

Multi-pin net の引き剥しを考慮した並列配線処理

佐野雅彦 高橋義造

徳島大学工学部

E-mail:{sano, taka}@n30.is.tokushima-u.ac.jp

概要

配線問題に於て配線品質を向上させる方法の一つにMulti-pin net (多端子ネット) の最短配線があり, Steiner tree問題として知られている. 本稿では, プロセッサ競合方式における多端子ネットの並列処理アルゴリズムとして, ネットを複数の端子グループに分割して並列処理する方法を提案する. このアルゴリズムは初期配線段階と改善配線段階から構成され, ネットの並列性とネット間の並列性の異なる並列性を用いて並列処理する. この並列処理方式の基本アルゴリズムと端子の分割アルゴリズムについて説明し, 並列計算機への実装について考察する.

Parallel Wire Routing with Ripup of Multi-pin net

Masahiko SANO Yoshizo TAKAHASHI

Department of Information Science and Intelligent Systems,
Faculty of Engineering, Tokushima University

E-mail:{sano, taka}@n30.is.tokushima-u.ac.jp

ABSTRACT

The problem of finding the minimal length wire routing of a multi-pin net in order to improve wiring quality belongs to the class of Steiner tree problems. This paper presents a new parallel processing algorithm for multi-pin net problems using the competing processors approach, in which a net is decomposed into several terminal groups. The algorithm consists of two procedures: 1) initial routing and 2) routing improvement by using two routing parallelisms, namely, net and among nets parallelisms. In this paper, the basic processing algorithm and the terminals grouping algorithm are described, and their parallel implementation on a multicomputer is discussed.

1. はじめに

配線問題における品質改善方法には配置段階における改善と配線段階における改善に分けられる。多端子ネット (Multi-pin net) は複数の端子間の接続を一本のネットで行うものであり、通常はCAD等が出力するネットリストに含まれているため、配線処理する側でもこれに対応する必要がある。最も簡単な多端子ネットの処理はpin-to-pinのネットに分割する方法であるが、例えば、最も簡単な方法である最短距離の端子同士を接続する方法では無駄な迂回経路が必要となり、配線品質の向上を阻害する。そこで、Steiner tree のように配線経路の途中から分岐する、より最適な経路を求める方法が必要である。

一方、我々は以前よりプロセッサ競合方式による並列配線問題において、配線品質向上のための並列配線アルゴリズムを研究している[1,2]。これまで、多端子ネットはpin-to-pinで処理を行ってきたが、考察の結果、配線品質の向上には更に最適な経路が得られる経路探索アルゴリズムが必要であると判断し、多端子ネットの並列処理方法について研究をしている。

本論文では多端子ネットを並列処理するためのネット分割による並列処理方法、及び、プロセッサ競合方式による並列改善処理においてどのように経路改善 (引き剥し再配線) すれば良いか、その一例として分割したそれぞれのグループがSteiner tree で配線・改善されるアルゴリズムを提案し、並列計算機への実装について考察する。

2. Multi-pin net (多端子ネット) の配線問題

多端子ネットの配線問題の例を図1に示す。この問題をpin-to-pinに分割して解いた場合の結果を(a)に、Steiner tree問題として解いた結果を(b)に示す。この結果で判るようにpin-to-pin分割では限界があり、より配線品質を向上させるためにはSteiner treeの様な配線経路途中から分岐する経路が必要である。しかしSteiner treeを求める問題はNP困難であるため[3]、端子数の増加によって計算量が大幅に増加する。しかし、実際には近似解を求める簡単な解法がある。こ

れは以下の方法で実現される[4]。まず、代表となる端子を選び、探索を始める。そして最初に到達した端子までの経路を確定する。次に、確定した経路から探索し、次に見つかった端子までの最短経路を確定する。そして、これまでに見つかった経路と合わせて次の探索の始点とする。この操作を繰り返すことによりpin-to-pinより改善された経路が得られる。

