

領域分割型並列配線システムの開発と評価

菊池 淳†, 白石 洋一††

†(株)日立製作所中央研究所, ††群馬大学工学部

概要

ビルディングブロック LSI のブロック間配線を対象とした並列配線システムを開発した。本システムでは、概略配線処理を行った後、配線領域を小領域に分割し、ネットワーク結合した複数の汎用 EWS を用いて、各小領域の詳細配線処理を並列に実行する。LSI テストデータを用いた評価実験の結果、従来逐次配線手法に対し、本システム (EWS 16 台構成) では、配線処理全体で約 3 倍、並列化を行った詳細配線処理で約 10 倍の高速化を確認した。

Abstract

We present a parallel routing system for inter-block routing problem of building-block model LSIs. This system divides routing area into sub-areas and distributes each sub-area to EWSs after global routing, and each EWS executes detail routing parallelly. Experimental results show that detail routing time on our system using 16 EWSs is 10 times shorter than the single execution.

1. はじめに

1.1 背景

LSI チップの設計に計算機を用いた DA/CAD ツールは今や必要不可欠となった。特に近年のチップの大規模化、複雑化の増大はその設計にとって深刻な問題であり、DA/CAD ツールの役割は今迄以上に重要になりつつある。実際、ゲート数増大は年率約 1.5 倍ペースで、今や 100 万ゲートチップも目前となりつつある。また、高密度レイアウト、動作速度最適化、多層配線モデル等の、同時に最適化する目的関数の高精度化、多種多様化はますます増大している。例えば、クロックスキュー、バスディレイ制御はピコ秒単位での制御が要求され、プロセスは 5~6 信号配線層が目前となっている。

LSI チップの自動レイアウト設計においても、問題規模、目的関数の複雑度が急増するに従い、より一層アルゴリズムの性能向上が不可欠となっている。一方で、性能向上のトレードオフとして処理時間の増大という問題が発生するようになった。

この問題を解決するために、我々はレイアウト問題を並列分散処理によって解くアプローチが有効であると考え、特に多大な処理時間を要する配線処理を例題としてその有効性を評価することにした。

レイアウト DA/CAD の分野では、並列分散処理アルゴリズムとしては既に幾つかのアプローチが提案されている [1]-[8]。それらのアプローチは、専用の並列計算機をベースに並列アルゴリズムを開発するものと [1]-[3][5][6][8]、既存の汎用計算機をベースとして既開発の逐次アルゴリズムを適用するもの [4][7] の 2 つのカテゴリに分類できる。本システムで採用するのは後者で、(1) 問題を分割して、(2) ネットワーク結合した複数台の EWS 上で、分割した問題を逐次アルゴリズムにより並列分散処理する、の 2 段階からなるアプローチである。それは、EWS の性能価格比が高いこと、汎用性が高く拡張性が大きいこと、過去に開発してきた逐次レイアウトプログラムの財産を活用できること、問題の小規模化により解の質と処理時間を大きく向上できること、等のメリットによる。

Development and Evaluation for Area Dividing Parallel Routing System

Jun KIKUCHI †, Yoichi SHIRAISHI ††

†Central Research Lab., HITACHI, Ltd., ††Faculty of Engineering, Gunma Univ.

このアプローチの課題として、(1)問題の分割、(2)小問題の境界条件設定、(3)並列分散処理制御を解決しなければならない。以下では、これらの課題を解決する各技術を示し、更にこれら技術を用いた本システムの実験的評価について述べる。

