

ビア数最小化とクロストークを陽に考慮したMCM配線手法

三好 哲也 小出 哲士 若林 真一 吉田 典可

広島大学 工学部

〒724 広島県東広島市鏡山一丁目4番1号

あらまし 近年、プリント基板に代わるICの実装方法としてマルチチップモジュール(MCM)が注目されており、それに伴ってMCM専用の自動配線手法の開発が要請されている。これまでにMCM用の配線手法が幾つか提案されているが、伝搬遅延やノイズの原因となるクロストークが考慮されていなかった。また、MCMの配線では、信号の反射の原因となるビア数を少なくすることが必要である。そこで、本稿ではビア数を4までに制限し、かつクロストークを考慮したMCM配線手法を提案する。提案手法は、ビア数0までの配線、垂直制約を考慮したトラック割り当て手法に基づくビア数2までの配線、線分探索法に基づくビア数4までの配線を順に実行することにより、ビア数の最小化を図る。

和文キーワード MCM, クロストーク, ビア数最小化, トラック割り当て, 線分探索法

An MCM Routing Method for Via Minimization Considering Crosstalk

Tetsuya MIYOSHI, Tetsushi KOIDE,
Shin'ichi WAKABAYASHI and Noriyoshi YOSHIDA

Faculty of Engineering, Hiroshima University

4-1, Kagamiyama 1 chome, Higashi-Hiroshima 724 JAPAN

Abstract In the past several years, Multi-Chip Module (MCM) has been introduced as an alternative packaging approach for Printed Circuit Board (PCB). But conventional routing methods of PCBs are inadequate to solve the MCM routing problem, because they do not take account of MCM technology. Some routing methods for MCM have also been proposed. However, they do not take account of the crosstalk and vias that cause the reflection of a signal or the timing delay. Thus, in this paper we present an MCM routing algorithm for via minimization considering crosstalk. Our method performs no-via-routing and 2-via-routing based on the track assignment considering vertical constraints, and 4-via-routing based on the line search algorithm.

英文 key words MCM, crosstalk, via minimization, track assignment, line search algorithm

1 まえがき

近年、VLSIの製造技術の進歩に伴って、より小型化、軽量化、高性能化された電子システムの開発が望まれている。従来より電子システムの実装手法としてプリント基板(PCB)が用いられているが、チップを実装したパッケージの占有面積が大きくなり、システムの小型化、及び信号の伝搬遅延を小さくすることに限界があった。そこで、PCBに代わる実装技術として、マルチチップモジュール(MCM)が注目されている。これは、高密度配線基板上に直接IC(ベアチップ)や部品を実装して1つのパッケージにする技術であり、従来のPCBに比べ、基板面積、配線長を大幅に改善することができる。また、配線間隔がシリコンオンシリコンモデルでは、 $10 \sim 30 \mu\text{m}$ 、thin filmモデルで $35 \sim 75 \mu\text{m}$ であり、高密度な配線も可能である。現在、MCMの規模はICの数 $4 \sim 50$ 、IC1個当たりのピン数 $80 \sim 400$ であり、近い将来ICの数 $100 \sim 150$ 個、IC1個当たりのピン数 $500 \sim 1000$ になると予想される[6]。よって、本稿では、このような大規模MCMに対する配線問題について考える。従来より、IC、PCBに対する配線手法が数多く提案されているが、以下のような理由により、MCMにそのまま適用することは不適當である。

- ICのように配線層数が多く、またあらかじめ決まっていない。
- PCBに比べピッチ間隔が小さく配線が密である。
- ハイパフォーマンスな回路を設計するためには、配線を損失のある伝送線路としてモデル化し、信号の反射やクロストークを考慮しなければならない。

MCM配線問題に対する従来法としては、次のような手法が挙げられる。3次元迷路法を基にした花房らの手法[7]、x-yレイヤペアに対し、層割り当てを行った後、各レイヤペアごとに2層配線を行なうHoらの手法[8]、ラバーバンドモデルを用いたDaiらの手法[5]などがある。さらに、最近の研究成果としてKhoorらの手法[9]がある。これは、配線領域を左から右へスweepしながら、2部グラフのマッチングを用いて端子のトラック割り当てを行った後、k-cofamilyを求めるアルゴリズムを用いて、垂直方向のインターバルのトラック割り当てを行うものである。しかし、これらの手法は信号の伝搬遅延やノイズの原因となるクロストークの影響を考慮していない。クロストークを考慮した配線手法としては、Choらの手法[4]やChenらの手法[2]がある。Choらの手法は配線領域を幾つかのチャンネルに分割し、概略配線を行った後、ネット間の平行配線長をを求める。これによりクロストーク制約を満たす配線間隔を求め、チャンネルルータGlitterを用いて配線を行う。しかし、この手法では、信号の伝搬遅延や反射の原因となるビア数について何も考慮して

