

配線折曲げを行うレイアウト
コンパクションの一手法

4R-5

瀬岡 満 安藤 宏
(沖 電 気 工 業 株 式 会 社)

1. はじめに

近年、ビルディングブロック方式のVLSIレイアウト設計用のコンパクタ^[1]が開発されてきている。これらのコンパクタにおいて配線の自動折曲げ処理が高圧縮の一つのキーポイントとなるが、効率的に高圧縮を実現する配線の自動折曲げアルゴリズムは、まだ、確立されたとは言い難い。今回、 x/y どちらか一方に可能な限り詰めることを保証する配線自動折曲げ機能を持つコンパクションアルゴリズムを開発したので報告する。

2. 配線自動折曲げ機能の必要性

配線の自動折曲げ機能は次の理由により有効である。図2.1に示すように、少数の配線がネックとなって他の配線領域には十分空いている領域があるのにそれ以上詰められないとき、配線の折曲げを行うことによりさらに詰められる場合があるからである。

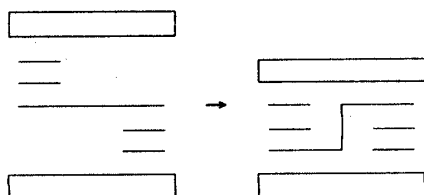


図2.1 配線折曲げによるネックの解消

3. 本アルゴリズムの特徴

従来より、効果的な配線折曲げ(ブロック全体をより詰めるのに役立つ折曲げ)を行うために、レイアウトをグラフで表現しクリティカルパス(コンパクション後のブロックの大きさを決めている配線、下位ブロック等のレイアウト要素群に対応)を求め、このクリティカルパス上にある配線を折曲げる手法^[2]が提案されてきている。この方法では、無駄な折曲げを行わないが、その代り配線の折曲げの都度、クリティカルパスを求める処理を繰返す必要がある。また、クリティカルパスが複数のパスから構成されているときなど、配線を折曲げる処理/折曲げられないと判定する処理は複雑なものとなる。

今回報告するアルゴリズムでは、配線の分割位置を局所的なレイアウト要素の相対的な位置関係のみによって決めて、まずコンパクションを行う。この方法では、不要な折曲げが発生するので、次に、不要な折曲げを除去する処理を行う。配線の折曲げ位置を局所的な情報だけから決めることによりアルゴリズムを簡単にすることが出来る。また、折曲げ位置を隣接するレイアウト要素を移動させるのに最小限必要な配線だけ移動させる位置としている。このことにより、 x/y どちらか一方に可能な限り詰めることを保証する。

4. コンパクションアルゴリズム

コンパクションは、移動させたいレイアウト要素(起点レイアウト要素)とその要求移動量を入力とし要求移動量を超えない範囲で起点レイアウト要素を可能な限り移動させるアルゴリズムをブロック全体に適用して行う。(設計しようとするブロックの外形辺のうち一方を位置固定、他方を起点レイアウト要素、要求移動量を $+\infty$ とする。)

次に、このレイアウト要素移動アルゴリズムについて述べる。まず最初は、配線の折曲げを行わない場合について述べる。ここでは、移動方向は y の正方向とする。(y軸に平行な配線は伸縮自在と考え無視する。)

起点レイアウト要素を要求移動量だけ移動させるには、移動方向に要求移動量だけの空いた領域を見付けていけばよい。これを移動制約関係探索という。移動制約関係とは、あるレイアウト要素を要求移動量だけ移動させるのに、他のレイアウト要素も移動させなければならないとき、これらは移動制約関係にあるという。本アルゴリズムではこの探索をグラフを用いて行う。

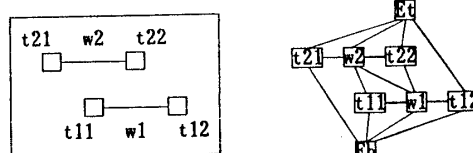


図4.1 レイアウト例とそのグラフ

グラフは、レイアウト要素を頂点、レイアウト要素間の隣接関係及び接続関係を枝としたグラフである。枝には要素間の空いた領域の大きさを重さとして付与する。さらに、枝にはその指すレイアウト要素等のx座標値を付与して順序付ける。

移動制約関係探索は起点レイアウト要素を探索木の根とし、探索木の葉と移動制約関係にある頂点を順に次の葉としていくことにより行う。移動制約関係探索の詳細については文献^[3]を参照のこと。

