The University of Osaka
Institutional Knowledge Archive

Title	ASICレイアウト設計の自動化に関する研究
Author(s)	神戸, 尚志
Citation	大阪大学, 1992, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3087996
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

ASICレイアウト設計の自動化に関する研究

1991年

神　戸　尚　志

ASICレイアウト設計の自動化に関する研究

1991年

神　戸　尚　志

内容梗概

本論文は、筆者がシャープ株式会社 技術本部 コンピュータシステム研究所および生産技術開発センターにおいて行ってきた「ASICレイアウト設計の自動化に関する研究」についてまとめたものであり、8章より構成される。以下、各章ごとにその概要を述べる。

第1章　緒論

ASICの技術動向、設計方式、設計手法について概説し、本研究の位置付け、目的、および課題を明確にする。

第2章　ASIC設計支援統合システム

ASIC設計全体の高効率化を達成するために、研究開発を進めているASIC設計支援統合システムについて考察する。機能論理設計、テスト設計、レイアウト設計など、ASICにおける各設計段階を総合的に支援するシステムについて述べる。

第3章　フロアプランニング手法

大規模なマクロセル方式ASICを設計する場合、フロアプラニング手法を用いてマクロセル間の大局的なレイアウトを最適化することが重要である。本章では、人工知能技法を応用したフロアプランニングシステムについて考察する。特にブロック配置・形状決定手法について詳細に述べ、実験によりその性能を評価する。

第4章　マクロセル方式自動配線手法

チップの各種設計階層において多く用いられる、マクロセル方式自動配線手法について述べる。特に詳細配線手法では、接続要求を満たすための初期配線とデザインルールを満たすための配線改善の、2つの処理工程に分けて実行することにより、問題の簡単化を図っている。

第5章　スタンダードセル方式セル配置手法

大規模スタンダードセル方式自動配置手法について述べる。2層メタルを用い

たセル上配線や電気的等価端子を利用する配線がチップ面積に影響を及ぼさないことに着目し、それらを相対的に増大させる配置改善手法を提案する。

第6章　スタンダードセル方式概略配線手法

大規模スタンダードセル方式自動配線手法について述べる。特にセルの持つフィードスルー領域を積極的に利用しつつ、スタンダードセル方式においてチップ面積増大の大きな要因である局所的な混雑の最小化を図る概略配線手法を考察する。

第7章　レイアウト設計支援統合化システム

自動設計ツールと会話型設計ツールの一体化を実現し、ASICレイアウト設計を総合的に支援することを目的として開発したシステムの構成、データベース、およびそのデータベース上で実現されたレイアウトツールの概要とその有効性について述べる。

第8章　結論

本研究で得られた結果を総括してまとめ、さらに今後の課題を述べる。

目次

第1章　緒論

1.1　ASICの技術動向

1.2　ASICの設計方式

1.2.1　ゲートアレイ方式

1.2.2　スタンダードセル方式

1.2.3　マクロセル方式

1.2.4　シリコンコンパイラ

1.3　ASIC設計手法

1.4　本論文の目的・課題とその主な内容

1.5　本論文の構成

第2章　ASIC設計支援統合システム

2.1　緒言

2.2　ASIC設計支援統合システム

2.3　機能論理設計支援システム

2.4　テスト設計支援システム

2.5　フロアプランニングシステム

2.6　自動レイアウト設計システム

2.7　会話型レイアウト設計支援システム

2.8　結言

第3章　フロアプランニング手法

3.1　緒言

3.2　フロアプランニングシステム

3.3　レイアウトモデル

3.4　ブロックの配置と形状の決定

3.4.1　人工知能技術の応用

3.4.2　配置手法
3.5　実験結果
3.6　結言

第4章　マクロセル方式自動配線手法
4.1　緒言
4.2　マクロセル方式レイアウトにおける配線問題
4.3　マクロセル方式自動配線手法の概要
4.4　チャネル決定
4.5　RCルータ
4.5.1　初期配線
4.5.2　配線圧縮
4.6　実験結果
4.7　結言

第5章　スタンダードセル方式セル配置手法
5.1　緒言
5.2　スタンダードセル方式におけるセル配置問題
5.3　シャフリング操作とその限界
5.3.1　シャフリング操作
5.3.2　シャフリング操作による配置手法—配置手法A
5.3.3　最適化の限界
5.4　縦方向ネット分布をも考慮した配置手法B
5.4.1　Tクラスタの生成
5.4.2　Tクラスタ内セル行割り付け
5.4.3　セル行形成及びセル行内概略順序決定
5.5　実験結果
5.6　結言

第6章　スタンダードセル方式概略配線手法

6.1　緒言

6.2　スタンダードセル方式における自動配線問題

6.3　概略配線手法

6.3.1　初期概略配線

6.3.2　概略配線の改善

6.4　チップ面積最小化のための機能

6.4.1　セルの局所的配置改善

6.4.2　周辺バッファとの一括配線

6.5　実験結果

6.6　結言

第7章　レイアウト設計支援統合化システム

7.1　緒言

7.2　ASICのレイアウト設計における諸問題

7.3　システム構成

7.3.1　ハードウエア構成

7.3.2　ソフトウエア構成

7.3.3　データベース構造

7.4　レイアウト設計ツール

7.4.1　小規模回路設計用レイアウトツール

7.4.2　中規模回路設計用レイアウトツール

7.4.3　大規模回路設計用レイアウトツール

7.4.4　会話型自動配線

7.5　システムの設計適用例

7.6　結言

第8章　結論

謝辞

本研究に関する関連発表文献

参考文献

第1章　緒論

1.1　ASICの動向

ASIC(Application Specific Integrated Circuits)とは、特定の用途あるいは特定のユーザのために、フルカスタムあるいはセミカスタム設計によって実現されるICの総称である[1]。ASICは1970年代後半から、情報処理技術、プロセス製造技術、CAD技術の進歩により急速に普及し、現在では産業機器分野、民生機器分野、OA機器分野等において広く活用されている。

ASICの設計技法はその普及の伸展や各種製造技術の進歩に伴い、次のように変化を遂げている[2]。

(1) 高集積化

製造プロセスにおける微細加工技術の発展により、1980年代初頭では2~3ミクロン程度であった精度が、ほぼ3年に2倍のペースで向上し、最近はサブミクロンに達している。これに伴い、ASICで実現可能な回路規模は、当初は数千ゲート程度であったものが、現在では数十万ゲートへと急速に増大している。

(2) 設計期間短縮

設計期間短縮へのユーザの要望はますます高くなり、ゲートアレイを例にとると、論理シミュレーション終了後からサンプル出荷までの期間が1980年代初頭には4週間が標準的であったものが、最近では1週間を切るものも出現している。

(3) 複合化

ASICは大規模化するだけでなく、多種多様に複合化する傾向も極めて強い。従来、ランダム回路のみをLSI化の対象としていたが、ROM、RAMなどのメモリを搭載したり、さらに最近はアナログ/ディジタル回路の混在、あるいはCPUコアの導入など、多機能化・高機能化が図られ、ASICに対する設計複雑度が急速に増大している。

1.2　ASIC設計方式

ASICは設計方式により大きく分類してゲートアレイ方式、スタンダードセル方式、マクロセル方式、シリコンコンパイラ方式、プログラマブル論理デバイスやプ

ログラマブル1チップマイコンを用いた方式などがある[3]。これらは開発期間、開発コスト、製造コスト、設計自由度、設計支援ツールなどの点から比較すると各々特徴があり、応用に適した設計方式が選択される。以下では、ゲートアレイ方式、スタンダードセル方式、マクロセル方式、シリコンコンパイラ方式など、ASICの代表的な設計方式について説明する。

1.2.1 ゲートアレイ方式

ゲートアレイ方式とは、例えば図1-1のように基本セルをあらかじめVLSIチップ上に形成しておき、基本セル内及びセル間の配線設計だけを施して所望のVLSIを実現する方式である。ただし、電源配線は、どのセルも電気的特性に関する仕様を満たすようにあらかじめ設計されている。このように、配線に関するマスクパターンのみを生成するだけで所望のVLSIが作成できるため、この方式は多品種少量のVLSIを短期間にしかも安価に製造するのに適している。

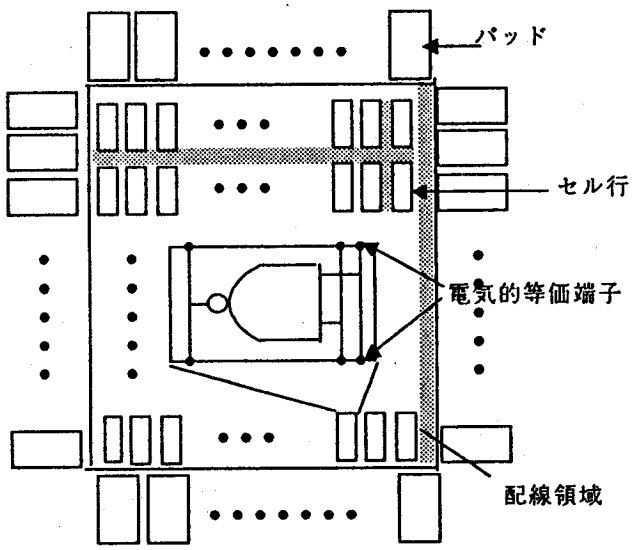


図1-1 ゲートアレイのレイアウトモデル

ゲートアレイの基本セルは、いくつかのトランジスタから構成され、各種論理機能をそれらの基本セルを用いて実現できるように、あらかじめレイアウト設計し、ライブラリに登録する。ゲートアレイの設計フローを図1-2に示す。まず、セルラ

イブラリを用いて所望の論理回路を設計する。次に、入力の各端子にそれぞれ加えるテストベクタとそれに対応する出力の期待値を作成し、論理シミュレーションを行い、回路動作を確認する。論理シミュレーションにおいては各セルに仮想配線長の負荷を付与して遅延を考慮したタイミングシミュレーションも実行する。論理設計と検証の試行を繰り返し、期待した機能および性能が得られた後に、レイアウト設計に移る。レイアウト設計では、自動配置配線設計を実行し、その結果から得られた実配線遅延に対するタイミングシミュレーションを通して動作確認を行った後に、製造を開始する。

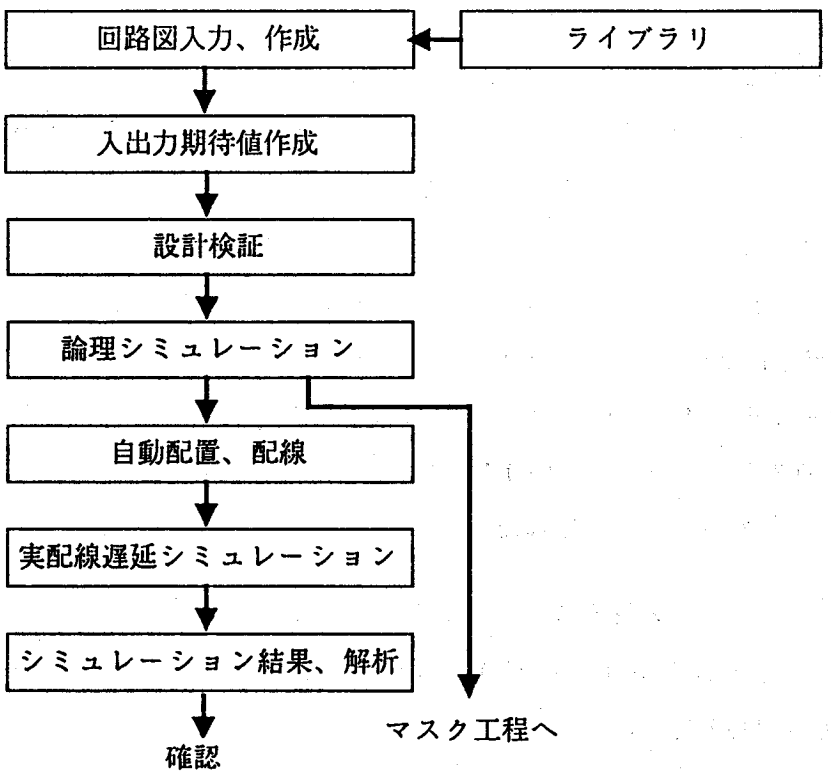


図1-2　ゲートアレイの設計フロー

最近普及しつつあるゲートアレイに、全面素子型がある。これは上記の従来型ゲートアレイとは異なり、基本セルをチップ全面に隙間なく配置し、特別に配線用チャネルを定めずに、実際の配線時に必要に応じてチャネルを確保する設計方式である。この方式では、ROM、RAM、マクロセル等を複数本のチャネルを使って自

由な形状に設計することができる。このように全面素子型ゲートアレイはマクロセルの形状や配線用チャネルの使用に自由度があるため、レイアウト効率が向上し、従来よりもさらに大規模なゲートアレイが実現可能となった。

1.2.2 スタンダードセル方式

スタンダードセル方式は、ゲートアレイ方式と同程度の設計期間で、しかもより高密度なVLSIを設計するために考えられた方式である。

この方式では、まず数十種類の機能単位を、幅は異なるが、高さがほぼ同一の矩形領域内に収まるようにあらかじめレイアウト設計を行い、セルライブラリとして登録しておく。次に、与えられた論理設計仕様に基づいてこれらのセルに対する配置配線設計を行なって、所望のVLSIを実現する。セルに対する配置配線設計は、ゲートアレイの場合と同様に、セル列とそれらの間の配線領域を用いて行なう。

この方式とゲートアレイ方式との大きな相違点は、セルに対するマスクパターンをあらかじめ作成しないことであり、そのため通常は設計、製造に要する期間がやや長くなる。しかし、チャネルの寸法が固定されていないなど、より柔軟性のある高密度設計が可能であり、この方式の需要は根強い。

スタンダードセル方式には大別して次の2種類がある。

①ポリセル型スタンダードセル(図1-3(a))

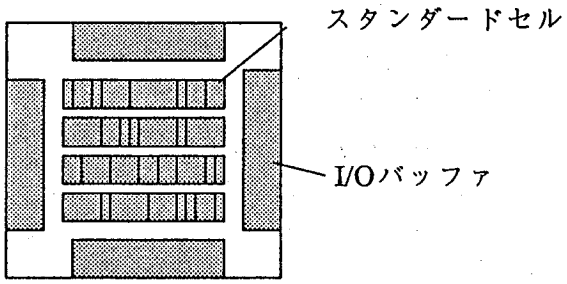
ほぼ同じ高さを持つ標準化された基本セルのみをアレイ状に配置配線する方式。

②ビルディングブロック型スタンダードセル(図1-3(b))

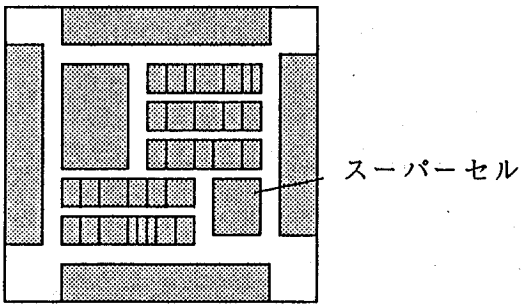
スタンダードセルアレイ中にROM、RAM、CPUコアなどのマクロセルを含むことが可能である方式。

1.2.3 マクロセル方式(図1-3(c))

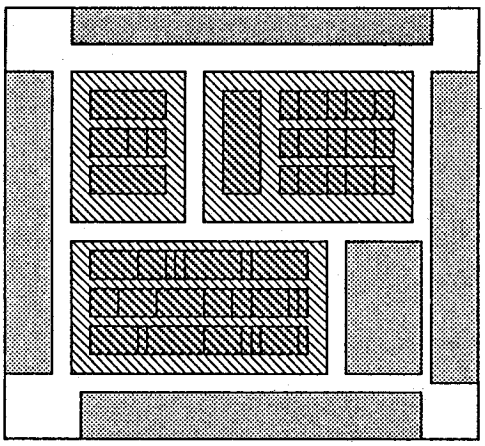
スタンダードセル方式VLSIにおいては、基本的に各セルの高さがほぼ一定であるため、レイアウト設計は行い易いが、個々のセルの論理機能が限定されたものであるから、チップ全体としての機能がかなりの制約を受ける。したがって、最近のASICの傾向として、ROM、RAM、あるいは各種論理機能を1チップ上で実現しな



(a) ポリセル型スタンダードセル方式



(b) ビルディング型スタンダードセル方式



(c) マクロセル方式

図1-3　スタンダードセル方式及びマクロセル方式のレイアウトモデル

ければならないが、この場合にはスタンダードセル方式には限界がある。マクロセル方式は、このような多種多様な論理機能を搭載するVLSIの設計のために導入された設計方式である。この方式では、さまざまな論理機能を相異なる形状や大きさ

の矩形領域で実現するマクロセルに対して、チップ面積最小という目的関数のもとにいかにして配置配線設計を実行するかということが主要な問題となる。

近年のVLSI技術の急速な進歩に伴ってVLSIの高集積化、したがって、1チップ上に搭載すべきシステムの大規模化が進行し、論理設計及びレイアウト設計の複雑さがますます増大しようとしている。これに対処する一つの有力な手段として、階層設計手法が取り入れられ、設計効率の向上に大きく貢献している。特にマクロセル方式VLSIの設計においてはそれが顕著である。

階層設計の利点として以下の点が挙げられる。

(1) 大規模なシステムを機能別に分割し階層構造化することによって、全体の設計工程が簡明となり、かつ処理時間と手間の削減が図れる。

(2) 大規模VLSIでは、配線の引回しによる遅延増加等に起因するタイミングエラーがよく問題となるが、階層構造化によってネットの広がる領域の囲み込みが可能となり、タイミング問題を軽減することができる。

(3) 設計自動化技法の進展によって、機能ブロック内のレイアウト設計を行う時機能に応じた自動化手法が適用でき、設計の最適化や効率化が可能となる。

1.2.4 シリコンコンパイラ

VLSIの大規模化と複雑化が進行し、“設計危機(design crisis)”と叫ばれているなかで、“理想の自動設計システム”として注目を集めているのが、シリコンコンパイレーション技法である[4]。“シリコンコンパイレーション”とは、ソフトウエアの分野におけるコンパイラが、FORTRANやC言語のような高級言語で記述されたプログラムを自動的に機械語に変換することを意味するのと同様に、ハードウエア設計において、「高位レベル言語で記述されたハードウエア記述を、自動的にシリコン構造データに変換する」ということを意味する。このような概念に基づいて多くの研究が行なわれてきたが、真の意味のシリコンコンパイレーション技法はまだ実現されていない。現存するシリコンコンパイレーションはモジュールに限定した設計支援システムであるといってよい。

モジュールコンパイレーションとは、設計者の与えた機能仕様やパラメータから

VLSIを構成するROM、RAM、データパス等の大規模なモジュール(あるいはマクロセル)のレイアウトパターンを、あらかじめ定められた設計規則に基づいて自動生成するツールである。このようなモジュールコンパイレーションによるVLSI設計においては、まず実現しようとするVLSIの各機能をモジュールに割り付けて自動生成し、ついでモジュールの配置並びにモジュール間の配線を実行するという手順をとる。

1.3 ASIC設計手法

ASICの大規模化、複合化はその設計手法に大きな影響を与えている。従来のゲートアレイやスタンダードセルでは論理設計とレイアウト設計における設計単位はゲートであった。論理設計は主としてゲートレベルの論理回路を作成することであり、レイアウト設計はそのゲートレベル論理回路を入力としてセル割り付け、配置・配線を行うことであった。これに反して、最近ではモジュールが論理設計およびレイアウト設計の設計単位となりつつある。ここでモジュールとはレジスタ、メモリ、演算器などの、ある程度以上の規模を有する機能単位をいう。したがって、論理設計においては個々のモジュールはある規模の論理機能を実現しているため、その記述はゲートレベルよりも機能レベルの方が適しており、論理検証もゲートレベルで行うよりも機能レベルで行う方がより簡明かつ高速である。さらに、新たに設計したモジュールや既設計のモジュールを配置配線してチップを構成する手法の方が大規模ASICの設計に適している。このように、ASIC設計手法はゲートレベルからモジュールベースへと全体的に変化しつつある[5]。

以下でモジュールベース設計の各設計段階について簡単に述べる。

(1) 機能レベル設計

機能レベル設計では、与えられた仕様を満足するハードウエアとして実現するための設計方式、すなわちアーキテクチャ、制御手続などを決定し、機能レベル検証により目的の仕様が論理的に実現できるかどうかを確認する。

(2) モジュール割り付け

この段階は論理的に検証された機能レベル記述を入力とし、これをスピードやタイミング等の電気的諸条件とチップ面積の妥当性を考慮しつつ、階層的モジュール構成に変換する。すなわち、個々の機能について、モジュールとして実現する設計手法を選択し、チップ全体のレイアウト方法を決定する。また、レイアウト手法が決定されたモジュールに対して、必要に応じて詳細な論理設計やレイアウト結果を想定した論理シミュレーションによる設計検証を行う。

(3) テスト設計

設計された論理回路が仕様を満足しているかどうかを検証するためのテストパターン(入力値及びそのときの出力期待値)を生成する。生成されたテストパターンが仕様をすべて満たすかどうかを評価することは難しく、一般に単一縮退故障を仮定した故障シミュレーションによる故障検出率をもって、これの評価とする。最近の大規模化、複合化したVLSIでは高い故障検出率のテストパターンを設計することが難しいため、テスト設計をより精密に行うべく、論理設計の段階から、いかに仕様を満たすかだけではなく、いかに検証するかをも考慮したテスト容易化設計が行われている。

(4) フロアプランニング

階層的なレイアウト設計においては、トップダウンに各階層でのモジュール間の相対配置と各モジュールの形状および入出力端子位置を求め、ボトムアップに詳細な配置配線を実行する。フロアプランニングとは、このレイアウト設計におけるトップダウンに行う設計工程の部分をいう。すなわち、階層的に設計された論理回路をモジュールとしてレイアウトする際に、面積や電気的特性等の最適化を目的として、各モジュールの形状や相対配置を定める工程である。

(5) 配置配線設計及びレイアウト検証

配置配線設計の段階では、フロアプランニングによってトップダウンに求められた各階層のモジュールの相対配置や端子位置に基づいて、ボトムアップに詳

細なレイアウト設計を行う。この段階でフロアプランニングによる面積見積りや配線領域見積りに、ある程度以上の誤差が生じる場合には当初のフロアプランを修正する。最終的に、得られたレイアウト結果に対してタイミングなどの仕様を満たすかどうかの検証を行う。また、レイアウト設計に人手が介入した場合には設計規則についての検証をも行う。

1.4 本論文の目的・課題とその主な内容

ASICの設計工程は、以上述べてきたように、高性能化、多機能化の要請に応じて多様化、複雑化している。このような状況下においてはASICの設計技法の高品質化、統合化がますます重要な課題となる。具体的には、設計技法の構築という観点から最も大きな問題は以下の点である。

(1) 設計複雑度の急激な増大

ASICの回路規模だけでなく設計複雑度の急激な増大により、CAD技法は従来の単純な延長ではその実用性に限界がある。そこで、ASICのための新たな設計技法として、トップダウン設計を支援するフロアプランニング、複雑化するレイアウトに対処するより高度な配置配線技法など、抜本的な研究開発が重要となる。

(2) サブミクロン製造プロセスの実用化

サブミクロン製造プロセスでは、配線遅延が回路動作に大きな影響を与える。このために、論理合成における遅延最適化、遅延を詳細に考慮したタイミングシミュレーション、実配線データからの遅延抽出、電源やクロックなど特定配線に対する設計支援などのように、各設計ツールにおいて回路動作を保障するための対策が不可欠となる。

(3) システム統合化

ASICの設計期間全体の中で、人手設計に依存している割合が依然として高い。よって、総合的に全設計工程の効率化を達成するには、自動化技法の高度

化、人手設計部分の高効率化、CADシステムの統合化などの研究開発が重要である。

本研究の目的は、特にレイアウト設計における、上記の課題について詳細な検討を行い、中でも重要な以下の項目について有効な手法を追求することにある。

(1) フロアプランニング

多様な評価関数を考慮しつつトップダウン設計の最適化を行うための、エキスパトシステムを導入した新しいAI手法を考察する。

(2) マクロセル方式自動配線設計

自動配線の全設計工程を、接続要求を満たすための初期配線とデザインルールを満たすための配線改善との、2つの処理工程に分割することによって問題の簡単化を図り、かつ高密度配線を実現する手法を考察する。

(3) スタンダードセル方式自動レイアウト設計

機能ブロックレイアウトによく用いられるスタンダードセル方式において、高密度レイアウトを実現させるために、チップ面積に影響を及ぼさないフィードスルーや電気的等価端子を利用した配置配線手法を考察する。

(4) レイアウト設計支援ツールの統合化

会話型設計ツールと各種自動設計ツールの一体化を実現するために、図形情報以外に素子情報、端子情報などを持たせたデータベースを構築し、レイアウトツール全体の統合化について考察する。

1.5 本論文の構成

本論文の構成は以下の通りである。

第2章では、ASIC設計全体の効率化を達成するために研究開発を進めてきたASIC設計支援統合システムについて概説する。ASICにおける各設計工程の支援ツールとして、

(1) 論理回路入力編集、機能論理回路シミュレータ、論理合成などから構成される機能論理設計支援システム、

(2) 故障シミュレータ、テストパターン自動生成、自動スキャン回路合成などの機能を持つテスト設計支援システム、

(3) モジュール配置および形状決定、概略配線などの機能から構成されるフロアプランニングシステム、

(4) 各種設計方式に対応した自動配置配線を行う自動レイアウト設計システム、

(5) 自動設計が困難な場合に人手設計を効率的に進めるための、シンボリックレイアウト、オンライン設計検証などからなる会話型レイアウト設計支援システム

などがあり、これらが一つのデータベースに統合化され、ASIC設計全体の設計品質の向上を図っている。

第3章では、大規模ASICを設計する際に不可欠なフロアプラニング手法について考察する。まず、フロアプランニングの対象となるレイアウトモデルについて述べ、ついでフロアプランニングにおいて複雑多岐にわたる設計条件を総合的に評価し最適化を図るために、エキスパートシステムによる手法とヒューリスティックな手法を組み合せた技法を提案する。すなわち、設計者のノウハウを組み入れたエキスパートシステムを用いて、各種設計条件を考慮し、モジュールの隣接関係を決定するための手法を述べ、さらに詳細配置を求めるための、接続関係を評価関数とするヒューリスティックな算法について述べる。本手法により設計者の要求する各種設計条件を考慮しつつ、チップ面積を最小化したモジュール配置が得られることを、いくつかの実験により示す。

第4章では、マクロセル方式レイアウトにおける自動配線手法について述べる。ASICに対する階層的レイアウト設計では、特にマクロセル方式が各種階層におけるレイアウト設計に多く用いられる。しかし、マクロセル方式配線では、一般には配線領域が複雑な形状となり、配線の最適化が困難であるという問題がある。そこで本章では、配線領域の凹凸を削減するチャネル定義手法と新しいチャネル配線手法を提案する。新しいチャネル配線手法は、多角形配線領域において高密度配線を達成

するために、トポロジカルな配線を決定する初期配線と、設計規則の許す範囲で配線の移動を反復する配線改善からなる。実験により本手法が配線領域形状の複雑さによる配線迂回や空領域を削減できることをも示す。

第5章では、大規模スタンダードセル方式レイアウトにおけるセル配置手法について述べる。初期解に依存せずセル配置最適化を行うことができるシャフリング手法を大規模スタンダードセル方式に適用できるように改良を施す。すなわち、大規模スタンダードセル方式では2層メタルを用いたセル上配線(フィードスルー)や電気的等価端子を利用する配線パターンがチップ面積に影響しないことを利用し、セル行に対して垂直方向にクラスタ配置を行った後に、シャフリング手法を適用する。従来手法との比較により、本手法が有効であることを明らかにする。

第6章では、大規模スタンダードセル方式レイアウトにおける自動配線手法について述べる。スタンダードセル方式では、セルがフィードスルーを許す領域を持っており、まずこのフィードスルーの効率的な利用が配線結果の改善に多く影響することを示す。次にフィードスルーを積極的に利用しつつ、スタンダードセル方式レイアウトにおいてチップ面積増大の大きな要因である局所的な混雑の最小化を図る概略配線手法を考察する。具体的には総配線長の最小化と局所的な配線混雑の減少を目的関数としてフィードスルー割り付けと幹線割り付けを行うものであり、あわせてチップ面積最小化に有効と考えられる配線途中でのセルの局所的な移動、あるいはチップ周辺での一括配線を考察する。

第7章では、自動設計が困難な場合に必要となる人手設計を支援することを目的としてレイアウト設計支援統合化システムを考察する。まず、自動設計手法が現在持つASICレイアウト設計における問題点を明らかにし、それらを解決するための統合化システムを提案する。ついで統合化システムのシステム構成、データベースにおける特徴を示し、さらにシステムに実装されているレイアウトツールの機能と性能評価を設計目的別に示し、あわせて性能評価を行う。

第8章では、本研究で得られた結果と課題を総括する。

第2章　ASIC設計支援統合システム

2.1　緒言

本章では、ASIC設計工程全体の効率化を達成するために研究開発を進めてきたASIC設計支援統合システムについて考察する。ASICにおける各設計段階の支援として、機能論理設計支援システム、テスト設計支援システム、フロアプランニングシステム、自動レイアウト設計システム、会話型レイアウト設計支援システムがあり、これらを、一つのデータベースに統合化してASIC設計全体の設計品質の向上を図っている。

2.2　ASIC設計支援統合システム

ASICの高密度化、大規模化、設計複雑化に伴い、設計自動化技術はさらに高性能化、高品質化を図らなければならない。このような背景から1980年に開発し実用化した自動レイアウト設計システムSHARPS[6]、1988年に開発した機能論理設計支援システムOLIVE2[7]などを統合し、ASIC設計の各段階を総合的に支援するシステムの構築を進めている。

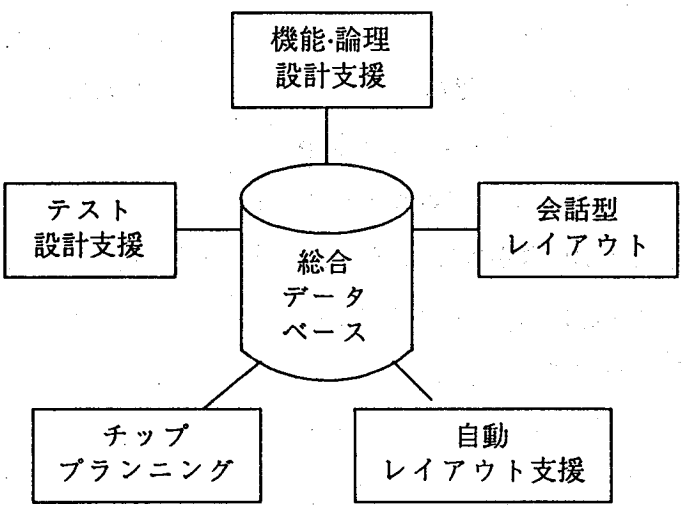


図2-1　ASIC設計支援統合システムの構成図

ASIC設計支援統合システムはゲートアレイやスタンダードセル方式VLSIの設計自動化だけでなく、人手設計によるカスタムVLSIの設計効率化をも目的としており、機能・論理設計支援システム、テスト設計支援システム、フロアプランニングシステム、自動レイアウト設計システム、会話型レイアウト設計支援システムの5つのシステムより構成されている。図2-1に示すように各設計支援システムは論理、レイアウト、テストなどの情報を持つ総合データベースを介して結合されている。また、これらは共通なユーザインタフェースを持ち、会話型処理関連ソフトウエアはワークステーションに搭載されている。シミュレーション、自動レイアウトなどのバッチ処理はネットワークを介して、データを大型計算機へ転送して実行する事ができる。

