

## ブロック上通過配線処理手法

木下善彦 永原 出 石井 真  
ソニー株式会社 半導体事業本部

本論文では、階層設計時に問題となるチップ面積の増大を防ぐために、既設計の下位ブロック上の配線領域を有効利用し、チップ面積の縮小を図るブロック上通過配線処理手法について述べる。

本手法は、既設計の下位ブロックのレイアウト情報を抽出・抽象化して配線禁止領域とし、それを避けてブロック上を通過するという方法を採用している。さらに、配線領域を図形として捉え、計算幾何学的手法によるグリッドレスルータを用いて径路を求めている。したがって、階層構造を保持したまま、少ないデータ量・処理時間で、柔軟にブロック上の配線径路を求めることができる。

実際のレイアウトデータに適用した結果、ブラックボックス扱いに比べて、チップ面積が最高15%減少した。

### Over-Block Routing Method ( in Japanese )

Yoshihiko KINOSHITA, Izuru NAGAHARA, Makoto ISHII  
Semiconductor Division, SONY Corporation  
4-14-1, Asahi-cho, Atsugi-shi, Kanagawa-ken, 243 Japan

This paper presents over-block routing method for hierarchical layout to reduce chip size utilizing the routable area over the laid out blocks.

We extract the protection area from the laid out blocks and apply the gridless router based on computational geometry to find the routes over them, which works very efficiently in terms of time and space.

The experimental results show the effectiveness of this method. The reduction rate ranges up to 15% on chip size compared with the conventional "black-box" approach.

## 1. はじめに

近年の半導体プロセス技術の進歩によるVLSIの集積度の向上に伴ない、システム全体を1チップ上に載せる事が可能になってきている。またエレクトロニクス産業が成熟化とともに多様化してきており、ユーザの最終製品の差別化に対する要求が強まってきた。さらにゲートレイやスタンダードセルなどのセミカスタムLSIは、開発コストの低減、設計工数の削減、TATの短縮などを可能にし、特にビルディングブロック型スタンダードセル方式のLSIは、このような性質を保ちながらフルカスタムLSIに近い性能を発揮できる。一方ワークステーションの普及により、手軽にLSIを設計できるように環境が整備されてきた。こういった背景から最近、特定用途向けの専用LSI(ASIC)が注目され始めており、それを支えるCADシステムの開発が望まれている。

ところがシステム全体を1チップ上に載せるというような大規模なLSIをワークステーション上で設計する場合、データ量や処理時間が深刻な問題となる。この問題を解決するための手法として、階層設計の手法が必要となってくる。特に論理設計と比べて多大なデータ量、処理時間を必要とするレイアウト設計では、階層設計の手法がより重要となってきており、このような階層設計をサポートするレイアウトシステムも数多く報告されている(1) - (6)。

しかし階層レイアウトを行なう場合、一般に下位のブロックをブラックボックスとして扱うため、チップ面積の増大という問題が起こる。

そこで本論文では、この問題を解決するために下位ブロックのレイアウト情報を抽出・抽象化して配線禁止領域とし、それを避けてブロック上を配線する手法を提案する。これによって下位ブロックの配線領域を有効利用でき、チップ面積の縮小が図れる。さらに下位ブロックのレイアウト情報をその配線禁止領域として扱っているためにデータ量が抑制され、ワークステーション上での本格的な階層レイアウト設計が可能となる。

## 2. 階層設計における自動配線手法

### 2.1 階層設計におけるメリットと問題点

階層設計を行なう事によるメリットとしては次のようなものが挙げられる。まず、データ量・処理時間が削減される事であり、また各ブロックごとに並行してレイアウトを行なえる事(7)、既設計されたブロックのレイアウト情報を別のレイアウトの時に再利用できる事(7)、などである。

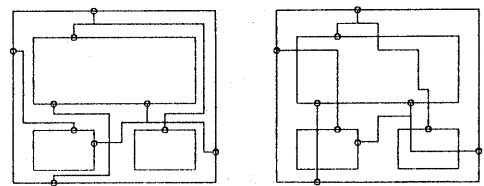
これに対して問題となる点は、特に顕著な例としてチップ面積の増大という事が挙げられる。一般に階層設計を行なう場合、下位のブロックをブラックボックスとして扱うために、それを挟んで両側にある端子間を結ぶ要求があった場合には配線がその周囲を迂回してしまい、チップ面積の増加をもたらす(図1(a))。

