Vol. 35 No. 12 情報処理学会論文誌 Dec. 1994

ハイブリッドIC平面レイアウト対話設計における配線位相を保つ端子移動アルゴリズム

村田 洋[†,††] 梶谷洋司[†,†††]

ハイブリッドICやプリント配線基板（PCB）の平面レイアウト対話設計において，端子の連続移動の問題を位相的に処理するレイアウトエディタのプロトタイプを開発した．本文ではその位相配線処理部分について詳しく述べる．配線の位相的経路情報を格納するデータベースとして，部品の配置に応じて領域を三角形分割し，配線はそれが通過する三角形の系列で表現する．そして，端子の連続移動を凸多角形の変形の系列で捉え，それぞれの変形はデータベースの部分変更で対応するワームクリープ法を提案する．従来の端子移動アルゴリズムに比べ，ワームクリープ法は(1)凸多角形端子を扱える，(2)端子を変形できる，(3)移動途中における停止を不要とした，という特長を持つ．

Interactive Terminal Sliding Algorithm for Hybrid IC Planar Layout

Hiroshi Murata[†,††] and Yoji Kajitani[†,†††]

A prototype layout transformation editor for hybrid IC's and printed circuit boards is developed. Its speed is realized by dividing the single terminal continuous sliding problem into the topological phase and the geometrical phase. This paper describes the topological phase algorithm. As for the topological database of the layout, the wiring space is triangulated and each wire topology is represented by a sequence of passing triangles. Then, we propose the terminal sliding algorithm called *Worm Creep Method*. It slides one terminal by a sequence of polygon deformations, and executes each deformation only by the local modification of the database. Worm Creep Method has following advantages to the previous alogrithms. (1) Each terminal can be a convex polygon, as well as a single point. (2) The designer can even deform the terminal shape as well as simply slide it. (3) The algorithm eliminates intermediate stops of the sliding terminal.

1. 緒　言

ハイブリッドICやプリント配線基板（PCB）の平面レイアウト設計の自動化は強い需要にもかかわらず遅れており，人手によるパターンの詰め込み作業が多く残されている．人手設計に匹敵する設計品質（基板寸法，製造コスト，電気的特性）を達成する自動設計ツールの実現は当面困難視されており，とりあえず対話型の設計支援ツールが求められている．人は，『配線を追従させながら部品配置を修正するならば，レイアウトはどのように変化するか』と考え，その変化に基づき彼の価値観に沿ってレイアウトの改善をはかるものである．そこで，部品の移動は設計者が指令し，配線の追従は自動計算に任せる**レイアウト連続変形エディタ**のプロトタイプを開発した．

それは，例えば**図1**のように使用される．レイアウトは，端子（凸多角形）と2つの端子を接続する配線（曲線）からなる平面回路である．設計者は，（仮の）配置配線済の初期レイアウト（図1(a)）を入力し，部分的な修正を繰り返して，最終的なレイアウト（図1(e)）を得る．一回の修正において，設計者は部品（端子の組）の並進（図1(a)→(b)）や回転（図1(c)→(d)）を指令できる．エディタは位相的経路を保って配線を修正する．

エディタの有効性は，修正単位機能と応答の速さに

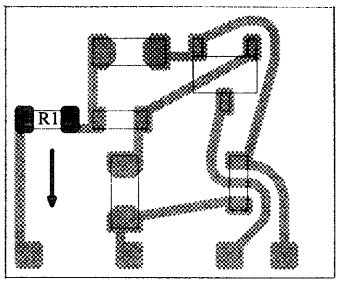
† 北陸先端科学技術大学院大学情報科学研究科
School of Information Science, Japan Advanced Institute of Science and Technology, Hokuriku
†† 株式会社村田製作所ファンクショナル・デバイス事業部
Functional Device Division, Murata MFG. CO., LTD.
††† 東京工業大学工学部電気電子工学科
Department of Electrical and Electronic Engineering, Faculty of Engineering, Tokyo Institute of Technology

かかっている．ここでは，エディタの単位機能として，ひとつの端子移動が指令されたとき，配線の位相的経路を保ち，かつ，あらかじめ指定された配線間隔を満たして配線を修正することを考えている．部品の間隔が狭すぎて配線間隔を保てない場合には，エディタは警告を出すかあるいは適宜間隔を狭めて配線するものとする．

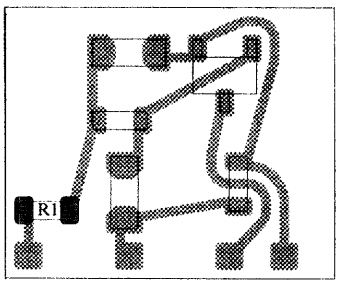
単位機能を実行する初等的なアルゴリズムとして，時間を刻んで配線の変形の過程を計算する方法がまず考えられる．しかし，刻々の配線の曲線を求めるのは，結果的に途中経過であるので無駄な計算であるだけでなく，膨大な座標計算が必要になり，とても実用的なエディタは得られない．高速処理のためにはデータの抽象化が必要である．アナログレイアウトの処理では連続曲線を扱うために膨大な座標データを処理しなければならないように思われるが，曲線そのものはデザインルールを満たすものとして周辺状況から推測できるものであるから常に保持していなければならないものではない．配線の曲線そのものを表す配線データ（以下で物理配線と呼ばれる）を，曲線の位相的経路は指定するが具体的な曲線そのものは未確定とする一種の概略配線データ（以下で位相配線と呼ばれる）に変換して扱えばよい．すると単位機能は3つの処理に分割されることになる．