1. 2 従来手法

本節では、配線処理中で並列化を実現している従来の手法を示す。従来、ベクトル化迷路法 [8]、並列オブジェクト線分探索法 [6]、局所概略配線並列化法 [7]、分散処理型詳細配線手法 [4] 等の手法が提案されている。ベクトル化迷路法は、配線径路探索時の波面拡大処理にベクトル処理を利用して、波面拡大処理をパイプライン処理によって行なうことにより、全体の処理を高速化する手法である。並列オブジェクト線分探索法は、1線分を1オブジェクトと定義し、ある線分に交差する全線分を求める際にオブジェクト間メッセージ通信を利用して、交差線分を並列的に求めることにより処理を高速化する手法である。局所概略配線並列化法は、配線領域を分割する粗い格子で定義される小領域毎並列独立に概略配線径路を求めることにより処理を高速化する手法である。分散処理型詳細配線手法は、配線領域を格子状に分割した小領域内の詳細配線径路を並列に探索することにより処理を高速化する手法である。

並列化手法はそれを実現するプラットフォームと強く結びついている。ベクトル化迷路法はベクトル化計算機上に、並列オブジェクト線分探索法は並列推論マシン上に、局所概略配線並列化法、領域分割並列化法は、汎用 EWS 上にそれぞれインプリメントされている。

1. 3 従来手法の問題点

上記ベクトル化迷路法、並列オブジェクト線分探索法は、専用プロセッサを必要とし利用環境が限定されてしまう。局所概略配線並列化法は、配線処理時間の内、概略配線処理時間の占める割合が通常 10~20% であり、概略配線処理の並列化は配線処理全体に対する処理時間短縮の効果が低い。また、分散処理型詳細配線手法では、格子状に配線領域の分割を行なっている。しかし、本システムで対象とするブロック間配線問題では、配線領域内にマクロブロックなどの大きな配線禁止領域が存在するなど、配線問題量が均一に分布していない。そのため、ブロック間配線問題を扱おうと

すると、格子状分割では各配線問題量が不均一になり、十分な並列化の効果が期待できない。

2. 配線モデル

本章では、本システムの対象とするブロック間配線問題の配線モデル、及び自動配線処理の流れについて述べる。

2. 1 ブロック間配線問題の配線モデル

本システムは、特にビルディングブロックモデル LSI におけるブロック間配線問題を対象とする。図 2.1 はブロック間配線問題の配線モデルである。複数ブロックで囲まれた矩形領域 (チャンネルと呼ぶ) とブロック上が配線領域である。また、異なるブロックの間を結線する同電位端子の集合をブロック間ネットと呼ぶ。

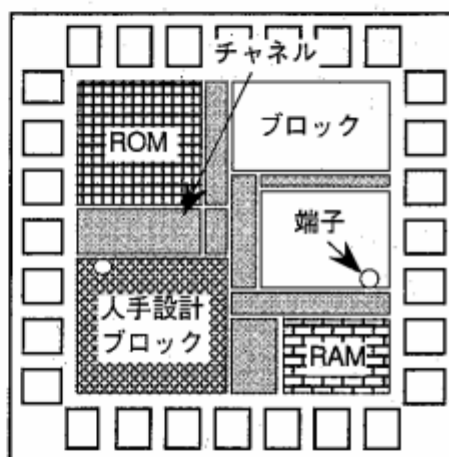


図 2.1 ブロック間配線問題配線モデル

2. 2 ブロック間自動配線処理の流れ

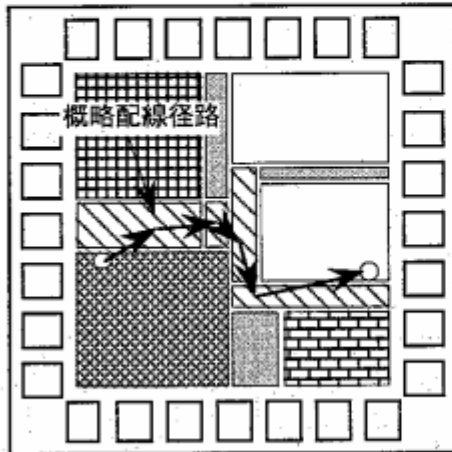
ブロック間自動配線処理は、概略配線処理と詳細配線処理の大きく 2 つの処理からなる。以下にそれぞれの概要を示す。