### 2.2 階層設計における自動配線手法の分類

前述の問題点を解決するためには下位ブロックの上を

配線が通過する必要がある(図1(b))。これに関してさまざまな方法が提案されており、これを分類すると次のようになる。

- 1) レイアウトの段階で全階層を展開して、チップレベルでの一括配線を行なう(1)。
- 2) 各ブロックごとにレイアウトを行ない、その上位ブロックで下位ブロックのレイアウト情報(セル配置や配線)を調べながら、ブロック内の配線領域を利用して配線を行なう(2)。
- 3) フロアプランの段階で概略配線を行ない、下位ブロック上を通過するネットに対してそのブロックの外周上に端子を作成し、下位ブロック内のレイアウト時にその詳細な経路を求める事によって、配線をブロックの中に取り込む(3)。



(a)ブラックボックス扱い (b)ブロック上を通過

図1. 階層設計時の問題点とその解決策

1), 2) はともにデータ量・処理時間が膨大となり効率的でない。また1) では論理設計時の階層構造を壊すことになり、前述の階層設計のメリット(並行レイアウト、既設計ブロックの再利用)を活かす事ができない。3) ではトップダウンにブロックの外周上に端子位置を決定するので、ブロック内のレイアウトが終わってからブロックの配置を変えたりするような場合に柔軟に対応できない。

このように、今までに提案された方法には上記のような問題がある。そこで本論文では

- a) 階層構造の保持
- b) データ量の削減と処理時間の短縮
- c) 配線経路の柔軟性

の3つの条件を満たすブロック上通過配線について提案する。

### 2.3 ブロック上通過配線の新手法

#### 2.3.1 概要

前節の3つの条件を満たすために本論文で提案するブロック上通過配線は、既設計の下位ブロックのレイアウト情報を配線禁止領域として抽出・抽象化し、それを避けながら下位ブロック上を配線が通過するというものである。本手法を用いる時の基本的な階層レイアウトの流れはトップダウンのチッププランとボトムアップの詳細レイアウトである(図2)。下位ブロックのレイアウトが終了したら配線禁止領域抽出プログラム(8)を実行する。上位ブロックでのレイアウトの時には下位ブロックのレイアウト情報としてこの配線禁止領域を使用し、下位ブロック上の配線領域を利用して配線を行なう。

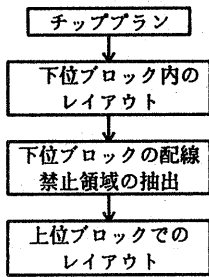


図2. 本手法を用いた階層レイアウトの流れ

### 2.3.2 特徴

本手法では上記のようなレイアウトの流れを採る事により、論理設計時の階層構造を保持したままレイアウトを行う事ができる。また、上位ブロックのレイアウトを行なう時には下位ブロックのレイアウトが終了し大きさ・形状が決定されているため、柔軟に配線経路を求める事ができる。

そして、下位ブロックのレイアウト情報としてそれが抽象化された配線禁止領域を用いているので、約20分の1のデータ量で済む<sup>(9)</sup>。

さらに、配線領域を図形として捉え、計算幾何学的手法によるグリッドレスルータを用いて経路を求めているので、プログラム実行時に使用される内部メモリも少なく、処理時間も迷路法などの格子展開法に比べて短い。

## 3. ブロック上通過配線

ここでは、まず本手法を一機能として含む自動配線プログラムが搭載されているレイアウトシステムの概要について述べる。次に、本手法が配線経路探索手法として用いている計算幾何学的手法によるグリッドレスルータについて述べ、そして本手法の処理手順について述べる。

### 3.1 レイアウトシステムの概要

本手法を一機能として含む自動配線プログラム<sup>(8)</sup>は、階層設計用スタンダードセルLSI統合レイアウトシステム ASTRO FLEX<sup>(6)</sup>に搭載されている。

