieben und die ansonsten erzeugte Ausgabedatei für Parameterwerte wird nicht erzeugt. Wird ein Dateiname ohne Extension angegeben, wird die Endung .set angehängt.

-b <batchfile>

Dieser Schalter spezifiziert den Namen der Datei für den Batchbetrieb und aktiviert den Batchbetrieb. Diese Batchdatei enthält die Namen der oben aufgeführten Ein- und Ausgabedateien. Wird ein Dateiname ohne Extension angegeben, wird die Endung `.flg` angehängt. Näheres zum Batchbetrieb ist dem Abschnitt A.5 zu entnehmen.

-P

Dieser Schalter läßt das Programm als stand-alone Plazierer arbeiten, der Verdrahter wird deaktiviert. Alle Dateiangaben und Parametereingaben, die nur den Verdrahter betreffen werden ignoriert. Wird auch der Parameter `-R` angegeben, wird weder plaziert noch verdrahtet.

-R

Dieser Schalter läßt das Programm als stand-alone Verdrahter arbeiten, der Plazierer wird deaktiviert. Alle Dateiangaben und Parametereingaben, die nur den Plazierer betreffen, werden ignoriert. Wird auch der Parameter `-P` angegeben, wird weder plaziert noch verdrahtet.

-B

Dieser Schalter aktiviert den Batchmodus, ohne die Notwendigkeit eine Eingabedatei zu spezifizieren. Es werden keine Benutzerabfragen durchgeführt. Alle nicht angegebenen Dateinamen und Parameterwerte werden mit den Defaultwerten belegt. Für Dateinamen ist der Default der Name der Basisdatei ohne Dateiendung und Pfadangaben. Wird auch der Basisdateiname nicht angegeben, so wird als Basisdatei `test.chp` im aktuellen Verzeichnis angenommen.

-I

Dieser Schalter deaktiviert den Batchmodus und forciert Benutzerabfragen. Wird eine Batchdatei angegeben, die die weiteren Dateinamen spezifiziert (Schalter `-b <batchfile>`), so wird der dadurch automatisch aktivierte Batchbetrieb wieder deaktiviert und der Benutzer nach allen nicht in der Parameterdatei angegebenen Parameterwerten gefragt. Analog werden die Dateinamen von nicht spezifizierten Ein- und Ausgabedateien erfragt.

A.2 Algorithmusparameter

Die verwendeten Algorithmen des Programmes `prob` lassen sich in ihrem Verhalten durch die Vorgabe von Parametern beeinflussen. Die Wahl der Parameter kann auf zwei verschiedene Arten erfolgen. Entweder interaktiv, das Programm fragt den Benutzer in diesem Falle nach allen benötigten Werten (siehe A.2.2.1), oder durch eine Parameterdatei (siehe A.2.2.2).

A.2.1 Die Parameter der Teilalgorithmen

Die Parameter sind in Gruppen eingeordnet. Jede Gruppe enthält die Parameter für den Algorithmus zur Lösung eines Teilproblems. Diese Gruppierung ist sowohl in der Parameterdatei als auch in der interaktiven Abfrage der Parameter zu finden. Die Parameter werden im folgenden erläutert.

A.2.1.1 Eigenschaften der Zellbibliothek

Die Parameter dieser Gruppe bestimmen die Charakteristik der verwendeten Zellbibliothek. Die Werte dürfen nicht verändert werden, wenn Ergebnisse bereits bestehender ESPRO Module weiterverarbeitet werden sollen.

Diese Parameter werden im interaktiven Modus nicht abgefragt und können nur über die Parameterdatei geändert werden.

PortRadius

Gibt den Radius einer Padzelle in Rastereinheiten an. Default ist PortRadius=14 .

CellHeight

Gibt die Höhe einer Standardzelle in Rastereinheiten an. Default ist CellHeight=13 .

CellWidthBias und CellWidthPin

In ESPRO sind die Zellbreiten direkt abhängig von der Anzahl der Pins. Diese beiden Parameter bestimmen die Breite der Zellen und die Lage der Pins entlang der oberen und unteren Kante. Die Breite einer Zelle mit PinCount Anschlüssen errechnet sich als $Width = CellWidthBias + CellWidthPin * (PinCount / 2)$.

Default ist CellWidthBias=1 und CellWidthPin=2 .

FTWidthBias und FTWidthPin

Analog den Parametern CellWidthBias und CellWidthPin geben diese Parameter die Breite der Feedthroughzellen an. Feedthroughs haben normalerweise genau zwei Pins, so daß eine Berechnung eigentlich nicht nötig ist, aus Analogie existieren diese Parameter jedoch. Default ist FTWidthBias=0 und FTWidthPin=1 .

FTWidth

Dieser Parameter gibt die Gesamtbreite einer Feedthroughzelle mit genau zwei Pins an. Der Wert muß mit denen für FTWidthBias, FTWidthPin zusammenpassen. Default ist FTWidth=1.

A.2.1.2 Hierarchische Netzplazierung

Die Parameter dieser Gruppe bestimmen die Bewertung der Partitionen während der Plazierung mittels hierarchischer Partitionierung, sowie die Abbruchbedingung.

TotalNetBalanceWeight, CutSizeWeight und ExtCutSizeWeight

Diese Parameter bestimmen die Bewertungsfunktion, die die Qualität einer Bipartition angibt. Die Bewertungsfunktion einer Bipartition der Menge M von Knoten des Graphen Y in die disjunkten Teilmengen A und B ist gegeben als

$$\begin{aligned} \text{Rating}(Y, M, A, B) = & \text{CutSizeWeight} * \# \text{geschnittene interne Verbindungen} \\ & + \text{ExtCutSizeWeight} * \# \text{geschnittene externe Verbindungen} \\ & + \text{TotalNetBalanceWeight} * |(|A| - |B|)|. \end{aligned}$$

Default ist TotalNetBalanceWeight=30, CutSizeWeight=20 und ExtCutSizeWeight=75 .

MaxLeafSize

Dieser Parameter gibt die Anzahl der Netze an, die sich in der kleinsten, durch eine Teilung erzeugten Kachel befinden dürfen. Befinden sich in einer Kachel mehr Netze, als

MaxLeafSize angibt, so wird mit der Partitionierung fortgefahren und diese Kachel wiederholt weiter halbiert. Default ist MaxLeafSize=1 .

A.2.1.3 Kräftebasierte Platzierung der Zellen

Die Kräftebasierte Platzierung wird wiederholt durchgeführt. Dabei bestimmen die Parameter dieser Gruppe, wie die Abbruchbedingungen aussehen. In der Startsituation sind die Zellen über die gesamte Ebene verteilt. Der Wertebereich der Koordinaten ist der mit 32 Bit vorzeichenlos darstellbare Bereich. Durch das Streben des Systems zum Minimum der inneren Energie, konzentrieren sich die Zellen in der Mitte der Ebene. Die Padzellen werden dabei jedoch am Rande der Ebene festgehalten. Es ergibt sich ein nahezu konstanter Skalierungsfaktor für den Durchmesser der von Zellen bedeckten Fläche von einem Iterationsschritt zum nächsten. Die Distanz, um die sich die Zellen während der Iterationen bewegen, muß innerhalb vorgegebener Grenzen bleiben. Das Intervall zulässiger Distanzen ist [MoveDistLimit, MoveDistMax]. Die untere Grenze ist fest, die obere gegeben durch:

$\text{MoveDistMax} = \text{MoveDistMaxBias} - \text{iterationNumber} * \text{MoveDistMaxGradient}.$

Diese Bedingungen gelten für maximale und durchschnittliche Distanz, um die sich die Zellen bei der Iteration bewegen.

MaxCentreIterations

Gibt die absolute Obergrenze an durchzuführenden Iterationen an. Nach MaxCentreIterations wird das Verfahren, unabhängig von allen anderen Randbedingungen, beendet. MaxCentreIterations darf nicht kleiner sein als MinCentreIterations. Default ist MaxCentreIterations=0.

MinCentreIterations

Gibt die absolute Untergrenze an durchzuführenden Iterationen an. Keine andere Abbruchbedingung wird überprüft, solange nicht mindestens MinCentreIterations durchgeführt wurden. Default ist MinCentreIterations=0.

MaxMoveDistLimit

Untere Schranke der Distanz, um die sich die am weitesten bewegte Zelle während der Iteration bewegt haben darf. Default ist MaxMoveDistLimit=0 .

MaxMoveDistMaxBias

Gibt den y-Achsenabschnitt der Geraden an, die die obere Schranke der Distanz definiert, um die sich die am weitesten bewegte Zelle während der Iteration bewegt haben darf. Ein Wert von Null deaktiviert den Test auf die obere Schranke der maximalen Zellbewegung. Default ist MaxMoveDistMaxBias=0 .

MaxMoveDistMaxGradient

Gibt die negative Steigung der Geraden an, die die obere Schranke der Distanz definiert, um die sich die am weitesten bewegte Zelle während der Iteration bewegt haben darf. Default ist MaxMoveDistMaxGradient=0 .

AvgMoveDistLimit

Untere Schranke der Distanz, um die sich die Zellen im Mittel während der Iteration bewegt haben dürfen. Default ist AvgMoveDistLimit=0 .

AvgMoveDistMaxBias

Gibt den y-Achsenabschnitt der Geraden an, die die obere Schranke der Distanz definiert, um die sich die Zellen im Mittel während der Iteration bewegt haben dürfen. Ein Wert von Null deaktiviert den Test auf die obere Schranke der durchschnittlichen Zellbewegung. Default ist AvgMoveDistMaxBias=0.

AvgMoveDistMaxGradient

Gibt die negative Steigung der Geraden an, die die obere Schranke der Distanz definiert, um die sich die Zellen im Mittel während der Iteration bewegt haben dürfen. Default ist AvgMoveDistMaxGradient=0.

AvgMoveDistShrinkLimit

Obere Schranke des Schrumpfungsfaktors. Der durchschnittliche Wert der Zellbewegungen nimmt von einer Iteration auf die nächste um einen nahezu konstanten Faktor ab. Dieser Faktor steigt zum Ende der Iterationen langsam an. Dieser Parameter ist eine Fließkommazahl im Intervall [0;1). Ein Wert von 1.0 deaktiviert diese Abbruchbedingung. Default ist AvgMoveDistShrinkLimit=0.98.

A.2.1.4 Erzeugen der Standardzellreihen

Die Parameter dieser Gruppe steuern das Aussehen der Standardzellplatzierung, die aus der zuvor ermittelten Anordnung der Zellen in der Ebene erstellt wird.

AspectRatio

Gibt das gewünschte Seitenlängenverhältnis x/y des Chips an. Dieses Verhältnis wird versucht durch Wahl der Zellreihenanzahl und Zellreihenlänge zu erreichen. Die endgültigen Ausmaße des fertigen Layouts können von diesem Seitenverhältnis abweichen. Default ist AspectRatio=1.0.

EstChannelWidthScale und PinsPerColumnScale

Skalierungsfaktoren für die geschätzte Kanalbreite bei der Initialschätzung der Chipgröße. Die Erste Größenabschätzung bei der Erstellung der Standardzellanordnung verwendet eine andere Schätzfunktion für den Platzbedarf von Zellen und Verdrahtungsregionen, als die folgenden Abschätzungen und kann daher hier gesondert parametrisiert werden. Die Kanalbreite wird dabei abhängig von der Anzahl der Netze im Kanal (skaliert mit EstChannelWidthScale) und der Anzahl der zu verdrahtenden Pins je Spalte im Kanal (skaliert mit PinsPerColumnScale). Default ist EstChannelWidthScale=10 und PinsPerColumnScale=20.

ChannelWidthScale

Skalierungsfaktor für die Kanalbreite horizontaler Kanäle bei der Chipgrößenabschätzung auf Basis einer (unvollständigen) Standardzellanordnung der Zellen. Die Kanalbreite wird geschätzt auf Basis der Anzahl im Kanal zu verdrahtenden Netze. Die angenommene Kanalbreite wird dabei mit ChannelWidthScale multipliziert. Default ist ChannelWidthScale=10

VertChannelWidthScale

Skalierungsfaktor für die Kanalbreite vertikaler Kanäle bei der Chipgrößenabschätzung. Die Kanalbreite wird geschätzt auf Basis der Anzahl der an der Chipkante positionierten Padzellen. Die angenommene Kanalbreite wird dabei mit VertChannelWidthScale multipliziert. Default ist ChannelWidthScale=1.

In der in der Arbeit evaluierten Version von prob ist dieser Parameter nicht durch eine Parameterdatei oder eine Abfrage im interaktiven Modus einstellbar.

RowUtilization

Angestrebte Zellreihenfüllung nach abgeschlossener Standardzellplatzierung. Die Zellreihen werden bis zu dem von RowUtilization angegebenen Füllungsgrad mit Zellen und Feedthroughs gefüllt. Die verbleibenden Zwischenräume können vom Verdrahter (zumindest in prob) für neue Feedthroughs und die Verschiebung von Zellen verwendet werden. Default ist RowUtilization=0.95.

MinRowUtilization

Faktor, der die zu erreichende Füllung der letzten Zellreihe angibt. Bei der Anordnung in Standardzellreihen muß die letzte Zellreihe mindestens zu einem Anteil gefüllt sein, der durch $\text{RowUtilization} * \text{MinRowUtilization}$ gegeben ist. Default ist MinRowUtilization=0.75.

GapSearchLimitCell

Gibt die Distanz in Rasterkoordinaten an, innerhalb der beim Einfügen einer Zelle in eine bestehende Zellreihe (während des Sweep-Line Algorithmus) nach einer möglichen Position für diese Zelle gesucht wird. Wird innerhalb des Limits keine passende Lücke gefunden, wird versucht durch Zellverschiebung eine passende Lücke zu schaffen. Default ist GapSearchLimitCell=30.

