

第五世代コンピュータプロジェクトと並列 LSI-CAD *

瀧 和男 †

神戸大学工学部 情報知能工学科 ‡

概要

第五世代コンピュータプロジェクトにおける並列 LSI CAD プログラムの研究開発について概要を報告する。処理速度の向上を目指した論理シミュレーションとブロック間配線では、256 台のプロセッサを用いて、1 台での実行に比べ、それぞれ 166 倍と 92 倍の良好な速度向上を実現した。前者では、タイムワープと呼ぶ分散時刻管理方式の実装により、後者では、多数の小さなオブジェクトの集まりとして配線問題を定義することにより、大きな並列性を実現することができた。また設計品質の向上を目指したセル配置では、組合せ最適化問題の確率的解法であるシミュレーテッドアニーリングについて、新しい並列アルゴリズムを提案した。良解への収束性が向上し、配置の設計品質向上が確認された。

第五世代コンピュータプロジェクトで開発された KL1 言語を用いることで、複雑なプロセス構造を持つ並列プログラムの開発が可能となった。

Abstract

Overview of the parallel LSI-CAD programs in the "Fifth Generation Computer Systems Project" are reported.

Time warp logic simulator and object-oriented parallel router, aiming at high speed processing, attained 166-folds and 92-folds speedup. Small granular object-oriented problem modeling extracted large parallelism. A new concurrent simulated annealing algorithm was also developed, which realized automated temperature scheduling and very good convergence capability.

KL1 programming, developed in the FGCS project, made it possible to develop large complexed parallel LSI-CAD programs.

1 まえがき

第五世代コンピュータプロジェクトおよび、その後継プロジェクトである第五世代コンピュータの研究基盤化プロジェクトでは、近未来の知識情報処理のための基礎技術確立を目指し研究開発を行ってきた。そこでは論理型言語の技術を中核に据え、並列処理のハードウェアから、オペレーティングシステム、各種知識処理の技術、知識処理/並列処理応用ソフトウェアに至るまで、総合的な研究と開発に取り組んできた [Kurozumi 92] [Furukawa 92] [Uchida 92] [FGCS 94]。並列推論マシン PIM は、その中で開発された MIMD 型の大規模並列計算機であり、プロジェクトの中核技術である並列論理型言語 KL1 を効率良く実行するための専用マシンであった [Taki 92a] [Chikayama 92] [Ueda 90]。

並列推論マシンシステムの評価と、知識処理および並列処理の研究のため、多くの応用プログラムが開発されてきたが [Nitta 92]、その中の重要な応用領域の一つに LSI CAD が含まれる。LSI の集積度が指数関数的に増大するのにもとない、より高速な自動設計、さらに LSI の高速動作を実現するより高

精度な自動設計が望まれるようになり、並列処理と知識処理を適用するのに極めて適当な分野と考えられるようになった。

そこで並列推論マシン PIM の上では、知識処理の研究に重点を置いた LSI CAD プログラム (例えば [Minoda 92] 他、全体の要約は [Nitta 92]) と、並列処理の研究に重点を置いたプログラム [Date 92a] について研究開発を進めた。知識処理と並列処理を同時に実現するにあたっての難しさについては [瀧 93] に詳述されている。本稿では紙面の関係から、並列処理に重点を置いたプログラムを取りあげ、論理シミュレーション、セル配置、ブロック間配線などの 4 種に関し、それらの特徴と研究開発の経緯、評価結果について紹介する。

以下では第 2 章で、LSI CAD プログラムの並列化にあたり目指した点と、4 種のプログラムの特徴および関連する研究について要約し、合わせて PIM の概要について述べる。3 章以下の各章で、それぞれのプログラムについて解説する。

2 プログラム開発の概要と特徴

2.1 並列化の取り組みと特徴

本稿で取り上げる 4 種類の LSI CAD プログラムは、並列処理によって処理時間を短縮し、より大規模の問題に対応可能とすること、もしくは並列処理

*Parallel LSI-CAD Research in the Fifth Generation Computer Systems Project

†Kazuo TAKI

‡Department of Computer and Systems Engineering, Faculty of Engineering, Kobe University

を活用して設計品質を高めることを目的としている。

これらは並列推論マシン、マルチ PSI および PIM [Taki 92a] の上に試作されたものであるが、プログラムの特徴としては、第一に、分散メモリ構造を持つある程度規模の大きい MIMD 型の並列計算機での実行を意識したこと、第二に、並列論理型言語 KL1 で記述されていることである。前者は、問題のモデル化とアルゴリズムの設計に影響を与えている。後者は、従来言語ではプログラム記述やデバッグの手間が大き過ぎるようなプログラム構造の取扱いを可能にするという特徴を持っている。後に述べる並列オブジェクトモデルに基づくプログラムは、KL1 言語が使えたことによって、自然な記述と効率の良い実行が可能になったものと考えている。[瀧 92b]