#### 3.1.1 システム構成と特徴

ASTRO FLEXのシステム構成を図3に示す。本システムは一元的なD/Bを核とし、チッププラン、自動配置、自動配線の他、レイアウト修正、検証、そしてシミュレータ・インタフェースなど数多くのアプリケーションを備えている統合レイアウトシステムである。

#### 3.1.2 レイアウトモデル

チップはI/Oセルと内部のファンクションブロック(以下FB)及びセルから構成されている(図4)。

FBとは、論理的に一つの機能を実現しているブロックで、その外部とのインタフェースは明確に定義されて

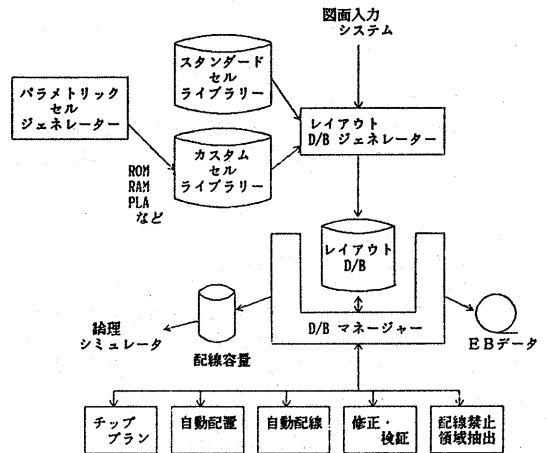


図3. システム構成図

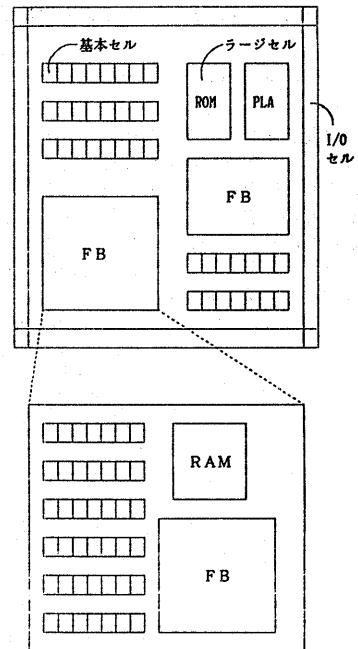


図4. レイアウトモデル

いる。これが一つの階層を成している。

セルは、一列状に並べられる基本セルと、ランダムに配置できるラージセルに分けられる。ROM, RAM, PLAなどがこのラージセルに相当する。

これらのFB・セルは、上位レベルのFB又はチップ内の任意の位置に配置できる。

また配線はポリ一層、アルミ二層の三層配線であり、水平方向を1stA1(1A1)、垂直方向をポリと2ndA1(2A1)を用いる。

### 3.1.3 自動配線プログラム

本手法を一機能として含む自動配線プログラムの処理手順は図5のようになっている。まず与えられたセル及びFBの配置に対し、チャンネルグラフを作成する。次にこのグラフ上でグローバル配線を行ない、概略的な配線経路を求める。そしてそれに基づきブロック上通過配線及び詳細配線を行なう。

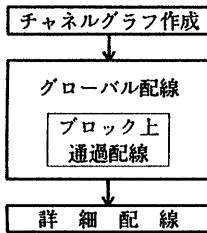


図5. 自動配線プログラムの処理手順

次に、各処理の内容について述べる。

#### (1) チャンネルグラフ

自動配線プログラムで用いるチャンネルグラフを図6に示す。各配線領域の交点、各FB・ラージセルの中心及びその上下左右、そして各セル列の上下にその幅から決められた数の節点を作成し、隣接する節点間に枝を設ける。各枝にはその長さを重みとして与え、またセル列上枝にはその枝を通過できる配線本数(容量)を与えておく。

