

矩形探索迷路法による高速配線手法

- (1) 伊藤卓司 佐藤康夫 池本康博 (2) 垣木信彦 (3) 南 英一
(4) 中村武典 田中 博 (5) 山田政浩 (6) 上矢 誠

- (1) 日立製作所デバイス開発センター
(2) 日立製作所中央研究所
(3) 日立製作所神奈川工場
(4) 日立コンピュータエンジニアリング
(5) 日立コンピュータエレクトロニクス
(6) 三倉エンジニアリング

配線可能領域を矩形単位に探索することによる、高速で汎用的な自動配線アルゴリズムを新たに提案する。配線経路を探索する際、その処理単位を領域とし、領域間経路を求め、その後領域内部において詳細な配線経路を決定する2段階配線手法を持つ。処理単位が領域であるため、一度に広い範囲が探索でき高速な処理が可能となる。また、探索時に幅の概念を持つため、ダイレイを考慮した配線等に有効であると考えられる。本手法を用い評価した結果、従来の迷路法を用いた処理に比べ、約25倍の処理速度を実現し、その高速性を確認した。

C. I. M. A. : Channel Intersection Maze Router

- (1) Takuji Itoh Yasuo Satoh Yasuhiro Ikemoto (2) Nobuhiko Kakigi (3) Eiichi Minami
(4) Takenori Nakamura Hiroshi Tanaka (5) Masahiro Yamada (6) Makoto Kamiya

- (1) Hitachi, Ltd. Device Development Centre
(2) Hitachi, Ltd. Central Research Laboratory
(3) Hitachi, Ltd. Kanagawa Works
(4) Hitachi Computer Engineering Ltd.
(5) Hitachi Computer Electronics Ltd.
(6) Mikura Engineering Co., Ltd.

The fast, general routing algorithm is proposed. This algorithm finds a route using a rectangle units which are the part of the routing area. This algorithm consists of two stages. First, it divides a routing area into rectangles, find the route through these rectangles. Second, determine the detailed route within the above connected rectangles. The fast search is accomplished because a large area enclosed by a rectangle can be searched at a time. The algorithm is implemented, and experimented. From the result, we could conclude that we had a 25 times of speed up from the ordinary maze router.

1. はじめに

VLSIにおける配線問題は、その質的変革とともに手法の見直しが必要となっている。すなわち、従来は指定されたピン間を出来るだけ短い配線長で結ぶことが必要で、技術的には未配線本数の低減、計算機処理時間の低減が主要課題であった。そこでアルゴリズムとしては、チャンネル割当法の様に高速でチップ全体の混雑度制御の容易な方法が主流となっていた。

しかし、VLSIの大規模化、高速化とともに、

(1) 多層配線化

3～5層以上の配線手法が必要である。

また、これに伴い半導体プロセス上のルール（コンタクト隣接制約等）も複雑化している。

(2) 敷詰め化、端子散在化

(1)により、セル間の隙間が無くても配線可能となり、チップ上にセル列間のチャンネル領域なるものが存在しないものもある。

(3) マクロセル多用化

均一なスタンダードセルだけでなく、大きさの様々なセルが配置されたモデルを扱う必要がある。

(4) デイレイ、特性考慮配線の必要化

ネットの長さを指定した配線、
配線幅を通常より太くした配線、
配線層を指定した配線、
バス状の配線、

差動信号のペア配線等、

これらは、一本あるいは複数本のネットの形状を制御する必要がある。

これらに対し、迷路法やラインサーチ法の様なネットあるいはピンペア単位に配線するアルゴリズムは、比較的上記の課題に対し対応しやすい性質を持っているが、処理時間の点で計算機が高速化されてきたとはいえ、まだまだ問題が多く、また(4)の処理課題に対しては、アルゴリズム本来の性質として実現できるものは少なく、特殊な専用処理を施し

