

MIMD 型並列計算機上の LSI ルーター ——PROTON——

山内 宗[†] 中田 登志之[†] 石塚 昭夫^{††}
西口 信行^{††} 小池 誠彦[†]

本論文では、我々が開発した並列ルーター PROTON (Parallel ROuTer ON a parallel machine) について述べる。PROTON は、64 台のマイクロプロセッサで構成された MIMD 型並列マシン Cenju の上に実装されている。配線速度を向上するために、ネット内並列性とネット間並列性の 2 つのレベルの並列性を用いている。ネット内並列性を引き出すために、配線層ごとに配線領域を、配線方向に細長い帯状の配線領域に分割し、各要素プロセッサを各帯状の配線領域に割り当て、配線領域内の探索処理を並列に行う。ネット間並列性については、概略配線経路が重ならないネットは同時に配線処理を行うことができるので、そのようなネットのグループを選び、同時に配線処理を行うことによって並列性を引き出している。さらに、PROTON の並列配線アルゴリズムは、従来の逐次型ルーターと同等の配線品質を保っている。また、PROTON では配線領域に関する情報が各要素プロセッサに分散配置されるので、要素プロセッサ数を増すことによって、より大規模の VLSI 配線問題も扱うことが可能となる。ネット内、ネット間の両方の並列性を合わせて用いることにより、中規模のチャネルレス・ゲートアレイ (1,537×1,790 グリッド, 12,591 ピンペア) を配線対象とした場合に 64 台のプロセッサで 43 倍の速度向上を得た。

A Parallel LSI Router PROTON on an MIMD Parallel Machine

TSUKASA YAMAUCHI,[†] TOSHIYUKI NAKATA,[†] AKIO ISHIZUKA,^{††}
NOBUYUKI NISHIGUCHI^{††} and NOBUHIKO KOIKE[†]

This paper describes a new parallel detailed router named PROTON. PROTON is implemented on an MIMD parallel machine named *Cenju* which consists of 64 micro-processors. In order to improve routing speed, PROTON incorporates two level parallelism, mainly intra-net parallelism and inter-net parallelism. In order to exploit intra-net parallelism, PROTON divides routing area into narrow stripes according to its orientation, assigns each processing element to each stripe and searches each stripe in parallel. As for inter-net parallelism, when the global routes of the nets have no intersection, multiple nets can be routed simultaneously. So, PROTON selects such nets and routes them simultaneously. These parallelization method do not sacrifice any quality of the routing result compared with conventional sequential routers. By combining intra-net parallelism and net level parallelism, we have achieved a speedup of 43 times using 64 processors for a medium scale channelless gate-array (1,537×1,790 grids, 12,591 pin pairs).

1. はじめに

配線処理は、VLSI を開発する際の重要な課題の 1 つとして挙げられるが、莫大な処理時間とメモリ空間を必要とする。従って、今後大規模な VLSI の配線を行うためには、配線アルゴリズムの並列化による高速化が重要である。また、大規模な VLSI の配線処理が

必要とする莫大なメモリ空間については、データを並列計算機の各プロセッサに分散配置するという方法を取り、プロセッサ数を増すことによって、より大きな VLSI の配線処理にも対応が可能となる。従って、今後 VLSI が大規模化するにつれて、配線処理の並列化は重要となると考えられる。

本論文では、以下の特徴を持つ並列 LSI ルーター PROTON (Parallel ROuTer ON a parallel machine)^{1)~4)} について述べる。

- 幅優先の探索にもとづいた並列線分探索法
- ネット内、ネット間の 2 レベルの並列性の抽出
- 分散共有メモリ型 MIMD (Multiple Instruction

[†] 日本電気(株) C&C システム研究所
C&C Systems Research Laboratories, NEC Corporation

^{††} 日本電気(株) ULSI システム開発研究所
ULSI System Development Laboratories, NEC Corporation

streams Multiple Data streams) 型の並列計算機 Cenju 上で動く並列ルーター

- 種々の複雑なデザイン・ルールを持つ多層配線層のチャンネルレス・ゲートアレイを配線対象とする詳細配線
- 従来の逐次型ルーターと同等の配線品質を保ちながら、高い並列性を抽出

従来の並列ルーターでは、十分な並列性を引き出せずにいるもの、高い並列性を引き出すために配線結果の品質（配線長、配線率）を犠牲にしているもの、あるいは、配線本数の制限や複雑なデザイン・ルールの対応が原因で実用レベルの配線対象を扱うことが困難なものが多かった。しかし、本論文で述べる LSI ルーター PROTON は、従来の逐次型ルーターの配線品質を犠牲にすることなく高い並列性を引き出し、しかも、複雑なデザイン・ルールを持つ実際の大規模ゲートアレイの配線処理が可能である。