LSI CAD プログラムの並列化の中でも、数百以上のプロセッサを用いて高い並列処理効率を目指す場合には、問題の解き方が明らかかなもの、あるいは問題のモデルが明確に規定できるものを選ぶよう心がけた。問題のモデル化自体が研究対象であって、しかもしばしばモデルの変更が必要な場合には、大規模並列処理で効率を上げるための検討を並行させることは極めて難しくなる。

2.2 4種の並列 LSI CAD プログラム

ここでは、次章以下で紹介する 4 種類のプログラムの特徴をまとめる。また第五世代コンピュータプロジェクト終了後の関連する研究の継続、成果についても簡単に触れる。

・論理シミュレーション：並列処理による高速処理性の実現を目指している。シミュレーション時刻の管理方式が処理効率を高める上で重要なため、分散メモリ構造の並列計算機に適する方式として、徹底した分散時刻管理を行なうタイムワープ方式を実現した。同方式の並列論理シミュレータを実装し厳密な評価まで行なったのは、著者の知る限りでは本システムが最初である。負荷分散方式等を工夫し、高い並列処理効率を実現している。KL1 言語を用いることにより、プログラム中の通信や同期の記述が簡素化され、バグの少ないプログラムを実現するのに有効であった [Matsumoto 92a, 松本 92b, Matsumoto 93]。

プロジェクト終了後は神戸大学にて研究を継続しており、汎用並列計算機上での実装、評価を行なっている [石原 94a]。マルチプロセッサ型ワークステーション・サーバ機上で、実用レベルの良好な並列シミュレーション性能が得られている。本ワークショップにて関連の報告がある [石原 94b]。

・セル配置：並列処理を活用して従来よりも優れた設計品質の実現を目指している。これを実現する

ために、組合せ最適化問題を確率的に解く、新しい並列アルゴリズムを開発し試験実装した。アルゴリズムは、シミュレーテッドアニーリング (SA) の並列版であるが、逐次 SA アルゴリズムで良い解を得るのに必須だった「温度」と呼ばれるパラメータの制御が、本並列 SA では一切不要になるという特徴を備えている。これにより、多様な配置問題において、疑似最適解への収束性が従来より向上することが期待されている。この並列 SA アルゴリズムは、分散メモリ構造の並列計算機で効率良く実行できる [木村 90, Kimura 91, Date 92a, 伊達 93b]。

プロジェクト終了後は、同じく神戸大にて関連研究を継続中である。ブロック配置問題への適用と、温度並列 SA アルゴリズム自体の最適化能力を実験により評価している [小西 94a]。その結果、1 CPU だけによる並行実行の場合でも、温度並列 SA は従来の逐次 SA に比べて、同じ時間をかけてより良い解へ収束するという極めて興味深い成果を得ている。本ワークショップで関連の報告がある [小西 94b]。

・並列オブジェクトモデルに基づくブロック間配線：並列処理による処理の高速化と大規模データへの適用を目指している。問題のモデル化段階で、より大きな並列性を内在させることができるように、配線の断片一本一本をすべて独立なオブジェクトとする特徴的なモデル化を行なった。そして、それらがメッセージを交換しながら短い配線経路を決定してゆく分散アルゴリズムを設計した。KL1 言語を用いることにより、このような多数の並列オブジェクトからなるプログラムの記述を効率良く行なうことができた。並列推論マシン PIM で実行することにより、並列オブジェクト間で、ある程度頻繁なメッセージ交換があるにもかかわらず、良好な並列処理効率が得られている [伊達 92b, Date 93a]。

タイムワープに基づく配線：並列処理による処理の高速化が最終目標であるが、当初の研究目的は、タイムワープがシミュレーション以外にも使える汎用的な並列性制御方式であることを示すことにあった。配線処理を並列化すると、一般にはネット間の競合の問題で、逐次配線に比べて配線品質が劣化する傾向が見られる。逐次配線で得られる配線結果との同一性を保証しながら、ネット間の並列配線を実現するためにタイムワープ機構を利用した。ちょうど論理シミュレーションにおいて、本来は厳密に時刻順を守ってゲート評価すべきところを、正しいという見込みの上で時刻を気にせずに複数ゲートを並行評価し、間違いが見つかった時にだけロールバック (履歴巻戻し、再実行) するという方法を、時刻では

なく配線順序に対して適用するものである。複雑な並列アルゴリズムであるが、KL1 言語によりバグ取りと性能調整が可能となった [松本 94a]。本ワークショップで関連の報告がある [Matsumoto 94b]。

