

# 距離最短とビア数削減を目指した多端子配線法 A wire length and via driven multi-terminal routing method

渡邊 真規                      豊永 昌彦  
Masaki Watanabe              Masahiko Toyonaga

高知大学理学部数理情報科学科  
Faculty of Science, Kochi University

## あらまし

LSI 製造技術の向上により電子回路の微細化が進み、配線長に起因する信号遅延が増大している。配線経路において折れ曲がりがあると、高抵抗なビアが挿入されるため、より高速な信号伝播のためには短距離で折れ曲がりの少ない配線が求められる。

そこで著者は、配線領域にラベル付けをおこなった最短となる擬似スタイナー点と経路選択および経路探索のラベル多重化による折れ曲がりの少ない配線手法を提案する。

本手法の特徴は、多端子の擬似スタイナー点と部分経路における L 型経路について最適に選択する点である。実験の結果、より配線長の短い多端子配線が確認できたので報告する。

## 1. はじめに

高度情報化を支える電子機器の高性能化が進められている。電子機器の情報処理は、主に LSI (大規模集積回路) と呼ばれる半導体回路により行われている。これら半導体回路は、高集積化、高性能化を目指して製造技術の微細化が進められている。

電子回路において信号伝播遅延が配線抵抗に比例することから、より高性能な LSI を提供するた

めには、より短い、抵抗の少ない配線設計技術が不可欠である。

LSI 回路設計において、配線経路の形状はレイアウト設計段階における配線設計で決定される。また配線層と層との接合は経路の折れ曲がり点でビア (スルーホール) を通じておこなう。ビアは、アルミニウムや銅などが使われる信号配線と異なり、タングステンなどが使われるため、配線に比べて抵抗値が高い (TSMC 社の 0.18 ミクロンメートルの製造では単位長さあたりの抵抗値が配線長の 100 倍にも達する)。従って、配線におけるビア数は信号遅延の主要な原因となり得る。

これとは別に、ビアは製造時の歩留まりに影響する。ビアの「穴あけ」が多数となると完全な製造が難しくなり不良品の原因となる。

従って、LSI 回路にとって折れ曲がりが少ない配線経路を実現する何らかの方法が必要である。

著者は本研究において、折れ曲がりを最小化する多端子配線手法を提案する。

提案するアルゴリズムは、トレースバックコードを多重化することで、より直線的な経路を選択するステップと、端子分布の中心部へと折れ曲がるような経路を選択することで、総配線長を短くするステップで構成され、より直線で最短な配線経路を自動設計することができる。本提案手法は、

ランダムに配した端子を使って確認された。

以下の構成は次のとおりである。

第 2 章は、研究の動機について説明する。第 3 章、第 4 章は従来の迷路配線法との違いを中心として本手法のアルゴリズムを説明する。第 5 章は、本手法で行った実験結果について述べる。最後の第 6 章で、本研究のまとめと今後の課題を述べる。

## 2. 研究の動機

自動配線法の 1 つである迷路配線法では、配線領域を設計ルール（配線と配線、ビアとビアの間隔などを定めた規則）を満たす間隔のグリッドで敷き詰めたグリッドモデルを用いる。始点から隣接するグリッドに後戻りラベル付け（トレースバック）を繰り返して、終点に到達したときに、そのラベルをたどって配線経路を求める。

配線領域内に始点から終点までの配線経路が存在すれば最短な配線経路が得ることが保証されている。

しかし、1 つのグリッドへのラベル付けは、始点から同じ距離であっても先にラベル付けした方向が優先される。従ってラベル付け順によっては、直線化できる経路であっても意図しない折れ曲がりが生じることがある。

そこで本研究室では、一昨年ラベル付けを多重化することで、折れ曲がりの少ない配線経路を得る 2 端子配線手法を提案した[2]。この手法は、トレースバックに際して多重化されたラベルのうち、「直線優先」でラベル方向を選択していくことで、直線を保つ配線経路を得る。

しかし、現実の回路の配線設計では、2 端子配線に限らず多数の端子を同時に配線する場合が多い。そこで、著者は、本論文において同手法[2]を多端

子配線へと拡張する方法を検討した。その結果、より直線的で配線長の短い多端子配線を行ううえで、多端子の一部となる 2 端子間において最短となる直線的な経路が複数存在することがわかった。