GapSearchLimitFT

Gibt die Distanz in Rasterkoordinaten an, innerhalb der beim Einfügen einer Feedthroughzelle in eine bestehende Zellreihe (während des Sweep-Line Algorithmus) nach einer möglichen Position für diese Zelle gesucht wird. Wird innerhalb des Limits keine passende Lücke gefunden, wird versucht, durch Zellverschiebung, eine passende Lücke zu schaffen. Default ist GapSearchLimitFT=20.

EqualCellDist

Boolscher Wert, der angibt, ob nach abgeschlossener Anordnung der Zellen in Reihen die in der Netzliste angegebenen Zellen (also ohne eingefügte Feedthroughs) in den Reihen gleichmäßig verteilt angeordnet werden sollen. Diese gleichmäßige Verteilung der Zellen über die Reihen kann für nachfolgende Verdrahter eine günstigere Startposition bieten. Default ist EqualCellDist=1.

A.2.1.5 Modifikationen des Verdrahters an der Platzierung

Die Parameter dieser Gruppe steuern die für den Verdrahter zulässigen Veränderungen an der Platzierung.

UtilTrackWeight

Gewichtungsfaktor der geschätzten benötigten Trackanzahl im Kanal. Dieser Parameter ermöglicht es, die bei der Globalverdrahtung geschätzte Kanalbreite für die

Anhang A Verwendung des Programmes prob

Kanalverdrahtung um den gegebenen Faktor zu verändern. Default ist `UtilTrackWeight=0.6`.

RelativeMovabilityLimit und AbsoluteMovabilityLimit

Die Beweglichkeit einer Zelle während der Globalverdrahtung hängt von der Anzahl der bereits angeschlossenen Netze. Wird entweder die absolute Grenze an bereits verdrahteten Netzen (`AbsoluteMovabilityLimit`) oder der Anteil an verdrahteten Pins zur Gesamtzahl an Pins dieser Zelle (`RelativeMovabilityLimit`) überschritten, ist die Zelle für den Verdrahter unbeweglich. Die aktuelle Version unterstützt noch keine Verschiebung von teilweise verdrahteten Zellen, Werte ungleich Null können zu unvorhersehbarem Verhalten des Verdrahters führen. Default ist `RelativeMovabilityLimit=0.0` und `AbsoluteMovabilityLimit=0`.

(Der Tippfehler in „MovabilityLimit“ ist dem Quellcode entnommen und Teil des Parameternamens!)

FTSearchDist

Gibt die Entfernung in Rasterschritten an, in der ausgehend von der Wunschposition für eine Feedthroughzelle nach einer möglichen Einfügeposition für diese gesucht werden soll. Default ist `FTSearchDist=25`.

FTSearchBeyondIntervall

Boolscher Wert, der angibt, ob die Suche nach einer möglichen Einfügeposition für eine Feedthroughzelle über die durch `FTSearchDist` gegebene Entfernung hinaus gesucht werden darf. Default ist `FTSearchBeyondIntervall=1` (entspricht TRUE).

ShiftCells

Boolscher Wert, der angibt, ob während des Verdrahtens prinzipiell Zellen bewegt werden dürfen, um innerhalb einer Zellreihe passende Lücken für Feedthroughzellen zu schaffen. Das Zulassen von Zellverschiebungen für Feedthroughpositionierung (`ShiftCells=1`) erhöht die Flexibilität, mit welcher der Verdrahter nach möglichen Verdrahtungswegen suchen kann, führt jedoch zu einer Veränderung der gegebenen Platzierung. Default ist `ShiftCells=1` (entspricht TRUE).

CandFTDist

Gibt die maximale Entfernung von bei der Globalverdrahtung als mögliche Feedthroughpositionen betrachteten Orte in einer Zellreihe an. Wenn zwei benachbarte Kandidatenpositionen für eine Feedthroughzelle weiter als `CandFTDist` auseinander liegen, wird zwischen ihnen eine weitere Kandidatenpositionen vorgesehen. Default ist `CandFTDist=15`.

StartCellMoveDist

Gibt die maximale Entfernung in Rastereinheiten an, in welcher nach einer günstigeren Position der Startzelle des gerade verdrahteten Zweipunktnetzes gesucht wird. Ein Wert von Null deaktiviert die Verschiebung der Startzelle. Default ist `StartCellMoveDist=0`.

EndCellMoveDist

Gibt die maximale Entfernung in Rastereinheiten an, in welcher nach einer günstigeren Position der ausgewählten Endzelle des gerade verdrahteten Zweipunktnetzes gesucht

wird. Ein Wert von Null deaktiviert die Verschiebung der Endzelle. Default ist `EndCellMoveDist=0`.

Der Verdrahter kann keine Veränderung an der gegebenen Plazierung vornehmen, wenn eine Parametereinstellung von `ShiftCells=0` und `StartCellMoveDist=0` und `EndCellMoveDist=0` vorgegeben ist.

A.2.1.6 Globalverdrahtung und Beurteilung der platzierten Netze

Die Parameter dieser Gruppe geben an, wie die im Routinggraphen repräsentierten möglichen Verdrahtungspfade bewertet werden. Jeder Pfad erhält als Kosten die Summe seiner Kanten- und Knotengewichte. Die Parameter geben die Gewichte der betrachteten Eigenschaften für Knoten und Kanten des Graphen an.

EdgeRatingLengthWeight

Gibt das Gewicht der Kantenlänge in Rastereinheiten für diese Kante des Graphen an. Die Kantenlänge wird für große Layouts größer sein als für kleine, daher kann hier gegebenenfalls eine Anpassung erforderlich sein, um das Verhältnis zum Einfluß der Kanalauslastung beizubehalten. Default ist `EdgeRatingLengthWeight=1`.

EdgeRatingAvgUtilWeight und EdgeRatingMaxUtilWeight

Diese Werte sind Gewicht für den Einfluß der Kanalauslastung auf dem von dieser Kante verwendeten Abschnitt des Kanals. Die Gewichte beziehen sich auf die durchschnittliche (`EdgeRatingAvgUtilWeight`) bzw. maximale (`EdgeRatingMaxUtilWeight`) Auslastung des Kanals. Die Kanalauslastung selbst wird gemessen mit einer Fließkommazahl im Intervall $[0;1]$ und ist damit, im Gegensatz zur Länge der Kante, unabhängig von der Chipgröße. Default ist `EdgeRatingAvgUtilWeight=25` und `EdgeRatingMaxUtilWeight=150`.

EdgeWeight

Gibt den Gewichtungsfaktor für den Einfluß einer Kante zu den Gesamtkosten des Pfades im Routinggraphen an. Die Kosten einer Kante werden durch die drei zuvor genannten Parameter ebenfalls beeinflusst. Default ist `EdgeWeight=10`.

NodeWeight

Gibt den Gewichtungsfaktor für einen Knoten entlang des betrachteten Pfades im Routinggraphen an. Die Knotenkosten sind bestimmt durch den Abstand des Knotens vom Netzmittelpunkt. Default ist `NodeWeight=5`.

NetExtentPerimeterWeight und NetExtentPerimeterRange

Diese Parameter geben die Möglichkeit, Pins, die sich nahe des Randes eines Netzes befinden, mit stärkerem Gewicht in die Bestimmung des Netzmittelpunktes eingehen zu lassen. Dies modelliert, daß eine Verschiebung dieser Pins mehr Einfluß auf die Ausdehnung des Netzes hat, als die Verschiebung eines Pins im Inneren des überdeckten Gebietes. Das Gewicht eines Pins, der nicht weiter als `NetExtentPerimeterRange` von der größten bzw. kleinsten x- oder y Koordinatenposition eines Pins des Netzes entfernt liegt, wird mit `NetExtentPerimeterWeight` multipliziert in der Bestimmung des neuen Netzmittelpunktes berücksichtigt. Default ist `NetExtentPerimeterWeight=5` und `NetExtentPerimeterRange=10`.

A.2.1.7 Interne Speicherverwaltung

Die Parameter dieser Gruppe sind nur über die Parameterdatei zugreifbar und dienen der Kontrolle der Speicherverwaltung für dynamische Datenstrukturen. An diesen Parametern sollten keine Änderungen vorgenommen werden, da sie die Programstabilität entscheidend beeinflussen. Nehmen Sie hier nur Änderungen vor, wenn Sie wissen, was Sie tun!

MaxExpandWhileRouting

Dieser Parameter gibt an, um welchen Faktor sich die Chipgröße, ausgehend von der bei der Platzierung ermittelten (oder eingelesenen) Größe, maximal ändern darf. Dieser Wert gibt somit an, um wieviel die betroffenen Datenstrukturen bewußt zu groß ausgelegt werden. In Systemen mit beschränkten Speicherressourcen kann eine Verkleinerung des Wertes zu Geschwindigkeitsvorteilen bei der Verdrahtung führen. Eine Vergrößerung des Parameterwertes kann Programmabbrüche während der Verdrahtung vermeiden. Default ist `MaxExpandWhileRouting=10.0`.

TrackArrayExpandStep

Gibt an, um welchen Faktor die Größe der Datenstrukturen für Informationen zu Tracks vergrößert werden, wenn die bisherige Größe nicht ausreicht. Dieser Parameter hat nur Einfluß auf den Verdrahter. Default ist `TrackArrayExpandStep=1.1`.

A.2.2 Formen der Parametereingabe

Alle Parameter für die in prob verwendeten Algorithmen können vom Benutzer vorgegeben werden. Die beiden Methoden sind die Parameterdatei und die interaktive Eingabe. Bei Programmstart wird der Benutzer neben den Namen der Ein- und Ausgabedateien auch nach den Werten der einstellbaren Parameter gefragt. Es wird jeweils der voreingestellte Wert angezeigt, der Benutzer kann diesen beliebig ändern. Wird eine Parameterdatei angegeben, liest das Programm alle darin enthaltene Parameter und ersetzt die Werte der Standardeinstellung durch diese. Werden in der Parameterdatei nicht alle Werte spezifiziert, werden die nicht angegebenen Werte anschließend interaktiv erfragt.

A.2.2.1 Interaktive Parametereingabe

Die Parameter sind in den oben eingeführten Gruppen strukturiert. Jede Gruppe wird zunächst angezeigt und der Benutzer kann entweder die Werte aller Parameter dieser Gruppe akzeptieren oder ablehnen. Werden die Werte der Parameter einer Gruppe abgelehnt, so erfolgt anschließend eine Aufforderung, jeden Wert neu einzugeben. Die bisher gültigen Werte (das sind die Defaultwerte oder Werte aus einer Parameterdatei) werden dabei wiederum angezeigt. Nach Eingabe aller Parameter der Gruppe wird die gesamte Gruppe erneut angezeigt und die Bestätigung der neuen Werte erwartet. Sind alle Parametergruppen bestätigt, beginnt der eigentliche Programmlauf.

A.2.2.2 Parameterdatei

Die Angabe einer Parameterdatei ermöglicht es zum einen, die Programmausführung im Batchbetrieb (siehe A.5) durchzuführen und zum anderen, die Ergebnisse vorheriger Programmläufe präzise zu reproduzieren. Dies zu unterstützen dient die in A.1 erwähnte

automatisch generierte Parameterdatei. Die Parameterdatei lehnt sich in ihrem Format an die bereits vor der Entstehung von prob existierenden ESPRO Dateiformate an.

Im Gegensatz zur interaktiven Parametereingabe, ermöglicht die Parameterdatei dem Benutzer die Charakteristika der modellierten Zellbibliothek zu verändern. Diese Gruppe wird im interaktiven Modus nicht abgefragt. Werden Veränderungen an dieser Gruppe vorgenommen, lassen sich die mit anderen ESPRO Modulen erzeugten Plazierungen nicht verwenden, da diese Änderungen eine Neuerung in prob darstellen. Diese Erweiterung ermöglicht es, mehr Variabilität innerhalb der Grenzen des ESPRO-Modells zu unterstützen und soll vor allem für weitere Arbeiten mit ESPRO mehr Flexibilität offerieren.

Die Parameterdatei besteht aus zwei Teilen, einem Kopfteil (Header) und dem Rumpf (Body).

Im Header steht eine Zeichenkette, die diese Datei als ESPRO Parameterdatei kennzeichnet. Außerdem wird der Name des Schaltkreises angegeben, zu dem diese Parametereinstellung gehört. Die Angabe der Schaltkreisnamen ist notwendig, der Name wird jedoch bei der Verwendung der Datei nicht auf Übereinstimmung mit dem Schaltkreisnamen aus der Basisdatei getestet, wie bei Plazierungsdateien. In automatisch generierten Parameterdateien wird der Name aus der Basisdatei verwendet, manuell erstellte Parameterdateien können hier eine beliebige (nicht leere) Zeichenkette enthalten.

```
>Configuration_of_Circuit: test1000
```

Abbildung 62: Kopfzeile einer Parameterdatei für den Schaltkreis mit dem Namen "test1000"

Der Rumpf enthält die anzuwendenden Werte für die Parameter. Jede Zeile enthält dabei den Namen und neuen Wert eines Parameters. Zeilen, die vom Programm ausgewertet werden sollen, also alle Zeilen mit Parameterangaben, müssen mit einem „>“ beginnen. Die Werteingabe erfolgt nach der Syntax > <Parametername>= <Wert>. Zu beachten ist, daß der Parametername und das Gleichheitszeichen nicht durch ein Leerzeichen getrennt sein dürfen. Die folgende Abbildung zeigt eine Zeile aus dem Rumpf einer Parameterdatei.

```
>AspectRatio= 1.000000
```

Abbildung 63: Zuweisung eines Parameterwertes im Rumpf einer Parameterdatei

Wird einem Parameter mehrmals ein Wert zugewiesen, so überschreibt die letzte Zuweisung alle zuvor vorgenommenen.