2.3 実行環境

本報告の中で実行・評価に使用した並列推論マシンは PIM/m と呼ばれるタイプで、256 台のプロセッサが 2 次元格子状のネットワークで接続されている。プロセッサ間通信はメッセージ交換によっているが、メッセージの送受は KL1 言語処理系が暗黙のうちに行ない、KL1 のプログラマはメッセージを陽に取り扱う必要がない。この点が、複雑な構造を持つ並列プログラムの記述を容易にするだけでなく、負荷分散記述の手間も軽減している [Taki 92a]。分散メモリ構造の計算機では、プロセッサ間通信の手間(時間)は一般に大きいと言われているが、言語処理系を効率良く実装する努力により、実用に耐え得る通信性能を実現している [瀧 92b]。要素プロセッサは、論理型言語の実行効率を高めるよう特別に開発されたものであり、65nS のクロックで動作する。プロセッサ毎に 80MB の大容量メモリを持つことも特徴である。

3 論理シミュレーション

3.1 背景

論理シミュレーションの目的は、設計回路の論理機能、および信号伝播タイミングの検証である。これは、LSI 設計の中で最も計算時間を必要とする工程の一つであることから、高速化が強く望まれている。論理シミュレータを並列プログラムとして実現する方法は、柔軟性の高い高速シミュレーションを実現する有望な方法である。

並列論理シミュレーションは、並列離散事象シミュレーションの問題として扱われる。並列離散事象シミュレーションでは、複数のオブジェクトが互いにメッセージを交換し、その状態を変化させていくことでシミュレーションが進行するように問題をモデル化する。メッセージは事象情報を持ち、その生起時刻が刻印されている(タイムスタンプ)。正しいシミュレーション結果を得るためには、各オブジェクトで時刻順にメッセージを処理していかなければならない。このため、何らかの時刻管理機構が必要になる。

本プログラムでは、分散メモリ構造を持つ MIMD 型計算機で効率的に動作することを目指し、完全な分散時刻管理機構としてタイムワープ機構を採用した。タイムワープ機構では、ロールバック処理が必

要であり、そのオーバーヘッドは大きいと予想されてきた。しかし、MIMD 計算機上での詳細な評価はほとんど行われていない。もし、ロールバックのオーバーヘッドを小さく抑えることができれば、タイムワープ機構は分散メモリ型 MIMD 計算機に適した時刻管理機構になると考えられる。このため、適切な負荷分散戦略を採用し、局所メッセージスケジューラ、アンチメッセージ削減機構を追加することで、効率的な並列論理シミュレーションの実現を試みた [松本 92b]。

3.2 システムの概要

本シミュレータは、組み合わせ回路、順序回路(非同期回路も含む)ともに扱うことができる。信号値モデルとしては、Hi, Lo, X(不定)を用いる 3 値モデルを、また、遅延値モデルとしては、各ゲートに異なる値を割り当てることができる非単一遅延モデルを採用した。

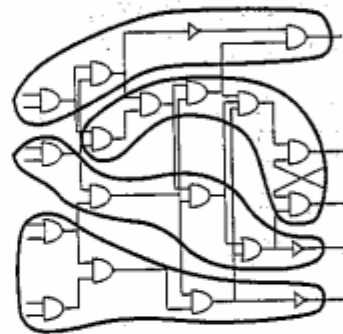


Fig.1 Cascading-Oriented Partitioning

図 1: 縦割り指向回路分割

本シミュレータでは、ゲートをオブジェクトとして扱っている。タイムワープ機構 [Jefferson 85] を用いた論理シミュレーションでは、各オブジェクトは時刻順にメッセージが到着するであろうという予測に基づき、履歴を保存しつつ、メッセージ処理を進める。もし、メッセージが遅れて到着した場合は履歴を巻き戻し(ロールバック)、そのメッセージが本来到着すべきであった時刻から処理をやり直す。また、誤って送信されてしまったメッセージに対しては、それらを取り消す役割を持つアンチメッセージを送り出す。

実際のプログラムでは処理効率を少し改善するため、ゲートを直接オブジェクトとして実現するのではなく、オブジェクトの状態をデータとして管理しオブジェクトの動きを模擬するプロセスを用意して、プロセッサに一つずつ配置した [瀧 92b]。別ブ

ロセッサ上にあるゲートオブジェクトとのメッセージ交換は、これらのプロセス間のメッセージ交換となる。

ロールバック処理のオーバーヘッドは、無視できない大きさであるため、高速シミュレーション実現のためには、このオーバーヘッドを削減する必要がある。また、PIMのような分散メモリ計算機を用いる場合、プロセッサ間通信オーバーヘッドも小さく抑える必要がある。本シミュレータでは、縦割り指向戦略に基づく負荷分散、局所メッセージスケジューラ、およびアンチメッセージ削減機構によって、プロセッサ間通信およびロールバックのオーバーヘッド削減を行っている。

負荷分散を行う時の目標は、(1) 負荷の均一化、(2) プロセッサ間通信低減、(3) 高い並列性抽出の3点となる。本シミュレータの開発に際して提案、採用した縦割り指向戦略は、縦方向に連なったゲートを切り出す操作を基本とした静的回路分割戦略である(図1)。この戦略は、短い処理時間で上記3点の目標をある程度満たす手法と考えられる。