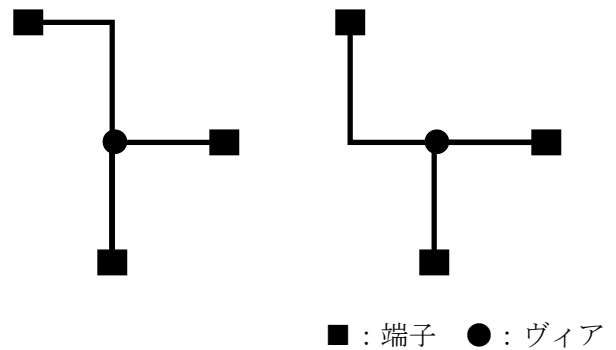


図 1 最適な経路

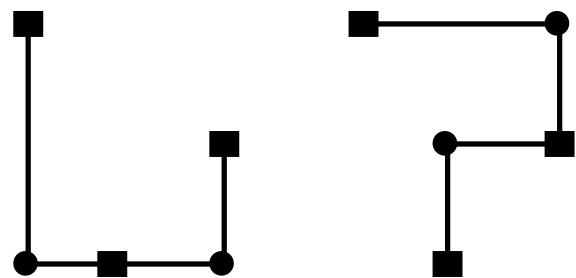


図 2 良くない経路

上記の図 1 に示したような場合は、どちらの経路も、総配線長を変えないので問題はないが、図 2 に示すような多端子配線においては、全体の配線長を考えたとき、その複数の経路のなかで最適な経路とそうでない経路とが生じてしまう。

そこで、著者は、まず多端子全体の領域内の分布を把握できるマップを生成し、そこからより短い経路を選べる経路選択法を提案する。

ここで端子フィールドマップと呼んでいる配線領域内の端子分布は、各グリッドに同グリッドから全端子への距離を記入したものであり、最小値

となるグリッドは、端子分布の中心部に相当する。つまり、数値の低いほうに向かうほど、より多くの他の端子に近い経路であると考えられる。そのため、多重のトレースバックコードでトレースバックする際、フィールドマップ上の数値の低いグリッドへの方向を優先して選択するようにすると、総配線長の短い配線になると予想される。

以下の章において、従来法のアルゴリズムと本手法を比較し、その優位性を説明する。

### 3. 従来の迷路配線法

#### 3.1. 従来の迷路配線法

迷路配線法は、まず配線情報から端子点をグリッド上に記入する。

##### 迷路配線法アルゴリズム

###### 1° 準備

- 1.1 配線情報の入力
- 1.2 ピンリスト $P2 = \phi$
- 1.3 開始点 $p_0$ として、 $P1 = \{p_0\}$ とする。

###### 2° 迷路配線法による経路検索

- 2.1 リスト $P1$ のすべての要素 $p$ について以下をおこなう。
  - 2.1.1 端子 $p$ の周囲のグリッド $p'$ とする。
  - 2.1.2  $p'$ が空きグリッドなら、TBコード書き、 $p'$ をリスト $P2$ に加える。
  - 2.1.3  $p'$ が終点ならば、3へ行く。
- 2.2 以上で得られたリスト $P2$ が空なら、「配線不能」として終了し、そうでなければ、 $P2$ を $P1$ にコピーして、 $P2 = \phi$ として、2.1へ戻る。

###### 3° トレースバック処理をおこなう

- 3.1 終点からTBコードをたどり始点までの経路を求める。

図3 迷路配線アルゴリズム

まず作業用リスト $P1$ ,  $P2$ を用意し、任意に決めた始点を $P1$ に登録する。

まず、リスト $P1$ に登録された端子を取り出しその隣接グリッドが障害物でなければ後戻りラベル（トレースバックコード）を設定していく。このとき、トレースバックコードを記入できたグリッド点をリスト $P2$ に登録する。この作業の途中で終

点が見つければ、終点からトレースバックコードをたどって経路を決定し、配線形状を出力する。リスト $P1$ のすべての要素の点について、以上の処理が終了したら、リスト $P2$ の要素をリスト $P1$ に移動させリスト $P2$ を空にし、再び2の作業からくり返す。