- 物理配線を位相配線に変換する（物理位相変換）
- 位相配線を端子移動に追従させる（位相配線処理）
- 位相配線を物理配線に変換する（位相物理変換）

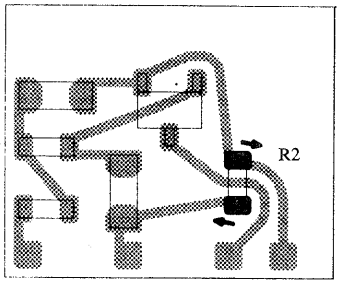
このような処理を位相的アプローチという．これに対し，物理配線を直接修正する物理的アプローチは，先に述べたように，計算量の限界のため柔軟なレイアウト修正が困難である．例えば，物理的アプローチの代表例として，現在広く用いられているCADシステムMagicにおけるplowing機能[1]では，端子移動は水平・垂直な方向に限られ，配線の形状も水平・垂直な直線分からなる折れ線に限られる．一方，上述のような位相的アプローチの例として提案されている，Dai, Kong, Jue, Sato[2]のMulti Chip Module用エディタは，端子を任意方向に移動でき，Valainis, Kaptanoglu, Liu, Suaya[3]のLSIリーフセル用コンパクタ



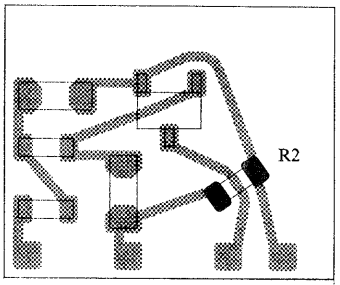
（a） R1の下方向への移動指令
（a） Command of R1 downward slide



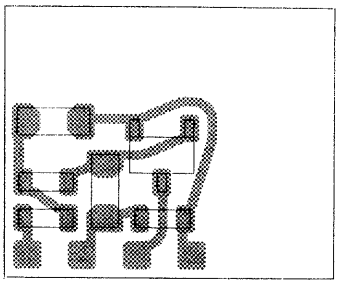
（b） R1の移動と配線追従の結果
（b） Slided R1 and accordingly shifted wires



（c） R2の回転指令
（c） Command of R2 rotation



（d） R2の回転と配線追従の結果
（d） Rotated R2 and accordingly shifted wires



（e） 最終レイアウト
（e） Final layout

図1 レイアウト連続変形エディタの使用例
Fig. 1 An interactive layout design scenario using a layout transformation editor.

は，端子を任意方向に移動でき，かつ，曲線配線を扱うことができる．しかし，高速応答が必要なエディタとして曲線配線を生成する位相的アプローチは複雑になることは避けられず，実現されていない．また，いずれの研究も端子を点と仮定するので，部品を基板に半田づけするために端子が凸多角形であることが本質的に必要であるハイブリッド IC や PCB のレイアウトに適用することはできない．このようにいずれの方法も従来のままでは不満である．

我々のエディタは位相的アプローチの側に立つが，端子形状を点から凸多角形に拡張し，かつ，計算の本質的な簡単化をはかっている．その中核である位相配線処理について報告する．

本文は，次のように構成されている．2章では，エディタの単位機能を定義し，それを物理位相変換，位相配線処理，位相物理変換に分割する手順を述べる．3章では，従来の物理位相変換の手法を，凸多角形端子を扱えるように拡張する．4章が本文の主体であり，端子の移動を，凸多角形の収縮・拡大の2変形に帰着することにより，従来のシステムでは免れ得なかった移動端子の途中停止を必要としない，より簡単で新しい位相配線処理のアルゴリズムを提案する．5章では，開発したエディタのプロトタイプについて報告し，本文の成果をまとめる．位相物理変換については現在研究中であり本文に含まれない．

2. エディタの機能と構成

エディタは回路のレイアウトの一部変更を繰り返すための対話型ツールであり，常にディスプレイに**図2**のようなレイアウトを表示している．回路とレイアウトをきちんと定義すれば次のようになる．

> 端子の名前の一覧と接続すべき端子の名前の対の一覧が回路である．回路のレイアウトとは，平面図形であり，端子は互いに重ならない凸多角形であり，2つの端子を接続する配線は互いに交差しない曲線であり，両者は基板と呼ばれる矩形上に配置されている．ここで，配線が2端子を接続するとは，曲線の両端点をそれぞれの端子凸多角形の外周上の任意の点に置くことをいう．

上の定義は例えば図2に実線で示す図形について述べたものである．（実際のエディタは，実線の図形に一定の幅を付けた図形（図で灰色の部分）を表示する．例えばデザインルールなどは実線上に換算適用されるので，本文ではこの幅については考えない．）