次に、配線の自動折曲げを行う場合について述べる。基本的には折曲げを行わない場合と同様であり前述の処理に次のふたつの処理を追加することにより行う。

(a) 配線分割処理 (b) 配線整形処理

配線分割処理は移動制約関係探索木を成長させるとき、次に成長させる葉が配線のとき行う。分割位置は親頂点を移動させるのに最小限必要な長さの配線だけ移動させる位置として決める。(図4.2参照) また、分割したときには分割配線と分割位置の上下に隣接するレイアウト要素との間に分割位置のx座標値を付与した枝をはる。

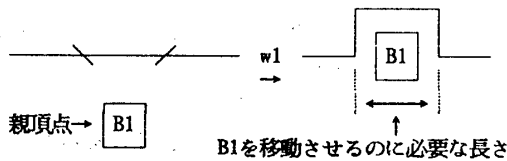


図4.2 配線折曲げ位置の決定

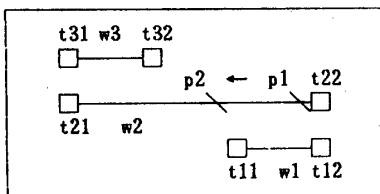


図4.3 配線折曲げ位置のシフト

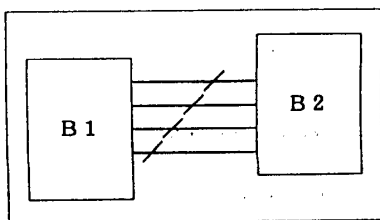


図4.4 下位ブロック間配線の折曲げ

次に、探索時の不要な分割の低減方法について述べる。図4.3のレイアウトにおいて、t11, t12, t22, w1が既探索、t22が親頂点でw2を分割しようとしているときを考える。前述のように分割位置を決めるとp1が分割位置となるがここで分割してもすぐw1によって押し上げられると判定出来る。(分割候補位置の下側にあるレイアウト要素が既探索で、両者間に大きさが1以上の空いた領域がないとき。)無駄と判定したときは分割位置を(この例では左側へ)シフトする。先の候補位置より左側にある枝の指す頂点を親頂点と見なして次の候補位置を決める。ここで、また、折曲げが有効かどうか同じ方法で判定する。このようにしてp2までシフトしてくる。p2では有効と判定されここで分割される。また、この処理を行うことにより、図4.4のように下位ブロックに複数の配線が並んで接続している場合(B1を親頂点としたとき)、図中に示す位置で分割が生じる。

このようにしてコンパクションを行った場合、不要な折曲がりが生じることがあるので配線の整形処理を行う。この配線整形処理は、前述のレイアウト要素移動アルゴリズムを個々の折曲がり配線に適用して行う。(起点レイアウト要素を折曲がり配線、要求移動量を折曲がり量とする。このとき、配線分割処理は行わない。)ただし、各レイアウト要素の探索回数を少なくするため、整形処理は折曲がり量の大きいものから順に行う。何故なら、折曲がり量の大きい部分配線の整形処理の過程で、より折曲がり量の小さい部分配線の整形も行われることがあるからである。

5. おわりに

本稿では、配線の自動折曲げ機能を持つレイアウトコンパクションアルゴリズムについて述べた。本アルゴリズムを既に開発したレイアウトコンパクタ^[3]に組み込み効果を確認した。

通常、コンパクションをしようとするとき、人間にはブロック全体を見てどの辺りを折曲げれば効果的か簡単にわかることが多い。従って全配線について分割位置探索を行わせるのではなく、人間が(個々の分割位置を指定するのではなく)分割位置探索領域をインタラクティブに指定することにより、少ない処理時間で品質のよい結果が得られる。今後、この機能の追加を行う予定である。

【参考文献】

[1] Ishikawa, M. et al., "An Automatic Compaction Method for Building Block LSIs" Proc. ISCAS, pp203-206, 1985
 [2] Hsueh, M. Y. et al., "Computer-Aided Layout of LSI Circuit Building-Blocks" proc. ISCAS, pp. 474-477, 1979
 [3] 瀬岡, 他「VLSIレイアウトシステム(VILLA)のレイアウトコンパクション手法」情処第32回全国大会論文集 pp1959-1960, 1986