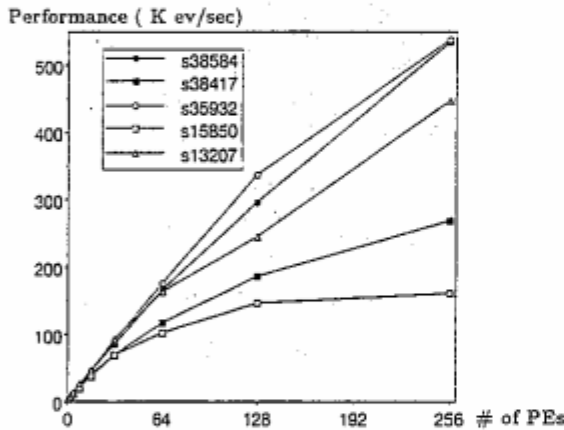


Fig:2 Speedup in Logic Simulator

図 2: シミュレーション性能とプロセッサ台数

3.3 計測結果

ISCAS'89 のベンチマークから 5 つの順序回路について、PIM/m 上でシミュレーションを行った。回路のゲート数は、およそ 1 万 2 千から 3 万 2 千である。

図 2 に性能とプロセッサ数の関係を示す。256 プロセッサを用いた場合、1 プロセッサの性能に比べて最大で 166 倍の性能向上(これを台数効果という)と、537K イベント/秒のシステム性能を得た。この台数効果は非常に良好であり、タイムワープ機構が効率良く動作したことを示す。システム性能につい

ても、ソフトウェア論理シミュレータとしては極めて良好な値と考えられる。なお、台数効果が頭打ちとなった回路については、解析の結果、問題の持つ並列性自身の小さいことが主な要因と分かっている [Matsumoto 92a]。

他の時刻管理法としての性能比較として、分散時刻管理法であるコンサーバティブ法と、集中時刻管理法のタイムホイル法のそれぞれとの性能比較も行った。前者はデッドロック回避のオーバーヘッドのため、性能が桁違いに悪く、後者はプロセッサ数 8 以下で本方式より高い性能を示すものの、プロセッサ数の多い並列実行では返って性能が低下することが分かり、本方式の優位性が確認された [松本 92b]。さらに、並列性が不十分な回路のシミュレーション時にロールバックが多発することを抑制し、ロールバックによる性能低下を軽減する方法についても報告している [Matsumoto 93]。

4 セル配置

4.1 背景

スタンダードセルと呼ばれる種類の LSI では、レイアウト設計において、ゲートやフリップフロップ等に対応するセルをチップ領域上に配置し、セルの端子どうしの配線経路を決定して目的の論理回路を構成する。このとき配線長をなるべく短くして、小さなチップ面積が実現できるように個々のセル位置を決定することが求められる。これがセル配置問題である。

LSI を構成するセル数は非常に多く、配線領域を最小化するためには膨大なセル配置の組合せを調べなければならない。このような組合せ最適化問題を解くのに、シミュレーテッドアニーリング (SA) [Kirkpatrick 83] と呼ばれる一種の確立的アルゴリズムが有効であることが知られている。

しかしながら、逐次の SA アルゴリズムでは、「温度」と呼ばれるパラメータの制御を注意深く行わなければ、最適に近い解に到達することができない。そこで、並列処理を活用して温度の制御をまったく不要にする新しい並列 SA アルゴリズムを提案した [木村 90] [Kimura 91]。これをセル配置問題に適用し、配置設計の品質向上を目指した。アルゴリズムが自動的に適切な温度スケジュールを構成するため、対象問題の性質が変化する場合にも、常に疑似最適解への良い収束性が期待される。本並列 SA は汎用的なアルゴリズムであり、規模の大きい組合せ最適化問題に広く適用可能である。

4.2 並列 SA アルゴリズム

本並列 SA は、別名、温度並列 SA という名で呼ぶこともある。異なるプロセッサに対して、高温から低温まで別々の一定温度を与え、それぞれで乱数を用いた配置改善を並行に進める。各プロセッサは、異なる中間解を持つことになる。に対してそして隣接する温度を担当するプロセッサ間で、一定期間ごとに、それぞれの配置の中間結果を比較する。このとき、温度の高い方の配置結果が低い方より良い評価値を示すならば、これらのプロセッサ間で配置結果を交換する。逆の場合は、確率的に交換を行なう。

この結果、高温を担当するプロセッサでは大域的な配置改善が、低温のプロセッサでは局所的な改善が行なわれ、処理の進行に伴って良い配置結果が適宜選択されることになる。良い配置結果は、図 3 に示すように、高温のプロセッサからしだいに低温のプロセッサへ配置改善をともしないながら移動してゆく。最終的には、一番低温を担当するプロセッサ上にもっとも配置改善の進んだ結果が現れることになる。このようにして、良い解に至るための温度スケジュールの自動化が実現される。