ない。また、2本のネット間の平行配線長を制約することによりクロストークを制約することができない。一方、Chenらの手法は、端子再分散手法で用いたodd-evenアルゴリズム[3]を拡張した手法であり、2次元格子を用いて配線を行うことを前提としているので、使用記憶容量と計算時間の点で大規模MCMには適用できない。

そこで、本稿では、大規模MCMに適用でき、かつビア数とクロストークを陽に考慮した配線手法を提案する。ビア数は2端子ネット1本あたり最大4個までとし、クロストークは2本のネット間の最大平行配線長によって制約する。

以下では2.で配線モデル及び問題の定式化について述べ、3.で配線手法を提案する。4.で実験結果、5.でまとめと今後の課題について述べる。

2 ビア数とクロストークを陽に考慮した配線問題

本稿で考えるMCMモデルはMCM-D(thin-film)[10]である。このモデルでは最上位層にベアチップを実装し、以下の層には、信号配線層、電源/グラウンド配線層があるものとする(図1参照)。配線は信号配線層で行うものとし、層間の接続にはビアを用いる。障害物としては、電源/グラウンド配線のためのスタックドビアなどがある。また、信号配線層は、2層ずつ組(レイヤペア)にしたHVモデル、配線領域には仮想グリッドを仮定する。

一般にMCMは高周波数回路の設計に用いられるものであり、よってMCMの配線問題では回路のパフォーマンスを考慮することが重要となる。正確なモデルでは熱特性や電気的特性をすべて考慮しなければならないが、本稿では、その中で、特にビア数とクロストークに注目する。ビアは信号の伝搬遅延及び反射の原因となるので、できるだけ少ない方が良く、クロストークノイズによる影響もできるだけ小さくしたい。ここで、クロストークとは平行に配線された2本の配線間に生じる雑音のことであり、どちらか一方の信号の立ち上がり、または立ち下がりによって、もう一方の信号線に誘導される電圧である。信号の進行方向と逆方向に伝搬していくバッククロストークと同方向に伝搬していくフォワードクロストークがあり、バッククロストークの大きさは相手側の信号

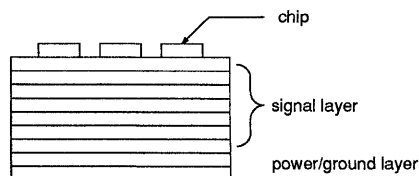


図1: MCMモデル

線の入力電圧に比例し、フォワードクロストークは平行配線長と信号の立ち上がり時間に比例する [12].

入力電圧、信号の立ち上がり時間は配線によって変化させることができないので、隣接した2つのネット間の最大許容平行配線長によってクロストークを制約する。クロストークを制約する別の方法として、配線間隔を広げる、グラウンドとの間隔を小さくするなどが考えられるが、グラウンドとの間隔は特性インピーダンスに関係し、自由に小さくすることはできない。また、今後さらに高密度化が進むことを考えるならば、配線間隔を広げるとは相当に限定される [11]. 従って、平行配線長によってクロストークを制約することは妥当である。一般に、パフォーマンスを考慮するならば、配線長、ビア数をできるだけ小さくするような配線を考えなくてはならず、生産コストを考慮するならば、配線層数をできるだけ少なくしなければならない。この2つは互いにトレードオフの関係にある。よって、本稿では、ネットの集合、端子の集合、クロストーク制約の集合が与えられたとき、回路のパフォーマンスと生産コストを最適にする各ネットの配線経路を求める問題を考える。問題の定式化を行う前に記号の定義を行う。

l_i : ネット n_i の配線長

v_i : ネット n_i のビア数

k : 配線層数

pl_{ij} : ネット n_i と ネット n_j 間の平行配線長

N_{CPL} : 平行配線長に制約のあるネットの集合

CPL_i : ネット n_i の最大平行配線長

クロストークを考慮した多層 MCM 配線問題 CMRP (Crosstalk driven MCM Routing Problem) を以下のように定式化する。

[問題 CMRP]

入力: ネットリスト N , 端子位置の集合 T , クロストーク制約の集合 CPL

出力: 配線経路の集合 P

目的関数:

$$\alpha k + \beta \sum_{n_i \in N} l_i + \gamma \sum_{n_i \in N} v_i \rightarrow \text{最小化}$$