2.3 機能・論理設計支援システム

一般に、複雑な設計を誤りなく行うためには、設計の初期の段階からその正しさを十分に確かめてから、次の段階へ進むことが必要である。従って、例えば論理設計に入る前に機能設計の評価を可能な限り行っておくことが必須である。このような観点から、本システムは機能モジュール内のゲートレベル論理設計やトランジスタ回路設計を支援するだけでなく、ブール関数、真理値表あるいは機能記述などを入力とする機能レベルの設計を可能とし、システム設計にも利用できるシステムとなっている。このほか本システムは以下の特徴を持つ。

(1) マルチウインドウを装備し、各ウインドウ間でのデータ受け渡しや、複数の異なった回路の編集作業を同時に行うことができ、階層設計が容易である。

(2) リアルタイムに接続や重なりなどの図面検証、自動編集ができ、短期間に誤りのない回路が作成できる。

(3) 論理シミュレータ[8]、タイミング検証[9]など各種検証ツール及びレイアウトシステムとのデータインターフェイスを持ち、エディタ中から利用できる。

(4) トランジスタの各種パラメータやセル名、IC名などデータベースへの書き込み事項を必要に応じて新たに追加することができ、VLSI、プリント基板レイアウトの設計情報を保有できる。

2.4 テスト設計支援システム

テストの困難性を推定する指数として、ゲート数/ピン数の比がよく使われる。これは回路の外界との切り口である入出力ピンから、どれ位多くの回路を制御し観測する必要があるかを表す指標となる。この数字が大きければ大きいほどテストは困難であるとされている。実際にパッケージなどの制約からピン数の伸びは抑えられており、テストの困難さは年々増大している。したがって、設計の段階からテスト容易性を考慮しない限り、満足なテストは不可能となっている。本システムはテスト設計を支援するためのツール群であり、機能論理設計支援システムと密接に結合している。以下に主なツールを紹介する。

(1) 故障シミュレータ[10]

テストパターンの有効性を故障検出率という形で定量的に把握し、故障診断用データを得るために故障シミュレータが用いられる。故障シミュレーションの出力は、与えられたテストパターンによる故障検出率、未検出故障の一覧表、およびどのパターンでどの故障が検出されるかを示す故障辞書である。これらのパターンをもとにして、テスト設計者はテストパターンが十分か否かの判定、不十分な場合のテストパターンの追加改良などを行う。

(2) テストパターン自動生成機能[11]

高い故障検出率のテストパターンを人手で設計することは極めて困難であり、多くの工数がかかる。特に多品種少量生産のASICではこのために工数を割くことは非常に難しく、CADのサポートが重要となる。テスト容易化設計の一つであるスキャンパス方式の実用化によって、従来組み合せ回路に限って有効であったテストパターン自動生成が順序回路にも適用可能となっている。スキャンパ

ス方式を適用した回路では、わずかな時間で人手設計をはるかに上回る故障検出率のテストパターンを自動生成することができる。

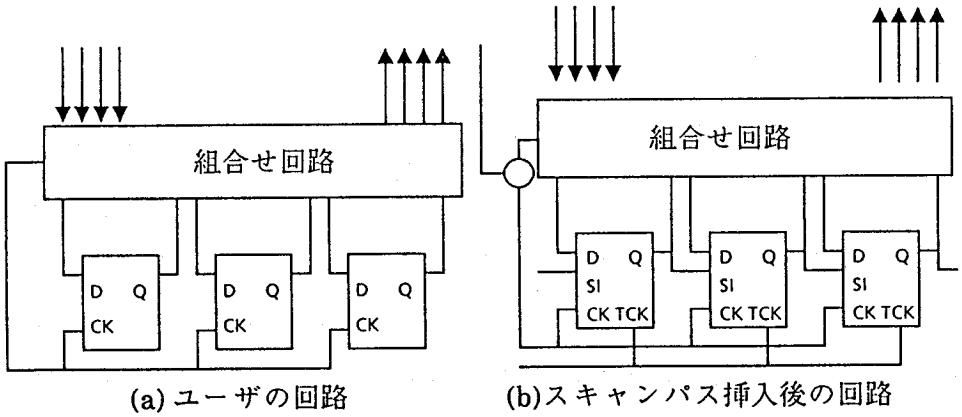


図2-2　スキャンパス回路説明図

(3)　スキャンパス設計支援機能

スキャンパス方式とは、例えば図2-2に示すように論理回路中に存在するフリップフロップなどの記憶素子が、システムクロックCKの入力時はユーザ所望の動作をし、テストクロックTCKの入力時はシフトレジスタとして動作する設計方式である。このような構成をとることによって組み合せ回路部分から独立して論理回路の内部状態の制御、観測が可能となるため、テスト設計が容易に行える。反面、スキャンパス回路を適用しないものに比べて回路構造が複雑になるため、設計者への負担増加、回路規模の増大を招いてしまう。そこで本機能ではテストパターン自動発生の過程で内部状態の観測もしくは制御が必要となったフリップフロップのみにスキャン機能を付加することで、必要最小限のスキャンパス回路を自動合成する。

2.5　フロアプランニングシステム[12]

本システムはVLSIのレイアウト設計の大局的決定を支援し、これを最適化するために次の4つの機能から構成される。

(a) モジュール配置機能

まず、チップレベルからトップダウンに、ある階層を構成するモジュールの相対位置を決定する。この段階では詳細なレイアウト設計がされていないモジュールが存在するため、モジュールの形状の決定を含めて最適化を図る配置手法を用いる必要がある。本サブシステムはこのためにルールベースによる自動配置機能を持つ[13]。本機能はルールベースで作成されているため、従来のヒューリスティック手法に比較して同時に多くの設計項目を評価しながら最適化を進める点に特徴がある。

(b) 概略配線機能[14]

電源、バス、クロックをはじめとする特別な電気的仕様を持つ配線設計、配線領域の見積もり、モジュールの端子位置決定などの問題がチップ面積最小化、電気的性能最適化のために重要となる。本システムは、このために以下の機能を持つ。

①人手もしくは自動で概略経路の指定(どのモジュール間の配線領域を通過するか)を行うことができる。また、モジュール配置が若干変更した時にも指定された配線経路を保存する機能を持つ[15]。

②モジュール間の各信号配線経路をチップ面積や配線長を考慮して最適化する。

③未設計のモジュールの端子位置はモジュールが配置される時点で、接続関係を持つ他のモジュールとの位置関係及び設定された概略配線経路をもとに最適化する。

(c) ユーザインターフェイス

データのロード・セーブ、自動配置・自動概略配線の起動、その結果や各種の評価関数による評価結果の表示、モジュールや配線領域の面積見積り表示、配置や配線に対して必要に応じた人手による変更などの機能がグラフィック端末を用いたエディタにコマンドとして備えられている。

(d) 配置配線システムの設計情報生成

本システムにおいて得られた配置・配線情報をレイアウトシステムに変換し、実際のレイアウト設計に反映させる。

本システム及びモジュール配置について第3章で詳細に述べる。

2.6 自動レイアウト設計システム

フロアプランニングシステムによりトップダウンに大局的なレイアウトが形成された後に、実際の自動レイアウトをボトムアップに行う。レイアウト手法としてゲートアレイ方式[16]、スタンダードセル方式[17]、マクロセル方式[18]などに対して自動配置配線プログラムが装備されている。

マクロセル間自動配線手法については第4章で、スタンダードセル方式セル配置手法は第5章で、スタンダードセル方式概略配線手法を第6章で詳細に述べる。

その他に、本システムにはモジュールコンパイレーションの環境として、レイアウト記述言語[19]に基づいて構築されたセルコンパイラ、データパスコンパイラ、PLAコンパイラなどを含む。

2.7 会話型レイアウト設計支援システム

高い集積度が要求されるASICにおいては人手によるレイアウト設計に頼らざるをえない。本システムではこれを支援する目的でレイアウトエディタ、シンボリックレイアウト及び自動コンパクション[20]、会話型自動配線[21]、オンライン設計検証(設計規則検証、接続抽出、レイアウトパラメータ抽出)、マスク生成などの機能が搭載されている。詳細は第7章で述べる。

2.8 結言

本章では本研究の背景となる、ASIC設計を総合的に支援する統合化システムの概要を述べた。本システムはゲートアレイやスタンダードセル方式VLSI設計だけでなく、カスタムVLSIの設計自動化および高効率化をも目的としており、機能・論理

設計支援システム、テスト設計支援システム、フロアプランニングシステム、自動レイアウト設計システム及び会話型レイアウト設計支援システムの5つのサブシステムより構成されている。

本システムはすでに開発実用化され、実際のASIC設計の自動化、高効率化に貢献している。

第3章　フロアプランニング手法

3.1　緒言

大規模なマクロセル方式ASICを設計する場合、まず、フロアプラニング手法を用いてマクロセル間の大局的なレイアウトを生成することが重要である。本章では、人工知能技法を応用したフロアプランニングシステムについて考察する。特にブロック配置・形状決定手法について詳細に述べ、実験によりその性能を評価する。

3.2　フロアプランニングシステム

近年ASICの高性能化・高機能化が進展するに伴い、自動レイアウト設計されたマクロセルあるいは機能ブロックや、ライブラリに登録されている各種設計資産などを組み合せて、階層的にチップを構成するという設計手法が一般的になりつつある。反面、階層化によって問題の局所化を行うため、レイアウト設計が完了していない機能ブロックを含む階層全体に対して、良いレイアウトの概略をあらかじめ求めておくことが新たに重要となっている。

フロアプランニングの目的は、詳細なレイアウト設計を行う前に、チップ面積、配線長、タイミングなど諸特性の推定・見積りを行うことによって、チップ全体について概略レイアウトを求め、レイアウト設計の方針を定めることである。

しかしながら、未設計ブロックを含んだフロアプランニングは、考慮すべき設計項目が多く、さらに、ブロック数や階層の深さの増大につれレイアウト設計の自由度が大きくなるため、人手による設計最適化には限界がある。ここに、有効な自動化手法の開発が強く求められるのである。

本章で提案するフロアプランニングシステムは、熟練していない設計者でも大規模なASICの実装設計を容易にかつ短期間で完了できることを目的として開発されたものであり、以下の特徴を持つ。

(1) 形状可変なブロックを含む自動配置、未設計ブロックの端子位置自動決定が可能である[14]。

(2) ブロック間配線領域の見積りができ、さらにその見積りに基づくブロック再配置によってチップ面積の最適化ができる。

(3) 配線長や経路予測をもとに、回路の正常動作を保証するためのタイミング条件に対する検証機能を備えている。

(4) フロアプランニングと実レイアウトとの間でアルゴリズムを共通化することによって、フロアプランニングの精度を保証する。

(5) 人手による設計最適化を支援するために、高機能なフロアプランエディタが用意されている。

(6) クロックなどの特別な信号線に対して、配線経路等の各種制約の指定および指定された制約を維持する機能を持つ[15]。

(7) フロアプランニングやレイアウト結果の情報を、階層毎に論理設計データベースへ反映し、実装レベルでの論理検証について精度向上を図っている。

(8) 各ブロックの面積や規模の調節を行うために、ブロックの合併・分割の操作ができる。

(9) フロアプランニング完了後は全階層のレイアウト設計が完全自動で行われ、人手介入を必要としない。

(10) ワークステーション上でレイアウト工程が全て処理可能である。

本システムの構成を図3-1に示す。

本システムは、各階層において各機能ブロックのレイアウト方式が既に決定されている論理設計データ(以下、レイアウトロジックと呼ぶ)を入力する。

レイアウトロジックは機能・論理設計支援システム[7]を用いて以下のような手順で設計される。

(1) 機能レベルでの設計及び検証

(2) 動作速度や面積等のレイアウトの条件に基づいた詳細な論理設計

例えば、論理式等で表現されたランダムロジック部をスタンダードセル方式によってレイアウト設計する場合には、人手設計または論理合成システム[22-24]により、スタンダードセルを割り付け、ネットリストを作成する。

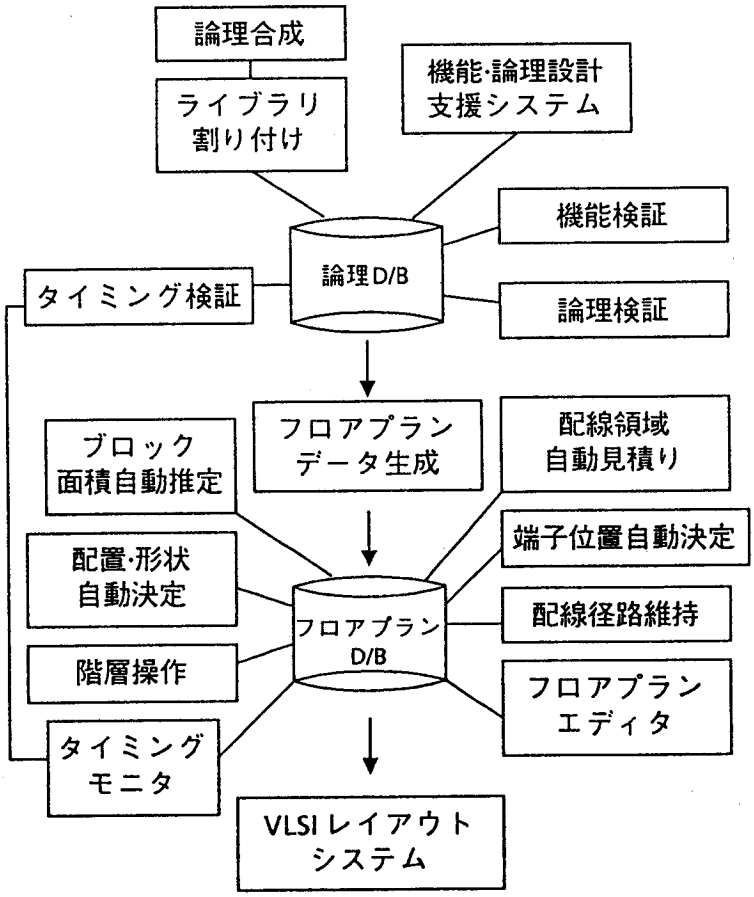


図3-1　システム構成

メモリやデータパス等をモジュールコンパイラを用いて生成する場合には、コンパイラに与える各種パラメータを決定する。

(3) チップの規模に応じてフロアプランニングの対象となる階層を決定する。例えば、ある機能ブロックAのゲート規模が大きい場合、それをさらにB1、B2などの幾つかの機能ブロックに分割し、機能ブロックAを新たにフロアプランニングの対象と指定し、B1、B2などの機能ブロックはスタンダードセル方式レイアウトの対象と指定する。

本システムでは、スタンダードセル方式レイアウト、モジュールコンパイラを用いたレイアウト、既設計のマクロセルを含むレイアウト、マクロセル方式レイア

ウトなどに適用可能である。

フロアプランニングシステムは、

(1) 設計の効率化を図るための自動プランニング機能

(面積自動推定、自動配置・形状決定、端子位置自動決定)

(2) 人手による最適化を行うための会話型機能

(階層操作、フロアプランエディタ)

(3) 品質を保証するための機能

(配線の経路指定・維持、タイミングモニタ)

等から構成されている。操作性の向上を図るため、すべての機能はフロアプランエディタ上でコマンド形式により起動される。

配置の自由度を高め無効領域の削減を図るために、ブロック配置は非スライシング構造をも取り扱う。また、チップ全体や未設計ブロックについてその縦横比を指定することができる。ただし、ブロック形状は矩形とする。

本システムを用いたフロアプランニング設計の手順は以下の通りである(図3-2)。

(1) レイアウトロジックを設計する。

(2) フロアプランデータベースを生成する。ここでは、すでに述べたように、指定されたレイアウト方式に沿って階層構造の変換も行う。

(3) 未設計ブロックに対して、ゲート数、ネット数、レイアウト手法、論理構造等の情報をもとにブロック面積の自動見積りを行う。

(4) 未設計ブロックの形状決定と全ブロックの自動配置を行う。

(5) ブロック間の配線予想に基づき、未設計ブロックの端子位置決定を行う。

(6) 配線領域の見積りを行う。見積りに従って、無効領域が減少するようにブロック位置・形状の最適化を行う。

(7) レイアウトシステムへデータを変換する。下位の階層より順にレイアウトを行う順序を定め、自動配置配線設計を行う。

フロアプランニングの各段階において、グラフィック画面による設計の評価を行う。必要に応じて未設計ブロックの分割・合併や、配置・端子位置の改善などの会話

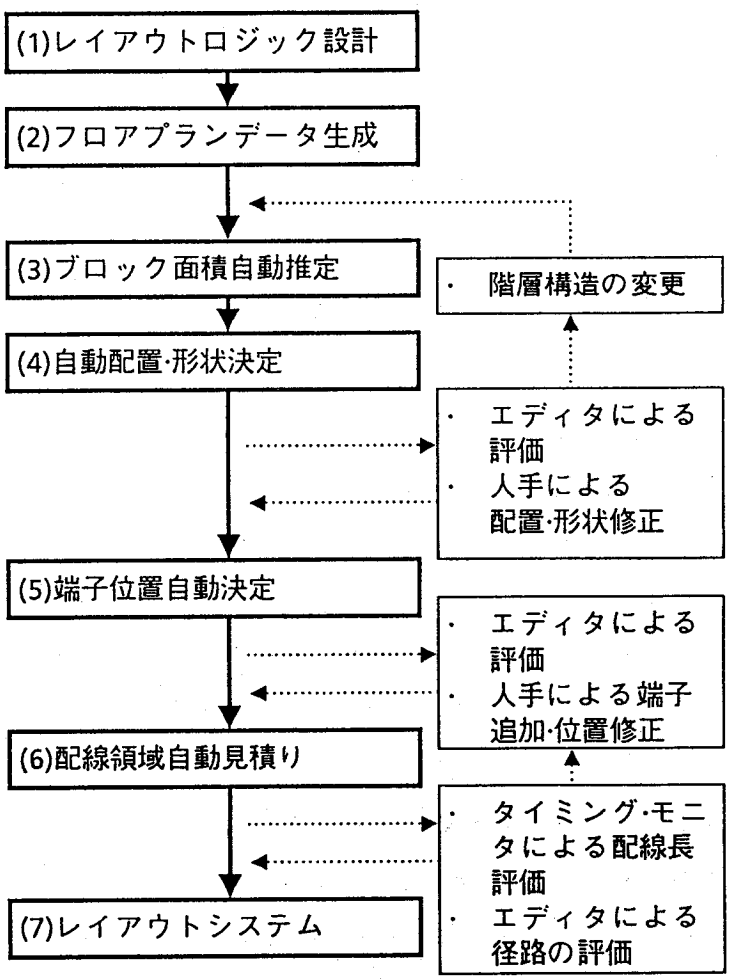


図3-2　フロアプランニング設計の手順

型機能を用いることができる。

配線長の増加によって遅延時間の増大が問題となるようなネットについては、配線領域の見積りと同時に仮想配線長の評価を行い、タイミングの検証を行う。タイミング制約に違反する場合には、配線経路の再指定、ブロックの再配置や分割·合併の対策を行う。また、電圧降下、発熱、ノイズ対策に必要な指定事項(配線幅や 配線経路の指定など)の設定も会話型機能で行う。指定された制約はその後のフロアプランニングや自動レイアウトにおいて維持される。未設計ブロックの分割·合併によって、階層構造の変更が行われると、階層間およびブロック間の接続関係が変わる。この時、論理設計によって定められたネット名を自動的に変更するが、変更さ

れたネットが論理設計のどの階層の、どのネットであったかを記述した対応表も併せて作成する。これにより、階層構造を変更した場合においても、レイアウトロジックへの情報のフィードバック(バックアノテーション)が容易に行える。

3.3 レイアウトモデル

本章で扱うフロアプランニングのレイアウトモデルについて述べる。チップは矩形であり、マクロセルとしての機能ブロックと4辺に置かれるI/Oバッファ列から構成される(図3-3)。各I/Oバッファの位置は、あらかじめ指定されているものとし、各機能ブロックは矩形であり、以下の2種類に分類される。

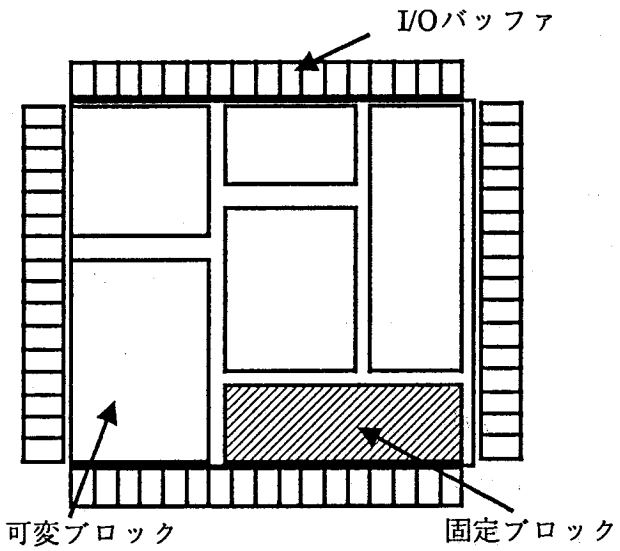


図3-3 レイアウトモデル

(i)固定ブロック：これは、形状が固定であり、かつブロックの各端子の位置が4辺上に固定されているようなものをいう。既に設計された機能ブロック、ROM、RAM、PLA等がこの固定ブロックとなる。

(ii) 可変ブロック：これは、形状の変更が可能であり、かつ各端子の位置は4辺上で自由に変更できるようなブロックである。その形状は、配置後に無効領域を削減するために変更される。また、その端子は、配置時にはすべ

てブロックの中心にあるものと想定し、全ブロック配置後にその具体的な位置を決定する。段数の変更により形状が変更できるようなスタンダードセルブロックがこの可変ブロックとなる。

3.4 ブロックの配置と形状の決定

以下では、AIを応用したブロックの配置と形状の決定手法について述べる。一般に、配置とは配置すべきセルの集合$X=\{x1,\cdots,xn\}$から、配置可能な位置(スロット)の集合$P(|P|\geqq n)$への単射である[25]。この配置問題は1つの2次割り付け問題として定式化されるが、その近似解を求めることさえもNP完全であることが知られている[25]。このような背景からフロアプランニングにおけるブロックの配置と形状の決定問題では、スライシング構造に基づく手法[26-28]、矩形双対グラフによる手法[29-32]、階層的手法[33]など、チップ面積や総配線長を評価関数としたヒューリスティックな手法が一般的であった。

しかし、実際にはASICが多機能化するに従って設計仕様が増大し、従来の設計自動化手法だけでは多様化する設計条件に対処することは困難となっている。例えば、デジタル/アナログ混在型ASICにおけるアナログブロックの配置では、発熱やノイズなどの影響を考慮する必要が生じたり、あるいは大規模ASICでは、電源やクロックのための各種電気的制約を満たすようにブロックを配置しなければならないなど、多種多様な設計仕様が新たに課せられるようになった。そこで、これらの各種設計仕様を満たしつつ配置を最適化する目的のために、人工知能技法を導入し、その有用性について検討する。

3.4.1 人工知能技法の応用

人工知能技法の応用には、次のような利点がある。第1に、設計者の意図やノウハウをルールとして表現すれば、ある定まったアルゴリズムにとらわれず、各種電気的制約をも考慮した配置設計が期待できる。第2に、新しいルールの追加やルールの修正によって、設計仕様の変更に対処することができ、従来のヒューリスティックアルゴリズムに見られないきめ細かな配置結果が期待できる。すなわち、

エキスパートシステムではルールの追加・修正は、アルゴリズムにおける処理の追加・修正に比べ極めて容易である。よって、アルゴリズム処理だけによるシステムでは一度完成してしまうと新しい処理機構を組み入れることが困難となるのに反して、ルールベースシステムを導入する場合には、種々の局所的な配置処理を容易に組み入れることが可能となり、人手設計において蓄積された知恵を具現することができる[34-36]。

本手法における考慮すべき項目として以下のものがあげられる。

1. ブロックに関する項目
   - 形状(固定、可変、大きさ、など)
   - 機能
   - 他ブロックとの接続
   - 電気的特性

2. ネットに関する項目
   - 種類(バス、電源、など)
   - タイミング(遅延)
   - 電気的特性

ルールは、IF-THEN形式で表現され、条件部には配置のための状態を記述し、結論部にはその処理を記述する。この時の考慮すべき項目によって配置の評価基準が決定される。現在のところ、おもに形状や接続に関するルールが主であるので、チップ面積と配線長を考慮した配置になる。例えば、バス信号(ネットの種類)について考慮したルールによって、バス配線を考慮した配置結果が期待できる。このように、配置の評価基準においてもルールの内容によって柔軟に対処する事ができる。

3.4.2　配置手法

配置・形状決定処理は、ルールベースによる併合処理とアルゴリズムによる詳細配置処理の2段階で行われる。(図3-4)

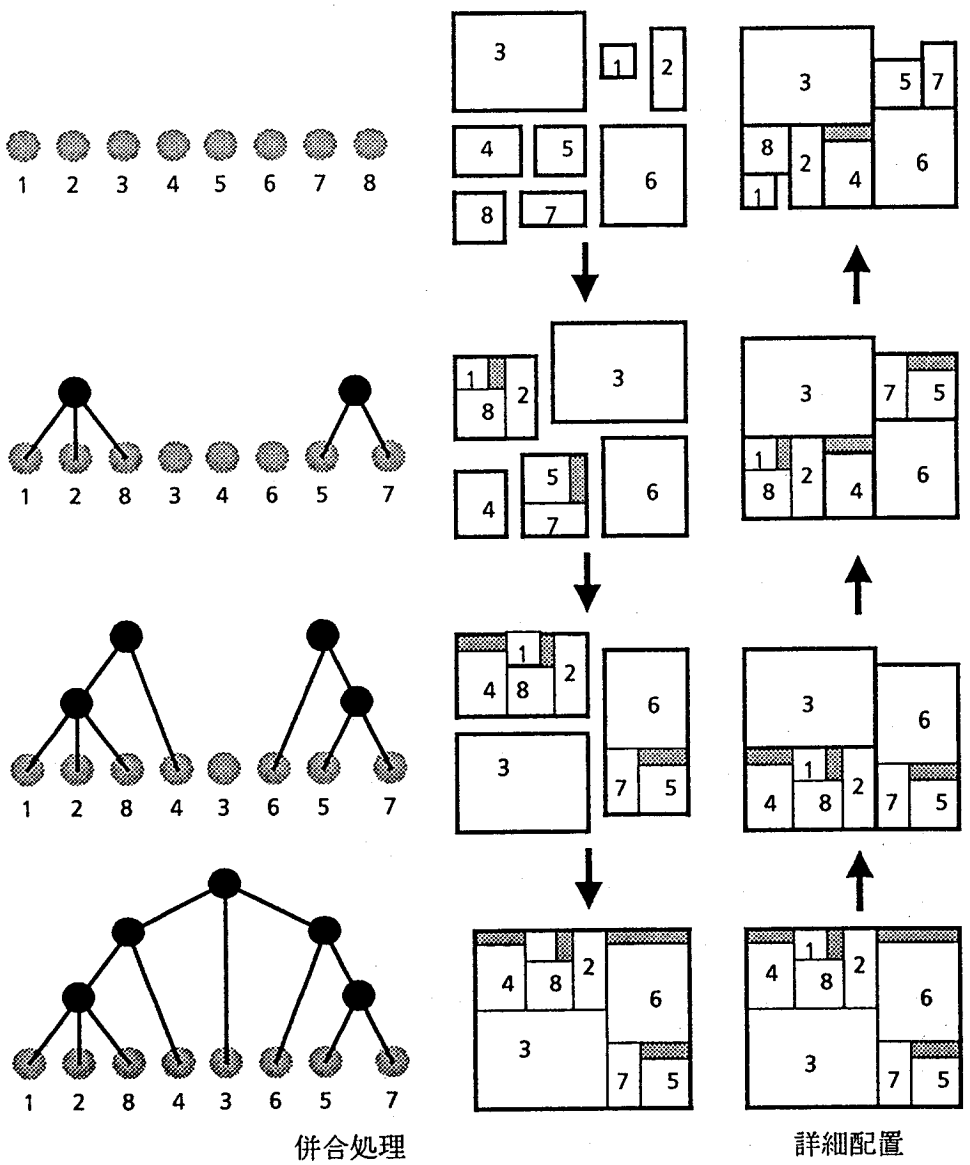


図3-4　配置手法

ルールベースによる併合処理

ルールベースによる併合処理は、記述されたルールに従ってブロック間の結合度、大きさ、形状、種類等の様々な項目を考慮したうえ、ブロック間の隣接関係を決定する。それを木構造で表現し、最終チップレイアウトの形状を見積る。ルールは、以下の3種類に分類される。(記述例を本章最後の付録に示す。)

(1) 戦略決定ルール

戦略決定ルールは配置するブロックの大きさ、種類等を評価してどの併合ルールを適用するかを決定する。

(2) 併合ルール

併合ルールは2または3個のブロックをそれらの結合度、大きさ、形状、種類等を考慮して選択し新しいブロックを生成する。この時、選ばれたブロックの中に可変ブロックが存在すれば、生成されるブロックの無効領域が最小となるようにその形状を決定する。

(3) バックトラックルール

併合処理は、結合度、大きさ、形状などを考慮して最も適切な2または3個のブロックを順次併合していくが、この結果は必ずしも最適解とならない。そこで、バックトラックルールは生成されたブロック中の無効領域や縦横比を評価して、その結果が許容範囲でなければ生成を中止し、新たな併合のプロセスに進むため、すべてのブロックの中から面積の大きなブロック、または内部結合度の小さいブロックを選択し分解する。

図3-5にバックトラックルールの動作例を示す。併合処理のある段階においてminimum clustering rule(付録参照)が適用され、あらたにブロックが1つ生成されたとする。この時、バックトラックルールの評価により、生成されたブロック中の無効領域や縦横比が許容範囲でないので生成を中止し、またもう1つのブロックも選ばれ分解される。そこで新たにBasic clustering rule(付録参照)が適用され、異なる併合のプロセスへ進む。