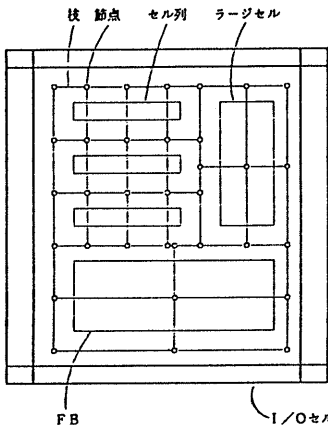


図6. 自動配線プログラムにおけるチャンネルグラフ

#### (2) グローバル配線

グローバル配線では図6のチャンネルグラフ上で概略的な配線経路を求める。

#### (3) ブロック上通過配線

グローバル配線の結果、FB・ラージセル上の枝を通過するネットに対してその詳細な経路を二層(1A1と2A1)配線を用いて求める。

#### (4) 詳細配線

グローバル配線、ブロック上通過配線によって求めた各セル列上及びFB・ラージセル上の端子位置を基にして詳細な経路をチャンネル配線によって求める。

## 3.2 配線容量を考慮した計算幾何学的グリッドレスルート

本手法では、配線経路探索手法として計算幾何学的手法によるグリッドレスルートを用いている。このルートは、線分探索法のように使用メモリが少なく、また処理時間が短く、さらに迷路法のように高い経路発見能力がある。

従来の計算幾何学的グリッドレスルート<sup>(10)</sup>では、1層目と2層目の配線領域の重なりを調べてグラフを作成し、そのグラフ上で経路を求め、その後配線領域の更新を行ない、再び経路を求めるという処理を行なっている。

本手法では、グラフの作成方法を変え、節点と枝に配線容量を与える事によって配線領域の更新といった操作をなくし、配線領域内の混雑度の均一化を図れるようにした。

### 3.2.1 基本アルゴリズム

本ルートの基本的なアルゴリズムを示すために図7(a)のような配線領域を考える。配線は水平方向を1A1、垂直方向を2A1を用い、あらかじめ1A1の配線領域が水平に、1A1の配線禁止領域が垂直に、2A1の配線領域が垂直にそれぞれ矩形分割されているとする。

〔性質1〕1A1の矩形配線領域のうち隣接するもの(図7(c))に着目する。もし、この2つの矩形 $R_{1a}$ 、 $R_{1b}$ のy座標の最小の辺から最大の辺までをカバーする2A1の矩形配線領域 $R_{2A}$ が存在する時、 $R_{1a}$ と $R_{1b}$ の境界線のうち $R_{2A}$ に含まれている部分 $L_{as}$ には2A1が垂直に通る事ができる。したがって、 $R_{1a}$ と $R_{1b}$ のうち $L_{as}$ の範囲にある部分 $R'_{1a}$ 、 $R'_{1b}$ からは、どの部分からもお互いに他方へ2A1を用いて配線する事ができ、その本数 $C_1$ は、

$$C_1 = L_{as} / 2A1 - 2A1 \text{の配線間隔}$$

〔性質2〕1A1の矩形配線禁止領域 $\bar{R}_1$ をまたがって隣接する1A1の矩形配線領域(図7(d))に着目する。もし、この2つの矩形 $R_{1c}$ 、 $R_{1a}$ のy座標の最小の辺から最大の辺までをカバーする2A1の矩形配線領域 $R_{2B}$ が存在する時、 $\bar{R}_1$ のうち $R_{2B}$ に含まれている部分 $\bar{R}'_1$ には2A1が垂直に通る事ができる。したがって、 $\bar{R}'_1$ の幅を $L_{ca}$ とすると、 $R_{1c}$ と $R_{1a}$ のうち $L_{ca}$ の範囲にある部分 $R'_{1c}$ 、 $R'_{1a}$ からは、どの部分からもお互いに他方へ2A1を用いて配線する事ができ、その本数 $C_2$ は、