2. 2. 1 概略配線処理

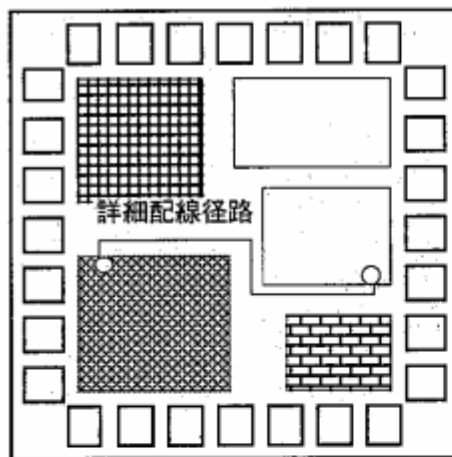
概略配線処理では、ブロック間ネットについてその配線径路をチャンネルを単位として決定する。この際の目的関数は配線長最小化ならびに配線混雑の平準化である。またチャンネルは各境界辺上に配線通過可能本数 (容量) を制約値として持ち、各ネットの配線径路は容量を超えてチャンネルを通過できない。本処理で得られる配線径路を概略径路と呼ぶ。概略径路の例を図 2.2 (a) に示す。

2. 2. 2 詳細配線処理

詳細配線処理では、概略径路に従って、チャンネル内での実際の配線径路を決定する。ここで配線径路が見つからなかったネットを未配線と呼ぶ。目的関数は未配線数最小化である。本処理で得られる配線径路を詳細径路と呼ぶ。詳細径路の例を図 2.2 (b) に示す。



(a) 概略配線径路例



(b) 詳細配線径路例

図 2.2 ブロック間配線処理の流れ

3. システム構成

本章では、本システムの並列処理プラットフォーム、及び配線処理の流れについて述べる。配線処理の個々の詳細は後章に記す。

3. 1 並列配線処理プラットフォーム

本システムのハードウェア構成を図 3.1 に示す。汎用 EWS (以下、ホストと呼ぶ) を複数、本システムでは現在最大 16 台、Ethernet を介して接続したネットワーク形態をとる。

接続されたホストは夫々ローカルディスクを持ち、各々に担当する配線領域についての入出力データを格納する。

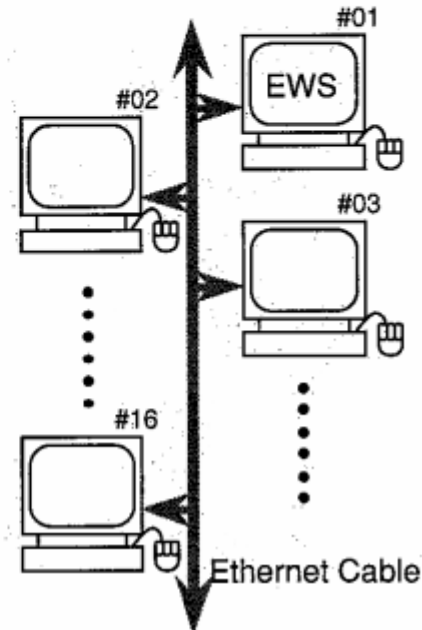


図 3.1 並列配線システムハードウェア構成

3. 2 配線処理の流れ

システム全体の処理の流れを図 3.2 に示す。

まず、概略配線フェーズにおいて、チャンネル生成を行なう。ここでは、チャンネル形状の決定、ならびに各境界边上の配線通過容量の設定を行なう。なお、ブロック間の領域だけでなく、ブロックもチャンネルとして定義する。続いて各ネットの概略配線径路を決定する。