#### 3.2. 直進性を重視した配線法

直進性を重視した配線法[2]は、配線対象となるネットの端子位置情報、障害物情報、配線領域情報を入力として、ネット端子間を配線する。基本アルゴリズムは迷路配線法（LEEアルゴリズム）である。ただし、距離が同じであれば、1つの隣接グリッドに2つのトレースバックコード（開始点への戻りマーク）を設定できる。

トレースバックする際には、向きの同じトレースバックコードを優先して経路を決定する。以下、アルゴリズムの概要を以下に示す。

##### 直進性を重視する迷路配線法アルゴリズム

###### 1° 準備

- 1.1 配線情報の入力
- 1.2 ピンリスト $P2 = \phi$
- 1.3 開始点 $p_0$ として、 $P1 = \{p_0\}$ とする。

###### 2° 迷路配線法による経路検索

- 2.1 リスト $P1$ のすべての要素 $p$ について以下をおこなう。
  - 2.1.1 端子 $p$ の周囲のグリッド $p'$ とする。
  - 2.1.2  $p'$ が空きグリッドなら、TBコードおよび開始点 $p_0$ から $p'$ までの距離 $d$ を書き、 $p'$ をリスト $P2$ に加える。
  - 2.1.3  $p'$ にTBコードが設定されており、 $p_0$ からの距離 $d$ が同じなら $p'$ に第2TBコードを設定して、リスト $P2$ に加える。
  - 2.1.4  $p'$ が終点ならば、3へ行く。
- 2.2 以上で得られた $P2$ が空なら、「配線不能」として終了し、そうでなければ、 $P2$ を $P1$ にコピーして、 $P2 = \phi$ として、2.1へ戻る。

###### 3° トレースバック処理をおこなう

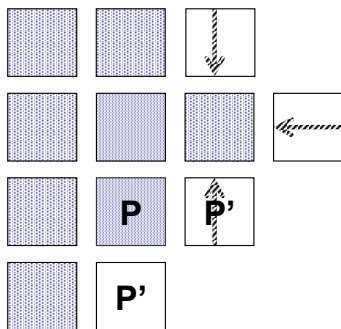
- 3.1 終点からTBコードをたどり始点までの経路を求める。多重のTBコードについては、同一方向を優先して戻る。

図4 直進性を重視した迷路配線アルゴリズム

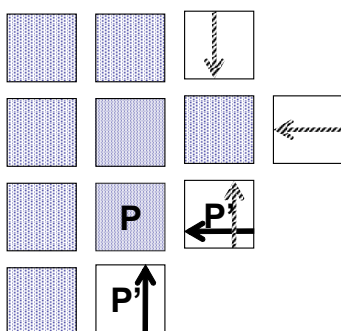
### (1)TB 選択方法

図 5(a)で、影の四角は、探索済の格子点、白四角は、探索対象の格子点とする。影矢印は、探索されることでトレースバックコードが設置された白四角である。今、すでに他の格子から TB コードが設定された白四角 P'と、単なる白四角に隣接する格子点 P が、各 P'にトレースバックを設定する場合を説明する図である。

従来の配線法では、影矢印の P'は、すでに他の格子からチェックされているため新たに TB コードを設定することができない。しかし、提案手法では、図 5(b)のように、2つまでの影矢印をつけることをゆるすことで、二重に TB を設定できる。これが本手法の TB コード設定の特徴である。



(a) P の隣接 P' に TB 定義された場合

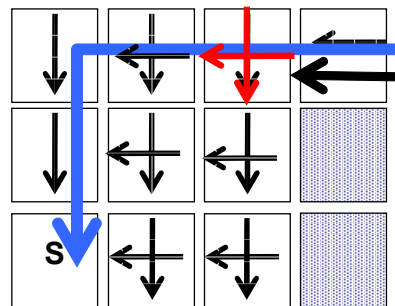


(b) P の隣接 P' に TB を二重で定義

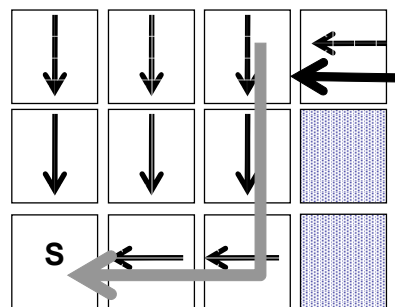
図 5 トレースバックの設定

### (2)経路決定方法の比較

下図 6 で説明する。まず目標地点からトレースバックされてくる。次に2つの道にさしかかる。ここで2つの TB コードが選択できるので、直線性を優先して TB されてきた同じ方向に進み、スタート地点に戻る。これが本手法による2つの TB コードで折れ曲がりの少ない方向を選択する方法である。従来の迷路配線法だと、図 6(b)のように、ただ1つの矢印に従って折れ曲がりの多い配線となってしまうことがわかる。