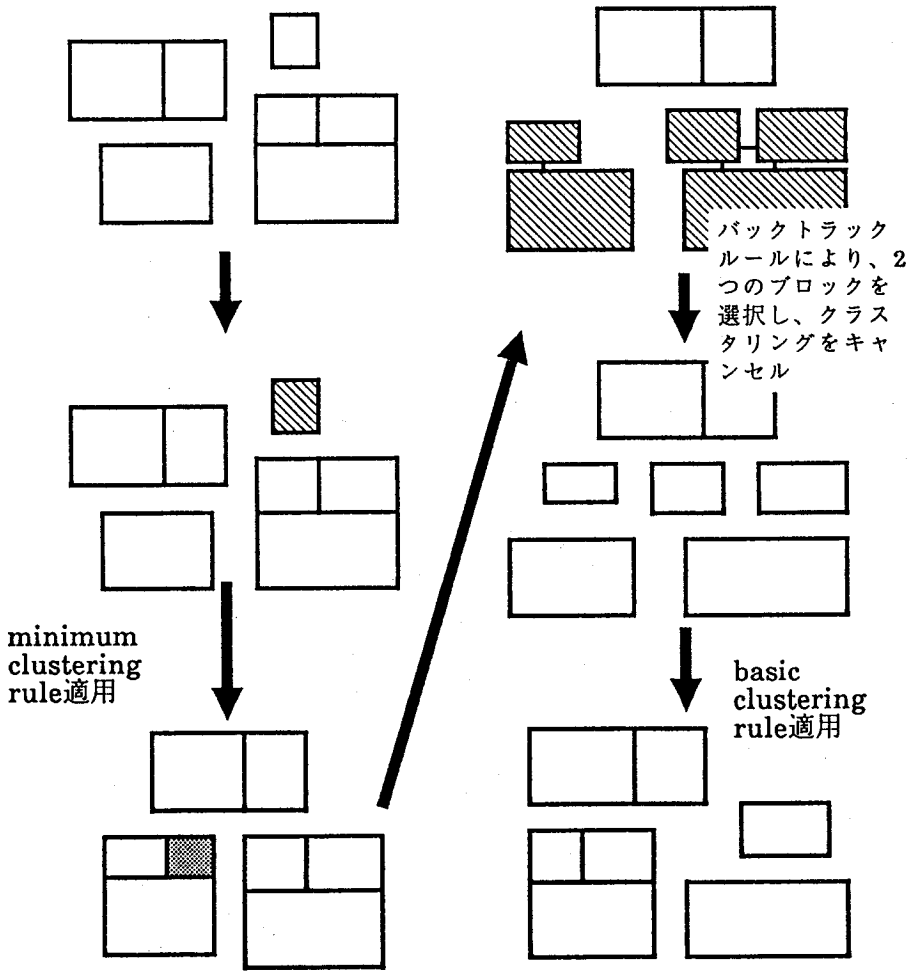


図3-5　バックトラックルールの適用例

<u>詳細配置アルゴリズム</u>

ルールベースによる併合処理によって得られた、ブロック間の隣接関係を表す木構造をトップダウンに探索し、各レベルのブロックの配置を、水平引力、垂直引力(図3-6)によって決定する。次に、固定ブロックの方向性を、仮想配線長が最小になるように決定する。ここでいう仮想配線長とは、端子間のマンハッタン距離の総和である。

図3-7にその配置手順の例を示す。3つのブロックA、B、Cの併合から新たにブロックを生成する場合、A、B、Cの配置方法は4通りに大別できる。例えば、図に示すように、配置タイプ1について詳細配置は次のようにして求められる。まずブ

$$\text{水平引力 } Fh(c) = \sum_i \sum_j (Phij - Phcj)$$

$$\text{垂直引力 } Fv(c) = \sum_i \sum_j (Pvij - Pvcj)$$

c : current module to be placed
i : all modules which connect to current module
j : all nets of current module
Phij: x position of I/O terminal with net j in module i
Pvij: y position of I/O terminal with net j in module i

図3-6　水平引力、垂直引力

ロックAとB、およびAとCとの間で水平引力が評価され、左右の位置関係が決定され、次にブロックB、Cについて垂直引力が評価され、それらの上下関係が決定される。図3-8に詳細配置アルゴリズムの記述を示す。

3.5　実験結果

表3-1に実験に用いた4種類のデータの内容を示す。ただし、データAは23個のブロックすべてが可変である場合と固定である場合の比較実験用のデータであるため、固定ブロック数や可変ブロック数は記さない。

まず、データAを用いて、トップダウンフロアプランニング手法とボトムアップフロアプランニング手法の比較を行った。表3-2に示すように、結果1はデータA中のブロックがすべて可変ブロックであると仮定し、フロアプランニング後に形状、I/O端子位置を定めた。一方、結果2は、データA中のブロックがすべて形状固定と仮定して本手法を適用したものである。結果1は結果2に対してチップ面積が約8%、総配線長で約30%減少しており、この実験からトップダウンフロアプランニング手法の優位性が定量的に確認できる。各々のレイアウトパターンを図3-9に示す。

次に本手法の性能を評価するために最適配置との比較を行った。データBは24個

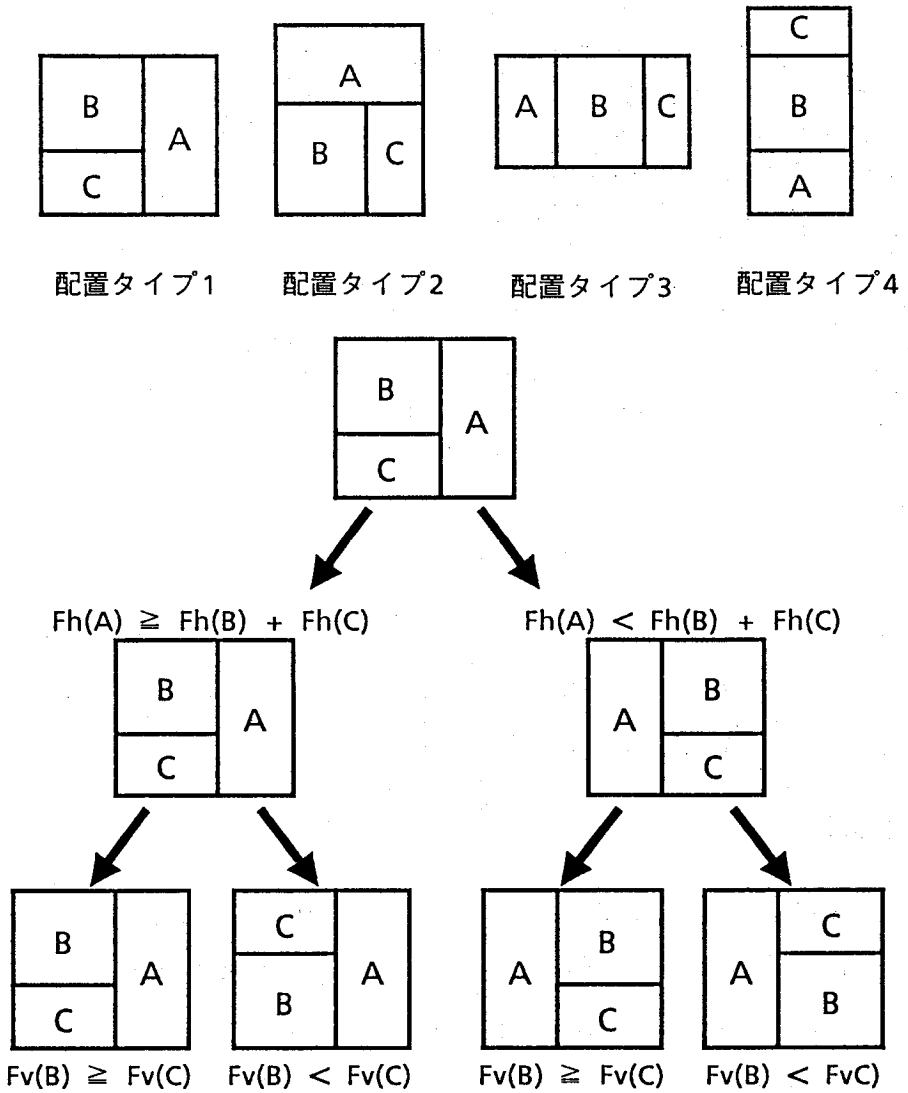


図3-7　詳細配置の例

の固定ブロックからなり、最適配置が図3-10(a)で与えられているデータである。ブロック間に引かれた線は接続要求を表し、太い実線は特に接続要求が強いことを示す。本手法を適用した結果を図3-10(b)に示す。総配線長は11%増大したが、チップ面積は最適解と一致し、太い実線で示した配線要求についての考慮は最適解とほぼ同等であり、本手法が十分実用的であると言える。結果3の配置はSUN 3/60Cワークステーション上でCPU時間で約140秒であった。

```
PROCEDURE placement ( )
BEGIN
    set the root block to S;
    WHILE ( any block in S has its son blocks )
    BEGIN
        select a block B in S, which is of the largest connection to others;
        smash B into son blocks
        and decide their positions
            according to horizontal or vertical forces;
    END;
END;
```

図3-8　詳細配置アルゴリズム

データ	ブロック数	固定 ブロック数	可変 ブロック数	ネット数
A	23	-	-	101
B	24	24	0	159
C	11	2	9	425
D	6	3	3	368

表3-1　実験データ

データ	チップ面積	総配線長
結果 1	35,441×34,561	2,441,993
結果 2	36,984×36,041	3,570,653

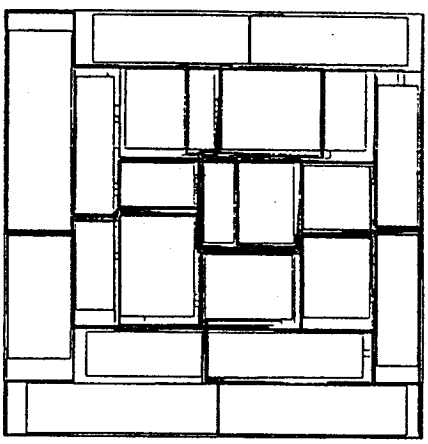
表3-2　実験結果

中間階層のフロアプランニングを行う場合、配置した結果のブロック形状が問題となる。データCにつき、縦横比を指定する実験を行った。図3-11は縦横比を 1.0 と 0.7 に指定した場合のレイアウト結果である。

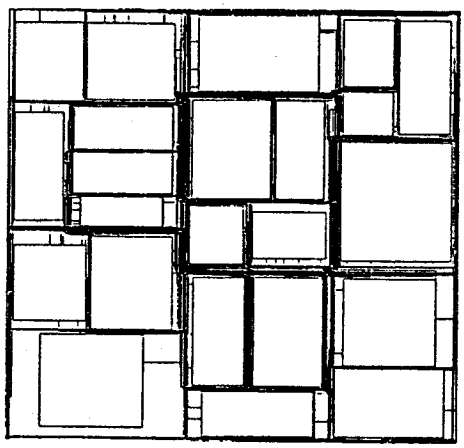
図3-12はデータDに対するチップレイアウトの結果であり、内部をスタンダードセル方式でレイアウトされる 3 個の可変ブロックと、すでに人手設計済みの 3 個の固定ブロックからなる。

3.6 結言

本章では、提案するフロアプランニングシステムの概要を述べ、特にトップダウン設計におけるブロック配置手法について詳細に述べた。本ブロック配置手法は、AI技法と組合せ算法を組み合せて用いることにより、多様な評価関数を考慮した配置最適化が可能であり、実験によりその性能を確認した。

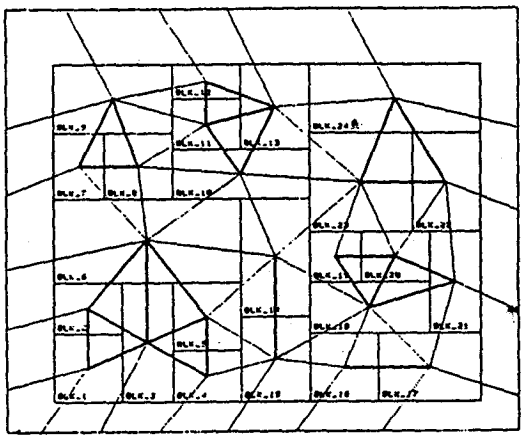


(a) 結果1

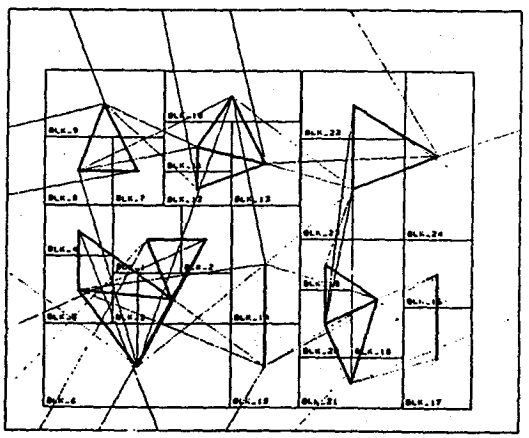


(b) 結果2

図3-9 データAのレイアウト結果

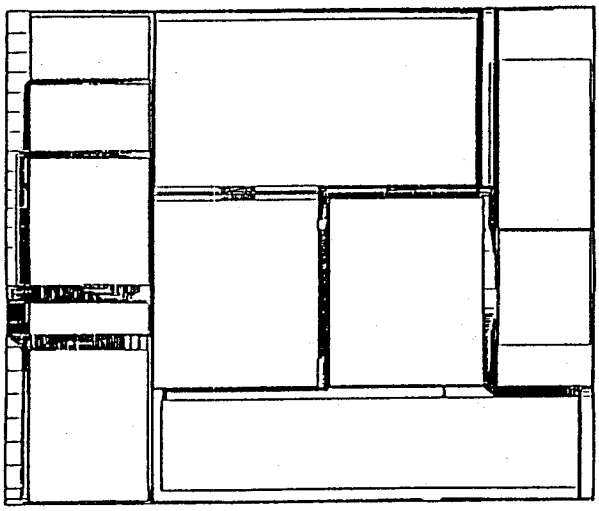


(a) 最適フロアプラン

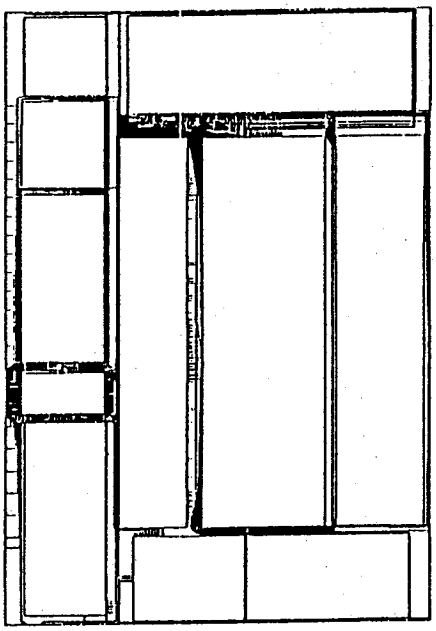


(b) 本手法によるフロアプラン結果

図3-10　データBのフロアプラン結果



(a) 縦横比 1.0



(b) 縦横比 0.7

図3-11 データCのフロアプラン結果

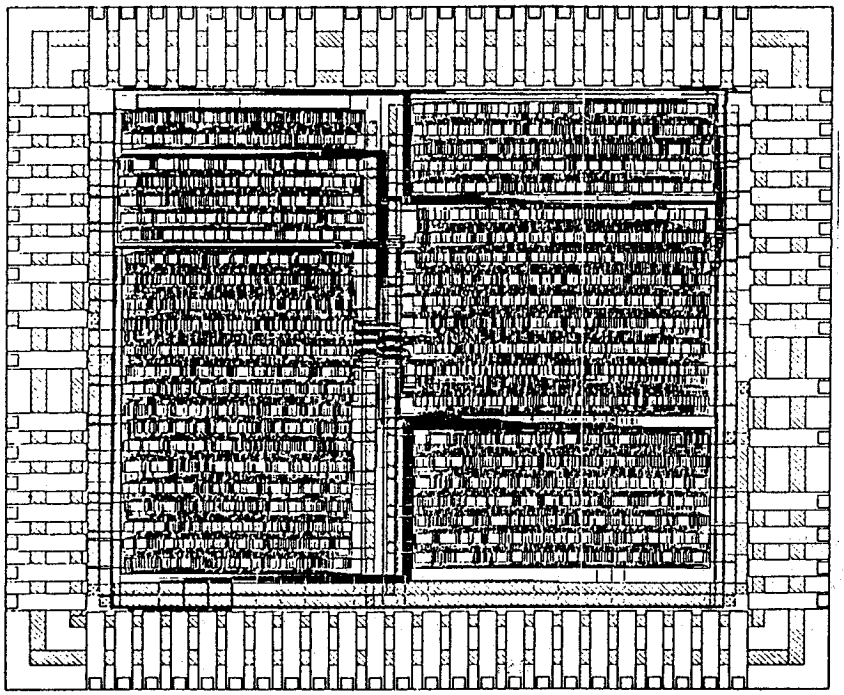


図3-12 データDに対するフロアプラン結果

[付録]

ルールベースの併合処理のおけるルール記述例を以下に示す。

(1) 戦略決定ルール

[Strategy Rule 1]
IF the rate of a minimum block to a maximum one is greater than a prescribed value,
THEN apply minimum block clustering rules.

[Strategy Rule 2]
IF there are more than two analog blocks,
THEN apply analog block clustering rules.

[Strategy Rule 3]
IF neither strategy rule 1 nor strategy rule 2 is selected,
THEN apply basic clustering rules.

(2) 併合ルール
[Basic Clustering Rule 1]
IF each of the three connection numbers among three blocks is grater than the mean value, and the area of dead spaces and the aspect ratio of a block to be obtained by clustering these three blocks are the best of all,
THEN group them into a new block, and estimate its shape from its placement types.

[Basic Clustering Rule 2]
IF any two of the three connection numbers among three blocks are both greater than the mean value, and the area of dead spaces and the aspect ratio of a block to be obtained by clustering these three blocks are the best of all,
THEN group them into a new block, and estimate its shape from its placement types.

[Basic Clustering Rule 3]

IF the connection number between two blocks is greater than the mean value, and the area of dead spaces and the aspect ratio of a block to be obtained by clustering these two blocks are the best of all,

THEN group them into a new block, and estimate its shape from its placement types.

[Basic Clustering Rule 4]

IF the area of dead spaces and the aspect ratio of a block to be obtained by clustering these three blocks are the best of all,

THEN group them into a new block, and estimate its shape from its placement types.

[Basic Clustering Rule 5]

IF the area of dead spaces and the aspect ratio of a block to be obtained by clustering these two blocks are the best of all,

THEN group them into a new block, and estimate its shape from its placement types.

[Minimum Clustering Rule 1]

IF each of the three connection numbers among the minimum size block and two others is greater than the mean value, and the area of dead spaces and the aspect ratio of a block to be obtained by clustering these three blocks are the best of all.

THEN group them into a new block, and estimate its shape from its placement types.

[Minimum Clustering Rule 2]

IF each of the two connection numbers between the minimum size block and another one is treater than the mean value, and the area of dead spaces and the aspect ratio of a block to be obtained by clustering these two blocks are the best of all,

THEN group them into a new block, and estimate its shape from its placement types.

[Minimum Clustering Rule 3]

IF any of the three connection numbers among the minimum size block and two others is greater than the mean value, and the area of dead spaces

and the aspect ratio of a block to be obtained by clustering these three blocks are the best of all,

THEN group them into a new block, and estimate its shape from its placement types.

[Minimum Clustering Rule 4]

IF the area of dead spaces and the aspect ratio of a block to be obtained by clustering the minimum size block and two others are the best of all,

THEN group them into a new block, and estimate its shape from its placement types.

[Minimum Clustering Rule 5]

IF the area of dead spaces and the aspect ratio of a block to be obtained by clustering the minimum size block and another one are the best of all,

THEN group them into a new block, and estimate its shape from its placement types.

[Analog Clustering Rule 1]

IF each of the three connection numbers among three analog blocks is greater than the mean value, and the dead spaces and the aspect ratio of a block to be obtained by clustering these blocks are the best of all,

THEN group them into a new block, and estimate its shape from its placement types.

[Analog Clustering Rule 2]

IF any two of the three connection numbers among three analog blocks are both greater than the mean value, and the area of dead spaces and the aspect ratio of a block to be obtained by clustering these three blocks are the best of all,

THEN group them into a new block, and estimate its shape form its placement types.

[Analog Clustering Rule 3]

IF the connection number between any two analog blocks is greater than the mean value, and the area of dead spaces and the aspect ratio of a block to be obtained by clustering these two blocks are the best of all,

THEN group them into a new block, and estimate its shape from its placement types,

[Analog Clustering Rule 4]

IF the area of dead spaces and the aspect ratio of a block to be obtained by clustering three analog blocks are the best of all,

THEN group them into a new block, and estimate its shape from its placement types.

[Analog Clustering Rule 5]

IF the area of dead spaces and the aspect ratio of a block to be obtained by clustering two analog block are the best of all,

THEN group them into a new block, and estimate its shape from its placement types.

(3) バックトラックルール

[Backtracking Rule 1]

IF a new block includes dead spaces, the total area of which exceeds a prescribed value, and the rate of a minimum block to a maximum one is greater than a specified value,

THEN cancel the clustering and decompose the maximum block into subblocks.

[Backtracking Rule 2]

IF a new block includes dead spaces, the total area of which exceeds a prescribe value, and the internal connection number of the block is the smallest of all,

THEN cancel the clustering and decompose the block into subblocks.

[Backtracking Rule 3]

IF the aspect ratio of a new block is not in the allowable range, and the internal connection number of the block is the smallest of all,

THEN cancel the clustering and decompose the block into subblocks.

[Backtracking Rule 4]

IF the aspect ratio of a new block is not in the allowable range and the rate of a minimum block to a maximum one is greater than a prescribed value,

THEN cancel the clustering and decompose the block into subblocks.

以上

第4章　マクロセル方式自動配線手法

4.1　緒言

本章では、マクロセル方式自動配線設計のための詳細配線手法について考察する。これは、接続要求を満たすための初期配線とデザインルールを満たすための配線改善との、2つの処理工程に分けることにより、問題を簡単化するとともに高密度配線の実現を図っている。さらに、幾つかの実験によりその性能を評価する。

4.2　マクロセル方式レイアウトにおける配線問題

マクロセル方式の配線設計は、通常は概略配線と詳細配線との2段階に分けて行う。概略配線では、全体の配線領域をいくつかの領域に分け、各々のネットの経路をこれらの領域に割当てる。一方、詳細配線では、各領域ごとに割り当てられたネットについてチャネルルータ[37-38]を用いた配線設計を行う。これはチャネルルータが100%配線を保証し、効率の良い配線が可能だからである。さらに、最近のチャネルルータではグリッドレス配線、任意幅の配線、多層化、凹凸境界線等に対応した手法が提案され[39-40]、さらに高機能化、高密度化が進んでいる。

しかしマクロセルレイアウトではセルの大きさにばらつきがあり、配線領域が複雑な形状となるため、従来のチャネルルータだけでは配線設計が不可能あるいは無効領域が不必要に生じる場合がある。その解決策として、スイッチボックスルータの適用が有効と考えられる。スイッチボックスルータは、これまでにいくつか手法が提案されているが[41-43]、それらのほとんどはグリッドを用いた単純なモデルでの配線であり、現実にはより複雑なグリッドレスモデルが求められている。

本章では、チャネル配線問題を初期配線処理と配線改善の2つの処理工程に分割して考察する。これにより、問題の簡単化を図られ、機能的に柔軟性のあるモデルを扱うことができる。さらにグリッドレス配線が実現され、効率的な配線が可能となる。

以下ではまず上記の手法を用いたマクロセル方式自動配線手法の概要について述べ、次に凹凸の少ないチャネルの定義法を述べる。最後に、マクロセル用チャネル

ルータ(以下、RCルータという)について2つの処理の概要および配線改善の詳細を述べる。

4.3 マクロセル方式自動配線手法の概要

以下にマクロセル方式自動配線手法の概要について述べる。まず、レイアウトモデルを図4-1に示す。各マクロセルの形状は矩形でその大きさと形状は任意である。また周辺には接続のための端子が存在する。ただし、ここでは配置の自由度を高め、無効領域の削減を図るため非スライシング構造をも扱う[44]。なお、マクロセル配置は、フロアプランニングにおいて最適化されており、各マクロセル間には、配線に必要な幅が確保されているとする[45]。

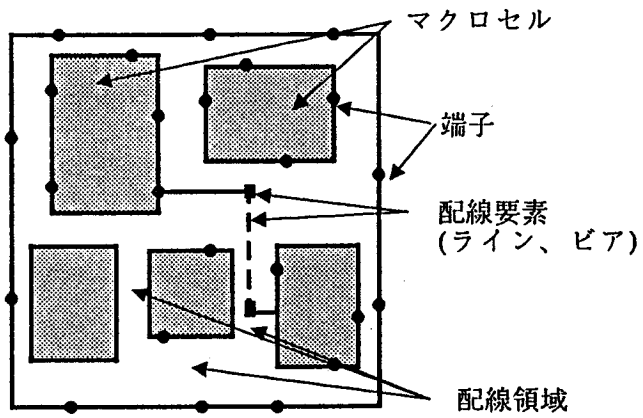


図4-1 マクロセル方式のレイアウトモデル

<u>処理手順</u>

マクロセル方式自動配線手法は、概略配線と詳細配線からなる二段階配線手法を用いており、処理の流れを以下に示す。

1) 概略配線

フロアプランニングシステムで作成された経路探索グラフを用いてコスト最小迷路法[46]により各配線の概略経路を決定する。

2) チャネルの決定と順序付け

チャネル配線を適用する範囲である配線領域を定め、チャネル配線の順序を決定する。処理の詳細は、次の4.4節で述べる。

3) チャネル配線

2)で得られた各領域について、4.5節に示すRCルータによりチャネル配線を行う。

4.4 チャネル決定

非スライシング構造をも扱ったチャネルの決定手法は、これまでに幾つか発表されている[47-48]。しかし、これらでは、従来のチャネル配線を行うことを前提としており、チャネル形状については考慮されてなく、既に述べたようにその凹凸により無駄な領域が発生しやすい。そこで、より効率のよい配線結果を得るため、以下の二点に注目する。

第一に、チップ面積を決定する重要な要因は、チャネルポジショングラフ[47]におけるクリティカルパス上のチャネル幅である。このため、それを配線領域に含むチャネルは、チャネル長を短くして、他の配線領域部分からの影響による配線領域幅の拡大を防ぐ必要がある。なお、チャネル長とは、配線領域として定義された領域の幹線方向の最大長である。

第二に、チャネルの十字形の交差部分(図4-2)やL形のチャネルの折れ曲がり部分(図4-3)などでは、配線領域の形状を考慮しないでチャネルを決定すると、ブロックの位置から凹凸の大きなチャネルが決定され、その配線領域は増大する。

以上のことから、本研究では、フロアプラングラフにもとづくチャネル決定及び順序付けは、文献[47]の手法に対してクリティカルパス情報および配線領域の形状をさらに考慮に入れるべく改良したものを用いる。

まず、チャネル領域の幾何学的な形状を定義する。ここでは簡単のために水平な直線チャネルに注目して説明する。いま注目するチャネルに面するマクロセルの集合をBとする。このチャネルに下辺および上辺が面しているマクロセルの集合をそれぞれB_{TOP}およびB_{BOTTOM}とする。$B_{TOP(BOTTOM)}$中にあるマクロセルiにおいて

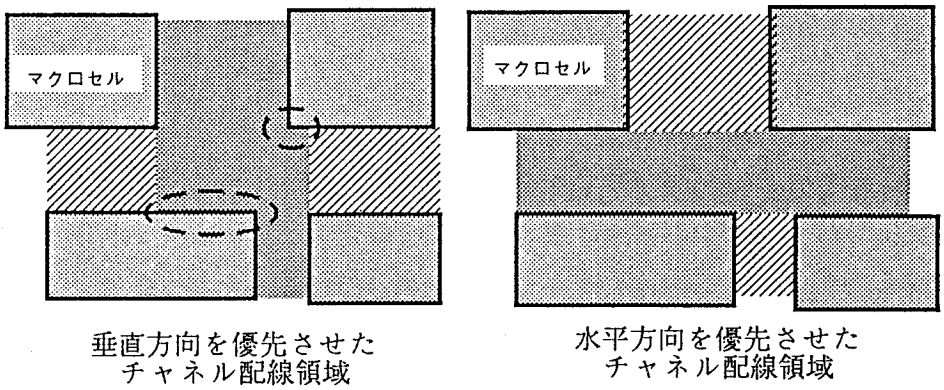


図4-2　チャネルの十字変換

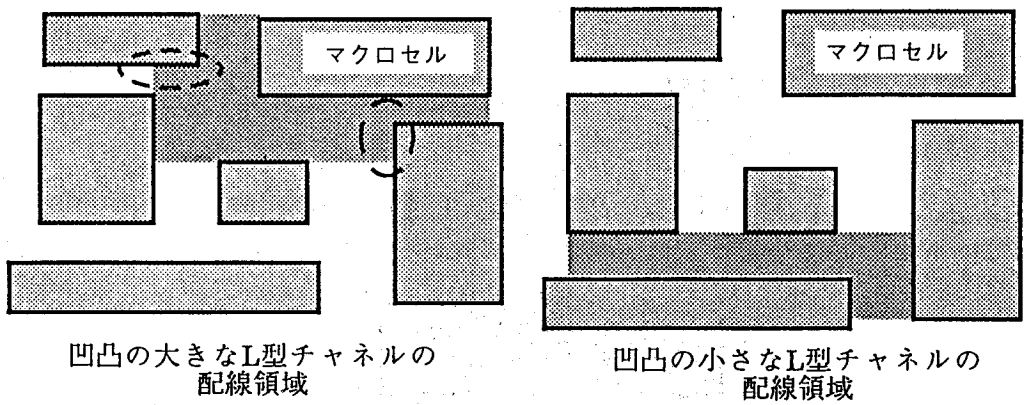


図4-3　L型チャネル

チャネルに面する辺の左右端の座標を$(X^i_{T(B)L}, Y^i_{T(B)})$, $(X^i_{T(B)R}, Y^i_{T(B)})$とする。また、チャネル領域の左下及び右上の座標を$X_{BL}, Y_B, X_{TR}, Y_T$とすると、最小チャネル領域を任意のセル$i \in B$について次のように定義する。

$$X_{BL} = \min(X^i_{BL}, Y^i_{T(B)})$$

$$Y_B = \max(Y^i_B)$$

$$X_{TR} = \max(X^i_{BR}, X^i_{TR})$$

$$Y_T = \min(Y^i_T)$$

また、最大チャネル領域を任意のセル$i \in B$について次式で表す。

$$X_{BL} = \max(X^i_{BL}, X^i_{TL})$$

$$Y_B = \min(Y^i_B)$$

$X_{TR} = \min(X^i_{TR})$

$Y_T = \max(Y^i_T)$

図4-4(a)は最小チャネル領域の例であり、図4-4(b)は最大チャネル領域の例である。従来のチャネルルータは最小チャネル領域を用いるが、本章のチャネルルータは最大チャネル領域を用いる。これにより、配線の迂回や無駄領域を削減することができる。

