

Pinzuordnungs-Algorithmen für hochkomplexe Area-Array-Komponenten

Tilo Meister, Jens Lienig, Institut für Feinwerktechnik und Elektronik-Design (IFTE), TU Dresden, Dresden
Gisbert Thomke, IBM Deutschland Entwicklung GmbH, Böblingen

Kurzfassung

Die Verdrahtbarkeit und das Verdrachtungsergebnis von Leiterplatten, Multi-Chip-Modulen und Single-Chip-Modulen, die mit Area-Array-Komponenten bestückt sind, hängen von der gewählten Pinbelegung dieser Komponenten ab. Im Gegensatz zum Pinzuordnungsproblem im VLSI-Entwurf, bei dem eine Zuordnung von einer Vielzahl (stellenweise Millionen) Komponenten mit jeweils wenigen Pins erfolgt, zeichnen sich die Area-Array-Komponenten auf den oben genannten Verdrachtungsträgern durch Tausende von Pins aus, deren Signalzuordnung zu optimieren ist. Dieses Pinzuordnungsproblem wurde bisher wenig beachtet und ist demzufolge weitestgehend noch durch eine manuelle Vorgehensweise gekennzeichnet.

Diese Arbeit stellt drei neue Pinzuordnungs-Algorithmen für die Signalbelegung bei Area-Array-Komponenten vor, welche die Zuordnung von Signalen zu physikalischen Pins automatisch erstellen und optimieren. Die Anwendung in realen Entwurfsumgebungen bestätigt ihre enorme Zeiteffizienz im Gegensatz zur bisher erfolgten manuellen Belegungsermittlung, welche im Durchschnitt einen Monat in Anspruch nimmt.

1 Einleitung

Die Komplexität elektronischer Baugruppen und damit die Anzahl der elektrischen Signale und die Anforderungen an diese nehmen ständig zu. Dadurch werden die Entwicklung und Produktion der Baugruppen immer aufwendiger. Der Entwicklungsprozess muss in vielen Einzelschritten, an denen mehrere Entwickler gleichzeitig arbeiten, durchgeführt werden. Zur Koordination der parallel arbeitenden Entwicklerteams ist das gesamte elektronische Gerät in kleinere, bewältigbare Teile zu zerlegen. Ebenso sind die Schnittstellen zwischen den Teilsystemen und damit den verschiedenen Entwicklungsteams frühzeitig im Entwicklungsprozess festzulegen. Fast immer werden die elektronischen Teilsysteme so ausgewählt, dass ihre Trennstellen mit elektrischen Anschlüssen, wie z.B. Steckern oder Pins von Bauelementen, zusammenfallen. Dadurch werden die Stecker und Pins sowohl zu den Schnittstellen zwischen den Teilsystemen des elektronischen Gerätes als auch zwischen den Entwicklern.

Der reibungslose Ablauf der parallelen Entwicklungsprozesse erfordert eine exakte, vollständige Definition der Schnittstellen, die im Laufe der Entwicklung der elektronischen Baugruppe möglichst nicht zu verändern sind. Jede Modifikation dieser Schnittstellendefinition beträfe viele Entwickler und würde großen Arbeits- und Zeitaufwand nach sich ziehen. In den meisten Fällen ist dazu sehr frühzeitig festzulegen, welches Signal auf welchem physikalischen Pin

(elektrischen Anschluss) eines Bauelementes übertragen werden soll. Diese Zuordnung von Signalen zu Pins wird als *Pinzuordnung* (engl. *pin assignment*) bezeichnet. Dabei kann die Anzahl der Pins eines einzigen Bauelementes einige Tausend betragen.

Die Zuordnung von Signalen zu Pins kann nicht willkürlich festgelegt werden, denn sie beeinflusst maßgeblich den Verdrahtungsaufwand und entscheidet auch darüber, ob die Baugruppe überhaupt verdrahtbar ist. Bei der stetig wachsenden Pinanzahl ist die bisher übliche manuelle Pinzuordnung nicht mehr zu bewältigen und ist so automatisch zu erzeugen und zu optimieren.

Dieser Beitrag stellt drei neue Algorithmen für die automatische Erstellung optimierter Pinzuordnungen vor, welche sich unter realen Entwurfsbedingungen bewährt haben und zur Ersetzung der bisherigen manuellen Pinzuordnung führen. Einsatz finden diese Algorithmen beim Entwurf von Leiterplatten, Multi-Chip-Modulen und Single-Chip-Modulen. In dieser Arbeit sind stets alle diese Entwurfsaufgaben gemeint, auch wenn im Folgenden nur Leiterplatten explizit genannt werden.

Der Beitrag ist wie folgt gegliedert. Im anschließenden Abschnitt 2 werden die bisherigen Arbeiten auf diesem Gebiet vorgestellt. Der Abschnitt 3 beschreibt das Pinzuordnungsproblem. Abschnitt 4 stellt die neuen Pinzuordnungs-Algorithmen vor. Abschnitt 5 präsentiert schließlich die mit den neuen Algorithmen erzielten Ergebnisse für industrielle Testdesigns.