本アルゴリズムは時間方向に一様であるため、利用者は任意の時点で処理を終了することができ、その時点での最良の (または最良に近い) 配置結果を最低温度を担当するプロセッサ上に得ることができる。本アルゴリズムの理論と取束性の証明が報告されている [木村 90] [Kimura 91]。

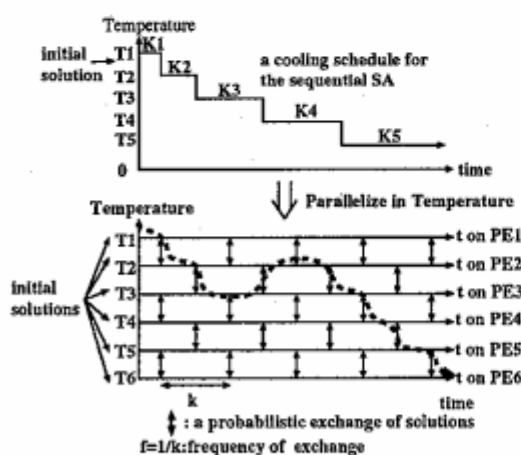


Fig:3 Temperature Parallel SA Method

図 3: 温度並列シミュレーテッドアニーリングアルゴリズム

4.3 システムの概要

本配置プログラムでは、マクロブロックを含まないスタンダードセル型の LSI を対象としている。配置対象のセルは、高さが一定で幅がまちまちな形状を持ち、それらの集合であるセル列がチップ上に複数構成されるように配置する。ランダムな初期配置から始めて、総配線長の見積り値が最小になるよう、並列 SA で配置を改善する。SA をスタンダードセルの配置に適用する場合の各種パラメータ等の設定方法は、[Sechen 85] に準じている。配置改善のためのセル位置の交換または移動は乱数を用いて行ない、高温域ではチップ全体を対象とする移動・交換を、低温域ではセル列内に限定した移動・交換を行なっている。

4.4 実行結果

初期の実験結果としては、グラフの分割問題に適用して逐次 SA より良好な解を得ている [木村 90]。

セル配置では、MCNC ベンチマークのデータ (125 セル, 147 ネット) 等を使用し、プロセッサ数 (温度数) を 63 とした。内側のループ (新しい配置の生成と受理の可否の計算) 100 回に対して 1 回の割合で隣接温度間の解交換を試み、ループ 20,000 回実行後の配置結果では、初期配置に比べてチップ面積が 56% の減少となった [Date 92a]。チップ面積はこの時点では、他所から報告済みの最良値よりははまだ大きかった。

その後の改良により、752 セル, 904 ネットのデータ (primary1) において、最小のチップ面積を実現した [伊達 93b]。処理時間の短縮が次の課題であるが、同期を必要としない単純な計算の部分で KL1 の処理が遅いことが分かっており、改良すべき点となっている。

5 ブロック間配線

5.1 背景

LSI の配線設計も、LSI 設計の中で多大な計算時間を必要とする工程の一つである。ますます大規模化する LSI を対象に高速の配線設計を実現するため、規模の大きな並列計算機で実行することのできる並列配線プログラムの実現を目指した。

ここでは、配線問題の中でも、配置済みブロックの端子間の配線を主な対象としたブロック間配線を扱っている。

配線アルゴリズムは、a. 計算の効率がよく、b. 配線率が高く、c. 適用範囲の広いことが望まれるが、逐次計算機用に工夫されたアルゴリズムで、そのまま大規模な並列処理に適用できるものはほとんど存

在しない。上記3点の要求を満たしつつ、大きな並列性を内在する配線アルゴリズムの実現を目指したいが、まったく新しいアルゴリズムの設計は容易ではない。そこで、基本的なアルゴリズムは、上記3点を満たす逐次アルゴリズムの、予測線分探索法[北沢 85]に基づきながら、問題のモデル化を全く見直すことにより、並列実行の可能性を高める方針を取った。

すなわち、並列オブジェクトモデルに基づく全く新しい配線問題のモデル化とプログラム化を試み、大規模な並列計算機での効率の良い並列実行を目指した。

5.2 並列オブジェクトモデルに基づくプログラミング

問題のモデル化：大きな並列性を内在することができるよう、比較的粒度の小さいオブジェクトが多数集まって、互いにメッセージを交換し合いながら問題を解いてゆくようなモデル化を行なった。すなわち、配線格子に存在するすべての既配線、未配線をオブジェクト(プロセス)に対応させ、互いに直交するオブジェクト間でメッセージ交換しながら、短い配線経路を探索する(図4)。

