

## ビアの削減を目的とした階層的概略配線手法

金沢 正博 田中 秀彦 佐藤 政生 大附 辰夫

早稲田大学理工学部

あらまし

ゲートアレイやビルディングブロック・レイアウトに対する概略配線手法として、線形割当てに基づく階層的概略配線手法が提案されている。この手法は配線領域全体を概略格子によって分割した矩形領域を処理単位としたもので、再帰的に領域を分割し、適当なコストに基づいて最終的に各ネットの通過する領域を決定する。しかし、この手法では、概略径路のパターンが複雑化することにより、この段階でのビアの数が多くなってしまう。そこで、端子位置主導型のコスト関数を導入することでビア削減を目指した概略配線手法を提案し、計算機実験によってその効果を確認する。また、この手法を、平面配線が得意で柔軟なトポロジカルなレイアウト表現を用いたスケッチレイアウトシステムのための概略配線手法に応用するため、層割当て手法と端子位置の決定手法について提案する。

## A Hierarchical Global Routing Algorithm with Via Reduction

Masahiro KANAZAWA, Hidehiko TANAKA, Masao SATO, and Tatsuo OHTSUKI

School of Science and Engineering, Waseda University  
3-4-1 Okubo, Shinjuku, Tokyo 169

Abstract

Several hierarchical global routing algorithms based on linear assignments have been proposed and widespread. They bi-partition a given routing region recursively and assign a global position for each net crossing a cut-line by using linear assignments according to defined costs. A drawback of the algorithms is to generate zigzag routing patterns, which could lead a lot of vias. A new hierarchical global routing algorithm which generates less vias by improving cost functions is presented in this paper. Experimental results show its efficiency. Algorithms for layer assignment and determining terminal positions are also presented. Those algorithms are necessary to apply the presented global router to sketch layout systems.

## 1 はじめに

ゲートレイやビルディングブロック・レイアウトに対する概略配線手法として、線形割当てに基づくトップダウン概略配線手法が提案されている [1][2]。これらの手法はいずれも配線領域全体を概略格子によって分割した矩形領域を処理の単位としたもので、適当なコストに基づいて各ネットの通過する領域を決定する。一般に概略配線の目的は、大規模な配線問題をいくつかの小問題に分割して詳細配線の負担を軽減すること、総配線長最小化や混雑度の平均化などの最適化を行うものであるが、これらの手法は、この条件をほぼ満たすような結果を得ている。[1]の手法に対して、[2]では、コスト関数と端子の割当て方に工夫を持たせ、混雑度の平均化と高い配線達成率を実現している。

高い配線率を達成するためには通常、配線層には縦もしくは横方向といった優先層を設定している。そこで、概略配線で求めた概略経路の縦の部分集合は縦の優先層に、横の部分集合は横の優先層に割当てておくことを考える。このとき、縦と横の部分集合の交点は、概略経路上での必要なビアとなる。ところで、配線の質を考えた場合、ビアは、電気的には遅延の原因となり、レイアウトの面から見れば面積を多く取るため、できるだけ少ないことが望ましい。ところが、このように割当てたとき、[2]では、縦と横の配線の部分集合が数多くできるため、ビアの数が多くなってしまふ。そこで本稿では、この手法に新たなコスト関数を導入することで、この手法の利点を保つことができ、かつ、ビアの削減を目指した概略配線手法を提案する。また、本手法を計算機に実装し、評価実験を行った結果についても報告する。スケッチレイアウトシステム [3] では、トポロジカルなレイアウト表現により、柔軟なレイアウト設計が可能で、平面配線に威力を発揮する。提案手法をこのシステムの概略配線に適用するためには、平面配線を得る必要があるが、そのために、層割当てと端子位置の決定という処理を行う必要がある。この層割当て手法と端子位置の決定手法も併せて提案する。