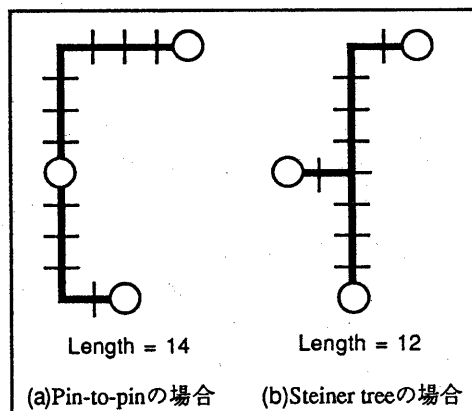


図1 多端子ネット問題の簡単な例

3. プロセッサ競合方式による並列配線処理

プロセッサ競合方式[1,2]は図2に示すマスター・スレーブモデルを用いている。このモデルは各種並列計算機に対する実装性を考慮して、共有メモリ、及び放送機構を使用していない。プロセッサ競合方式では、配線処理をスレーブプロセッサ (以下スレーブ) が担当し、その配線結果の評価、及び配線処理の割り当てはマスタープロセッサ (以下マスター) が行う。また、一本のネット単位で処理を割り当てるネット割り当て法を用いることにより、ネット間の並列性を用いた複数ネットの同時配線を行なう。

次に、プロセッサ競合方式における並列配線アルゴリズムを簡単に説明する。ここではネット数を n 本とする。マスターはこれらのネットを m 台のスレーブに各々割り当てる (但し、 $m < n$)。そして割り当てられたスレーブは配線処理を開始する。やがて配線処理を終えたスレーブは配線結果をマスターに送り、マスターはこれを評価する。その後、マスターは次のネットを割り当てる。このとき、未処理のネットを優先し、

未処理のネットが無ければ既に配線処理されているネットから改善可能なネットを選択して割り当てる。スレーブはこのネットデータについて配線処理を行う。

プロセッサ競合方式ではこの処理を反復して行うことにより経路を改善する。経路の改善は各スレーブで行われるので複数のネットを並列に改善できる。

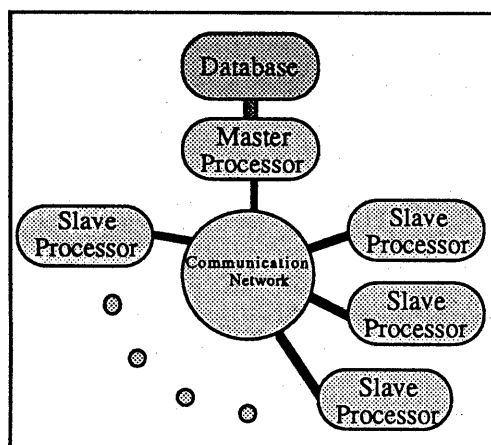


図2 プロセッサ競合方式

・経路の並列改善[2] 一般に、逐次配線アルゴリズムを並列化した場合では配線結果の重複を許容しないため、配線領域等のデータの一貫性が容易に保たれる。しかし、並列計算機では同期待ちの増加の原因となり性能を低下させる。そこで、アルゴリズムの並列性を重視して、従来とは逆に配線結果の重複を許容することにより配線結果の矛盾を配線結果の重複として取り扱う。その結果、配線結果の矛盾の問題は配線経路の改善という形式に吸収される。つまり、配線経路の改善を多数のプロセッサで並列処理する方式となる。

4.多端子ネットの並列配線処理

ここではプロセッサ競合方式への実装を前提とした、多端子ネットを並列に処理するための基本的なアルゴリズムについて説明する。

まず、多端子ネットの並列配線処理を初期配線と改善配線の2段階に分ける。初期配線段階ではネットを複数の端子グループに分割してネットの並列性とネット間の並列性を用いた並列配

線処理を行う。一方、改善配線段階では配線経路の改善を枝、接続点、及び端子の単位で並列処理する。この改善配線も2つの並列性を用いて処理される。

4.1 初期配線

本節では多端子ネットを複数の端子グループに分割して並列配線処理を行う初期配線について説明する。まず、グループ化の理由について述べ、次に端子の分割方法、ネットの前処理、及び、異なるプロセッサ割当てを行った場合の分割方法の違いについて述べる。