第2章では、従来の配線アルゴリズムとその並列化について述べ、第3章では、PROTON で採用した基本配線アルゴリズムについて述べ、第4章では、PROTON における配線アルゴリズムの並列化手法について述べ、第5章では、PROTON を実装した並列マシン Cenju の概要について述べ、第6章では、PROTON の評価結果について述べ、第7章では、本論文の結論について述べている。

2. 従来の配線アルゴリズムとその並列化

2.1 従来の配線アルゴリズム

従来よく知られている配線アルゴリズムとしては、

1. 格子展開法（迷路法）⁷⁾
2. 線分探索法¹¹⁾

3. チャンネル配線法¹⁸⁾

の3種類がある。各配線アルゴリズムの特徴を表1に示す⁸⁾。

格子展開法や線分探索法のように配線処理を1ネットずつ行うアルゴリズムでは、1つのネットの配線処理に内在するネット内並列性と、複数のネットを同時に配線することによって得られるネット間並列性の2つのレベルの並列性が内在すると考えられる。しかし、チャンネル配線法は、チャンネル内のすべてのネットの配線処理を同時に行うので、ネット内並列性やネット間並列性を用いた並列化というものは存在しない。チャンネル配線法は、グラフ理論を用いて配線処理を行うので、グラフ処理の並列化というアプローチも取り得るがそのような例は無く、むしろ、配線領域間の並列性の一種が用いられている¹⁹⁾。もちろん、配線領域間の並列性は、チャンネル配線法に限らず用いることが可能である¹⁴⁾。表2に、(1)ネット内、(2)ネット間、(3)配線領域間の3種類の並列性と各配線アルゴリズムの問題点についてまとめる。

格子展開法のネット内並列性（波面伝搬の並列処

表1 各配線アルゴリズムの特徴
Table 1 Features of routing algorithms.

	格子展開法	線分探索法	チャンネル配線法
径路の特徴	最短径路	最少折れ曲がり径路	端子列間の配線
メモリ使用量	多	中	少
配線領域の制約	制約なし	制約なし	障害物不可 配線間隔一定
デザイン・ルールへの対応	問題が多い	容易	問題が多い
配線速度	遅い	やや遅い	速い

表2 ネット間、ネット内そして配線領域間の並列性の問題点
Table 2 Problems of intra-net, inter-net and inter-routing-area parallelism.

	格子展開法	線分探索法	チャンネル配線法
ネット内並列性	波面のプロセッサへの効率良い割り当て困難 →プロセッサ稼働率 低 処理粒度が細かい →MIMD に不適	プロセッサ間通信が多く なりやすい	—
ネット間並列性	複数ネットを複数のプロセッサが協調しながら配線 →プロセッサ台数増に連れて配線率、配線の品質が低下（逐次処理と比較して）	—	—
領域間並列性	配線領域の分割方法が配線結果に悪影響を及ぼしやすい	—	チャンネル領域間の並列性を利用 各チャンネルの配線処理の順番が配線結果の品質に影響を与える

理)を用いた例は、比較的多く見られる^{9),10)}。処理の粒度が細かいので Routing Processor⁹⁾, MAPLE¹⁰⁾ のどちらも専用のプロセッサを用いている。

線分探索法のネット内、ネット間並列性を用いた例もいくつかある^{12),13)}。これらは、複数のプロセッサが互いに協調しながら複数ネットの配線処理を同時に進行するので、逐次処理した結果と比較した場合に、配線品質や配線率に関して何の保証も無く、実用化という点では問題が多い。

2.2 PROTON における配線アルゴリズムの選択

本システムで配線の対象としているのは主にゲートアレイ、特に、近年大規模ゲートアレイの主流となっているチャンネルレス型のゲートアレイ (Sea-of-gates) である。チャンネルレス型のゲートアレイは、「チャンネルの領域が必ずしも明確ではない」、あるいは、「チャンネル内に障害物が存在する」等の特徴を持っている。従って、配線アルゴリズムを選択するに当たっては、そのように多様なデザイン・ルールに対応することができるかどうかが重要である。その他にもメモリ使用量、プロセッサ稼働率、処理粒度などがアルゴリズム選択の重要な判断基準となる。表 3 に、各配線アルゴリズムの特徴を比較したものを示す。

この表 3 を見て、さらにチャンネルレス型ゲートアレイの細かいデザイン・ルールにも対応する必要があるという点を考慮に入れると、基本配線アルゴリズムとして、線分探索法を選択するのが最も適していることがわかる。そこで、本論文では、基本配線アルゴリズムとして線分探索法を選択し、MIMD 型並列計算機に適した新たな並列線分探索手法を提案する。提案する並列線分探索手法は、ネット内並列性とネット間並列性の 2 種類の並列性を併用することによって、高い並列性を得ている。並列化手法の詳細については後述するが、ネット内並列性に関しては、配線領域をプロセッサへ割り当てる方式に工夫を加えることによってプロセッサ間通信を減らし、高い並列性を得ている。さらに、ネット間並列性については、概略配線経路が重な