## 2 線形割当てに基づく概略配線手法

まず、線形割当てに基づく概略配線手法 [1][2] について述べる。この概略配線手法は、配線順序で結果が大きく変わってしまうようなネット 1 つずつの配線ではなく、一括してすべてのネットの概略経路をトップダウンに決定することで大域的に最適化が考慮され、混雑度の平均化が配線領域全体にわたって図ることができる。

### 2.1 レイアウトモデル

扱う多層配線のモデルは、以下の通りである。レイアウトの対象としては、MCM、高密度プリント配線板などを考えている。

1. 配線領域の面積、形状および各ネットの端子の位置はあらかじめ決定されている。端子は最上位の配線層に置かれ、ビアを用いて下位の層に引き出すことができる。ビアは任意の層間に設けることができる。
2. 層数については 2 層以上であるということ以外、特に制限を設けない。各層における配線方向は優先方向を持つものとする。

準備として、以下の用語を定義する。

**概略格子** 概略配線のためにチップ上に仮想的に設定した粗い格子。概略格子は端子や禁止領域の配置とは無関係に設定される。

**領域** 概略格子によって分割された小領域。処理の単位となる。

**格子境界** 領域を構成する境界線分。各境界は、属性として配線容量を与えられている。

**仮端子** 配線が境界を通過する場合にその境界上に割当てられる仮想的な端子。概略経路はこの仮端子の並びとして表現される。

**カットライン** 配線領域を 2 分割するための水平または垂直の線分。カットラインは必ず格子境界上に引くものとする。

**オーバーフロー** カットラインを横切るネットの総数がカットライン全体の配線容量を超えたためにそのままではネットを割当てることができない状態。

図 1 にモデルの概念図を示す。

### 2.2 概略配線アルゴリズム

まず、入力と出力についてまとめる。

**Input** 各ネットの端子位置。

**Output** 境界の各ネットの概略経路。層は考慮しない。

すべての格子境界で配線容量を超えないように、各ネットの概略経路を決定する。目的関数は仮想総配線長の最小化および混雑度の平均化である。この手法は、配線領域を再帰的に 2 分割し、その分割線上のネット通過位置を線形割当て [4] によって決定する

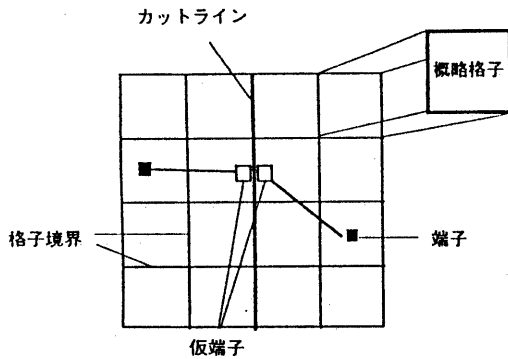


図 1: 概略配線モデル

という高速なトップダウン処理に基づいている。配線の通過境界の決定は、カットラインによって領域を再帰的に2分割する際に、そのカットラインと交差するネットの仮端子をカットライン上の境界に割当てることによって行われる。この手法は、以下のような特徴がある。

1. トップダウン的な再帰処理に基づいているためアルゴリズムが比較的単純である。
2. 領域全体わたって配線の混雑度を平均化できる。
3. 複数のネットを同時に処理するため、配線順序による影響を回避できる。
4. 仮端子を割当て際のコスト設定によりさまざまな結果を得ることができる。

### 2.2.1 処理の概要

概略配線アルゴリズム [2] は、基本的には文献 [1] の方法に基づいたものだがカットライン上の配線容量の計算と、格子境界への仮端子の割当て処理に改良を加えてある。この手法の基本操作はカットラインを通過するネットすべてを同時に考慮する一括処理に基づいたもので、ネット毎にスタイナー木を作るような逐次的な処理ではない。この基本操作を再帰的に繰り返すことにより、最終的に全境界上に仮端子が割当てられる。各レベルにおいて、ネットが通過する格子境界は線形割当てによって決定される。以下にアルゴリズムの全体の流れを示す。