Alle Zeilen, die nicht mit einem „>“ beginnen, werden ignoriert und können als Kommentarzeilen genutzt werden. Automatisch erzeugte Parameterdateien nutzen dies, um mit Kommentarzeilen die verschiedenen Parametergruppen zu trennen.

A.3 Beispiel einer Parameterdatei

Hier wird eine komplette Parameterdatei aufgelistet, um ohne Zugriff auf den Inhalt der CD eine Anschauung zu liefern.

```
>Configuration_of_Circuit: test1000

# ----- parameters characterizing the cell library -----
>PortRadius= 14
>CellHeight= 13
>CellWidthBias= 1
>CellWidthPin= 2
>FTWidthBias= 0
>FTWidthPin= 1
>FTWidth= 1
# ----- parameters for hierarchical net-partitioning algorithm -----
>Placement= 1
>Routing= 1
>TotalNetBalanceWeight= 30
>CutSizeWeight= 20
>ExtCutSizeWeight= 75
>MaxLeafSize= 1
# ----- parameters for force directed placement -----
>MaxCentreIterations= 0
>MinCentreIterations= 0
>MaxMoveDistLimit= 0
>MaxMoveDistMaxBias= 0
>MaxMoveDistMaxGradient= 0
>AvgMoveDistLimit= 0
>AvgMoveDistMaxBias= 0
>AvgMoveDistMaxGradient= 0
>AvgMoveDistShrinkLimit= 0.980000
# ----- parameters for row organization -----
>AspectRatio= 1.000000
>EstChannelWidthScale= 10.000000
>PinsPerColumnScale= 20.000000
>ChannelWidthScale= 10.000000
>RowUtilization= 0.950000
>MinRowUtilization= 0.700000
>GapSearchLimitCell= 30
>GapSearchLimitFT= 20
>EqualCellDist= 1
# ----- parameters for placement modification by router -----
>UtilTrackWeight= 0.600000
>RelativeMovabilityLimit= 0.000000
>AbsoluteMovabilityLimit= 0
>FTSearchDist= 25
>FTSearchBeyondIntervall= 1
>ShiftCells= 1
>CandFTDist= 15
>StartCellMoveDist= 0
>EndCellMoveDist= 0
# ----- parameters for global routing -----
>EdgeRatingLengthWeight= 1
>EdgeRatingAvgUtilWeight= 25
>EdgeRatingMaxUtilWeight= 150
>EdgeWeight= 10
>NodeWeight= 5
>NetExtentPerimeterWeight= 5
>NetExtentPerimeterRange= 10
```



```
# ----- parameters for internal memory organization -----
>MaxExpandWhileRouting= 10.000000
>TrackArrayExpandStep= 1.100000
```

Abbildung 64: Eine Konfigurationsdatei für prob

A.4 Eine Beispielsitzung im interaktiven Modus

```
ole@p3:~/da/sw/data > ../bin/prob -c test100.chp
Welcome to prob, a part of the ESPRO sytem.

Result of Command Line interpretation:
Name of placement file [ test100.pla ] correct ? (y/n) y
Name of placement after routing file [ test100.plr ] correct? (y/n)
y
Name of layout file [ test100.lay ] correct ? (y/n) n
Enter filename (default extension .lay): test100.la6p

Layoutfilename: test100.la6p

----- parameters for hierarchical net-partitioning algorithm -----
TotalNetBalanceWeight= 30
CutSizeWeight= 20
ExtCutSizeWeight= 75
MaxLeafSize= 1

Do you want to change any of the above values ? (y/n) n

----- parameters for force directed placement -----
MaxCentreIterations= 0
MinCentreIterations= 0
MaxMoveDistLimit= 0
MaxMoveDistMaxBias= 0
MaxMoveDistMaxGradient= 0
AvgMoveDistLimit= 0
AvgMoveDistMaxBias= 0
AvgMoveDistMaxGradient= 0
AvgMoveDistShrinkLimit= 0.980000

Do you want to change any of the above values ? (y/n) y

Please enter the new values
MaxCentreIterations= 0 new value: 1000
MinCentreIterations= 0 new value: 10
MaxMoveDistLimit= 0 new value: 20
MaxMoveDistMaxBias= 0 new value: 0
MaxMoveDistMaxGradient= 0 new value: 0
AvgMoveDistLimit= 0 new value: 10
AvgMoveDistMaxBias= 0 new value: 0
AvgMoveDistMaxGradient= 0 new value: 0
AvgMoveDistShrinkLimit= 0.980000 new value: 1

----- parameters for force directed placement -----
MaxCentreIterations= 1000
MinCentreIterations= 10
MaxMoveDistLimit= 20
MaxMoveDistMaxBias= 0
MaxMoveDistMaxGradient= 0
AvgMoveDistLimit= 10
```

Anhang A Verwendung des Programmes prob

```
AvgMoveDistMaxBias= 0
AvgMoveDistMaxGradient= 0
AvgMoveDistShrinkLimit= 1.000000
```

Do you want to change any of the above values ? (y/n) n

```
----- parameters for row organization -----
AspectRatio= 1.000000
EstChannelWidthScale= 10.000000
PinsPerColumnScale= 20.000000
ChannelWidthScale= 10.000000
RowUtilization= 0.950000
MinRowUtilization= 0.700000
GapSearchLimitCell= 30
GapSearchLimitFT= 20
EqualCellDist= 1
```

Do you want to change any of the above values ? (y/n) y

```
Please enter the new values
AspectRatio= 1.000000 new value: 1
EstChannelWidthScale= 10.000000 new value: 10
PinsPerColumnScale= 20.000000 new value: 20
ChannelWidthScale= 10.000000 new value: 10
RowUtilization= 0.950000 new value: 0.95
MinRowUtilization= 0.700000 new value: 0.75
GapSearchLimitCell= 30 new value: 30
GapSearchLimitFT= 20 new value: 20
EqualCellDist= 1 new value: 0
```

```
----- parameters for row organization -----
AspectRatio= 1.000000
EstChannelWidthScale= 10.000000
PinsPerColumnScale= 20.000000
ChannelWidthScale= 10.000000
RowUtilization= 0.950000
MinRowUtilization= 0.750000
GapSearchLimitCell= 30
GapSearchLimitFT= 20
EqualCellDist= 0
```

Do you want to change any of the above values ? (y/n) n

```
----- parameters for placement modification by router -----
UtilTrackWeight= 0.600000
RelativeMovabilityLimit= 0.000000
AbsoluteMovabilityLimit= 0
FTSearchDist= 25
FTSearchBeyondIntervall= 1
ShiftCells= 1
CandFTDist= 15
StartCellMoveDist= 0
EndCellMoveDist= 0
```

Do you want to change any of the above values ? (y/n) n

```
----- parameters for global routing -----
EdgeRatingLengthWeight= 1
EdgeRatingLengthUtilWeight= 25
```

```
EdgeRatingAvgUtilWeight= 25
EdgeRatingMaxUtilWeight= 150
EdgeWeight= 10
NodeWeight= 5
```

```
----- parameters for placed net rating -----
```

```
NetExtentPerimeterWeight= 5
NetExtentPerimeterRange= 10
```

```
Do you want to change any of the above values ? (y/n) n
```

```
Reading Basefile
Reading cell connectivity list...
Cell 99... done
Reading netlists...
74 ths net ... done
```

```
Level 0
Iteration: 1 2 3
```

```
Level 1
Iteration: 1 2 3 4
```

```
Level 2
Iteration: 1 2 3 4 5
```

```
Level 3
Iteration: 1 2 3 4 5 6
```

```
Level 4
Iteration: 1 2 3 4 5
```

```
Level 5
Iteration: 1 2 3 4 5
```

```
Level 6
Iteration: 1 2
```

```
Level 7
Iteration: 1 2 Locating Cells... done
Iterating centre Algorithm
```

```
0 Iteration 1: MaxDist: 1260213407, AvgDist: 361105179 Max/Avg
3.490 Shrink: 0.084 MaxLimit 4294967295 AvgLimit 4294967295
1 Iteration 2: MaxDist: 400522155, AvgDist: 147793782 Max/Avg
2.710 Shrink: 0.409 MaxLimit 4294967295 AvgLimit 4294967295
2 Iteration 3: MaxDist: 201264765, AvgDist: 120115041 Max/Avg
1.676 Shrink: 0.813 MaxLimit 4294967295 AvgLimit 4294967295
```

```
. . . . [here are some lines intentionally missising] . . .
```

```
98 Iteration 99: MaxDist: 27, AvgDist: 20 Max/Avg 1.350 Shrink:
0.870 MaxLimit 4294967295 AvgLimit 4294967295
99 Iteration 100: MaxDist: 23, AvgDist: 17 Max/Avg 1.353 Shrink:
0.850 MaxLimit 4294967295 AvgLimit 4294967295
100 Iteration 101: MaxDist: 20, AvgDist: 14 Max/Avg 1.429 Shrink:
0.824 MaxLimit 4294967295 AvgLimit 4294967295
Ordering Cells... done
estimated chip extent : (703 , 674)
possible pad slots : (15 , 14)
number of rows required : 3 with max. length 641
```

Anhang A Verwendung des Programmes prob

```
last row contains to few cells or no cells at all
estimated chip extent : (536 , 593)
possible pad slots    : (11 , 12)
number of rows required : 3 with max. length 474
all cells placed, last row filled, test aspect ratio
estimated chip extent : (536 , 593)
possible pad slots    : (11 , 12)
number of rows required : 3 with max. length 474
the result is satisfying
estimated chip extent : (536 , 593)
possible pad slots    : (11 , 12)
number of rows required : 3 with max. length 474
estimated average channel width: 162
Initializing Net extent... done
Writing placement file
Writing Placement:
    writing pad locations
    writing cell positions
    writing cellrows
writing placement complete.

Starting routing
Initializing all Nets in row-system ... done
GlobalCoreRouting
    routing Cell 95 :    done
Verifying net conectivity
    checking net 74    done
Routing pad cells
Routing horizontal channels:
    horizontal channel 0
    horizontal channel 1
    horizontal channel 2
    horizontal channel 3

Routing vertical channels
    left vertical channel
    right vertical channel
Updating global layout data (vias)
Updating global layout data
Writing placement modified by the router to file
Writing Placement:
    writing pad locations
    writing cell positions
    writing cellrows
writing placement complete.
Writing layoutfile
Writing layout file
    writing pad locations
    writing x-segments
    writing y-segments
    building ordered list of vias
    writing vias
writing layout finished

Time required for Placing:          0: 0' 0"

Time required for Global-Routing:   0: 0' 0"
Time required for Channel-Routing:  0: 0' 0"
-----
Time required for Routing:          0: 0' 0"

Final chip extent in grid coords: ( 536, 743)
Total number of tracks inserted into channels: 25
```

```
Total number of tracks removed from channels: 6
Netto Track statistic: 19
Average netlength: (without,with) pads: 1.593 , 1.755
Maximum netlength: (without,with) pads: 4.165 , 4.187

That's It. Bye
ole@p3:~/da/sw/data >
```

A.5 Batch-Betrieb

Der Batch-Betrieb wird durch die Kommandozeilenschalter `-B` oder `-b <batchfile>` aktiviert. In dieser Betriebsform werden keine Benutzerinteraktionen durchgeführt. Alle nicht angegebenen Dateinamen und Parameterwerte werden mit den Defaultwerten belegt. Für Dateinamen ist der Default der Name der Basisdatei ohne Dateiendung und Pfadangaben. Wird auch der Basisdateiname nicht angegeben, so wird als Basisdatei `test.chp` im aktuellen Verzeichnis angenommen. Ist die spezifizierte Parameterdatei vollständig (und wurden die Werte nicht im forcierten interaktiven Modus verändert), so wird keine Log- und keine Parameterdatei für diesen Programmlauf erzeugt.

Diese Betriebsform ermöglicht es, bei Verwendung der bei einem vorherigen Programmlauf von `prob` generierten Parameter- und Logdatei das zuvor produzierte Layout zu reproduzieren. Dies mag in Anbetracht der Größe der Layoutdateien für größere Entwürfe (>1000 Zellen) von Interesse sein.

Im Batch-Betrieb ist es auf bequeme Art möglich eine größere Anzahl von Layouts zu produzieren, sofern dies gewünscht ist, z.B. für die Ermittlung von günstigen Parametereinstellungen.

A.6 Beispiel einer Batch- bzw. Logdatei

Hier wird ein Beispiel einer automatisch generierten Logdatei angeführt. Diese Dateien sind im Format identisch zu den Dateien für den Batchbetrieb. Diese Dateien enthalten die Dateinamen aller verwendet bzw. erzeugten Dateien, sowie das Datum der Programmausführung und den Namen des bearbeiteten Schaltkreises. Der Aufbau ist nicht an die anderen Dateien für ESPRO angelehnt.

```
logfile for PROB execution: test100

Date: Mon Oct 29 08:59:32 2001

>BaseFileName= ../test100.chp
>PlacementFilename= test100-part-row2-no.pl6
>PlacementFileName= test100-part-row2-no.plr6
>LayoutFileName= test100-part-row2-no.la6p
>ConfigFileName= ../param/t-pr-part-row2-no.set
```

Abbildung 65: Dateilog-File für eine der Programmausführungen von `prob`, mit Verwendung beider enthaltener Module, die Daten für Tabelle 11 lieferte.