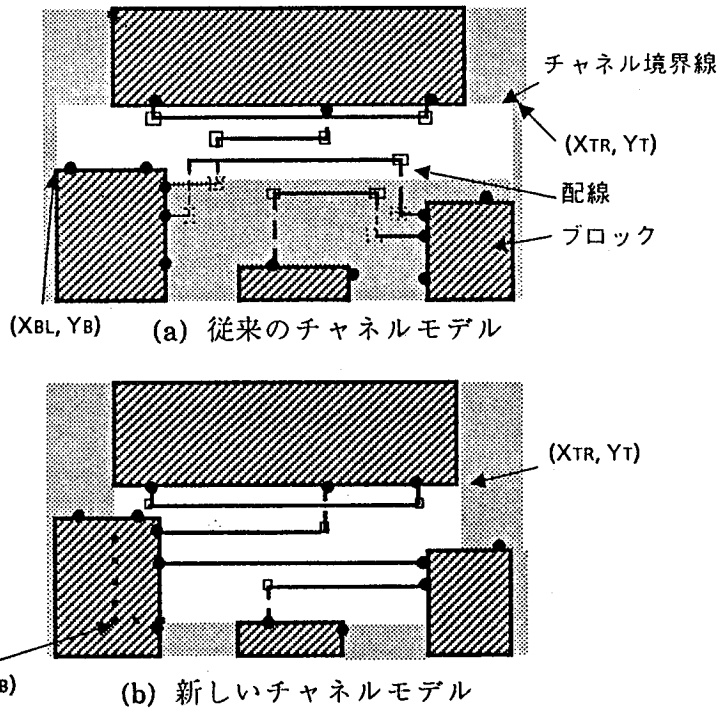


図4-4　チャネルモデル

さらに、チャネルを決定する時、チャネル形状の凹凸を定量的に考慮するために次式を定義する。

$IR = \{\Sigma(Y^i_T - T) + \Sigma(B - Y^i_B)\} / L$

ここで、$T = \min(Y^i_T)$, $B = \max(Y^i_B)$, $L = T - B$

本式はチャネル形状に凹凸が大きいほど大きな値を持つので、チャネル決定アルゴリズムでは2個以上の候補が存在する場合IRの値がより小さいものを選択する。このIRを用いて凹凸の最小化を図るチャネル定義アルゴリズムを図4-5に示す。

```
(The channel definition algorithm)
WHILE channel candidate exists
   Search the candidate for wall slicing.
   IF there are some candidates C,
      THEN
         IF there are candidates C' which have no critical
             subchannel,
            THEN C=C'
         END IF
         choose one for which IR is minimum from C.
         and define it a straight channel.
      ELSE search the totally independent candidate
            for corner cutting.
         IF there are any candidates C,
            THEN
            IF there are candidates C' which have no critical
                subchannel,
               THEN C=C'
            END IF
            choose one for which IR is minimum from C.
            ELSE search vertical or horizontal independent
                  candidates.
               IF there are candidates C' which have no
                  critical subchannel,
                  THEN C=C'
               END IF
               choose one for which IR is minimum from C.
         END IF
         define an L-shape channel.
   END IF
END WHILE
```

図4-5　チャネル定義アルゴリズム

4.5 RCルータ

本手法は、1)すべてのネットに対して位相と層を決める初期配線、2)配線の絶対位置を決める配線改善の2つの処理工程からなる。詳細配線問題は、NP完全である事

がすでに知られている[49]。そこでここでは、初期配線では配線トポロジーの最適化を、配線改善ではすべての配線要素がデザインルールを満たす事を目的関数とし、問題に適したヒューリスティック算法をそれぞれ適用し、全体の最適化を図る。

4.5.1 初期配線

初期配線の目的は、すべてのネットについて配線経路を見出すことであり、基本的には各ネットの相対位置と配線とビアの層を決める処理である。この段階ではシンボリックパターンの配線経路、すなわち配線の中心線、端子、およびビアの中心点が連結され、他のネットの経路と短絡さえしなければよく、各層でのデザインルールを無視する。例えば図4-6の例では、実パターンでは、ネットAとネットBは重なっているが、シンボリック図では同層での重なりはないため、初期配線の条件を満たしている。

シンボリックパターンの配線経路を採用することにより処理は簡単になるが、最終的な配線品質を考えた場合、この段階でなるべく最終的に定まる配線経路に近い位置を求めておいたほうが望ましい。そこで、以下では適当な間隔の仮想グリッド上で配線を行い、各配線要素(配線、ビア)の初期配置を決める。グリッド上にない端子は他の端子と短絡しない最も近いグリッドに引き出しておき、この仮想グリッド上で従来のスイッチボックス配線手法[42]を適用する。もし未配線があれば必要なだけグリッドを増やすことにより配線経路を見出し、未配線が無くなるまでこれを繰り返す。図4-7は、ネットAが未配線となった場合のグリッドの追加を示している。

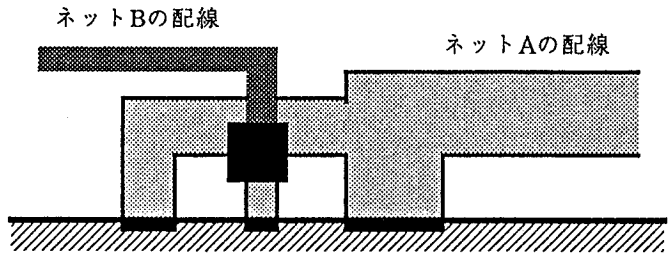
4.5.2 配線改善

配線改善は、初期配線の結果得られた配線接続を保ちながら配線に折れ曲がりを随時挿入しつつ、配線要素を縦方向または横方向に移動することによって、デザインルールに違反している部分を取り除く処理である。

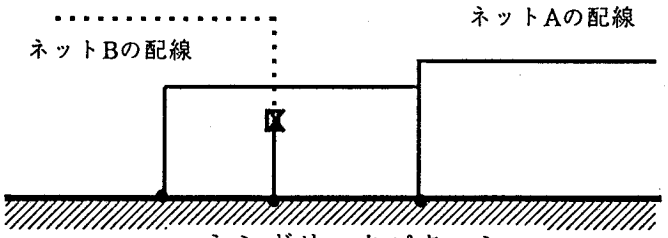
処理手順は以下のとおりである。

1) 配線移動の実行回数と方向を設定する。

2) 配線移動を実行する。

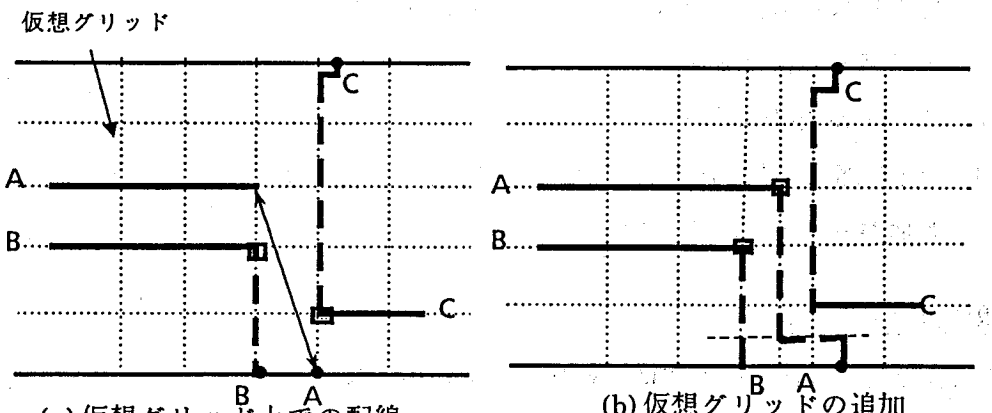


実パターン



シンボリックパターン

図4-6　初期配線



(a) 仮想グリッド上での配線　　(b) 仮想グリッドの追加

図4-7　グリッドの追加

3) デザインルールに対する違反を調べる。違反がなければ5)へ。

4) 実行回数が指定回数に達していれば、配線不可能として、6)へ。
それ以外は、方向を90度回転して設定し、2)へ。

5) 配線結果出力

6) 処理終わり

処理手順2)の一次元の配線移動は、チャネルコンパクション手法[50-51]を基本とした処理を行っている。チャネルコンパクションとはチャネル配線において、チャネル幅を可能なかぎり縮小することをいう。本手法は、これにスイッチボックス問題への適用及び多角形領域へ拡張のための操作を加えることによりグリッドレス配線を実現し、かつ一次元の配線移動を4方向に適用することによりデザインルール違反を徐々に取り除いていくという方法を採用している。以下にこれに関する操作について記述する。ここでは配線移動の方向はすべて下方向の場合を考えることとする。

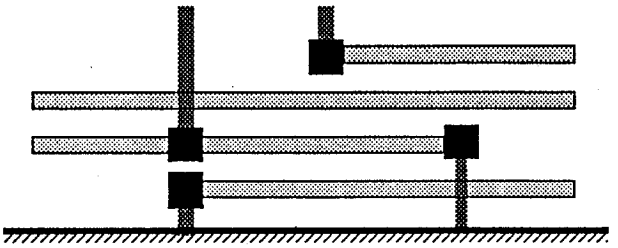
配線移動処理

1) チャネル配線結果に対して、幹線(ビアを含む)をy座標に関して昇順にソーティングする。
2) 下方の幹線から順に、次の操作を行う。下端のチャネル境界線または下方の他の幹線との間にデザインルール違反が起こらない最も下の位置に配線経路を下げる。(図4-8(b))
3) チャネル幅を決定する。
4) 2)と逆順に、各配線要素に対して、可能ならば配線経路を直線化し、不要な折れ曲がりを取り除く。(図4-8(c))

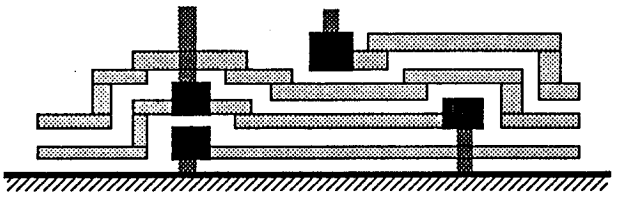
配線要素の位置決定

配線要素の位置決定の操作は、移動する側に境界線を設定して行う。配線要素は、この境界線との間でデザインルールを満たす範囲内で移動可能である。本手法は、多層化に対応するためこの境界線を各層ごとに持っている。処理前に領域境界線の最も下の辺を各層の境界線として設定し、配線要素の移動により境界線を順次更新する。

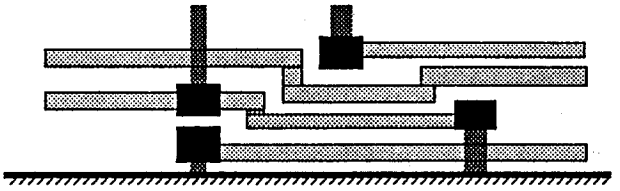
例えば図4-9の場合、配線要素の位置決定操作のある段階において、次に移動の対象となる配線要素をネットkとする(図4-9(a))。図4-9(b)はネットkのa層の配線を移動させた図である。移動した配線の輪郭線は、a層境界線として新たに加わる。この



(a)チャネル配線結果



(b)下方向へネットを沿わせた結果



(c)配線の直線化

図4-8　配線改善

時、更新された境界線情報には、配線のネット名と1つ前の境界線情報も記憶しておく。ネットkのビアは、ビアが関係する2つの層境界線を参照して、各境界線に対してデザインルールを満たす最も下の位置に移動する。図4-9(c)のビア移動では、a層境界線上に同ネットの境界線があるため、その部分だけ1つ前の境界線が参照される。移動後、2つの層の境界線が更新される。図4-9(d)のb層境界線の場合も、同ネットの境界線は1つ前の境界線が参照され、ネットkのb層配線が移動する。なおビアとの配線は、後処理により追加される。

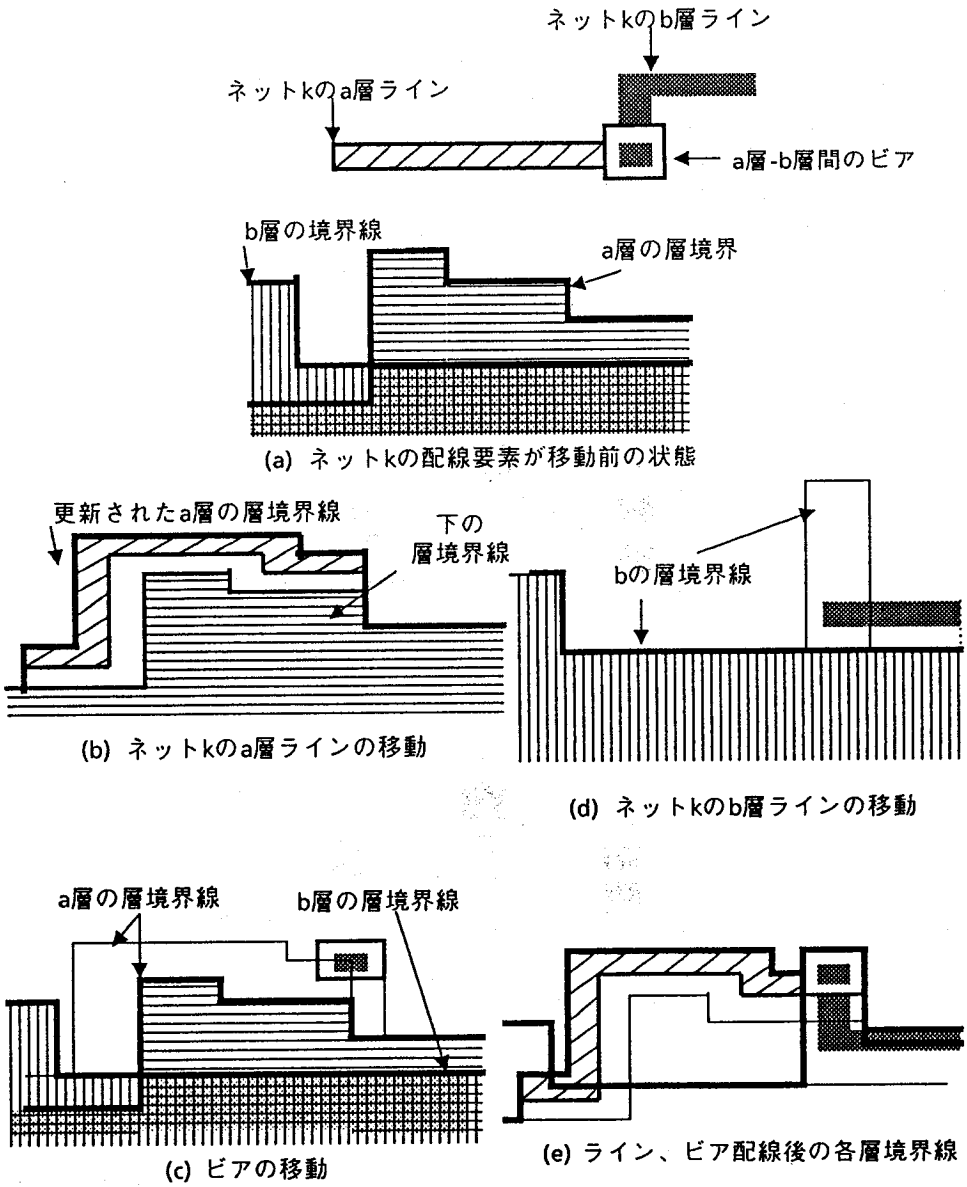


図4-9　ネットkの配線要素移動による層境界線の更新

領域境界線付近の配線移動処理

一般にチャネルコンパクション手法は、すべての要素のy座標は固定されていないという制限がある。本モデルでは垂直辺上の端子に接続されている配線経路が存在することや固定された領域内に配線経路を引かなければならないことから、この制限を満たしていない。このため処理の途中において続行が不可能となる場合があ

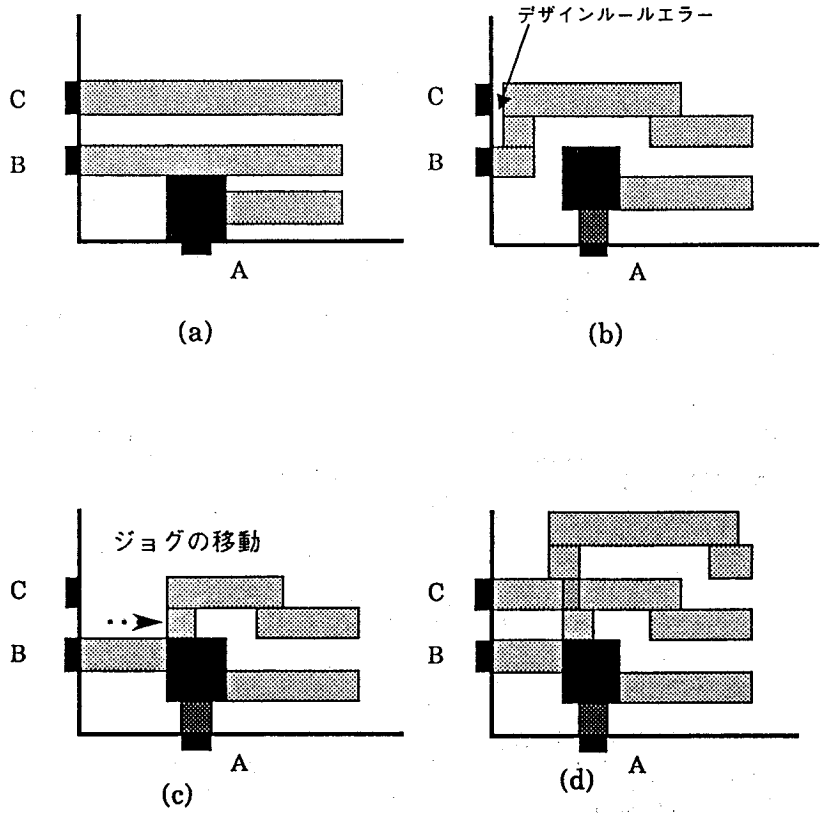
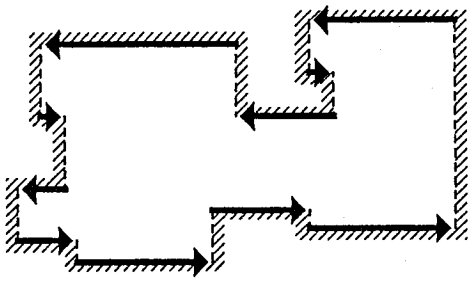


図4-10　配線領域の境界線付近の処理

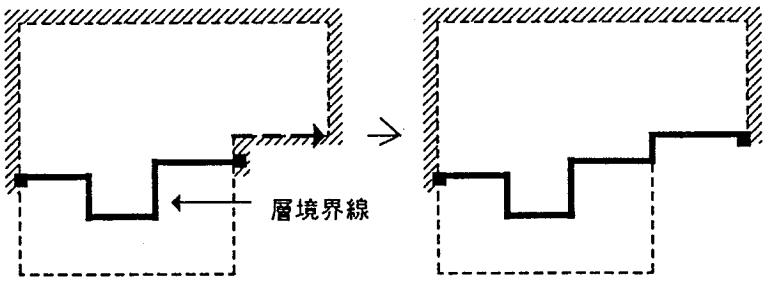
る。例えば図4-10の場合、ネットBの配線を移動した時点で、領域境界線の垂直辺とネットBの間にデザインルール違反が生じ、かつネットCが配線できなくなってしまう(図4-10(b))。これを防ぐために本手法では、折れ曲がりと垂直領域境界線との間にデザインルール違反が生じた場合、配線要素間の相対位置が変わらない範囲で、その領域境界線と最も離れた位置に折れ曲がりを移動する(図4-10(c))。この部分のデザインルール違反は他方向の配線移動の処理により取り除かれる。また配線要素が配線領域から溢れてしまう場合は、デザインルールを無視して、配線要素を領域内に移動し、後に行われる他方向の配線移動によりこの違反を解消させる。

多角形領域対応

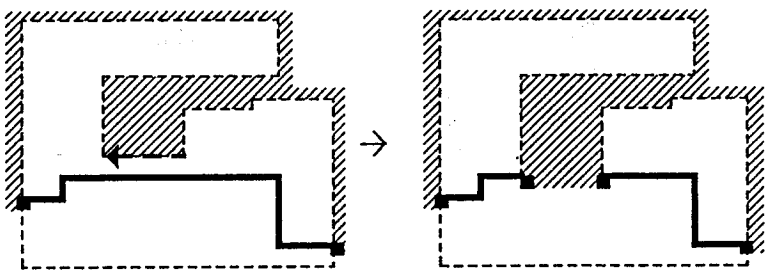
配線領域が多角形の場合、一般に層境界線には凹凸が存在する。これに対処するた



(a) 配線領域の境界線の方向づけ



(b) 右向き境界線の更新



(c) 左向き境界線の更新

図4-11　多角形領域での層境界線の更新

めに本手法では、配線領域境界線の各水平線分を1つの配線要素とみなして、すべての層境界線の更新処理の対象とする。しかし、配線領域境界線は移動処理の対象とはできないことから他の配線要素とは異なり、移動不可能とする。また層境界線の更新処理のために方向性を与える(図4-11(a))。この方向性は、領域境界線に沿って反時計回りの方向を与え、通常の配線要素と共にy座標でソーティングしておく。右向き領域境界線は上側が配線領域であるため、層境界線の更新対象に領域境界線が含まれる時、各層境界線の追加、延長または併合を行い、左向きの領域境界線は下側が配線領域であるため、層境界線の更新対象に領域境界線が含まれる時、各層境界線の削

除、縮小または、分割を行う。

図4-11(b)は配線要素を配置している途中段階の1つの層境界線であり、次に領域境界線が層境界線更新の対象となる例である。この領域境界線は、右向きであるため、層境界線として追加される。また図4-11(c)では、左向きであるため、その位置の層境界線が削除されている。なおこの処理はすべての層について行われる。

配線領域内に障害物がある場合は、障害物境界線に時計まわりの方向を与えて水平線分を取り出し、同様な処理を行う。

4.6 実験結果

配線移動処理を3種類のテストデータに対して適用した。

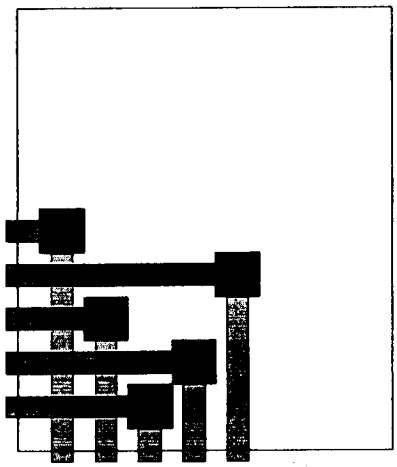
図4-12のデータは、配線領域のコーナー部において端子間隔が配線の最小間隔に等しい場合を想定したものである。本手法の適用により、4回の配線移動処理ですべてのデザインルール違反を除去することができた。これは従来のスイッチボックスルータでは配線不可能な例である。本手法では領域境界線付近でデザインルール違反が途中生じるが、他方向の配線移動処理により解消されている。処理時間は、HP9000/850(6MIPS)のCPU時間で4.5秒であった。

図4-13のデータは、スイッチボックス配線問題でよく用いられる例題[50]に対して端子間隔を小さくする変更を加えたものである。各層の配線の線幅:2メッシュに対して、ビアサイズ:4x4メッシュ、デザインルールが各層:2メッシュ、端子の中心間隔:5メッシュとした。つまり、ビアの中心間隔は6メッシュ必要である。初期配線では隣接するビア間にすべてデザインルール違反がある。本手法の適用により、4回の配線移動処理で解を得られ、従来手法より小さい領域で配線が可能であることが判明した。CPU時間は、73.4秒であった。

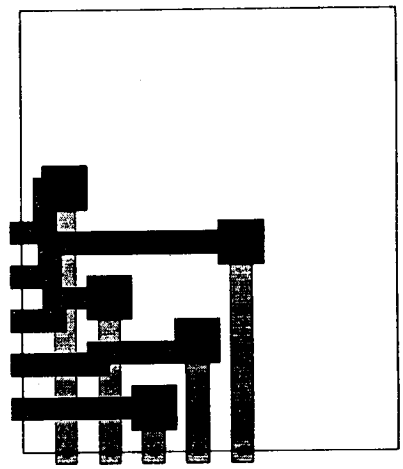
図4-14のデータは、配線領域に凹凸があり、ネット数26、端子数66、2層配線のデータである。本手法の適用により、4回の配線移動処理で解が得られた。CPU時間は29.2秒であった。

4.7 結言

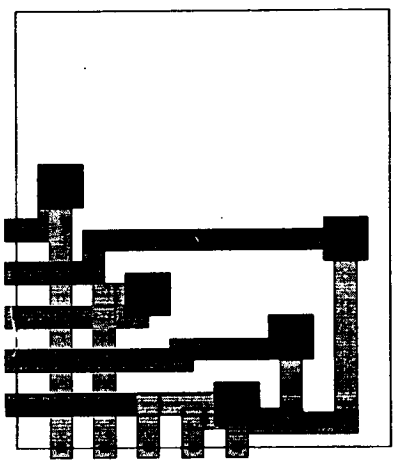
本章では、マクロセル方式自動配線手法としてチャネル定義と詳細配線手法について述べた。特に詳細配線では、接続要求を満たすための初期配線とデザインルールを満たすための配線改善の、2つの処理に分けて実行することにより、より高品質な配線経路が効率的に得られることがわかった。



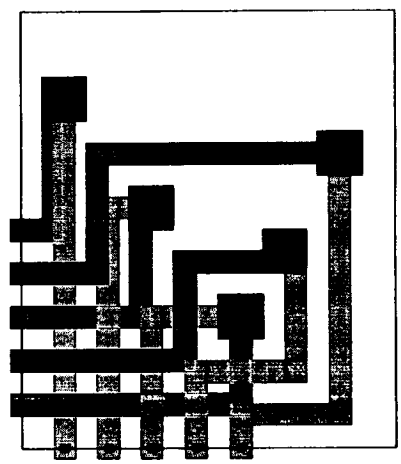
(a)　初期解



(b)　途中経過1

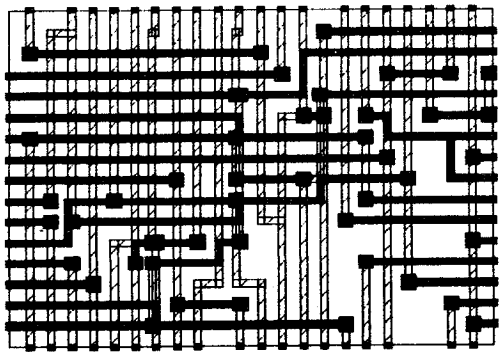


(c)　途中経過2

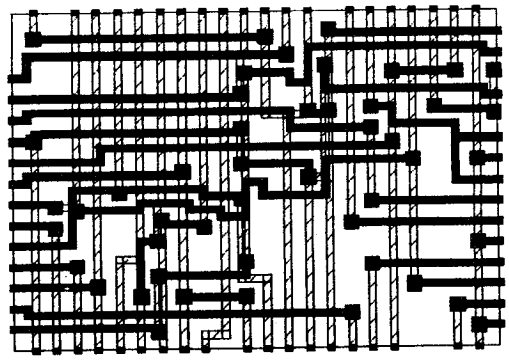


(d)　配線移動終了後

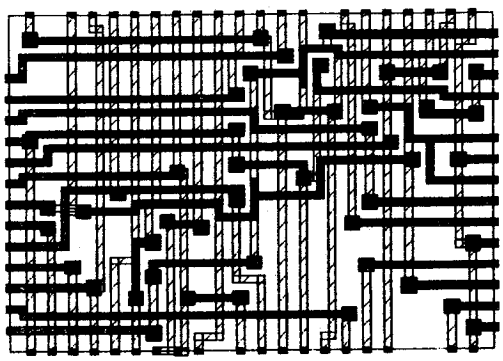
図4-12　コーナー部の配線例



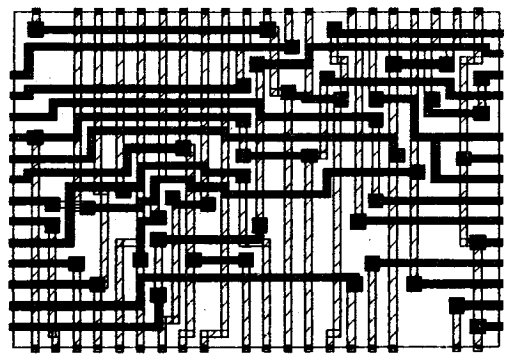
(a)　初期解



(b)　途中経過1

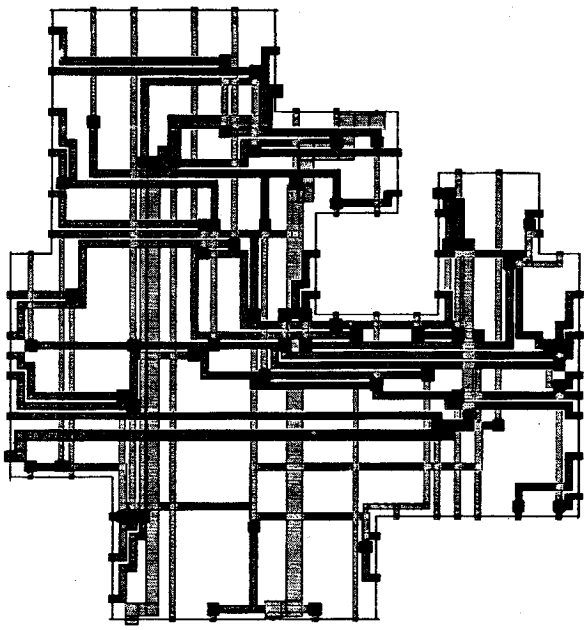


(c)　途中経過2

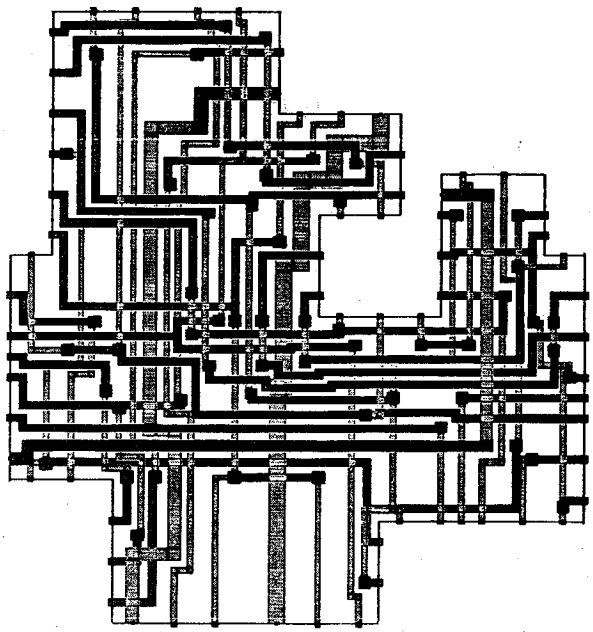


(d)　配線移動終了後

図4-13　difficult switchbox問題の配線例



(a)　初期解



(b)　配線移動終了後

図4-14　多角形領域での配線例

第5章　スタンダードセル方式セル配置手法

5.1. 緒言

スタンダードセル方式レイアウトにおいて、メタル2層配線を行う場合、セル上配線や電気的等価端子を利用した配線設計を行うことによってチップ面積を削減することが可能となる。この点に着目し、総配線長を最小化しつつ、セル上配線を効率的に利用するために、ネットごとの配線要求の分布がセル行に対して相対的に垂直方向に長くなるよう、セル配置を求めるヒューリスティック手法について考察し、幾つかの実験によりその性能を確認する。

5.2　スタンダードセル方式におけるセル配置問題

スタンダードセル方式VLSIは、設計期間の短縮、製造コストの低減の利点を持つため、その需要は依然として増大傾向にある。しかし、ますます進行する多機能化、大規模化に伴ってチップ面積や配線長の増大、処理時間の長大化、メモリなどの既設計機能ブロックを含むレイアウト設計への対応、さらに電気的特性の考慮など、レイアウトシステムには多くの課題がある。

スタンダードセル方式の配置問題を考えるために、図5-1に示すようなレイアウトモデルを導入する。

1) 配置領域の外形は矩形で、その4辺上に外部入出力端子を持つ。
2) 配置されるセルは、ほぼ同じ高さであるがその幅は任意とする。また上下辺には接続のための電気的に等価な入出力端子を持つ。また、各セルには、セル上配線禁止領域が定義されており、その他の部分は2層メタル配線で通過可能である。
3) セルは何行かのアレイ状に配置され、隣接する行間はチャネルと呼び、セル間の垂直領域と共に配線領域として用いる。