#### 概略配線アルゴリズム

- Step 1 チップ全体の領域を待ち行列 Q に登録する。  
 Step 2 Q が空でない場合、次に分割する配線領域を Q から取り出す。Q が空の場合、終了。

- Step 3 カットラインの位置を決定する。  
 Step 4 カットラインにより対象とする配線領域を部分領域 R1 と R2 に分割する。  
 Step 5 カットラインを横切るネットを求める。  
 Step 6 ネットをカットライン上の各境界に割当てた場合のコストを計算する。  
 Step 7 線形割当てによりネットの通過するカットライン上の概略境界を決定する。  
 Step 8 ネットの割当てられた境界の両側に仮端子を置く。  
 Step 9 部分領域 R1 または R2 がカットラインにより分割可能である場合、Q に登録する。Step 2 へ。

### 2.2.2 カットライン位置の決定

この手法ではオーバーフローの発生を極力抑えるために、最も混雑度の高い位置からカットラインを引くように工夫している。これは領域の分割レベルが比較的浅いほうが混雑度を平均化するのが容易だからである。具体的には、カットラインの配線容量を  $C$ 、カットラインにより分割された部分領域の両方に端子（あるいは仮端子）を持つネットの総数を  $D$  としたとき、 $C - D$  が最小となるようなカットラインを選択している。

### 2.2.3 仮端子の境界への割当て

カットラインの両側に端子を持つ全ての配線の仮端子を適当な境界に線形割当てによって割当てる。線形割当てには、文献 [4] のアルゴリズムを用いている。いま、カットラインを通過するネットの集合を  $N = 1, 2, \dots, n$ 、カットライン上の境界の集合を  $S = 1, 2, \dots, m$  とし、境界  $j$  の配線容量を  $c_j$ 、ネット  $i$  を境界  $j$  に割当てたときのコストを  $a_{ij}$  とする。またネット  $i$  が境界  $j$  に割当てられていたら 1 そうでなければ 0 をとるような値を  $p_{ij}$  と表すことにすると、ここで言う線形割当ては以下のように定式化できる。

$$\text{minimize } \sum_{i=1}^n \sum_{j=1}^m a_{ij} p_{ij}$$

$$\text{subject to } \sum_{i=1}^n p_{ij} \leq c_j (j = 1, 2, \dots, m)$$

以下では、境界コスト  $a_{ij}$  の算出法について説明する。このアルゴリズムは、コストの定義によって配線結果が変わるという性質を持つが、ここでは総配線長の最小化を狙って、コストを設定している。まず、注目しているカットラインの片側の部分領域において、カットラインを横切る、あるネットの端子（あるいは仮端子）全てを包含する最小の矩形を考える。この矩形をカットライン上へ平行に写像した場合に矩形の内側に含まれる格子境界には最小コスト 0 を与え、外側にある境界には遠ざかるにつれて大きくなるようなコストを与える (図 2)。

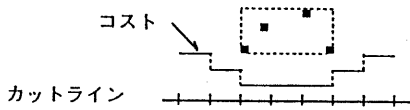


図 2: コスト関数

これは、矩形の外側の境界に仮端子が割当てられた場合には、迂回した径路をとることになり配線長が増加すると考えられるためである。また、矩形の外側の境界ではコストの増分を一定ではなく、カットラインまでの距離が小さいネットほどより大きいものとする。これはカットラインまでの距離が小さいネットが迂回した径路をとった場合、他の配線の障害となる可能性が高いからである。文献 [2] では、より厳密に、この矩形とコスト関数の生成をカットラインの両側で行い、算出された 2 つのコスト値を足し合わせて境界のコストとしている。

このようにコストが定められれば、線形割当てによってコスト最小になるような仮端子の位置が決定されることになる。

#### 2.2.4 混雑度の平均化

文献 [2] では、概略配線アルゴリズムにおける Step 8 と Step 9 の間に、混雑度の平均化のために仮端子の再割当てを行っている。線形割当てでは、等コストの解が存在する場合、どの割当てが選択されるかは不定である。そこで、コストが増加しない範囲では、割当ての結果で得られる配線長はほぼ同等であると考えられるため、まず、線形割当てによるコストの総和が最小となる割当てを行った後に、そのコストを増加させない範囲で仮端子の移動を行って、カットライン上の混雑度を平均化する。