B. PARAMETERWERTE DER EVALUIERUNG

Dieser Anhang enthält Tabellen mit Wertebestellungen der Parameter, die im Laufe der Evaluierungsphase verwendet wurden.

Tabelle B- 1: Werte der Parameter zur Beurteilung der Pfade beim Globalverdrahten.

Parametername	Werte									
LengthWeight	1									
LengthUtilWeight	0	5	10	15	20	25	50	100	200	500
AvgUtilWeight	25									
MaxUtilWeight	0	50	100	150	200	250	300	500	1000	
EdgeWeight	10									
NodeWeight	0	1	2	5	10	15	20	50	100	

Tabelle B- 2: Verwendete Gewichtungsfaktoren für die Bewertung des Längeneinflusses auf die Pfadbewertung

Parametername	Werte									
LengthWeight	1	2	3	5	10	15	20	30	40	50

Tabelle B- 3: Untersuchte Parametrisierungen für die Platzierungsmodifikation durch den Verdrahter

Parametername	Werte											
FTSearchDist	0	25	50	75	100	150	200	250	300	350	400	500
StartCellMoveDist	0	25	50	75	100	150	200	250				
EndCellMoveDist												
ShiftCells	0 (False)			1 (True)								

Anhang B Parameterwerte der Evaluierung

Die folgende Tabelle gibt die untersuchten Parametrisierungen der Transformation der Vorplazierung in eine Standardzellplazierung an. Für die Erzeugung der zugrundeliegenden Vorplazierung wurden folgende Werte verwendet:

TotalNetBalanceWeight=30, CutSizeWeight=20 und

ExtCutSizeWeight=10.

Tabelle B- 4: Parametrisierung zur Evaluierung der Eigenschaften des Plazierungsschrittes zur Überführung der Vorplazierung in eine Standardzellplazierung

Parametername	Werte							
AspectRatio	1.0							
EstChannelWidthScale	10							
PinsPerColumnScale	20							
ChannelWidthScale	10							
RowUtilization	0,75	0,80	0,90	0,95				
MinRowUtilization	0,70							
GapSearchLimitCell	0	10	20	30	50	75	100	
GapSearchLimitFT	0	10	20	30	50	75	100	150
EqualCellDist	False		True					

Tabelle B- 5: Parametrisierung zur Evaluierung des Plazierungsschrittes der hierarchischen Partitionierung.

Parametername	Werte											
CutSizeWeight	1	2	3	5	10	15	20	30	50	75	100	
ExtCutSizeWeight	1	2	3	5	10	15	20	30	50	75	100	
TotalNetBalanceWeight	1	2	3	5	10	15	20	30	50	75	100	

Die Parametrisierung der kräftebasierten Plazierung ermöglicht die Definition mehrerer Abbruchkriterien. Zu den in der folgenden Tabelle gezeigten Parametern wurden für die anderen Abbruchkriterien folgende Wertebelegungen verwendet:

MaxMoveDistLimit=0, MaxMoveDistMaxBias=0,

MaxMoveDistMaxGradient=0, AvgMoveDistLimit=0,

AvgMoveDistMaxBias=0, AvgMoveDistMaxGradient=0.

Die folgende Tabelle gibt die Werte für die durchgeführten Iterationszahlen bei der kräftegesteuerten Plazierung an. Die beiden genannten Parameter sind stets auf den gleichen Wert gesetzt.

Tabelle B- 6: Evaluierte Iterationsanzahlen der kräftegesteuerten Plazierung.

Parametername	Werte											
MaxCentreIterations	0	5	10	15	20	30	50	75	100	125	150	175
MinCentreIterations		200	225	250	275	300	350	400	450	500	600	700

C. ERGEBNISSE DER PARAMETRISIERUNG

Dieser Anhang enthält Tabellen mit Zwischenergebnissen, die im Ergebniskapitel aus Gründen der Übersicht nicht erschienen sind. Die Tabelleneinträge für Parameterwerte erfolgen in der für den jeweiligen Parameter in Anhang A definierten Einheit. Angaben zu Längen verstehen sich in mm, Flächen sind in mm² angegeben. Alle Zeitangaben verstehen sich in Sekunden.

C.1 Parametrisierung des Verdrahters

Die folgenden beiden Tabellen zeigen Zwischenergebnisse zu den Parametrisierungen gemäß der Tabelle B- 1.

Tabelle C- 1: Für die Netzliste test500 ist für jedes Knotengewicht die jeweils bezüglich maximaler Netzlänge bzw. benötigter Chipfläche beste Parametrisierung der Pfadbewertung aufgelistet.

Rang Maximale Netzlänge	LengthUtilWeight	MaxUtilWeight	NodeWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche	Rang Chipfläche
1	20	150	5	18.548	8.502	30.011	6
2	500	150	2	18.760	8.532	30.086	8
3	100	100	10	18.872	8.626	30.285	10
4	15	150	1	19.006	8.542	30.086	8
5	15	0	20	19.012	8.745	30.785	14
6	10	50	2	19.207	8.435	29.836	1
7	15	500	0	19.408	8.453	29.936	3
8	5	0	1	19.861	8.502	29.911	2
9	10	50	0	19.884	8.421	29.761	4
10	25	0	20	19.610	8.677	30.585	11
11	500	500	50	19.872	8.774	30.835	15
12	5	0	10	19.576	8.524	30.011	6
13	25	0	100	19.951	8.747	30.934	16
14	5	0	5	20.180	8.480	29.861	5
15	50	150	50	20.236	8.723	30.660	12
16	200	100	100	20.320	8.726	30.710	13

Die Tabelle C- 1 ist zeilenweise nach steigender maximaler Netzlänge sortiert, die rechteste Spalte gibt die Rangfolge der Parametrisierungen bezüglich der Chipgröße an.

Anhang C Ergebnisse der Parametrisierung

Die durchschnittliche Netzlänge zeigt keine signifikante Abhängigkeit von der Parametrisierung und wird als Auswahlkriterium für die core-dominierten künstlich erzeugten Netzlisten nicht weiter betrachtet.

Tabelle C- 2: Eine Auswahl der 160 betrachteten Parametrisierungen zur Pfadbewertung, nach der Variation des Längengewichtungsparameters (vergleiche Abschnitt 5.3.1.1) für die Netzliste `test500`. Die Tabelle ist zeilenweise aufsteigend nach dem Kriterium der maximalen auftretenden Netzlänge sortiert.

Rang	Maximale Netzlänge	LengthWeight	LengthUtilWeight	MaxUtilWeight	NodeWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	1	20	150	5	18.548	8.502	30.011	
2	1	500	150	2	18.760	8.532	30.086	
3	2	5	0	10	18.788	8.506	29.936	
4	15	25	0	100	18.805	8.527	29.986	
5	20	500	150	2	18.822	8.509	29.986	
6	3	5	0	10	18.855	8.471	29.811	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
160	10	10	50	0	20.784	8.572	30.185	

Tabelle C- 3: Eine Auswahl der 160 betrachteten Parametrisierungen zur Pfadbewertung für die Netzliste test500, die auch Grundlage für Tabelle C- 2 sind, nach der Variation des Längengewichtungsparameters (vergleiche Abschnitt 5.3.1.1). Die Tabelle ist zeilenweise aufsteigend nach dem Kriterium der benötigten Chipfläche sortiert.

Rang Chipfläche	LengthWeight	LengthUtilWeight	MaxUtilWeight	NodeWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	50	15	0	20	18.861	8.329	29.511
2	15	15	0	20	20.063	8.401	29.561
3	10	25	0	20	20.504	8.397	29.586
4	20	15	0	20	20.012	8.395	29.611
5	30	15	0	20	19.738	8.395	29.611
6	5	5	0	5	19.699	8.425	29.636
7	40	50	150	50	19.235	8.412	29.636
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
160	2	25	0	100	19.733	8.813	31.059

Anhang C Ergebnisse der Parametrisierung

Tabelle C- 4: Eine Auswahl der Parametrisierungen der bezüglich der maximalen Netzlänge betrachteten Parametrisierungen zur Platzierungsmodifikation, ausgehend von der Pfadbewertung zur Verringerung maximaler Netzlängen (vergleiche Tabelle C- 2) für die Netzliste test500.

Rang	Maximale Netzlänge	FTSearchDist	StartCellMoveDist EndCellMoveDist	ShiftCells	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	25	0	False	18.548	8.502	30.011	
2	50	0	True	18.771	8.495	30.061	
3	50	0	False	18.849	8.508	30.036	
4	25	0	True	18.945	8.525	30.011	
5	100	0	False	19.112	8.652	30.460	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
25	75	250	False	20.627	9.622	33.032	
26	25	25	True	20.756	8.895	30.984	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
192	100	150	False	26.620	10.897	36.252	

Tabelle C- 5: Eine Auswahl der Parametrisierungen der bezüglich der Chipgröße betrachteten Parametrisierungen zur Platzierungsmodifikation, ausgehend von der Pfadbewertung zur Verringerung der benötigten Chipfläche (vergleiche Tabelle C- 3) für die Netzliste test500.

Rang Chipfläche	FTSearchDist	StartCellMoveDist EndCellMoveDist	ShiftCells	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	25	0	False	18.861	8.329	29.511
2	25	0	True	18.877	8.334	29.536
3	50	0	True	19.694	8.478	29.911
4	50	0	False	19.694	8.498	29.936
5	100	0	False	19.979	8.545	30.036
:	:	:	:	:	:	:
23	25	25	True	21.108	8.786	30.610
:	:	:	:	:	:	:
192	100	100	False	22.125	11.007	36.552

Tabelle C- 6: Für die Netzliste pk690 ist für jedes Knotengewicht die jeweils bezüglich maximaler bzw. durchschnittlicher Netzliste beste Parametrisierung der Pfadbewertung aufgelistet

Rang	Maximale Netzlänge	LengthUtilWeight	MaxUtilWeight	NodeWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche	Rang Durchschnittliche Netzlänge
1	500	0	0	6.339	1.970	25.008	2	
2	25	0	1	6.563	1.988	25.008	9	
3	25	5	20	6.579	2.007	25.008	13	
4	15	0	10	6.591	1.988	25.008	9	
5	500	200	50	6.596	2.014	25.008	14	
6	200	200	2	6.607	2.031	25.008	16	
7	5	250	5	6.619	2.004	25.008	12	
8	500	500	100	6.647	2.022	25.008	15	
9	25	0	5	6.658	1.975	25.008	4	
10	25	0	10	6.742	1.970	25.008	2	
11	0	0	1	6.764	1.984	25.008	7	
12	15	5	20	6.809	1.984	25.008	7	
13	100	0	50	6.876	1.980	25.008	5	
14	20	250	0	6.948	1.951	25.008	1	
15	200	10	100	6.948	1.982	25.008	6	
16	25	0	2	7.055	1.993	25.008	11	

Die Tabelle C- 6 ist, genau wie Tabelle C- 1, zeilenweise nach zunehmender maximaler Netzlänge sortiert. Die rechteste Spalte gibt die Rangfolge nach dem zweiten betrachteten Qualitätskriterium, hier die durchschnittliche Netzlänge, wieder. Die zuvor als Bewertungskriterium betrachtete Chipfläche wird durch die Verdrahtungsqualität nicht beeinflusst (pad-domiertes Layout).

Tabelle C- 7: Eine Auswahl der 160 betrachteten Parametrisierungen zur Pfadbewertung, nach der Variation des Längengewichtungsparameters (vergleiche Abschnitt 5.3.1.1) für die Netzliste pk690. Die Tabelle ist zeilenweise aufsteigend nach dem Kriterium der maximalen auftretenden Netzlänge sortiert.

Rang	Maximale Netzlänge	LengthWeight	LengthUtilWeight	MaxUtilWeight	NodeWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	1	500	0	0	6.339	1.970	25.008	
2	15	15	50	20	6.496	1.985	25.008	
3	1	20	250	0	6.501	1.975	25.008	
4	1	500	0	0	6.507	1.963	25.008	
5	10	15	0	10	6.518	2.004	25.008	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
160	3	200	500	2	7.781	2.030	25.008	

Anhang C Ergebnisse der Parametrisierung

Tabelle C- 8: Eine Auswahl der 160 betrachteten Parametrisierungen zur Pfadbewertung für die Netzliste `test500`, die auch Grundlage für Tabelle C- 7 sind, nach der Variation des Längengewichtungsparameters (vergleiche Abschnitt 5.3.1.1). Die Tabelle ist zeilenweise aufsteigend nach dem Kriterium der durchschnittlichen Netzlänge sortiert.

Rang	Durchschnittliche Netzlänge	LengthWeight	LengthUtilWeight	MaxUtilWeight	NodeWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	10	500	0	0	6.792	1.932	25.008	
2	50	20	250	0	6.769	1.940	25.008	
3	30	500	0	0	6.535	1.944	25.008	
4	50	500	0	0	7.496	1.948	25.008	
5	1	20	250	0	6.948	1.951	25.008	
6	15	20	250	0	6.831	1.952	25.008	
7	20	20	250	0	7.178	1.956	25.008	
8	15	500	0	0	6.859	1.957	25.008	
9	10	100	0	50	6.613	1.959	25.008	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
160	10	200	200	2	7.396	2.032	25.008	

Tabelle C- 9: Eine Auswahl der Parametrisierungen der bezüglich der maximalen Netzlänge betrachteten Parametrisierungen zur Platzierungsmodifikation, ausgehend von der Pfadbewertung zur Verringerung maximaler Netzlängen (vergleiche Tabelle C- 7) für die Netzliste pk690.

