

端子回りの接続構造から結線可能性を判定する方法

2K-8

江藤 博明 濱 利行

日本アイ・ビー・エム株式会社 東京基礎研究所

1 はじめに

片面プリント基板の配線は、「それぞれの配線が互いに交差せず、かつすべての配線は与えられた配線領域に収まる」という制約を充足する配線問題である。特にジャンパの使用を許さない場合は、与えられた端子と配線からなるグラフの平面性を判定するNP完全問題となる。しかも配線の組合せからなる探索空間の大きさに比べて許容解の数が非常に少ないため、ヒューリスティックによる探索の絞り込みでは解を求める事は困難である。本稿では、一本の配線について、その近傍の配線との平面性判定を使用した配線可能性のテスト方法を述べる。この手法はバス構造に対して無駄な配線の発生を抑えるため、グローバルルータにおける配線の組合せから生ずる探索空間を効果的に絞り込む。

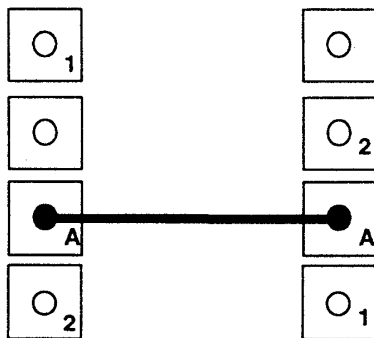


図 1: バスの配線

バス構造の配線問題を図1を例にして説明する。ここで隣り合っている4個の端子の間には配線を通す余裕は無いとする。端子Aを図のように配線すると端子1と端子2を同時に配線することはできなくなる。リップアップルータでは、交差が生じた時に交差した配線のいずれかを引き剥し、再配線を行なう事で交差違反の解消を図る。この交差違反の改善をバスに適用すると再配線したものが隣にある残りのバスに対して交差をするため、バスに関わる全ての配線の交差違反を解消する事は

Testing routability of a wire by checking the structure around terminals
Hiroaki Etoh, Toshiyuki Hama
Tokyo Research Laboratory, IBM Japan

困難である。そのためバス配線はグローバルルータの探索で処理するのではなく、前処理で特別なバス配線プログラムを使用するものもある。

ここで提案するのは、具体的な配線経路(配線パターンと呼ぶ)を作成する事無しに交差競合を見つける手法である。図1を例にすると、端子Aの配線と端子Aから連結(間に配線の余裕のない状態)しているものの集まりは位相的に同型な図2(以降、連結端子表現と呼ぶ)に置き換えられる。図1の端子1のすべての配線パターンは連結端子表現では直線として表現される。逆に図2で交差する線は実際の配線パターンでも交差する。したがって、連結端子表現で配線不可能ならば端子Aの配線パターンは配線不可能である。

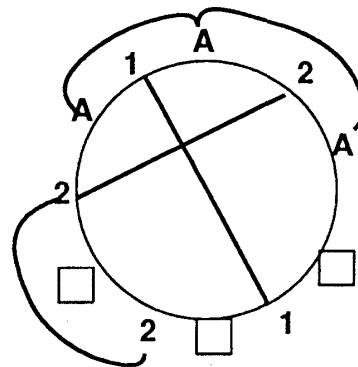


図 2: 連結端子表現例

本方式の利点は次の4点である。(1)バス構造を特別視することなく、位相的な構造で扱うためあらゆるタイプのバス構造に対処可能、(2)グローバルルータにおける無駄な配線候補の排除、(3)配線探索とバスの処理を分離しない事で、配線領域が配線によって埋まることで発生するバス構造に対処可能、(4)配線パターンを作成しない事から交差テストが高速である。節3では交差テストを配列操作に置き換えるアルゴリズムを示す。