$$C_2 = L_{ca} / 2A1 - 2A1 \text{の配線間隔}$$

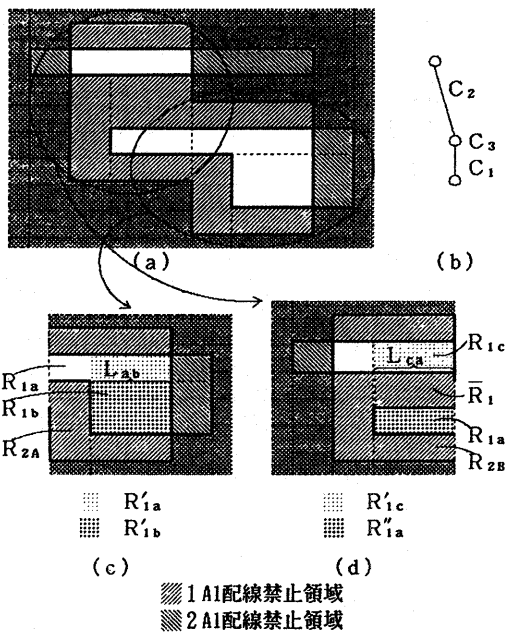


図7. 配線領域の例(1)

【性質3】1AIの矩形配線領域 $R_{1a}$ 内では1AIが水平方向に自由に通ることができる。そしてその本数 $C_3$ は、  

$$C_3 = R_{1a} \text{の高さ} / 1AI-1AI \text{の配線間隔}$$

性質1~3より、 $R_{1c}$ から $R_{1b}$ へは  $\min(C_1, C_3, C_2)$ 本の配線が通れることになる。さらに、図8のように $L_{ab}$ と $L_{ca}$ が共有部分 $L_{abc}$ を持つ時には、その部分では1AI配線を使わずに2AIのまま直進する事ができ、その本数 $\chi$ は、  

$$\chi = L_{abc} / 2AI-2AI \text{の配線間隔}$$
  
 である。したがってこの場合、 $R_{1c}$ から $R_{1b}$ へは  $\chi + \min(C_1 - \chi, C_3, C_2 - \chi)$ 本の配線が通れることになる。

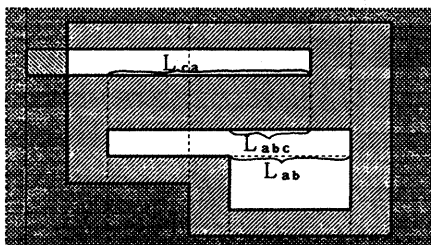


図8. 配線領域の例(2)

上記の性質を利用して効率よく配線径路を求めるために、1AIの矩形配線領域を節点に対応させ、性質1, 2を満たすならば対応する節点間に枝を設ける。さらに、上記の配線本数 $C_1, C_2$ を枝、 $C_3$ を節点の配線容量として与える。そして、各節点には水平方向に1AIが通れる $y$ 座標を与え、各枝には垂直方向に2AIが通れる $x$ 座標を与える。こうして得られたグラフ(図7(b))上で、最短経路アルゴリズムを適用する事により2つの節点間の概略な配線径路が求まり、節点と枝に与えられた1AIと2AIの通れる座標を基にして詳細な径路が求まる。

### 3.2.2 ラベル付けと逆追跡

最短経路アルゴリズムにおけるラベル付けと逆追跡の操作について詳しく述べる。  
 いま、図9のような配線領域を考え、節点 $N_1$ から次の節点 $N_2$ にラベルを付けるとする。もしその間の枝 $E_1$ の容量が0ならば、その節点 $N_2$ にはラベルを付けない。また、節点 $N_2$ にラベルが付いた場合には同時にその間の枝 $E_1$ に与えられた2AI通過可能位置の中から1つを選んでおく。そしてその節点 $N_2$ からさらに次の節点 $N_3$ へのラベル付けを行なう時に、その間の枝 $E_2$ に与えられた2AIの通過可能位置の中に前の枝 $E_1$ の時に選ばれた2AI通過可能位置と同じ $x$ 座標をもつものがあればそれを採用し、なければ $N_2$ の容量が0の時は $N_3$ にラベルを付けず、それ以外の時は任意の2AI通過可能位置を選ぶ。径路が発見されたら逆追跡を行なうが、この時に各枝で選ばれた2AI通過可能位置を調べながら径路として採用し、枝の容量を1つずつ減らしていく。そして連続する2つの2AI通過位置が同じ $x$ 座標ならばその間の節点の容量は変えず、異なれば節点の容量を1つ減らし、節点に与えられた1AI通過可能位置の中から1つを選び、径路として採用する。このような操作により不必要な折れ曲がりを減らす事ができ、通過する配線本数を増やす事ができる。