次に、詳細配線フェーズに移る。詳細配線処理を並列化するため、その前処理として配線領域分割処理と境界端点決定処理を行なう。

配線領域分割処理では、配線領域を小領域に分割し、そこを通過する概略配線径路から 1 台の EWS が担当すべき部分配線問題を作成する。

境界端点決定処理では、各小領域の境界辺を通過する概略配線径路の詳細通過位置を決定する。

更に各 EWS に部分配線問題のデータを送付して、各 EWS 独立に詳細配線処理を実行する。全 EWS での詳細配線終了後、配線結果を統合して初期配

線問題に対する詳細配線結果を作成する。この時点で径路未決定ネットが存在している時は、配線領域全体を対象とした詳細配線処理を再度行なう。

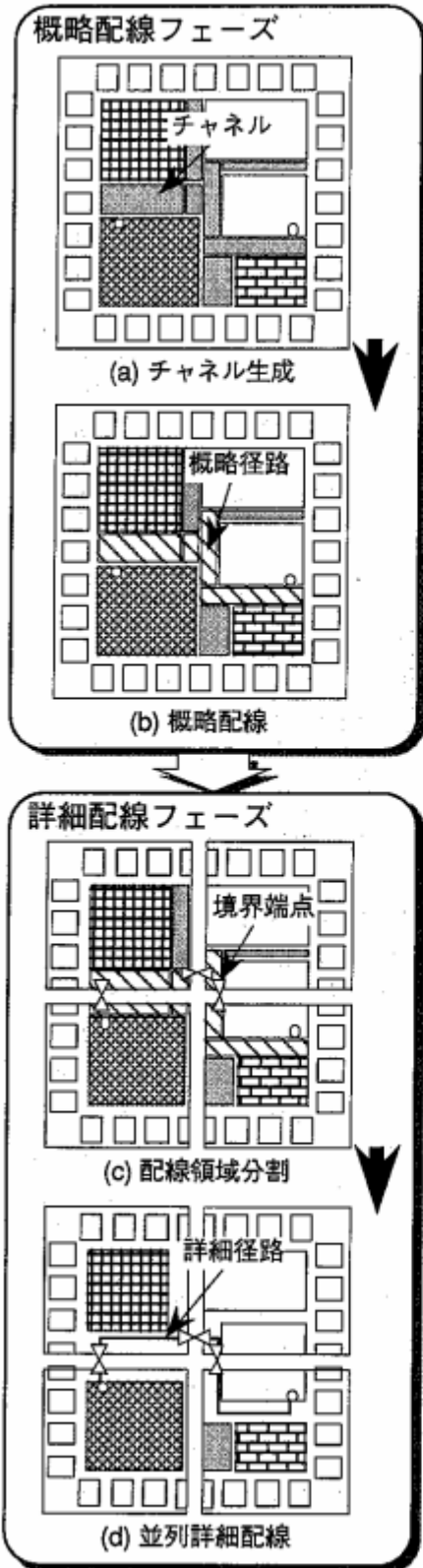


図 3.2 本システムの配線処理の流れ

4. 並列配線システムの特徴的技術

4.1 最適化技法を用いた概略配線技術

本システムにおける概略配線処理は、ネットワークフローアルゴリズムに基づいている。図 4.1 は本概略配線処理の流れを模式的に示したものである。まずチャンネルを節、チャンネルの隣接関係を枝とするグラフを生成し、配線問題をネットワークフロー問題に定式化する（同図 (a), (b)）。次にネットワークフロー問題を解き（同図 (c)）、得られたフローを概略配線径路に変換する（同図 (d)）。

ネットワークフローという一括意思決定アルゴリズムを用いることで、従来、概略配線処理で課題とされていた、ネットの処理順序の問題を解決し、全てのネットの相互影響を考慮した配線結果を得ることができる。