2 Bisherige Arbeiten

Auf dem Gebiet der Pinzuordnung für VLSI-Design existieren eine Vielzahl von Veröffentlichungen. So beschreiben [1] [2] ein typisches VLSI-Pinzuordnungs-Problem, das Channel Pin Assignment. Ein weiteres Pinzuordnungsproblem beim VLSI-Design ist die Optimierung der Pinzuordnung von Standardzellen- und Makrozellen-Designs [3] [4]. Auch für FPGA-Designs wurden Untersuchungen zur Pinzuordnung betrieben [5] [6]. Alle diese Arbeiten konnten die Qualität eines Designs, oft durch die Gesamtverbindungslänge ausgedrückt, durch Optimierung der Pinzuordnung verbessern.

Das Pinzuordnungs-Problem auf Leiterplattenebene unterscheidet sich jedoch wesentlich von der Problemstellung bei VLSI- und FPGA-Designs. Beim VLSI-Design muss eine Pinzuordnung für bis zu Millionen von Komponenten mit jeweils wenigen Pins gefunden werden. Im Gegensatz dazu ist beim Erstellen einer Pinzuordnung für Komponenten auf Leiterplatten eine Pinzuordnung für relativ wenige Komponenten (einige Hundert) mit jeweils sehr vielen Pins (bis zu mehreren Tausend) gesucht.

In früheren Arbeiten wurden für das Pinzuordnungsproblem auf Leiterplatten Heuristiken entwickelt, die basierend auf der Topologie von Pins eine Pinzuordnung ermitteln, die eine möglichst geringe Anzahl Signalkreuzungen haben und somit eine gute Verdrahtbarkeit aufweisen [4] [7] [8] [9] [10]. Die vorgestellten Methoden sind jedoch nur für Bauelemente mit jeweils wenigen Pins geeignet.

Aktuelle Veröffentlichungen beschränken sich auf die Pinzuordnung von VLSI-Designs. Eine Ausnahme stellt die in [11] beschriebene Pinzuordnung für Single Chip Module dar, welche jedoch nur speziell für vollkommen symmetrische Single Chip Module geeignet ist und sich für das hier behandelte Problem der hochzahligen Pinanschlüsse nicht eignet.

3 Pinzuordnungs-Problem

Pins sind die elektrischen Anschlussstellen zwischen den Komponenten einer elektronischen Baugruppe. Zum einen dienen Pins als definierte Schnittstelle zwischen den einzelnen Komponenten einer Baugruppe. Zum anderen sind Pins Übergangsstellen zwischen unterschiedlichen Herstellungstechnologien einer Baugruppe.

Im Allgemeinen versteht man unter Pinzuordnung die Zuweisung von logischen Signalen bzw. Netzen zu fest platzierten Pins. Im Rahmen dieser Veröffentlichung wird zur einfacheren Veranschaulichung eine andere, jedoch äquivalente Betrachtungsweise gewählt. Bei dieser gehört jedes Pin zu einem bestimmten Netz, ohne eine feste Position zu besitzen. Damit kann jedes Pin eine Position aus einer vorgegebenen Menge von Pinpositionen einnehmen.

Das Pinzuordnungs-Problem besteht also darin, für jedes Pin die gültige Position zu finden, welche design-spezifische Optimierungskriterien bestmöglich erfüllt. Das Optimierungskriterium im Rahmen dieser

Arbeit ist die Verdrahtbarkeit (engl. routability) der Schaltung, ausgedrückt durch die Anzahl der bei der Verdrahtung zu erwartenden Signalkreuzungen.

Aufgrund der Komplexität des Pinzuordnungs-Problems liegen den hier vorgestellten Algorithmen folgende entwurfsmethodischen Randbedingungen zugrunde:

Eingabedaten:

Als Eingangsinformation werden dem Algorithmus zwei unabhängige Mengen von gültigen Pinpositionen mit jeweils p Pinpositionen und p Zweipunkt-Netze übergeben.

Ausgabedaten:

Das Ergebnis des Pinzuordnungs-Algorithmus ist die Pinzuordnung für die p Zweipunkt-Netze. Dabei wird einem Pin jedes Netzes eine Position in der ersten Menge von Pinpositionen und dem zweiten Pin eine Position in der zweiten Menge zugeordnet. Jede der Pinpositionen nimmt genau ein Pin auf. Die Pins aller Netze werden in jeder Hinsicht als gleichberechtigt behandelt, d.h. es erfolgt keine Netzpriorisierung.

Optimierungsziel:

Ziel des Pinzuordnungs-Algorithmus ist es, eine Pinzuordnung zu finden, die eine bestmögliche Verdrahtbarkeit gewährleistet, wobei diese durch die bei der Verdrahtung zu erwartende Anzahl von Signalkreuzungen ausgedrückt wird.