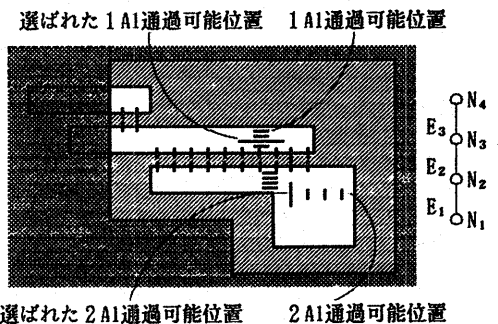


図9. 配線領域の例(3)

### 3.3 ブロック上通過配線の処理手順

グローバル配線の結果、FB・ラージセル(以下、単にブロック)上の枝を通過するネットに対してその詳細な径路を二層(1AIと2AI)配線を用いて求める。ブロック上通過配線は各ブロックごとに処理され、そ

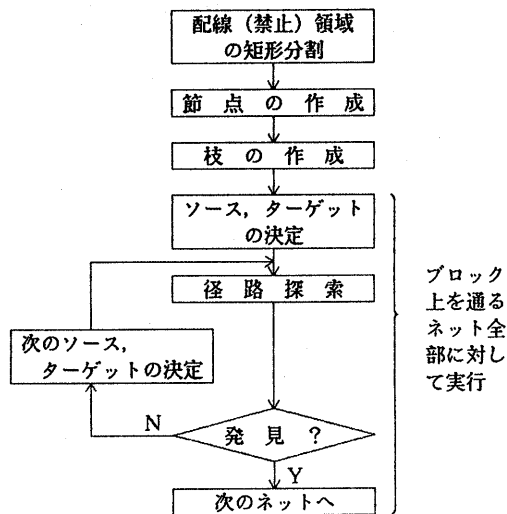


図10. ブロック上通過配線の処理手順

の処理手順は図10のようにになっている。各処理の内容は次の通りである。

i) 配線(禁止)領域の矩形分割

1 A1の配線領域を水平に、1 A1の配線禁止領域を垂直に、2 A1の配線領域を垂直にそれぞれ矩形分割する(図11(a),(b))。これらは、plane sweep法によって処理される。

ii) 節点の作成

1 A1の矩形配線領域を節点に対応させる。また、2 A1の矩形配線領域のうちでブロックの上下辺に接するものがあるならば、その辺上に節点を作成し、1 A1の矩形配線領域のうちでブロックの左右辺に接するものがあるならば、その辺上に節点を作成する(図11(d))。

そして、上下辺上を除くすべての節点にそこを通過できる1 A1の配線本数を容量として与え、その通過可能位置を与えておく。

iii) 枝の作成

隣接する1 A1の矩形配線領域、あるいは矩形配線禁止領域をまたがって隣接する1 A1の矩形配線領域に対して、それらをカバーしている2 A1の矩形配線領域があるならば、対応する2つの節点間に枝を作成する。またブロックの上下辺上に作成された節点に対しても、それに対応する仮想的な1 A1の矩形配線領域を考え、上記の方法で枝を作成する。さらに、左右辺上に作成された節点に対しては、それを作成する原因となった1 A1の矩形配線領域に対応する節点との間に枝を作成する(図11(d))。