以下に混雑度平均化アルゴリズムを示すが、仮端子の割当ては既に終了しているものとする。

また注目しているカットライン上で最も混んでいる境界の混雑度を MAXD、最も空いている境界の混雑度を MIND とする。割当てられたネットは待ち行列 Q に格納されている。

#### 混雑度平均化アルゴリズム

**Step 1** Q からネットを一本取り出す。もし Q が空なら終了。

**Step 2** 注目しているネットの仮端子 t をコストの増加なしに割当てられるような境界が他に存在すれば Step 3 へ存在しなければ Step 5 へ。

**Step 3** 仮端子 t を取り除いた境界の混雑度  $\geq$  MIND かつ仮端子を新たに割当てた境界の混雑度  $\leq$  MAXD ならばこの変更を実際に行い Step 4 へ。そうでなければ Step 2 へ。

**Step 4** MIND, MAXD を更新する。

**Step 5** Step 1 へ。

### 2.3 従来のコスト関数と配線形状

チャンネル配線手法に見られるように、多層配線においても縦配線と横配線の優先層を設けるのが配線率を上げるためには適当であると思われる。この場合、概略配線で求めた概略径路を縦配線の部分は縦の優先層に、横配線の部分は横の優先層に割当てて配線を行う。縦と横との接続する部分、つまり、曲がりの部分には必ずビアが生成される。したがって、各ネットはできるだけ縦横の部分集合、つまり曲がりが少ないことが望ましい。ところが、従来の方法では配線径路がジグザグになる傾向があり、配線パタンの複雑化や必要以上のビアの発生という欠点があった。

線形割当てに基づく概略配線に重要なものとして、コスト関数の設定方法がある。コスト関数の設定の仕方は配線結果に大きく影響を及ぼす。このコスト関数が配線形状も左右する。配線がジグザグになるのは、従来のコスト関数に原因がある。配線領域を分割していったときの、ある 2 端子ネットを考えたときのコスト関数およびその配線形状の例を図 3 に示す。このコスト関数はカットラインに対して両側のコスト関数をマージしたものである。

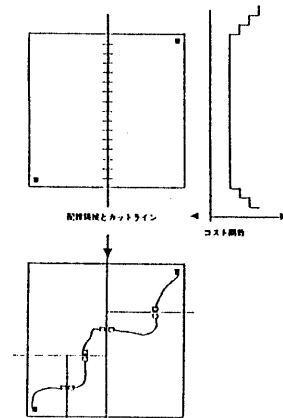


図 3: コスト関数と配線形状 (従来手法)

カットラインに対して、端子が含まれる矩形領域を写像したとき、その間の格子境界にネットを割当てたときのコストは等しくなる。それ以外の部分では、コストは矩形が写像された部分から離れるほど高くなる。従って、コスト最小化を目指す線形割当てで

は、結果として仮端子の割当ては矩形が写像された部分の格子境界のどれかに割当てられる可能性が高くなる。このとき、このコストが等しい各格子境界には、ネットが同じ確率で割当てられることになる。言い換えれば、この各格子境界に対して、あるネットを考えたとき、このネットはランダムに割当てられることになる。これが、領域を階層的に分割するたびに繰り返される結果、ジグザグになると考えられる。そのようなネットはクリティカルになる可能性が大きいにもかかわらず、端子が配線領域全体に散らばっている、いわゆる端子間距離の大きなネットほど、配線遅延の原因となるビアを生む曲がりの多い配線形状になる可能性が大きくなってしまふ。混雑度の平均化においては、この大きな自由度が良い結果を導くが、逆にビアの数が多いという点から見れば、詳細配線における混雑度を上げてしまうことになる。