(a) 2つの TB コードでより直線を選ぶ



(b) 1つの TB コードで経路を選ぶ

図 6 トレースバック中の経路の選択

## 4. 直進性を重視した多端子配線法

### 4.1. 多端子配線における課題

本章では、2端子限定であった直進性を重視した配線法[2]を、多端子配線へと拡張する。

多端子配線法は、2つ以上の各端子毎に経路探索、トレースバックを端子の数だけおこなうことで多端子に対応する。

以下、アルゴリズムの概要を以下に示す。

#### 1° 準備

- 1.1 配線情報の入力
- 1.2 端子リスト  $P1 = \{p1, \dots, pn\}$
- 1.3 ピンリスト  $P3 = \phi$
- 1.4  $p1$  を開始点  $p0$  として、 $P2 = \{p0\}$  とする。
- 1.5 終点リスト  $T = \{p2, \dots, pn\}$  とする。

#### 2° 迷路配線法による経路検索

- 2.1 リスト  $P2$  のすべての要素  $p$  について以下をおこなう。
  - 2.1.1 端子  $p$  の周囲のグリッド  $p'$  とする。
  - 2.1.2  $p'$  が空きグリッドなら、TBコードおよび開始点  $p0$  から  $p'$  までの距離  $d$  を書き、 $p'$  をリスト  $P3$  に加える。また、ビア数も設定する。
  - 2.1.3  $p'$  にTBコードが設定されており、 $p0$  からの距離  $d$  が同じなら  $p'$  に第2TBコードを設定して、リスト  $P3$  に加える。また、ビア数も設定する。
  - 2.1.4  $p'$  が  $T$  の要素ならば、 $p'$  を終点とし、3へ行く。
- 2.2 以上で得られた  $P3$  が空なら、「配線不能」として終了し、そうでなければ、 $P3$  を  $P2$  にコピーして、 $P2 = \phi$  として、2.1へ戻る。

#### 3° トレースバック処理を行う

- 3.1 終点からTBコードをたどり始点までの経路を求める。多重のTBコードについては、同一方向を優先して戻る。
- 3.2 始点と終点を  $P1$  から削除し、 $P1$  が空になれば終了する。そうでなければ、 $P1$  の残りの要素から昇順に開始点  $p0$  とし、始点、終点、経路となったグリッドを終点リスト  $T$  に加え、2に戻る。

図7 直進性を重視した多端子配線法

ただし、このままのアルゴリズムでは、部分的な2端子間におけるL字形経路において、最適な経路とそうでない経路を判別できない可能性がある。2端子では問題とならないが、多端子においては全体の配線長に影響する。そこで多端子において配線長がより短くなる配線方法について見当が必要となる。

## 4.2. 端子フィールドマップを利用した配線法

本章では、著者が提案した「直進性を重視しつつ、総配線長がより短い経路を選択する配線法」[1]について述べる。

この配線法は、端子分布を距離で表す端子フィールドマップを作成し、そのマップを利用して、総配線長のより短い経路選択をおこなう手法を加えたものである。

これにより、前章4.1.で紹介した多端子配線法では判別できなかった、配線の部分的なL字形経路に関して、総配線長がより短くなるような経路を選択できる。

以下、アルゴリズムの概要を以下に示す。

#### 1° 準備

- 1.1 配線情報の入力
- 1.2 端子リスト  $P1 = \{p1, \dots, pn\}$
- 1.3 ピンリスト  $P3 = \phi$
- 1.4  $p1$  を開始点  $p0$  として、 $P2 = \{p0\}$  とする。
- 1.5 終点リスト  $T = \{p2, \dots, pn\}$  とする。