そして、左右辺上の節点との間に作成された枝を除くすべての枝に、そこを通過できる2 A1の配線本数を容量として与え、その通過可能位置を与えておく。

iv) ソース・ターゲットの決定

グローバル配線でブロック上を通過するネット及びその通過辺が決定されているので、そのうちの任意のネットを選び、径路探索のためのソース・ターゲットの節点を決定する。

グローバル配線の結果、図12のような結果が得られたとすると、上辺の一番右にある節点と下辺の一番左にある節点をソース及びターゲットとする。

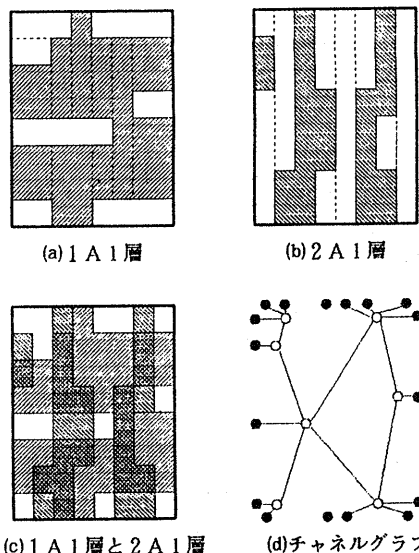
v) 径路探索

チャンネルグラフ(図11(d))上で、ソース、ターゲット間の概略な配線径路を最短径路アルゴリズムにより求める。

径路が発見できたならば、枝に与えられた2 A1の通過可能位置及び必要ならば節点に与えられた1 A1の通過可能位置により、詳細な配線径路を求める。端子の位置は、上下辺の節点に対してはその接続する枝に与えられた2 A1の通過可能位置、左右辺の節点に対してはその節点に与えられた1 A1の通過可能位置の中から選ぶ。

径路が発見できない時には、別のソース、ターゲットを選び、再び径路探索を行なう。

尚、ブロック上の径路が発見できなかったネットに対しては、ブロックをブラックボックス扱いにしてグローバル配線をやり直す。



○ 配線領域節点  
● ブロック周囲辺上節点

図11. ブロック上通過配線におけるチャンネルグラフ

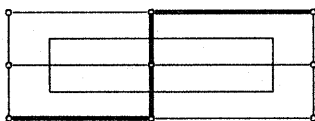


図12. グローバル配線の結果の例

#### 4. 適用例

本手法を実際のレイアウトデータに適用した結果を表1に示す。また、プロット図を図13に示す。

ブラックボックス扱いにした時に比べて、チップ面積は3~15%減、配線長は13~27%減、処理時間は最悪でも2倍に収まっている。また、ブロック上通過要求のあったネットのうち、実際にその径路が求まったのは約65%である。

この結果は、本手法の有効性を十分に示すものである。特に、処理時間の速さと径路発見能力の高さは、十分実用に供するものであると思われる。

#### 5. おわりに

既設計ブロックの配線領域を有効利用し、チップ面積の縮小を図るブロック上通過配線について述べた。

本手法は、階層構造を保持したまま、少ないデータ量・処理時間で、柔軟に配線径路を求める事ができ、また実際のレイアウトデータに適用した結果、その有効性が確認できた。現在、本手法を実際のLSI設計に適用し始めている。

今後の課題としては、ブロック上の配線径路探索時の混雑度の考慮、ブロック周囲上の端子位置の決定方法などが残されている。

#### 謝 辞

日頃御指導頂く井上隆秀設計技術部長に感謝いたします。また本論文をまとめるに当たり貴重な御助言を頂いた小山正弘氏をはじめ、レイアウト関係の方々に感謝いたします。さらに、本手法の開発に当たり御討論頂いた当社情報処理研究所隅田哲夫氏、並びに本論文作成に御協力頂いた小清水伸子氏に感謝いたします。

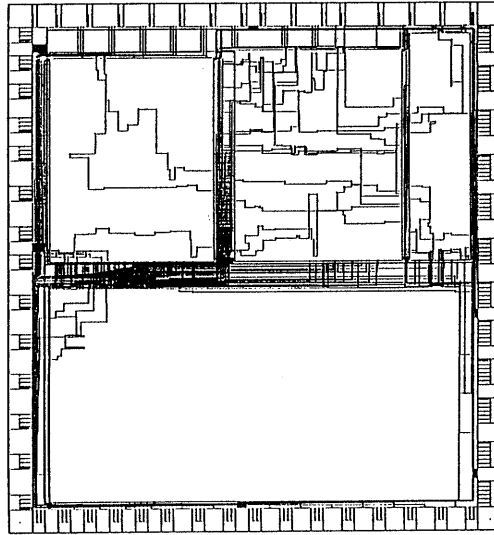
#### 【参考文献】

- (1) 神戸, 藤原, 岡田, 小嶋, 西岡, “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SHARPS IIについて”, 信学技報, CA S 85-141.
- (2) 寺井, 小澤, 坂田谷, 湯山, “大規模フルカスタムLSIの自動レイアウト設計方式”, 情報処理学会論文誌, Vol. 26, No. 1, pp. 76-84.
- (3) 三橋, 山田, 樋渡, 村方, 田中, 南, “カスタムVLSIのための統合レイアウトシステム”, 信学技報, CA S 85-140.
- (4) 北沢, 安達, 上田, “大規模高密度レイアウトプログラム: ALPHA-II”, 設計自動化, 19-4.