この配線形状を改良する方法として、概略配線が終わった後に、ネットの径路を1つずつ修正する方法も考えられるが、逐次処理となるため、逐次処理特有の問題が発生するため、よい方法とはいえない。処理中、つまり、コスト関数の工夫によって配線形状を改良する方法を以下に述べる。

## 2.4 コスト関数の工夫による配線形状の改良

カットラインに対して端子が写像される位置の格子境界に端子が割当てられていくなれば配線は曲がりの少ない形状になるはずである。そこで、端子が写像される格子境界に対しては他よりもコストを低くする。端子を含む格子境界へ仮端子が割当てられれば、ネットはなるべく曲がらないような径路を取ることが考えられ、ネットの部分集合が少なくなることにより、ビアになる曲がりの数の削減が期待できる。この端子位置主導型のコスト関数を採用した場合の予想される配線形状を図4に示す。ここでのコスト関数も、前節と同様カットラインの両側のコスト関数をマージしたものである。

ここで、端子の割当てのコスト生成について詳しく述べる。カットラインに対する端子の写像を行う場合、実際の端子（実端子）と仮端子の写像がある。

実際の端子が写像される格子境界に対しては、端子の矩形領域よりもさらに低いコストを与える。この様子を図5に示す。

これに対して、仮端子の端子が写像される格子境界に対しては、端子の矩形領域よりもさらに低いコストにする場合と、矩形領域と同様のコストを与える2つの場合がある。現在のカットラインに対して平行した格子境界上に仮端子を持っている場合（図6）には、コスト関数の設定は実端子の場合と同様に、仮

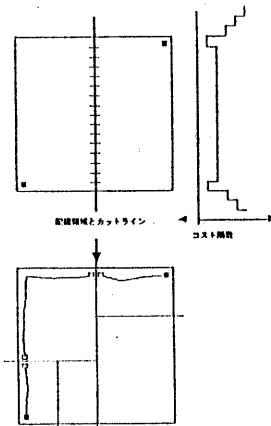


図4: コスト関数と配線形状（提案手法）

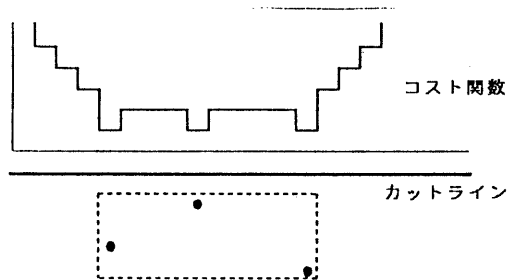


図5: コスト関数

端子が写像される格子境界に対して、矩形領域よりもさらに低いコストを与える。これは、この格子境界に割当てが行われれば曲がりの数を減らす可能性が高いからである。

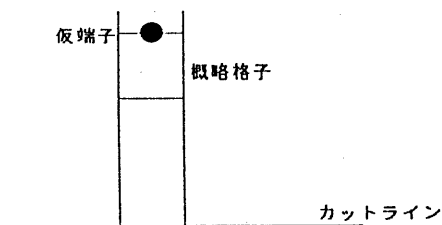


図6: 仮端子とカットライン（1）

現在のカットラインに対して垂直な格子境界上に仮端子を持っている場合（図7）には、仮端子が写像される矩形領域よりも低いコストは与えない。なぜなら、この格子境界の位置に割当てられたとしても曲がりの数を減らす可能性はないからである。

以上のようにコスト関数を設定することによって、曲がりの数が減ることになり、概略配線で発生するビ

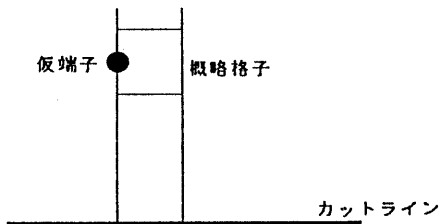


図 7: 仮端子とカットライン (2)