設計者はディスプレイに表示された現在時刻 τ_1 のレイアウトを観察し，ひとつの端子 T を任意の方向へ任意の距離だけ移動する指令 M を，例えばマウスを用いて発行する．エディタは時刻 τ_2 で実行を完了する．**図3**（a）は現在のレイアウトと端子 T の移動指令 M（図中に破線で示す）の例である．端子 T が指令により移動し，レイアウト中のすべての配線が回路としての接続を保ちデザインルールを満たして変形するならば，端子移動が完了した時刻 τ_2 に図（b）のようなレイアウトが得られるであろう．このような手続きを単端子移動として定義する．これは，変形の途中状態の出力は行わないという意味で，レイアウトを離散的に変更する手続きである．しかし，連続的な制約条件を表現するために時間の概念を導入して，時刻 τ におけるレイアウトを $\mathcal{L}(\tau)$ と表記する．

手続き：単端子移動

入力：$\mathcal{L}(\tau_1)$，端子 T の移動指令 M

出力：$\mathcal{L}(\tau_2)$

例外処理：T が基板から逸脱する指令および T が他の端子に衝突する指令（**図4**（a）参照）は不正とする．（画面に理由を表

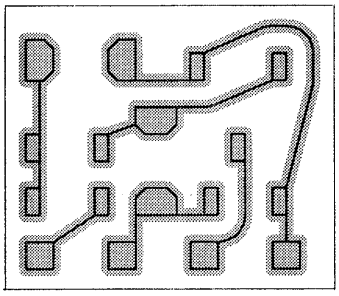
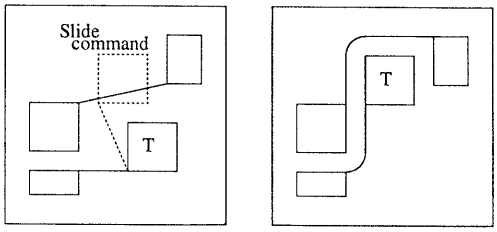


図2 レイアウト
Fig. 2 Layout.



（a） 端子移動前
（a） Before terminal slide

（b） 端子移動後
（b） After terminal slide

図3 単端子移動
Fig. 3 Single-terminal-slide.

示し，指令を捨てる．）

制約 1：$\tau \in [\tau_1, \tau_2]$ に対して，$\mathcal{L}(\tau)$ は連続的に同一回路を表す（連続制約）

制約 2：可能な場合必ず，$\mathcal{L}(\tau_2)$ における配線は所定の配線間隔を保つ（デザインルール制約）

端子移動指令 M の与え方はいろいろ考えられるが，ここでは，図3（a）に破線で示すように，端子 T の移動完了時 τ_2 における端子の凸多角形と，端子 T のひとつの頂点が時間 $[\tau_1, \tau_2]$ に描く直線分を指定する．我々の移動法によれば，この直線分をいわばすり抜けて，端子が移動される．この特徴により，狭い障害物の間をすり抜けたり端子の形状を変更する指令が可能になる．（例を図4（b）に示す．）従って正確には一頂点指定端子移動変形指令と呼ぶべきであろうが，簡単のため**端子移動指令**と呼ぶ．

さて，エディタは概略次のように構成される．設計開始時に設計者が入力するレイアウトにおいて，配線の曲線そのものを表すデータ（物理配線データ）を，配線の位相的経路は指定するが曲線は指定しない配線データ（位相配線データ）に変換する（物理位相変換）．ここで，配線の位相的経路とは，Leiserson, Maley[4] にならい，レイアウトの端子を固定し配線のみを連続制約を保って変形して互いに移りあえる配線曲線をすべて同一視した配線の経路概念である．物理配線のかわりに位相配線を用いたレイアウトを $\dot{\mathcal{L}}$ と書く．このあとに続く単端子移動を2つの手続きに分けて順に実行する．

手続き 1：位相配線処理

入力：$\dot{\mathcal{L}}(\tau_1)$，端子 T の移動指令 M

出力：$\dot{\mathcal{L}}(\tau_2)$

例外処理：T が基板から逸脱する指令および T が他の端子に衝突する指令は不正とする．

制約 1：$\tau \in [\tau_1, \tau_2]$ に対して，$\mathcal{L}(\tau)$ は連続的に同一回路を表す（連続制約）

位相配線処理は，デザインルールは考慮せずに，端子の移動の問題に専念する．従来2つのアルゴリズム[2),5)] が提案されているが，点形状の端子しか移動できず，かつ，非常に複雑である．凸多角形形状の端子を非常に簡単に移動できる新しいアルゴリズムを4章で提案する．従来法との違いは5章で考察する．