以下で、ネットとは同電位に接続すべき端子の集合を言う。

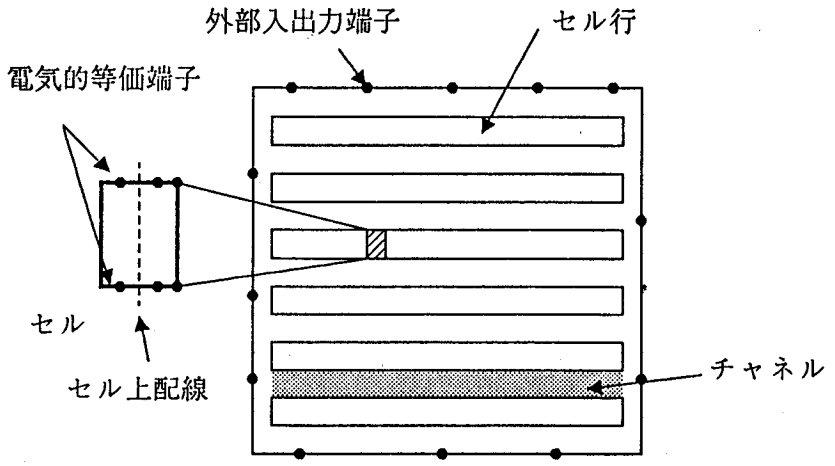


図5-1　レイアウトモデル

5.3　シャフリング操作とその限界

ここでは、シャフリング操作による配置手法の概要と最適化を行う上での限界となる要因について述べる。

M. Hanan等の配置手法の性能比較[52]によれば、一般に従来のヒューリスティック手法の配置結果は初期配置に強く依存している。これは、従来の改善配置手法が1つのセルもしくはセルの系列に対して、セル交換、移動を行うものであるため、局所解に落ち込みやすいことに起因している。こうした問題点を解決するため、セルの集合単位(以後クラスタと呼ぶ)での改善配置を行うシャフリング操作[53]による配置手法を考察する。

5.3.1　シャフリング操作

クラスタ単位での改善配置を行うシャフリング操作について以下に記述する。いま、配置されるべきセルは、すでにいくつかのクラスタに分割され、1次元配列として初期配置されているとする。シャフリングの基本操作は、以下のステップからなる(図5-2)。ここでクラスタ数をkとする。

1) クラスタC_i、C_{i+1}を一時的に1つのクラスタ$C_{i,i+1}$に併合する。

2) $C_{i,i+1}$以外のクラスタをそのままにして$\Sigma_{1\leq i\leq k}H_i$が最小となるように、$C_{i,i+1}$を$C_i'$、$C'_{i+1}$に再分割する。

この基本操作をi=1からk-1まで繰り返す。次に逆順にi=k-1から1まで繰り返す。これを1回のシャフリング操作とする。ここで、H_iとはクラスタC_iを貫通し、クラスタC_i中のどのセルにも端子を持たないネットの総数をいう。

外部端子への配線要求をもシャフリング操作中で考慮するために、外部端子を仮想的なセルとしてその相対位置をもとにクラスタに割りつける。ただし、外部端子の相対位置を変更しない場合は、シャフリング操作中で外部端子のクラスタ割り付けを変更しない。

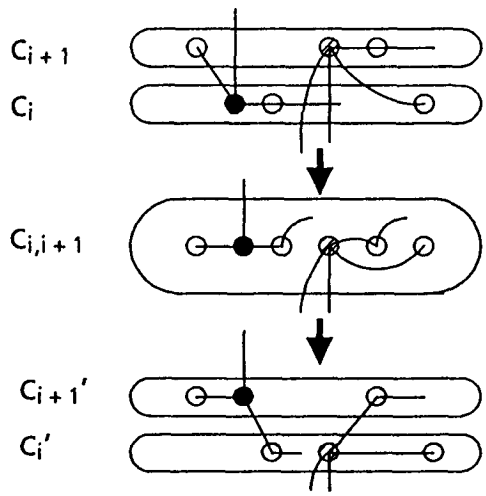


図5-2　シャフリングの基本操作

5.3.2　シャフリング操作による配置手法 ー 配置手法A

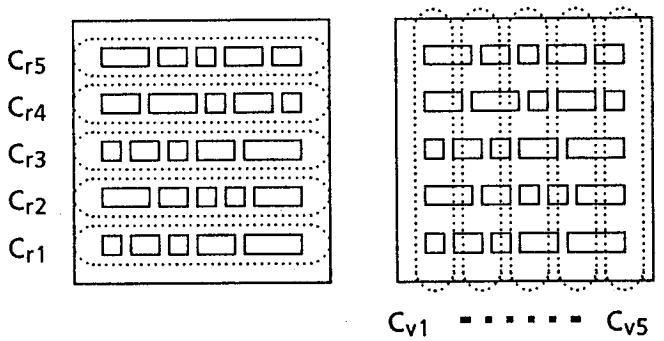
いま、シャフリング操作による手法 ー 配置手法Aを次の処理A1~3から構成する。

処理A1：セル行へのセル割り付け

処理A2：セル行内の概略順序決定

処理A3：セル行内の詳細順序決定

処理A1では、配置対象のセルを行へ割り付け、次に各セル行をクラスタ($C_{r1},\cdots,C_{rm}$;m:セル行数)としてシャフリング操作を行い、各セルのY座標(すなわち、帰属するセル行)を決定する。(図5-3-a、m=5の場合)



a)セル行割り付け　　b)セル行内の順序決定

図5-3　クラスタの構成

処理A2では、縦方向に等間隔の幅の領域を考え、この領域内に含まれるセルの集合をクラスタ($C_{v1},\cdots,C_{vn}$;n:クラスタ数)とし、処理A1で得られた各セル行への割り付けを保存しながら、このクラスタに対してシャフリング操作を行い、各セルのX座標を決定する。(図5-3-b、n=5の場合)

処理A3では、各々の行内で隣接セルの交換や反転により局所的改善を行う。ただし、評価関数として次式で定義される総幹線長Lを用いる。

$$L=\sum_{Nj}\left(\mathrm{MAX}_{ni\in Nj}(X(ni))-\mathrm{MIN}_{ni\in Nj}(X(ni))\right)$$

ここでX(ni)は、ネットNjの端子niのX座標を示す。

このように、配置手法Aの処理内容は、2次元領域への配置を2つの1次元配置問題に分解し、それぞれの配置改善をシャフリング操作を用いて行うというものである。

5.3.3　最適化の限界

この配置手法Aによるセル配置最適化の性能を検討するために、結線要求の分布について考察する。処理A1でのセル割り付けにおいては、いくつかのセル行にまたがるネットに対して、その行の個数を極小化することを目的関数としている。したがって、この段階で得られるセルの割り付けにおいては、セル行を貫通するネット、すなわち上方、下方のどれかのセル行に端子を持つものの、その行には端子を持たないネット、ができるだけ発生しないように配慮されている。よって、実際に配線処理を行った場合でも、図5-4に示すような1チャネルネットや2チャネルネットが多く発生する。ここで、1チャネルネットとは隣接する2つのセル行にまたがって端子を持つネット、2チャネルネットとは、1つのセル行のみに端子を持つネットを言う。

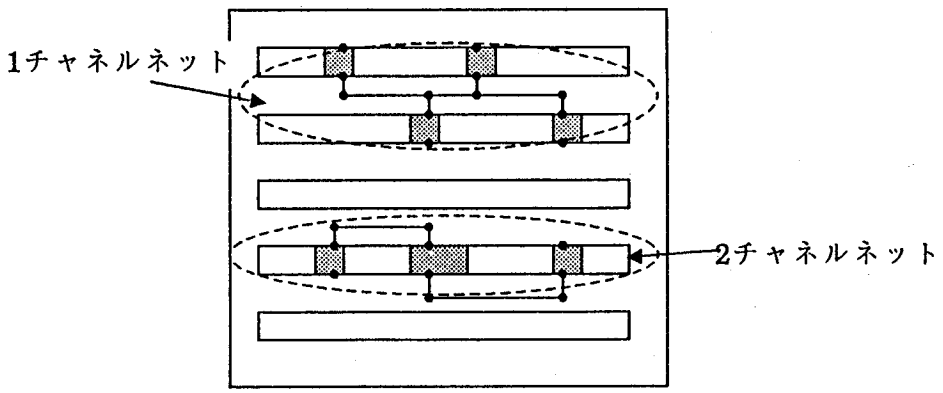


図5-4　実配線パターン

配置手法Aで得られる解が1チャネルネットや2チャネルネットが多く発生するという傾向は、1つのセル行に多くのセルが含まれるような大規模データになる程、さらに顕著になることが実験により判明している。

セル上配線やセル間配線が行えないモデルに対しては配置手法Aが有効であるが、本文で述べるメタル2層配線を用いる場合は、各ネットの幹線が横方向に長くな

ることになり、スタンダードセル方式VLSIではチップ面積の増大、ゲートアレイ方式VLSIでは未配線の原因となる。

5.4　縦方向ネット分布をも考慮した配置手法B

本章では、配置手法Aでの最適化の限界を克服する配置手法Bについて述べる。配置手法Aの問題点は、各ネットに対する配線要求が少数のチャネルに集中することである。これは、一つのネットの広がりを、それに含まれる端子をすべて包含する最小矩形で表した場合に、図5-5で示すような横方向に広がったネット(横方向ネットという)が多くなるという傾向を示す。特に、チップあるいは機能ブロックの縦横比が横方向に長くなるような場合、あるいはチップに搭載する回路が大規模になる場合にこの傾向は顕著となる。

従って、これらの横方向ネットを抑止する配置手法Bについて考察する。

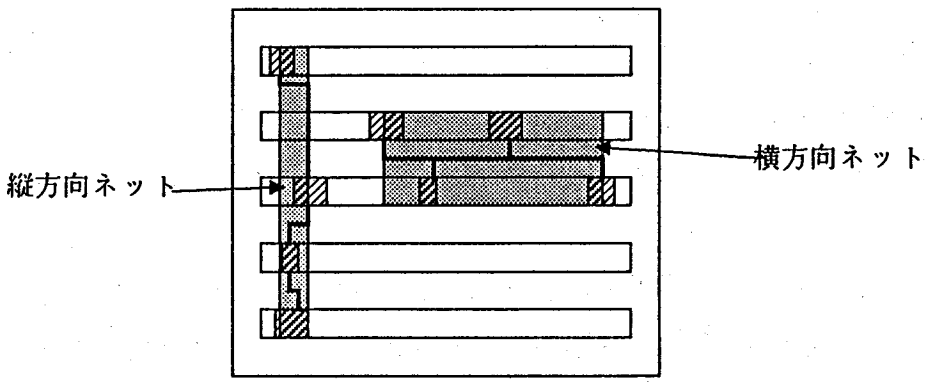


図5-5　ネットの広がり

これは、総配線長の最小化を図るだけでなく、幹線と直交する縦方向に広がったネット(縦方向ネットという)を相対的に高い比率で含むような配置を求めることをめざすものである。なぜなら、縦方向ネットは、実配線時にセル上配線や電気的等価端子を用いるため、面積の増加に影響を与えにくいからである。

そこで、配置手法Bでは横方向ネットを相対的に削減させるために、セル行割り付けを行う前の段階で、セル集合全体を互いに結合度が低い幾つかのクラスタ(以

下、Tクラスタ)に分割し、縦方向に分割されたチップの各部分領域に割り付ける。

配置手法Bは、4つの処理からなる。

処理B1 :Tクラスタ生成

処理B2 :Tクラスタ内セル行割り付け

処理B3 :セル行形成及びセル行内概略順序決定

処理B4 :セル行内詳細順序決定

処理B1では、セル行と直交するようなTクラスタを生成する。処理B2では、各Tクラスタ内でセル行割り付けを行う。処理B3では、処理B2で生成した各Tクラスタ内のセル行から全体のセル行を再形成した後に、セル行内概略順序決定を行う。処理B4では、配置手法Aの処理A3と同様にセル行内での詳細順序決定を行う。

以下では、配置手法Aと手続きの異なる処理B1~3について主に述べる。

5.4.1 Tクラスタの生成

Tクラスタは、チップ上を等間隔な幅で、垂直方向に分割した各領域へのセルの初期割り付けにより生成される。この分割は、ネットの水平方向の広がりに制限を与えるもので、ここでのセル割り付けが大局的なネットの分布を決定すると考えられる。

Tクラスタの生成においては、その目的関数は、各Tクラスタ間の接続要求数を最小化することである。そこで、本処理での評価関数は、Tクラスタ間のカットラインを横切るネットの総和を用いる。

Tクラスタの数N_Vの決定は、次式により求める。

$$N_V = W_{rw} / W_{vc}$$
$$= (W_{avcl} \times Ntlcl) / (m \times k \times W_{avcl})$$

但し($k \times W_{avcl} \geqslant W_{clmx}$)の条件を満たすものとする。

ここで

W_{rw} :セル行の長さ　　W_{vc} :縦方向クラスタの横幅

W_{avcl} :セルの平均横幅　　$Ntlcl$:総セル数

m　：セル行数　　k　：セル行割り付けの最小個数

W_{clmx}: セルの最大横幅

である。また、セル行割り付けの最小個数kとは、処理B2のセル行割り付け時に、シャフリング操作により改善が行うことのできるセルの個数である。この値は、実験から求めた値(5~10)を用いる。

処理の手続きは、図5-6に示すようにMin-Cut法においていわゆるbisection法[54]を適用する。

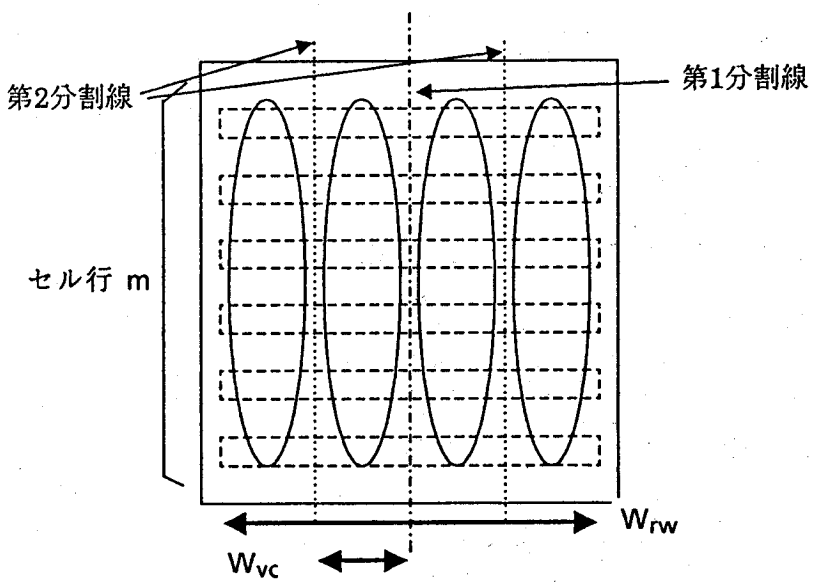


図5-6　Tクラスタの構成(N_v=4)

5.4.2. Tクラスタ内セル行割り付け

各Tクラスタ毎のセル行割り付けは、初期割り付けの後に、シャフリングによる割り付け改善を行う配置手法Aが基本になっている。ただし、あるTクラスタでのセル行割り付けは、他のTクラスタでのネット分布を考慮することにより、異なるTクラスタ間で共有するネットの広がりを小さくすることができる。そこで注目するTクラスタ以外のTクラスタ中でのネットは、注目するTクラスタの外部端子として認識する。また、配線処理において、セル行の長さが異なると行の両端で無効領

域を生じやすい。そこで、注目するTクラスタ以外のセル行長を参照しながら全体のセル行長が均一となるようにセル行の生成を行う。

セル行割り付け処理は、次の2つの手続きから成る。

[ステップ1] 初期セル行生成手続き

指定されたTクラスタについて、クラスタ成長法を用いてセル集合全体を分割した後に行への割り付けを行う。

[ステップ2] セル行生成改善手続き

指定されたTクラスタ以外のセルについて、ネットの分布から仮想的な外部端子位置、及び各セル行の長さを求める。

次に注目するTクラスタについて、シャフリング操作によりセル行への割り付け改善を行う。

ステップ2でのTクラスタの処理の順番は、① 右端から左方向へ、② 左端から右方向へを交互に行う。また、ステップ2は各Tクラスタでのセル行への再割り付けを改善がなくなるもしくは指定された回数が完了するまで処理を繰り返す。

5.4.3 セル行形成及びセル行内概略順序決定

セル行への割り付けにより決定した各Tクラスタ内のセル行をもとに、各Tクラスタのx方向の相対位置関係を保存しつつ全体のセル行を形成する。

配置手法Aにおいて、セル行への割り付け及びセル行内の概略順序決定では、対象となるクラスタの数を動的に変更する階層的処理を行っていた。これは、大規模ASICに対して、クラスタの形成を階層化することで、大局的配置を優先して求め、徐々に局所的配置の最適化を進めるために行っていた[53]。

配置手法Bは、各セル行内での大局的配置は、Tクラスタにより与えられている。従って、初期クラスタ数をTクラスタN_vより大きな値から始める。また、セル行内概略順序決定のための初期クラスタをクラスタ成長法で生成するが、処理B1で求めたネットの分布領域を保存するため、クラスタの候補を各Tクラスタ内に制限する。以上の初期クラスタ生成処理の後に、シャフリング操作によりセル行内での概

略位置改善を行う。

本処理において、シャフリング操作は、全体のセル行から生成したクラスタ全体に対して行うため、Min-Cut法のように1度分割されたものが改善されず、最適化が困難となる問題等も解消している。

以上の処理の後に、各セル行内について、隣接セルの交換、反転により詳細位置の決定を行う。

5.5 実験結果

配置手法A、Bは、スタンダードセル方式VLSI以外にレイアウトモデルとしてアレイ構造を持つゲートアレイ方式VLSIについても適用可能である。そこで、配置手法Aと配置手法Bとの比較を上記の2種類のVLSIデータについて適用した。そのデータを表5-1、結果を表5-2に示す。ここで結果における面積、配線長は第6章で考察する配線プログラムにより実際にレイアウトしたものである。

データ	ブロック数	ネット数	ゲート数 (2入力NAND換算)
A (S.C.)	1,895	1,849	5,000
B (S.C.)	2,428	2,570	7,500
C (G.A.)	1,452	1,566	4,255
D (G.A.)	1,921	2,095	5,000
E (G.A.)	3,090	3,335	10,000

表5-1 データ
(データ欄のS.C.はスタンダードセル方式
G.A.はゲートアレイ方式を示す)

また、他の配置手法との比較として、スタンダードセル方式VLSIのデータについて、シミュレーティッドアニーリング法(S.A.法)[55]との面積、処理時間の比較を表5-3に示す。シミュレーティッドアニーリング法は、局所解に陥らず、配置最適化を進める有効な手法の一つとされている。また、配置手法Aと配置手法Bによるy方向のカットライン数を図5-7、レイアウト結果を図5-8に示す。

	データA		データB	
	面積 (mesh^2)	配線長 (mesh)	面積 (mesh^2)	配線長 (mesh)
手法A	1.325×10^8	4,646,810	1.509×10^9	21,242,360
手法B	1.221×10^8 (-8%)	3,922,320 (-16%)	1.354×10^9 (-11%)	17,324,340 (-19%)

表5-2-a　スタンダードセル方式LSIにおけるレイアウト結果の比較

	データC	データD	データE
手法A	2,202,000	4,127,000	4,887,000
手法B	2,004,000 (-9%)	3,911,000 (-6%)	4,539,000 (-8%)

表5-2-b　ゲートアレイ方式LSIにおける
配線長の比較 (単位はmesh)

	面積 (mesh^2)	処理時間 (sec.)
S.A.法	1.267×10^8	22,740
手法B	1.221×10^8 (-3%)	6,084 (-74%)

表5-3　S.A.法との比較 (データA)
[CPU : 5MIPS相当の計算機]

実験結果が示すように、配置手法Bは配置手法Aに比べて、ネットの分布領域を考慮することでレイアウト面積全体で約10%、配線長で約20%改善されている。また、S.A.法と比較した場合でも面積でほぼ同等の結果が得られ、処理時間で3倍の時間短縮が図れることを考慮すると充分実用に耐える手法と思われる。

5.6 結言

本章では、スタンダードセル方式VLSIレイアウト設計において、より大規模なデータに対するセル自動配置手法について提案した。本手法の特長は、配置処理の段階からメタル2層配線を用いたセル上配線や電気的等価端子を利用する配線がチップ面積削減に有用であることに着目し、それらが相対的に多くなるような配置改善を行うことができる点にある。本手法の性能評価を、配置手法A及びS.A.法との比較実験により行い、その有効性を示した。

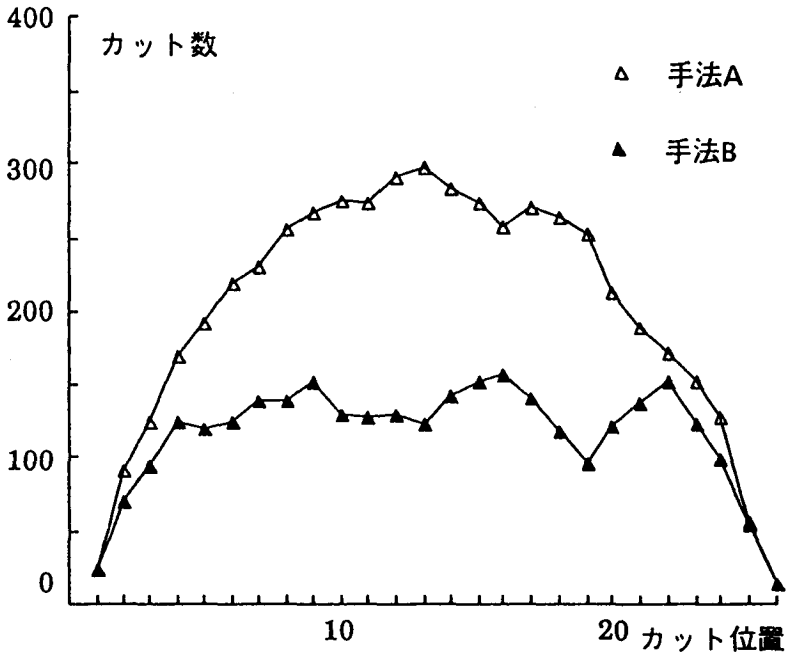


図5-7　カットライン数の比較(データA)

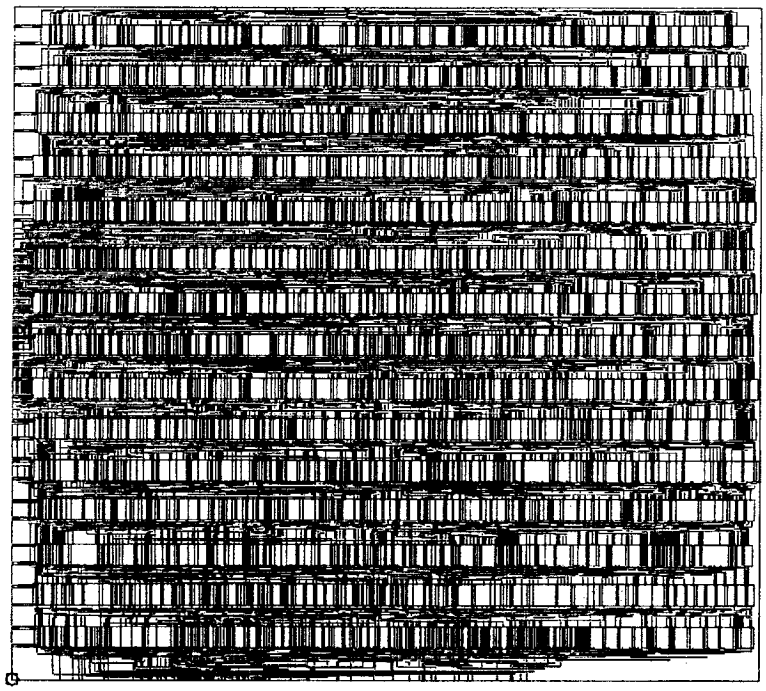


図5-8　配置手法Bによるレイアウト結果

第6章　スタンダードセル方式自動配線手法

6.1　緒言

本章では、大規模スタンダードセル方式レイアウトにおける自動配線手法について述べる。特にセルの持つフィードスルー領域を積極的に利用しつつ、スタンダードセル方式レイアウトにおいてチップ面積増大の大きな要因である局所的な混雑の最小化を図る概略配線手法について考察する。さらにチップ面積最小化に有効と考えられる配線処理中でのセルの局所的な移動、チップ周辺での一括配線手法などについて考察する。

6.2　スタンダードセル方式における自動配線問題

VLSI製造技術の進歩に伴って、ASICの大規模化、高性能化が進み、またユーザからのニーズもますます多様化することにより、ASICの設計は複雑化、長期化してきている。このような状況で設計期間を短縮し、設計品質を向上させるために、ASIC設計自動化技術をより発展、充実されることが不可欠である。スタンダードセル方式VLSIは、ゲートアレイに比較して製造期間はやや長くなる欠点はあるが、チップサイズがより小さくなることから需要が増大しつつある。こうした点からスタンダードセル方式VLSIに対する効率的な自動配線手法の開発が重要な問題となっている。

スタンダードセル方式VLSIのレイアウトモデルを図6-1に示す。チップは内部機能ブロックとその周辺に配置されたI/Oインターフェイス用バッファ列から構成されている。内部機能ブロックは、その内部においてセルが水平列を形成しており、各セルは接続情報としての端子情報とフィールドスルーが可能な位置情報をもつ。ここで、ネットとは等電位となるべき端子の集合であり、フィールドスルーとは配線する時、配線があるセル列を貫通することをいい、図6-2に示すようにセル間を使用する場合とメタル2層配線を利用してセル上を通過する場合の２種類がある。ま

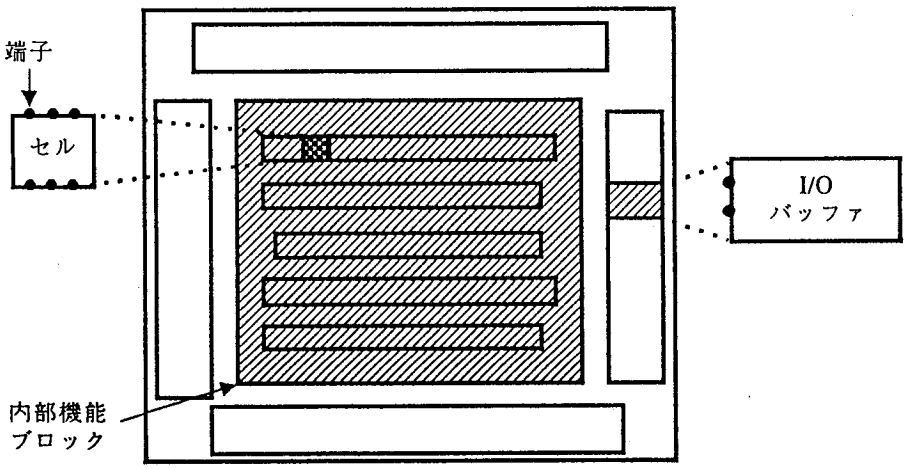


図6-1　レイアウトモデル

た、幹線とは端子及びフィールドスルーを水平(垂直)チャネル内で配線するときの水平(垂直)線分を言う。

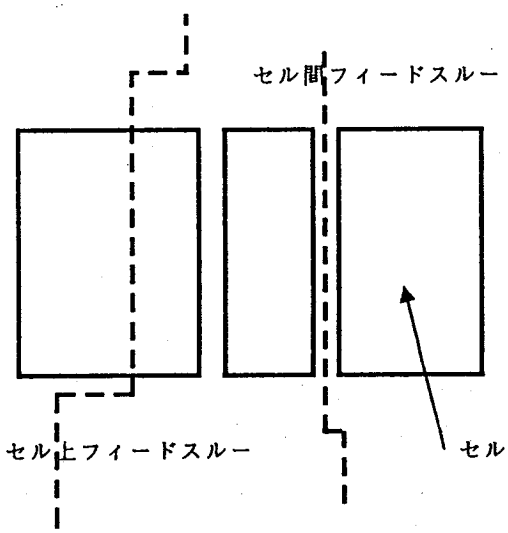


図6-2　フィードスルー

スタンダードセル方式、特にメタル2層を用いる場合、セル間のフィードスルーに加えてセル上でフィールドスルーも可能であることから、これを効果的に使用することは総配線長の最小化や局所的混雑の緩和に極めて有効であると考えられる。例えば、図6-3に示すようなネットの配線を行う場合、図6-3(a)に示すように二つの

チャネルで長い幹線を用いて配線するのではなく、図6-3(b)に示すようにフィードスルーを使用すれば水平方向の配線経路が削減され、したがって、水平幹線を割り付けるチャネルが減り、局所的混雑の緩和も容易になる。

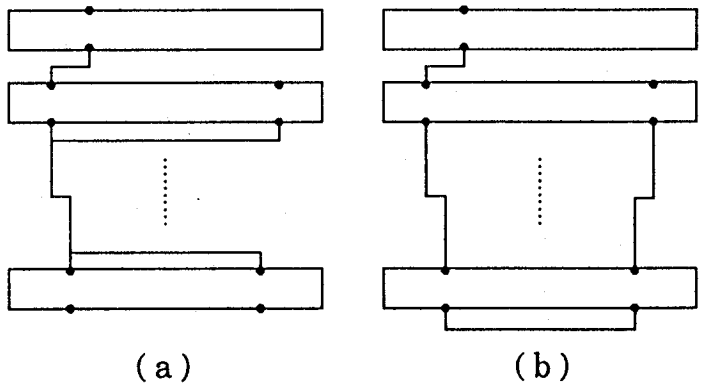
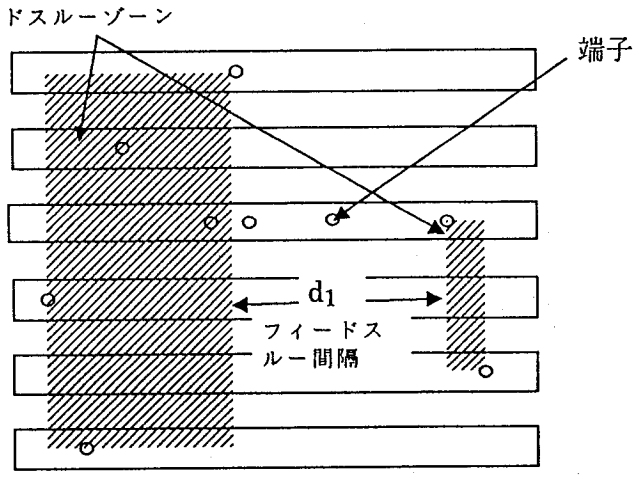


図6-3　フィードスルーの使用例

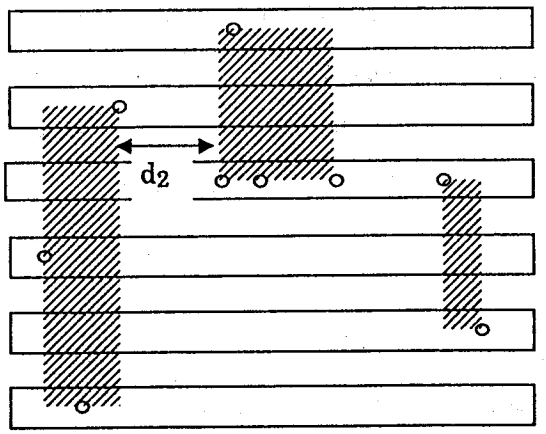
従来、スタンダードセル方式における概略配線はゲートアレイと同様にチャネルへの経路割り付け問題として定義され、ゲートアレイ用の手法に対してその目的関数に、チップサイズに影響を与えるチャネルの最大混雑度の総和などを加味して利用する場合が多かった[56]。しかし、上記の観点から、スタンダードセル方式における概略配線では、セル上もしくはセル間のフィードスルー位置を最適化することが最も重要な問題である。

