

L形チャネル配線のためのグリーディ算法

大中友志* 金整範* 佐藤政生** 大附辰夫*

* 早稲田大学理工学部

** 拓殖大学工学部

L S Iのレイアウト設計において、チャネル配線領域の定義と配線順序の決定は特にビルディングブロック方式において重要である。与えられた配置が直線によって再帰的に二分割できるスライシング構造の場合には配線領域の定義と配線順序の決定を容易に行うことができるが、ノンスライシング構造の配置の場合に100%配線を保証するためにはL形チャネルを導入する方法しか知られていない。そこで本稿では、L形チャネルを配線するグリーディな算法を提案する。提案する算法によって生成される配線パターンはごく一部斜めのものを含む可能性はあるが、基本的には縦配線を一方の層、横配線を他方の層とする縦横原則配線モデルを採用している。また、計算機による実験結果を報告する。

A Greedy Algorithm for L-shaped Channel Routing

Yuji Ohnaka*, Jung-Bum Kim*, Masao Sato**
and Tatsuo Ohtsuki

* School Of Science and Engineering, Waseda Univ.

3-4-1, Ohkubo, Shinjuku, Tokyo, 160 Japan

** Faculty of Engineering, Takushoku Univ.

The concept of L-shaped channels was first introduced by W.-M. Dai et al. in 1985 to generate a feasible routing order for nonslicing-structure placement in building-block layout design. A greedy algorithm is presented in this paper for routing L-shaped channels. The algorithm first routes the corner of an L-shaped channel, then horizontal and vertical subchannels. During the process the channel may be extended if there is not enough space to route it completely. Experimental results show that the algorithm provides good solutions to the L-shaped channel-routing problems.

1. はじめに

ビルディングブロック方式LSIのレイアウト設計は、通常、フロアプラン、概略配線、チャンネル配線領域の定義と配線順序の決定、チャンネル配線という流れに沿って行われる。フロアプランによって定められた配置が直線によって再帰的に二分割できる場合には、配線領域の定義と配線順序の決定は容易に行うことができる。しかし、そうでない（ノンスライシング構造の配置）場合には、チャンネル配線領域の配線順序付けにサイクルが生じるために順序が決定されないか、配線が可能である保証のないスイッチボックスを定義しなくてはならない。この問題を解決するために、ノンスライシング構造の配置においても100%配線を保証するチャンネル配線領域の定義と配線順序の決定方法が提案されている[1]。この方法では、図1のようにL形チャンネルを導入することにより配線順序のサイクルを解消している。L形チャンネルは初期の領域では配線不可能であっても、既配線チャンネルを再配線することなく拡大可能であるという点においてスイッチボックスとは本質的に異なる。

L形チャンネル配線に関する文献は[2]のみである。ここでは、幹線を一方の層、支線を他方の層とする縦横原則配線モデルと、支線のほとんどに斜め配線を用いる配線モデルが扱われている。前者のモデルにおける配線算法はL形チャンネルを配線するのに何回かの繰り返し配線操作を必要とし、後者のモデルは現実のものとは言い難い。そこで本稿では、繰り返すことなくL形チャンネルを配線するグリーディな算法を提案する。提案する算法によって生成される配線パターンはごく一部斜めのものを含む可能性はあるが、基本的には縦配線を一方の層、横配線を他方の層とする縦横原則配線モデルを採用している。最後に計算機による実験結果を報告する。

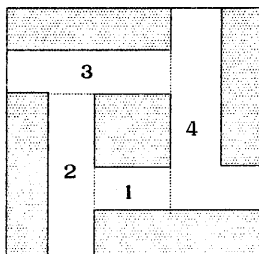


図1 ノンスライシング構造におけるチャンネルの定義と順序づけ

2. L形チャンネルの定義と前提条件

L形チャンネルは配線領域の定義のされ方によって4種類考えられるが、すべて図2の形ものを回転移動したものと考えることができるので、本稿では、一般性を失うことなく、図2の形のものに関してのみ論じる。