手続き 2：位相物理変換

入力：$\dot{\mathcal{L}}(\tau_2)$

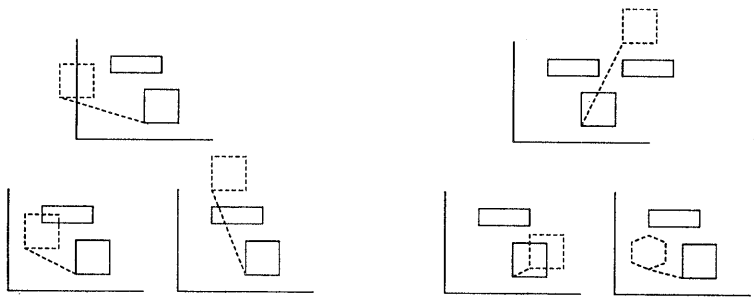
出力：$\mathcal{L}(\tau_2)$

制約 2：可能な場合必ず，$\mathcal{L}(\tau_2)$ における配線は所定の配線間隔を保つ（デザインルール制約）

位相物理変換は，端子は固定されたものとして，デザインルールの問題に専念する．たとえば文献 2), 4) のアルゴリズムにより，設計ルールを満たした垂直水平な線分からなる物理配線を生成できる．また，文献 5) では曲線配線を生成することが試みられている．ただし，端子を点と仮定するので，端子凸多角形の外周上において配線接続位置を最適化することはできない．この点について拡張が必要であるが，本文では扱われず，今後の課題として残される．

3. 物理配線から位相配線への変換

物理配線から積極的に曲線の形状情報を脱落させて，位相的経路のみを表現するデータを得る．以下，位相配線データを2つの段階により導入する．



（a） 不正な指令（基板逸脱および他端子衝突）
（a） Illegal commands (out of the substrate or collision with other terminal)

（b） 正常な指令
（b） Legal commands

図 4 移動指令
Fig. 4 Slide commands.

第1の段階では，端子配置を反映した三角形分割を作成する．レイアウトのすべての端子凸多角形のすべての頂点を与点集合として，基板上の全領域が三角形に分割されるように，点間を交差しない辺で結ぶ．ただし，端子凸多角形の辺は，常に三角形分割の辺として採用する．例えば，図3(a)のレイアウトは，**図5**のように三角形分割される．三角形分割を，ひとつの三角形が指定されたとき，それに辺で接する3つの隣接三角形を返すことができるデータ構造で表す．

制約条件として一部の分割辺があらかじめ指定される三角形分割は，計算幾何学において“制約付三角形分割”と呼ばれている．分割アルゴリズムについては，漸近的には文献7)の分割統治に基づくアルゴリズムが高速である．しかし，ここでは逐次与えられる入力に基づいて，現在得られている三角形分割を部分的に更新する方法を採る．エディタにおける対話的な処理では，平均的な応答時間はそのほうが速いであろうと考えるからである．あらかじめ基板の矩形を対角線で2つに分割した三角形分割を作成しておき，3つの手続き一点挿入，辺挿入，点削除一により部分的更新を行う．点挿入は現在の三角形分割に新たな1点を挿入する手続きであり，指定点を含む三角形を探索し，それを3つに分割する．辺挿入は挿入済の2点間に辺を実現する手続きであり，実現したい辺に交差する辺を削除して，現れた多角形領域を再分割する．点削除は，挿入済の1点を削除する手続きであり，指定点とそれに接続する辺を削除して，現れた多角形領域を再分割する．

三角形分割の辺を，配線が自由に交差できる**自由辺**（図に破線で示す）と，それ以外の**非自由辺**（実線で示す）に分類する．非自由辺は端子凸多角形の辺か対角線のいずれかである．三角形を作る3辺のうち自由辺の数を，その三角形の**自由度**と呼ぶ．自由度1の三角形が存在すれば非凸な端子が存在することになるので，三角形の自由度は 0，2，3 のいずれかである．図5にはいくつかの三角形について自由度が示してある．

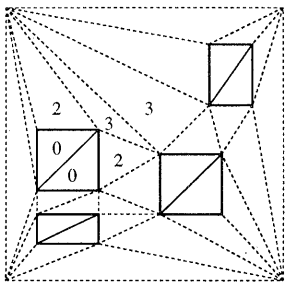


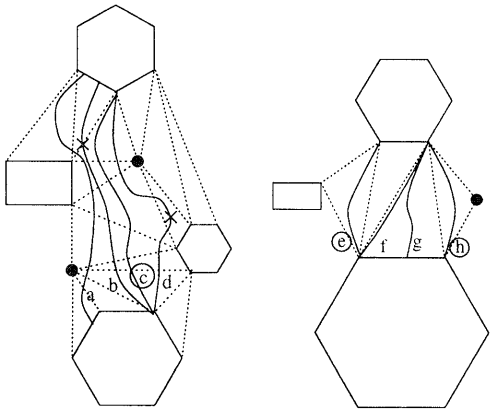
図5 三角形分割
Fig. 5 Triangulation.