らないネットの配線処理を同時に行うことによって、逐次計算機上の配線結果と同等の配線経路の品質や配線率を得ることを保証した (従来の研究では、逐次処理をした場合と同等の品質を得るのが困難であった)。

3. 基本配線アルゴリズム

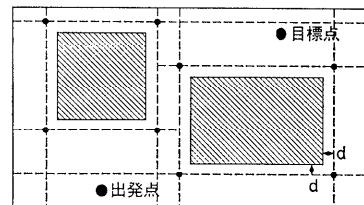
前述のように、並列 LSI ルーター PROTON の基本配線アルゴリズムとしては、線分探索法を用いることにした。線分探索法にもいろいろな種類があるが、実際に用いた配線アルゴリズムは「改良線分探索法」^{15),16)} を基本としたものである。このアルゴリズムを採用したのは、(1) $O(n \log^2 n)$ の処理時間 (探索線 n 本、通常の線分探索法が $O(n^2)$)、(2) 配線経路があれば必ず見つける、(3) $O(n)$ の記憶容量、などの特徴を有し、従来の線分探索法と比較した場合に処理速度やメモリ消費量の点で優れているからである。

改良線分探索法の処理は概ね、以下のように進められる。まず、コーナー・ポイント (CP) を障害物から d (障害物と配線の許容距離) だけ離れた位置に生成し、障害物等を回避するエスケープ・ライン (EL) を CP から生成する (図 1 参照)。そして、出発点からは探索線、目標点からは目標線を生成する。次に、探索線と交差するエスケープ・ラインあるいは目標線を探し、交差するエスケープ・ラインは次のステップの新たな探索線とする。この処理を、探索線が目標線と交差するまで繰り返す。そして、探索線が目標線と交差した場合には、配線経路が見つかったことになり、探索線を逆向きにたどって配線経路を推定する。探索線と交差するエスケープ・ラインが全くなくなっても目標線にたどり着かなかった場合には、配線経路が無いということの意味する。このように配線経路を探すことにより、最少折れ曲がりの経路が求まる。

ただし、オリジナルのアルゴリズムでは、実際の

表 3 各配線アルゴリズムの比較
Table 3 Comparison among routing algorithms.

	格子展開法	線分探索法	チャンネル配線法
メモリ使用量	×	○	○
デザイン・ルールへの対応	×	○	×
プロセッサ稼働率	×	○	○
処理粒度	×	○	○



● コーナー・ポイント (CP)
--- エスケープ・ライン (EL)
■ 障害物
図 1 改良線分探索法

Fig. 1 Improved line search algorithm.

ゲートアレイが有する複雑なデザイン・ルールへの対応がなされていないのでそれらの点については、改良を施している。

改良線分探索法のアルゴリズムは大きく分けて以下の3つのフェーズで構成されている。

1. 前処理 (障害物の周囲にコーナー・ポイント, エスケープ・ラインを生成)
2. 探索処理 (エスケープ・ラインの中から配線経路を探す)
3. 後処理 (新たに生成された配線経路は障害物となるので, その周囲のコーナー・ポイント, エスケープ・ラインを更新する)

これらのフェーズのうち, 探索処理が最も重い (全体の処理時間の90%以上を占める)。従って, 配線処理の高速化を図るためには, 探索処理部分をうまく並列化することが重要である。なお後述するように前処理, 後処理についても並列化している。

4. 配線アルゴリズムの並列化

前述のように, 改良線分探索法は, 前処理, 探索処理, 後処理の3つのフェーズで構成されている。この章では, 前処理や後処理の並列化, そして, 最も処理が重い探索処理の並列化について述べる。

4.1 配線領域の割り当て

配線の対象となるゲートアレイは, 配線層ごとに配線の向き (主軸方向) が決まっている。そこで PROTON では, 図2に示すように, 配線領域をその層の主軸方向に沿って帯状に分割して, 各プロセッサ (PE) にそれぞれの配線領域を割り当てる。このように各 PE が配線領域を分割して受け持つことによって, 後

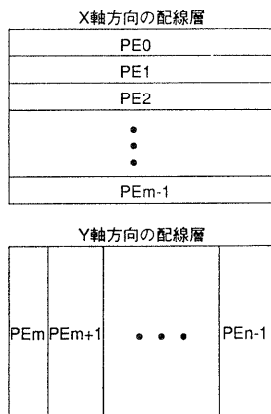


図2 配線領域の各 PE への割り当て

Fig. 2 Routing area assignment for each PE.