て工夫する必要がある。

本稿では、矩形探索迷路法（C.I.M.A.法：CHA—NNEL INTERSECTION MAZE ROUTER）と呼ぶ、新配線手法を提案する。これは、迷路法が点、ラインサーチ法が線単位に経路探索を行なうのに対し、面で経路探索を行なうため、従来の同種の手法より高速であるとともに、探索時の経路幅の概念を持つため、(4)で述べた課題をアルゴリズムそのものの性質として扱える特徴を持っている。面に着目した配線としては、文献(1)等があり、特定モデルの最適解を与えて優れた論文であるが、本稿とは扱う問題が異なるため、本稿で提案している経路幅や2段階配線の概念は無い。

以下、矩形探索迷路法の詳細、及び評価結果について報告する。

2. レイアウトモデル

本手法が想定している、レイアウトモデルを以下に示す。これらのモデルは、現実の配線プロセスや設計ルールを充分カバーする必要がある。

2.1 多層配線モデル

多層配線モデルを以下のように定義する。x方向の格子座標系に対し各配線層は $aK+b$ 、 $cL+d$ 、 $eM+f$ 、 $gN+h$ 、…（ K 、 L 、 M 、 $N=1, 2, \dots$ ）の様に、等間隔で配線する。配線方向は主として、奇数層が横、偶数層が縦という具合に交互に変わる。

配線幅が太くなる場合や、グリッドフリーで配線する場合は上式の様に表現できなくなるが、ここでは簡単のため省略する。

2.2 隣接制約ルール

(1) コンタクト制約ルール

各層間を接続するコンタクトを図2.1の様に CT_1 、 CT_2 、 CT_3 …と呼ぶ。隣接ルールとしては、 CT_i と CT_j （ $i, j=1, 2, 3, \dots$ ）の間の隣接禁止条件が与えられる。なお、この条件は異なる信号線間の場合と、同一信号線内の場合で

一般に異なった制限となることに注意する必要がある。

離れた配線層間を一気に結ぶ貫通コンタクトは、プリント基板の世界では用いられるが、LSIを対象とする本稿ではとりあえず考えない。

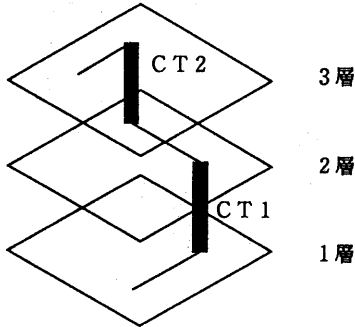


図2.1 コンタクト

(2) 配線隣接ルール

各層の配線パターンに対し、図2.2の様に同層ボタン(コンタクトも含む)の隣接禁止条件が与えられる。この条件を考慮することにより、配線幅が太い配線や、グリッドフリー配線に容易に対応可能となる。



図2.2 配線隣接制約

2.3 禁止領域モデル

チップ上、配線層毎に矩形の集合として禁止領域が定義されている。形状に関する制約は一切無い。

3. 矩形探索迷路法(C.I.M.A.法)のアルゴリズム

図3.1に全体の処理フローを示す。全体は大きく分けて、配線可能未使用領域(=全体領域-禁止領域)を、配線矩形の直和に分割しデータ構造を作成する処理(ステップ1)、配線矩形間の交差を求

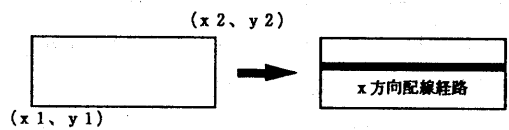
めながら、配線矩形間経路を求める処理(ステップ2)、求めた配線矩形間経路の内部で最適詳細経路を決定する処理(ステップ3)、求めた経路を配線禁止領域とし、データ構造の中にフィードバックする処理(ステップ4)の、4つの部分に分れる。