Aus diesen Festlegungen folgt, dass der Pinzuordnungs-Algorithmus eine eindeutige Zuordnung zwischen den Pinpositionen der beiden ihm übergebenen unabhängigen Mengen ermittelt. Die zwei einander zugeordneten Pinpositionen werden dann jeweils für die zwei Pins eines Netzes verwendet. Die Zuordnung eines konkreten Netzes zu einem bestimmten Pinpaar ist für die hier vorgestellten Algorithmen unerheblich, da keine Netzpriorisierung erfolgt, also alle Netze gleich behandelt werden.

Bild 1 zeigt beispielhaft zwei Pinzuordnungs-Proble-

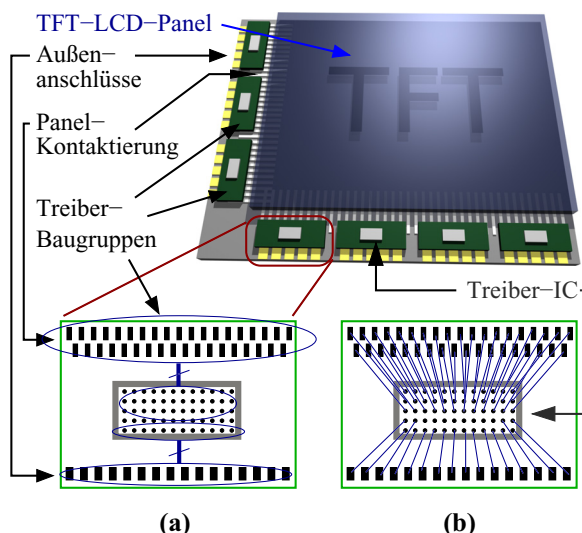


Bild 1. Vereinfachte Darstellung des Pinzuordnungs-Problems anhand der Pins einer Treiber-Baugruppe für TFT-LCD-Panele. (a) Zwei Pinzuordnungs-Probleme, (b) eine mögliche Pinzuordnung dargestellt durch die Luftlinien der Netze.

me anhand einer Treiber-Baugruppe für ein TFT-LCD-Panel. Die elektrischen Kontakte des TFT-LCD-Panels bzw. die Außenanschlüsse definieren eine Menge von Pinpositionen. Die Pins des Treiber-ICs legen jeweils die zweite Menge von Pinpositionen fest. Ziel der Pinzuordnung ist es, eine Zuordnung zwischen den Pinpositionen des Panels bzw. der Außenanschlüsse und denen des Treiber-ICs zu finden, so dass eine bestmögliche Verdrahtung der Signale auf der Treiber-Baugruppe möglich ist.

Wenn n die Anzahl der Zweipunkt-Netze eines Pinzuordnungs-Problems ist, dann gibt es $n!$ verschiedene Pinzuordnungen. Bereits für die kleine Anzahl von $n=25$ ergeben sich so $1,55 \cdot 10^{23}$ mögliche Pinzuordnungen. Da n für aktuelle Designs in der Größenordnung von Tausend liegt, ist es unmöglich, das Pinzuordnungs-Problem umfassend zu lösen, womit heuristische Vorgehensweisen bzw. Vereinfachungen anzuwenden sind.

4 Pinzuordnungs-Algorithmen

Die in diesem Abschnitt vorgestellten Pinzuordnungs-Algorithmen zielen darauf, Pinzuordnungen durch eine gezielte Minimierung der Signalkreuzungen zu optimieren. Signalkreuzungen werden dabei durch Schnittpunkte der Luftlinien (Abschn. 4.1 und 4.3) und Manhattan-Verbindungen (Abschn. 4.2) modelliert.

4.1 Pinzuordnung ohne Signalkreuzungen in euklidischer Geometrie

Bei dieser Methode werden Signalkreuzungen als Schnittpunkte der Luftlinien von Netzen abgeschätzt und minimiert.

Beginnend mit einer Ausgangs-Pinzuordnung sind dazu solange die Positionen von Pins zu vertauschen, bis keine Kreuzungen mehr vorliegen. Dazu wird, ausgehend von einem (willkürlich festgelegten) ersten Netz, überprüft, ob dessen Luftlinie die eines anderen Netzes schneidet. Wird eine solche Kreuzung gefunden, werden die Positionen von zwei Pins der beiden Netze so vertauscht, dass diese Kreuzung entfernt wird. Ergeben sich für dieses Netz weitere Signalkreuzungen, werden diese genauso aufgelöst. Anschließend ist das zweite Netz auf Kreuzungen zu überprüfen, bis schließlich alle Netze abgearbeitet sind. Dieser Vorgang wird iterativ so lange wiederholt, bis alle Kreuzungen entfernt sind.