L形チャンネルとそれに関する語句を以下に定義する。

- ① L形チャンネルは文献[1]の方法によって定義されたものであり、配線不可能ではない。これは配線領域の定義方法によって保証されている[3]。
- ② L形チャンネルは、1つのinternal boundaryと1つのexternal boundary、それに2つのopen endsから構成される。internal boundaryの垂直・水平部分をそれぞれ V_i, H_i 、external boundaryの垂直・水平部分をそれぞれ V_e, H_e とし、 V_i と H_i の交点を C_i 、 V_e と H_e の交点を C_e とする（図2参照）。
- ③ internal boundaryは任意の方向に移動できる。ただし、 V_i, H_i が同時に同方向に移動するものとする。external boundaryは固定されており、移動できない。
- ④ boundary上の端子は固定されており、open ends上の端子は浮動である。このとき、端子は V_i, V_e では垂直辺上のみ、 H_i, H_e では水平辺上のみ存在する。

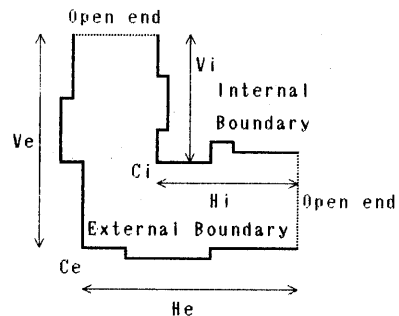


図2 L形チャンネル

また、本稿で扱うL形チャンネルの前提条件を以下に示す。

- a) 配線や端子、ビアはすべて格子構造に基づくものとする。
- b) 配線領域は2層で、基本的には配線領域全体に対し水平方向の配線に1層目を、垂直方向の配線に2層目を用いる。ただし、一部斜め配線（45°）を許す。

3. アルゴリズムの概要

提案する算法では、L形チャネルを配線する場合には、角の領域が最も混雑するという経験に基づいて次のような段階で配線を行う。

Phase 1. V_i, V_e 間、 H_i, H_e 間の配線混雑度に基づき、配線容量が配線要求を満足するように、internal boundary と external boundary の相対位置を設定する。

Phase 2. C_i から、水平・垂直方向に external boundary に交わるまでカットラインを引き、L形チャネルを図3のように3つの領域 α 、 β 、 γ に分割する。このとき各カットラインに対して、それと交差する必要があるネットを求め、そのネットの浮動仮端子をカットライン上に設ける。

Phase 3. 領域 α （隣り合う2辺に端子を持つチャネル）の配線を行う。初期設定された領域内で配線できなかったときには、internal boundary を移動して領域を広げ配線を完了させる。

Phase 4. 領域 β 、 γ （3辺に端子を持つチャネル）の配線を行う。

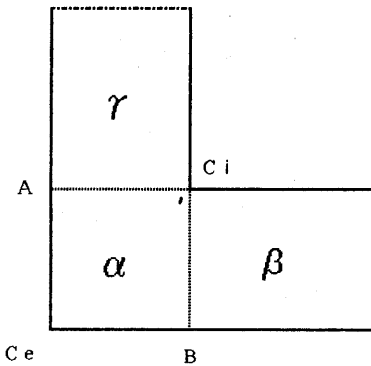


図3 チャンネル分割

以下の節では、Phase 3、Phase 4 についての詳細なアルゴリズムをそれぞれ述べる。

4. 領域 α の配線算法

領域 α は隣合う2辺（ $A C_e$ と $B C_e$ ）に固定端子、他の2辺（ $A C_i$ と $B C_i$ ）に浮動仮端子が存在する adjacent-side channel である。L形チャネルでは最も混雑するL形の角における配線方法が重要なので、本算法では α の中でも左下（ C_e ）から右および上に向けて

配線処理を行う。このとき、領域 α に含まれる全ての配線ネットをいくつかのタイプに分類し、まず領域を無駄にすることなく他のネットに無関係に配線できるものを優先的に配線し、次に残りのネットを互いに領域を有効に利用できるように配線していく。 α におけるコラムおよびトラックを図4のように定義する。

