

## 対話型2点間配線プログラムの開発

8N-2

竹田 淳 定兼 利行 寺井 正幸

三菱電機(株)

## 1 はじめに

人手レイアウト設計は今日でも必要不可欠であり、特にアナログLSIのレイアウト設計は、大部分を人手で行なっているのが現状である。人手レイアウト設計では配線に要する時間の占める割合が大きい。従って、人手レイアウトを行なう場合、配線の効率化は設計時間の短縮にとって極めて効果的である。しかし、これまでに実用化されている自動配線プログラムのほとんどは、対話的な指定ができなかったり、レイアウト以外に回路図や接続情報を必要とするため、人手レイアウトに適用するのは難しい。そこで、我々は、人手レイアウト設計効率化のための対話型2点間配線プログラム(以下、2点間ルータ)を開発し、レイアウトエディタから呼び出すことのできるコマンドとして実現した。2点間ルータは、ユーザが対話的に指定する2点間の一層配線を自動生成するものであり、オプションとして、配線の途中経路や、配線が可能な範囲や、配線を禁止する領域をユーザが指定することもできる。これらの指定をすることにより、ユーザである設計者は、人手で行なうのと同じぐらい柔軟に配線を行なうことができる。

2点間配線のアルゴリズムは大別すると、迷路法[1]、線分探索法[2]、線分展開法[3]、タイル探索法[4]などがある。このうち、迷路法は計算時間とメモリを大量に消費する。これは対話処理を行なうには致命的である。線分探索法の場合、混雑したレイアウトでは迷路法よりも計算時間が長くなる。線分展開法とタイル探索法は、共に、経路が存在すれば必ず配線可能であるし、計算時間やメモリ使用量もそれほど大きくない。タイル探索法は、配線領域を長方形分割し、長方形探索により配線経路を見つけるものである。タイル探索法ではデータとして基本的に長方形しか必要としないので、インプリメンテーションが容易である。このため、我々は配線アルゴリズムにタイル探索法を採用した。特に、2点間ルータでは、高速処理を実現するために、長方形探索の各時点で始点からの経路長が最短の長方形を選択するアルゴリズムを用いた。これは、グリーディなアルゴリズムであり、最短経路を必ずしも保証しないが、経路長の計算を各長方形について一回しか行なわないので、計算時間が短くなる。この結果、2点間ルータは、約100素子からなる回路ブロックに対しても10秒以内で配線可能であり、大規模回路に対しても適用可能である。

## 2 2点間配線プログラムの概要

2点間ルータでは、配線を行なう前に前処理を必要とする。前処理では配線領域をいくつかの長方形に分割する。ここでは、前処理と配線処理の機能の概要を説明をする。

## 2.1 前処理

前処理時には、ユーザは配線レイヤと配線幅を入力する。また、必要ならば配線可能範囲や配線禁止領域も

入力する。配線可能範囲とは、2点間ルータの生成する配線の中心線が通ることのできる領域(配線領域)の外枠のことであり、配線禁止領域とは、2点間ルータの生成する配線の中心線が通ることのできない領域のことであり、ユーザが特に配線可能範囲を指定しない時は、対象となるレイアウト図形を囲む範囲が配線可能範囲になる。配線レイヤの既配置の図形とその周辺で、そこに配線すればスペースシグナル違反になる領域は、自動的に配線禁止領域になる。前処理では、これらの入力を用いて、レイアウト図形から図形演算により配線領域を抽出する。更に、抽出された配線領域をいくつかの長方形に分割する。前処理によりいくつかの長方形に分割した後の配線領域の略図を図1に示す。2点間ルータは、長方形分割された配線領域内に配線の中心線を置く。このため、生成された配線はスペースシグナルを満たす。

2点間ルータで配線を行なった後には、生成された配線の周辺は配線禁止領域になるので、配線領域を更新する必要があるが、この配線領域の更新処理は、生成された配線の周辺のみで行なうことによって、高速に実行することができる。

## 2.2 配線処理

2点間ルータは、ユーザが配線の始点(S)と終点(T)の座標、及びSとTにおける配線方向をマウスで指定すると、SとTを結ぶ一層配線を生成する。経路が存在する場合には、必ず見つけて配線することができる。

途中経路の指定をする場合には、ユーザは、配線の始点(S)と終点(T)の座標、SとTにおける配線方向、及び途中経路の概略位置(A1, A2, ...)をマウスで指定する。この時には、本プログラムは、A1, A2, ...をそれぞれ含む各長方形(前処理で配線領域を分割した時に得られる)を通してSとTを結ぶ一層配線を生成する。この指定方法により、ユーザは配線のトポロジーを決めることができる。2点間ルータによる配線例の概略図を図2に示す。

## 3 配線アルゴリズム

ここでは2点間ルータの配線アルゴリズムについて説明する。まず、途中経路の指定をしない場合について考える。2点間ルータでは、始点Sから終点Tまでの経路探索は、長方形の探索で行なう。アルゴリズムの説明のために、前処理で得られた各長方形と始点Sと終点Tを節点とし、隣接している節点の間に辺を持つグラフVを用いる。

始点Sから終点Tまでの経路探索は、グラフVにおける節点Sから節点Tまでの探索である。節点Sから隣接する節点(長方形または終点T)を優先度に基づいて探索して行き、終点Tにまで到達できれば逆探索により詳細な配線経路を決定し、到達できなければ経路が存在しないので、配線不可能であるというメッセージを表示する。ここで、探索する節点の優先度は、その節点が表す長方形(または終点T)までの始点Sからの経路長で決められる。この経路長が短い節点ほど高い優先度が与えられ、先に探索される。ただし、各節点までの経路は、その節点に隣接する節点が最初に探索された時に一意的に決定する。たとえば、別の経路でその節点に到達で