以下、各処理について詳細に説明する。

3.1 配線矩形定義

図3.2は配線矩形の定義を示す。配線矩形 $L(x_1, y_1, x_2, y_2, \lambda)$ とは、対角頂点 (x_1, y_1) 、 (x_2, y_2) を持つ矩形で配線層 λ に属する。配線層 λ が例えば横方向の層の場合、配線矩形 L の中(境界を含む)で横方向の配線を任意位置に引くことが出来る。また、図3.3に示すように横型の配線矩形 $M(x_1, y_1, x_2, y_2, \lambda_1)$ と、縦型の配線矩形 $N(x_1, y_1, x_2, y_2, \lambda_2)$ が交差し、 λ_1 と λ_2 が隣り合う層、すなわち $|\lambda_1 - \lambda_2| = 1$ の場合、交差部領域内の任意の位置にコンタクトを作成可能で、図3.3の様に λ_1 層と λ_2 層を接続する配線を引くことが出来る。交差条件としては、図3.4の様に部分的な交差、あるいは片方の配線矩形が他方の配線矩形に完全に含まれる場合もある。

(1) 横型配線矩形 $M(x_1, y_1, x_2, y_2, \lambda_1)$



(2) 縦型配線矩形 $N(x_1, y_1, x_2, y_2, \lambda_2)$

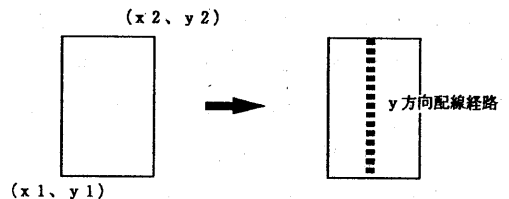


図3.2 配線矩形

3.2 配線矩形作成

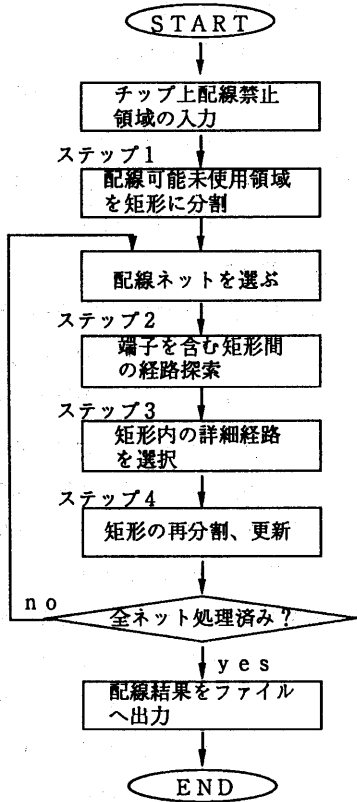


図3.1 矩形探索迷路法アルゴリズムフロー

配線層毎に、配線矩形を作成する。図3.5は禁止領域を除いた部分を配線矩形に分ける方法を示す。横方向の配線層の場合は横に、縦方向の配線層の場合は縦に分割する。このようにして作成された配線矩形は、3.1の定義を満たすことは明らかである。横方向の配線矩形の集合をH、縦方向の配線矩形の集合をVと定義する。

配線矩形のデータ量は図3.5から明らかのように、 $(4 \times \text{禁止領域数})$ 以下となるが、実験の結果禁止領域数以下となることがわかっている。