・グループ化の理由 我々は多端子ネットを並列処理する方法として、以下の理由からネットを複数の端子グループに分割して複数のプロセッサに割り当てる方法を用いる。

- ・1台のプロセッサに1本のネットを割り当てた場合、そのネットは逐次処理されるためにネット間の並列性しか利用できない。
- ・一方、ネットを分割して複数のプロセッサに割り当てた場合、ネットの並列性とネット間の並列性の2つの異なる並列性を用いた並列配線処理が可能になり、プロセッサの利用率向上が期待できる。
- ・また、1台で処理する場合に比べて配線領域の範囲が小さくなるために処理時間が短くなる。その結果、配線領域の更新間隔も短くなり、他のプロセッサが参照するデータが最新のものである確率が高くなる。これはプロセッサ競合方式におけるプロセッサ間の配線結果の矛盾抑制に有利である。
- ・グループに分けることで最適な経路を得ることが難しくなるが、後述する改善配線によって準最適な経路が得られる。

・端子の分割 端子の分割には幾つか方法がある。前述のようにSteiner tree問題はNP困難であるが、以下の理由から、分割されたグループの端子数を3以下とする。

- ・端子数が多いと各プロセッサの計算量が増大し、かつ、これらの端子は逐次処理される必要があり、処理時間が長くなる。その

結果、配線領域更新時間が長くなる。

- 一方、幾つかのグループに分解した場合、各グループでは最適経路で配線できる。しかし、グループ間の配線は最適経路になる保証はないが、改善配線において準最適な経路に改善することが可能である。
- 端子数を3以下にした場合、図3に示すように簡単にSteiner tree問題を解くことが出来る。(a)は全ての端子が端子を囲む長方形の周上にあり、かつ、各々の座標軸が同じでない場合である。この場合、長方形の頂点に位置する端子以外の端子から反対側の辺に垂線を延ばし、二つの垂線の交点を接続点とすれば良い。(b)は二つの端子が同座標軸上にある場合で、残る端子から延ばした垂線と、軸の交点を接続点とすれば経路長は最小となる。また、端子の一つが長方形の辺上でない場合はその端子が接続点となる(c)。これらの最短配線経路長は端子を囲む長方形の縦横の辺の長さの和となる。