(α, β, γ は定数)

制約:

$$\max_{n_j \in N} \{pl_{ij}\} \leq CPL_i \quad (n_i \in N_{CPL})$$

3 アルゴリズム

問題 CMRP に対し、次のようなヒューリスティックアルゴリズムを提案する。あらかじめ、配線層を2層ずつ

のペアとし、それぞれをレイヤペアと呼ぶ。各ネットに対し、最小木アルゴリズムを用いて2端子分解を行った後、レイヤペア毎にクロストーク制約を満たすように2層配線を行う（以降ではネットはすべて2端子ネットとする）。ここで配線できなかったネットについては、続くレイヤペアで配線することとする。この操作を全てのネットが配線されるまで続ける。各レイヤペアの2層配線では、ビア数の少ない配線または優先度の高いネットから順に配線する。本手法では、ビア数を4までに制限した手法を提案する。この手法はビア数0の配線、ビア数2までの配線、ビア数4までの配線の3段階からなる。クロストークとビア数を陽に考慮したMCM配線アルゴリズム V4CMRA (Via-four Crosstalk driven MCM Routing Algorithm) を以下に示す。

[アルゴリズム V4CMRA]

Step1 各ネットの2端子分解

Step2 各ネットの優先度を決定

Step3 ビア数0の2層配線

Step4 ビア数2までの2層配線 (V2TA)

Step 4.1 水平 (垂直) トラック割り当て

Step 4.2 垂直 (水平) トラック割り当て

Step5 ビア数4までの2層配線 (V4LS)

Step6 全てのネットが配線できれば終了、そうでなければ、水平方向と垂直方向の配線順序を入れ替え、次のレイヤペアに処理を移し Step2 へ。

3.1 ビア数0の配線

ネットの優先度の高いものから順に配線する。配線パターンは各ネットに対して一意であるので、配線可能かどうかを決定していくだけで配線を行うことができる。

3.2 ビア数2までの配線

ここでは、垂直制約を考慮しながら水平トラック割り当てと垂直トラック割り当てを行う。水平トラック割り当てには配線領域の下から上へトラック割り当てを行う上方向へのトラック割り当てと、上から下へ順に割り当てる下方向へのトラック割り当てがある。垂直方向のトラック割り当ても同様に右方向へのトラック割り当てと左方向へのトラック割り当てがある。合計4方向のトラック割り当てがあるが、どの割り当ても等価なので、ここでは上方向へのトラック割り当てのみ説明する。アルゴリズムを説明する前にここで用いるいくつかの用語の定義を行う。

$x(p_i), y(p_i)$: ネット n_i の端子を p_i とし、その x 座標、 y 座標をそれぞれ $x(p_i), y(p_i)$ とする。

障害物: 他のネットの端子、他のネットのビア、及び入力で与えられた障害物を合わせて障害物と呼ぶ。

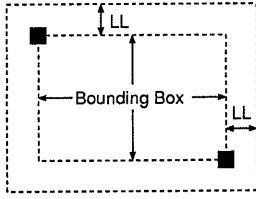


図 2: 配線長の制約

$up(p_i)$, $down(p_i)$, $left(p_i)$, $right(p_i)$: 端子 p_i の上隣の障害物の y 座標を $up(p_i)$, 下隣の障害物の y 座標を $down(p_i)$, 左隣の障害物の x 座標を $left(p_i)$, 右隣の障害物の x 座標を $right(p_i)$ とする。

トラック t で処理中: ネット n_i の端子を p_1, p_2 とし, $y(p_1) \leq y(p_2)$ であるとする。ネット n_i が未配線であり, $y(p_1) - LL \leq t \leq y(p_2) + LL$ であるならば, ネット n_i はトラック t において処理中であるという。(LL は 2 つの端子を囲む最小矩形を越えて配線できる長さであるとする (図 2).) 例えば, 図 3 ではネット 1, 2, 5 はトラック t で処理中であるが, ネット 3, 4 は処理中でない。

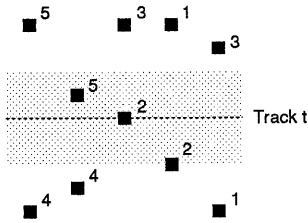


図 3: トラック t で処理中

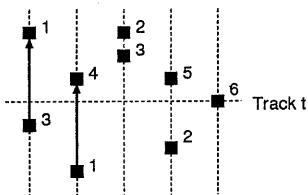


図 4: 垂直制約の例

トラック t に引き出し可能: ネット n_i がトラック t で処理中であり, かつネット n_i の端子 p_i に対して, $y(p_i) \leq t < up(p_i)$, または $down(p_i) < t \leq y(p_i)$ ならば, 端子 p_i はトラック t に引き出し可能であるという。