Ist n die Anzahl der Netze, dann sind pro Wiederholung (Iteration) $\frac{1}{2}(n^2+n)$ Überprüfungen auf Kreuzungen nötig. Die Anzahl der notwendigen Wiederholungen hängt von der Ausgangs-Pinzuordnung ab. Allgemein gilt, dass die maximale Anzahl notwendiger Wiederholungen durch die Anzahl Signalkreuzungen des Netzes mit den meisten Kreuzungen definiert ist.

Ausgehend von jeder beliebigen Pinzuordnung findet dieser Algorithmus eine Pinzuordnung, bei der die Netze frei von Signalkreuzungen sind. Existieren mehrere Pinzuordnungen ohne Signalkreuzungen, dann hängt die gefundene Lösung von der Ausgangs-

Pinzuordnung und von der Abarbeitungsreihenfolge der Netze ab. Es ist also zufällig, welche der möglichen Pinzuordnungen ermittelt wird, da die Abarbeitungsreihenfolge der Netze zufällig ist.

Bild 2 zeigt ein einfaches eindeutiges Beispiel zur Verdeutlichung dieses Algorithmus.

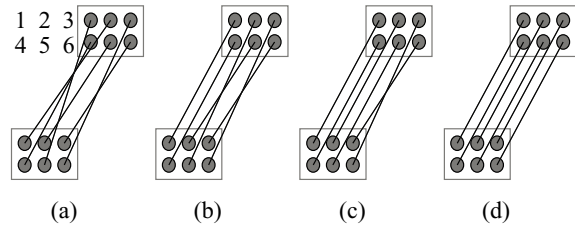


Bild 2. (a) Ausgangs-Pinzuordnung, (b) Pins an der Position 1 und 2 in der oberen Menge wurden vertauscht, (c) Pins 2 und 5 wurden vertauscht, (d) Pins 3 und 6 wurden vertauscht.

4.2 Pinzuordnung ohne Signalkreuzungen in Manhattangeometrie

Zur Berücksichtigung der Manhattangeometrie einer Verdrahtung lassen sich Signalkreuzungen auch als Schnittpunkte der jeweiligen halben Umfänge der Manhattan-Verdrahtungsfenster zweier Netze ableiten. Die so abgeschätzten Signalkreuzungen werden analog zu dem in Abschnitt 4.1 vorgestellten Pinzuordnungs-Algorithmus komplett entfernt, wobei es keine wesentlichen Unterschiede zu der dort beschriebenen Rechenkomplexität gibt.

Für die Abbildung eines Netzes als halben Umfang des Manhattan-Verdrahtungsfensters existieren grundsätzlich zwei Varianten (**Bild 3**). Um diese Netz-Abbildung für den Pinzuordnungs-Algorithmus einzusetzen, ist vorab zu definieren, welches Pin des Netzes *Pin 1* ist und in welcher Richtung *Pin 1* verlassen wird. In dieser Arbeit ist immer das Pin, welches zu der ersten Menge von möglichen Pinpositionen gehört, *Pin 1* (*Pin 2* ist folglich der zweiten Menge zugeordnet).

Entsprechend der beiden möglichen Richtungen, in denen *Pin 1* eines Netzes auf dem Rechteckumfang verlassen werden kann, lassen sich so zwei unabhängige Varianten der Pinzuordnung erstellen.

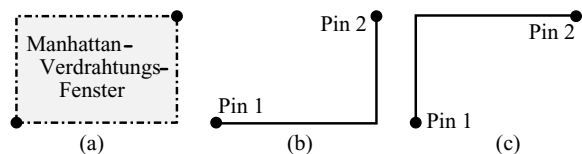


Bild 3. Zwei Varianten (b) und (c) zur Abbildung eines Netzes als halber Umfang seines Manhattan-Verdrahtungsfensters (a).

4.3 Projektion der Pins auf eine Gerade

Bei dieser Methode werden Signalkreuzungen als Schnittpunkte der Luftlinien von Netzen abgeschätzt und minimiert. Im Gegensatz zur Methode in Abschnitt 4.1 werden hier jedoch die Pinpositionen auf