ア数の削減が期待できる。この従来のコスト関数による概略配線と提案するコスト関数による概略配線の実験結果を以下に述べる。

## 2.5 計算機実験

提案したコスト関数を採用した線形割当てに基づく概略配線手法を Sun Spark Station 2 (28.5MIPS) に C 言語で実装した。プログラムは、概略格子、配線層の数、端子の位置及びネットリストを入力とし、各格子境界に仮端子を割当ててるものである。入力データとしては、一様乱数によって発生させた 2 端子~6 端子の 300~3,000 ネットとした。概略格子の数は  $10 \times 10$  とした。これらのデータに対して従来手法 [2] と提案手法とで概略配線を行った結果を表 1 に示す。表中、総配線長とは端子が概略格子の中心にあるとみなして算出した仮想配線長の総和であり、ビア数とは概略配線径路の縦と横の部分集合の結合点の数のことである。

いずれのデータに対しても総配線長はほぼ同様かそれ以下、ビアの数では提案手法が従来手法の 6 割から 8 割弱に削減されていることが分かる。また、従来手法による配線形状の例を図 8 に示す。提案手法による配線形状の例を図 9 に示す。

この実験結果から言えることは、端子位置主導型のコスト関数を導入することにより、縦横の部分集合が減る効果もたらし、電気的には遅延の原因になり、またレイアウトの面では面積を多く使うビアの削減が可能になるということが言える。提案手法はコスト関数で配線形状の改良を図っているため、一括してネットを扱うので処理速度が非常に速い配線手法の特徴を損なうことなくビア数の削減に成功している。

提案手法は、ビアの数の削減を可能にし、電気的な特性を向上させる手法であると考えられ、平面配線が得意であるスケッチレイアウトシステムのための概略配線手法として組み込むことが考えられる。そのためには、各概略格子毎の平面配線解を求めるこ

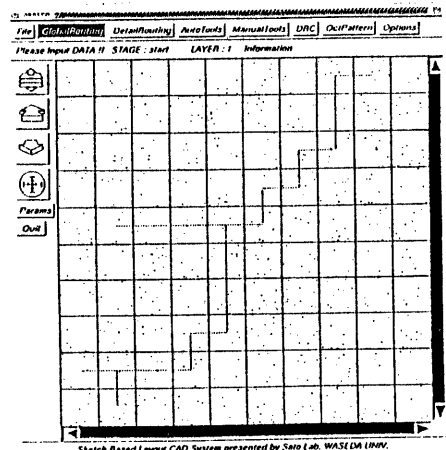


図 8: 従来手法による配線形状

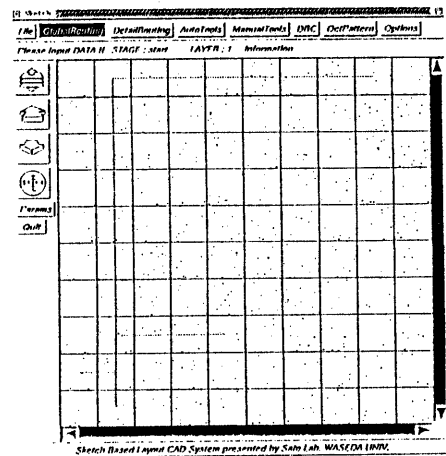


図 9: 提案手法による配線形状

とが必要となるため、この概略配線後に、層割当てと端子位置の決定処理を行うことで平面配線解を得る手法を考える。以下、層割当て手法と端子位置の決定手法について述べる。

## 3 層割当て

線形割当てによる概略配線では、平面に層情報を落として概略配線を行っている。つまり、各層毎の容量の和が格子境界上の容量になっている。本稿では縦横の優先層に概略径路を割当てていくことを前提としている。これは、各層毎の配線率を向上させるためであり、各概略格子における交差問題を回避するためである。層の数が 2 層であれば概略径路の割当ては自明であるが、3 層以上の場合には、層割当てが必要になる。