垂直制約: ネット n_i の端子を p_1, p_2 ($y(p_1) < y(p_2)$) とし, ネット n_j の端子を p_j とする。次の 3 つの条件が成り立つ時, ネット n_i と n_j はトラック t において垂直制約があるという, n_j は n_i の後に配線しなければ

ならない。(1) $y(p_1) < y(p_j) = up(p_1)$, または $y(p_2) < y(p_j) = up(p_2)$ であるとする。(2) ネット n_i がトラック t で処理中である。(3) $up(p_1) > down(p_2)$ である。例えば, 図 4 では, ネット 3 が ネット 1 を, ネット 1 が ネット 4 を制約している。ネット 2 は (3) の条件を満たさないのでネット 2 と ネット 5 の間には垂直制約はない。

インターバル: トラック t 上の 2 つの x 座標 x_1, x_2 によってインターバルを定義し, それを $int(x_1, x_2, t)$ と表す。

トラック t に割り当て可能: 2 端子分解後のネット n_i の端子を p_1, p_2 とする。 p_1, p_2 がともにトラック t に引き出し可能であり, トラック t 上のインターバル $int(x(p_1), x(p_2), t)$ 上に障害物が存在しない場合, ネット n_i はトラック t 上に割り当て可能であるという。

ビア数 2 までの配線は垂直制約を考慮したトラック割り当て手法により経路を決定する。まず, トラック t に対し, 引き出し可能な端子の集合を求め, 2 つの端子が引き出し可能になったネットの集合を求める。次に, 求めたネットがトラック t に割り当て可能であるかどうか調べ, 割り当て可能なネットの集合 (インターバル) を $C(t)$ を求める。さらに Left Edge アルゴリズム [10] を用いて, $C(t)$ の中から実際に割り当てるネットを決定する。クロストークは Left Edge アルゴリズムで割り当てるネットを決定する直前に制約を満たしているか調べ, もし, 満たしていれば, そのネットを割り当て, そうでなければ, 次の候補に対して同様の操作を行う。以上の処理を全てのトラックの割り当てが終わるまで繰り返す。このアルゴリズムをアルゴリズム V2TA (Via-two Track Assignment) と呼ぶ。詳細を以下に示す。

[アルゴリズム V2TA]

- Step 1 トラック $t = 0$
- Step 2 トラック t に引き出し可能な端子の集合 $D(t)$ を求める。
- Step 3 垂直制約を考慮しながら, $D(t)$ より割り当て可能なネットの集合 $C(t)$ を求める。
- Step 4 Left Edge アルゴリズムを用いて $C(t)$ の中から実際に割り当てられるネットの集合を求める。この時, 割り当てようとするネットに対してクロストーク制約を満たしているかどうか調べながらトラック割り当てを行う。
- Step 5 $t = G$ (配線領域のグリッド数), または, 全てのネットの配線が完了していれば終了。そうでなければ, Step 6 へ。
- Step 6 $t = t + 1$ として Step 2 へ。

実際に Left Edge アルゴリズムを用いてトラック割り当てをしている様子を図 5 に示す。現在トラック t のトラック割り当てを行っているものとする。ネット 1 はトラック $t-1$ で、ネット 2 はトラック t で先に割り当てられたとする。今、ネット 3 を割り当ててことを考え、もし、ネット 1、ネット 2 に対してクロストーク制約を満たしているならば、ネット 3 を割り当てて。満たしていないならば、次の候補であるネット 4 についてクロストーク制約を満たしているか調べる。

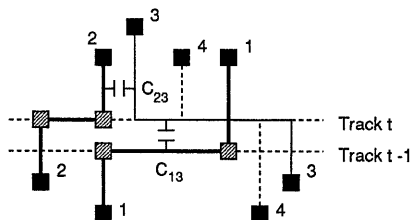


図 5: Left Edge におけるクロストークの考慮

3.3 ビア数 4 までの配線

アルゴリズム V4CMRA では、ビア数 2 までの配線の後、残されたネットに対してビア数 4 までの配線を行う。本稿では、Step 4 の配線手法として、線分探索法 [10] を基にした配線手法を提案する。ネットの配線順序は優先度によって決定する。優先度は制約の厳しさと仮想配線長から決定する。これは制約の厳しいネットを後で配線しようとする、配線領域のほとんどを探索するため多くの計算時間を必要とするからである。アルゴリズムの説明をする前に幾つかの定義を行う。