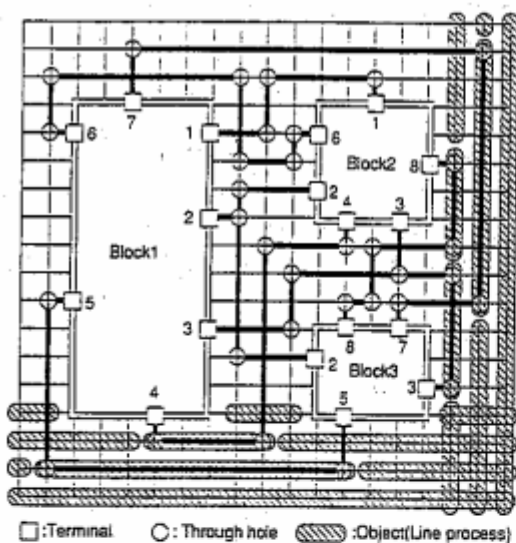


Fig:4 Concurrent Object-Oriented Model for a Routing Problem

図4: 並列オブジェクトモデルに基づく配線問題のモデル化

並列アルゴリズム：基となった予測線分探索法は逐次アルゴリズムであるが、これを個々のオブジェクトが独立に並行動作できる分散アルゴリズムに作り直した[伊達 92b]。

予測線分探索法では、配線の開始点から終了点に向かって短い経路を探索するとき、折れ曲がり1段分の先読みを行ないながら、先読み結果のもっとも良い方向へ経路を決めてゆく。一種の山登り法であり、必ずしも最短経路とならないが、折れ曲がりの少ない配線結果が得られる。二重探索を避けるフラグとバックトラックなどを導入し、経路が存在すれば必ず発見できることを保証している。

本並列アルゴリズムでは、開始点と接続する未配線オブジェクトに対し、「ある終了点を目標に配線せよ」という配線要求メッセージを送ることで処理が開始される。同オブジェクトは、直交する全ての未配線オブジェクトに、「もっとも終了点に近い点を報告せよ」という先読み依頼メッセージを渡し結果を待つ。全ての応答があると、もっとも終了点に近いことを報告したオブジェクトと接続関係を結んで自らは既配線となる。接続した先の未配線オブジェクトに対しては配線要求メッセージを送る。これを繰り返して、山登り法で終了点を目指す[伊達 92b]。

未配線オブジェクトは、配線が決まると、既配線と未配線のオブジェクトに分裂する。バックトラックがあると、分裂していたものが合併する。このようにオブジェクトの数と接続関係は、動的に変化する。

本プログラムでは、2種類の並列性を実現している。先読み部分の並列処理と、異なるネット(異なる始点・終点間の配線)の同時配線である。

5.3 システムの概要

本システムは、ブロック間配線を対象として、配線格子を用いる2層配線を行なう。オブジェクトは、KL1のプロセスとして実現した。配線を表現するオブジェクトの他に、配線格子に対応するオブジェクトを設け、配線格子上の複数の配線オブジェクトの管理と、メッセージの仲介の役割を持たせた。また大規模データに対応できるように、オブジェクトの接続構造を工夫している[Date 93a]。

負荷分散に関しては、通信の局所性と負荷の均等化を考慮し、配線格子オブジェクトとその上の配線オブジェクトをグループとして、グループ単位に巡回的にプロセッサへ割り付けた。

5.4 計測結果

LSIの実データを用いて、次の三つの観点からシステムを評価した。(1) データ規模と台数効果との関係、(2) 並列性と配線率との関係、(3) 汎用計算機との性能比較。

図5は、本配線プログラムをプロセッサ数を変化させて実行させたときの台数効果を表している。256

台のプロセッサを用いて、最大 92 倍の良好な台数効果を得た [FGCS デモ資料 92]。しかしながら、並列性を上げるため同時配線するネットの数を増すほどに、ネット同士の配線領域の取り合いにより、配線率が低下する傾向が見られた [伊達 92b]。

この問題を解決するためのアルゴリズムの改良を行ない、最終的に、逐次アルゴリズムと同じ配線率、配線結果を保証できるようになった。このとき、格子サイズ 2746×3643、ネット数 556 の実 IC データについて、IBM3090 上の同様のアルゴリズムの FORTRAN プログラムに比較し、約 6 倍高速であるとの測定結果を得た。この時の台数効果は、256 プロセッサで約 37 倍であったが、ネット数の大きいデータを用いることによりさらに増加の見込みである [Date 93a]。