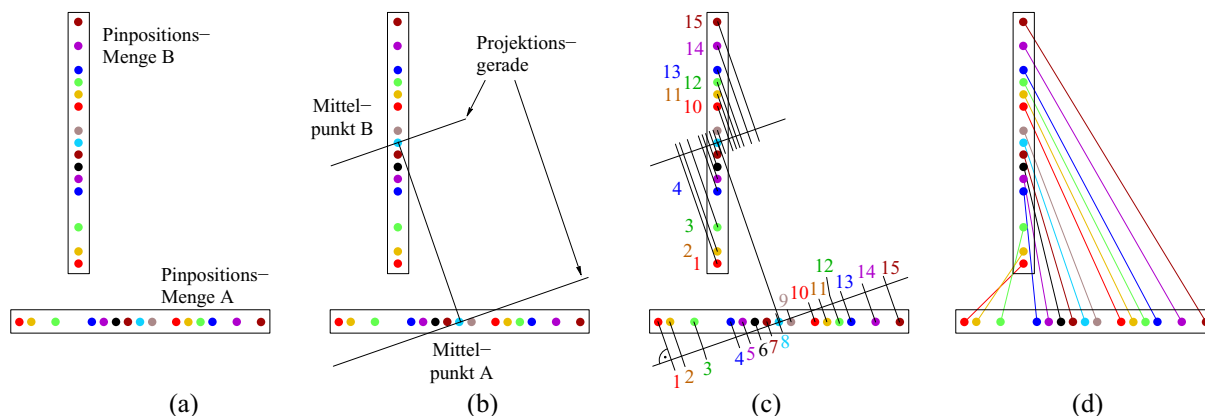


Bild 4. Pinzuordnung durch Projektion auf eine Gerade. (a) Ausgangssituation, (b) Projektionsgerade, (c) Projektion der Pinpositionen und Reihenfolge der projizierten Pinpositionen, (d) Pinzuordnung.

eine Gerade projiziert. Diese Projektionsgerade steht senkrecht auf der Verbindungslinie der Mittelpunkte der Mengen von Pinpositionen. Damit die Verbindungslinie und dadurch die Projektionsgerade sinnvoll bestimmt werden können, müssen die Mittelpunkte der Mengen jedoch hinreichend unterschiedlich sein.

Die Pinpositionen jeder Menge werden dann anhand der Reihenfolge der projizierten Positionen durchnummeriert. Fallen projizierte Pinpositionen aufeinander, so wird zufällig eine Reihenfolge für diese Pins ausgewählt. Jeweils die beiden Pinpositionen mit der gleichen Nummer werden den beiden Pins des gleichen Netzes zugewiesen. **Bild 4** zeigt diesen Pinzuordnungsprozess.

Mittels des Skalarproduktes kann die Reihenfolge der projizierten Pinpositionen effizient mit nur zwei Multiplikationen und einer Addition pro Pinposition bestimmt werden und eignet sich deswegen für Pinzuordnungs-Aufgaben mit vielen Netzen.

Die erstellten Pinzuordnungen weisen nur wenige Kreuzungen in den Luftlinien der Netze auf und sind deswegen eine günstige Ausgangs-Pinzuordnung für den in Abschnitt 4.1 vorgestellten Pinzuordnungs-Algorithmus.

Es lässt sich leicht zeigen, dass die Projektionsgerade nicht notwendigerweise für beide Mengen von Pinpositionen gleich sein muss und auch nicht zwangsläufig senkrecht zur Verbindungslinie der beiden Mengen von Pinpositionen zu orientieren ist. Besitzt beispielsweise ein Bus eine bevorzugte Richtung zum Ausfächern, so kann die Projektionsgerade senkrecht zu dieser Richtung liegen. Auf diese Weise wird eine Pinzuordnung erstellt, die in der endgültigen Verdrahtung wenige Signalkreuzungen aufweist. Dies trifft insbesondere bei Spezialfällen zu, wie z.B. die Umführung von Bauelementen, bei denen die anzuschließenden Pins auf der abgewandten Seite liegen.

5 Experimentelle Ergebnisse

Die vorgestellten Pinzuordnungs-Algorithmen wurden anhand von industriellen Layouts getestet. Da nur wenige vorangegangene Arbeiten zur automatischen Pinzuordnung auf Leiterplatten existieren, stehen kei-

ne allgemein anerkannten Benchmarks zur Verfügung. Daher wurden die Pinzuordnungs-Algorithmen mittels industrieller Designs von IBM getestet. Die hier vorgestellten repräsentativen Ergebnisse stammen von einem Single Chip Module (SCM) und einem Ausschnitt aus einem Leiterplattenlayout (KARTE).

Das Testlayout SCM besteht aus einem Flip-Chip-montierten Die mit 1058 Signalpins, welcher im Zentrum auf der Oberseite eines Umverdrahtungsträgers sitzt. Die Unterseite des Umverdrahtungsträgers ist ganzflächig mit 1058 Signalpins und 1058 Spannungs-/Massepins belegt. Im Umverdrahtungsträger wird jedes Chipsignal-Pin mit einem Signal-Pin auf der Unterseite kontaktiert. Die Pinzuordnungs-Algorithmen wurden angewendet, um eine optimierte Zuordnung zwischen Chip-Pins und den Pins auf der Unterseite des Verdrahtungsträgers zu erstellen.

Das Testlayout KARTE ist, vergleichbar mit **Bild 1**, ein Layout, in dem die Pinzuordnung zwischen einer Steckerleiste an der Unterseite einer Leiterplatte und einem darüber liegenden Prozessor erfolgen soll. Das Testlayout KARTE enthält 1120 Netze, für die eine optimierte Pinzuordnung gesucht ist.