Rang	Maximale Netzlänge	Pfadbewertung	FTSearchDist	StartCellMoveDist EndCellMoveDist	ShiftCells	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	edge	25	50	False	6.306	1.920	25.008	
2	edge	25	0	False	6.339	1.970	25.008	
3	edge	150	25	True	6.434	1.950	25.008	
4	edge	250	75	True	6.451	1.992	25.008	
5	node	25	0	False	6.496	1.985	25.008	
6	node	25	50	False	6.512	1.950	25.008	
:	:	:	:	:	:	:	:	
384	node	400	200	True	13.645	2.142	25.008	

Tabelle C- 10: Eine Auswahl der Parametrisierungen der bezüglich der durchschnittlichen Netzlänge betrachteten Parametrisierungen zur Platzierungsmodifikation, ausgehend von der Pfadbewertung zur Verringerung maximaler Netzlängen (vergleiche Tabelle C- 7) für die Netzliste pk690.

Rang	Durchschnittliche Netzlänge	Pfadbewertung	FTSearchDist	StartCellMoveDist EndCellMoveDist	ShiftCells	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	edge	25	50	False	6.306	1.920	25.008	
2	edge	50	25	False	6.585	1.931	25.008	
3	edge	50	50	True	7.133	1.935	25.008	
4	edge	100	0	False	7.317	1.940	25.008	
5	edge	75	25	False	6.714	1.940	25.008	
:	:	:	:	:	:	:	:	
11	node	25	50	False	6.512	1.950	25.008	
:	:	:	:	:	:	:	:	
384	node	25	250	False	10.638	2.201	25.008	

Anhang C Ergebnisse der Parametrisierung

Die folgende Tabelle zeigt die Ergebnisse der Parametrisierung des in prob implementierten Verdrahtungsalgorithmus.

Tabelle C- 11: Für künstlich generierte Netzlisten ermittelte Parametereinstellungen für den Verdrahtungsalgorithmus

Parametername	Bestes Ergebnis bezüglich des betrachteten Kriteriums nach der Parameterevaluation				Bestes Ergebnis mit Veränderung der Plazierung	
Zellverschiebungen für die Feedthroughsuche	nicht zugelassen		zugelassen		gefordert	
Qualitätskriterium	Max. Netzlänge	Chipgröße	Max. Netzlänge	Chipgröße	Max. Netzlänge	Chipgröße
FTSearchDist	25	25	25	25	25	25
StartCellMoveDist	0	0	0	0	25	25
EndCellMoveDist	0	0	0	0	25	25
ShiftCells	False	False	True	True	True	True
LengthWeight	1	50	1	50	1	50
LengthUtilWeight	20	15	20	15	20	15
AvgUtilWeight	25	25	25	25	25	25
MaxUtilWeight	150	0	150	0	150	0
EdgeWeight	10	10	10	10	10	10
NodeWeight	5	20	5	20	5	20
Dateiname :	t-r-max-no.set	t-r-size-no.set	t-r-max-yes.set	t-r-size-yes.set	t-r-max-move.set	t-r-size-move.set

Tabelle C- 12: Für praktische Netzlisten ermittelte Parametereinstellungen für den Verdrahtungsalgorithmus. Das betrachtete Qualitätskriterium ist die maximale Netzlänge.

Parametername	Bestes Ergebnis ohne Betrachtung der Pfadposition				Bestes Ergebnis mit Betrachtung der Pfadposition	
Modifikation der Plazierung	zugelassen		Nicht zugelassen			
Zellverschiebungen für die Feedthroughsuche zugelassen	Nein	Ja	Nein	Ja	Nein	Ja
FTSearchDist	25	25	25	25	25	25
StartCellMoveDist	50	50	0	0	0	0
EndCellMoveDist	50	50	0	0	0	0
ShiftCells	False	True	False	True	False	True
LengthWeight	1	1	1	1	3	3
LengthUtilWeight	500	500	500	500	15	15
AvgUtilWeight	25	25	25	25	25	25
MaxUtilWeight	0	0	0	0	50	50
EdgeWeight	10	10	10	10	10	10
NodeWeight	0	0	0	0	20	20
Dateiname :	p-r-max-move.set	p-r-max-move2.set	p-r-max-no.set	p-r-max-yes.set	p-r-max-node1.set	p-r-max-node2.set

C.2 Parametrisierung des Plazierers

Dieser Teilanhang zeigt die Ergebnisse der Evaluierung der Parametrisierung des Plazierers. Er teilt sich in drei Abschnitte, je einer mit den Ergebnissen zu den in den Abschnitten des Teilkapitels 6.2.

C.2.1 Parametrisierung der Generierung einer Standardzellplatzierung

Die Tabelle C- 13 und die Tabelle C- 14 geben für eine 95%-ige Reihenauslastung (RowUtilization=0,95) eine Auswahl der Parametrisierungen wieder. Die Ergebnisse für geringere Reihenauslastungen als 90% wurden nicht weiter betrachtet, da geringe Reihenauslastungen zu größerem Flächenbedarf führen. Von den insgesamt 112 Standardzellplatzierungen konnten 74 mit dem `dkrouter` verdrahtet werden.

Tabelle C- 13: Resultate für Parametrisierungen zur Erzeugung einer Standardzellplatzierung gemäß Tabelle B- 4, für 95%ige Reihenauslastung für die Netzliste `test500`, geordnet nach maximaler Netzlänge.

Rang	Maximale Netzlänge	GapSearchLimitCell	GapSearchLimitFT	EqualCellDist	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	75	20	True	46.403	22.144	38.619	
2	30	10	True	48.013	21.274	37.415	
3	100	100	True	48.052	21.887	37.969	
4	20	150	True	48.169	21.574	37.849	
5	10	150	True	48.717	21.409	37.657	
6	0	10	True	48.829	22.137	38.041	
7	50	150	True	48.929	21.650	37.487	
8	30	30	True	48.996	21.602	37.174	
9	20	75	True	49.309	21.464	37.271	
10	100	150	False	50.131	22.190	39.164	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
15	100	150	True	51.612	21.489	37.006	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
74	10	20	False	70.227	22.751	39.263	

Tabelle C- 14: Resultate für Parametrisierungen zur Erzeugung einer Standardzellplazierung gemäß Tabelle B- 4, für 95%ige Reihenauslastung für die Netzliste test500, geordnet nach benötigter Chipfläche.

Rang Chipfläche	GapSearchLimitCell	GapSearchLimitFT	EqualCellDist	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	100	150	True	51.612	21.489	37.006
2	0	100	True	56.308	21.510	37.174
3	30	30	True	48.996	21.602	37.174
4	20	20	True	49.309	21.464	37.271
5	75	10	True	53.658	21.504	37.295
6	75	75	True	56.504	21.308	37.295
7	0	50	True	57.560	22.035	37.415
⋮	⋮	⋮	⋮	⋮	⋮	⋮
74	10	100	False	56.822	22.507	40.159

Anhang C Ergebnisse der Parametrisierung

Die Tabelle C- 15 und Tabelle C- 16 geben analog den obigen Ergebnissen die Resultate für die Netzliste pk690 wieder. Für die weitere Parametrisierung werden wiederum nur die Reihenauslastungen von 95% und 90% betrachtet, die folgenden Tabellen zeigen die Resultate für eine Reihenauslastung von 95%. Von den insgesamt 112 Standardzellplatzierungen konnten 108 mit dem dkrouter verdrahtet werden.

Tabelle C- 15: Resultate für Parametrisierungen zur Erzeugung einer Standardzellplatzierung gemäß Tabelle B- 4, für 95%ige Reihenauslastung für die Netzliste pk690, geordnet nach maximaler Netzlänge.

Rang	Maximale Netzlänge	GapSearchLimitCell	GapSearchLimitFT	EqualCellDist	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	50	20	True	17.653	3.789	38.856	
2	30	10	False	18.341	3.775	41.036	
3	30	75	True	21.320	3.843	39.215	
4	75	50	True	22.315	3.900	40.378	
5	50	10	False	23.109	3.730	39.248	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
15	10	20	False	24.160	3.752	39.366	
16	100	100	True	24.473	3.804	38.926	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
31	10	20	True	25.915	3.821	38.571	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
108	30	50	True	35.021	3.945	41.193	

Tabelle C- 16: Resultate für Parametrisierungen zur Erzeugung einer Standardzellplazierung gemäß Tabelle B- 4, für 95%ige Reihenauslastung für die Netzliste pk690, geordnet nach benötigter Chipfläche.

Rang Chipfläche	GapSearchLimitCell	GapSearchLimitFT	EqualCellDist	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	30	150	True	28.772	3.826	38.520
2	10	120	True	25.915	3.821	38.571
3	75	150	True	28.408	3.742	38.615
4	30	0	True	27.475	3.790	38.739
5	20	150	True	25.585	3.744	38.758
6	50	20	True	17.653	3.789	38.856
7	20	10	True	23.484	3.800	38.878
8	50	100	True	29.750	3.817	38.900
9	100	150	True	24.473	3.804	38.926
⋮	⋮	⋮	⋮	⋮	⋮	⋮
34	10	20	False	24.160	3.752	39.366
⋮	⋮	⋮	⋮	⋮	⋮	⋮
108	20	75	False	31.600	3.875	41.731

Die folgenden beiden Tabellen (Tabelle C- 17 und Tabelle C- 18) geben eine Auswahl geeigneter Parametrisierungen zur Generierung der Standardzellenplazierung aus der Vorplazierung wieder. Die grau unterlegte Spalte enthält die für weitere Betrachtungen der Plaziererparametrisierungen verwenden Werte.

Anhang C Ergebnisse der Parametrisierung

Tabelle C- 17: Für theoretische Netzlisten ermittelte Parametereinstellungen für das Erzeugen einer Standardzellplazierung aus der Vorplazierung. Als Qualitätskriterien wurden sowohl die maximale Netzlänge als auch die Chipgröße berücksichtigt.

Parametername	Werte					
	Bestes Resultat		Kleine Suchradien		Große Suchradien	
Besonderheit						
Äquidistante Zellabstände gefordert	Ja	Nein	Ja	Nein	Ja	Nein
AspectRatio	1,0	1,0	1,0	1,0	1,0	1,0
EstChannelWidthScale	10	10	10	10	10	10
PinsPerColumnScale	20	20	20	20	20	20
ChannelWidthScale	10	10	10	10	10	10
RowUtilization	0,95	0,95	0,95	0,95	0,95	0,95
MinRowUtilization	0,7	0,7	0,7	0,7	0,7	0,7
GapSearchLimitCell	30	30	0	0	100	100
GapSearchLimitFT	20	20	10	10	150	150
EqualCellDist	True	False	True	False	True	False
Bezeichnung :	t-p-row1	t-p-row2	t-p-row3	t-p-row4	t-p-row5	t-p-row6

Tabelle C- 18: Für praktische Netzlisten ermittelte Parametereinstellungen für das Erzeugen einer Standardzellenplazierung aus der Vorplazierung. Als Qualitätskriterien wurden sowohl die maximale Netzlänge als auch die Chipgröße berücksichtigt.

Parametername	Werte					
	Bestes Resultat		Kleine Suchradien		Große Suchradien	
Besonderheit						
Äquidistante Zellabstände gefordert	Ja	Nein	Ja	Nein	Ja	Nein
AspectRatio	1,0	1,0	1,0	1,0	1,0	1,0
EstChannelWidthScale	10	10	10	10	10	10
PinsPerColumnScale	20	20	20	20	20	20
ChannelWidthScale	10	10	10	10	10	10
RowUtilization	0,95	0,95	0,95	0,95	0,95	0,95
MinRowUtilization	0,7	0,7	0,7	0,7	0,7	0,7
GapSearchLimitCell	50	50	10	10	100	100
GapSearchLimitFT	20	20	20	20	150	150
EqualCellDist	True	False	True	False	True	False
Bezeichnung :	p-p-row1	p-p-row2	p-p-row3	p-p-row4	p-p-row5	p-p-row6

C.2.2 Parametrisierung der hierarchischen Partitionierung

Die in den folgenden beiden Tabellen grau unterlegt dargestellte Parametrisierung wurde als Beispiel zur Evaluierung der kräftegesteuerten Platzierung anhand der Netzliste test500 verwendet. Von den 1331 untersuchten Parametrisierungen konnten 538 mit dem dkrouter verdrahtet werden.

Tabelle C- 19: Resultate für Parametrisierungen der hierarchischen Partitionierung gemäß Tabelle B- 5, für die Netzliste test500, geordnet nach durchschnittlicher Netzlänge.

Rang	Durchschnittliche Netzlänge	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	20	75	30	36.704	16.985	25.978	
2	2	5	30	41.886	17.087	26.299	
3	20	30	15	43.350	17.100	26.270	
4	15	10	3	48.292	17.127	26.247	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
535	100	10	30	72.385	22.805	45.114	
536	30	3	100	69.836	22.808	45.890	
537	100	20	15	71.435	23.057	44.867	
538	30	5	75	80.669	23.236	46.608	

Tabelle C- 20: Resultate für Parametrisierungen der hierarchischen Partitionierung gemäß Tabelle B- 5, für die Netzliste test500, geordnet nach benötigter Chipfläche.