3.3 配線矩形間経路探索

対象ネットに対して以下の処理を行なう。

- (1) 始点Sの座標を (x_0, y_0) 、層を α_0 とする。始点S自体は禁止領域として登録されているため、Sに隣接する配線矩形を探す。

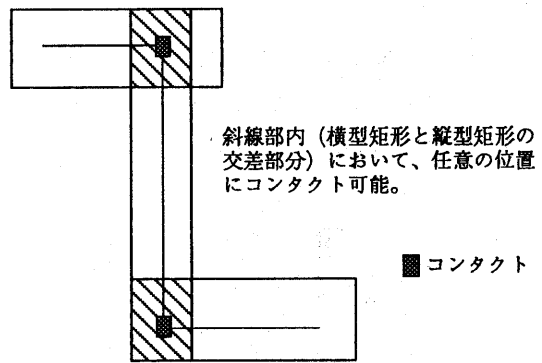


図3.3 矩形交差

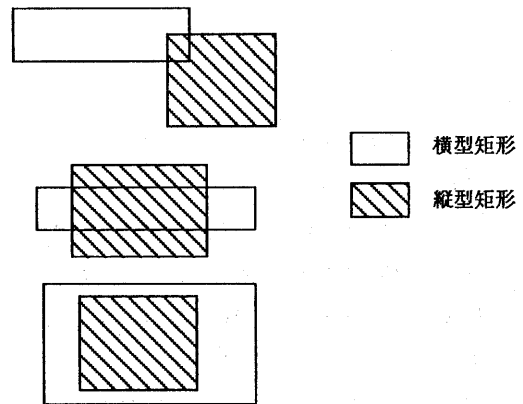
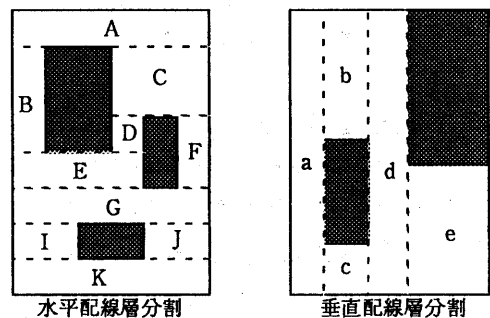


図3.4 矩形の交差の模様



横型矩形H = {A, B, C, ...}

縦型矩形V = {a, b, c, ...}

(■ : 配線禁止領域、点線 : 矩形分割線)

図3.5 矩形分割

隣接するとは、配線矩形 $L(x_1, y_1, x_2, y_2, \ell)$ が、

a: コンタクトを介す場合

$$x_1 \leq x_0 \leq x_2, y_1 \leq y_0 \leq y_2, \\ |\ell - \ell_0| = 1 \text{ なる場合}$$

b: コンタクトを介さない場合

$$\ell = \ell_0 \text{ で、}$$

$$\cdot x_0 = x_1 - 1, y_1 \leq y_0 \leq y_2$$

$$\cdot x_0 = x_2 + 1, y_1 \leq y_0 \leq y_2$$

$$\cdot y_0 = y_1 - 1, x_1 \leq x_0 \leq x_2$$

$$\cdot y_0 = y_2 + 1, x_1 \leq x_0 \leq x_2$$

上記の条件のいずれかを満たすことである。

条件を満たす L の集合を L_0 とする。

(2) L_i の各要素 L に対し、評価値 $W(L)$ を計算する。

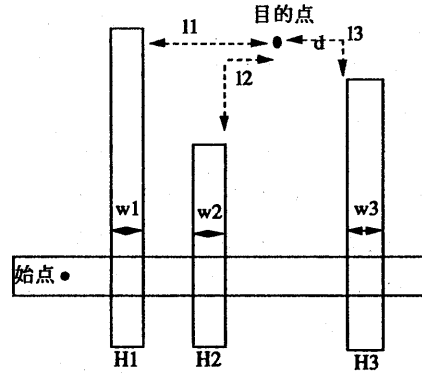
$W = f$ (目的点への距離、迂回距離、配線矩形の幅、配線層、etc) として定義され、関数 f は適用システムによって、様々な形を持つものであるが、例えば各項目は以下の意味を持つ (図3.6参照)。

- ・目的点への距離: 始点から目的点へより近づく経路を良い経路とする。
- ・迂回距離: 迂回により配線長が長くなるような考慮する。
- ・配線矩形の幅: 一般に幅が広いほど、経路発見の可能性が高い。また、通常幅より太い配線やバス配線の経路を探すときは、指定幅以上の配線矩形を探す必要がある。
- ・配線層: 層により配線抵抗が異なることを考慮する場合、階層レイアウトで処理中の階層で使える層が決まっている場合等考慮する。

今、 W の値が小さい程評価値が良いと考える。

例えば、

$$W = A \times (\text{目的点への距離}) + B \times (\text{迂回距離}) \\ - C \times (\text{配線矩形幅}) \quad A, B, C > 0$$



