

## 配線経路探索高速化のためのクリティカルカット削減方法

2K-9

濱利行 江藤 博明

(株)日本アイ・ビー・エム 東京基礎研究所

## 1 はじめに

近年、電子機器の小型軽量化にともなって内蔵されるプリント基板も高密度化の傾向にある。この要求を満たすために、プリント基板用自動配線システムには自由角度による高密度配線の機能が求められる。しかしながら、90度、45度配線に比べて、自由角度による配線では、設計規則を満たした配線経路探索に必要な処理が複雑で重い。本稿では、現在開発中の自由角度自動配線システムで利用した高速化の一手法について報告する。

位相的な配線経路が与えられた時、設計規則（ここでは配線間に必要な間隔をあけること）を満たした物理的な配線が存在するためには、すべてのクリティカルカットが配線の間隔を確保できるだけの長さを持っていれば十分であることが知られている [1]。プリント基板の場合には端子が点に在しているため、1つの端子からの見通しがよく、クリティカルカットの数は端子の2乗に近くなる。しかし、実際には検証する必要のないクリティカルカットが存在する。開発中のシステムでは、配線経路の探索中に通過するクリティカルカットがあれば、その時点で設計規則の検証を行なう方法を探っている。このため、これらの不要なクリティカルカットを前もって除くことにより配線経路探索を高速化できる。以下では、検証の必要のないクリティカルカットの条件と生成時の同定方法について詳しく述べる。

## 2 不要なクリティカルカット

クリティカルカットとは2つの物体間の最短のカットのことであるが、各カット  $C$  に対して次の量を定義する。

- $Cap(C)$ : カットを横切ることの出来る最大配線幅
- $MaxFlow(C)$ : カットを横切る配線幅の上限

この2つの量の間に、

$$MaxFlow(C) \leq Cap(C) \quad (1)$$

A method for reducing the number of critical cuts for improving the performance of routing path search  
Toshiyuki HAMA (hama@trl.ibm.co.jp) Hiroaki ETOH (etoh@trl.ibm.co.jp)  
IBM Research, Tokyo Research Laboratory, 1623-14 Shimotsuruma, Yamato, Kanagawa 242, Japan

という関係があればこのカットは検証する必要がないと分かる。ここで、 $Cap(C)$  は既知なので、 $MaxFlow(C)$  について考える。任意の連続するカットの列で両端がこのカット  $C$  と等しいものを  $C_1, \dots, C_n$  とすると、

$$MaxFlow(C) \leq \sum_{i=1}^n MaxFlow(C_i) \quad (2)$$

であるので、他のカットの  $MaxFlow$  が既知であれば、それによって押えることが出来る。これから、

$$\sum_{i=1}^n MaxFlow(C_i) \leq Cap(C) \quad (3)$$

であれば、そのクリティカルカットは検証の必要がない。

式 (3) を利用するためには、 $MaxFlow$  が既知のカットが必要であるが、次のようにして作ることができる。端子を点とみなして配線領域を三角形分割しているものとして、各々の三角形について考えてみる。辺  $C$  が配線領域の外周になっている場合、この辺を配線が横切ることはないので  $MaxFlow(C) = 0$  である。また、三角形の1辺だけが外周の場合には、クリティカルカットがその対頂点からの垂線になっている場合があり、このクリティカルカット  $C'$  は検証することにすれば、他の2辺の  $MaxFlow$  は、 $Cap(C')$  に等しくなる。どちらの場合も  $MaxFlow$  が決まった辺は、式 (1) を満たしている。さらに、このような式 (1) を満たす辺を含む三角形で式 (2) を利用することにより、他の2辺のどちらかの  $MaxFlow$  をより小さく押えられる可能性がある。このようにして、逐次的に三角形の各辺の  $MaxFlow$  を決定できる。また、その他の任意のカット  $C$  についても、もしそのカットを必ず検証することにすれば、 $MaxFlow(C) = Cap(C)$  と決定できる。

## 3 クリティカルカットの生成

クリティカルカットを見つけるアルゴリズムは、各頂点を中心とした平面走査を用いるのが一般的であるが、クリティカルカットの同定と前節で述べた条件に基づいた要不要の判断を同時に行ないたい。平面走査では、クリティカルカットの同定時に既に走査が済んだ平面内の情報しか利用できない。そこで、平面操作よりもやや効率は落ちるが、配線領域の三角形分割の結果を利用する方法を採用した。