割当ての方法として、ネットを基準にして割当て

表 1: 実験結果

ネット数	端子数	総配線長		ビア数	
		従来手法	提案手法	従来手法	提案手法
300	2,3,4	1631750	1610000	1310	728
400	2,3,4,5	2510500	2460500	1861	1131
1000	2,3,4,5	5102000	5078000	2009	1587
1000	2,3,4,5,6	5158000	5133500	2026	1576
2000	2,3,4,5	10170500	10094500	4263	3160
2000	2,3,4,5,6	10293000	10217500	4258	3228
3000	2,3,4,5	15052000	14953000	5997	4637
3000	2,3,4,5,6	15117500	15117500	6132	4734

る方法と、ネットが分割された格子列を基準にして割当てる方法が考えられる。

### 3.1 ネットを基準にした層割当て

ネットを基準に割当てる方法では、ネットを順次取り出して、ネットを割当てるベア層を決める。ネットが割当てられるすべての境界において容量違反がなければ問題は無い。違反があればそのネットの部分集合を分割し、容量違反を起こさないように割当てる。このとき各境界において、各層の合計は容量以内に収まっているため、割当ては常に行うことができる。ただし、概略配線で求めたネットの部分集合がさらにいくつか分割される。

#### [層割当てアルゴリズム 1]

- Step 1 すべてのネットをキューに登録する。
- Step 2 キューからネットを1つ取り出す。ネットがなければ層割当て終了。
- Step 3 取り出したネットの概略経路を縦と横の部分集合に分割する。
- Step 4 ネットを割当てるベアの2層を決定する。違反がなければ Step 2 へ。違反があれば Step 5 へ。
- Step 5 ネットを分割し、違反がないように割当てる。Step 2 へ。

### 3.2 格子列を基準にした層割当て

ネットの部分集合を割当てる方法では、格子の列を考え、その格子間にまたがる部分集合を格子の列に沿って順次重ならないように求めていく方法（チャンネル配線でのレフトエッジ手法的な考え方）によって、各層の各格子境界においての容量違反が起こらないように割当てができる。部分配線がさらに細くなることは、先程とはかなり減少すると思われるが、ネッ

トそれぞれについて考えた場合には、同じネットが各層わたって部分集合を持つことになる。

#### [層割当てアルゴリズム 2]

- Step 1 未探索の格子列を1つ取り出す。格子列がなければ層割当て終了。
- Step 2 格子列に含まれるネットの部分集合をキューに登録する。
- Step 3 キューからネットを1つ取り出す。ネットがなければ Step 1 へ。
- Step 4 ネットの部分集合を割当てる層を決定する。違反がなければ Step 3 へ。違反があれば Step 5 へ。
- Step 5 そのネットを違反がないように分割し、割当てる。Step 3 へ。

## 4 端子位置の決定

層割当て後に、各層においての格子境界に割当てられている仮端子の相対順序と必要なビアの位置の決定をする必要がある。これらが具体的に決まっていじめて、各格子毎が平面配線可能な状態になり詳細配線ができる状態になる。

端子位置の決定の方法は次のようになる。配線領域は規則的な格子に分割されているものとし、x座標の方向の優先層に対しては、それと平行した格子の列をy座標の小さい順に取り出す。その格子の列に配線の部分集合を持つネットに対し、平面掃引法によってそれらの部分集合の径路とビアの位置決定と部分集合同士の交差によって生じるビアの作成と位置決定等を行う。この作業を層毎に順番に進めていく(図 10)。

格子の列を取り出したときの具体的な処理は以下のようなになる。

- ・端子のリストを作成する。

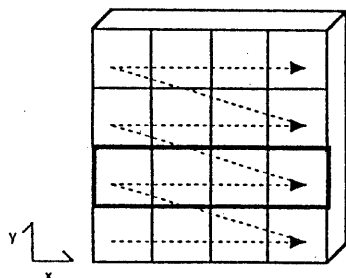


図 10: 格子列の取出し方

- ・ビアのリストへの追加をする。
- ・掃引をしながらネットの相対位置を決定する。
- ・具体的な座標値でのビア等の位置指定と経路の出力。

これらの内容について以下で述べる。

#### 4.1 端子のリストの作成

x方向の掃引を考えると、x座標の昇順に、同じx座標においてはy座標の昇順に、端子を対象にリストを作成する。このリストを基に、平面掃引を行い経路の決定がされることになる。