そこで、本章ではスタンダードセル方式VLSIにおける概略配線問題を、ネットにフィードスルーを割り付ける問題と、各幹線をチャネルに割り付ける問題との2つに分割し、これらの問題に対して、総配線長およびチャネル混雑度を評価関数として用いた概略配線の一手法を提案する。概略配線手法を述べる前に、必要な幾つかの用語を定義する。いくつかの端子間をフィードスルーを用いて配線する場合、それらの端子を含む最小矩形領域をフィードスルーゾーンと言う。また、フィードスルー間隔とは、あるネットにおいてフィードスルーを同じセル列で複数用いる時、それらに対応するフィードスルーゾーンの距離の最小値を言う。例えば、図6-4では、あるネットの分布についてフィードスルー間隔をd_1もしくはd_2とした場合、

各々図6-4(a)、図6-4(b)中、斜線で示すフィードスルーゾーンが設定される。



(a)フィードスルー間隔がd_1の場合



(a)フィードスルー間隔がd_2の場合

図6-4　フィードスルーゾーン

ネットは次の3つのタイプに分類される(図6-5)。すなわち、隣接する2つのセル列にのみ端子をもつネットを1チャネルネット、1つのセル列のみに端子をもつネットを2チャネルネット、これら以外のネットをグローバルネットという。

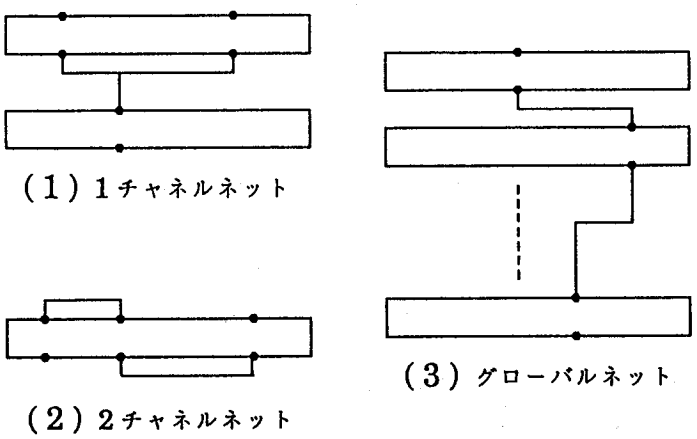


図6-5　ネットの分類

6.3　概略配線手法

提案する手法は、初期配線と逐次改善からなる。逐次改善は、フィードスルーの直線化と初期配線で与えられた混雑度をもとにした再配線による手法を用いる。ここでは、メタル2層配線を用いる場合について述べる。セル上配線が禁止されている場合はフィードスルーをセル間に限定することにより同様の手法が適用できる。

6.3.1　初期概略配線

初期概略配線は、まず各ネットごとにフィードスルーゾーンを求め、それにセル上もしくはセル間のフィードスルーを割り付ける。次に各チャネル内で端子またはフィードスルーを相互に接続する幹線をチャネルに割り付ける。以下では、各ステップについて処理の概略を述べる。

<u>フィードスルー間隔(F間隔)の設定</u>

まず、データによりフィードスルーの使用可能数が様々であることからフィードスルーの使用を制御する値を求める。各ネットを配線するために必要なフィードスルーの数のすべてのネットに関する総和をNFとし、セル上フィードスルーが可

能な位置の総数をPFとしたとき、フィードスルー間隔を、セル列の平均幅をa·PF / NF(aは定数)で割った値とする。

フィードスルーゾーンの抽出

次にF間隔を用いてネット毎にフィードスルーゾーンを求める。

[操作1]　ネットを構成する端子をx座標で昇順に並べる。

i←2

[操作2]　i番目の端子について以下の処理を行う。

i)　ゾーンが存在しない場合

i-1番目とi番目の端子間配線にフィードスルーが必要であるなら、2つの端子を含む最小矩形でゾーンを生成する。

ii)　ゾーンが1つ以上存在する場合

i番目の端子からF間隔以内にゾーンが存在するならば、その端子が含まれるようにゾーンを拡張する。

さもなくば、新たにゾーンを生成する。

[操作3]　i←i+1

すべての端子を検索したならば、終了する。

さもなくば、操作2へ。

フィードスルーの割り付け(図6-6)

横方向の幅が狭くかつ縦方向の長さが長いゾーンほど配線が蛇行して配線長を増大させ、チャネル混雑度をあげる確率が高いことから、各ゾーンを以下のように順序付けし、各ゾーンについて割り付けを行う。

[操作1]　フィードスルーゾーンを横方向の幅が狭く、かつ縦方向の長さが長い順に順序付けする。

[操作2]　各ゾーンについて以下の処理を行う。

i)　ゾーンがセル列の中心より左(右)に存在すればゾーンの左(右)端よりフィードスルーを検索し、見つかればそれを割り付ける。

ii) もしゾーン内で見出せない場合にはゾーンを一定値だけ両横方向に広げて、i)を繰り返す。

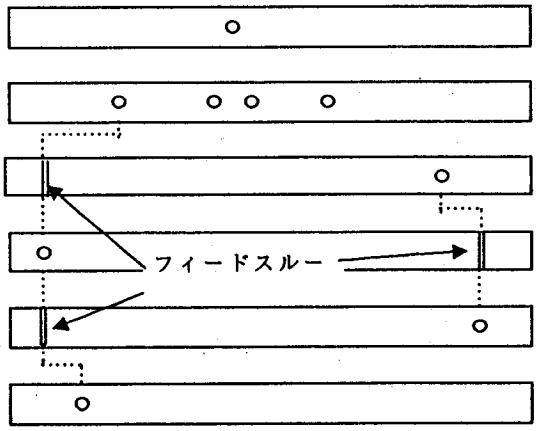


図6-6　フィーソスルーゾーンの割り付け

幹線のチャネルへの初期割り付け

フィードスルーが割り付けられた段階で各ネットについての概略配線は、フィードスルー端子を含むセル列上の端子をどのチャネル内で接続するかという、いわゆる幹線のチャネル割り付けの問題となる。ここでは、チャネルの局所的混雑を緩和することにより、すべてのチャネルでのトラック使用総数を最小化することが目的となる。

ネットは最初に1チャネルネット、次にグローバルネット、最後に2チャネルネットの順で割り付けを行う。1チャネルネットは割り付けるチャネルが一意的に決まることから最初に割り付ける。2チャネルネットは他のタイプのネットの割り付けによって得られた混雑度をもとに、幹線長の増大を伴うことなしに2つの隣接するチャネルのどちらかを選択するので、最後の処理の対象となる。

グローバルネットの幹線割り付けは、注目しているネットの端子が存在するチャネルすべての中から、その割り付けによって最も混雑度を増加させないチャネルを

求め、そのチャネルに幹線を割り付ける。次にそのチャネルから上または下の方のチャネルへ順次幹線を割り付けて行く(図6-7に例示)。

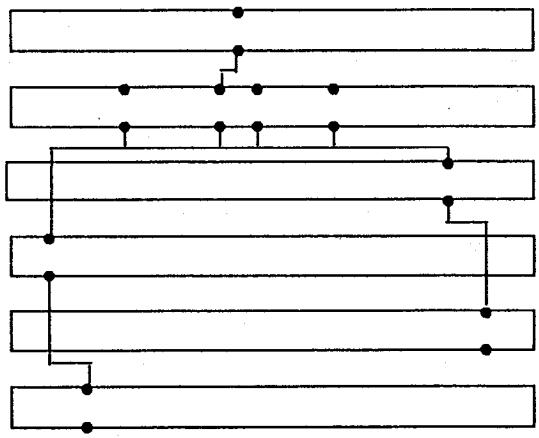


図6-7　幹線の初期割り付け

6.3.2.　概略配線の改善

概略配線の改善は、得られた初期解に対して総配線長の減少とチャネルの混雑度の減少を目的とし、フィードスルー再割り付け、論理的等価端子の交換、幹線再割り付けなど3つの操作からなる。

フィードスルー再割り付け

初期概略配線においてフィードスルーはゾーン内で検索し割り付けているため、ゾーン幅が広い場合に配線が蛇行する場合がある。これを図6-8のようにフィードスルーを再割り付けし直線化する。この操作により配線長を減少させ、チャネルの混雑度も減少させることが可能である。

各セル列の各フィードスルーに対し、それを同じセル列に存在する未使用のフィードスルー位置に移動させ、そのネットの総幹線長が削減できたならば、それを採用する。この操作をまず各セル列中で左から右へ、セル列を下から上に順に行い、次に右から左へ、上から下へ行う。

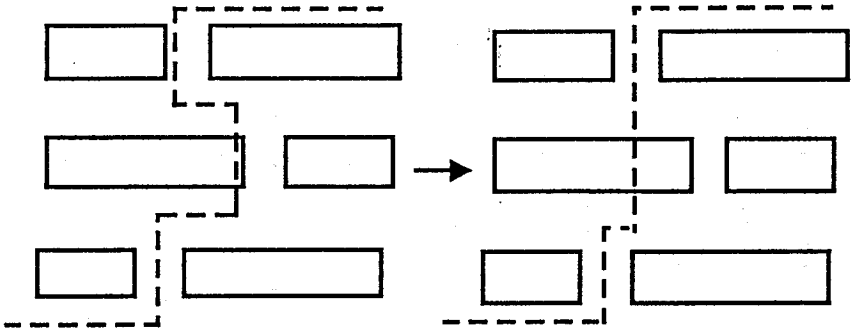


図6-8　フィードスルーの再割り付け

論理的等価端子の交換

各セルは論理的等価端子を持っており、それらの交換により配線長及びチャネルの混雑度を減少させることができる。そこで、各セルの論理的等価端子対の各々に対して一度ネット割り付けを交換し、配線長が減少したならば交換を採用するという操作を改善がなくなるまで反復する。(図6-9)。この手法は、フィードスルー割り

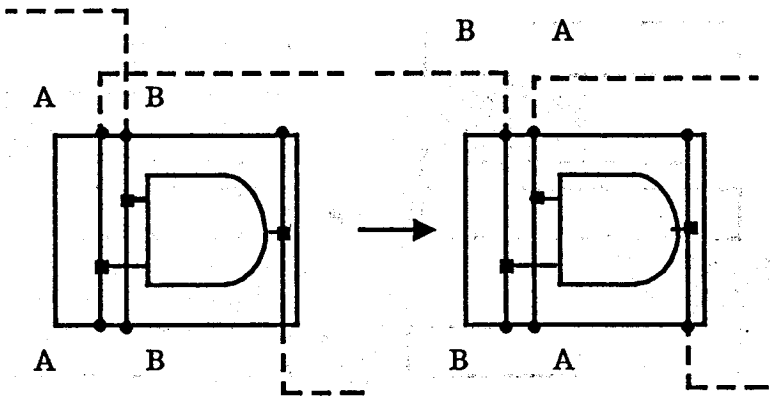


図6-9　論理的等価端子の交換

付けにより与えられた2つの隣接するセル間のフィードスルーの順序付けにも適用する(図6-10)。

幹線再割り付け(図6-11)

チャネルの中で混雑している部分を通っている幹線の中で隣接するチャネルへ移動可能な幹線を選び、再割り付けを試みることにより、チャネルの局所的混雑を緩

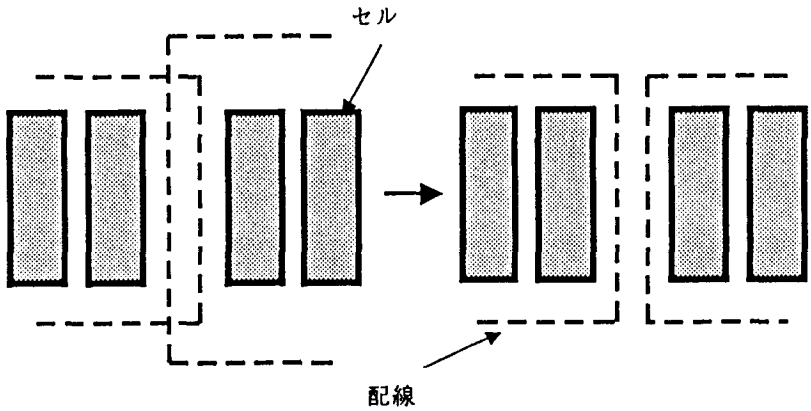


図6-10　セル間フィードスルーの交換

和する。各幹線をすべての支線との交点で分割し、個々に上下のチャネルに移動可能か、さらにそれによりチャネル混雑度が減少するかを評価し、有効な場合は移動させる。

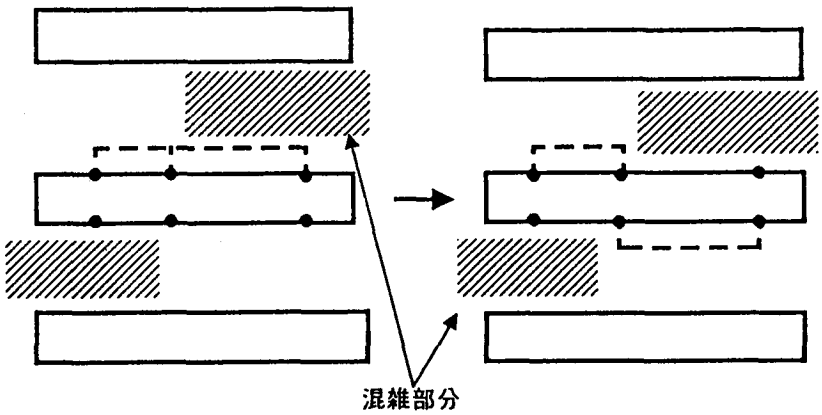


図6-11　幹線の再割り付け

6.4　チップ面積最小化のための機能

本章で提案している配線手法はすでに述べた概略配線の他に、チップ面積をより削減するため、また、設計期間を短縮する為に幾つかの付加機能を持つ。

6.4.1 セルの局所的配置改善

スタンダードセル方式の場合、各セル列の長さが異なるため、左右の周辺垂直チャネルにおいて配線に使われない無駄な領域が生じやすくなる。特に、セル上配線が禁止されている場合、フィードスルーはすべてセル間に割り付けられるため、配線結果が各セル列の幅に大きく影響する。セル間フィードスルーによるセル列幅の変化を配置設計の段階で正確に評価することは困難である。

本手法では、この無駄な領域を少なくするため初期概略配線が終わった後に左右の垂直チャネルの配線領域と各セルの長さを見積もり、その結果に整合するように各セルの再配置を行う(図6-12)。処理手順は次の通りである。

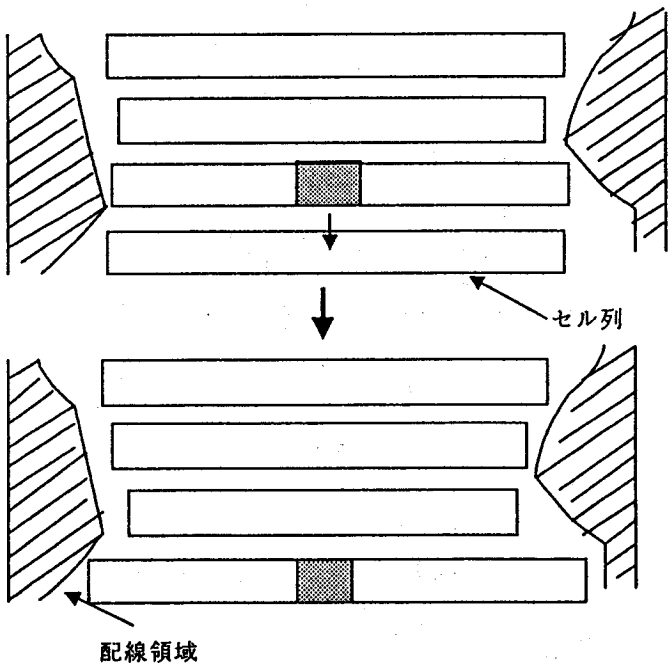


図6-12　セルの配置改善

[操作1]　初期概略配線の結果により、左右の垂直チャネルの配線に使われる領域を求め、各セル列の幅に加える。

[操作2]　以下の操作をセル列幅がもはや削減されなくなるまで行う：最も幅の広いセル列の中から適当な幅のセルを選び出し、隣接するセル列へ移動させ

る。この時候補として選ぶセルは、チャネルの隣接部分の混雑度が小さく、移動することによりフィードスルーが増えないものとする。

6.4.2 周辺バッファとの一括配線

従来のスタンダードセル方式の配線手法は、まず下位の階層として、内部機能ブロック内のセル間配線を行い、内部機能ブロックの大きさ、形状が決定した後、上位の階層でI/Oインターフェイス用バッファを周辺に配置し、それと内部機能ブロックの間を配線する方法が一般的であった(図6-1)。しかし、このように階層に分けて別々に配線をおこなった場合、階層の境界上に無駄な配線が生じ、チップ面積を増大させる一因となることがある(図6-13)。

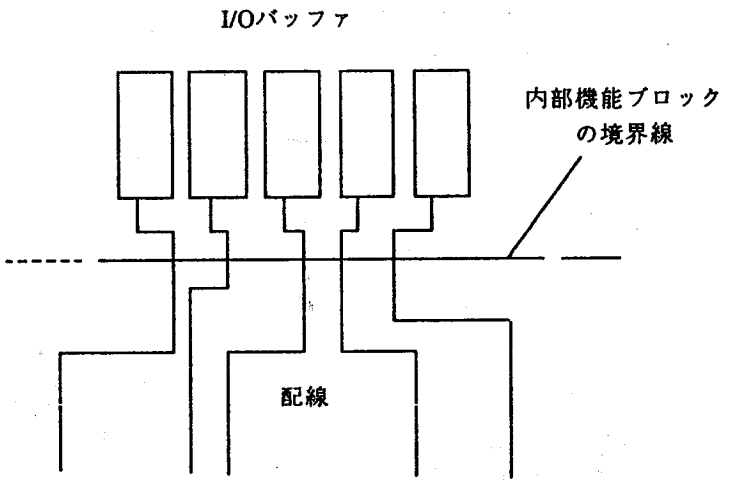


図6-13 階層間の無駄な配線

本手法では内部機能ブロック、周辺バッファとの一括配線を行っている。この手法は次の手順からなる。

[操作1] I/Oバッファの端子位置に対応して内部機能ブロックの外部端子の位置を一時的に決める。

[操作2] 内部機能ブロック内の配線を行う。

[操作3] 内部機能ブロックの形状、大きさと電源線、グランド線の配線領域を考慮して、4辺にあるI/Oバッファ列を以下のように配置する。

i) 各I/Oバッファを、対応する内部機能ブロックの辺と互いの中点が一致するように配置する。

ii) この時、バッファ列同士に重なりがあれば、バッファ列を辺に沿って移動させることによりこれを解消する(図6-14)。

iii) チップの各コーナーにおいて電源線、グランド線が配線できるかをチェックし、可能でないなら配線領域を確保するためにバッファ列を移動させる。

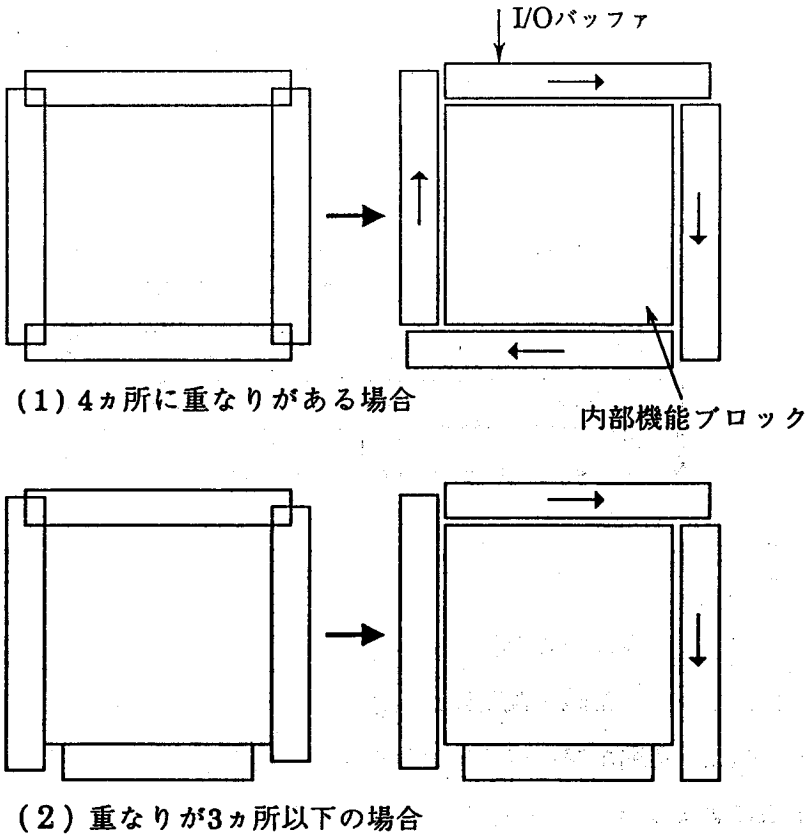


図6-14　周辺バッファの重なり解消例

[操作4]　内部機能ブロックの外部端子位置修正を行う。

[操作5]　内部機能ブロック周辺のチャネルを再配線する。

[操作6]　I/Oバッファと内部機能ブロック間の配線、電源線、グランド線の配線を行う。

6.5　実験結果

本手法を用いて2つのデータについて実験を行った結果を表6-1に示す。なお、プログラミング言語はFORTRAN77を用い、計算機は1MIPSのワークステーションを使用した。

ここでデータAはセル上フィードスルーが可能であるデータ、データBはセル上フィードスルーが禁止されているデータである。実行方法はそれぞれ次のとおりである。

	データA	データB
ブロック数	462	477
ネット数	618	553
端子数	5,791	3,432

方法1：従来の配線プログラム[57]

方法2：本手法(初期概略配線のみ)

方法3：本手法(初期概略配線＋概略配線改善)

方法4：本手法(初期概略配線＋概略配線改善＋セル配置改善)

また本手法によるレイアウト結果を図6-15に示す。本データの処理時間は12分3秒(cpu time)で、方法1の約1/2である。

6.6　結言

本文では、スタンダードセル方式VLSI設計の自動配線におけるフィードスルー割り付けを用いた概略配線手法とそのいくつかの改善手法について考察した。実験結果が示すように本手法は十分実用に耐える良好な結果を出している。特にフィード

データA	方法 1	方法 2	方法 3	方法 4
サイズ	2,206x2,124 (1)	1,914x2,100 (0.86)	1,884x2,044 (0.82)	1,848x1,988 (0.78)
水平トラック数	237(1)	231(0.97)	208(0.87)	203(0.86)
配線長	383,572 (1)	336,513 (0.88)	320,496 (0.84)	319,529 (0.83)

データB	方法 1	方法 2	方法 3	方法 4
サイズ	1,160x1,380 (1)	1,156x1,428 (1.03)	1,148x1,360 (0.98)	1,124x1,356 (0.95)
水平トラック数	176(1)	188(1.07)	171(0.97)	170(0.97)
配線長	161,100 (1)	169,688 (1.05)	159,290 (0.99)	160,420 (0.99)

スルー割り付けは、メタル2層の場合には、極めて有効な方法であることが示された。またI/Oバッファとの一括配線などの自動処理により人手の介入がほぼなくなり、設計期間を大幅に短縮できた。

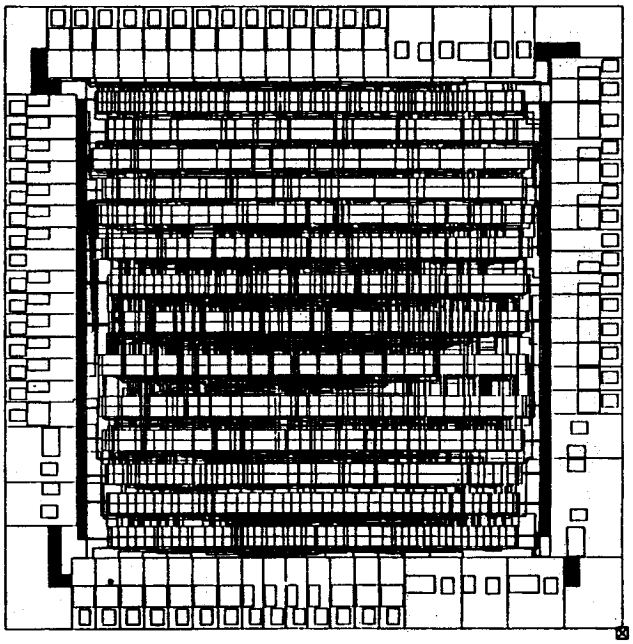


図6-15　本手法による配線例

第7章　レイアウト設計支援統合化システム

7.1　緒言

本章では、自動設計ツールと会話型設計ツールの一体化を実現し、ASICレイアウト設計を総合的に支援することを目的として開発したシステムの構成、データベース、およびこのデータベース上で実現されたレイアウトツールの概要とその有効性を確認した実験結果について述べる。

7.2.　ASICのレイアウト設計における諸問題

ASIC製造技術の急速な進歩により、ASICの設計複雑度が飛躍的に増大している。しかし、一方、多品種少量化の進展、製品の短命化により、ASICの設計短縮が強く望まれている。ASIC設計の各種設計工程の中で、レイアウト設計は全体の設計期間に最も大きく影響を与える工程のひとつであり、レイアウト設計の期間短縮を目的として、これまで自動配置配線やモジュールコンパイラなど多くの自動レイアウト手法が開発され実用化されている[58]。

しかし、このような設計自動化技術が進展しているにもかかわらず、以下に述べる理由から人手によるレイアウト設計が依然として実行されている。

(1) 自動設計手法を用いた場合、人手設計同等のチップ面積削減を図ることは現状では困難であり、メモリ部やCPUコア部などのマクロセルでは人手によるレイアウト設計が依然として高密度化に寄与している。

(2) 自動レイアウト手法が適用できるASICはレイアウトモデルや電気的特性面で大きな制約を受ける。例えば、アナログ回路では、人手レイアウト設計が不可欠であり、システムオンチップを実現するうえで、設計期間短縮の障害となる。

(3) 基本セルライブラリは、絶えず変化する製造技術に伴う設計規則の更新により常に再設計が必要であるため、人手設計に頼らざるを得ず、その開発は膨大な作業となっている。

このような問題点からASIC設計者は自動設計ツールだけでなく、同時に会話型の設計ツールをも利用できる総合的なレイアウト設計環境を必要としている。

このような問題に対処し、総合的なレイアウト設計環境を実現するために、会話型設計ツールと自動設計ツールを一体化し、かつディジタルVLSIだけでなく、アナログVLSIをも同一環境上で設計できるようにレイアウトシステムを構築した。

本システムは次の特徴を持つ。

(1) データの一貫性を保持するために、すべてのレイアウトツールは単一のデータベースの上で一体化されている。また、データベースは、レイアウトパターン情報以外にトランジスタ、抵抗などの素子情報や接続情報、さらに端子情報などを持ち、レイアウト検証の高速化、自動レイアウトツールの適用を可能としている。

(2) シンボリックレイアウトや会話型自動配線などにおいてマスク図形の接続判定機能を付加し、ディジタルVLSIだけでなく、複雑なマスク図形を用いるアナログVLSIのレイアウト設計の自動化/効率化をも可能としている。

(3) レイアウトエディタはレイアウトパターン編集機能だけでなく、他の各種レイアウトツールを利用するときのユーザインターフェース機能を持つ。これにより、設計者は基本セルからチップ全体のレイアウトパターンに至るまで、一つのデータベース上で、かつ共通のユーザインタフェースを用いて、自動的にあるいは会話的に設計することが可能である。

(4) 本システムは、VLSI製造に関する情報をシステムから独立させて、テクノロジーファイルとして保持しているので、素子抽出規則、素子特性計算規則、設計規則などが登録、変更可能であり、ツールの使用に影響しない。

以下、本章では、まずシステムの概要とデータベース構造を述べたのち、レイアウト設計用ツールの概要を示し、データベースの特徴を活用して実現された幾つかのツールについて性能評価を行う。最後に本システムを用いて設計した事例を示し、システムの有効性を確認する。

7.3 システム構成

本システムのハードウエア、ソフトウエア構成およびデータベース構造について述べる。

7.3.1 ハードウエア構成

本システムは次の3種類のハードウエアから成るコンピュータネットワーク上で実現されている。第1は直接設計者が使用するエンジニアリングワークステーションであり、主に会話型の処理が行われる。第2はスーパーミニコンピュータであり、セルライブラリ管理、データベース管理、ネットワーク管理などを行う。第3は汎用大型計算機で、自動レイアウトなど大規模データ処理を行うバッチジョブを主な対象としている。これらはローカルエリアネットワークにより、相互に接続されている。

7.3.2 ソフトウエア構成

本システムは単一のデータベースのもとに、セル設計からマスクデータ作成に至るまでのレイアウト設計の各工程を支援するツール群がすべて一体化されている(図7-1)。設計者が最もなじみやすいレイアウトエディタをユーザインターフェースとしており、各ツールをエディタ上でコマンドを介して利用できるようになっている。実際にはすべてのツールを一つのプログラムに組み込むことは実行モジュールのサイズの問題があり不可能であるため、各ツールのサイズ、処理時間に応じて次の3通りの方法を用いて結合している。

(1) レイアウトエディタ中に直接組み込む。

(2) 子プロセスとして独立実行させ、終了時に親プロセスであるレイアウトエディタに、割り込みをかけ、結果を受け入れるか取り消すかの選択を設計者が行う。

(3) バッチジョブとして、必要に応じた性能を持つ計算機上で実行させる。

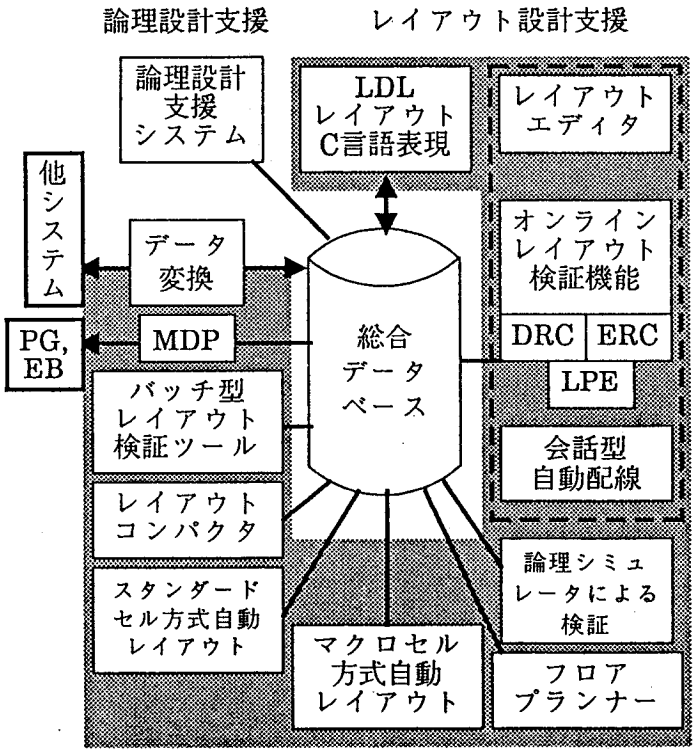


図7-1　システムのソフトウエア構成

7.3.3　データベース構造

本システムのデータベースの特徴は、素子情報、接続情報、端子情報を含んだレイアウトデータを保有している点にある。従来のレイアウトデータベースは、単にマスクデータを作成するための図形の集まりであったが、本システムでは素子として意味を持つ図形データの集合をレイアウトの単位とし、これに素子情報を属性として持たせて回路確認を効率化することにより、レイアウト検証の高速化を図っている。されに端子情報を用いて、設計自動化ツールの直接的な適用を可能としている。データベースの論理構造は、以下の項目であらわされている(図7-2)。