述のように前処理, 後処理, そして探索処理の並列化が効率良く行えると共に, 今後配線対象の VLSI が大規模化した際に, プロセッサ数を増大することによって対応できるというメリットも生じる。

4.2 前処理, 後処理の並列化

改良線分探索法においては, 探索処理が90%以上の処理量を占めているので, その部分を並列化することが重要である。しかし, 前処理や後処理の部分も並列化することができれば, なお一層の高速化が図れる (特に, プロセッサ数が多い場合には, 逐次部分をなるべく減らすことが高い台数効果を得るという点で重要である)。

PROTON では, 前述のように配線領域を細長い帯状に領域に分割して, 各プロセッサに割り当てている。従って, 前処理に関しては, 各プロセッサは, 自分が受け持っている配線領域に属する障害物の周囲のコーナー・ポイント, エスケープ・ラインの生成を行う。このように前処理の並列化を行うと, プロセッサ間の通信が全く必要無いので, 高い台数効果が得られる。

後処理は, 新たに生成された配線経路の周辺のコーナー・ポイントやエスケープ・ラインを更新することになるので, その新たな配線経路が通っている配線領域を割り当てられているプロセッサが, それらの更新処理を行う。従って, 後処理に関しては, 新たな配線経路が通っている配線領域を担当しているプロセッサの台数だけの並列性がある。また, この処理も前処理と同様にプロセッサ間通信はほとんど無い (新たに生成された配線経路を各プロセッサに教えるために, 最初に一度だけ通信が必要となる) ので高い台数効果が得られる。

4.3 探索処理の並列化

PROTON では, ネット内並列性とネット間並列性の2種類の並列性を併用することによって高い台数効果を得ている。これは, 配線処理の中で最も重い処理部分である探索処理に内在する並列性である。以下に, PROTON におけるネット内並列性とネット間並列性について述べる。

4.3.1 PROTON におけるネット内並列性

PROTON の基本配線アルゴリズムである改良線分探索法の処理の中で最も重い部分は, エスケープ・ラインの中から配線経路を探索する部分である。従ってこの部分を並列化してネット内並列性を得ることが配線処理の高速化に効果的である。改良線分探索法における配線経路の探索は, 探索線と交差 (直交) する

エスケープ・ラインあるいは目標線を breadth-first-search の戦略で探すという処理を基本としている。つまり、あるレベルの探索線と交差するエスケープ・ラインや目標線（すなわち次のレベルの探索線）を列挙する必要がある。PROTON では、この処理をうまく並列化するために、前述のように配線領域を細長い帯状の領域に分割して、各プロセッサに割り当てている。そして、各プロセッサは、それぞれが割り当てられた配線領域内での線分の探索を独立に担当することによって、探索処理が並列化される²⁾。並列に探索している様子を図3に示す。例えば、PE0~PE3は、探索線 #1 と交差するエスケープ・ラインを並列に探すといった具合に探索処理を並列化している。

4.3.2 PROTON におけるネット間並列性

一般的に、LSI の自動配線は概略配線と詳細配線の2つのフェーズで構成されていることが多い。これは、概略配線を用いることによって、探索空間を絞り込んで詳細配線の処理時間を削減し、配線の順番が配線の品質、配線率に与える影響を減らすためである。

しかし、詳細配線は概略配線の領域内で行われるということを考えると、概略配線領域が重ならないネットは同時に詳細配線処理を行うことが可能であることがわかる。そこで PROTON では、概略配線の領域が重ならないそれらのネットを並列に配線することにより、ネット間の並列性を引き出している（正確に

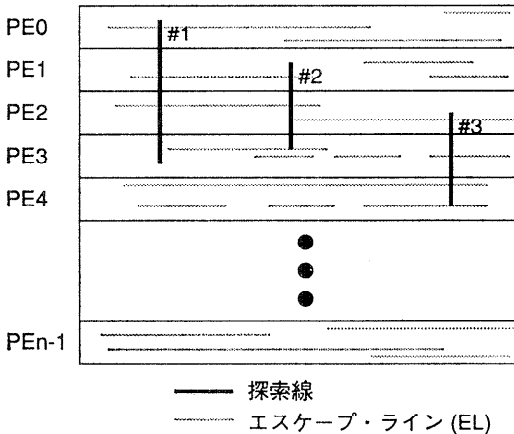


図3 ネット内並列性
Fig. 3 Intra-net parallelism.

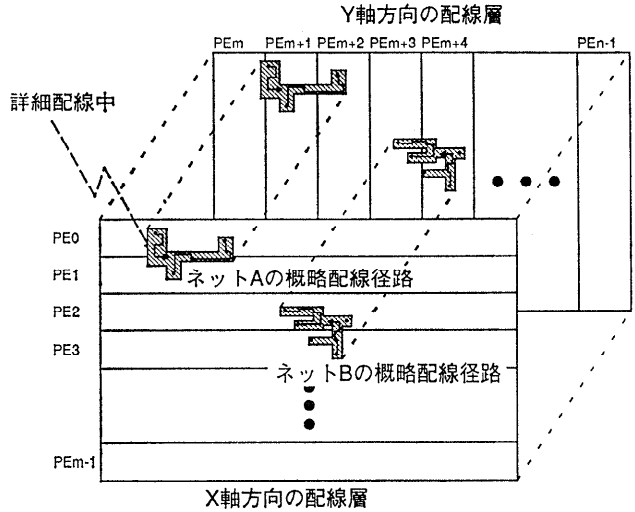


図4 ネット間並列性
Fig. 4 Inter-net parallelism.