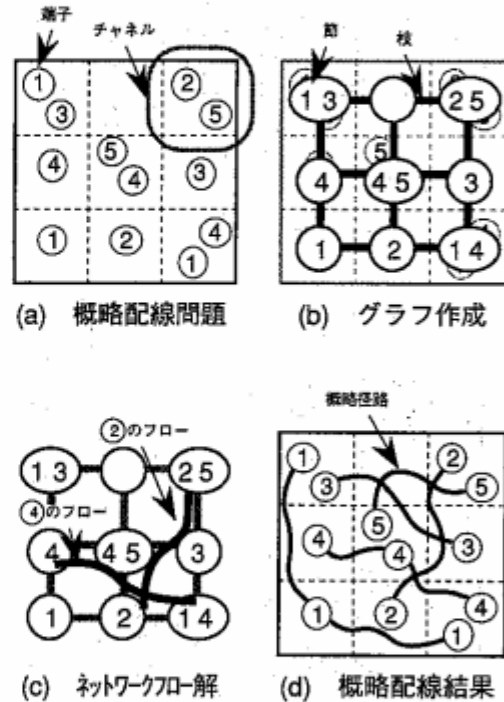
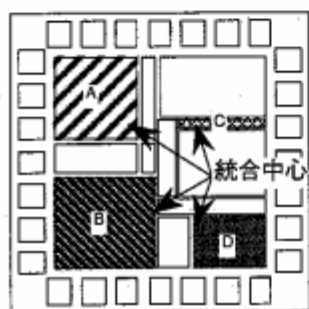


図 4.1 本システムにおける概略配線処理の流れ

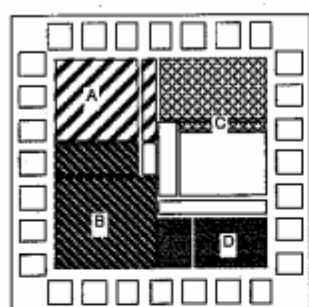
4.2 負荷均等化配線領域分割技術

並列化効率向上には、各プロセッサにかかる負荷を均等化することが必要である。配線領域を分割し、各分割領域をプロセッサに割り当てることで、配線処理の並列化を行なう本システムでは、各分割領域の配線問題の規模（以下、配線負荷と呼ぶ）を揃えるように配線領域分割を行なうことで並列化効率の向上を図る。

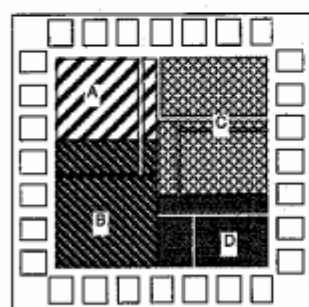
本分割処理では配線負荷を配線本数と仮定し、概略配線情報を有するチャンネルを単位としてチャンネルを統合することで分割領域を構成する。本処理の流れを図 4.2 に示す。まず、統合中心となるチャンネルを所望の分割数と同数設定する。更に、統合中心の周囲に隣接するチャンネルの中から統合時の配線負荷の増分が最も小さいものを統合する。この統合処理を未統合のチャンネルがなくなるまで繰り返す。このように配線負荷増分の小さいチャンネルを一つずつ統合することで、精度良く配線負荷を算出することができる。更に、チャンネル統合中に、全分割領域の配線負荷の最大値を目標値と定め、配線負荷の目標値を逐次更新しながら、配線負荷を目標値に合わせるようにチャンネル統合を行なうことで、分割領域間の配線負荷バランスをとることが可能となる。



(a) 統合中心設定



(b) 配線増分最小チャネル統合



(c) チャンネル統合結果

図 4.2 負荷均等化配線領域分割処理の流れ

4. 3 境界端点位置決定技術

分割領域の詳細配線処理を並列実行するためには、ネットの通過位置（境界端点）を予め決めておく必要がある。その境界端点位置の善し悪しは、未配線の発生に大いに影響を及ぼす。未配線を抑さえるように境界端点位置を決定するために、分割領域を通過する配線ネットと境界上の通過可能位置の間に配線処理に与える影響の大きさ（以下、通過困難度と呼ぶ）を割付コストとして定義し、全配線ネットの割付コストの総和の最小化を目的関数とする割り付け問題を解く。