(1)　定義情報

定義には、コンポーネントとセルの２種類がある。コンポーネントとは、回路素子として意味を持つレイアウトの最小単位である。素子としてトランジスタ、コン

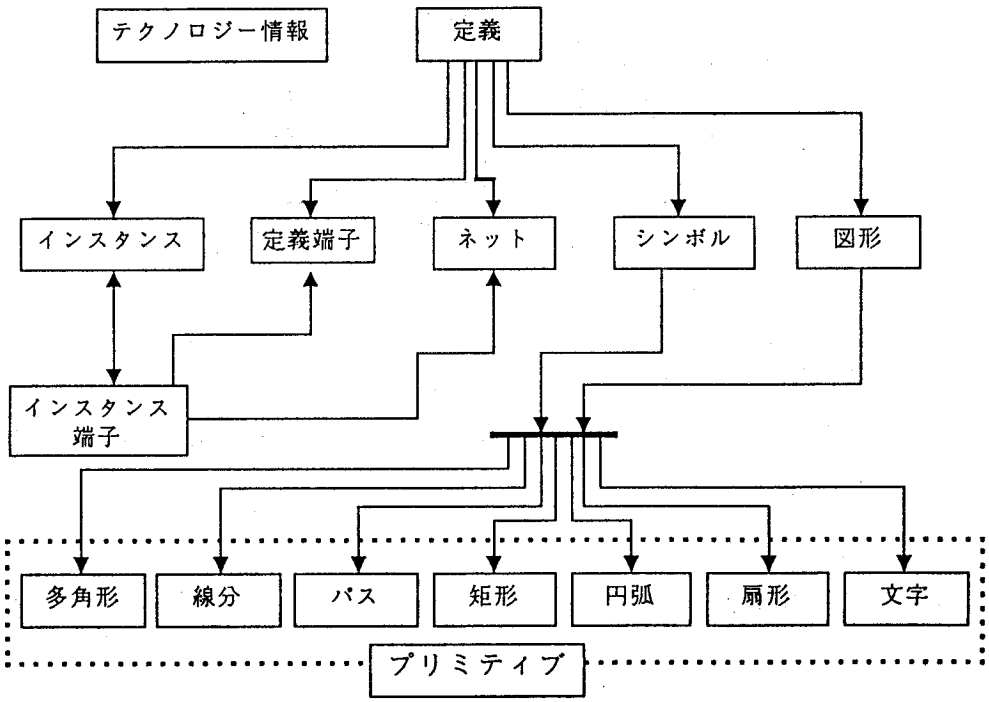


図7-2 データベースの論理的構造

タクト、配線などがあげられる。また、セルとは回路としての機能を持つレイアウトパターンの単位であり、コンポーネントおよびセルから構成される。

定義情報は各セルおよびコンポーネントについて設計情報を管理する。設計情報として、図形情報、インスタンス情報、シンボル情報、ネット情報、端子情報、および定義自身の属性情報を持つ。属性としては、定義の種類、構成するデータを囲む最小の最外殻、使用しているマスク層、バージョンなどがある。

コンポーネントは、特に素子情報をその属性として保有する。また、コンポーネントは、伸縮仕方についての属性を持ち、基本形状を保持しながら、配置された座標により大きさや形状を可変することができる(以下、これをSコンポーネントという、図7-3)。

(2) 図形情報

図形情報は、コンポーネントのレイアウトパターンを与える。

(3) インスタンス情報

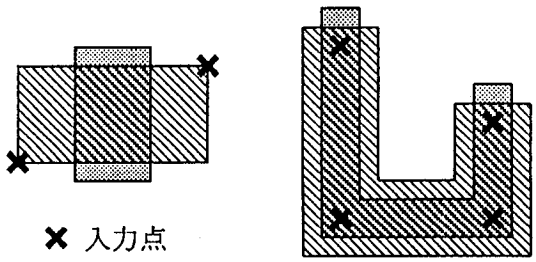


図7-3　Sコンポーネントの例

インスタンス情報は、セルを構成する子セルやコンポーネントの配置情報などを持ち、セルの階層やアレイ構造などを表現する。コンポーネントの場合は、属性として接続情報を持つ。

(4)　シンボル情報

シンボル情報はシンボリックレイアウトなどの場合のシンボルデータを格納する。

(5)　端子情報

端子情報には、定義ごとに存在する「定義端子」と、インスタンスごとに存在する「インスタンス端子」がある。定義端子は定義外とのネット情報を管理し、インスタンス端子は、インスタンスを構成要素として持つ定義内でのネット情報を管理する。また、インスタンス端子は属性として接続情報を持つ。

(6)　ネット情報

各定義ごとに信号名を管理し、定義端子、インスタンス端子から参照される。ネット情報は同電位に配線する要求を信号名により表しており、一方接続情報は実際に要素間の接続状態を表すものである。

(7)　プリミティブ

プリミティブには、多角形、線分、パス、矩形、円弧、扇形などがあり、シンボル情報や図形情報の構成要素となる。

(8)　テクノロジー情報

設計規則、素子抽出規則、素子特性計算規則などが記述されており、データベースに付随する製造情報を表す。

本システムでは、以上のデータ構造により、図形情報、接続情報、階層情報および素子情報が統一的に矛盾なく表現されている。

7.4 レイアウト設計ツール

ここでは、レイアウト設計の各段階における、本システムの設計支援ツールの概要を示すとともに、7.3節で述べたデータベース構造を用いて、特に検証の効率化や自動レイアウトとの一体化を図った幾つかの設計ツールの機能について、従来手法と対比しながら、性能評価を行う。計算機はマイクロVAX2(0.9MIP) を用いた。

ASICにおけるレイアウト設計は、次の3つの場合に大別できる。

(A) プリミティブを用いて小規模回路を高密度にレイアウト設計する場合。

(B) シンボリック設計手法を用いて短期間に比較的密度の高いレイアウト設計を行う場合。

(C) 大規模回路を自動レイアウトツールを駆使して設計する場合。

以下では、上記の設計目的ごとに、レイアウトツールの概要について述べる。

7.4.1 小規模回路設計用レイアウトツール

レイアウト設計(A)では、デザインルールを守りつつ最小の領域に納まるようにレイアウトパターンを作成し、そのパターンを実際に製造したときの回路特性を検証する。本システムではこの段階は会話的設計が主体であり、これを支援するレイアウトエディタと、デザインルールや回路特性に関する検証を効率的に行うオンラインレイアウト検証ツールを準備している。

レイアウトエディタは、円滑なレイアウト設計を行うための次のような機能をユーザインターフェースとして備えている(図7-4).

(1) マルチウインドウによる複数セルの高速表示[59]および同時修正

(2) メニュー、キーボード、ファンクションキー、シンボルコマンドなどを用いた多入力方式

(3) 階層設計のサポート

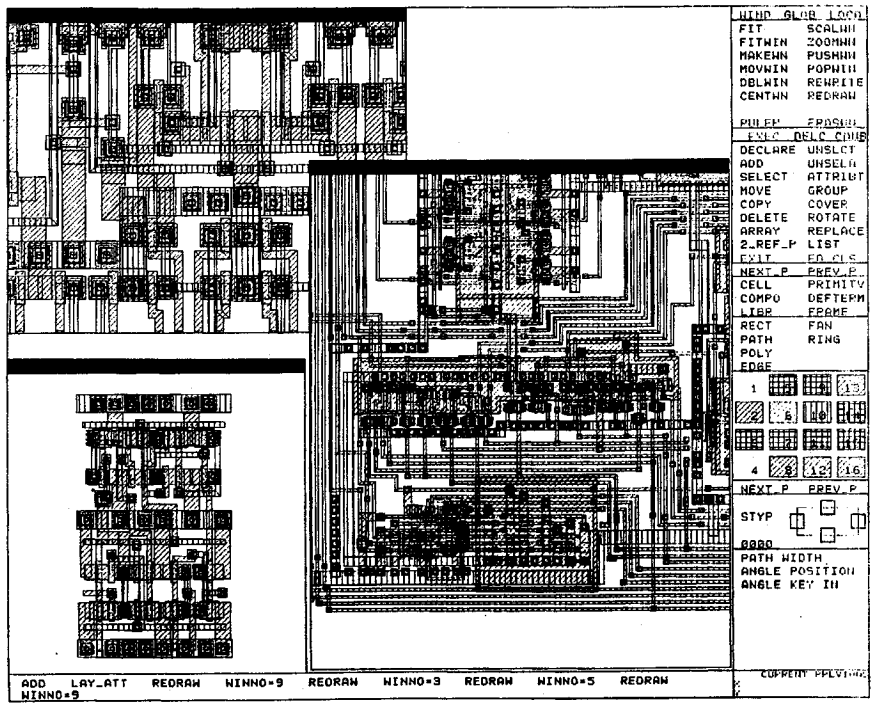


図7-4　レイアウトエディタ

(4) レイアウト設計中の論理回路図参照/照合

レイアウト検証はこれまで、レイアウト設計完了後にバッチ処理で行われていた。しかし、レイアウト設計が完了した時点で設計ミスが発見された場合、膨大な修正作業が発生することが多く、設計の長期化の大きな要因であった。これに対処するため、本システムではオンラインで動作する設計検証ツールを装備している。すなわち、設計中の一部のデータに対してのみレイアウトエディタのコマンド形式でデザインルールチェック、電気的ルールチェック(図7-5)、レイアウトパラメータ抽出などの検証を高速に実行し、結果として得られた誤り箇所をグラフィックディスプレイに表示する。これにより設計者は設計途中で設計誤りを修正でき、レイアウト完了後の詳細な検証時の誤り箇所を大幅に削減できる。

ここでは特に電気的ルールチェック(ERC)について特徴を述べ、その性能評価を示す。ERCは、従来、次の手順で処理される[60]。

(1) マスクデータ間の図形演算により、素子認識を行う。例えば、MOS回路の場合ポリシリコン層と拡散層との交差部分を見つけ、その部分をMOSトランジスタのゲートとする。

(2) 次に素子間の接続を抽出する。これは、図形間の接続関係を調べることにより、全図形を等電位図形群(ノード)に分類し、ノード番号付けを行う。

(3) 得られたノード番号をもとにして、ショート、オープンなどの電気的ルール検査を行う。

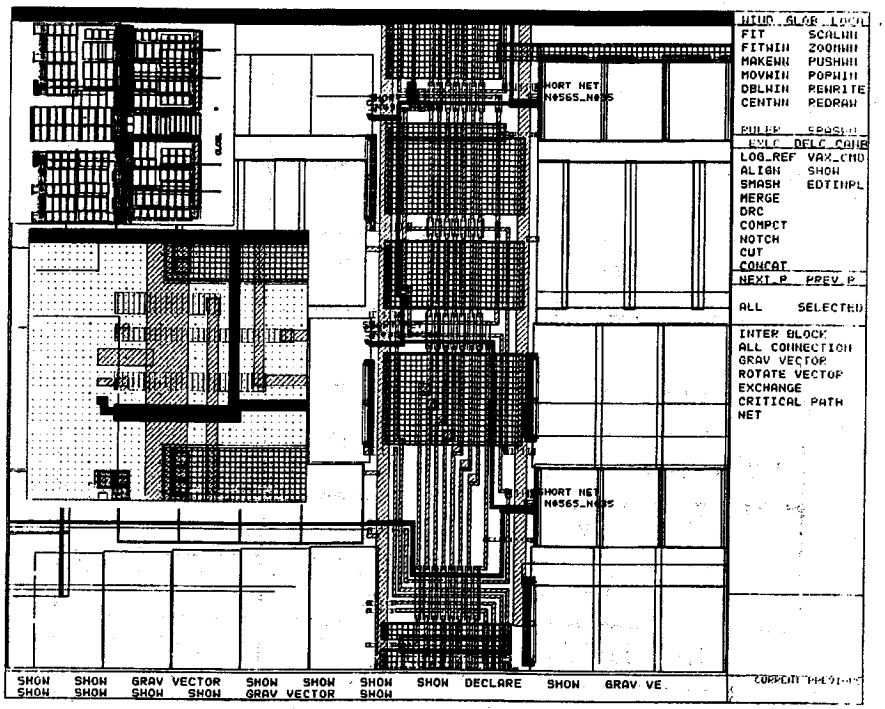


図7-5　オンラインERCの実行例

従来の手法では、マスクデータに修正があるごとに、上記の処理をすべて行わなければならず、膨大な処理時間を要する。本システムでは、以下に示す実現方法を取ることで、大幅に高速化され、オンライン処理が可能となっている。

(1) 素子認識は、各コンポーネントの持つ素子情報を用いて行われ、図形演算を不要としている。

(2) 接続抽出も、コンポーネントやセルの端子との間で行われ、一度得られた接続情報は、セルあるいはコンポーネントが修正されない限り、データベース中に記憶される。

(3) 接続情報は、端子およびコンポーネント単位で記憶されているので、階層的に利用できる。

表7-1に、幾つかのセルに対して行った接続抽出の処理時間を示す。この表から中規模のセルに対して短時間で接続情報が得られ、オンライン化が可能であることがわかる。

コンポーネントより構成されるセルの場合	セルA	セルB
トランジスタ数	22	62
CPU TIME	0'21"27	1'41"14

子セルより構成されるセルの場合	セルC	セルD
ネット数	68	516
インスタンスセル数	28	440
CPU TIME	0'17"50	4'19"18

表7-1　接続抽出の実験結果

7.4.2 中規模回路用レイアウトツール

レイアウト設計(B)では、論理的に誤りのないレイアウト設計を短い期間内で完成することが要求される。本システムではこのために、扱えるシンボルやコンパクション手法を機能拡張したシンボリックレイアウト機能を有している。また、レイアウトパターンから接続情報、レイアウトパラメータを抽出し、これをゲート論理

に変換し、論理シミュレーションを実行させることができる。設計者は波形で表示されたシミュレーション結果を確認するか、論理設計時に得られたシミュレーション結果と比較することでレイアウト結果についての動作検証を行うことができる[61]。ここでは、特にレイアウトコンパクションについて、特徴を述べ、性能評価を行う。

従来、シンボリックレイアウト手法[62]は、以下に示す点において実用上問題がある。第1にコンパクタにおいて扱うことができるレイアウトデータが、マスク図形レベルのレイアウト設計に比較してかなり制限されている。例えば、シンボルは形状が異なれば新たに定義を必要とし、シンボル間の接続位置も固定的に定められている。第2に、シンボリックレイアウトツールとマスクレベルのレイアウトツールが独立して存在し、両者間ではデータ変換が必要である。

本システムのコンパクタ[63]に対して次のような機能拡張がなされている。

(1) Sコンポーネントへの対応

本コンパクタでは個々の素子を、7.3.3節の「データベース構造」で示したSコンポーネントを用いて設計することができる。コンパクションは配線以外のSコンポーネントに対しては識別し、大きさ、形状を変えない。これは素子の特性を保持するためである。

(2) 素子の認識

本コンパクタは、あらかじめ指定したコンポーネントの組み合せに対して、特定の位置関係制約を設定することができる。例えば、トランジスタ素子をゲートコンポーネントと配線コンポーネントとの組合せで定義できるが、この場合、コンパクタは両者の相対位置関係を変えない。これにより、コンポーネントの重ね合わせによって素子を構成するレイアウト設計に対してもコンパクションを適用できる。素子はテクノロジーファイルの記述に基づき、コンポーネント間で図形演算処理を行うことで認識される。

(3) マスク図形の接続判定

本コンパクタは、シンボリックレイアウトに対して適用するが、コンポーネント間のスペーシングや接続関係の判定は、実際のマスク図形の位置に基づいてお

り、コンポーネント間の位置関係を与えるためのルールは特に作成する必要がない。また、コンポーネント間の接続もマスク図形の重なりから自動的に検出される。

(4) エディタに対する考慮

本コンパクタは、データベース中にコンパクション用の特別なデータを加える必要がない。コンパクションの実行は子プロセルを生成して行われるため、実行中も他のセル編集が可能である。

表7-2に本コンパクタによるレイアウト圧縮の例を示す。幾つかの機能拡張を行っているにもかかわらず、実験結果は短時間で高い圧縮率を示している。面積比は人手がシンボリックレイアウトを行ったサイズに対する比較である。

	データ1	データ2	データ3
データサイズ	MOS トランジスタ 16	MOS トランジスタ 22	マクロセル4個 スタンダードセル100個
コンパクション方向	下+左	左+下	上+左
面積比	52.4%	46.0%	84.6%
処理時間	35”28	2’15”28	50’06”45

表2 コンパクションの実験結果

7.4.3 大規模回路用レイアウトツール

レイアウト設計(C)では、自動レイアウトツールとしてスタンダードセル方式レイアウト用自動配置配線(第5章)、マクロセル方式レイアウト用の自動配置配線(第4章)などがあり、さらに階層的レイアウト設計を最適化するためにルールベースを用いたフロアプランニング機能が用意されている(第3章)。

また、モジュール自動生成機能としてレイアウト記述言語(LDL)を備えている。これは本システムのレイアウトデータをLDL表現に自動変換したのち、座標値などの変数化を行ったLDL記述を作成し、それをモジュール自動生成機能のソースコードとするという機能である[19]。

7.4.4 会話型自動配線

レイアウト設計(A)、(B)、(C)に共通する問題としては配線作業の効率化がある。一般に配線作業は多くの時間を要する作業であり、特に遠距離配線を行う場合、折れ曲がり点の座標の確認、デザインルールへの考慮など入力編集が時間を要し、誤りが生じやすい。本システムではこうした人手による配線作業を支援する目的で会話型自動配線機能を実現している。

本機能は次のような会話型機能を装備している。

① 配線経路の指針となる中間点指定、

② 設計者が意図する配線を選択可能とするための経路複数出力、

③ 配線不可能な場合の探索済み領域表示、

④ 2点間、点対ネット、ネット全体などの配線方法の選択、

⑤ 1層、2層配線の選択、

⑥ 配線経路改善機能

などである。

一般に、ネットごとに配線を順次行う場合、以前に行った配線が次の配線時に障害となる。配線経路改善機能は、配線済みの経路を設計者の指定する方向(上下左右)に、デザインルールに従って最大限に移動させることができる。

人手設計の場合、マスクパターン中に斜めを含む任意形状の多角形を含むことが多い。そこで、本システムの会話型自動配線では、任意形状の多角形をいくつかの矩形に分解し近似的に表現して取り扱う。

配線アルゴリズムは、高速性と使用メモリへの考慮から線分探索法[64]を採用している。処理の高速化と遠回りな配線経路の発生を極力防ぐ目的で、探索領域、探索方

向を段階的に制御している。すなわち、第1段階では始点終点を囲む最小矩形領域内でのみ経路探索を行い、経路が発見できない場合、順次この制限を暖める方法を用いている。

表7-3に、幾つかのデータに対する経路探索時間を示しているが、結果は十分な高速性を示している。

データ	障害物数	トランジスタ数	LSIの種類
データ1	7,324	10K	CMOS 8BIT CPU
データ2	4,412	40K	CMOS 信号処理
データ3	5,838	4K	I^2L コントローラ

表7-3(A)　実験データ

データ	CPU TIME
データ1	8
データ2	35
データ3	2

表7-3(B)　会話型配線の実験結果

7.5　システムの設計適用例

本システムは、実際にASIC設計に利用されており、設計の効率化、期間短縮に貢献している。

8ビットのCPUコア、ROM、RAM、信号処理機能ブロック(約1万ゲート)と入出力処理ブロックからなるCMOSカスタムASICの例では、多くのブロックのセルはシンボリック設計、ブロックおよびチップは自動配線によりレイアウト設計が行われ、得られたASICのチップサイズは、人手設計を行った場合の見積りに比べ数%増

大したが、セルライブラリ開発後のレイアウト設計期間は検証を含め約1.2ヵ月で完了できた(図7-6)。

バイポーラアナログLSIのブロック設計への適用例として、トランジスタ数25個、抵抗数11個のテレビ用アナログスイッチで、以下の手順で設計された。

(1) 論理回路設計支援システムを用いて、上記の回路を設計(図7-7)。

(2) 論理回路データベースからレイアウトデータベースに素子及び接続情報を自動変換する。ここで論理回路データベース中の各素子は、対応するレイアウトコンポーネントをあらかじめ指定されている。また、ネット情報は、レイアウトコンポーネント中の端子に信号名として与えられる。配置は回路図の相対配置を保存し、かつカレントミラー回路などを構成するトランジスタについては自動的に認識されグループ化され、隣接配置される。

(3) レイアウトコンパクションおよび配置改善が会話的に行われる。

(4) 素子間配線は会話型自動配線機能を用いて行われる。ネット全体の自動配線をまず行い、配線できない部分に対して会話型配線機能を用いて接続される。

本モジュール設計ではほぼ人手同等の面積を持つモジュールを約1/2のレイアウト設計期間で完了することができた。レイアウト結果を図7-8に示す。

7.6 結言

本システムは、データベースに図形情報以外に素子情報、端子情報などを持つことにより、会話型設計ツールと自動設計ツールの一体化を実現した。また、シンボリックレイアウトや会話型自動配線などにおいてマスク図形の接続、非接続などの考慮を付加し、ディジタルVLSIだけでなく、複雑なマスク図形を用いるアナログVLSIのレイアウト設計の自動化/効率化をも可能とした。本章では、レイアウト設計を総合的に支援するために開発した本システムの概略について述べた。また、実際に設計に適用した結果、本システムによりASICのレイアウト設計が従来の設計に対して、得られるレイアウト面積はほぼ同等で、レイアウト設計期間では、セル(マクロセルを含む)設計およびその検証において約30~50%、チップ設計およびその検証において約50~70%、全体として約40~60%削減できた。会話型設計に自動化

ツールを結合させることにより配線処理を効率化したことや、オンライン検証により設計誤りに対する早い段階での設計フィードバックを可能としたことにより設計の高度化を図ったことが本システムの大きな特徴である。

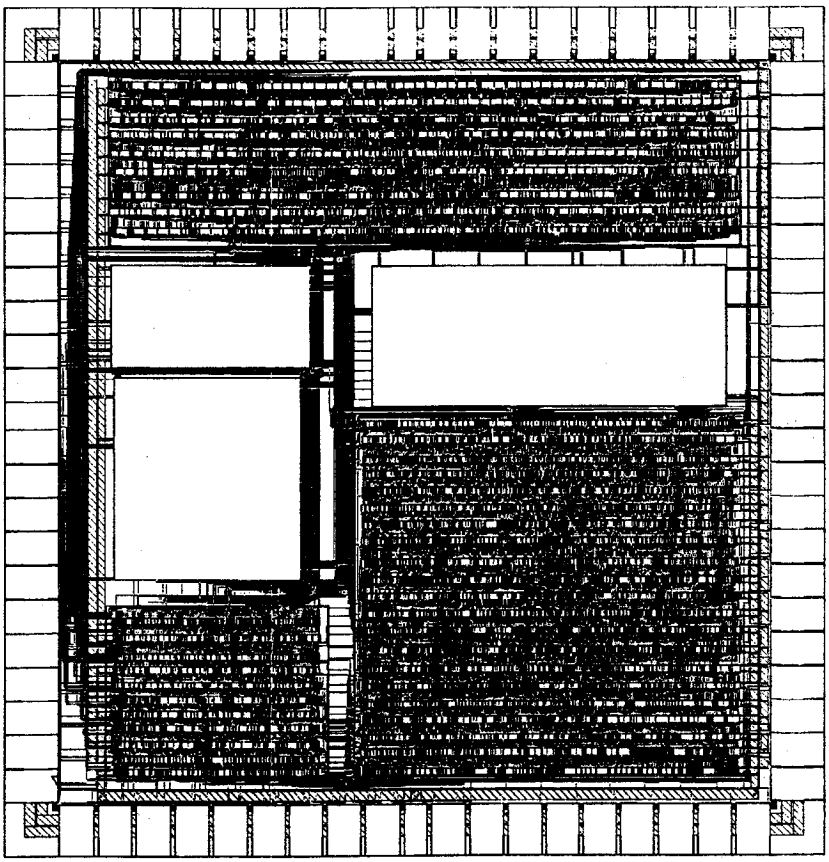


図7-6　CMOSカスタムASICのレイアウト結果

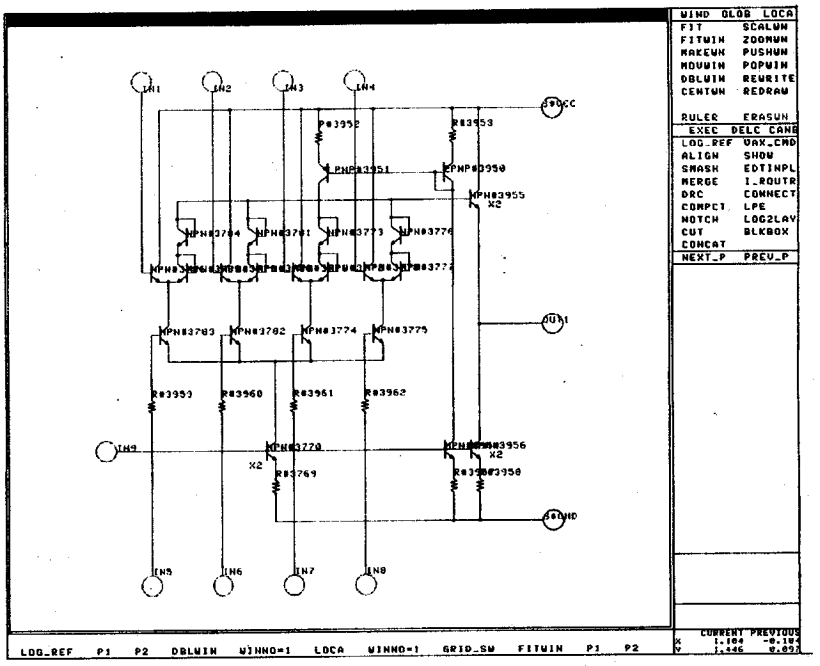


図7-7　トランジスタ回路の例

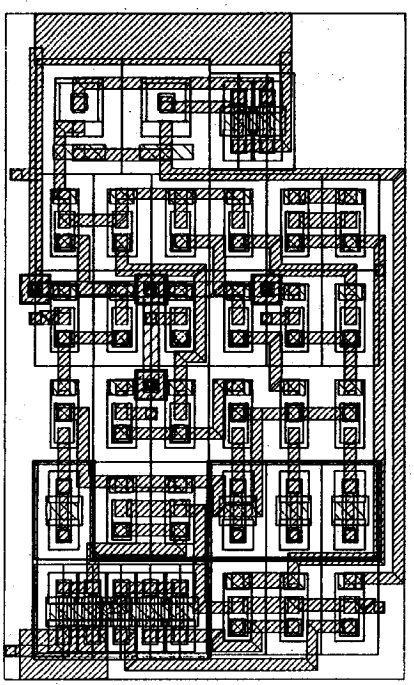


図7-8　アナログモジュールのレイアウト結果

第8章　結論

本研究の目的は、ASICの高密度設計において決定的な役割を果たすレイアウト設計自動化、効率化の技法について考察を行うことであった。そこで、特に、大局的なレイアウト最適化を達成するためのフロアプラニング手法、マクロセル方式自動配線手法、大規模スタンダードセル方式レイアウトにおける自動配置配線手法について考察するとともに、これらと会話型設計支援ツールとを一体化したレイアウト設計支援統合化システムの研究について述べた。

本研究で得られた主要な結果を以下に要約する。

(1) 大規模ASICを設計する場合に、大局的なレイアウト最適化の達成に重要なフロアプラニング問題に対して、複雑多岐にわたる設計条件を総合的に評価し最適化を図るためにAI技法を応用したモジュール配置法を考察した。本手法は、モジュールの隣接関係決定手法と詳細配置を求めるためのヒューリスティックな算法とからなる。特に、モジュールの隣接関係決定手法では、各種設計条件を考慮し隣接配置が望ましいモジュールの組み合せを求めるため、設計者の知識を組み入れたエキスパートシステムを活用している。本配置法は設計者の要求する各種設計条件を考慮しつつ、チップ面積を最小化したモジュール配置が達成できることをいくつかの実験により確認した。

(2) マクロセル方式レイアウトにおける自動配線問題に対して、トポロジカルな配線を決定する初期配線と、設計規則を満足させるために配線の移動を反復する処理からなる新しいチャネル配線手法を考察した。本手法が、配線領域形状の複雑さによる配線迂回や空き領域を最小化できることを実験により確認した。

(3) 大規模スタンダードセル方式レイアウトにおける自動配置配線問題に対して、二つの手法を記述した。まずセル配置問題について、2層メタルによるセル上配線や電気的等価端子を利用する配線がチップ面積の削減に有用であることに着目し、セル行に対して垂直方向にクラスタ配置を行った後に、シャフリング手法を適用し、ネットごとの配線要求の分布がセル行に対して相対的に垂直方向に長くなるセル配置を求める手法を考察し、幾つかの実験によりその性能を確認した。次に概略配線

問題について、セルの持つフィードスルー領域を積極的に利用しつつ、チップ面積増大の大きな要因である局所的な混雑の最小化を図る手法を考察した。あわせてチップ面積最小化に有効と考えられる配線途中でのセルの局所的な移動、チップ周辺での一括配線手法についても述べた。

(4) 自動設計が困難な場合に必要となる人手設計を支援することを目的として構築したレイアウト設計支援統合化システムについて考察した。本システムは、データベースに図形情報以外に素子情報、端子情報などを保有し、会話型設計ツールと自動設計ツールの一体化を実現している。また、シンボリックレイアウトや会話型自動配線など各種ツールにおいて、マスク図形の接続判定機能を付加し、ディジタルVLSIだけでなく、複雑なマスク図形を用いるアナログVLSIのレイアウト設計の自動化/効率化をも可能とする。これらの機能により、本システムがASICレイアウト設計全体の自動化/効率化を強力に支援していることを、実際のASICレイアウト設計に適用し実証した。

今後に残された課題としては、以下のものがある。

(1) 論理設計の自動化

ASICの大規模化はレイアウト設計だけでなく、論理設計においても設計期間の長期化や設計品質の低下などの問題が発生している。レイアウト設計自動化の進歩は論理設計の長期化をさらに顕在化させている。今後、動作レベル、レジスタトランスファレベルなどの各種自動論理合成技術の研究を進める必要がある。

(2) 電気的特性を考慮した自動設計

サブミクロンプロセス技術の実用化により、配線遅延の電気的特性への影響がさらに大きくなり、また大規模化により電源、クロックなどの供給に付随して重要なレイアウト問題が発生する。本研究ではフロアプランニングシステムにおける経路や配線幅指定および維持機能を実現することにより、ある程度の電気的特性を保証しているが、今後は自動配置配線においても電気的特性の考慮が必要である。

(3) CAD統合化のためのフレームワーク技術

CADツールが豊富になりかつ進歩が激しくなる今日、統合化データベースの拡張

性や共通ユーザインターフェイスの提供など、CADシステムの統合化技術が大きな問題である。さらに、複雑多種なCADツールを用いて設計する場合にその工程管理やデータのバージョン管理も重要である。このようなCADツール全体の実装環境とも言うべきフレームワーク技術の確立が望まれている。

謝辞

本論文の作成に当たり、終始御懇切な御指導と御教示を賜った大阪大学工学部情報システム工学科 白川功教授に厚く感謝を申し上げます。論文をまとめるにあたり、有益な御指導と御援助を賜った大阪大学工学部電子工学科 児玉慎三教授、情報システム工学科 寺田浩詔教授、電子工学科 熊谷貞俊教授に厚く感謝申し上げます。

筆者が大阪大学在学中から、つねづね御指導、御教示を頂いている大阪大学工学部電子工学科 尾崎弘名誉教授、応用物理工学科 樹下行三教授、明治大学理工学部情報科学科 藤原秀雄教授、中央大学理工学部電気·電子工学科 築山修治教授に深く感謝致します。

また本研究の遂行にあたり、シャープ株式会社 佐々木正元副社長、岡野孝作前副社長、浅田篤副社長、技術本部 藤本一郎本部長、同本部 河田亨副本部長、情報システム事業本部 西岡郁夫副本部長、IC事業本部超LSI開発研究所 犬伏恒雄副所長、生産技術開発推進本部 生産技術研究所 藤原康宏所長、同研究所 第5研究部 佐原謙一部長には本研究の機会を与えて頂き、生産技術開発推進本部生産技術開発センター 道畑一三所長、同本部CAEセンター 西田久生所長、同本部FA事業部CAD事業センター第1開発室 栗本卓治室長、技術本部 情報技術開発センター第3開発部 千葉徹部長、IC事業本部IC開発センター第3技術部 久保登部長には御理解ある御配慮、御指導、御鞭撻を頂きました。ここに深く感謝します。

研究を行うにあたってVLSI設計技術、VLSICAD実用化に関しては同社IC事業本部ロジック技術センター 石野喜英 所長、同北森宏 副所長、同第3技術部 佐藤雄一 副参事、IC事業本部メモリ技術センター 佐野健二 所長、IC開発センター 宮田慶一 所長、IC事業本部超LSI開発研究所第6開発室 藤井克正 主任研究員、生産技術開発推進本部 CAEセンターICCAE推進室 田中隆吉 室長に、本研究の個々の課題については同本部生産技術研究所第5研究部谷貞宏 係長、藤本徹哉 係長、藤原紳一 主任、富田常雄 副主任、竹田信弘 副主任、野田浩明 副主任をはじめ第5研究部のみなさんに御協力と御援助を頂きました。厚くお礼申し上げます。

本研究に関する発表文献

I. 論文(学会論文誌)

[1] 木村征二, 神戸尚志, 千葉徹, 犬伏恒雄: "LSI設計における配置問題に対する一手法," 電子情報通信学会論文誌(C), Vol. J66-C, No. 12, pp.1148-1155, (1983-12).