5.1 Qualität der Pinzuordnungen

Zur Bewertung der Qualität der Pinzuordnungen werden neben der bereits erwähnten Anzahl der Signalkreuzungen die Gesamtlänge der Netze, die Standardabweichung der einzelnen Netzlängen und die finalen Verdrahtungsergebnisse des SPECCTRA-Autorouters verglichen. Die Signalkreuzungen werden als Schnittpunkte der Luftlinien der Netze bestimmt. Die Gesamtlänge der Netze ist sowohl in Manhattangeometrie (halber Umfang des die Pins umschließenden Rechtecks) als auch in euklidischer Geometrie (Länge der Luftlinien) angegeben (Tabelle 1).

Die Algorithmen 4.1 und 4.2 benötigen eine Ausgangs-Pinzuordnung. Die in dieser Arbeit vorgestellten Ergebnisse basieren auf einer mit Algorithmus 4.3 erstellten Ausgangs-Pinzuordnung.

Die globalen Minima für die Manhattan-Gesamtnetzlänge und für die euklidische Gesamtnetzlänge lassen sich unabhängig voneinander durch Lösen eines linearen Zuordnungsproblems, z.B. mittels des sog. Unga-

Tabelle 1. Testergebnisse für SCM und KARTE

Testdesign	Pinzuordnungs-Algorithmus	Manhattan-Gesamtlänge der Netze	Manhattan-Standardabweichung	Euklidische Gesamtlänge der Netze	Euklidische Standardabweichung	Signalkreuzungen
SCM	Unabhängige globale Optima	18523		13997		0
	Algorithmus 4.3	24308	11,12	18538	7,81	44686
	Algorithmus 4.1	18927	10,09	14272	7,06	0
	Algorithmus 4.2 (Variante 1)	22226	12,65	17506	7,27	43483
	Algorithmus 4.2 (Variante 2)	21282	12,59	16625	9,11	33699
KARTE	Unabhängige globale Optima	112359		101877		0
	Algorithmus 4.3	112392	14,44	101888	12,73	9703
	Algorithmus 4.1	112359	15,23	101877	12,78	0
	Algorithmus 4.2 (Variante 1)	115789	27,52	103808	15,37	144994
	Algorithmus 4.2 (Variante 2)	112523	22,23	102556	14,41	83716

Tabelle 2. Verdrahtungsergebnisse für SCM

Pinzuordnungs-Algorithmus	Geroutete Gesamtverdrahtungslänge	Vias
Minimale Manhattan-Gesamtnetzlänge	19542	1508
Minimale Euklidische Gesamtnetzlänge	19045	1242
Algorithmus 4.3	25516	2654
Algorithmus 4.1	19432	1412
Algorithmus 4.2 (Variante 1)	23945	2328
Algorithmus 4.2 (Variante 2)	22440	2066

rischen Algorithmus [12], bestimmen. Weil keine allgemein anerkannten Benchmarks für die Pinzuordnung verfügbar sind, sind diese globalen Optima in Tabelle 1 in der Zeile „Unabhängige globale Optima“ als Bewertungsgrundlage angegeben.

Die Pinzuordnungs-Algorithmen wurden in ANSI-C implementiert und auf einem 2.0 GHz PC ausgeführt. Die Laufzeit der Algorithmen für die hier angegebenen Testdesigns lag unter einer Sekunde. Größere Testdesigns zeigten, dass die Laufzeit für die Projektion der Pinpositionen mit der Anzahl der Netze skaliert und im Allgemeinen schneller als das Entfernen von Signalkreuzungen erfolgt. Dieses Verhalten entspricht auch der Komplexität der Algorithmen von $O(n^2)$ für Algorithmen 4.1 und 4.2 und $O(n)$ für Algorithmus 4.3, wobei n die Anzahl der Netze ist.

Die Testergebnisse zeigen, dass für SCM die Ergebnisse von Algorithmus 4.3 weit von den jeweiligen globalen Optima entfernt sind. Das liegt daran, dass die beiden Mengen von Pinpositionen (Die-Pins und Pins auf der Unterseite) fast deckungsgleich sind und sich damit die Projektionsgerade nicht sinnvoll bestimmen lässt. Für KARTE sind die Ergebnisse von Algorithmus 4.3 für die Netzlänge nur 0,03% und 0,01% vom Optimum entfernt. Von den vorgestellten Algorithmen erzeugt dieser Algorithmus für KARTE die geringste Standardabweichung der einzelnen Netzlängen, was einen eventuell notwendigen Längenabgleich innerhalb eines Busses begünstigt.

Algorithmus 4.1 liefert für alle Testdesigns gute Ergebnisse. Im Fall KARTE wird durch diesen Algorithmus eine Pinzuordnung erstellt, die bezüglich Manhattan-Netzlänge, euklidischer Länge und Anzahl der Signalkreuzungen optimal ist. Für SCM liegen die Manhattan-Netzlänge und die euklidische Länge der

Netze nur 2,1% und 2,0% von den globalen Optima entfernt. Auch ist Algorithmus 4.1 derjenige, der für SCM die geringste Standardabweichung erreicht.