レベル i のエスケープライン：レベル $i-1$ のエスケープラインから出された垂直線分の集合である。但し、レベル 1 のエスケープラインはソース s 、あるいはターゲット t を通過する垂直・水平線分である。

SS^i, TS^i ：ソース側のレベル i のエスケープラインの集合を SS^i 、ターゲット側のレベル i のエスケープラインの集合を TS^i とする。それぞれの要素をそれぞれ、 ss_j^i, ts_j^i と表す。

交点に対応するパス： SS^3 と TS^2 との交点と SS^3 を発生した ss_j^3 から決定されるソース s からターゲット t へのパス。

まず、2 端子分解されたネット n_i に対して、ソース s とターゲット t を決定する。ソース s とターゲット t からクロストーク制約を満たす範囲でレベル 1 のエスケープライン SS^1, TS^1 を発生させる。次に、 SS^1, TS^1 からクロストーク制約を満たす範囲でレベル 2 のエスケープライン SS^2, TS^2 を発生させる。さらに、 SS^2 からクロストーク

制約を満たす範囲でレベル 3 のエスケープライン SS^3 を発生させた後、 SS^3 と TS^2 の交点を求め、その交点に対応するパスに沿ってクロストーク制約を満たしているかどうか調べる。もし、制約を満たしていれば、配線経路を決定する。これを未配線の全てのネットに対して繰り返す。アルゴリズム V4LS (Via-four Line Search algorithm) の詳細を以下に示す。

[アルゴリズム V4LS]

- Step 1 未配線ネットを優先度によって整列する。 $i = 1$
- Step 2 これから処理を行うネット n_i に対して、ソース s とターゲット t を決定する。
- Step 3 クロストーク制約を満たす範囲でソース側からレベル 1 のエスケープライン SS^1 を発生させる。交点が見つければ、Step 10 へ。
- Step 4 クロストーク制約を満たす範囲でターゲット側からレベル 1 のエスケープライン TS^1 を発生させる。交点が見つければ、Step 10 へ。
- Step 5 クロストーク制約を満たす範囲でソース側からレベル 2 のエスケープライン SS^2 を発生させる。交点が見つければ、Step 10 へ。
- Step 6 SS^2 をターゲットの距離について増大順に整列する。
- Step 7 クロストーク制約を満たす範囲でターゲット側からレベル 2 のエスケープライン TS^2 を発生させる。
- Step 8 $j = 1$
- Step 9 $ss_j^2 \in SS^2$ に対して、レベル 3 のエスケープライン SS^3 を発生させ、 TS^2 との交点の集合を求める。
- Step 10 各交点に対応するパスに対し、クロストーク制約を満たしているかどうか調べる。制約を満たしたパスが見つければ、Step 11 へ。見つからなければ、 $j = j + 1$ として Step 9 へ。
- Step 11 すべてのネットに対して処理が終れば終了。そうでなければ、 $i = i + 1$ として Step 2 へ。

線分探索法を基にしたビア数 4 までの配線手法を実行例を図 6 に示す。図 6(a) はネット n_i に対し、ソース s とターゲット t を決定し、レベル 1 のエスケープライン SS^1, TS^1 を発生させたところである。図 6(b) でレベル 2 のエスケープライン SS^2, TS^2 を発生させている。さらに、図 6(c) でレベル 3 のエスケープライン SS^3 を発生させる。そして、 SS^3 と TS^2 の交点を求める。最後に図 6(d) で交点に対応するパスに対してクロストーク制約を満たしているかどうか、もう一度調べ、制約を満たしていれば、ネット n_i に対する経路が求まったことになる。クロストーク制約については、各レベルのエスケープラインを発生させるときに調べる。しかし、1 つのネットが 2 つの線分と平行に配線されていることがあるので、最後に