Rang Chipfläche	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	50	75	30	45.296	17.207	25.676
2	20	75	30	36.704	16.985	25.978
3	2	5	75	43.781	17.145	25.995
4	5	20	15	52.691	17.275	26.079
⋮	⋮	⋮	⋮	⋮	⋮	⋮
535	30	2	3	62.055	22.708	45.282
536	20	3	100	69.836	22.808	45.890
537	20	3	75	53.524	22.269	46.055
538	30	5	75	80.669	23.236	46.608

Die in den folgenden drei Tabellen grau unterlegt dargestellte Parametrisierung wurde als Beispiel zur Evaluierung der kräftegesteuerten Platzierung anhand der Netzliste pk690 verwendet. Von den 1331 untersuchten Parametrisierungen konnten 1168 mit dem dkrouter verdrahtet werden.

Tabelle C- 21: Resultate für Parametrisierungen der hierarchischen Partitionierung gemäß Tabelle B- 5 für die Netzliste pk690, geordnet nach maximaler Netzlänge.

Rang	Maximale Netzlänge	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	15	20	30	15.115	2.547	29.157	
2	1	5	5	15.272	2.803	31.372	
3	2	10	10	15.272	2.803	31.372	
4	3	15	15	15.272	2.803	31.372	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
20	2	3	50	16.099	2.580	27.814	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
1163	100	100	5	51.814	5.601	41.475	
1164	1	30	50	52.859	4.406	44.100	
1165	2	2	3	54.625	5.801	45.072	
1166	10	10	15	54.625	5.801	45.072	
1167	20	20	30	54.625	5.801	45.072	
1168	50	50	75	54.625	5.801	45.072	

Anhang C Ergebnisse der Parametrisierung

Tabelle C- 22: Resultate für Parametrisierungen der hierarchischen Partitionierung gemäß Tabelle B- 5 für die Netzliste pk690, geordnet nach durchschnittlicher Netzlänge.

Rang	Durchschnittliche Netzlänge	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	10	1	5	20.381	2.297	30.682	
2	20	2	10	20.381	2.297	30.682	
3	30	3	15	20.381	2.297	30.682	
4	100	10	50	20.381	2.297	30.682	
5	15	2	100	18.956	2.355	28.978	
6	20	2	15	17.933	2.360	28.565	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
71	15	20	30	15.115	2.547	29.157	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
102	2	3	50	16.099	2.580	27.814	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
1164	50	50	100	34.451	5.722	44.873	
1165	2	2	3	54.625	5.801	45.072	
1166	10	10	15	54.625	5.801	45.072	
1167	20	20	30	54.625	5.801	45.072	
1168	50	50	75	54.625	5.801	45.072	

Tabelle C- 23: Resultate für Parametrisierungen der hierarchischen Partitionierung gemäß Tabelle B- 5 für die Netzliste pk690, geordnet nach benötigter Chipfläche.

Rang Chipfläche	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	10	2	5	20.554	2.466	24.697
2	100	20	50	20.554	2.466	24.697
3	1	20	50	22.695	3.213	26.034
4	100	20	75	22.472	2.576	26.071
5	15	3	10	20.012	2.584	26.320
6	75	15	50	20.012	2.584	26.320
⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	2	3	50	16.099	2.580	27.814
⋮	⋮	⋮	⋮	⋮	⋮	⋮
63	15	20	30	15.115	2.547	29.157
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1164	75	5	5	20.571	3.406	51.851
1165	50	15	75	34.893	4.691	57.718
1166	5	2	10	31.824	4.521	57.932
1167	50	20	100	31.824	4.521	57.932
1168	5	2	20	34.390	4.749	58.784

C.2.3 Parametrisierung der kräftegesteuerten Platzierung

Die Parametrisierung der kräftebasierten Platzierung ermöglicht die Definition mehrerer Abbruchkriterien. Zu den in den folgenden Tabellen gezeigten Iterationsgrenzen wurden für die anderen Abbruchkriterien folgende Wertebelegungen verwendet:

MaxMoveDistLimit=0, MaxMoveDistMaxBias=0,
MaxMoveDistMaxGradient=0, AvgMoveDistLimit=0,
AvgMoveDistMaxBias=0, AvgMoveDistMaxGradient=0,
AvgMoveDistShrinkLimit=0.980.

Durch diese Parametrisierung werden alle Abbruchkriterien mit Ausnahme der Iterationsanzahl und der Größenänderung deaktiviert. Eine gleichbleibende Verschiebungsentfernung der Zellen ist ein Indiz für Oszillation. Der gewählte Wert AvgMoveDistShrinkLimit=0.980 sichert einen Abbruch in diesem Falle.

Tabelle C- 24: Entwicklung der Platzierung für die Netzliste test500 über die Iterationen der kräftegesteuerten Platzierung. Die hier nicht aufgeführten Iterationsanzahlen aus Tabelle B- 6 führten zu für den dkrouter unverdrahtbaren Platzierungen.

Anzahl durchgeführter Iterationen	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
0	36.704	16.985	25.978
5	43.574	19.840	28.643
10	-	-	-
15	45.631	19.495	28.159
30	44.737	19.639	28.900
50	57.745	20.547	29.932
75	-	-	-
100	-	-	-
200	46.358	19.506	28.308
225	47.537	19.121	28.023
250	49.544	18.550	26.854
⋮	⋮	⋮	⋮
350	45.816	19.637	28.630
400	54.296	19.210	28.100
450	44.513	19.288	28.234
⋮	⋮	⋮	⋮

Die beiden in den folgenden Tabellen grau unterlegten Parametrisierungen wurden zur weiteren Verwendung des Plazierers ausgewählt. Es ist jeweils eine Parametrisierung mit und eine Verwendung der kräftegesteuerten Plazierungsmethode.

Tabelle C- 25: Vergleich der Resultate für verschiedene Iterationsanzahlen der kräftegesteuerten Plazierung (vergleiche Tabelle B- 6) anhand einer Stichprobe von hierarchisch partitionierten Plazierungen, für die Netzliste test500, geordnet nach maximaler Netzlänge.

Rang	Maximale Netzlänge	Iterationsanzahl	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	0	20	75	30	36.704	16.985	25.978	
2	0	5	20	75	38.202	17.316	26.169	
3	0	1	2	1	40.019	17.805	27.502	
:	:	:	:	:	:	:	:	
10	15	2	5	30	41.752	19.535	28.881	
:	:	:	:	:	:	:	:	
26	450	20	75	30	44.513	19.288	28.234	
27	500	20	75	30	44.513	19.288	28.234	
28	600	20	75	30	44.513	19.288	28.234	
29	700	20	75	30	44.513	19.288	28.234	
:	:	:	:	:	:	:	:	
73	250	20	75	30	49.544	18.550	26.854	
:	:	:	:	:	:	:	:	
430	5	50	100	50	62.910	20.181	30.838	

Tabelle C- 26: Vergleich der Resultate für verschiedene Iterationsanzahlen der kräftegesteuerten Platzierung (vergleiche Tabelle B- 6) anhand einer Stichprobe von hierarchisch partitionierten Platzierungen, für die Netzliste test500, geordnet nach durchschnittlicher Netzlänge.

Rang	Durchschnittliche Netzlänge	Iterationsanzahl	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	0	20	75	30	36.704	16.985	25.978	
2	0	2	5	30	41.886	17.087	26.299	
3	0	20	30	15	43.350	17.100	26.270	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
18	250	20	75	30	49.544	18.550	26.854	
19	20	5	20	15	48.890	18.655	28.062	
20	200	5	20	15	54.614	18.655	27.914	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
49	450	20	75	30	44.513	19.288	28.234	
50	500	20	75	30	44.513	19.288	28.234	
51	600	20	75	30	44.513	19.288	28.234	
52	700	20	75	30	44.513	19.288	28.234	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
430	50	20	75	30	57.745	20.547	29.932	

Tabelle C- 27: Vergleich der Resultate für verschiedene Iterationsanzahlen der kräftegesteuerten Platzierung (vergleiche Tabelle B- 6) anhand einer Stichprobe von hierarchisch partitionierten Platzierungen, für die Netzliste test500, geordnet nach benötigter Chipfläche.

Rang Chipfläche	Iterationsanzahl	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	0	50	75	30	45.296	17.207	25.676
2	0	20	75	30	36.704	16.985	25.978
3	0	2	5	75	43.781	17.145	25.995
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
13	250	20	75	30	49.544	18.550	26.854
14	0	1	2	1	40.019	17.805	27.502
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	30	50	75	30	47.470	18.825	27.526
21	75	2	5	75	47.392	19.206	27.754
22	75	5	20	15	44.262	19.344	27.802
23	400	5	20	15	45.944	18.909	27.856
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
46	300	50	75	30	44.446	19.353	28.211
47	450	20	75	30	44.513	19.288	28.234
48	500	20	75	30	44.513	19.288	28.234
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
104	5	50	100	50	62.910	20.181	30.838

Tabelle C- 28: Entwicklung der Platzierung für die Netzliste pk690 über die Iterationen der kräftegesteuerten Platzierung. Parametrisierungen aus Tabelle B- 6, die zu für den dkrouter unverdrahtbaren Platzierungen führten, sind mit Strichen gekennzeichnet.

Anzahl durchgeführter Iterationen	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
0	15.115	2.547	29.157
5	17.989	2.388	30.319
10	18.838	2.437	30.999
15	21.320	2.708	31.945
20	43.977	2.746	32.072
30	-	-	-
50	21.035	3.364	40.230
75	31.477	3.356	40.401
100	45.117	3.403	40.620
125	23.601	3.318	40.405
150	16.798	3.380	39.594
175	15.440	3.414	40.249
200	28.134	3.480	40.443
225	29.566	3.469	40.223
250	20.051	3.490	39.823
275	22.695	3.585	40.112
300	-	-	-
350	28.833	3.529	39.293
400	27.480	3.535	40.230
450	27.151	3.590	40.278
500	-	-	-
600	27.587	3.502	40.112
700	27.123	3.531	39.749

Die in den folgenden Tabellen grau hervorgehobene Parametrisierung stellt die ausgewählte Parametrisierung dar, die für den Plazierer verwendet wird, sofern die kräftegesteuerte Methode angewendet werden soll.

Tabelle C- 29: Vergleich der Resultate für verschiedene Iterationsanzahlen der kräftegesteuerten Plazierung (vergleiche Tabelle B- 6) anhand einer Stichprobe von hierarchisch partitionierten Plazierungen für die Netzliste pk690, geordnet nach maximaler Netzlänge.

Rang	Maximale Netzlänge	Iterationsanzahl	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	5	2	10	10	12.024	2.522	32.316	
2	5	3	15	15	12.024	2.522	32.316	
3	5	10	50	50	12.024	2.522	32.316	
4	5	15	75	75	12.024	2.522	32.316	
5	5	20	100	100	12.024	2.522	32.316	
6	150	1	20	50	14.903	3.710	32.104	
7	500	10	1	5	15.048	3.526	26.763	
8	500	20	2	10	15.048	3.526	26.763	
:	:	:	:	:	:	:	:	
19	175	15	20	30	15.440	3.414	40.249	
20	0	3	30	2	15.529	2.673	34.531	
21	250	3	30	2	15.607	3.637	47.664	
22	700	10	1	5	15.786	3.553	26.922	
23	700	20	2	10	15.786	3.553	26.922	
:	:	:	:	:	:	:	:	
425	20	100	20	50	50.478	3.301	33.728	
426	75	10	2	5	51.176	3.606	34.420	
427	75	100	20	50	51.176	3.606	34.420	
428	150	10	2	5	51.780	3.766	34.555	
429	150	100	20	50	51.780	3.766	34.555	
430	50	15	2	100	63.279	3.901	41.603	

Tabelle C- 30: Vergleich der Resultate für verschiedene Iterationsanzahlen der kräftegesteuerten Platzierung (vergleiche Tabelle B- 6) anhand einer Stichprobe von hierarchisch partitionierten Platzierungen für die Netzliste pk690, geordnet nach durchschnittlicher Netzlänge.

Rang	Durchschnittliche Netzlänge	Iterationsanzahl	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	0	10	1	5	20.381	2.297	30.682	
2	0	20	2	10	20.381	2.297	30.682	
3	0	30	3	15	20.381	2.297	30.682	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
16	10	100	20	50	15.255	2.449	26.652	
17	5	20	2	15	17.977	2.461	30.365	
18	0	10	2	5	20.554	2.466	24.697	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
173	500	20	2	10	15.048	3.526	26.763	
174	50	20	100	100	36.564	3.526	44.513	
175	500	30	3	15	15.048	3.526	26.763	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
429	250	15	2	100	34.010	4.124	41.273	
430	0	1	1	5	32.165	4.888	32.644	

Tabelle C- 31: Vergleich der Resultate für verschiedene Iterationsanzahlen der kräftegesteuerten Platzierung (vergleiche Tabelle B- 6) anhand einer Stichprobe von hierarchisch partitionierten Platzierungen für die Netzliste pk690, geordnet nach benötigter Chipfläche.

Rang Chipfläche	Iterationsanzahl	CutSizeWeight	ExtCutSizeWeight	TotalNetBalanceWeight	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	0	10	2	5	20.554	2.466	24.697
2	0	100	20	50	20.554	2.466	24.697
3	0	1	20	50	22.695	3.213	26.034
4	0	100	20	75	22.472	2.576	26.071
5	20	10	1	5	21.549	2.713	26.082
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
25	500	20	2	10	15.048	3.526	26.763
26	500	30	3	15	15.048	3.526	26.763
27	500	100	10	50	15.048	3.526	26.763
28	5	3	50	50	16.977	2.648	26.799
29	700	10	1	5	15.786	3.553	26.922
30	700	20	2	10	15.786	3.553	26.922
31	700	30	3	15	15.786	3.553	26.922
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
427	500	3	30	2	18.430	3.640	47.920
428	400	3	30	2	18.067	3.687	47.951
429	350	3	30	2	22.097	3.692	47.999
430	450	3	30	2	22.187	3.677	47.999

Anhang C Ergebnisse der Parametrisierung

Die Parametrisierungsdateien zu den in Tabelle C- 32 und Tabelle C- 33 angegebenen Parametrisierungen verwenden für die Erzeugung der Standardzellplatzierung die in Tabelle C- 17 und Tabelle C- 18 grau hervorgehobenen Wertebelegungen.