[2] T. Kambe and T. Tomita: "A floorplanning scheme for VLSI design," The Institute of Electronics, Information and Communication Engineers Transactions, Vol.E71, No.12, pp.1236-1242, (1988-12).

[3] 山田晃久, 築山修治, 白川功, 神戸尚志: "知識ベースシステムによるCMOS論理セル自動生成," 電子情報通信学会論文誌(A), Vol.J72-A, No.1, pp.95-104, (1989-01).

[4] 神戸尚志, 谷 貞宏, 小嶋 格, 冨田常雄, 森本清巳: "VLSIレイアウト設計のための統合化支援システム," 情報処理学会論文誌, Vol.31, No.3, pp.351-359, (1990-03).

[5] T. Kambe, T. Okada, S. Fujiwara, and C. Yoshioka: "A routing method for macro cell VLSI layout," The Institute of Electronics, Information and Communication Engineers Transactions, Vol.E73, No.12, pp.1979-1988, (1990-12).

II. 論文(査読付き国際学会)

[1] T. Chiba, N. Okuda, T. Kambe, I. Nishioka, T. Inufushi, and S. Kimura: "SHARPS: A hierarchical layout system for VLSI," Proc. 18th Design Automation Conference, pp.820-827 (1981).

[2] T. Kambe, T. Chiba, I. Shirakawa, S. Kimura, T. Inufushi, and N. Okuda : "A placement alogorithm for polycell LSI and its evaluation," Proc. 19th Design Automation Conference, pp.655-662 (1982).

[3] T. Chiba, N. Okuda, T. Kambe, I. Nishioka, T. Inufushi, and S. Kimura: "A general cell layout system for VLSI," Proc. IEEE Int. Symp. on Circuits and Systems, pp.1013-1016 (1982).

[4] T. Chiba, N. Kubo, T. Kambe, I. Nishioka, T. Inufushi, and S. Kimura: "SHARPS: A CAD system for VLSI," Proc. IEEE Int. Conf. on Computer Design, pp.532-535 (1983).

[5] T. Kambe, T. Okada, T. Chiba, and I. Nishioka : "A global routing scheme for polycell LSI," Proc. IEEE Int. Symp. on Circuits and Systems, pp.187-191 (1985).

[6] T. Kambe, M. Matumoto, and I. Kojima: "A comprehensive set of layout design tools for VLSI," Proc. IFIP TC-10 Working Conf., pp.113-125 (1988).

[7] T. Kambe and T. Tomita: "A floorplanning scheme for VLSI design," Proc. JTC-CSCC88, pp.293-298 (1988).

[8] K. Okuda, H. Noda, H. Matsumoto, T. Matsumoto, T. Kambe, and H. Kitamori: "Automatic test pattern generation for ASICs with partial scan path designs," Proc. Int. workshop on design and test of ASICs, pp.59-61 (1990).

[9] T. Okada, A. Yamada, T. Kambe, and I. Shirakawa: "A analog module generator," Proc. Third Int. Forum on ASIC and Transducer Technology, pp.94-100 (1990).

[10] C. Yoshioka, T. Okada, S. Fujiwara, and T. Kambe: "A routing method for macro cell VLSI layout," Proc. JTC-CSCC90, pp.507-512 (1990).

[11] S. Fujiwara, T. Okada, C. Yoshioka, A. Yamada, and T. Kambe: "A routing method for macro cell VLSI layout," Proc. IEEE Int. Symp. on Circuits and Systems, pp.pp.23-26 (1991).

III. 解説論文(学会誌)

[1] 神戸尚志, 白川 功:“ASICの設計環境,” 電子情報通信学会誌, Vol.73, No.10, pp.1058-1065 (1990-10).

IV. 論文(査読付き国内学会)

[1] 神戸尚志, 富田常雄: “VLSIチッププランニングにおけるAI応用の検討,” 電子情報通信学会第1回回路とシステム軽井沢ワークショップ, pp.229-234 (1988).

[2] 神戸尚志, 長尾明, 山内貴行, 松本美佐代: “アナログLSIにおけるレイアウト自動化に関する手法,” 電子情報通信学会第2回回路とシステム軽井沢ワークショップ, pp.168-175 (1989).

[3] 神戸尚志, 岡田時仁, 藤原紳一, 吉岡智良: “マクロセル方式レイアウト対応チャネルルータの一手法,” 電子情報通信学会第3回回路とシステム軽井沢ワークショップ, pp.310-317 (1990).

V. 論文(研究会等)

[1] 神戸尚志, 千葉徹, 奥田登, 西岡郁夫, 犬伏恒雄, 木村征二: “LSIの階層的設計のための会話型自動配線プログラム,” 信学技報, CAS80-51 (1980).

[2] 千葉徹, 奥田登, 神戸尚志, 西岡郁夫, 犬伏恒雄, 木村征二: “VLSIレイアウト設計システムSHARPSについて,” 情報処理学会電子装置設計技術研究会資料, 8-1 (1981).

[3] 神戸尚志, 中畑元行, 久保登, 千葉徹, 犬伏恒雄, 木村征二: “高速CMOSゲートアレイシステム,” シャープ技報, 第24号, pp.35-41 (1980).

[4] 谷貞宏, 神戸尚志, 白川功, 尾崎弘: “レイアウト設計における配置手法の評価,” 信学技報, CAS82-146 (1983).

[5] 神戸尚志, 米田浩, 岡田時仁: “ポリセル方式LSIにおける一配線手法,” 情報処理学会夏期シンポジウム, (1984).

[6] 岡田時仁, 神戸尚志, 西岡郁夫: “スタンダードセル方式LSIの一配線手法,” 信学技報, SSD85-82 (1985).

[7] 神戸尚志, 藤原紳一, 小嶋格, 西岡郁夫: "ビルディングブロック型スタンダードセル方式LSIの一配置手法," 情報処理学会設計自動化研究会資料, 30-4, (1986).

[8] 神戸尚志, 藤原紳一, 岡田時仁, 小嶋格, 西岡郁夫: "ビルディングブロック型スタンダードセル方式LSIの自動レイアウトシステムSHARPSIIについて," 信学技報, CAS85-141 (1986).

[9] 山内貴行, 谷貞宏, 神戸尚志, 西岡郁夫: "ストレッチシンボル対応レイアウトコンパクタ," 情報処理学会設計自動化研究会資料, 34-9 (1986).

[10] 神戸尚志, 藤原紳一, 岡田時仁, 小嶋格: "ビルディングブロック型スターダードセル方式LSIの自動レイアウトシステム," シャープ技報, 第36号, pp51-56 (1986).

[11] 神戸尚志, 谷貞宏, 小嶋格, 富田常雄, 吉富久子, 吉岡智良, 山内貴行, 松本美佐代, 原嶋勝美, 長尾明: "超LSI設計支援システム," 信学技報, VLD87-9 (1987).

[12] 原嶋勝美, 若林謙二, 神戸尚志, 重弘裕二, 白川功: "C言語によるレイアウト記述の一手法," 信学技報, CAS87-14 (1987).

[13] 富田常雄, 神戸尚志: "ビルディングブロック方式LSIにおけるルールベースを用いた一配置手法について," 信学技報, VLD87-96 (1987).

[14] 藤本徹哉, 野田浩明, 神戸尚志: "制御信号系にもとづくタイミング検証," 信学技報, VLD87-107 (1987).

[15] 野田浩明, 松本敏行, 藤原紳一, 藤本徹哉, 奥村憲三, 神戸尚志: "スキャン構造を意識したテストパターン生成," 信学技報, VLD87-110 (1987).

[16] 山内貴行, 岡田時仁, 三浦順子, 神戸尚志: "フロアプランにおける配線経路指定の一手法," 情報処理学会設計自動化研究会資料, 43-3 (1988).

[17] 長尾明, 小嶋格, 神戸尚志: "任意角度辺を含むLSIマスクパターンの拡大縮小手法," 情報処理学会設計自動化研究会資料, 43-4 (1988).

[18] 神戸尚志, 山内貴行, 松本美佐代, 長尾明: "アナログLSI設計支援システムの検討," 情処学会DAシンポジウム88 (1988).

[19] 竹田信弘, 三浦順子, 神戸尚志: "CMOS用多段論理合成," 信学技報, VLD88-83 (1988).

[20] 神戸尚志: "ASIC設計支援統合システム," シャープ技報, 第40号, pp.81-86 (1988).

[21] 富田常雄, 松本美佐代, 岡田時仁, 神戸尚志: "VLSIにおけるフロアプランニング," 情報処理学会設計自動化研究会資料, 46-2 (1989).

[22] 藤原紳一, 神戸尚志: "大規模LSI設計における配置問題に対する一手法," 信学技報, VLD88-86 (1989).

[23] 山内貴行, 富田常雄, 松本美佐代,神戸尚志: "VLSIフロアプランニング," 信学技報, VLD89-44 (1989).

[24] 竹田信弘, 三浦順子, 神戸尚志: "論理合成技術," シャープ技報, 第43号, pp.73-76 (1989).

[25] 三浦順子, 竹田信弘, 神戸尚志: "多段論理合成における時間最適化の一手法"," 信学技報, VLD89-85 (1989).

[26] 岡田時仁, 神戸尚志: "グリッドレススイッチボックス配線の一手法," 情報処理学会設計自動化研究会資料, 51-4 (1990).

[27] 神戸尚志, 岡田時仁, 藤原紳一, 吉岡智良: "A routing scheme for macro cell VLSI layout," 信学技報, VLD90-37 (1990).

[28] 藤本徹哉, 神戸尚志: "許容関数に基づく組合せ論理回路の遅延最適化" 信学技報, VLD90-90 (1990).

[29] 藤原紳一, 岡田時仁, 吉岡智良, 中谷美千世, 神戸尚志: "チッププランニングシステムにおけるマクロセル間配線の一手法について," シャープ技報, 第47号, pp.32-38 (1990).

[30] 藤本徹哉, 神戸尚志: "論理回路の自動合成における遅延時間最適化手法," シャープ技報, 第49号, pp.31-36 (1991).

[31] 吉岡智良, 山田晃久, 中谷美千世, 藤原紳一, 神戸尚志: "マクロセル方式高密度配線手法について," 情報処理学会設計自動化研究会資料, 58-6 (1991).

[32] 神戸尚志: "ASICレイアウト設計支援技術の現状," 信学技報, VLD91-41 (1991).

VI. 論文(講演会、口頭発表)

[1] 神戸尚志, 千葉徹, 西岡郁夫, 犬伏恒雄, 木村征二: "SHARPSの初期配置プログラムIPLACEについて," 昭和56年電子通信学会総合全国大会, 436 (1981).

[2] 西本猛史, 神戸尚志, 木村征二, 青木良正: "CMOSランダムロジックの論理抽出法," 昭和56年電子通信学会総合全国大会, 454 (1981).

[3] 千葉徹, 奥田登, 神戸尚志, 西岡郁夫, 犬伏恒雄, 木村征二: "階層的LSI設計システムSHARPSについて," 昭和56年電子通信学会総合全国大会, 435 (1981).

[4] 千葉徹, 奥田登, 神戸尚志: "LSI設計のためのCADシステムSHARPS," 昭和58年電子通信学会情報·システム部門全国大会, S1-4 (1983).

[5] 谷貞宏, 神戸尚志, 吉田健一, 小嶋格, 千葉徹, 犬伏恒雄: "LSIのワイヤボンディング設計のためのシミュレータ," 昭和58年電子通信学会情報·システム部門全国大会, S1-8 (1984).

[6] 宮野宏, 中谷仁継, 神戸尚志, 斉藤寿士, 犬伏恒雄: "ゲートアレイ用論理設計支援システムGENESIS(1)ーシステム構成とその特徴についてー," 情報処理学会第30回全国大会, 4H-4 (1986).

[7] 林省吾, 神戸尚志, 犬伏恒雄: "ゲートアレイ用論理設計支援システムGENESIS(4)ー会話型自動配線プログラムについてー," 情報処理学会第30回全国大会, 4H-7 (1986).

[8] 神戸尚志, 藤原紳一, 岡田時仁, 小嶋格: "ビルディングブロック型スタンダードセル方式LSIの自動レイアウトシステム," 昭和61年電気関係学会関西支部連合大会, G6-3 (1986).

[9] 谷貞宏, 小嶋格, 森本清己, 吉田健一, 神戸尚志, 西岡郁夫: "超LSI設計支援システム(1)ーシステム構成ー," 情報処理学会第33回全国大会, 5R-6 (1986).

[10] 谷貞宏, 富田常雄, 森本清己, 神戸尚志, 西岡郁夫: "超LSI設計支援システム(2)ーレイアウトエディターー," 情報処理学会第33回全国大会, 5R-7 (1986).

[11] 山内貴行, 神戸尚志, 西岡郁夫: "超LSI設計支援システム(3)ーレイアウトコンパクターー," 情報処理学会第33回全国大会, 5R-8 (1986).

[12] 虫上秀昭, 長尾明, 谷貞宏, 神戸尚志, 西岡郁夫,今井正治: "超LSI設計支援システム(5)ーオンラインDRCー," 情報処理学会第33回全国大会, 5R-10 (1986).

[13] 原嶋勝美, 谷貞宏, 神戸尚志: "レイアウト記述言語を用いたセルコンパイラ," 昭和62年電子情報通信学会総合全国大会, 334 (1987).

[14] 藤本徹哉, 山内貴行, 西本包子, 神戸尚志: "論理シミュレーションによるレイアウト検証," 昭和62年電子情報通信学会総合全国大会, 343 (1987).

[15] 松本敏行, 松本比呂志, 吉田茂, 神戸尚志: "未検出故障解析を容易にした故障シミュレータ," 昭和62年電子情報通信学会総合全国大会, 318 (1987).

[16] 松本美佐代, 小嶋格, 神戸尚志: "VLSIレイアウト設計における会話型自動配線," 情報処理学会第35回全国大会, 6H-6 (1987).

[17] 岡田時仁, 小嶋格, 神戸尚志: "スタンダードセル方式LSIにおける一配線手法," 昭和63年電子情報通信学会春季全国大会, A-268 (1988).

[18] 山内貴行, 岡田時仁, 神戸尚志: "フロアプランにおける配線経路指定の一手法," 昭和63年電子情報通信学会春季全国大会, A-267 (1988).

[19] 長尾明, 小嶋格, 神戸尚志: "任意角度を含むLSIマスクパターンの拡大縮小手法," 情報処理学会第36回全国大会, 4Y-1 (1988).

[20] 谷貞宏, 竹田信弘, 大塚玄洋, 西本包子, 原嶋勝美, 山口雅之, 神戸尚志: "機能論理設計支援システム," 情報処理学会第36回全国大会, 2X-4 (1988).

[21] 藤原紳一, 岡田時仁, 竹田信弘, 神戸尚志: "階層的スタンダードセル方式LSIレイアウト手法の評価," 情報処理学会第36回全国大会, 3Y-4 (1988).

[22] 神戸尚志: "ASICにおけるレイアウト設計の現状," 電子情報通信学会「ASIC設計技術の現状と動向」講習会, pp.19-30 (1988).

[23] 富田常雄, 神戸尚志: "VLSIにおけるフロアプランについて," 昭和63年電子情報通信学会秋季全国大会, SA-6-2 (1988).

[24] 奥井千香, 富田常雄, 小嶋格, 神戸尚志: "階層的データ構造における表示高速化の一手法について," 情報処理学会第37回全国大会, 5U-8 (1988).

[25] 長尾明, 山内貴行, 松本美佐代, 下西之園正明, 神戸尚志: "アナログLSIにおけるモジュール設計支援システムについて," 1989年電子情報通信学会春季全国大会, SA-7-3 (1989).

[26] 原嶋勝美, 谷貞宏, 山口雅之, 神戸尚志: "論理設計支援システムにおけるリアルタイム図面検証," 1989年電子情報通信学会春季全国大会, A-253 (1989).

[27] 神戸尚志, 野田浩明: "ASIC設計法とそのツール," 電子情報通信学会関西支部「ASIC技術とその応用」専門講習会, pp.41-55 (1989).

[28] 岡田時仁, 神戸尚志: "グリッドレススイッチボックス配線の一手法," 1990年電子情報通信学会春季全国大会, A-121 (1990).

[29] 山田晃久, 岡田時仁, 神戸尚志: "アナログ機能ブロック内配置の一手法," 1990年電子情報通信学会春季全国大会, SA-3-4 (1990).

[30] 長尾明, 山内貴行, 乙井美佐代, 神戸尚志: "アナログ機能ブロックのレイアウト設計支援システム," 情報処理学会第42回全国大会 5J-2 (1991).

[31] 竹田信弘, 原嶋勝美, 山内順子, 神戸尚志: "論理回路の時間最適化手法," 情報処理学会第42回全国大会 2J-6 (1991).

[32] 中谷美千世, 吉岡智良, 藤原紳一, 岡田時仁, 神戸尚志: "チッププランニングシステムにおけるマクロセル間配線の一手法について," 情報処理学会第42回全国大会 6J-1 (1991).

[33] 野田浩明, 高橋瑞樹, 横山昌夫, 神戸尚志: "レジスタ転送レベル自動合成システム(1)ーシステム構成" 1991年電子情報通信学会春季全国大会, A-122 (1991).

[34] 高橋瑞樹, 野田浩明, 横山昌夫, 神戸尚志: "レジスタ転送レベル自動合成システム(2)ーデータパス合成" 1991年電子情報通信学会春季全国大会, A-123 (1991).

参考文献

第1章

[1] 本間明："ASICとその歴史," ASICデザインハンドブック, サイエンスフォーラム社, pp11-24 (1989).

[2] 平井誠一："ASICの展望," 電子情報通信学会誌, Vol.71, No. 2, pp. 172-180 (1988).

[3] 内田幸正："ASIC設計方式," ASICデザインハンドブック, サイエンスフォーラム社, pp91-126 (1989).

[4] D. Johannsen : "Bristle blocks : a silicon compiler," Proc. 16th Design Automation Conference, pp310-313 (1979).

[5] 神戸尚志："ASICにおけるレイアウト設計," 電子情報通信学会『ASIC設計技術の現状と動向』講習会資料, pp.19-30 (1988).

第2章

[6] T. Chiba, N. Okuda, T. Kambe, I. Nishioka, T. Inufushi, and S. Kimura: "SHARPS: a hierarchical layout system for VLSI," Proc. 18th Design Automation Conference, pp820-827 (1981).

[7] 谷貞宏, 竹田信弘, 大塚玄洋, 西本包子, 原嶋勝美, 山口雅之, 神戸尚志："機能·論理設計支援システム," 情報処理学会第36回全国大会, pp1909-1910, (1988).

[8] 吉田茂, 西本猛, 久保登, 千葉徹："MOSLSI用会話型論理シミュレーションの一手法," 信学技報, CAS83-193 (1984).

[9] 藤本徹哉, 野田浩明, 神戸尚志："制御系に基づくタイミング検証," 信学技報, VLD87-107 (1987).

[10] 松本敏行, 松本比呂志, 吉田茂, 神戸尚志："未検出故障解析を容易にした故障シミュレータ," 昭和62年電子通信学会全国大会, P2-122 (1987).

[11] 野田浩明, 松本敏行, 藤原紳一, 奥村憲三, 神戸尚志："スキャン構造を意識したテストパターン生成," 信学技報, VLD87-110 (1987).

[12] 富田常雄, 松本美佐代, 岡田時仁, 神戸尚志："VLSIにおけるフロアプランニング," 情報処理学会 設計自動化研究会資料, 46-2 (1989).

[13] T.Kambe and T. Tomota. : "A floorplanning scheme of VLSI design," IEICE Trans., Vol. E 71, No. 12, pp. 1236-1242 (1988).

[14] 富田常雄, 松本美佐代, 岡田時仁, 神戸尚志 : "VLSIにおけるフロアプランニングについて－配線領域見積り,端子位置決定－," 情報処理学会 設計自動化研究会資料, 46-2 (1989).

[15] 山内貴行, 岡田時仁, 三浦順子, 神戸尚志: "フロアプランにおける配線経路指定の一手法," 情報処理学会 設計自動化研究会資料, 43-3(1988).

[16] I. Kojima, N. Kubo, M. Hayashi, T. Chiba, and I. Nishioka: "Global routing scheme for gate array layout," Proc. 7th Europian Conf. Circuit Theory and Design (1985).

[17] 神戸尚志, 藤原紳一, 岡田時仁, 小嶋格, 西岡郁夫: "ビルディングブロック型スタンダードセル方式LSIの自動レイアウトシステムSHARP2について," 信学技報, CAS85-141 (1986).

[18] S. Kimura, N. Kubo, T. Chiba, and I.Nishioka : "An automatic routing scheme for general cell LSI," IEEE Trans. on CAD, Vol. CAD-2, No.4, pp285-292 (1983).

[19] 原嶋勝美, 若林謙二, 神戸尚志, 重弘裕二, 白川功 : "C言語によるレイアウト記述言語の一手法," 信学技報, CAS87-14 (1987).

[20] 山内貴行, 谷貞宏, 神戸尚志, 西岡郁夫 : "ストレッチシンボル対応レイアウトコンパクタ," 情報処理学会 設計自動化研究会資料, 43-9 (1987).

[21] 松本美佐代, 小嶋格, 神戸尚志: "VLSIレイアウト設計における会話型自動配線," 情報処理学会第35回全国大会, pp2441-2442 (1987).

第3章

[22] 竹田信弘, 三浦順子, 神戸尚志 : "CMOS多段論理合成," 信学技報, VLD88-83 (1988).

[23] 三浦順子, 竹田信弘, 神戸尚志: "多段論理合成における時間最適化の一手法"," 信学技報, VLD89-85 (1989).

[24] 藤本徹哉, 神戸尚志: "許容関数に基づく組合せ論理回路の遅延最適化" 信学技報, VLD90-90 (1990).

[25] 白川功 : "実装設計における配置配線技法の動向," 電子通信学会誌, Vol. 61, No. 3, pp.245-255 (1978).

[26] R. H. M. Otten : "Automatic floorplan design," Proc. 19th Design Automation Conference, pp.261-267 (1982).

[27] R. H. M. Otten : "Efficient floorplan optimization," Proc. ICCAD, pp.499-502 (1983).

[28] D. F. Wong and C. L. Liu : "A new algorithm for floorplan design," Proc. 23rd Design Automation Conference, pp.101-107 (1986).

[29] W. R. Heller, G. Sorkin, and K. Maling : "The planar package planner for system designers," Proc. 19th Design Automation Conference, pp.253-260 (1982).

[30] K. A. Kozminski and E. Kinnen : "An algorithm for finding a rectangular dual of a planar gragh for use in area planning for VLSI integrated circuits" Proc. 21th Design Automation Conference, pp.655-656 (1984).

[31] J. Bhasker and S. Sahni : "A linear algorithm to find a rectangular dual of a planar triangulated graph," Proc. 23rd Design Automation Conference, pp.108-114 (1986).

[32] B. Lokanathan and E. Kinnen : "Performance optimized floor planning by graph planarization," Proc. 26th Design Automation Conference, pp.116-121 (1989).

[33] W. M. Dai, B. Eschermann, E.S. Kuh and M. Pedram : "Hierarchical placement and floorplanning for BEAR," IEEE Trans. on CAD, Vol. 8, No. 12, pp. 1335-1349 (1989).

[34] T. Saito, H. Sugimoto, M. Yamazaki, and N. Kawato :" A rule-base logic circuit synthesis system for CMOS gate arrays," Proc. 23rd Design Automation Conference, pp. 594-600 (1986).

[35] G. Odawara, K. Iijima, and Wakabayashi :"Knowledge-based placement technique for printed wiring boards," Proc. 22nd Design Automation Conference, pp. 616-622 (1985).

[36] H. Watanabe and B. Ackland :"Flute - a floorplanning agent for full custom VLSI design," Proc. 23rd Design Automation Conference, pp. 601-607 (1986).

第4章

[37] T. Yoshimura and E. S. Kuh : "Efficient algorithms for channel routing," IEEE Trans. on CAD, Vol. CAD-1, No. 1, pp. 25-35 (1982).

[38] M. Burstein and R. Pelavin : "Hierarchical wire routing," IEEE Trans. on CAD, Vol. CAD-2, No. 4, pp223-234 (1983).

[39] A. Sangiovanni-Vincentelli, M. Antomauro, and J. Reed : "A new guidless channel router the second (YACR-Ⅱ)," Proc. ICCAD, pp. 72-75 (1984).

[40] H. Chen and E. S. Kuh : "Glitter: A gridless variable width channel router," IEEE Trans. on CAD, Vol. CAD-5, No. 4, pp459-465 (1983).

[41] R. Joobbani and D. P. Siewiorek : "Weaver : A knowledge based routing expert," IEEE Design & Test, Vol. 3, No. 1, pp12-33 (1986).

[42] Y. Shin and A. Sangiovanni-Vincentelli : "MIGHTY : A 'rip-up and reroute' detailed router," Proc. ICCAD, pp. 2-5 (1986).

[43] J. P. Cohoon and P. L. Heck : "Beaver : A computational-geometry-based Tool for switchbox routing," IEEE Trans. on CAD, Vol. CAD-5 , No. 4, pp459-465 (1983).

[44] 山内貴行, 藤原紳一, 岡田時仁, 吉岡智良, 神戸尚志 : "ビルディングブロック型LSIレイアウトシステム," 情報処理学会第39回全国大会 (1989).

[45] 山内貴行, 富田常雄, 松本美佐代, 神戸尚志 : "VLSIフロアプランニングシステム," 信学技報, VLD89-44, pp. 47-53 (1989).

[46] C.Y. Lee, :"An Algorithm for path connection and its applications," IRE. Trans. EC-10, pp. 346-365 (1961).

[47] W. -M. Dai, T. Asano, and E. S. Kuh : "Routing region definition and ordeing sheme for building-block layout," IEEE Trans. on CAD, Vol. CAD-4, No. 3, pp189-196 (1985).

[48] 福井正博, 山本敦志, 山口龍一, 羽山繁, 間野洋治郎 : "階層レイアウトシステムにおける自動配線の一手法," 信学技報, CAS85-149 (1986).

[49] M. T. Shing and T. C. Hu : " Computational Complexity of Layout Problems," in Advances in CAD for VLSI, Vol..4 ' Layout design and verification' ed. T. Ohtsuki, pp.199-266 (1986).

[50] D. N. Deutsch : "Compacted channel routing," Proc. ICCAD, pp. 223-225 (1985).

[51] 佐藤政生, 中島伸佳, 金整範, 大附辰夫 : "レイアウトシステムにおける多機能チャネルスペース," 信学技報 VLD88-11, pp, 81-87 (1988).

第5章

[52] M. Hanan, P. Woff, and B. J.Agule : "A study of placement technique," Journal of Design Automation and Fault-Tolerant Computing, 1, 1 (1976).

[53] 木村征二, 神戸尚志, 千葉徹, 犬伏恒雄 : "LSI設計における配置問題に対する一手法," 信学論(C), Vol. J66-C, No. 12, pp.1148-1155 (1983-12).

[54] M. Breuer : "Min-cut placement ," Journal of Design Automation and Fault-tolerant Computing, 1, 4, Oct. (1977).

[55] C. Sechen and A. Sangiovanni-Vincentelli : "The timberwolf placement and routing package," IEEE Journal of Solidstate Circuits, Vol. SC-20, No.2, pp.510-522, (1985).

第6章

[56] K. Aoshima and E. S. Kuh : "Multi-channel optimization in gate array layout," Proc. IEEE Int. Symp. on Circuits and Systems, pp.1005-1008 (1983).

[57] S.Kimura, N. Kubo, T. Kambe, and I. Nishioka : "An automatic routing scheme for general cell LSI," IEEE Trans. on CAD. Vol. CAD-2, No. 4 1983.

[58] A. Sangiovanni-Vincentelli : "Automatic layout of integrated circuits," in Design systems for VLSI circuits, ed. G. D. Micheli, pp.113-196 (1987).

[59] 奥井千香, 富田常雄, 小嶋格, 神戸尚志: "階層的データ構造における表示高速化の一手法について," 情報処理学会第37回全国大会, 5U-8 (1988).

[60] 森本清己, 吉田健一, 内芝伸彦, 田中伸一, 三宅隆一郎 : "超LSI設計支援システム(4)ー階層的ERCー," 情報処理学会第33回全国大会, 5R-9 (1986).

[61] 藤本徹哉, 山内貴行, 西本包子, 神戸尚志 : "論理シミュレーションによるレイアウト検証," 昭和62年電子情報通信学会全国大会 (1987)

[62] D. A. Mlynski and C.-H. Sung : " Layout compaction," in Advances in CAD for VLSI, Vol..4 ' Layout design and verification' ed. T. Ohtsuki, pp.199-266 (1986).

[63] 山内貴行, 谷貞宏, 神戸尚志, 西岡郁夫 : "ストレッチシンボル対応レイアウトコンパクタ," 情報処理学会設計自動化研究会資料, 34-9 (1986).

[64] D. W. Hightower : " A solution to line-routing problems on the continuous plane," Proc. 6th DA workshop, pp.1-24 (1969).