#### 2° 端子フィールドマップの生成

- 2.1 各端子について、配線領域に距離を記録したマップを作成する。
- 2.2 作成した全マップを足し合わせる。

#### 3° 迷路配線法による経路検索

- 3.1 リスト  $P2$  のすべての要素  $p$  について以下をおこなう。
  - 3.1.1 端子  $p$  の周囲のグリッド  $p'$  とする。
  - 3.1.2  $p'$  が空きグリッドなら、TBコードおよび開始点  $p0$  から  $p'$  までの距離  $d$  を書き、 $p'$  をリスト  $P3$  に加える。
  - 3.1.3  $p'$  にTBコードが設定されており、 $p0$  からの距離  $d$  が同じなら  $p'$  に第2TBコードを設定して、リスト  $P3$  に加える。
  - 3.1.4  $p'$  が  $T$  の要素ならば、 $p'$  を終点とし、4へ行く。
- 3.2 以上で得られた  $P3$  が空なら、「配線不能」として終了し、そうでなければ、 $P3$  を  $P2$  にコピーして、 $P2 = \phi$  として、3.1へ戻る。

#### 4° トレースバック処理を行う

- 4.1 終点からTBコードをたどり始点までの経路を求める。多重のTBコードについては、同一方向を優先して戻る。ただし、端子フィールドマップの数値が異なる場合、数値の低い方向を優先する。
- 4.2 始点と終点を  $P1$  から削除し、 $P1$  が空になれば終了する。そうでなければ、 $P1$  の残りの要素から昇順に開始点  $p0$  とし、始点、終点、経路となったグリッドを終点リスト  $T$  に加え、3に戻る。

図8 端子フィールドマップを利用した配線法



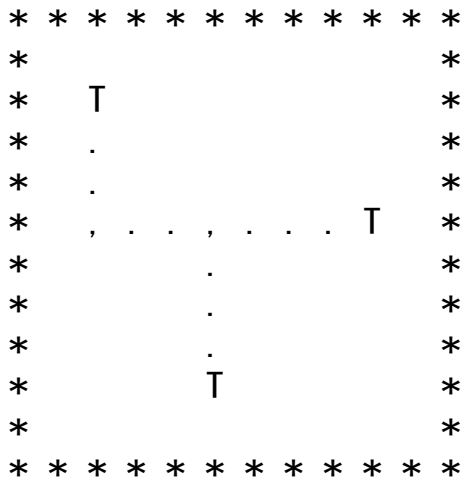


図 11 端子フィールドマップを活用した配線結果

#### 4.3. ヴィア数の計測も含めた提案手法

本章で著者は、[1]の配線法に、各グリッドにヴィア数を記録する手法を加えることで、より良く直進性を重視し、総配線長を減らす配線法を提案する。トレースバックする際には、向きの同じトレースバックコードを優先して経路を決定するだけでなく、2方向のヴィア数を比較し、数値の低い方向を優先する。また、ヴィア数が同じであれば、端子フィールドマップ上の数値の低い方向を優先する。以下、アルゴリズムの概要を以下に示す。

#### 1° 準備

- 1.1 配線情報の入力
- 1.2 端子リスト  $P1 = \{p1, \dots, pn\}$
- 1.3 ピンリスト  $P3 = \phi$
- 1.4  $p1$  を開始点  $p0$  として、 $P2 = \{p0\}$  とする。
- 1.5 終点リスト  $T = \{p2, \dots, pn\}$  とする。

#### 2° 端子フィールドマップの生成

- 2.1 各端子について、配線領域に距離を記録したマップを作成する。
- 2.2 作成した全マップを足し合わせる。

#### 3° 迷路配線法による経路検索

- 3.1 リスト  $P2$  のすべての要素  $p$  について以下をおこなう。
  - 3.1.1 端子  $p$  の周囲のグリッド  $p'$  とする。
  - 3.1.2  $p'$  が空きグリッドなら、TB コードおよび開始点  $p0$  から  $p'$  までの距離  $d$  を書き、 $p'$  をリスト  $P3$  に加える。また、ヴィア数も設定する。
  - 3.1.3  $p'$  に TB コードが設定されており、 $p0$  からの距離  $d$  が同じなら  $p'$  に第2TBコードを設定して、リスト  $P3$  に加える。また、ヴィア数も設定する。
  - 3.1.4  $p'$  が  $T$  の要素ならば、 $p'$  を終点とし、4へ行く。
- 3.2 以上で得られた  $P3$  が空なら、「配線不能」として終了し、そうでなければ、 $P3$  を  $P2$  にコピーして、 $P2 = \phi$  として、3.1へ戻る。