第2の段階では，配線が通過する三角形の系列に注目して，位相配線のデータを構成する．配線がひとつの三角形を通過するとは，その三角形の内部を通過することを意味する．（三角形の外周に接するだけであれば通過とはみなされない．）レイアウトに含まれるそれぞれの物理配線 w について，w と位相的経路が等しい曲線のうち，第1の段階で作成した三角形分割上で，自由度3の三角形を少なくともひとつは通過する曲線は必ず存在する．端子凸多角形は重ならないとしているからである．そのような曲線のなかで通過する三角形数が最小である曲線を，配線 w の位相的経路を代表する曲線として選び，w の **slacked rubber band equivalent** (**SRBE**) と呼ぶ．SRBE を，次の参照を可能とするデータ構造で表す．

参照 1：ひとつの SRBE が通過する三角形を列挙する．

参照 2：ひとつの三角形を通過する SRBE を列挙する．

物理配線を上のデータに変換する方法を考察する．**図6**(a)において，a, b, c, d は互いに位相的経路が等しい曲線（**同位相配線**という）である．このうち，c が SRBE であり図中に○印を付してある．実際，c の通過する三角形数は6個と a, b, c, d 中最小であり，自由度3の三角形を（4つ）通過している．ここで，c



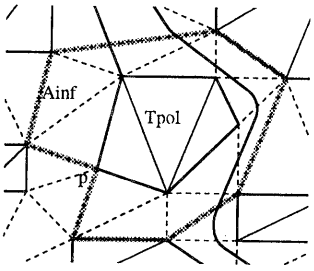
(a) 場合1
(a) Case 1
(b) 場合2
(b) Case 2

図6 同位相配線と SRBE (○印)
Fig. 6 Homotopic equivalent curves and SRBE (○ mark).

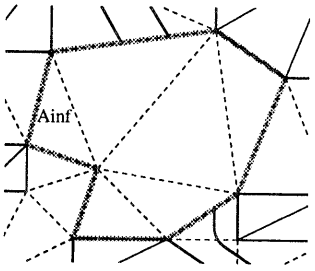
の通過する三角形は a, b, d のいずれもが共通して通過していることに注目する．いいかえれば，c を次第に変形して a, b あるいは d のようにしたときに曲線は新たな三角形を通過するようになる．この新たに通過する三角形は曲線が変形するに伴って単純に増加する．このことは，新たな通過三角形の発生の仕方を2通りに分類して考えれば，容易に理解できる．ひとつは c を d のように変形する場合である．この場合，新たな三角形に侵入してすぐ戻ることになる．もうひとつは c を b のように，端子接続部を端子の辺に沿ってずらす場合である．この場合も多くずらせばそれだけ新たな三角形を通過する．以上の考察により，a, b, c, d のどれが与えられても c の通過する三角形系列を出力することができる，グリーディな最短化アルゴリズムが得られる．入力の同位相配線が通過する三角形の系列から，“無駄な三角形”（c が変形する過程で新たに通過することになった三角形）をすべて取り除けばよい．無駄はその発生に応じて簡単に除去できる．d の無駄は，図中の×印の位置において，通過三角形系列が x, y, x の形の部分列を含んでいることで発見し，x, y, x の中央の y および一方の x を取り除く．b の無駄は，同様に図中の×印の位置において，通過三角形系列中で端から2つの三角形の共有辺，すなわち曲線が最も端子寄りで交差する自由辺が，接続する端子から発していることで発見し，系列端の三角形を取り除く．さて，現在得られている三角形系列の長さが2以上であり，端の三角形の自由度が2であれば，その三角形は b で見た例と同様に必ず除去される．従って，無駄をすべて除いた結果2つ以上の三角形が残った場合には，それはSRBEの通過三角形系列である．しかもそれは同位相配線中唯一である．

グリーディな最短化の結果，ただひとつの三角形が残されたとする．図6（b）のような場合である．この場合は，2端子間を結ぶ自由辺に一致する同位相配線 f が存在する．f の通過三角形系列は空列であり，従って同位相配線中最短かつ唯一であるが，我々のデータ構造では扱いに不便である．そこで，f のような配線は除外すべく，SRBEを定義している．f 以外の e, g, h はいずれも三角形をひとつだけ通過する同位相配線である．このうち e と h は，通過三角形の自由度が3であるから，いずれもSRBEである．g の通過三角形の自由度は2であるから，g はSRBEではない．図から想像できるように，2つの端子が自由辺で結ばれているときには，両端子間には自由度2の三角形が存在し得る（図の例では2個）が，その数は一意には定まらない．しかし，それらを挟む形で，自由度3の三角形がちょうど2個存在する．これがSRBEの定義において自由度3の三角形を選ぶ理由である．仮に，グリーディな最短化の結果得られている三角形が g の通過三角形であれば，自由辺で隣接する三角形を順に探索して，e あるいは h のうちいずれか一方の三角形を発見する．

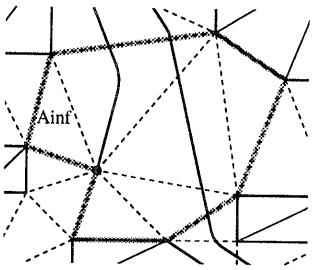
配線の位相的経路を代表する曲線をslacked rubber band equivalentと呼ぶのは，端子を点と仮定したときではあるが，Leisersonら[4]が最短曲線rubber band equivalent (RBE) を導く過程でSRBEをいわば最短化する前のRBEとして捉えており，それにならった



（a） 影響域 A_{inf}
（a） Influential area A_{inf}



（b） 三角形分割の修正
（b） Re-triangulation of A_{inf}



（c） SRBE データの修正
（c） Re-built SRBEs inside A_{inf}

図 7 単端子収縮手続き
Fig. 7 Procedute “single-terminal-shrinkage”.