は、概略配線領域を受け持つプロセッサが重ならないネットを並列に配線する)。

PROTON でどのようにネット間の並列性を用いているかを図4に示す。

同時に配線可能なネットのグループを1つの RU (Routing Unit) として扱い、1つの RU の配線が終わったらすべての PE が同期を取り、次の RU の配線へと進む。その様子を図5に示す。この図は PE 台数が8台の場合であり、9つのネット(#1~#9)は、各々の概略配線に従って3つの RU (RU0~RU2) に分けられている。

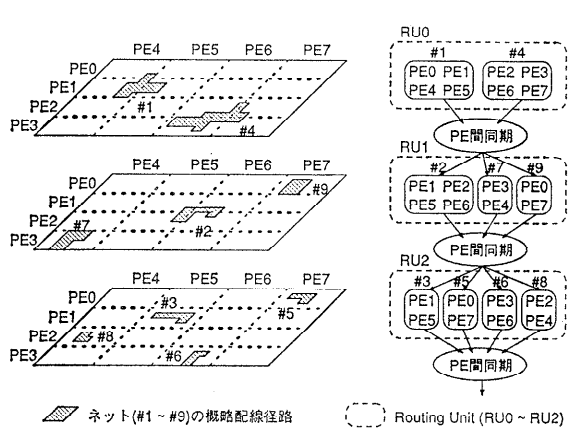


図5 ネット間並列性と RU
Fig. 5 Inter-net parallelism and RU.

4.4 ネット内、ネット間並列性の併用について

PROTON では、ネット間の並列性とネット内の並列性を併用している。これは、複数のネットを同時に配線する際に、各々のネットの配線についてはネット内の並列性を用いているということである。従って、概略配線領域が広くて同時に配線処理を行うことが可能なネットの本数が少ない場合（ネット間の並列性が低い場合）には、逆に高いネット内並列性が得られ、平均的に常に高い並列性が得られるという特徴がある。

逐次型計算機上で改良線分探索法を実行した場合の処理時間と、前処理や後処理の並列化、探索処理におけるネット内並列性を用いて並列処理した場合の処理時間および、さらにネット間並列性も用いた場合の処理時間を比較した結果を図6に示す。なお、図6では簡単のために各ネットの配線処理時間は一定となっているが、実際はばらつきがあり、プロセッサ間の同期のオーバーヘッドも生じる。

4.5 ネット間並列性の抽出

PROTON では、概略配線経路を受け持つプロセッサが重ならないネットを並列に配線することによってネット間並列性を得ている。従って、高いネット間並列性を得るためには、同時に配線することが可能なネットのグループ (RU) をどのように選ぶかが重要である。各ネットの間では概略配線経路の重なり以外には依存関係も無いので、RU の選択はスケジューリングというよりはマッピングに近い処理となる。このようなマッピングには、bin-packing アルゴリズム¹⁷⁾が適している。ただし、単純に概略配線経路の重なりだけでマッピングをすると、負荷分散のばらつきが生じるおそれがあるので PROTON では、それ以外にネットを構成するピンペア数 (端子数)、概略配線経路の面積等も考慮に入れるように bin-packing のアルゴリズムに改良を加えている。図7に bin-packing を用いて RU のマッピングを行った結果を示す。これは、横1行が1つの RU を表し、配線すべきネットを割り当てられたプロセッサを矩形で囲んで示している。図7を見ると、ネット間並列性に関しては、プロセッサの稼働率をかなり高くすることが可能であることがわかる。

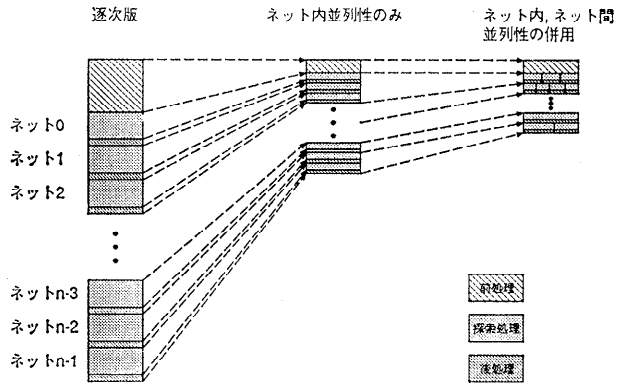


図6 並列化した場合の処理時間の比較
Fig. 6 Comparison of processing time between sequential router and parallel router.

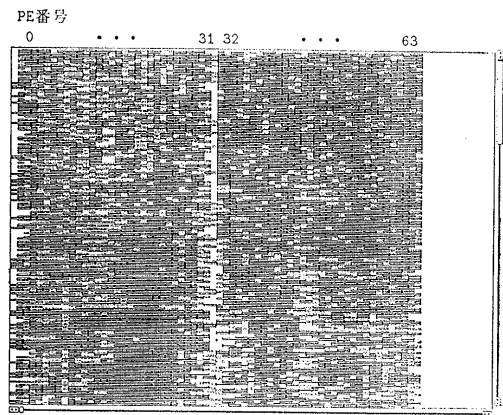


図7 bin-packing の結果
Fig. 7 Result of bin-packing.