Speedup

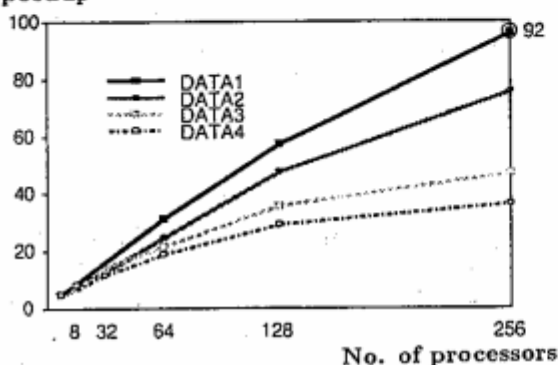


Fig.5 Speedup in Routing

図 5: 配線問題における台数効果

6 むすび

第五世代コンピュータプロジェクトの並列推論マシン上に試作された LSI CAD プログラムのうち、並列処理の活用に重点を置いたプログラムについて、概要を報告した。

処理速度の向上を目指した論理シミュレーションとブロック間配線では、256 台のプロセッサを用いて、1 台での実行に比べ、それぞれ 166 倍と 92 倍の良好な速度向上を実現した。前者では、タイムワープと呼ぶ分散時刻管理方式の実装により、後者では、多数の小さなオブジェクトの集まりとして配線問題を定義することにより、大きな並列性を実現することができた。また設計品質の向上を目指したセル配置では、組合せ最適化問題の確率的解法であるシミュレーテッドアニーリングについて、新しい並列アルゴリズムを提案した。良解への収束性が向上し、配置の設計品質向上が確認された。

第五世代コンピュータプロジェクトで開発された KL1 言語を用いることで、複雑なプロセス構造を持つ並列プログラムの記述が容易になり、バグの少ないプログラム記述が可能となった。また、負荷分散の記述を変えながら並列処理効率を改善することの手間も改善された。多数の小さなオブジェクトでプログラムを構成する手法は、KL1 言語の特徴を反映したものの一つである。また並列推論マシン PIM を用いた場合、プロセッサ間のメッセージ通信に関する応答性が良く、データ量の小さいメッセージがある程度高い頻度で発生する応用 (例えばブロック間配線) でも、効率良く実行できることが確認された。

絶対性能に関しては、配線問題において、汎用大型計算機に比べ、同じ設計品質で 6 倍の高速性が確認されたが、これはまだ十分に満足できる性能比ではない。KL1 言語の実装効率の向上と、負荷分散手法等の改善による並列処理効率のさらなる向上が、今後の課題として残されている。この中で前者については、汎用計算機上の KL1 処理系 KLIC [FGCS 94] として、改善が試みられている。

全体として、従来の常識で考えると、複雑過ぎて開発がたいへんではないかと思われるようなアルゴリズムやプログラム構造が、過大な苦勞なく、並列プログラム化でき負荷分散の調整もできるようになってきたと考えている。上述の課題について改善が進むと、いよいよ LSI CAD の様々な部分について、実用を目指した効率の良い並列プログラムの開発が現実のものになると期待している。

謝辞

本稿の大半は、著者が (財) 新世代コンピュータ技術開発機構 (ICOT) に在籍中の仕事、および著者の所属した LSI CAD グループの仕事に基づいている。ICOT の関係各位に深く感謝する。

参考文献

- [Chikayama 92] T. Chikayama. Operating System PIMOS and Kernel Language KL1. In *Proc. of the Int. Conf. on Fifth Generation Computer Systems*, 1992.
- [Date 92a] H. Date, Y. Matsumoto, K. Kimura, K. Taki, H. Kato and M. Hoshi. LSI-CAD Programs on Parallel Inference Machine. In *Proc. Int. Conf. on Fifth Generation Computer Systems 1992, (FGCS'92)*, July 1992, pp.237-247.
- [伊達 92b] 伊達博, 大嶽能久, 瀧和男: “並列オブジェクトモデルに基づく LSI 配線プログラム”, 情処論, Vol.33, No.3 (1992), pp. 378-386.
- [Date 93a] H. Date and K. Taki. A Parallel Lookahead Line Search Router with Automatic