ものである．また，本文以前に SRBE によって配線の位相的経路を代表させた例として文献 3) がある．ただし，端子をやはり点と仮定している．

4. 端子移動の位相配線処理

端子の連続的な移動を，端子の収縮・拡大に帰着して実行するアルゴリズムを構成する．

4.1 端子の収縮と拡大

凸多角形 T_{pol} である端子をそのひとつの頂点 p に収縮する単端子移動を**単端子収縮**という．逆に，1点 p である端子をその点をひとつの頂点とする凸多角形 T_{pol} に拡大する単端子移動を**単端子拡大**という．単端子収縮あるいは単端子拡大により三角形分割が修正されねばならないことは自明であろう．ここで，三角形分割上で T_{pol} の頂点 p 以外の部分と接触する辺を除いたとき現れる多角形を A_{inf} とおく．**図7**(a)はその例である．

定理 1 単端子収縮・単端子拡大の位相配線処理は，A_{inf} 内を一回だけ三角形再分割するデータ修正により実行できる．

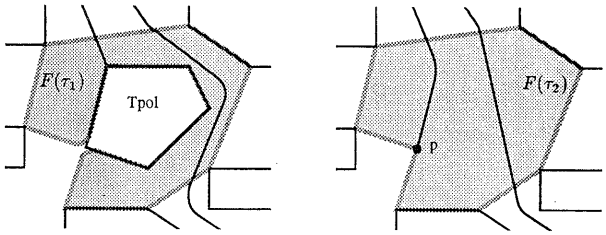
この事実から，A_{inf} を影響域と呼ぶ．収縮の場合について，アルゴリズムを示し，上の定理を証明する．収縮前の端子凸多角形を $T(\tau_1)=T_{pol}$ とし，収縮の目標となる頂点を $T(\tau_2)=p$ とする．図7は説明のための図である．

(**step 1**) A_{inf} 内を三角形再分割することにより，$\tau=\tau_2$ における三角形分割を得る．(図7(b))

(**step 2**) 端子の収縮に伴って，影響域 A_{inf} から端子内部を除いた領域 F (以後，影響自由域) が変形したと考えて，A_{inf} 内について SRBE データを再作成する．(図7(c))

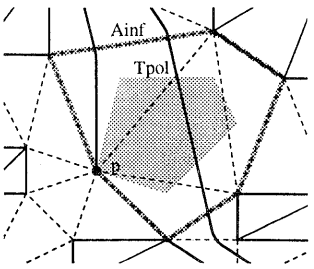
$\tau=\tau_1$ において，レイアウトは三角形分割データと SRBE データにより表されている．このうち三角形分割データについては，step 1 により $\tau=\tau_2$ におけるデータが得られることは容易に理解できる．以下，step 2 により $\tau=\tau_2$ における SRBE データが得られることを示す．

端子 T が $T(\tau_1)=T_{pol}$ から $T(\tau_2)=p$ に連続的に収縮することを，影響自由域 F が $F(\tau_1)$ から $F(\tau_2)$ に連続的に変形することに置き換えて考える．図7の例題では，**図8**(a)に示す $F(\tau_1)$ が，端子を押し潰すように変形して A_{inf} 一杯に広がり，図8(b)に示す $F(\tau_2)$

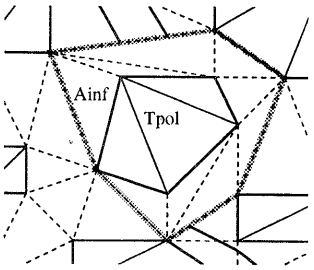


(a) 変形前 (b) 変形後
(a) Before deformation (b) After deformation

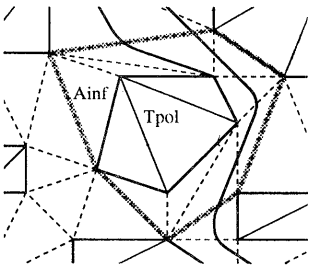
図 8 単端子収縮により変形する自由域
Fig. 8 Free area being deformed by single-terminal-shrinkage.



(a) 影響域 A_{inf}
(a) Influential area A_{inf}



(b) 三角形分割の修正
(b) Re-triangulation of A_{inf}



(c) SRBE データの修正
(c) Re-built SRBEs inside A_{inf}

図 9 単端子拡大手続き
Fig. 9 Single-terminal-expansion.