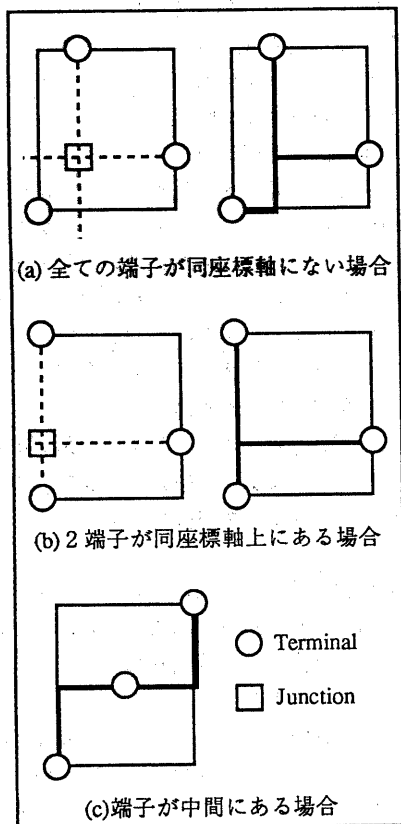


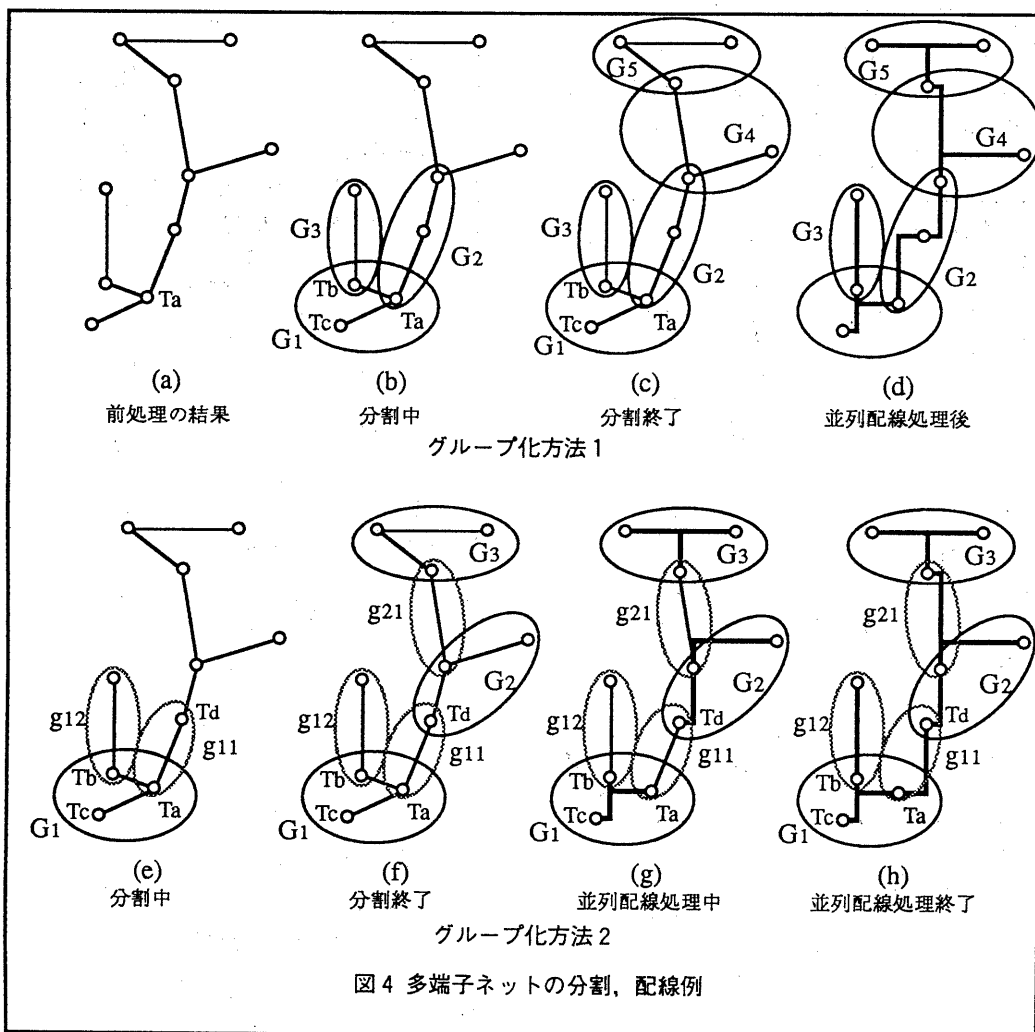
図3 3端子のSteiner tree

・前処理 端子を効果的にグループ化するための前処理について説明する。ここでは全ての端子が無閉塞な経路で接続されるグラフを生成する。グラフ生成の戦略としては最短経路を結ぶ方法を用いる。これは端子間距離の短いものから順に接続し、全ての端子が一つのグラフになるまで行う方法である。

・端子のグループ化 グループに分割されたネットを並列処理するには2つの方法が考えられる。グループ化方法1は1回の並列処理でネット経路探索する方法、方法2は2ステップの並列処理で経路探索する方法である。以下に、各々の方法におけるグループ化について述べる。

・グループ化方法1 この方法を図4を用いて説明する。(a)は前処理を行った結果を示している。まず、代表端子を決める。ここでは、他の端子への枝数が最も多い端子とする。もし、同数の端子が複数あれば適当に選択する(図中Ta)。この端子から接続されている隣接端子のうち、距離の短い端子Tb、Tcを選びグループ1(図中G1)とする。次に、G1からG1以外に隣接端子を持つ端子Taを選択し、G1以外の端子から距離が近く、かつ、隣接している2端子を選択しG2とする(b)。残りのG1の端子Tbについても同様の操作をする。以下、これを繰り返して未選択の端子が無くなるまで行う(c)。この様に分けられたグループを各々異なるプロセッサに割り当てることにより並列処理を行う。この結果、各グループ内の配線結果はSteiner treeとなる(d)。この方法の問題点は、異なるグループが同時に配線処理されたとき、グループ間の共通端子付近で配線経路の矛盾が起きる場合である。これはプロセッサ競合方式に実装することを前提としているために、各スレーブプロセッサ間のデータの一貫性を保証していないことが原因である。

・グループ化方法2 この方法では、前者の問題点を回避すべく、並列配線を2ステップに分けて行うものである。そのための分割方法を図4(e-h)に示す。前処理後の状態は(a)と同じである。まず、前述の方法と同様にグループG1