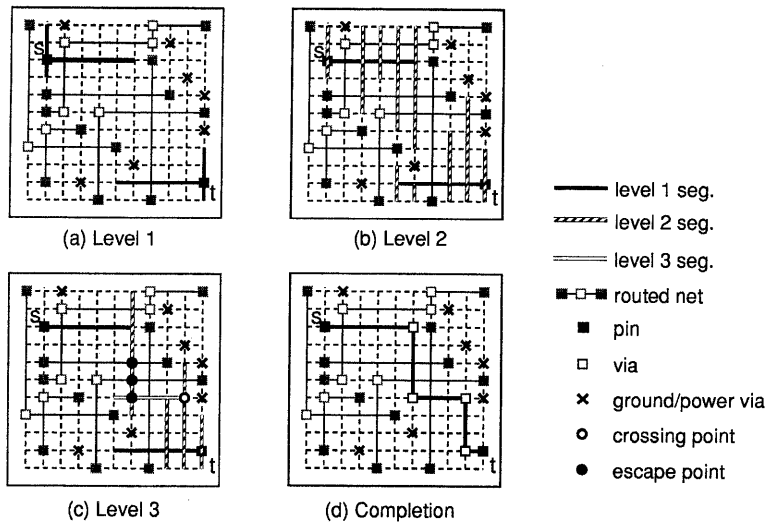


図 6: ビア数 4 までの配線

もう一度クロストーク制約を満たしているかどうか調べている。

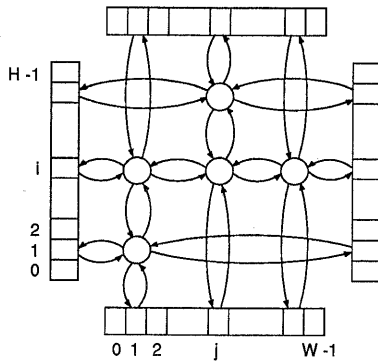


図 7: 双方向リストを用いたデータ構造

3.4 データ構造

提案手法を計算機上で実現するために、端子、ネットの情報以外に以下に示す 2 種類のデータ構造を用いる。

3.4.1 隣接した障害物

隣接した障害物を求めるために、図 7 に示すような 2 次元の双方向リストを用いる。各セルは端子、ビア、障害物に対応し、各セルは 4 方向へのポイント、4 方向の配線の有無、端子情報へのポイントを持つ。このデータ構造により、各端子に隣接した端子、ビア、及び障害物が $O(1)$ の計算時間で分かる。 $L = \max\{H, W\}$ (H は配線領域の高さ、 W は配線領域の幅を示す) とし、 n をネット数

とすると、ビア数は最大 4 なので空間複雑度は $O(L+n)$ である。

3.4.2 平行に配線したネット

1. で示したデータ構造だけでは、平行に配線されたネットを調べることは難しい。提案手法は端子再分散層 [3, 8] を仮定していないので、1トラックに端子が集中していることが多い。よって、1トラックあたりの端子、及びビア数は $O(n)$ であり、この時の計算複雑度は $O(n)$ である。そこで、図 8 に示すように各トラックにおける水平線分をヒープ探索木 [1] を用いたデータ構造によって表現する。例えば、水平線分の両端の x 座標をヒープ探索木のキーとすると、任意の区間 (x_1, x_2) と共通部分をもつ線分を $O(k + \log n)$ で列挙することができる。 k は列挙される線分の数であり、 n はネット数 (ヒープ探索木の要素の数) である。このデータ構造の空間複雑度は $O(L+n)$ である。さらに、このデータ構造に長さ 0 のインターバルとして端子を挿入し、1. で示したデータ構造と組み合わせることにより任意の点から、配線可能な区間を求めることも可能である。

3.5 計算複雑度

先へのべたように、 n はネット数、 $L = \max\{H, W\}$ とする。ビア数 0 までの配線は優先度の順に配線するだけなので $O(n)$ ができる。ビア数 2 までの配線において、1トラックあたりの端子、ビア、及び障害物の数は $O(n)$ とする。トラック t に端子を引き出す操作は $O(L)$ 、割り当て可能なネットを決定する操作は $O(n)$ ができる。Left Edge アルゴリズムが $O(n \log n)$ 、この時、1回のクロス