になると考える．変形の間，F は穴のない多角形である．穴のない多角形領域を通過する曲線の位相的経路は，曲線と多角形外周との交点をデータ化することにより把握できる．そこで，三角形再分割の前にあらかじめ F と SRBE の交点位置を求めておけば，再分割実行後の各三角形について SRBE が通過するか否かを判定することができる．

拡大のアルゴリズムを得るには，上記の収縮のアルゴリズムにおいて，収縮を拡大に置き換えればよい．証明も収縮の場合と同様に行える．**図9**に，収縮で用いた例を逆にした単端子拡大の例を示す．

4.2 ワームクリープ法による端子の移動

単端子収縮と単端子拡大を交互に繰り返し適用することにより，あたかも尺取り虫が移動するがごとく端子を移動する**ワームクリープ法**と呼ぶ単端子移動の位相配線処理アルゴリズムを提案する．単端子移動の定義に沿って，端子 T が，時刻 τ_1 において凸多角形 T_1 であり，$[\tau_1, \tau_2]$ の時間に1頂点が直線分 $\overline{pq}$ を描くように移動し，時刻 τ_2 において凸多角形 T_2 になるとする．

手続き：ワームクリープ法 (Worm Creep Method)

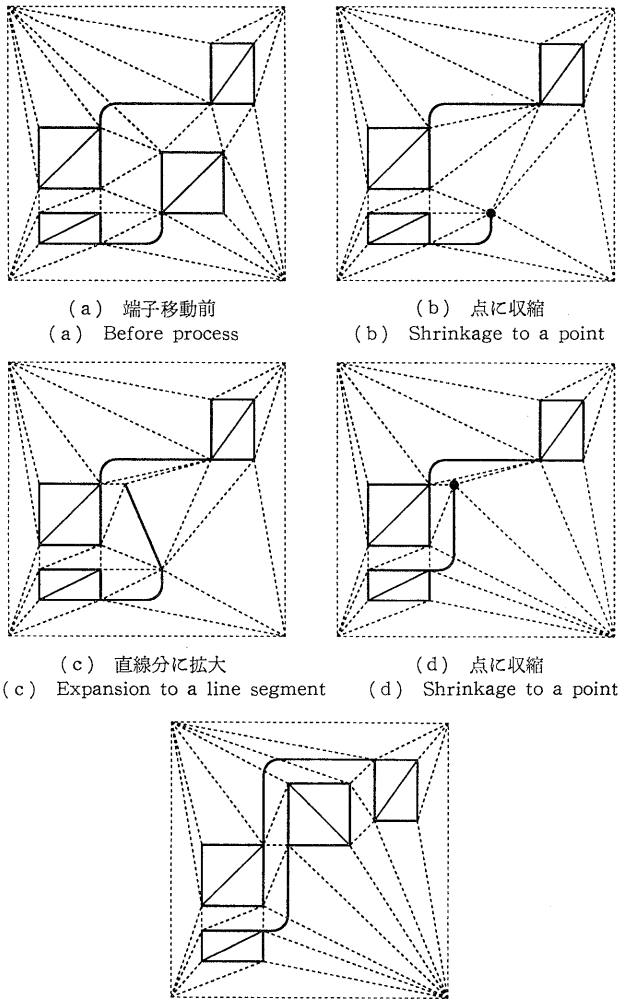
(**step 1**) 端子 T を凸多角形 T_1 から点 p にする単端子収縮を実行する．

(**step 2**) 端子 T を点 p から直線分 $\overline{pq}$ にする単端子拡大を実行する．

(**step 3**) 端子 T を直線分 $\overline{pq}$ から点 q にする単端子収縮を実行する．

(**step 4**) 端子 T を点 q から凸多角形 T_2 にする単端子拡大を実行する．

図3の例題を実行する場合のワームクリープ法の実行過程を**図10**に示す．図10(a)は，時刻 τ_1 におけるレイアウトデータ $\mathcal{L}(\tau_1)$ を表している．図10(a)において矩形である端子が，左下の1点に収縮し（図10(b)），左上方向に伸びて1本の直線分に拡大し（図10(c)），左上の1点に縮み（図10(d)），最後に目標の多角形に膨んで移動を遂げる（図10(e)）．レイアウトには2本の SRBE が含まれるが，それらは端子の収縮・拡大の各段階で修正され，時刻 τ_2 におけるレイアウトデータが図10(e)のように得られる．



(a) 端子移動前 (a) Before process
(b) 点に収縮 (b) Shrinkage to a point
(c) 直線分に拡大 (c) Expansion to a line segment
(d) 点に収縮 (d) Shrinkage to a point
(e) 凸多角形に拡大し移動完了 (e) Expansion to a polygon and complete process

図10 ワームクリープ法の振舞
Fig. 10 Process of Worm Creep Method.

5. 考察と結言

凸多角形端子と曲線配線からなるハイブリッド IC や PCB の平面レイアウトの対話設計において，設計者が指定する部品の移動に対応して，周囲の配線が経路を保って追従するように見える「レイアウト連続変形エディタ」を高速に実現することを目的とし，従来の位相配線データを凸多角形の端子を扱えるように拡張し，配線位相を保つ端子移動の新しいアルゴリズムとしてワームクリープ法を提案した．

従来，2つの位相配線処理のアルゴリズムが提案さ

れている（文献 2), 5)）．それらは，データベースの変更なしに配線位相を保てる限界まで端子を移動し，一旦端子を途中停止する．そして，データベースを部分的に再構成し，移動を再開する．これを端子が目標点に達するまで繰り返す．停止回数や計算時間は報告されていないが，データベースの状況に応じて非常に多くの途中停止をする可能性がある．さらに，停止点の座標を計算する上で丸め誤差が発生することが避けられないと思われ，それによるデータ構造の矛盾発生を回避するための繁雑な処理が端子を停止するたびに必要になるであろう．ワームクリープ法が従来手法と本質的に違うのは，端子が刻々移動するのではなく，一度に目標点まで延びることである．この違いにより，上に述べた種々の困難を招く端子の途中停止を不要とすることができた．その上で，凸多角形端子を扱え，端子を変形できる，という従来手法にない機能的柔軟さをもつことができた．