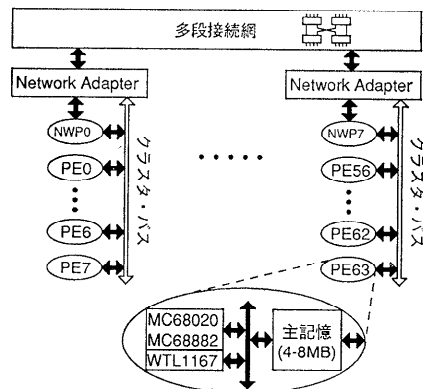


図8 Cenju のブロック図
Fig. 8 Block diagram of Cenju.

5. 並列マシン Cenju

PROTON を実装した並列マシン Cenju^{6),6)} は、次の特徴を有する。Cenju は、8個のクラスタで構成され、各クラスタ間はパケット交換の多段網で接続されている。各クラスタは、バスで接続された8台の要素プロセッサ (PE) で構成されている (図8参照)。メモリ空間は各 PE に分割された分散共有メモリの構成をとっている。また、各 PE 間の通信、同期は共有メモリや remote procedure call(rpc), barrier 等を用いて実現することが可能である。

6. PROTON の配線結果および評価

本論文では、以下の2種類のゲートアレイを用いて PROTON の評価を行った。

- ゲートアレイ A (2,832×2,742 格子, 2,495 ネット (4,454 ピンペア), 障害物 57,487 個)
- ゲートアレイ B (1,537×1,790 格子, 5,842 ネット (12,591 ピンペア), 障害物 151,982 個)

前述のように、PROTON では、ネット間並列性を抽出するために bin-packing アルゴリズムを改良したものをを用いている。それによって得られた RU の数は、ネット間の並列性の目安となる。ゲートアレイ A とゲートアレイ B の各々について RU の数を表4に示す。PE 台数が1台の場合は、RU の数はネットの総数と同じになる (PE が1台の場合は同時に1つのネットしか配線できないため)。また、ネットの総数を RU の数で割った値は、ネット間の並列性の平均値となる。従って、この表を見るとゲートアレイ A よりもゲートアレイ B の方がネット間の並列性の平均値が高いということがわかる。

表5に PROTON の配線結果 (実行時間, 配線率, 台数効果) を、図9に台数効果のグラフを示す。そして、図10に、ゲートアレイ B を配線した結果 (見やすさのために配線径路のみを示す) を示す。表5や図9を見ると、ゲートアレイ A では、台数効果が17倍程度

で飽和してしまっているが、ゲートアレイ B では、PE 数が64台の時に43倍の台数効果が得られている。ゲートアレイ A では、セルが配線領域の中心部により多く存在しているために、PROTON のような領域分

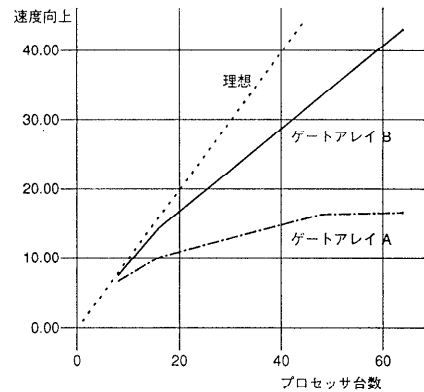


図9 台数効果
Fig. 9 Speedup.

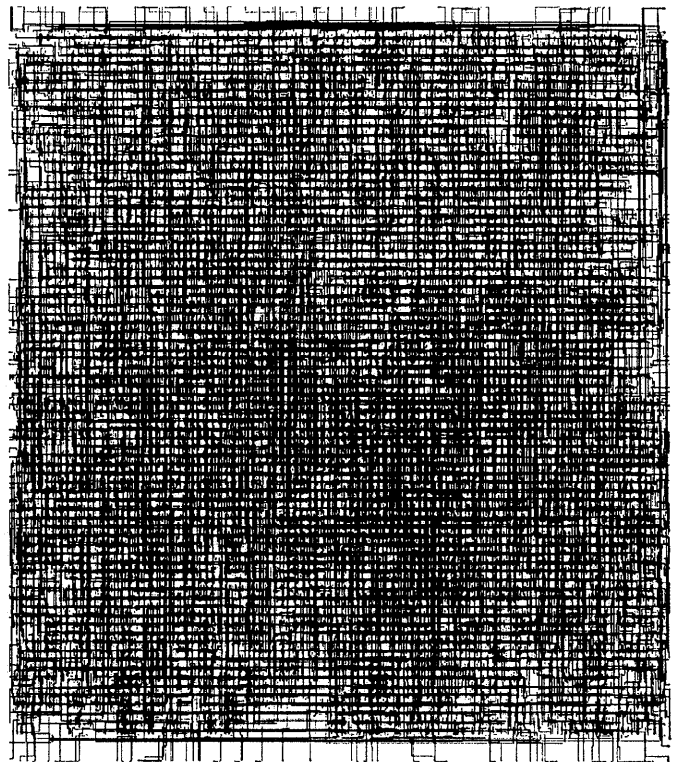


図10 ゲートアレイ B を配線した結果 (見やすさのために配線径路のみを示す)
Fig. 10 Routing result of gate array B.