を選ぶ。次にG1以外の接続端子を持つ端子Taを選び、Taとその枝に接続される端子Tdでg11を構成する。残りのTbについても同様にして、g12を生成する(e)。ここまではPhase Iとする。Phase II ではg11,のG1に属さない端子Tdを新たなグループG2の基準点とし、G1の選択要領でグループの残りの端子を選択する(f)。以下Phase I,IIを繰り返して全ての端子をグループ化する。並列配線するときには、最初のステップ1はPhase Iで選択されたグループG1,G2,G3を並列処理し(g)、次のステップ2でPhase IIで選択されたg11, g12, g21を並列処理する。ただし、ステップ2では、異なるグループの端子を接続するので、グループ間の配線処理となる(h)。

4.3 改善配線

改善配線は初期配線により発生した交差・接触のある経路、及び品質の悪い経路の改善を目的とする。改善配線は改善可能な経路の検出、経路の改善、改善結果の評価の3つに分かれ、これらを反復処理することにより経路の改善を行う。まず経路の改善方法について述べ、次に、改善可能な経路の検出、及び評価方法について説明する。

・経路の改善 我々は経路の改善を1)枝の改善、2)接続点の改善、3)端子の改善の3段階に分ける。枝とは端子(接続点を含む)から端子までの経路とし、接続点とはビアを含むネットの分岐点を指す。なお、説明の簡略化のため一層の場合に限定する。

1)枝の改善 まず、改善すべき枝を引き剥す。その結果、ネットは2つの部分ネットに分割されるので、分割されたネット間の再配線により枝の経路改善を行う(図5)。この操作により2つの部分ネットが最短経路で接続される。

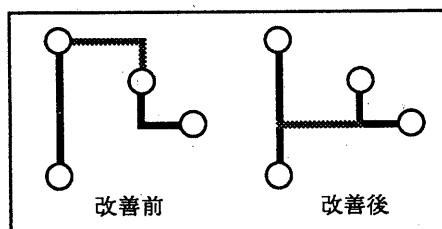


図5 枝の改善

2)接続点の改善 接続点の改善は図6のような不適当な接続点の改善を行う。改善方法は、まず接続点を除去し、その接続点と接続している枝を他の接続点及び端子までを引き剥して再配線する。但し、Steiner treeを生成するために引き剥す枝数は3以下にする。

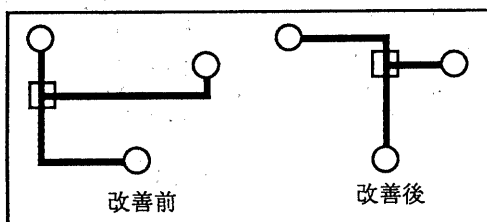


図6 接続点の改善

3)端子の改善 端子の改善では、端子周辺の経路の改善を目的とする。図7では他グループの端子を共有する場合の端子周辺の経路改善例を示す。図中、端子Taは他のグループg11, g12の端子を共有しており、ここではg11, g12の経路を引き剥す(a)のグレーの部分)。その結果、接続すべき点はTa, Td, Teとなり、その再配線結果は(b)となる。なお、枝が2本の端子の改善では(図(c)の端子Td) 2本共引き剥して再配線する(d)。