配線ネットの境界上通過可能位置に対する通過困難度の要因として、次の3点をコスト化し、線形結合して割付コストとした。（図 4.3 参照）

- (1) 予測配線径路上の配線障害物の数と大きさ
- (2) 予測配線径路と配線障害物との距離
- (3) 分割境界線を挟んだ結線端子間の位置関係

4. 4 優先探索木による高速線分探索技術

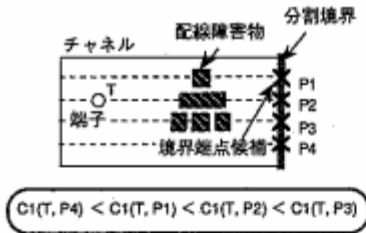
詳細配線手法には高速線分探索手法を用いている。線分探索手法は、探索開始点から探索目標点まで、障害物を回避しながら直線分の伸長と折曲げを繰り返して径路を求めるものである。本システムで用いた手法は、障害物データの生成法と構造を改善し、線分探索法で最も処理時間のかかる線分と障害物との衝突判定の高速化を図っている。

4. 5 並列配線制御技術

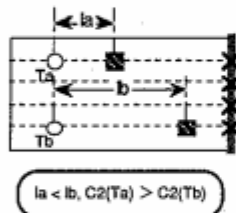
本並列配線システムでは並列処理の制御にクライアント・サーバモデルを採用している。

本並列配線処理では、クライアントプロセスのデータ要求に対して、サーバプロセスが配線データの分配を行ない、各クライアントの実行制御を行なう。各クライアントはデータを受け取り、サーバからの「実行許可」が得ると詳細配線処理を実行し始める。そして詳細配線処理が終了次第、サーバに「終了報告」を行なう。

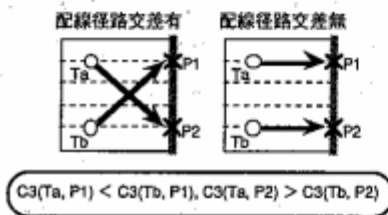
このようにプロセス間の通信は、「データ要求」、「実行許可」、及び「終了報告」に限定され、並列配線実行時の通信は一切ない。このように通信回数をできる限り低減することで、汎用 EWS における通信機能の貧弱さによる通信オーバーヘッド増大を回避する。



(a) 予測径路上の配線障害物の分布



(b) 予測径路と配線障害物との距離



(c) 結線端子の位置関係

図 4.3 通過困難度の定義

5. 評価結果

本章では、LSI テストデータを用いた実験結果を示し、本システムにおける並列化効果について考察する。実験に用いた LSI テストデータの諸元を表 5.1 に示す。なお、本実験で用いた EWS は Sun Microsystems 社の SPARCstation 10 である。

表 5.1 LSI テストデータ諸元

要求配線数	端子数	ブロック数	チャンネル数
2397	4277	16	65

5.1 全体評価

LSI テストデータに対する EWS 16 台構成の本システムならびに従来チップ一括手法における配線処理時間と未配線数を表 5.2 に示す。表に示すように本システムにより、全体の配線処理時間を従来手法の約 3 分の 1 に削減することができた。また、未配線数が 5 本発生したが、これは今後の概略配線アルゴリズム、及び詳細配線アルゴリズムの改良を進めることで十分対処できる量であると考えられる。

更に表 5.3 に本システム及び従来一括手法の配線処理時間の内訳を示す。表に示されるように、一括処理で配線処理時間全体の約 9 割を占めていた詳細配線処理を約 10 倍高速化することができた。この結果は、配線処理の中で詳細配線処理を並列化対象とした、本システムのアプローチの有効性を示している。しかし、総配線処理時間の内、約 3 割を未配線追加処理によって費やされている。今後、配線アルゴリズムを改良し、未配線を削減することで、更に配線処理時間を削減することが可能である。