表 4 RU とネット間並列性
Table 4 RU and inter-net parallelism.

PE 台数	ゲートアレイ A		ゲートアレイ B	
	Routing Units	ネット間並列性(平均)	Routing Units	ネット間並列性(平均)
1	2,495	1.00	5,842	1.00
8	1,518	1.64	2,098	2.78
16	1,164	2.14	1,391	4.20
32	968	2.58	1,035	5.64
48	883	2.82	874	6.68
64	875	2.85	845	6.91

表 5 配線結果
Table 5 Routing result.

PE 台数	ゲートアレイ A		
	配線時間(秒)	配線率(%)	速度向上
8	1,744.12	92.8	6.71
16	1,162.20	93.6	10.08
32	883.98	93.8	13.25
48	720.92	94.1	16.24
64	709.85	94.5	16.50

PE 台数	ゲートアレイ B		
	配線時間(秒)	配線率(%)	速度向上
8	33,461.12	97.6	7.50
16	17,497.42	97.7	14.33
64	5,831.07	98.5	43.02

割にもとづいた並列処理では負荷分散に偏りが生じてネット内並列性が損なわれたことが原因と考えられる。また、ゲートアレイ A では、各ネットの大きさが大きく、ネット間並列性があまり高くないということも原因として考えられる。一方、ゲートアレイ B は、配線領域に様に各セルが分布しており、各ネットの大きさも小さいのでネット内、ネット間共に高い並列性があり、43倍という高い台数効果が得られたと考えられる。また、表 5 を見ると、PE 台数を増やすにつれて配線率が若干良くなる傾向がある。これは RU のマッピングを行う際に、配線率を上げるために小さいネットを優先的に配線する等の工夫を加えており、PE 台数が多い場合(すなわち、1PE の受け持つ配線領域が小さい場合)の方が PE の割り当てと対象ネットの大きさの対応が正確になり、マッピングの工夫の効果が良く現れるためと考えられる。

7. おわりに

本報告では、(1)分散共有メモリ型並列マシン Cenju 上に実装、(2)実用規模の多層チャネルレス・ゲートアレイをサポート、(3)ネット内、ネット間の 2 レベルの並列性を併用、(4)従来の逐次型ルーターと同等の配線品質を保ちながら、高い並列性を抽出、という特徴を持つ並列 LSI ルーター PROTON の配線アルゴリズムならびに、その評価結果について述べた。そして、ネット内、ネット間の 2 レベルの並列性を併用したことにより、PE 台数 64 台の時に 43 倍の台数効果を達成した。

今後の課題としては、より高い配線率を得るために、引き剥し法等の検討が挙げられる。

謝辞 本研究の機会を与えて頂き、また有益な示唆を頂いた日本電気(株)研究開発グループ 石黒支配人、同 C&C システム研究所 山本所長、同 ULSI システム開発研究所 森野所長、佐々木 CAD 統括部長、ハードウェア製作に尽力された同 C&C システム研究所 三野輪一氏に深謝いたします。

参考文献

- 1) Yamauchi, T., Ishizuka, A., Nakata, T., Nishiguchi, N. and Koike, N.: PROTON: A Parallel Detailed Router on an MIMD Parallel Machine, *ICCAD-91*, pp. 340-343 (1991).
- 2) 山内, 中田, 石塚, 西口, 小池: 並列シミュレーションマシン Cenju 上の LSI ルーターの評価, 第 40 回情報処理学会全国大会論文集, 5L-6 (1990).
- 3) 山内, 中田, 石塚, 西口, 小池: 並列マシン Cenju 上の LSI ルーター—PROTON—, 第 44 回情報処理学会全国大会論文集, 3D-8 (1992).
- 4) 山内, 中田, 石塚, 西口, 小池: MIMD 型計算機上の LSI ルーター—PROTON—, 並列処理シンポジウム JSPP '92, pp. 445-452 (June 1992).
- 5) Nakata, T., Tanabe, N., Kajihara, N., Matsushita, S., Onozuka, H., Asano, Y. and Koike, N.: Cenju: A Multiprocessor System for Modular Circuit Simulation, *Computing Systems in Engineering*, Vol. 1, No. 1, pp. 101-109 (1990).
- 6) 中田ほか: 並列回路シミュレーションマシン Cenju, 情報処理 (30 周年記念特集号), Vol. 31, No. 5, pp. 593-601 (1990).
- 7) Lee, C. Y.: An Algorithm for Path Connections and Its Applications, *IRE Trans. Electron. Comput.*, Vol. EC-10, pp. 346-365 (1961).
- 8) 大附: 配線処理手法, 情報処理, Vol. 25, No. 10, pp. 1090-1099 (1984).