#### 4° トレースバック処理を行う

- 4.1 終点から TB コードをたどり始点までの経路を求める。多重の TB コードについては、まず二方向のヴィア数を比較し少ない方向を優先して戻る。ヴィア数が同じときは、端子フィールドマップの数値の低い方向を優先する。
- 4.2 始点と終点を  $P1$  から削除し、 $P1$  が空になれば終了する。そうでなければ、 $P1$  の残りの要素から昇順に開始点  $p0$  とし、始点、終点、経路となったグリッドを終点リスト  $T$  に加え、3に戻る。

図 12 直進性を重視する多端子配線法

#### (1) ヴィア数の設定

配線をより正確にするため、経路探索のたびにグリッドにヴィア数を書き込む。ヴィア数は水平方向と垂直方向の2種類に分けて設定する。ヴィア数の設定は、TB コードを設定するときに同時におこなう。設定方法は、まず、あるグリッド  $p$  から周囲のグリッド  $p'$  へ TB コードを設定する。そして、 $p$  の TB コードと  $p'$  の TB コードを比べ、同じ方向であれば  $p$  と同じヴィア数を登録し、違う方向であれば  $p$  のヴィア数に+1して登録する。このとき、 $p'$  に設定された TB コードの方向と同じ方向でヴィア数を登録する。

## 5. 実験

単純に多端子化した配線法および提案手法を C 言語 (Cygwin) で実装し、様々な端子分布について実験をおこなった。実験機として、CPU は Celeron(R) 2.53GHz、メモリは 2.54GHz、992MB RAM を使用した。

提案手法の有効性を確かめるため、TB コードの多重化および端子フィールドマップを使用せず、ビア数の計測もおこなわない単純に多端子化した迷路配線の結果と、提案手法の結果とを比較する。ここでは、前者を単純な多端子迷路配線、後者を提案手法と呼ぶことにする。

障害物のない例として図 13 を示す。端子数は 4 である。図 13(a) に単純な多端子迷路配線の例を、図 13(b) に提案手法による配線の例を示す。

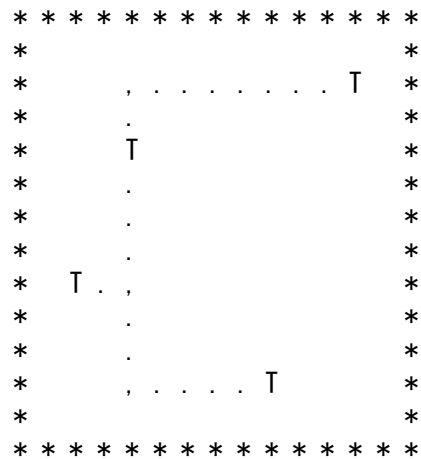
この例は、ビア数に違いはないが、図 13(a) における上部の配線形状（上側凸の L 字部）に対して、図 13(b) では中心部に向かう配線形状が達成されている（下側凸の L 字部）。これにより、配線長が 21 から 20 に短縮されている。

図 14 に障害物のある例を示す。端子数は 3 である。図 14(a) の障害物の左側の経路が、図 14(b) では、直進性を重視されて折れ曲がりが減っている。具体的には、総配線長は同じで、ビア数が 4 から 3 に削減されている。

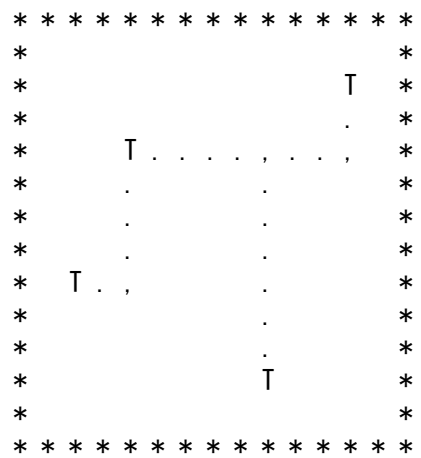
図 15 に、ビア数および総配線長の両方が改善されている例を示す。端子数は 4 である。単純な多端子迷路配線の例を図 15(a) に示し、提案手法による配線結果を図 15(b) に示す。図 15(a) における上部の配線形状（上側凸の L 字部）に対して、図