- Ripup-and-reroute. In *Proc. European Design Automation Conference EDAC'93*, March 1993.
- [伊達 93b] 伊達博, 瀧和男: “温度並列シミュレートッド・アニーリング法に基づくスタンダードセル配置プログラム”, DA シンポジウム'93 論文集, 1993年8月, pp.173-176.
- [FGCS 94] *Proc. International Symposium on Fifth Generation Computer Systems 1994*, December 1994.
- [FGCS デモ資料 92] 第五世代コンピュータ国際会議 1992, デモンストレーション説明資料, pp.63-ICOT, Tokyo, 1992.
- [福井 89] 福井: “バーチャルタイムアルゴリズムの改良”, 情処論, Vol.30, No.12 (1989), pp. 1547-1554.
- [Furukawa 92] K.Furukawa. Summary of Basic Research Activities of the FGCS Project. In *Proc. of the Int. Conf. on Fifth Generation Computer Systems, 1992 (FGCS'92)*, Tokyo, July 1992, pp.20-32.
- [石原 94a] 石原義勝, 瀧和男, 高垣雅弘: “汎用並列計算機上のバーチャルタイムによる並列論理シミュレータ”, DA シンポジウム'94 論文集, 1994年8月, pp.235-240.
- [石原 94b] 石原義勝, 瀧和男: “汎用並列計算機上の並列論理シミュレータ -タイムワープに基づく方法とルーズな時刻同期による同期的方法-”, FGCS'94 ワークショップ「並列/分散処理による LSI-CAD」論文集, 1994年12月
- [Jefferson 85] D. R. Jefferson. Virtual Time. *ACM Transactions on Programming Languages and Systems*, Vol.7, No.3 (1985), pp. 404-425.
- [木村 90] 木村宏一, 瀧和男: “時間的一様な並列アニーリングアルゴリズム”, 信学会ニューロコンピュータ研究会資料 NC90-1, 1990年5月.
- [Kimura 91] K. Kimura and K. Taki. Time-homogeneous Parallel Annealing Algorithm. In *Proc. IMACS'91*, 1991. pp. 827-828.
- [Kirkpatrick 83] S.Kirkpatrick, C.D.Gelatt and M.P.Vecchi. Optimization by Simulated Annealing. *Science*, Vol.220 No.4598, 1983, pp.671-681.
- [北沢 85] 北沢仁志: “高配線率線分探索の一手法”, 情処論, Vol.26 No.11, 1985年11月, pp.1366-1375.
- [小西 94a] 小西健三, 瀧和男: “温度並列シミュレートッド・アニーリング法の評価 - LSIブロック配置問題に適用して-”, DA シンポジウム'94 論文集, 1994年8月, pp.223-228.
- [小西 94b] 小西健三, 瀧和男: “温度並列シミュレートッド・アニーリング法の応用 -ブロック配置問題への適用と評価-”, FGCS'94 ワークショップ「並列/分散処理による LSI-CAD」論文集, 1994年12月
- [Kurozumi 92] T.Kurozumi. Overview of the Ten Years of the FGCS Project. In *Proc. FGCS'92*, Tokyo, July 1992, pp.9-19.
- [Minoda 92] Y.Minoda, S.Sawada, Y.Takizawa, F.Maruyama and N.Kawato. A Cooperative Logic Design Expert System on a Multiprocessor. In *Proc. FGCS'92*, Tokyo, July 1992, pp.1181-1189.
- [Matsumoto 92a] Y. Matsumoto and K. Taki. Parallel logic Simulator based on Time Warp and its Evaluation. In *Proc. FGCS'92* Tokyo, July 1992, pp.1198-1206.
- [松本 92b] 松本幸則, 瀧和男: “バーチャルタイムによる並列論理シミュレーション”, 情処論, Vol.33 No.3, 1992年3月, pp.387-396.
- [Matsumoto 93] Y. Matsumoto and K. Taki. Adaptive Time-Ceiling for Efficient Parallel Discrete Event Simulation. In *Proc. of Western Multi-Conference on Computer Simulation* San Diego, U.S.A., January 1993.
- [松本 94a] 松本幸則, 瀧和男: “タイムワープ機構の新しい応用 -並列無格子配線-”, 情処論, Vol.35 No.4, 1994年4月, pp.666-676.
- [Matsumoto 94b] Y. Matsumoto. Applications of the Time Warp Mechanism to LSI-CAD. In *Proc. FGCS'94 Workshop on Parallel and Distributed LSI-CAD*, Tokyo, December 1994.
- [Nitta 92] K. Nitta, K. Taki and N. Ichiyoshi. Experimental Parallel Inference Software. In *Proc. of the Int. Conf. on Fifth Generation Computer Systems*, 1992.
- [Sechen 85] C.Sechen and A.Sangiovanni-Vincentelli. The TimberWolf Placement and Routing Package. In *IEEE Journal of Solid-State Circuits* Vol.SC-20 No.2, 1985, pp.510-522.
- [Taki 92a] K. Taki. Parallel Inference Machine PIM. In *Proc. FGCS'92* Tokyo, July 1992, pp.50-72.
- [瀧 92b] 瀧和男, 市吉伸行. マルチ PSI における並列処理とその評価 -小粒度高並列オブジェクトモデルに基づくパラダイムについて-. 信学論 D-I, Vol.J75-D-I No.8, pp723-739, 1992年8月.
- [瀧 93] 瀧和男編. 第五世代コンピュータの並列処理bit別冊, 共立出版, 1993年7月.
- [Uchida 92] S.Uchida. Summary of the Parallel Inference Machine and its Basic Software. In *Proc. FGCS'92* Tokyo, July 1992, pp.33-49.
- [Ueda 90] K. Ueda and T. Chika-yama. Design of the Kernal Language for the Parallel Inference Machine. *The Computer Journal*, Vol.33 No.6, 1990, pp.494-500.