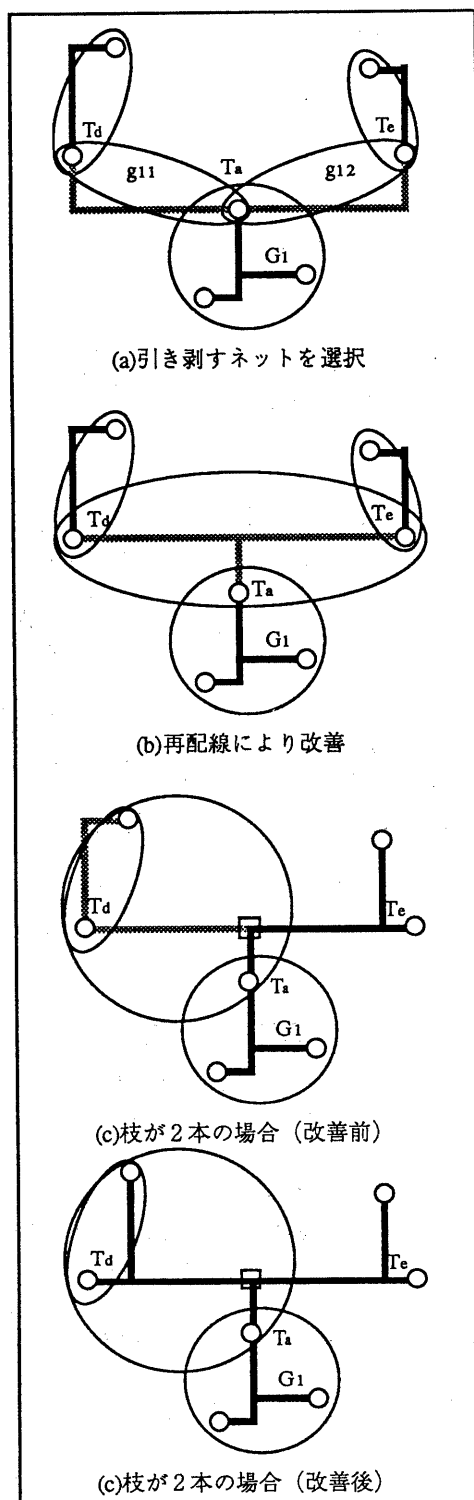


図7 端子における改善

4.4 改善の可能性

本節では経路改善の可能性について考察する。可能性の検出には以下に示す評価方法を用いる。評価は配線コストを基に計算し、用いる配線コストは配線距離、交差・接触数等である。

・枝の評価 枝の評価は次の式で各枝毎に計算される。この評価値が1のとき、最適となる。

$$\text{枝の評価値} = \frac{\text{実際の配線長}}{\text{理想配線長}} + \text{交差・接触コスト}$$

・接続点の評価 接続点では接続点から延びる枝のコストの合計を接続点の評価値とする。但し、理想配線長はSteiner tree で得られる長さとする。この場合も最適経路の評価値は1である。

$$\text{接続点の評価値} = \frac{\text{配線長の合計}}{\text{理想配線長}} + \text{交差・接触コストの合計}$$

・端子の評価 端子の場合は端子から、他の端子までの枝の評価値の合計で評価する。例えば図8の端子Taに着目すると、この端子の評価値の計算範囲は図に示された斜線部分になり、評価値は以下の式で計算される。図8の場合の最適値は枝数が4本なので4である。

$$\text{端子の評価値} = \text{各枝の評価値の合計}$$

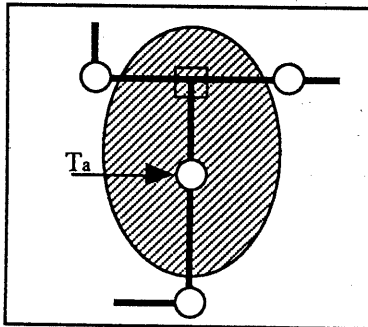


図8 端子の評価値の計算範囲

また評価値とは別に、異なるグループを並列に配線すると不適切な配線結果になる場合がある。特に隣接するグループでは起き易い。そのため、異なるグループで共通な端子やステップ2で配線される端子は全て改善の対象になり得る。

・改善結果の評価 再配線の結果は前述の評価方法により、枝、接続点、端子毎に評価されるが、ネット全体の評価値は枝の評価値を合計したものをを用いる。

4.5 経路矛盾の取扱

プロセッサ競合方式では、図9に示すように、同じネットの異なるグループの配線結果が重複する場合がある。この場合、あるグループの配線結果に重複が検出された場合、経路の状況を調べた上で残りの部分を捨てることにより経路の整合性を保証する。図では評価している経路は太線で示されており、で重複が検出された点から残りの経路（グレーの部分）を捨て、既に存在している経路と接続する。

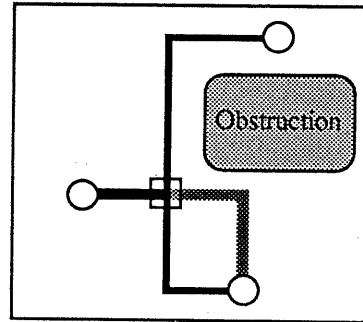


図9 経路矛盾の解決

5. 実装

この章では、提案してきた多端子ネットの並列処理アルゴリズムの実装について考察する。実装はプロセッサ競合方式を対象としているが、他の方式に対しても実装可能である。

プロセッサ競合方式では、複数のネットを同時に配線処理し、並列に経路改善を行う並列配線処理方法を用いている。この方式では図10に示すように、並列配線処理の前半部分の並列性は高いが、後半部分では低下する。これは、配線処理が進行すると改善すべき経路が減少し、改善処理の並列性が低下するためである。