評価値 $W = f$ (目的点への距離、迂回距離、矩形幅)

$$W(H1) = f(11, 0, w1)$$

$$W(H2) = f(12, 0, w2)$$

$$W(H3) = f(13, d, w3)$$

図3.6 評価値 W

とおく。

L_i の要素から、 $W(L)$ が最小のもの L_i を選ぶ。

(3) L_i が存在しない場合は L_{i-1} について

(2) を繰り返す。

(4) L_i が目的点に隣接する場合、配線矩形間経路発見として 3.4 へ進む。

(5) L_i が目的点に隣接しない場合、 L_i と隣合う層で交差する配線矩形を求める。

$H \in L_i$ の場合、 $V \in L_i$ の交差配線矩形
 $V \in L_i$ の場合、 $H \in L_i$ の交差配線矩形
 の関係にある。

求めた配線矩形の集合を L_{i+1} と定義する。

(6) (2) へ戻る。

3.3 詳細配線経路決定

次に 3.2 で求めた配線矩形間経路の中で、詳細な配線経路を決定する 2 段階配線を行なう。ここでは、3.1 の配線矩形定義から明らかなように、詳細な配線経路の存在は保証されており、複数の詳細な配線経路の内から最も良いものを選ぶことを行なう。これはラインサーチ法とは本質的に異なる点である。ラインサーチ法は経路探索時に、1本の経路

を決定しながら進むが、決定した経路が最終的に良いかどうか途中では予測が難しい。これに対し本稿の手法は配線矩形という形で実質的に複数の経路を同時探索しているため、経路探索時に1本を無理して選ぶ必要がなく、経路が保証された後、これから述べる様に決定すればよい。処理は目的点に隣接する配線矩形から、順番に始点に隣接する矩形まで行なう(図3.7参照)。

- ・配線矩形が探索点(始点、目的点)に隣接するとき、詳細な配線経路は探索点の位置に依存する。
- ・配線矩形が探索点に隣接しないとき、その内部において全体の配線長が最も短くなるように、詳細な配線経路を決定する。そのとき、2.2で述べた隣接制約を考慮して決定する。隣接制約により、詳細な配線経路が当該位置に決定できないとき、次善の位置(制約を満足する範囲で最も配線長が短くなる位置)に経路を決定する。

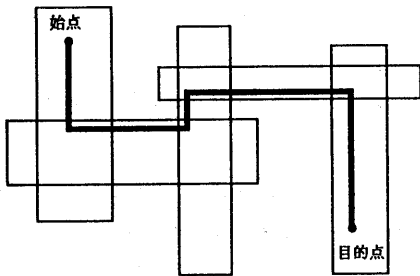


図3.7 詳細な配線経路の決定

3.4 配線矩形の分割・更新

ステップ3.3で得た詳細な配線経路は、次のネットの処理時には配線禁止領域として扱う必要がある。従って、配線矩形内部の詳細な配線経路を配線禁止とし、3.2と同様に配線矩形を分割更新する。この時、ステップ3.3で求めた配線経路について、隣接制約を考慮して、分割更新を行なう。配線矩形の分割・更新の様子を図3.8に示す。

上記、3.1から3.4までの矩形探索迷路法の処理の様子を、図3.9に示す。

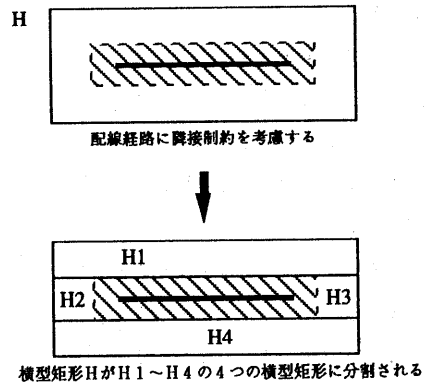


図3.8 配線矩形の分割・更新

4. アルゴリズムの拡張

ダイレイ、特性考慮配線に対するアルゴリズムの拡張性について、以下に示す。

4.1 幅広配線

配線幅を通常より太くした配線に対し、矩形探索迷路法においては、経路をある幅を持った配線矩形という領域で探索するため、容易に幅を持った配線経路を扱うことが出来る。

今、配線幅 $a (> 1)$ の配線経路を求めたいとき具体的な処理方法を以下に述べる。矩形間経路を求めるとき、その内部において幅 a の詳細な経路を決定する必要があるため、幅 a 以上の配線矩形を用いて矩形間経路を求める。その後、求めた矩形間経路