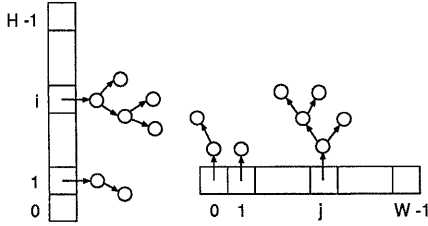


図 8: ヒープ探索木を用いたデータ構造

トークチェックが $O(\log n)$ である。よって、1トラックあたりの計算複雑度は $O(L + n \log n)$ である。さらに、これらを L 回繰り返すので、ビア数 2 までの配線における計算複雑度は $O(L^2 + Ln \log n)$ である。ビア数 4 までの配線において、端子を囲む最小矩形の 1 辺の長さは $O(L)$ であるとする。レベル 1 のエスケープラインを発生する際、クロストークチェックに $O(\log n)$ かかる。レベル 2 のエスケープラインを発生する際、矩形の 4 辺に対して線分を発生させるので、 $O(L \log n)$ かかる。レベル 3 のエスケープラインを発生する際、レベル 2 のエスケープラインに対してそれぞれレベル 3 のエスケープラインを発生させるので、 $O(L^2 \log n)$ かかる。よって、ネット 1 本あたりの計算複雑度は $O(L^2 \log n)$ である。これをネットの数だけ繰り返すので、 $O(L^2 n \log n)$ である。したがって、1 層あたりの計算複雑度は $O(L^2 n \log n)$ となり、これを k 層配線するので、アルゴリズム V4CMRA の計算複雑度は $O(kL^2 n \log n)$ となる。

3.6 アルゴリズムの変形

以上で述べたアルゴリズム V4CMRA を基に、さらにその変形として、ネットの優先度によって引き剥がし再配線を行うアルゴリズムを提案する。この手法を V4CMRA2 と呼ぶ。まず、ネットの引き剥がし方法について説明する。ネット n_i とネット n_j 間にクロストーク違反がある時、ネット n_i とネット n_j の優先度によって引き剥がすべきネットを決定する。今、ネット n_i を配線しているとする。配線途中に違反があったとしても、ネット n_i の優先度の方が高い限り、ネット n_i の配線を行う。そして、ネット n_i の配線経路が決まった後、ネット n_i との間に違反があるネットを引き剥がす。引き剥がされたネットは次の繰り返しで配線を行う。これを配線されるネットが増えなくなるまで繰り返す。以下にアルゴリズム V4CMRA2 の詳細を示す。配線したネット数を a 、引き剥がされたネット数を b とする。

[アルゴリズム V4CMRA2]

- Step 1 2 端子分解
- Step 2 優先度決定 $r = r' = 0$
- Step 3 ビア数 0 までの配線 $r = r + a - b$
- Step 4 ビア数 2 までの配線 $r = r + a - b$

Step 5 ビア数 4 までの配線 $r = r + a - b$

Step 6 $r > r'$ ならば、 $r' = r$ として、Step 3 へ。でなければ、次の層に処理を移し Step 2 へ。

4 実験

提案アルゴリズムを C 言語を用い、SPARC Station2 (28.5MIPS, 64MB) 上で実現した。入力データとして、表 1 に示すような 2 つのランダムデータと 3 つのベンチマークデータを用いた。ランダムデータは 2 端子ネット、3 端子ネット、4 端子ネットをそれぞれ 70%、20%、10% とし、乱数を用いて発生させた。#net はネット数、#pin は端子数、#net2 はネットの 2 端子分解後のネット数、grid はグリッド数を示す。

表 1: 入力データ

data	#net	#pin	#net2	grid (H×V)
data1	2000	4123	2123	1067×1067
data2	4000	8260	4260	1067×1067
mcc1-75	802	2495	1693	599×599
mcc2-75	7118	14659	7541	2032×2032
mcc2-45	7118	14659	7541	3386×3386

クロストークを考慮した場合と考慮していない場合についてそれぞれ比較実験を行った。クロストークを考慮していない場合には、Khoo らの手法 [9]、本手法における線分探索法を基にした手法 (V4LS) のみを適用した場合 (この手法だけでも適用可能である)、及び提案手法 (V4CMRA) の 3 手法について比較を行った。クロストークを考慮していない場合の実験結果を表 2 に示す。クロストークを考慮した場合については、提案手法である V4CMRA についてのみの実験を行った。クロストーク制約は約半数のネットに対し、仮想配線長の 5 分の 1 を上限として乱数により最大許容平行配線長を決定した。クロストークを考慮した場合の実験結果を表 3 に示す。LengthLB は配線長の下界値であり、次に示すものを用いた。

$$LB = \sum_{n_i \in N} \max\{HP(i), \frac{2}{3}MST(i)\}$$

$HP(i)$ はネット n_i の端子すべてを囲む最小矩形の半周近似であり、 $MST(i)$ はネット n_i の最小生成木より求められた仮想配線長である。#via はビア数の総和であり、ここでは配線途中に挿入されたビア以外に、2 端子分解した後のネットの両端点を、すべてスタックドビアとして加えたものである。

以上の結果より提案手法は Khoo らの手法に比べ、ビア数が最大 4.0%、総配線長で最大 3.0%、計算時間についても大幅な改善することができた。線分探索法のみを用いた手法 (V4LS) に対しても、ビア数が最大 12.3% 減少していることが分かる。クロストークを考慮した場合