本文ではひとつの端子の移動だけをとり上げたが，一般的には移動対象は部品であり，複数端子をまとめて移動する必要がある．この機能は単端子移動を繰り返すことで実現できる．ただし，部品を構成する端子間で衝突が発生しないように，かつ，できるだけ少ない回数で移動できるように工夫する必要がある．例えば，複数の端子をある方向に並進するときは，進行方向前方の端子から移動する．

本手法を多層の場合に拡張することは比較的容易である．層間をつなぐビアを，関係するすべての層の同位置に存在する端子として扱えばよい．特定の層においてビアの移動が指令されたときには，関連するすべての層においてそのビアに相当する端子を移動する．

実験のために，ワームクリープ法を組み込んでエディタのプロトタイプを開発した．そして，ADEE ジャパン '93 (Automated Design and Engineering for Electronics JAPAN '93) に出展し，Sun Sparcstation II (25 MIPS) 上で図 1 に示す実演を行った[6]．実演では実際の設計を模して合計 23 回の部品移動を指令した．各部品移動指令を与えてから新たなレイアウトが表示されるまでの応答時間は 0.15 秒〜0.84 秒と満足できる結果を得た．

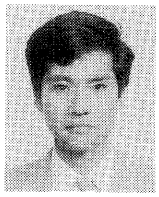
一方，ワームクリープ法で得られた位相配線をデザインルールを満たした曲線に変換する段階（位相物理変換）については，プロトタイプでは便宜的なアルゴリズムを用いており，必ずしもデザインルールを満たせない．位相物理変換は，SRBE の通過三角形の範囲を越えて曲線の経路を探索する必要があるので自明な処理ではないが，端子形状が点の場合には文献 2), 4), 5) にアルゴリズムが提案されている．これらのアルゴリズムを端子凸多角形の外周上において配線接続位置を最適化できるように拡張することは今後の課題として残されている．

謝辞 本研究の機会を与えてくださった(株)村田製作所および同社ファンクショナルデバイス事業部ハイブリッド商品統括部 枝 茂男 部長に感謝する．なお，本研究の一部は文部省科学研究費一般研究（B）(05452209) および CAD 21 研究体（於 北陸先端科学技術大学院大学）の援助による．

参 考 文 献

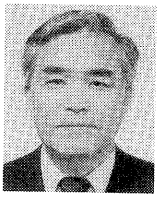
1) Scott, W. S. and Ousterhout, J. K.: Plowing: Interactive Stretching and Compaction in Magic, *21st IEEE Design Automation Conference*, pp. 166-171 (1984).
2) Dai, W. W., Kong, R., Jue, J. and Sato, M.: Rubber Band Routing and Dynamic Data Representation, *IEEE International Conf. on Computer Aided Design*, pp. 52-55 (1990).
3) Valainis, J., Kaptanoglu, S., Liu, E. and Suaya, R.: Two-Dimensional IC Layout Compaction Based on Topological Design Rule Checking, *IEEE Trans. Computer Aided Design*, Vol. CAD-9, No. 3, pp. 260-275 (1990).
4) Leiserson, C. E. and Maley, F. M.: Algorithms for Routing and Testing Routability of Planar VLSI Layouts, *Proc. 17th Ann. ACM Symp. on Theory of Computing*, pp. 69-78 (1985).
5) Liu, E.: Two Dimensional IC Layout Compaction, Ph. D. dissertation, Univ. Calgary (1986).
6) 村田 洋，梶谷洋司：CALTETT：アナログレイアウトエディタ，電子材料，Vol. 32, No. 1, pp. 142-145 (1993).
7) Chew, L. P.: Constrained Delaunay Triangulations, *Algorithmica*, Vol. 4, pp. 97-108 (1989).

（平成 6 年 3 月 10 日受付）
（平成 6 年 9 月 6 日採録）



村田　洋（正会員）

1980年金沢大学工学部電気工学科卒業後，(株)コンピュータアプリケーションズ勤務を経て，1984年より(株)小松村田製作所に勤務し現在に至る．1994年北陸先端科学技術大学院大学修士課程修了，現在博士課程在学中．業務としてハイブリッドIC，プリント配線基板等の設計に携わる一方，大学においてそのCADシステムの研究に従事．



梶谷　洋司（正会員）

1969年東京工業大学大学院博士課程修了．工学博士．同年東京工業大学助手．同助教授，教授（電気電子工学科）を経て，1991年から北陸先端科学技術大学院大学教授および東京工業大学教授(併任)．東京工業大学創造プロジェクト研究体CAD 21 研究長．組合せアルゴリズム理論の応用，特に，VLSIのレイアウト設計の理論とその計算機による実現の研究に従事．1969年，1973年，1985年度論文賞を電子情報通信学会から受賞．1992年IEEE Fellow．電子情報通信学会，IEEE会員．