15(b) では中心部に向かう配線形状が達成されている（下側凸の L 字部）。これにより若干の配線長の短縮ができています。具体的な数値は、図 15(a) の配線長が 21 であるのに対して、提案手法の図 15(b) では 19 である。また、ビア数が 5 から 4 に削減できている。

図 16 に、ビア数は減るが総配線長の増える例を示す。端子数は 5 である。図 16(a) の例では、ビア数は 8、総配線長は 35 である。図 16(b) の例では、ビア数は 5、総配線長は 38 である。



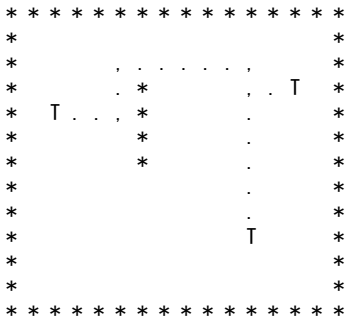
(a) 単純な多端子迷路配線の例



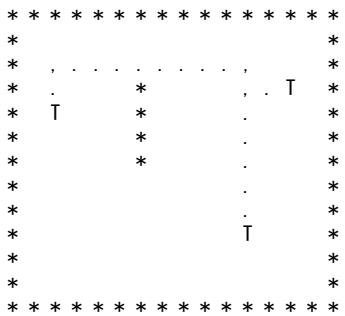
(b) 提案手法による配線の例

図 13 障害物のない例



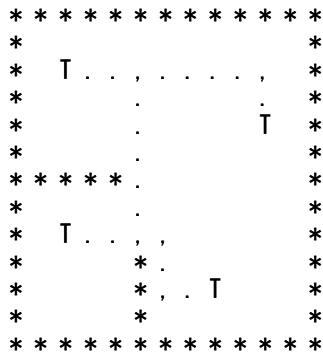


(a) 単純な多端子迷路配線の例

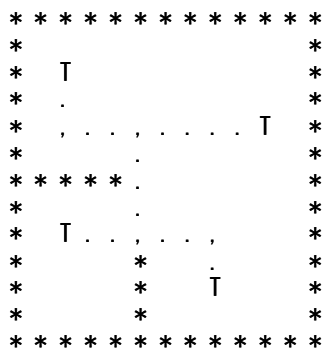


(b) 提案手法による配線の例

図 14 ヴィア数のみが改善されている例

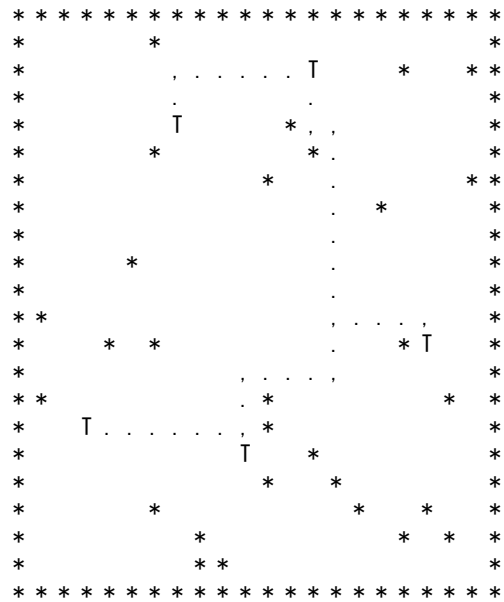


(a) 単純な多端子迷路配線の例

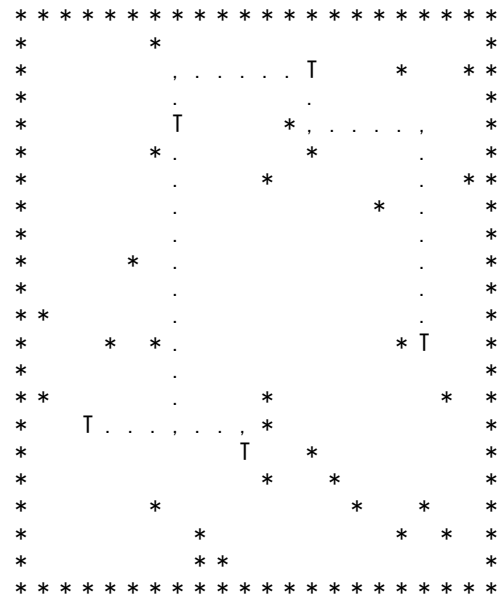


(b) 提案手法による配線の例

図 15 ヴィア数と総配線長が改善されている例



(a) 単純な多端子迷路配線の例



(b) 提案手法の例

図 16 ヴィア数は減るが配線長の増える例

```

* * * * *
*
*
*   T . ,
*
*   .
*
*   .
*   . . . . . T
*   . * * * * *
*   . . . . .
*
*           T
*
* * * * *