表 5.2 本システムと従来手法の性能比較

	本システム	従来一括手法
処理時間 (秒)	4099	12160
未配線数 (本)	5	0

表 5.3 本システムにおける配線処理時間内訳

	チップ一括 概略配線	並列詳細 配線処理	チップ一括 未配線追加	その他 配線領域分割 境界端点候補 ブランチ	全体
従来一括手法 処理時間 (秒)	1146	11014	-	-	12160
(処理時間割合)	(9%)	(91%)	(-%)	(-%)	(100%)
本システム 処理時間 (秒)	1146	1096	1330	527	4099
(処理時間割合)	(28%)	(27%)	(32%)	(13%)	(100%)
高速化率	1.0	10.0	-	-	3.0

5.2 負荷分散効果

配線領域分割処理における負荷分散効果について考察する。配線領域を 16 分割する際に、本分割処理を用いた場合と、配線領域を各分割領域の面積が等しくなるように 4 × 4 分割を行った場合の、各分割領域の配線本数ならびに詳細配線時間を表

5.3 及び図5.1に示す。表5.4から本手法が配線本数、配線時間の分散低減に対して効果があることがわかる。また、図5.1では配線本数と配線時間との間に大きな相関があることが示されており、本分割手法の有効性が確認できる。

表5.4 16分割領域における配線本数と詳細配線処理時間の分布

分割領域番号	負荷分散考慮領域分割		分散考慮なし領域分割	
	配線数	処理時間(秒)	配線数	処理時間(秒)
1	258	43.1	24	1.3
2	291	164.5	31	2.9
3	301	73.5	32	2.0
4	304	209.0	154	12.3
5	306	92.1	158	14.5
6	309	463.3	201	43.6
7	319	83.5	203	178.9
8	326	36.6	218	52.0
9	330	646.3	220	25.1
10	334	373.8	287	471.0
11	336	89.9	298	88.2
12	347	71.5	359	85.9
13	349	77.8	421	239.0
14	365	40.1	496	391.5
15	384	105.0	516	1312.5
16	433	388.0	620	1047.1

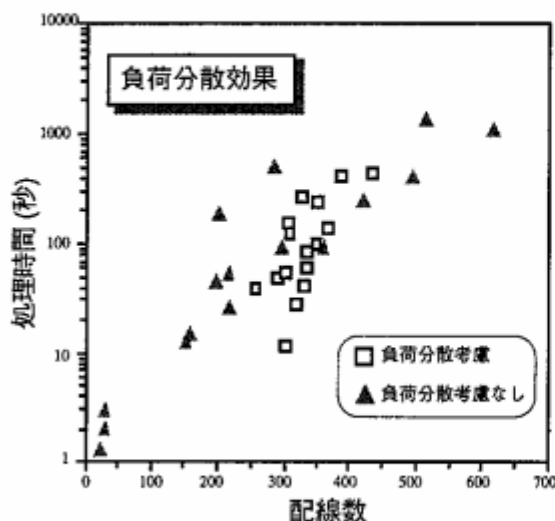


図5.1 配線本数と詳細配線処理時間の分布

5.3 並列化効果

上記 LSI テストデータに対して、配線領域を分割しない場合（逐次処理）、4分割した場合、8分割した場合、及び16分割した場合の詳細配線処理時間を表5.5、及び図5.2に示す。配線領域を分割することにより、詳細配線処理時間は確実に削減されており、並列化効果を確認することができる。しかし、8分割以上の分割になると、並列化効果が小さくなってしまふ。これは、より分割数が大きくなったときに、各分割領域の配線問題量の不均一が大きくなることの原因である。また、分割数が増えると同時に、決定すべき境界端点数も増え、未配線が多数発生してしまふ。そのために費やされる未配線追加時間も並列化効果の劣化の一因となっている。

表5.5 並列化効果

分割数	分割なし	4分割	8分割	16分割
初期配線時間(秒)	11014	2209	2005	1096
未配線追加時間(秒)	0	733	2006	1330
合計処理時間(秒)	11014	2942	4011	2426