今回提案した多端子ネットの並列配線処理方法は4章で説明したように初期配線と改善配線に分割している。初期配線は並列配線序盤における並列性を向上させる目的で行っており、改

善配線は並列性の低下する後半の段階で、処理を細かく分割することにより並列性の向上させることを目的としている。

次に、マスタ・スレーブの処理割り当てについて考察する。初期配線では、マスタは分割されたグループをスレーブに割り当てる。このとき、一本のネットの分割されたグループを一度に配線処理する場合は、グループ化方法1を用いて分割し、2ステップで配線処理する場合はグループ化方法2を用いる。後者の場合、各ネット毎に処理ステップが正しく守られるようにグループの割当てを行う。

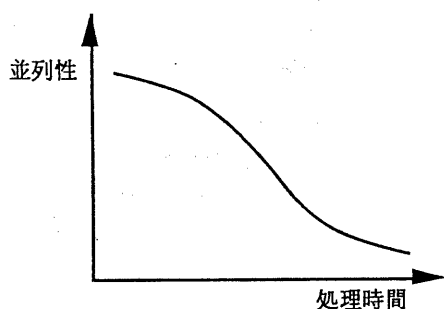


図10 経路改善における並列性の変化

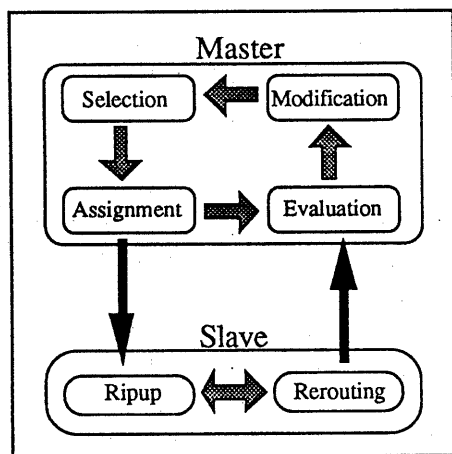


図11 改善配線における動作の概略図

初期配線が終り次第、改善配線に移る。改善配線では経路の不適當な部分の引き剥し・再配線によりネットの経路を改善する。これはプロセッサ競合方式で用いている並列経路改善方式と基本的に同じ方法であるために容易に実装で

きる。また、必要に応じてネット全体を引き剥す。この処理割当ては、初期配線と基本的に同じで、マスタが改善部分の選択、割り当て、改善結果の評価を実行し、スレーブはマスタから割り当てられた改善部分を引き剥し・再配線によって改善する。この処理の概略を図11に示す。

6. まとめ

本稿では引き剥しを考慮した多端子ネットの並列配線処理をプロセッサ競合方式で効率良く処理するためのネット分割方法、及び、枝、接続点、端子毎の経路改善方法について提案を行った。

本方式が従来の方法と異なる点は、ネットの並列性とネット間の並列性の異なる2つの並列性を含んだ並列経路改善アルゴリズムを使用している点である。このアルゴリズムでは改善可能な部分が減少するに連れて経路改善の並列性が低下する問題があるが、改善の単位を枝、接続点、端子の3段階に分け、細かな単位で改善を行うことにより並列性の低下を防ぐ。

現在、このアルゴリズムは実装中である。

7. 参考文献

- [1] 佐野雅彦, 高橋義造: 分散メモリ型と共有メモリ型マルチプロセッサによる並列配線処理の性能評価, 情報処理学会論文誌, vol.33, No.3, pp. 369-377, (1992).
- [2] 佐野雅彦, 高橋義造: プロセッサ競合方式による並列配線処理 ～引き剥し処理による品質改善～, JSPP'93, pp. 331-338, (1993).
- [3] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, San Francisco, (1979).
- [4] S.B. Akers, "Routing," in Design Automation of Digital Systems, vol. 1, Breuer, M.A. (ED).