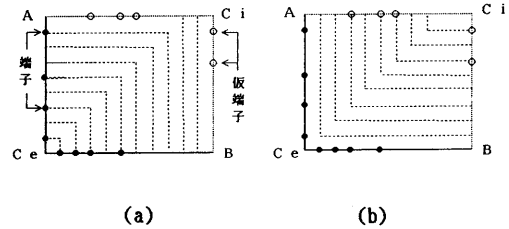


図4 領域 α 内での(a)コラムと(b)トラック

以下に配線算法を示す。

Step 1. 領域 α に含まれる全てのネットを、その端子の位置によって次の7つのタイプに分類する（図5）。

- Type I ... 2端子ネットで、 $A C_e$ 辺と $B C_i$ 辺に端子を持つもの
- Type II ... 2端子ネットで、 $B C_e$ 辺と $A C_i$ 辺に端子を持つもの
- Type III ... 4端子ネットで、4辺全てに端子を持つもの
- Type IV ... $A C_e$ 辺に端子を持ち、 $B C_e$ 辺に端子を持たないネット（ただし、TYPE I を除く）
- Type V ... $B C_e$ 辺に端子を持ち、 $A C_e$ 辺に端子を持たないネット（ただし、TYPE II を除く）
- Type VI ... $A C_e$ 辺にも $B C_e$ 辺にも端子を持つネット（ただし、TYPE III を除く）
- Type VII ... $A C_e$ 辺にも $B C_e$ 辺にも端子を持たないネット

以上のようなタイプ分けで、Type I, II, IIIのネットは領域を無駄にすることなく他のネットに無関係に一意的に配線できるネットである。Type IV, V, VIのネットはそれぞれ異なる配線パターンが考えられ、お互いの配線に影響を与えるネットであり、その配線パターンの組合せ方によっては配線領域を無駄に使う危険性がある（例、図6）。また、Type VIIのネットは他のタイプのネットの配線パターンを決定した後で配線することになる。

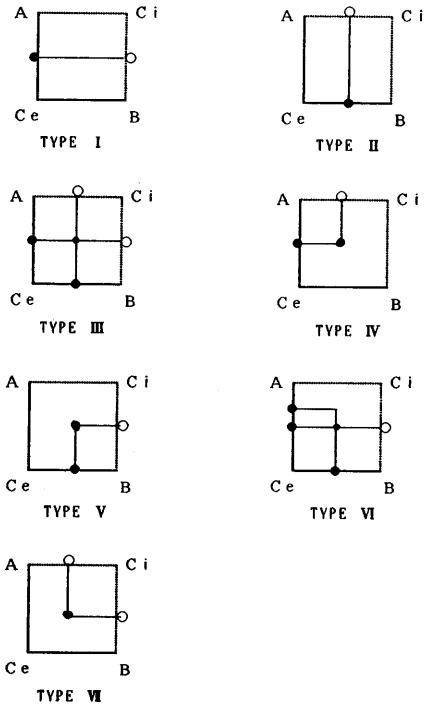


図5 領域αにおける7つのネットタイプ

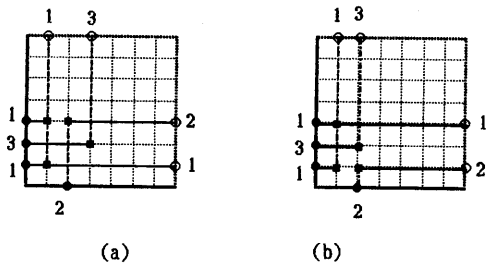


図6 (a)準最適配線と(b)最適配線

Step2. Type I, II, IIIのネットを、その固定端子が存在するコラムおよびトラックに割り当てる。以下では議論を簡単にするために、これらのコラムおよびトラックに関する記述を省略する。

Step3. Type IV, V, VIのネットを配線する。これらは、図7のようにCeから右および上方向にコラム毎に、また各コラムに於いては、左から右および下から上に向かって格子点毎に処理される。配線算法の詳細を述べる前に語句を定義しておく。