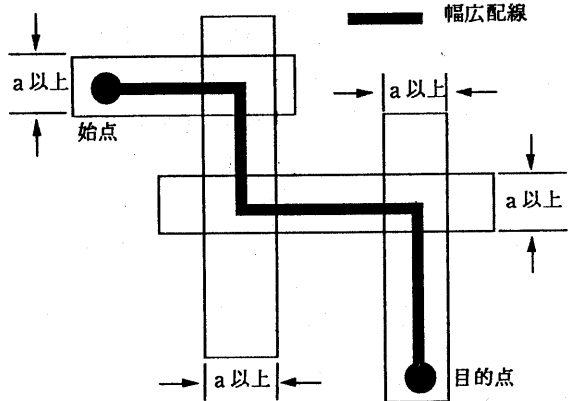


図4.1 幅広配線

せて、100Kゲート規模のCMOSLSIのブロック間配線に適用した。本システムでは最初に矩形探索迷路法で配線し、残った配線を従来の迷路法で配線した。その結果、LSIのレイアウト設計における処理時間を従来の1/5に低減した。

6. おわりに

本稿では、多様化するLSIレイアウト設計時の配線処理問題に焦点を当てて、機能的に汎用性が高い配線処理手法を提案した。本手法は以下のような特長を持つ。

1. 配線経路の探索を領域（矩形）で行なう。よって、一度に広い範囲を探索でき、高速な処理を実現する。
2. 領域で経路を探索するため、処理途中においては詳細な配線経路を決定する必要がないため、汎用性が高い。

7. 参考文献

1) 佐藤、降矢、大附：計算幾何学的手法を用いた

二層ルータ；電子通信学会 設計自動化（1984）

2) C. Y. LEE：An Algorithm for Path Connections and its Application；IRE Trans. on Electronic Computers, Vol.EC-10, pp.346-365（1961）

3) D.W. Hightower：A Solution to Line-Routing Problem on the Continuous Plane；Proc.6th Design Automation Workshop, pp.1-24,（1969）

4) J. Soukup：Fast Maze Router；Proc.15th Design Automation Conf., pp.100-102（1978）

5) Hashimoto, Stevens：Wire Routing by Optimizing Channel Assignment within Large Apertures；Proc.8th Design Automation Workshop, pp.155-169（1971）

6) 佐藤、岸田、高亀、池本、早瀬、山田：複数アルゴリズムを組み合わせた配線手法；情報処理学会 第26回 全国大会（1986）

7) 佐藤、湯山、岩倉、三宅、西山、池本：大規模LSI向3層分割配線手法；情報処理学会 設計自動化35-4（1986）

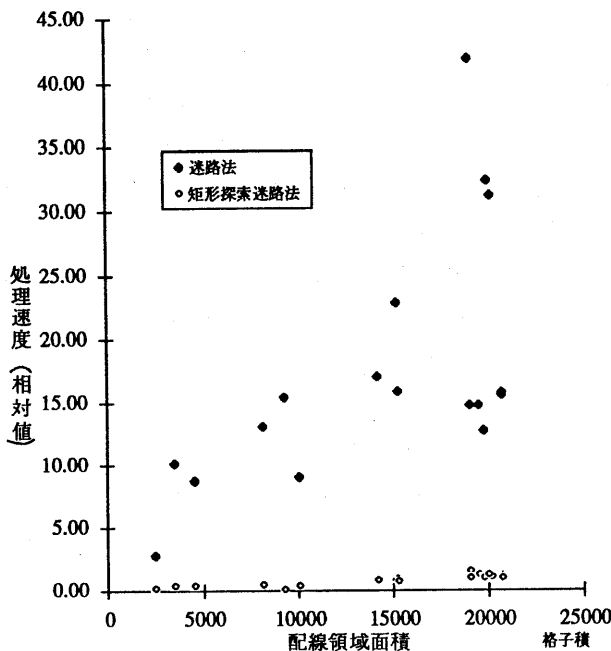


図5.1 自動配線処理時間