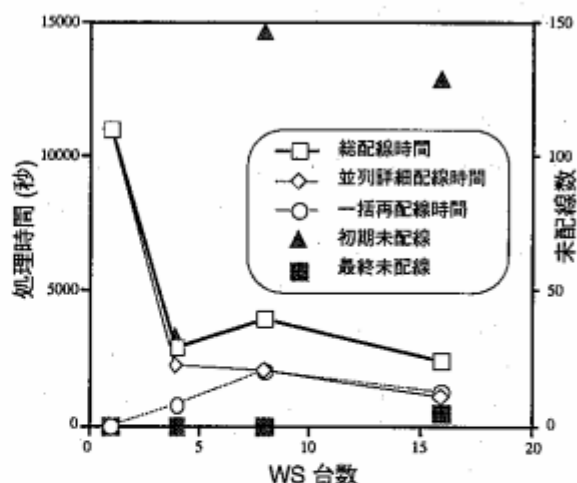


図5.2 詳細配線処理における並列化効果

6. おわりに

6.1 まとめ

ビルディングブロック LSI のブロック間配線を対象とした並列配線処理システムを開発した。本システムでは、概略配線処理を行った後、ネットワーク結合された複数台の EWS を用い、配線領域を小領域に分割し、各小領域毎並列に詳細配線処理を行うことを特徴とする。

LSI テストデータを用いた評価実験の結果、EWS 16 台を用いた本システムによって、従来逐次一括配線手法に対し、全配線時間を約 3 倍、並列化した詳細配線時間を約 10 倍高速化することが確認できた。

6.2 今後の課題

第 5 章で述べたように、本並列配線システムでは配線領域分割数が増加するに従い、未配線の発生、負荷分散の不均一化等の理由から、並列化効果が小さくなってしまふ。しかし、今後、LSI の規模増大、複雑化が進むと、今以上の設計時間短縮の要求が高まり、より多くの計算機による並列化が求められてくる。そこで、上記の問題を解決するために、

- (1) 概略／詳細配線技術の改良
- (2) 境界端点決定技術の改善
- (3) 配線領域分割技術の高精度化

が必要であると考えられる。

参考文献

- [1] Olukotun and Mudge : "A Preliminary Investigation into Parallel Routing on a Hypercube Computer," Proc. 24th DAC, pp.814-820 (1987).
- [2] Rose : "Locusroute: A Parallel Global Router for Standard Cells," Proc. 25th DAC, pp.189-195 (1988).
- [3] Kawamura, Shindo, Shibuya, Miwatari and Ohki : "Touch and Cross Router," Proc. IEEE ICCAD90, pp.56-59 (1990).
- [4] 羽根, 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, VLD91-84, pp.33-40 (1991).
- [5] Yamauchi, Nakata, Koike, Ishizuka and Nishiguchi : "PROTON: A Parallel Detailed Router on an MIMD Parallel Machine," Proc. IEEE ICCAD91, pp.340-343 (1991).
- [6] Date, Matsumoto, Kimura, Taki, Kato and Hoshi : "LSI-CAD Programs on Parallel Inference Machine," Proc. FGCS'92, pp.237-247 (1992).
- [7] 青木, 村方 : "ゲート敷き詰め方式ゲートアレイ用の並列概略配線手法," 第 4 回回路とシステム軽井沢ワークショップ予稿集, pp.373-378 (1992).
- [8] Miki : "A Fast Vectorized Maze Routing Algorithm on a Supercomputer," IEICE Trans. Fundamentals, Vol.E77-A, No.12 (1994).
- [9] 小島, 鈴木, 佐藤, 大附 : "ヒープ探索木を用いた改良線分探索法," 信学技報, VLD88-9, pp.65-72 (1988).
- [10] 菊池, 白石 : "領域分割型並列配線における負荷均等化手法," DA シンポジウム '93 論文集, pp.177-180 (1993).