処理を行う格子点をPc、左隣の格子点(処理済)をPl、下隣の格子点(処理済)をPbとする。各格子

点は、その1層目2層目を占めるネット番号とそれらのネットが上および右に進むかどうか、さらにビアを設けるかどうかといった情報を保持している。また、Pcと同じ列で下側にある格子点の中で右に向かう配線を持たないもの、Pcと同じ行で左側にある格子点の中で上に向かう配線を持たないものの数をそれぞれPcにおける水平配線余裕、垂直配線余裕といい、最も左の端子がPcより右側にあるネットの数、最も下の端子がPcより上側にあるネットの数を、それぞれPcにおける水平配線要求、垂直配線要求と呼ぶ。

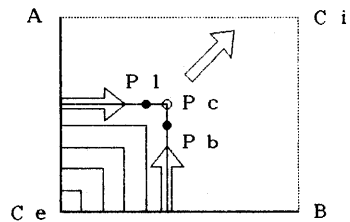


図7 領域αにおける配線方法

P1, P2における配線の情報により、Pcに関する処理は以下の5つに場合分けされる。

Case1. P1, P2からPcに向かって異なるネットが延長されている場合

Pcより右および上にそれぞれそのまま延長する(図8参照)。

Case2. P1, P2からPcに向かって同一のネットが延長されている場合

Pcで結線してビアを設け、そのネットのタイプに基づいて以下の(a)-(c)のいずれかの操作を行う。

(a) Type IVのネット

次のような場合に上に延長する。

- ・Pcの行より上側に端子があり、Pcと同じ行で左側の格子点の中にそのネットを上側に延長する配線(サブネット)が存在しない場合。

さらに次のような場合には右にも延長する。

- ・Pcの列より右側に端子がありPcと同じ列で下側の格子点の中にそのネットを右側に延長する配線(サブネット)が存在しないときに、Pcの行より上側にネットを延長していない場合、もしくは、Pcの行より上側に

ネットを延長しているが〔水平配線余裕〕 \geq 〔水平配線要求〕を満たしている場合。

上記の場合以外はどちらにも延長しない。

(b) TypeVのネット

直線 $y = x$ に対称にTypeIVと同様の処理を行う。

(c) TypeVIのネット

次のような場合に上に延長する。

- ・ P_c の行より上側に端子があり P_c と同じ行で左側の格子点の中にそのネットを上側に延長する配線(サブネット)が存在しないときに、 P_c の列より右側に端子がない場合、もしくは、 P_c の列より右側に端子があるが〔垂直配線余裕〕 \geq 〔垂直配線要求〕を満たしている場合。

さらに上述のTypeIVのネットを右に延長するのと同じ条件が成り立つ場合には、このネットを右に延長する。

上記以外の場合どちらにも延長しない。

Case3. P_1 からのみ P_c に向かってネットが延長されている場合

ネットのタイプに基づいて以下のいずれかの操作を行う。

(a) TypeIV, VIのネット

Case2 (a), (c)の条件に付け加え、さらに次のような場合に右に延長する。

- ・ P_c と同じ列で下側の格子点の中に同じネットを右側に延長する配線が存在する場合

(b) TypeVのネット

Case2と同様の処理を行う。

Case4. P_b からのみ P_c に向かってネットが延長されている場合

直線 $y = x$ に対称にCase3と同様の処理を行う。

Case5. どちらからもネットが延長されていない場合
次の格子点の処理に移る。

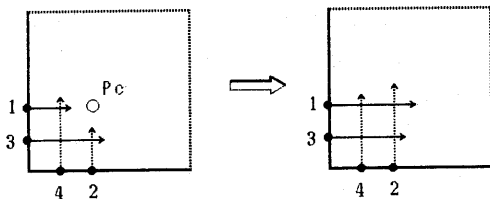


図8 格子点 P_c における操作

Step4. TypeVIIのネットを配線するために、図9のように仮想端子 S 、 T を設け、 S T を結ぶ最大本数の節点素なパスを求める。もし、この本数がネット数より大きければ配線を行う。そうでなければ、可能な範囲で配線を行う。