きる場合でも、一度決定した経路を後から変更することはない。従って、各節点までの距離も、最初に経路を決定する時に一度計算されるだけである。後から見つかる経路の方が距離が短くなる可能性があるが、2点間ルータでは、途中経路指定ができるので、応答速度を重視して、必ずしも最短経路になるとは限らないが、各節点の距離は一度しか計算しないことにしている。配線のアルゴリズムを図3に示す。ここで、 $U$  は  $S$  からの経路が求まっているがまだ探索されていないグラフ  $V$  の節点の集合を表し、 $Neigh(z)$  は、節点  $z$  に隣接し、経路が決まっていない節点の集合を表す。実際には、節点の集合  $U$  は Heap で実現している。この時、本配線アルゴリズムの計算時間は、 $O(N \log N)$  (但し、 $N$  は  $S$  からの経路が求まっている長方形の数) になる。

途中経路の概略位置の指定をする場合も、基本的には上記のアルゴリズムを用いる。途中経路の概略位置を  $A1, A2, \dots$  とすると、まず始点  $S$  から  $A1$  を含む長方形までの経路を上記のアルゴリズムで求め、次に  $A1$  を含む長方形から  $A2$  を含む長方形までの経路を上記のアルゴリズムで求め、以下同様に経路を求めていき、最終的に終点  $T$  までの経路を求める。

#### 4 適用結果

我々は2点間ルータをレイアウトエディタから呼び出すコマンドとして実現した。この2点間ルータを実際の回路ブロックに適用した時の実行時間を表1に示す。表1において、前処理時間は、対象レイアウト全体に対して行なった場合を示している。配線時間は、配線長が長くなるような10本の異なる配線を行なった内で、もっとも時間が掛かった場合の値を示している。この時、途中経路指定は行っていない。これらはすべて、ほとんど完成されたレイアウトに対するものであり、レイアウト設計の初期では、前処理時間、配線時間もかなり短くなる。また、配線可能範囲や配線禁止領域の指定により部分的な前処理を行えば、前処理時間、配線時間もさらに短くなる。計算機には、SUN4/260を用いた。

表 1: 2点間配線プログラムの実行時間

| 回路    | 素子数   | 前処理時間 | 配線時間 |
|-------|-------|-------|------|
| アナログ1 | 66    | 7秒    | 6秒   |
| アナログ2 | 56    | 5秒    | 7秒   |
| デジタル1 | 100   | 6秒    | 4秒   |
| デジタル2 | 約1.7K | 40秒   | 25秒  |

#### 5 おわりに

本稿では、アナログLSI等の人手レイアウト設計を効率化するために開発した、2点間ルータについて説明した。2点間ルータは、すでに画像処理用LSIのブロック間配線に適用されている。今後、アナログLSIのレイアウト設計にも適用していく予定である。

#### 謝辞

本開発を企画され、御指導下された、当社佐藤興二郎長に感謝致します。

#### 参考文献

[1] C. Y. Lee, "An Algorithm for Path Connection and Its Applications," IRE Trans. on Electronic Computers, pp.346-365, 1961.

[2] K. Mikami and K. Tabuchi, "A Computer Program for Optimal Routing of Printed Circuit Conductors," IFIP Congress 68, pp.1475-1478, 1968.  
 [3] W. Heyns, W. Sansen, and H. Beke, "A Line-Expansion Algorithm for the General Routing Problem with a Guaranteed Solution," Proc. 17th Design Automat. Conf., pp.243-249, 1980.  
 [4] 坂中, 小島, 佐藤, 大附 "配線問題におけるタイル探索法の評価", 信学技法, VLD87-115, 1987.

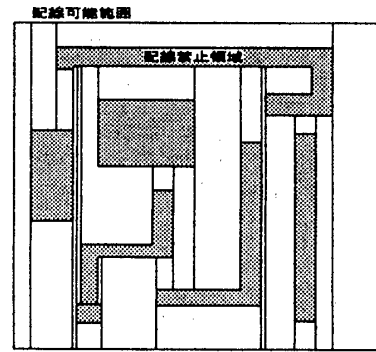


図 1: 前処理による配線領域の長方形分割

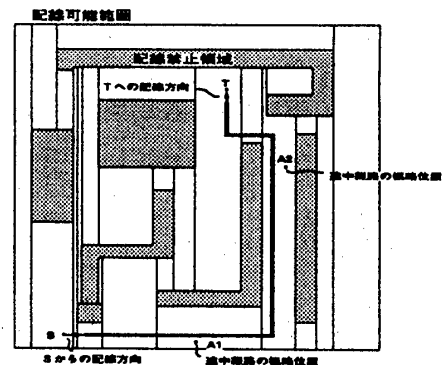


図 2: 2点間配線プログラムによる配線の概略図

```

procedure route(V : graph; S, T : node of V);
var
  U : set of nodes of V;
  y, z : node of V;
begin
  U := {S};
  for U ≠ ∅ do
  begin
    z := U の要素で S から最短のもの;
    U := U - {z};
    if z = T then
    begin
      経路を逆探索して配線を生成;
      終了;
    end;
    for y ∈ Neigh(z) do
    begin
      S から z を通る y までの経路とその距離を求める;
      U := U + {y};
    end;
  end;
  配線不可能と表示;
  終了;
end{route};
    
```

図 3: 配線アルゴリズム