Tabelle C- 32: Für theoretische Netzlisten ermittelte Parametereinstellungen für den Platzierungsalgorithmus zur Generierung der Vorplatzierung.

Parametername	Werte	
Bevorzugtes Qualitätsmerkmal	Alle Kriterien	
Kräftegesteuerte Platzierung angewendet	Nein	Ja
CutSizeWeight	20	20
ExtCutSizeWeight	75	75
TotalNetBalanceWeight	30	30
MaxLeafSize	1	1
MaxCentreIterations	0	250
MinCentreIterations	0	250
MaxMoveDistLimit ¹⁾	0	0
MaxMoveDistMaxBias ¹⁾	0	0
MaxMoveDistMaxGradient ¹⁾	0	0
AvgMoveDistLimit ¹⁾	0	0
AvgMoveDistMaxBias ¹⁾	0	0
AvgMoveDistMaxGradient ¹⁾	0	0
AvgMoveDistShrinkLimit	0.980	0.980
Bezeichnung :	t-p-all-part	t-p-all-force

Anmerkung: 1) Ein Parameterwert von Null deaktiviert diese Abbruchbedingung

Tabelle C- 33: Für praktische Netzlisten ermittelte Parametereinstellungen für den Plazierungsalgorithmus zur Generierung der Vorplazierung.

Parametername	Werte			
Kräftegesteuerte Plazierung angewendet	Nein			Ja
Bevorzugtes Qualitätsmerkmal	Alle Kriterien	Maximale Netzlänge	Chipgröße	Alle Kriterien
CutSizeWeight	2	15	100	20
ExtCutSizeWeight	3	20	20	2
TotalNetBalanceWeight	50	30	50	10
MaxLeafSize	1	1	1	1
MaxCentrelterations	0	0	0	500
MinCentrelterations	0	0	0	500
MaxMoveDistLimit ¹⁾	0	0	0	0
MaxMoveDistMaxBias ¹⁾	0	0	0	0
MaxMoveDistMaxGradient ¹⁾	0	0	0	0
AvgMoveDistLimit ¹⁾	0	0	0	0
AvgMoveDistMaxBias ¹⁾	0	0	0	0
AvgMoveDistMaxGradient ¹⁾	0	0	0	0
AvgMoveDistShrinkLimit	0.980	0.980	0.980	0.980
Bezeichnung :	p-p-all-part	p-p-max-part	p-p-size-part	p-p-all-force

Anmerkung: 1) Ein Parameterwert von Null deaktiviert diese Abbruchbedingung

C.3 Parametrisierung der Kombination von Plazierer und Verdrahter

Tabelle C- 34: Vergleich der Resultate für verschiedene Parametrisierungen der gemeinsamen Verwendung von Plazierer und Verdrahter von prob anhand der Netzliste test500, geordnet nach maximaler Netzlänge. Erläuterung der Bezeichnungen der Parametrisierungen gemäß Tabelle 1 in Abschnitt 5.3.

Rang maximale Netzlänge	Plazierungsmethode für die Vorplazierung	Erzeugen der Standardzellplazierung	Plazierungsmodifikation durch den Verdrahter	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	part	row2	no	18.676	7.215	105.462
2	part	row2	yes	18.682	7.235	105.462
3	part	row3	yes	19.079	7.395	105.462
4	part	row1	no	19.084	7.323	105.462
⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	part	row5	move	20.851	7.400	105.462
19	force	row4	no	21.572	8.175	109.938
20	force	row5	no	21.661	8.154	108.361
21	force	row5	yes	21.661	8.154	108.361
22	force	row3	yes	22.053	8.182	109.938
⋮	⋮	⋮	⋮	⋮	⋮	⋮
36	force	row2	move	23.623	8.326	109.174

Tabelle C- 35: Vergleich der Resultate für verschiedene Parametrisierungen der gemeinsamen Verwendung von Plazierer und Verdrahter von prob anhand der Netzliste test500, geordnet nach durchschnittlicher Netzlänge. Erläuterung der Bezeichnungen der Parametrisierungen gemäß Tabelle 1 in Kapitel 5.3.

Rang	durchschnittliche Netzlänge	Plazierungsmethode für die Vorplatzierung	Erzeugen der Standardzellplatzierung	Plazierungsmodifikation durch den Verdrahter	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	part	row2	no	18.676	7.215	105.462	
2	part	row2	yes	18.682	7.235	105.462	
3	part	row6	yes	20.264	7.283	105.462	
4	part	row5	no	20.370	7.302	105.462	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
18	part	row2	move	19.319	7.514	105.462	
19	force	row1	no	22.500	8.075	109.174	
20	force	row1	yes	22.533	8.090	109.174	
21	force	row5	no	21.661	8.154	108.361	
22	force	row5	yes	21.661	8.154	108.361	
23	force	row4	no	21.572	8.175	109.938	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
35	force	row5	move	22.695	8.389	108.361	
36	force	row6	move	23.495	8.481	108.361	

Tabelle C- 36: Vergleich der Resultate für verschiedene Parametrisierungen der gemeinsamen Verwendung von Plazierer und Verdrahter von prob anhand der Netzliste test500, geordnet nach Chipgröße. Erläuterung der Bezeichnungen der Parametrisierungen gemäß Tabelle 1 in Kapitel 5.3.

Rang	Chipgröße	Plazierungsmethode für die Vorplazierung	Erzeugen der Standardzellplazierung	Plazierungsmodifikation durch den Verdrahter	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	part	row1	move	19.185	7.364	105.462	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
1	part	row2	no	18.676	7.215	105.462	
1	part	row6	yes	20.264	7.283	105.462	
19	force	row5	move	22.695	8.389	108.361	
19	force	row5	no	21.661	8.154	108.361	
19	force	row5	yes	21.661	8.154	108.361	
19	force	row6	move	23.495	8.481	108.361	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
36	force	row4	yes	22.338	8.190	109.938	

Tabelle C- 37: Vergleich der Resultate für verschiedene Parametrisierungen der gemeinsamen Verwendung von Plazierer und Verdrahter von prob anhand der Netzliste pk690, geordnet nach maximaler Netzlänge. Erläuterung der Bezeichnungen der Parametrisierungen gemäß Tabelle 1 in Kapitel 5.3.

Rang maximale Netzlänge	Plazierungsmethode für die Vorplatzierung	Erzeugen der Standardzellplatzierung	Plazierungsmodifikation durch den Verdrahter	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	part	row5	move	10.817	0.973	65.540
2	part	row1	yes	10.923	0.991	65.540
3	part	row2	yes	10.928	1.003	65.540
4	part	row1	no	10.940	0.993	65.540
5	part	row3	no	10.951	0.990	65.540
6	part	row3	yes	10.979	0.984	65.540
:	:	:	:	:	:	:
18	part	row6	no	11.789	1.022	65.540
19	force	row4	no	18.855	1.683	143.365
20	force	row6	no	19.403	1.707	143.365
21	force	row3	yes	19.548	1.676	143.365
:	:	:	:	:	:	:
36	force	row2	yes	21.208	1.729	146.599

Tabelle C- 38: Vergleich der Resultate für verschiedene Parametrisierungen der gemeinsamen Verwendung von Plazierer und Verdrahter von prob anhand der Netzliste pk690, geordnet nach durchschnittlicher Netzlänge. Erläuterung der Bezeichnungen der Parametrisierungen gemäß Tabelle 1 in Kapitel 5.3.

Rang	durchschnittliche Netzlänge	Plazierungsmethode für die Vorplazierung	Erzeugen der Standardzellplazierung	Plazierungsmodifikation durch den Verdrahter	Maximale Netzlänge	Durchschnittliche Netzlänge	Chipfläche
1	part	row5	move	10.817	0.973	65.540	
2	part	row1	move	11.236	0.974	65.540	
3	part	row3	move	11.141	0.976	65.540	
4	part	row3	yes	10.979	0.984	65.540	
5	part	row3	no	10.951	0.990	65.540	
:	:	:	:	:	:	:	
18	part	row6	move	11.471	1.038	65.540	
19	force	row3	move	20.582	1.663	143.365	
20	force	row4	move	19.755	1.673	143.365	
21	force	row3	yes	19.548	1.676	143.365	
:	:	:	:	:	:	:	
36	force	row2	yes	21.208	1.729	146.599	

D. ERGEBNISSE DER KORRIGIERTEN VERSION VON PROB

Dieser Anhang enthält Tabellen mit Ergebnissen der einer Version von prob, die den in Kapitel 6.3 erwähnten Implementierungsfehler in der Kompaktifizierung nicht mehr aufweist. Die Tabelleneinträge für Parameterwerte erfolgen in der für den jeweiligen Parameter in Anhang A definierten Einheit. Angaben zu Längen verstehen sich in mm, Flächen sind in mm² angegeben. Alle Zeitangaben verstehen sich in Sekunden.

D.1 Verdrahterergebnisse

Tabelle D- 1: Ergebnisse des in prob implementierten Verdrahters für verschiedene Plazierer mit der Parametrisierung ohne jegliche Platzierungsmodifikation. Die Längenangaben verstehen sich in mm, die Flächenangaben in mm².

Plazierer	Great Deluge			Fuzzylogik Plazierer			Genetischer Plazierer		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
test100	4.310	1.837	3.217	3.622	1.605	3.033	4.584	1.784	3.189
test200	7.138	3.755	7.440	7.586	3.413	7.059	6.937	3.331	7.012
test500	17.977	8.647	30.347	17.396	7.585	28.169	18.564	8.500	30.011
test1000	45.441	19.593	69.333	35.295	16.261	60.145	46.889	18.880	66.950
test3000	128.41	50.78	1732.32	116.59	35.63	1302.13	124.78	48.83	1673.54
pk246	1.912	0.458	3.561	1.895	0.215	3.045	2.638	0.547	3.825
pk276	-	-	-	1.179	0.256	3.342	3.281	0.902	6.036
pk690	6.445	2.039	26.391	4.522	0.592	23.543	6.339	1.970	25.008

Tabelle D- 2: Ergebnisse des in prob implementierten Verdrahters mit Zulassen von Zellverschiebungen um neue Feedthroughpositionen zu erhalten, jedoch ohne weitere Platzierungsmodifikationen für verschiedene Plazierer. Die Längenangaben verstehen sich in mm, die Flächenangaben in mm².

Plazierer	Great Deluge			Fuzzylogik Plazierer			Genetischer Plazierer		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
test100	4.310	1.837	3.217	3.622	1.605	3.033	4.623	1.785	3.189
test200	7.166	3.743	7.426	7.586	3.413	7.059	6.909	3.324	6.998
test500	17.838	8.660	30.372	17.525	7.580	28.142	18.961	8.523	30.011
test1000	45.413	19.606	69.367	35.278	16.257	60.111	46.906	18.884	66.983
test3000	126.23	50.64	1727.78	116.18	35.69	1303.63	129.25	49.00	1677.68
pk246	1.917	0.466	3.561	1.906	0.219	3.045	2.683	0.553	3.865
pk276	-	-	-	1.179	0.254	3.342	3.360	0.901	6.036
pk690	6.456	2.068	26.391	4.500	0.595	23.543	6.524	1.953	25.008

Tabelle D- 3: Ergebnisse des in prob implementierten Verdrahters mit Modifikationen an der Platzierung für verschiedene Plazierer. Die Längenangaben verstehen sich in mm, die Flächenangaben in mm².

Plazierer	Great Deluge			Fuzzylogik Plazierer			Genetischer Plazierer		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
test100	4.908	1.943	3.299	3.885	1.820	3.290	4.645	1.951	3.362
test200	7.703	4.002	7.711	7.697	3.699	7.476	7.340	3.643	7.319
test500	17.888	8.847	30.702	16.558	7.743	28.482	20.756	8.895	30.984
test1000	45.083	19.536	68.796	39.253	16.810	61.785	47.627	19.489	68.338
test3000	127.92	50.94	1735.35	114.25	36.23	1321.43	125.21	49.24	1684.46
pk246	2.202	0.487	3.561	2.476	0.266	3.045	2.884	0.594	4.039
pk276	-	-	-	1.342	0.267	3.342	3.852	0.913	6.036
Pk690	6.725	2.059	26.391	4.489	0.620	23.543	6.306	1.920	25.008

D.2 Ergebnisse des Gesamtverfahrens

Tabelle D- 4: Ergebnisse des kompletten in prob implementierten Verfahrens mit Platzierung und Verdrahtung in einem Programmaufruf. Die Längenangaben verstehen sich in mm, die Flächenangaben in mm².

Plazierungs- methode	Nur hierarchische Partitionierung			Hierarchische Partitionierung und kräftegesteuerte Methode		
Kriterium	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche	Maximale Netzlänge	Durchschn. Netzlänge	Chipfläche
test100	3.723	1.584	4.009	4.198	1.618	4.087
test200	17.759	7.076	36.096	8.435	3.596	9.861
test500	8.066	3.291	9.147	22.086	8.172	41.185
test1000	47.772	17.580	88.900	45.788	17.783	91.020
test3000	217.121	40.258	2059.365	198.445	42.358	2203.069
pk246	1.459	0.262	3.500	1.392	0.311	3.696
pk276	1.699	0.338	3.517	2.052	0.469	6.265
pk690	4.768	0.730	25.745	6.999	0.926	21.951

Die in Tabelle D- 4 gezeigten Ergebnisse zeigen, daß die in Kapitel 6.3 beschriebene extreme Verdrößerung der Chipfläche nach Behebung des Fehlers in der Kompaktifizierung, nicht mehr auftritt. Die immernoch relativ große Chipfläche resultiert aus der Bevorzugung kurzer Netzlisten zu Ungunsten einer gleichmäßigen Kanalauslastung.