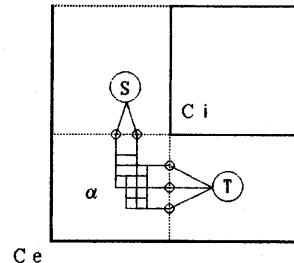


図9 TypeVIのネットの配線

Step5. 以上のようにして配線ができた場合にはStep6へ。配線できなかった場合はinternal boundaryを必要だけ移動させることにより領域を拡大する(図10)。拡大後、カットライン上に再び仮端子を設け、拡大した配線領域(拡大した部分のみ)を対象に本節の算法を適用する。この操作をすべてのネットが配線されるまで繰り返す。

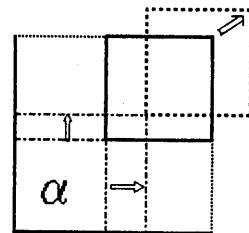


図10 領域 α の拡大

Step6 (後処理). 領域 α 内の配線終了後、領域 β および領域 γ の垂直(水平)制約条件グラフを作成する。このとき、カットライン上の仮端子列は考慮せずに作成する。次に、垂直(水平)制約条件グラフ上においてカットライン上の端子の親子関係を探索し、これらの関係グラフを作成する(図11)。この関係グラフがサイクルを持つ場合、サイクルを解消するような端子順序を求め、領域 α 内でこの順序で配線可能な場合には再配線を行う。

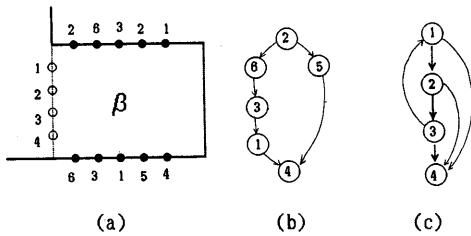


図11(b)カットライン上の仮端子を除いた領域βにおける垂直制約条件グラフと(c)カットライン上の仮端子の関係グラフ

以上で領域αの配線は終了する。ここで重要なことは、以上の操作によって領域α内のネットが、領域は拡大する可能性はあるが、必ず結線されることである。

5. 領域β, γの配線算法

領域βと領域γは、どちらか一方を先に配線し、その後でもう一方の配線を行う。配線順序は、配線が次の基準により困難であると考えられる方から先に行なう。

- ・カットライン上の端子が第4節Step6の処理後も関係グラフ上でサイクルを構成している。
- ・領域αの配線後に定められた配線領域の配線容量と混雑度の差が小さいもの

以下では、領域βを先に配線し、その後で領域γを配線するものとして議論を行う。

Step1 (領域βの配線)。領域βは、上下及び左の3辺に固定端子を持つチャンネルである(領域αから延長されているネットは固定端子となる)。この領域は左端から右のopen sideに向けてコラム毎にGreedy Router[5]を用いて配線する。このとき、設定された領域内で配線できないネットが現われたならば、その時点でinternal boundaryを上方向に移動させ、トラックを一番上に追加し配線が続ける(図12)。

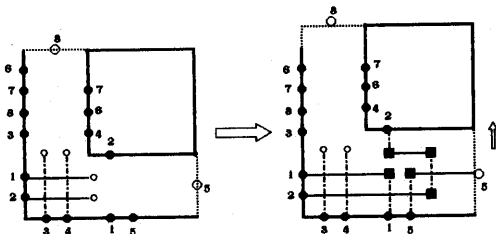


図12 領域βの配線

Step2 (領域γの配線)。領域βの配線終了後新たに設定された領域を領域γとする。この領域も3辺に固定端子を持つチャンネルである。γは下端から上方のopen sideに向けてコラム毎にGreedy Routerを用いて配線する。このとき、設定された領域内で配線できないネットが現われたならば、その時点でinternal boundaryを右上45°方向に移動させ、トラックを一番右に追加し、配線が続ける。移動の際には、処理済みの配線は端子と結線している部分のみを斜め線を用いて延長し、残りの配線は端子位置を更新してから再び垂直水平線分のみを用いて配線を行う(図13)。