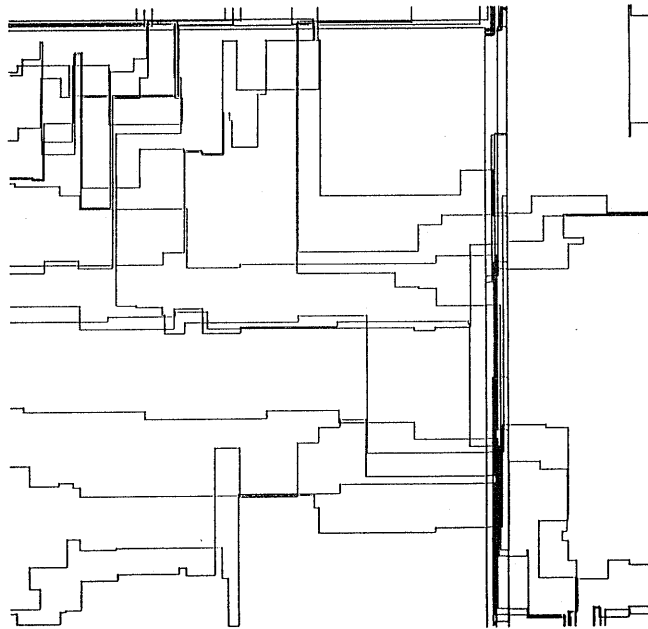
- (5) 荘司, 諸沢, 安藤, 大津, “VLSIレイアウトシステム(VILLA)の構成”, 第29回情処全大, p. 1671-1672.
- (6) 佐藤, 木下, 上高原, 永原, 石井, 小山, “スタンダードセルLSI統合レイアウトシステム(1) —概要—”, 第33回情処全大, pp. 2277-2278.
- (7) 上高原, 木下, 佐藤, 永原, 石井, 小山, “スタンダードセルLSI統合レイアウトシステム(2) —階層設計手法—”, 第33回情処全大, pp. 2279-2280.
- (8) 木下, 永原, 石井, “スタンダードセルLSI統合レイアウトシステム(4) —自動配線手法—”, 第33回情処全大, pp. 2283-2284.
- (9) 永原, “スタンダードセルLSI統合レイアウトシステム(5) —配線禁止領域自動抽出—”, 第33回情処全大, pp. 2285-2286.
- (10) 佐藤, 大附, “グリッドレス・ルータ —格子を用いない二層配線径路探索手法—”, 電子通信学会論文誌(D), Vol. J69-DNo. 5, pp. 802-809.

表1. 本手法の適用例

データ	ゲート数	ネット数	チップサイズ (mm <sup>2</sup> )			総配線長 (mm)			経路発見能力			処理時間 (μVAX II)	
			ブラックボックス扱い	ブロック上通過配線	減少率	ブラックボックス扱い	ブロック上通過配線	減少率	ブロック上通過要求ネット数	ブロック上通過ネット数	発見率	ブラックボックス扱い	ブロック上通過配線
			A	4900	180	7,800×8,130	7,460×7,940	6.59%	610.65	500.33	18.1%	151	76
B	1600	524	2,384×2,820	2,080×2,764	14.50%	389.66	284.0	26.9%	47	45	95.7%	28分02秒	29分53秒
C	2600	120	3,314×5,080	3,202×5,090	3.19%	183.87	160.57	12.7%	38	34	89.5%	5分42秒	12分42秒



(a)チップ



(b)拡大図

図13. プロット図 (表1のデータA)