配線領域は、端子も点とみなし、基板、配線禁止領域

の外周を制約枝とする制約付き Delaunay 三角形分割により領域分割する [2]。この時、クリティカルカットは、2つの頂点、あるいは、頂点と制約枝を結ぶ線分となる。配線領域をグラフ（三角形を節とし隣接する三角形の境界を枝とする）と見なすことにより、各節から始めて縦型探索を行ないながら、逐次的にクリティカルカットを同定することが出来る。図 1 は、頂点 S からのクリティカルカットを求めるために右側の三角形 1 から始めて、隣接する三角形 2、3 へ至った様子を示している。三角形 2 で S-T1 間のクリティカルカットが同定され、三角形 3 で S-T2 間のクリティカルカットが同定される。探索範囲を限定するために探索の進行とともに始点 C の頂点からの可視範囲（図の斜線部分）を更新していき、三角形の全体が不可視になった時点で探索を打ち切る。

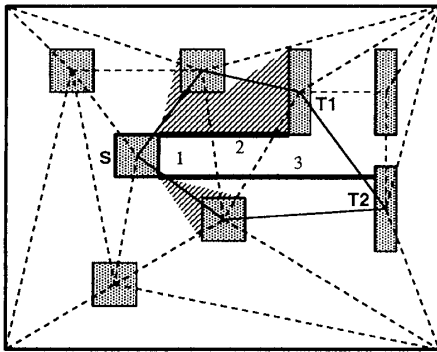


図 1: クリティカルカットの同定

クリティカルカット同定の手順から明らかであるように、クリティカルカットが同定された時点で、そのカットを取り囲む三角形の列が得られている。図 2 に示すように、これら三角形からなる領域の外周を注目しているカットの左右に分け (l1, l2, l3 と r1, r2, r3)、式 (3) により要不要の判定をする。ところが、S から探索を始めたとなると、L, R のクリティカルカットについては、この時点で同定と要不要の判定は済んでいる。必要であると判定されたクリティカルカットが存在する場合を考えてみる。カット L が必要であったとすると、前節で述べたように、 $\text{MaxFlow}(L) = \text{Cap}(L)$  とできる。また、

$$\text{Cap}(L) < \text{MaxFlow}(l1) + \text{MaxFlow}(l2)$$

であったから、 $\text{MaxFlow}(C)$  の上限は、

$$\text{MaxFlow}(l1) + \text{MaxFlow}(l2) + \text{MaxFlow}(l3)$$

よりも、

$$\text{MaxFlow}(L) + \text{MaxFlow}(l3)$$

を用いる方が有利である。このようにして、探索を進めながら左右両側の  $\text{MaxFlow}$  を更新していけば、より厳密な要不要の判定が可能である。

さらに、すべての頂点からの探索を行なえば、一つの

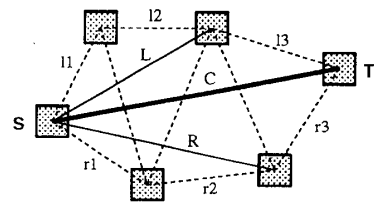


図 2: Maxflow の更新

クリティカルカットは両方向から要不要の判定を行なわれる。クリティカルカットの同定だけが目的ならば両方向からの探索は無駄であるが、不要なカットの削減のためには効果的である。

## 4 評価とまとめ

本方式を実際のプリント基板で実行した結果を表 1 に示す。A、B、C、D はベンチマーク用の比較的小規模な基板であるが、E、F は家電製品に実際に組み込まれている基板のデータを使用している。最低でも検証すべきクリティカルカットの数は 4 分の 1 に減っている。

基板	端子数	削減前	削減後	削減率
A	70	1352	161	0.88
B	83	1077	199	0.82
C	115	1533	376	0.75
D	126	2399	585	0.76
E	363	8266	1781	0.78
F	756	19053	3116	0.84

表 1: クリティカルカット削減率

配線経路探索と同時に設計規則を検証するシステムでは、クリティカルカットの容量の検証は処理の大きな部分を占めているので、全体の高速化にかなり有効であった。

## 参考文献

- [1] C. E. Leiserson and F. M. Maley. Algorithms for routing and testing routability of planar vlsi layouts. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pp. 69-78. ACM, 1985.
- [2] Y. Lu and W. Dai. A numerical stable algorithm for constructing constrained delaunay triangulation and application to multichip module layout. In *Proceedings of 1991 International Conference on Circuits and Systems*, pp. 644-647, June.