- 9) 橋, 鈴木, 大賀, 中島, 大附: 並列ルーティング・プロセッサの試作研究, 情報処理学会論文誌, Vol. 27, No. 6, pp. 639-647 (1986).
- 10) 河村, 進藤, 澁谷, 三渡, 大木, 土肥: 超並列配線マシン MAPLE-RP, 並列処理シンポジウム JSPP '91, pp. 373-379 (May 1991).
- 11) 石塚, 野田, 西口: 大規模 3 層ゲートアレイの配線手法, 信学技報, VLD 87-97 (1987).
- 12) 伊達, 大嶽, 瀧: 並列オブジェクトモデルに基づく LSI 配線プログラム, 並列処理シンポジウム JSPP '91, pp. 381-388 (May 1991).
- 13) 高橋, 佐々木: 競合プロセッサ群による配線問題の並列処理, 情報処理学会計算機アーキテクチャ研究会資料, 82-7 (Apr. 1990).
- 14) 羽根, 油井, 島本, 白川, 西口: 分散処理を用いた多層 VLSI 配線システム, 信学技報, CAS 91-90 (1991).
- 15) Suzuki, K., Ohtsuki, T. and Sato, M.: A Gridless Router: Software and Hardware Implementations, *VLSI '87*, pp. 121-131 (1987).
- 16) Lipski, W.: Finding a Manhattan Path and Related Problems, *NETWORKS*, Vol. 13, pp. 399-409 (1983).
- 17) Baker, B. S., Coffman, E. G. Jr. and Rivest, R. L.: Orthogonal Packings in Two Dimensions, *SIAM J. Comput.*, Vol. 9, pp. 846-855 (1980).
- 18) Yoshimura, T.: An Efficient Channel Router, *21st Design Automation Conference*, pp. 38-44 (1984).
- 19) Sagar, V. K. and Massara, R. E.: Exploiting Parallelism in Routing for a Hierarchical VLSI Design Environment, *Proc. 1989 European Conference on Circuit Theory and Design*, IEE, Brighton, U. K. (Sep. 1989).

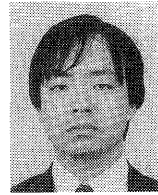
(平成 4 年 9 月 14 日受付)

(平成 4 年 12 月 10 日採録)



山内 宗 (正会員)

昭和 39 年生. 昭和 63 年東京大学大学院工学系研究科修士課程修了. 同年日本電気(株)入社. 現在同社 C&C システム研究所コンピュータ・システム研究部勤務. 並列計算機システム, CAD の研究に従事.



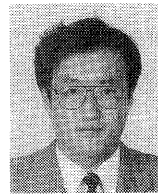
中田登志之 (正会員)

昭和 32 年生. 昭和 57 年京都大学大学院工学研究科修士課程修了. 昭和 60 年同大学院博士後期課程単位取得退学. 同年日本電気(株)入社. 京都大学工学博士. 現在同社 C&C システム研究所コンピュータシステム研究部研究課長. 並列計算機システムの研究に従事. 昭和 61 年度本学会論文賞受賞. 30 周年記念論文入賞. 電子情報通信学会会員.



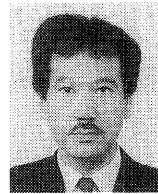
石塚 昭夫

昭和 35 年生. 昭和 59 年東京工業大学大学院理工学研究科修士課程修了. 同年日本電気(株)入社. 現在同社 ULSI システム開発研究所 CAD 基礎開発部主任. ゲートアレイ自動レイアウトシステムの開発に従事.



西口 信行

昭和 29 年生. 昭和 54 年大阪大学大学院工学研究科前期課程修了. 同年日本電気(株)入社. 現在同社 ULSI システム開発研究所 CAD 基礎開発部勤務. LSI の CAD システムの研究に従事. 電子情報通信学会会員.



小池 誠彦 (正会員)

昭和 22 年生. 昭和 45 年東京大学工学部電気工学科卒業. 昭和 47 年同大学院修士課程修了. 同年日本電気(株)に入社. 以来並列計算機システム, 論理シミュレーションエンジン (HAL), 並列回路シミュレーションマシン (Cenju) などの研究開発に従事. 最近では, 並列計算機のアーキテクチャ, CAD マシン, AI システム, ニューラルネットワークなどの研究に興味を持つ. 現在, 同社 C&C システム研究所コンピュータシステム研究部長. 工学博士. 著書「CAD マシン」(電子情報通信学会編, オーム社). 電子情報通信学会会員. 昭和 59 年度情報処理学会論文賞受賞. 平成 2 年本学会創立 30 周年記念論文入選.