Anhang D Ergebnisse der korrigierten Version von prob

Die folgende Abbildung 66 zeigt ein mit der korrigierten Version von prob erzeugtes Layout für die Netzliste test500. Es wurde mit der gleichen Parametrisierung, wie das in Abbildung 58 gezeigte Layout erstellt.

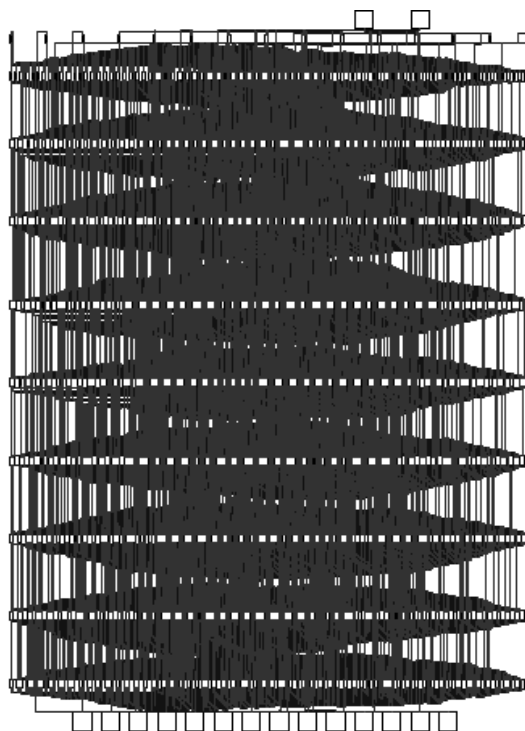


Abbildung 66: Das mit der korrigierten Version von prob erstellte Layout der Netzliste test500. Die Chipfläche ist nicht größer, als für das Layout benötigt.

E. INHALT DER BEIGEFÜGTEN CD

Auf der dieser Arbeit beigefügten CD finden sich folgende Inhalte:

`\index.html` Der Inhalt als Hypertext

`\doc` Diese Arbeit in Form eines PDF-Dokumentes

`\sw` Für die Evaluierung verwendete Software und Daten

`\bin` Ausführbare Programme für Linux und MS-DOS (ESPRO-Module)

`\data` Daten der Evaluierung

`\old-layout` Chiplayouts, erzeugt von anderen ESPRO-Plazierern

`\p690` Parameterevaluation mit der Netzliste `pk690`

`\param` Parameterdateien für `prob` mit den ermittelten Parametrisierungen.

`\placement` Plazierungen, erzeugt von anderen ESPRO-Plazierern

`\placer-results` Ergebnisse des implementierten Plazierers, Verdrahtung mit `dkrouter`

`\prob-corrected` Ergebnisse des Gesamtverfahrens, Plazierer und Verdrahter von `prob`. Erzeugt mit korrigierter Version von `prob`, ohne Implementierungsfehler in der Kompaktifizierung.

`\prob-results` Ergebnisse des Gesamtverfahrens, Plazierer und Verdrahter von `prob`

`\router-corrected` Ergebnisse des Verdrahters mit anderen ESPRO-Plazierern. Erzeugt mit korrigierter Version von `prob`, ohne Implementierungsfehler in der Kompaktifizierung.

`\routed-results` Ergebnisse des Verdrahters mit anderen ESPRO-Plazierern

`\t500` Parameterevaluation mit der Netzliste `test500`

`\src` Der Quellcode des implementierten Prototyps `prob`. Zu bekannten Fehlern siehe die Datei `knownbugs.txt`

`\ref\references.html` Das Literaturverzeichnis mit Links zu den Internetquellen

Die in vielen Verzeichnissen existierende Datei `dirinfo.txt` enthält weitere Informationen zu den Daten, die sich in dem jeweiligen Verzeichnis befinden.

F. LITERATURVERZEICHNIS

- [Alp98] Charles J. Alpert: "The ISPD98 Circuit Benchmark Suite", Proceedings International Symposium on Physical Design, 1998.
- [Ber89] Claude Berge: "Hypergraphs", North Holland, Amsterdam, 1989.
- [Bre77a] M. A. Breuer: "A class of min-cut placement algorithms", Proceedings Design Automation Conference, 1977.
- [Bre77b] M. A. Breuer: "Min-cut placement", J. B. Design Automation and Fault-Tolerant Computing", 1977.
- [Brü93] Rainer Brück: "Entwurfswerkzeuge für VLSI-Layout: Methoden und Algorithmen für den rechnergestützten Entwurf von VLSI-Layout", Carl Hanser Verlag, München, 1993
- [CKM00] Andrew E. Caldwell, Andrew B. Kahng und Igor L. Markov: „Can Recursive Bisection Alone Produce Routable Placements ?“, ACM/IEEE Design Automation Conference 2000.
- [CKM00a] Andrew Caldwell, Andrew B. Kahng und Igor Markov: „VLSI CAD Bookshelf“, <http://vlsicad.eecs.umich.edu/BK>, 2000 (Juni 2001).
- [CoL84] E. G. Coffman Jr. and M. A. Langston: "A Performance Guarantee for the Greedy Set-Partitioning Algorithm", Washington State University, Pullman, Washington, 1984.
- [Des87] M. Desochers: "A note on the partitioning shortest path algorithm", Centre for Mathematics and Computer Science, Stichting Mathematisch Centrum, Amsterdam, 1987.
- [Die89] Rainer Dietz: "Ein dynamisches Modell für die globale Verdrahtung anwendungsspezifischer integrierter Schaltungen", VDI-Verlag, Düsseldorf, 1989.
- [Dij59] E. W. Dijkstra: „A note on two problems in connexion with graphs“, Numerische Mathematik, 1:269-271, 1959.
- [DKS98] Wilm Donath, Prabhakar Kudva, Leon Stok, Paul Villarubia, Lakshmi Reddy, Andrew Sullivan, Kanad Chakraborty: "Transformational Placement and Synthesis", Proceedings of the Design, Automation and Test in Europe, 1998 (DATE '98)
- [ESS91] European Silicon Structures Ltd.: „Solo 1400 Release 3.1 Reference Manual“, Berkshire, 1991.
- [GaJ79] Michael R. Garey und David S. Johnson: „Computers and Intractability. A Guide to the Theory of NP-Completeness“, W. H. Freeman and Company, New York, 1979.
- [Ger95] M. Gerloff: „Clusterverfahren im VLSI-Standardzellenlayout“, Diplomarbeit, Universität Hamburg, 1995.
- [Ger99] Sabih H. Gerez: „Algorithms for VLSI Design Automation“ John Wiley & Sons Ltd., Chichester, 1999.
- [Gil93] Wolfgang K. Giloi: „Rechnerarchitektur“, Zweite Auflage, Springer-Verlag, Berlin, 1993.

- [Hag95] Klaus ten Hagen: „Abstrakte Modellierung digitaler Schaltungen“, Springer-Verlag, Berlin, 1995.
- [Het98] Asmus Hetzel: “A Sequential Detailed Router for Huge Grid Graphs”, Proceedings of the 1998 Design Automation and Test in Europe, 1998 (DATE '98).
- [HuL99] Sung-Woo Hur und John Lillis: “Relaxation and Clustering in a Local Search Framework: Application to Linear Placement”, ACM/IEEE Design Automation Conference 1999.
- [Int00] Pressemitteilung der Intel Corporation vom 7. November 2000: “Intel Completes 0.13 Micron Process Technology Development“, <http://www.intel.com/pressroom/archive/releases/cn110700.htm> (Juni 2001)
- [Int01] Intel Corporation: „Moore’s Law“, <http://www.intel.com/research/silicon/mooreslaw.htm> (Juni 2001)
- [Jen97] Heiko Jenke: „Implementierung und Evaluierung von innovativen Heuristiken zum Placement & Routing“, Diplomarbeit, Universität Hamburg, 1997.
- [Jun94] Dieter Jungnickel: “Graphen, Netzwerke und Algorithmen”, BI-Wissenschaftsverlag, Mannheim, 1994.
- [Kah95] Olav Kahlbaum: „Implementierung und Evaluierung von Standard-Heuristiken zum Placement & Routing“, Studienarbeit, Universität Hamburg, Sept. 1995
- [Kah97]. Olav Kahlbaum: „Ein Tool zur Verifikation von Standardzelllayouts“, Diplomarbeit, Universität Hamburg, 1997.
- [Kar01] George Karypis: „Publications Related to Graph Partitioning“, <http://www-users.itlabs.umn.edu/~karypis/publications/partitioning.html> (Juni 2001)
- [Kas95] O. Kastek: „Nachoptimierung von Standardzellentwürfen“, Diplomarbeit, Universität Hamburg, 1995.
- [Kla94] D. Klawan: „ESPRO - Ein Programmpaket für Placement & Routing beim Standardzellen-Entwurf“, Diplomarbeit, Universität Hamburg, 1994.
- [LaW00] Minghorng Lai and D. F. Wong: „Maze Routing with Buffer Insertion and Wiresizing“, ACM/IEEE Design Automation Conference 2000.
- [Len90] Thomas Lengauer: “Combinatorial Algorithms for Integrated Circuit Layout”, Teubner, Stuttgart, 1990.
- [LLL96] Jianmin Li, John Lillis, Lung-Tien Liu und Chung-Kuan Cheng: „New Spectral Linear Placement and Clustering Approach“, ACM/IEEE Design Automation Conference 1996
- [Mar93] Peter Marwedel: „Synthese und Simulation von VLSI-Systemen“, Carl Hanser Verlag München Wien, 1993.
- [Oht86] T. Ohtsuki: „Layout Design and Verification“, Advances in CAD for VLSI, Volume 4, North-Holland, 1986.
- [RaB91] M. Raith und M. Bartholomeus: „A new hypergraph based rip-up and reroute strategy“, ACM/IEEE Design Automation Conference 1991
- [Rau00] Reinhard Rauscher: ” Results in Combining Innovative Placement Heuristics“, Universität Hamburg, Noch unveröffentlicht.

- [RKB96] Reinhard Rauscher, Dieter Klawan und Hans-Jürgen Bandelt: „Results given by a New Evaluation System for Placement and Routing Heuristics“, 22nd EUROMICRO Conference '96
- [SENR98] Baruch Schieber, Guy Even, Joseph Naor and Satish Rao: “Fast Approximate Graph Partitioning Algorithms”, IBM Thomas J. Watson Research Center, Yorktown Heights, NewYork, 1998.
- [She89] A. Sherman: „VLSI Placement and Routing: The PI Project“, Springer Verlag, Berlin, 1989
- [She95] Naveed Sherwani: “Algorithms for VLSI Physical Design Automation”, Kluwer Academic Publishers, Dordrecht, The Netherlands, Second Edition, 1995.
- [Str01] Ludger Struewing: “Der Scan-Line Algorithmus”, olli.informatik.uni-oldenburg.de/Grafiti3/grafiti/flow8/page9.htm, 2001 (Oktober '01)
- [TAB00] S. Tyagi, M. Alavi , R. Bigwood, T. Bramblett, J. Brandenburg, W. Chen, B. Crew, M. Hussein, P. Jacob, C. Kenyon, C. Lo, B. McIntyre, Z. Ma, P. Moon, P. Nguyen, L. Rumaner, R. Schweinfurth, S. Sivakumar, M. Stettler, S. Thompson, B. Tufts, J. Xu, S. Yang und M. Bohr: “A 130 nm Generation Logic Technology Featuring 70nm Transistors, Dual Vt Transistors and 6 layers of Cu Interconnects”, IEEE International Electron Devices Meeting (IEDM), December 2000, <http://www.intel.com/research/silicon/ieee/ieee7.htm> (Juni 2001)
- [Téb93] Attila T. Téby: “Automatische Layout-Methoden für transparente Standardzellen”, VDI Fortschrittberichte, VDI-Verlag, 1993.
- [Tri87] Stephen M. Trimberger: “An introduction to CAD for VLSI”, Kluwer Press, Boston, 1987.
- [Vij88] Gopalakrishnan Vijayan: “Generalization of Min-Cut Partitioning to Tree Structures and ist Applications”, IBM Thomas J. Watson Research Center, Yorktown Heights, NewYork, 1988.
- [Vyg98] Jens Vygen: „Algorithms for Detailed Placement of Standard Cells“, Proceedings of the 1998 Design Automation and Test in Europe, 1998 (DATE '98)
- [WWW84] Y. F. Wu, P. Widmayer, C. Wong: “A Faster Approximation Algorithm for the Steiner Problem in Graphs”, IBM Thomas J. Watson Research Center, Yorktown Heights, NewYork, 1984.
- [Zob94] George Winston Zobrist: „Routing, Placement and Partitioning ...“, Norwood: Ablex, 1994
- [ZWL99] Hai Zhou, D.F. Wong, I-Min Liu und Adnan Aziz: “Simultaneous Routing and Buffer Insertion with Restrictions on Buffer Locations”, ACM/IEEE Design Automation Conference 1999.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, die vorliegende Arbeit selbständig verfaßt und mich keiner anderen, als den angegebenen Quellen und Hilfsmittel bedient zu haben.

Hamburg, den November 2001

(Ole Blaurock)