#### 4.2 ビアのリストへの追加

先程の端子のリストにビアのノードを加える。ビアの位置をなるべく端子の座標に近づけるように座標を取れば無駄な配線経路を生むことが少ないため、ビアの位置はなるべく端子の座標値に近いものを選択することにする。

ここでできたリストをもとに掃引を行う。なお、このリスト生成は[5][6]に従えば、 $O(n \log n)$ の手間で行うことができる。ここで、 $n$ は端子数である。

#### 4.3 掃引による端子位置の決定

端子、ビアのリストができたら、掃引リストを作り掃引を開始する。x座標に従って端子ビアリストのリストが現れたら、掃引リストとそのx座標に含まれる端子、あるいはビアとの比較を行う。比較の方法は、両者に共通のネットを探して、ビアと端子のネットであれば結線する。座標値を基に次の掃引リストへ引き込まれるネットの平均的な振り分けをそのx座標値で行う。このような処理をリストが終わるまで行う。

#### 4.4 配線解の出力

掃引をすべての格子列について行えば、お互いのネットの相対位置（仮端子位置）とビアの位置が特定される。

以上の処理により、各格子毎の平面配線可能な解が求まることになり、スケッチレイアウトシステムのための概略配線手法としてその機能が果たせることになる。

## 5 おわりに

本稿では、線形割当てに基づく概略配線手法に焦点をあてた。従来手法は、配線領域全体にわたり配線の混雑度を平均化でき、ネット全体を一括に処理ができるものの、配線混雑度の平均化ゆえ、概略配線の配線パターンが複雑化しビアの増加するという問題があった。この問題を解決するために、ビア削減が期待できる端子位置主導の新たなコスト関数を提案した。このコスト関数を導入することによって、大幅なビア削減が期待できることを実験によって確かめた。さらに、平面配線得意とする柔軟な配線設計が可能でトポロジカルなスケッチレイアウトシステムでの概略配線手法として本手法を適用するために、本手法の後に、層割当て手法と、その後の処理である端子位置の決定手法を行うことで平面配線解を得る方法を提案した。

今後の課題は層割当てと端子位置の決定を含んだ概略配線手法をシステムに統合し、様々なレイアウトに対して評価することである。

## 謝辞

本研究は、文部省科学研究費補助金：奨励研究(A)06855045(平成6年度)「柔軟性の高いLSIレイアウト設計手法に関する研究」の援助のもとに行われたものである。

## 参考文献

- [1] U.P. Lauther: "Top Down Hierarchical Global Routing for Channelless Gate Arrays Based on Linear Assignment," *Proc. VLSI 87*, pp.109-120 (1987).
- [2] 長谷川 他: "線形割当てに基づいた概略配線手法の実装と評価," *信学技法 VLD90-97*, pp.1-8 (1991).
- [3] 田中博, 金沢正博, 田中秀彦, 佐藤政生, 大附辰夫: "スケッチ表現に基づく多層配線システム," *情報処理学会研究報告 DA70-9*, pp.63-70 (1994).
- [4] R.E.Burkard and U.Derigs: "Assignment and Matching Problems: Solution Methods with Fortran Programs," Springer-Verlag (1980).
- [5] 粟島, 佐藤, 大附: "拡張平面掃引法によるコンパクト制約グラフ生成," *情報研報 DA58-1*, (1991).
- [6] 佐藤, 大附: "Enhanced Plane-Sweep Methods for LSI Pattern Design Problems," *信学技法 CAS86-199*, pp.87-94 (1986).