Für die in Tabelle 1 aufgeführten Bewertungskriterien erzielt Algorithmus 4.2 schlechtere Ergebnisse als die Algorithmen 4.1 und 4.3. Wie im folgenden Kapitel gezeigt wird, eignet sich dieser Algorithmus zur Beeinflussung der automatisch erstellten Verdrahtung.

5.2 Auswirkungen auf die Verdrahtung

Die Ergebnisse in Tabelle 1 und die Verdrahtungsergebnisse (siehe Tabelle 2) zeigen, dass die Minimierung der Netzlängen und die Minimierung der Kreuzungen der Luftlinien zu ähnlichen Pinzuordnungen und zu ähnlichen Verdrahtungsergebnissen führen. Dabei hat die Pinzuordnung durch Entfernen der Signalkreuzungen den Vorteil, dass sie deutlich schneller ($O(n)$ bzw. $O(n^2)$) zu bestimmen ist als die Minimierung der Gesamtnetzlänge ($O(n^3)$, siehe [12]).

Die Pinzuordnungen durch Entfernen der Signalkreuzungen in Manhattangeometrie lassen sich durch eine zweidimensionale Darstellung der Kreuzungsdichte der Luftlinien detailliert untersuchen. **Bild 5** zeigt diese Dichteverteilung für die Variante, bei der in (a) die Chip-Pins horizontal verlassen und in (b) vertikal verlassen werden. Es ist zu erkennen, dass sich die Signalkreuzungen in unterschiedlichen Bereichen des Designs konzentrieren.

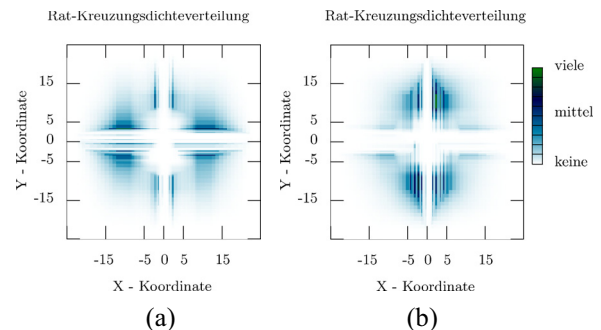


Bild 5. Dichteverteilung der Kreuzungen der Luftlinien (Rats) für SCM nach dem Entfernen der Signalkreuzungen mit Algorithmus 4.2. (a) Chip-Pins horizontal verlassen, (b) Chip-Pins vertikal verlassen.

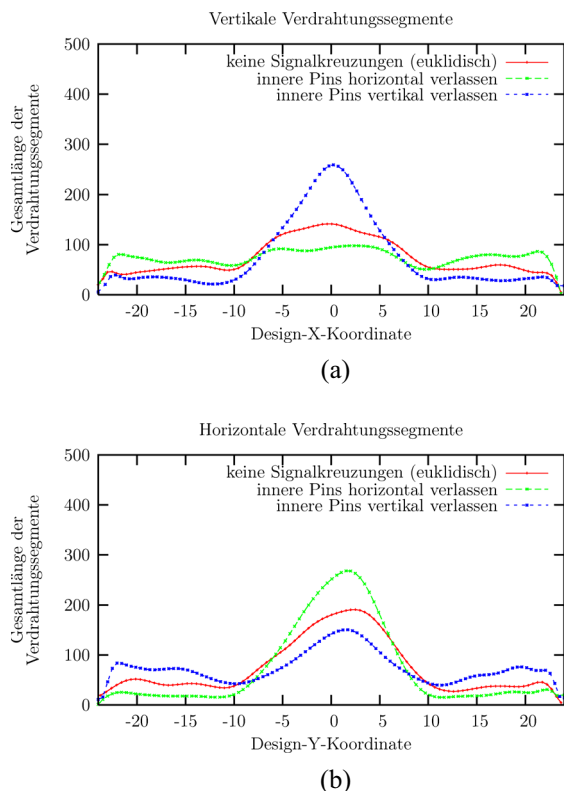


Bild 6. Verteilung der (a) vertikalen und (b) horizontalen Verdrahtungssegmente nach dem Autorouten.