多層配線への適用を考えると、各層で平面配線を追及する事でスルーホールの使用を抑えることに利用できる。

2 連結端子表現

連結端子表現とは、ある配線を中心に互いに連結した端子の集まりについて、その周囲を反時計回りに辿る事で得られる端子列のことである。連結した端子とはその端子間に他の配線の通る余地の無いものを言う。配線の通る余地の判断は端子および障害物間のクリティカルカットで(配線容量-配線フロー)と(Min 配線フロー)との比較で決定する。この配線の通る余地の無いクリティカルカットに着目し、連結した端子を次の基準で繰り返し選択する。

基準1 対象クリティカルカットの相手側の端子、

基準2 対象クリティカルカットを通過する配線で接続される端子、

基準3 配線されている場合、配線で接続される端子

クリティカルカットの作成はグローバルルータの前段階で完了しており [1]、かつ不必要なクリティカルカットを削減したため連結端子を求める事は困難ではない。

反時計回りの探索で求められた端子リストは、同一端子の異なる面や既に配線されている端子が複数回出現することがある(図2の参照)。これらは結線済みということで連結端子表現の外側に配線しておく。また連結端子表現で配線する相手のない端子は配線テストの考慮外であるためリストから外して構わない(図2の四角の端子)。

3 接続テスト

接続テストは連結端子表現上で交差しない結線パターンを求めることである。ここで一つの結線は連結端子表現を二つに分割することに注目すると、交差の無い結線パターンは連結端子表現を結線数+1個に分割することが判る。そこで連結端子表現を配列に格納し、結線に対応して配列を分割する事で配線可能性テストは高速に実現できる。

一つの端子に対して結線箇所は複数存在する。なぜならば端子はバスに対して二つの接続箇所を持ち、また配線済みの端子では端子数だけ接続箇所を持つからである。これらの結線箇所は本来一つの端子であるのでいずれかで一つで結線をすればよい。接続テストはいずれかの結線で分割できれば合格なので、すべての結線箇所を試さなければならない。ここではバックトラック探索による接続テストアルゴリズム1を示す。

連結端子構造を求めるには連結端子集合の周囲の端子をリストアップするのだが、連結端子集合では内部に配線用の領域を持つ場合がある。この場合には内部の

配線領域を含めた上で接続テストしなければならない。内部の配線領域でも連結端子構造は作成できる。複数の連結端子構造の接続テストは接続テストアルゴリズムのステップ3の状況に相当する。したがってステップ3以降の処理でこの接続テストは対処される。

接続テスト Algorithm1

端子配列には接続状況を示す属性を付与する。同一の属性値を持つ端子は接続していることを示すものとする。

1. 端子配列から一つ端子 A を選択する。
2. その端子 A に接続する端子候補を作成する。同一の端子番号を持ち、かつ異なる属性値を持つ端子を拾い上げる。
3. 端子候補から一つの端子 B を取り出す。端子候補が空の時には、a) 端子 A が連結端子構造の外側に配線を持つときにはステップ1で次の候補を試す、b) 存在しないならば戻り値として失敗を返す。取り出した端子 B の結線は端子 A の属性値を端子 B に設定することで処理する。
4. 端子 A から端子 B までの時計回りの端子配列と端子 B から端子 A までの時計回りの端子配列を作成する。二つの端子配列が空の場合は戻り値として成功を返す。
5. それぞれの端子配列について接続テストをする。どちらか一方が失敗した時にはステップ3に戻る。それ以外は成功を返す。失敗した時には、端子 B の属性値を以前の値に戻す。

4 簡略テスト

本稿では連結端子構造を求めるのに3つの基準を上げたが、クリティカルカットが求められていないシステムでは端子選択基準1(連結した端子)だけから作成する連結端子構造だけでも図1のような端子が隣り合わないバスについても対処可能である。

配線領域に比較して配線の数が多い場合には、容量の一杯になったクリティカルカットが増加する為、連結端子構造を求めるコストは大きくなる。この場合でも端子選択基準1だけに切り替えるというヒューリスティックでコストを下げられる。

参考文献

- [1] 濱利行, 江藤博明, “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”, 情報処理学会第52大会全国大会論文集, 1996.