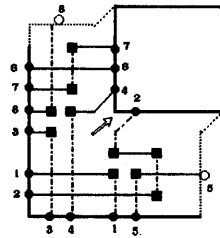


図13 領域γの配線

6. 実験結果

提案した配線算法をVAX11/785、UNIX 4.3 BSD上においてC言語を用いてインプリメントした。図14にある例題(Deutschのdifficult exampleを88カラムの所で折り曲げたL形チャンネル)を配線した結果を示す。このときに費やした時間は約6秒であった。本算法による配線結果は格子構造に基づいたものであるが、格子構造に基づかない文献[1]による算法による結果とほぼ同程度の配線領域で結線を行っている。

7. おわりに

本稿では、L形チャンネル配線問題に焦点をあて、このチャンネルを配線する高速なグリーディ算法を提案した。また、計算機実験結果より本算法によって短時間のうちに良好な配線結果が得られることを示した。今後の課題としては、(1)3層配線モデルへの拡張、(2)グリッドレス化、(3)コンパクト[6]を利用した配線算法などが挙げられる。

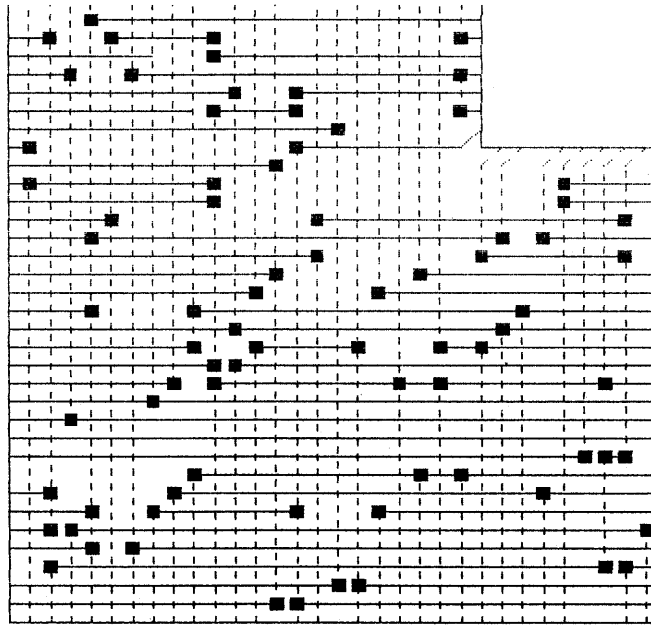


図14 配線例 (一部)

<謝辞>

本研究の一部は、文部省科学研究費補助金：奨励研究 (A)63750329 (昭和63年度)「VLSI設計における計算幾何学の応用に関する研究」の援助のもとに行われたものである。

参考文献

- [1] W.M.Dai, T.Asano and E.S.Kuh, "Routing Region Definition and Ordering Scheme for Building-Block Layout", IEEE Trans. Computer-Aided Design, Vol.CAD-4, No.3, pp.189-197, July 1985.
- [2] H.H.Chen, "Routing L-Shaped Channels in Non-slicing-Structure Placement", in Proc.23th ACM /IEEE Design Automation Conf.pp.152-158,1987.
- [3] M.Sato, "Consideration on L-shaped Channels (1)-(3)", Design Notes on BEAR, UC Berkeley, 1986.
- [4] S.Even and R.E.Tarjan, "Network Flow and Testing Graph Connectivity", SIAM J.Comput.,Vol.4, No.4, 1975.
- [5] R.L.Rivest and C.M.Fiduccia, "A 'Greedy' channel router", Proc. 19th Design Automation Conf., pp.418-424, June 1982.
- [6] 佐藤、中島、金、大附、"レイアウト・システムにおける多機能チャンネル・スペーサ"、信学技報 VLD 88-11, pp.81-87, April 1988.