表 2: 実験結果 (クロストークを考慮しない場合)

data	method	#layer	length(λ)	length LB(λ)	#via	time(s)
data1	V4LS	4	1480862	1465428	10269	53
	Ours	4	1481678		9276	82
data2	V4LS	4	4360584	4320840	21646	262
	Ours	4	4360500		18984	291
mcc1-75	[K _{hoo} 93]	4	394272	314947	6993	180
	V4LS	4	381408		7437	20
	Ours	4	382290		6812	34
mcc2-75	[K _{hoo} 93]	6	5559479	5362181	36438	3960
	V4LS	8	5434109		37639	438
	Ours	6	5431801		34983	708
mcc2-45	[K _{hoo} 93]	4	9130705	8935372	36473	5820
	V4LS	4	9032821		38304	579
	Ours	4	9033099		36277	888

表 3: 実験結果 (クロストークを考慮した場合)

data	#layer	length(λ)	#via	time(s)
data1	6	1480706	9528	251
data2	6	4359862	19086	1275
mcc1-75	6	381613	6812	49
mcc2-75	8	5433499	35430	1943
mcc2-45	4	9035811	36514	1420

についても、層数はほとんど同じで、かつ実用的な計算時間で解を求めることができた。

5 あとがき

本稿では、クロストーク制約とビア数を陽に考慮した新しい MCM 配線手法を提案した。提案手法はビア数を 4 までに制限し、かつ大規模 MCM にも適用可能である。 n をネット数、 L を配線領域の高さと幅の大きい方とすると、空間複雑度は $O(L+n)$ 、計算複雑度は $O(kL^2n \log n)$ である。実験結果より、クロストークを考慮しない場合において、提案手法 (V4CMRA) は従来法より配線層数は同じで、かつビア数を最大 4.0%、配線長を最大 3.0% 減少させることができた。さらに、最大許容平行配線長によるクロストーク制約を 100% 満たし、かつ実用的な計算時間によってクロストークを考慮することができた。よって、提案手法が有効であると言える。今後の課題としては、隣接した mトラックにおけるクロストークを考慮すること、配線長制約を考慮すること、多端子ネットへの拡張などが挙げられる。

謝辞

実験結果を得るのにあたり御協力いただいた本学学部生加藤昭君に感謝致します。

参考文献

[1] 浅野 哲夫: “計算幾何学”, 浅倉書店 (1990).

[2] H. H. Chen and C. K. Wong: “Wiring and crosstalk avoidance in multi-chip module design”, Proc. IEEE Custom Integrated Circuits Conference, pp. 28.6.1–28.6.4 (1992).

[3] J. D. Cho and M. Sarrafzadeh: “The pin redistribution problem in multichip modules”, Proc. 4th Annual IEEE International ASIC Conference and Exhibit, pp. 9–2.1–9–2.4 (1991).

[4] J.-D. Cho, K.-F. Liao and M. Sarrafzadeh: “Multilayer routing algorithm for high performance MCMs”, Proc. 5th Annual IEEE International ASIC Conference and Exhibit, pp. 226–229 (1992).

[5] W. W.-M. Dai, T. Dayan and D. Staepelaere: “Topological routing in SURF: generating a rubber-band sketch”, Proc. 28th Design Automation Conference, pp. 39–44 (1991).

[6] W. Daum, W. E. Burdick Jr. and R. A. Fillion: “A chips-first multichip module technology”, IEEE Computer, Vol. 26, No. 4, April, pp. 23–29 (1993).

[7] A. Hanafusa, Y. Yamashita and M. Yasuda: “Three-dimensional routing for multilayer ceramic printed circuit boards”, Proc. Int’l Conference on Comput.-Aided Des., pp. 386–389 (1990).

[8] J.-M. Ho, M. Sarrafzadeh, G. Vijayan and C. K. Wong: “Layer assignment for multichip modules”, IEEE Trans. Comput.-Aided Design of Integrated Circuits & Syst., Vol. 9, No. 12, pp. 1272–1277 (1990).

[9] K.-Y. Khoo and J. Cong: “An efficient multilayer MCM router based on four-via routing”, Proc. 30th Design Automation Conference, pp. 590–595 (1993).

[10] N. A. Sherwani: “Algorithms for VLSI Physical Design Automation”, Kluwer Academic Publishers (1993).

[11] 高木 清: “多層プリント配線板製造技術”, 日刊工業新聞社 (1993).

[12] 山崎, 足立, 山崎, 若狭: “電子回路のノイズ技術”, オーム社 (1981).