```

(a) TB コードの多重化と端子フィールドマップのみの配線

```

* * * * *
*
*
*   T
*
*   .
*
*   .
*
*   .
*   . * * * * *
*
*   .
*
*   .
*   . . . . . T . . . . .
*
* * * * *

```

(b) ヴィア数の判定も加えた配線法

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 25 22 1f 1e 1d 1e 1f 20 1f 1e 1d 1e 1f 22 25 28 0
0 22 1f 1c 1b 1a 1b 1c 1d 1c 1b 1a 1b 1c 1f 22 25 0
0 1f 1c . 18 17 18 19 1a 19 18 17 18 19 1c 1f 22 0
0 1e 1b 18 17 16 17 18 19 18 17 16 17 18 1b 1e 21 0
0 1d 1a 17 16 15 16 17 18 17 16 15 16 17 1a 1d 20 0
0 1c 19 16 15 14 15 16 17 16 15 14 15 . 19 1c 1f 0
0 1d 1a 17 16 15 0 0 0 0 0 15 16 17 1a 1d 20 0
0 1e 1b 18 17 16 15 14 13 14 15 16 17 18 1b 1e 21 0
0 1f 1c 19 18 17 16 15 14 15 16 17 18 19 1c 1f 22 0
0 20 1d 1a 19 18 17 16 . 16 17 18 19 1a 1d 20 23 0
0 23 20 1d 1c 1b 1a 19 18 19 1a 1b 1c 1d 20 23 26 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

(c) 端子フィールドマップ

図 17 ヴィア数の記録が必要である例

図 17(a)は、TB コードの多重化と端子フィールドマップ値の判定で配線をおこなった例で、図

17(b)は、さらにヴィア数の判定を加えた提案手法による例である。端子数は 3 である。図 17 では、障害物付近の端子フィールドマップ値が低いため、(a)ではそちらに折れ曲がってしまい無駄な折れ曲がりが増えてしまうことを示している。グリッドにヴィア数を記憶することで、(b)で示されているようにヴィア数が 5 から 2 へ改善されている。ただし、配線長は 17 と 19 で、(a)のほうが短い。

## 6. まとめ

障害の分布を反映して直進性を重視する（折れ曲がり数を削減する）多端子配線手法を検討した。実験例によれば、ヴィア数の削減とともに総配線長の短い経路、またはどちらか一方が改善された経路を得られることがわかった。

提案手法を用いても障害物の配置や配線する順番によっては、総配線長は減るがヴィアは増えるもの、逆にヴィアは減るが総配線長が増えるものがあるので、今後、それらをうまく配線する方法を検討したい。

## 参考文献

[1] 渡邊真規, 池野陽輔, 豊永昌彦 “障害回避を回避する直線指向の配線法” 平成 19 年度電気関係学会 四国支部連合大会 1-4 (2007 9 月 29 日)

[2] 池野陽輔, 坂東修司, 豊永昌彦 “直進性を重視する迷路配線法の検討” 平成 18 年度電気関係学会 四国支部連合大会 1-2(2006 9 月 26 日).

[3] 武永秀, 豊永昌彦, “配線層数の最小化を目指す多層再配線手法” 電子情報通信学会技術報告 VLD2005-7, pp7-12 (2005 年 5 月 20 日).

[4] 豊永昌彦, 武永秀, 國信茂郎 “再レイアウトに向けた単純多層迷路配線の有効性の考察” 情報処理学会 DA シンポジウム 2004 論文集, Vol. 2004, No. 8, pp. 115-120 (2004).

[5] 武永秀, 豊永昌彦 “単一ネット修正を指向した多層配線法の一手法” 平成 15 年度電気関係学会四国支部連合大会 1-12 (2003 10 月 12 日).