Um darüber hinaus konkretere Qualitätseinschätzungen dieser Pinzuordnungen treffen zu können, erfolgte eine automatische Feinverdrahtung des Testdesigns SCM mit dem SPECCTRA-Autorouter von Cadence. Von diesen Verdrahtungen wurden die Längen der horizontalen und vertikalen Verdrahtungssegmente über der Designhöhe und -breite analysiert. Die **Bilder 6 (a)** und **(b)** zeigen die Verteilung der jeweiligen Verdrahtungssegmente. Im Zentrum (Koordinatenbereich von etwa -5 bis 5) befindet sich der Flip-Chip auf der Oberseite des Umverdrahtungsträgers. Es ist deutlich zu erkennen, dass der Algorithmus 4.2 die Verteilung der vertikalen und horizontalen Verdrahtungssegmente beeinflusst. Wird für Algorithmus 4.2 die Variante der Manhattangeometrie angenommen, bei der die inneren Pins (Chip-Pins) horizontal verlassen werden, so erzeugt der Autorouter eine Verdrahtungsgeometrie, bei der das Ausfächern unter dem Flip-Chip vorrangig in horizontaler Richtung stattfindet. Analoges gilt für das vertikale Verlassen der inneren Pins.

Der Autorouter selbst führt während der automatischen Verdrahtung Analysen der Verdrahtungsdichten und Signalkreuzungen durch. Diese Analysen fließen in die Erstellung der Verdrahtung ein. Der Algorithmus 4.2 beeinflusst den Autorouter so, dass die detaillierte Verdrahtung eines Netzes etwa der Manhattan-Geometrie entspricht, die während der Pinzuordnung angenommen wurde.

6 Zusammenfassung

In dieser Arbeit wurden drei Algorithmen vorgestellt, die automatisch eine Pinzuordnung erstellen. Keiner der Pinzuordnungs-Algorithmen liefert für alle Bewertungskriterien und alle Testdesigns gleichzeitig die besten Ergebnisse, so dass sich ein anwendungsspezifischer Einsatz empfiehlt.

Der Pinzuordnungs-Algorithmus 4.1 erzeugt für alle Pinzuordnungs-Aufgaben gute Ergebnisse bezüglich der Gesamtverdrahtungslänge und der Signalkreuzungen. Der Algorithmus 4.3 findet sehr schnell eine Pinzuordnung mit wenigen Signalkreuzungen, wobei sich eine Richtung für das Ausfächern berücksichtigen lässt. Für fast deckungsgleiche Mengen von Pinpositionen ist dieser Algorithmus jedoch nicht anwendbar.

Der Algorithmus 4.2 beeinflusst den Autorouter so, dass dieser eine Verdrahtung erstellt, die in der Nähe der Pins bevorzugt aus horizontalen bzw. vertikalen Verdrahtungssegmenten besteht.

Mit den hier vorgestellten Algorithmen können optimierte Pinzuordnungen für hochkomplexe Area-Array-Komponenten auf Leiterplatten innerhalb von Tagen erstellt werden, für die bisher Bearbeitungszeiten von etwa einem Monat notwendig waren.

Die vorgestellten Algorithmen befinden sich derzeit bei IBM zum Entwurf industrieller Designs im Einsatz und haben dort ihre Zuverlässigkeit und Zeiteffizienz bewiesen.

7 Literatur

- [1] Y. Cai and D. Wong. Optimal channel pin assignment. CAD of Integrated Circuits and Systems, IEEE Transactions on, 10(11):1413–1424, November 1991.
- [2] T. Koide, S. Wakabayashi, and N. Yoshida. An optimal channel pin assignment with multiple intervals for building block layout. DAC, 1992. EURO-VHDL '92, EURO-DAC '92. European, 348–353, Sept. 1992.
- [3] J. Westra and P. Groeneveld. Towards integration of quadratic placement and pin assignment. VLSI, 2005. Proceedings. IEEE Computer Society Annual Symposium on, 284–286, Mai 2005.
- [4] J. Lienig. Layoutsynthese elektronischer Schaltungen – Grundlegende Algorithmen für die Entwurfsautomatisierung, Springer Verlag, 2006.
- [5] S. Hauck and G. Borriello. Pin Assignment for Multi-FPGA Systems. CAD of Integrated Circuits & Systems, IEEE Transactions on, 16(9):956–964, Sept. 1997.
- [6] W. Mak and C. Lai. On Constrained Pin-Mapping for FPG-PCB Codesign. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 25(11):2393–2401, 2006.
- [7] N. L. Koren. Pin assignment in automated printed circuit board design. Proceedings of 9th workshop on Design automation, 72–79, 1972.
- [8] L. Mory-Rauch. Pin assignment on a printed circuit board in Design Automation. Proceedings of the 15th conference on Design automation, 70–73, June 1978.
- [9] T. D. Am, M. Tanaka, and Y. Nakagiri. An approach to pin assignment in printed circuit board design. ACM SIGDA Newsletter, 10(2):21–33, 1980.
- [10] H. Brady. An approach to topological pin assignment. CAD of Integrated Circuits and Systems, IEEE Transactions on, 3(3):250–255, July 1984.
- [11] M. Yu and W. Dai. Pin assignment and routing on a single-layer pin grid array. Proceedings of the ASP-DAC '95/CHDL '95/VLSI '95, 203–208, 1995.
- [12] Harold W. Kuhn. The Hungarian Method for the assignment problem. Naval Research Logistic Quarterly, 2:83–97, 1955.