

4V-9

並列図形処理システム

辻村 亮

富士通株式会社

町田 泰秀

1. はじめに

近年のLSIの集積度の増大に伴い、設計パターンのデザインルールチェック(DRC)、製造データ変換等の図形処理の処理時間が問題となっている。図形処理の並列化手法については、並列処理の問題点とされる負荷分散については、チップ全体を複数の可変サイズの領域に分割する可変領域設定法で、またプロセッサ間通信については、分割領域が各々の周辺のパターンを参照する境界パターン参照法で解決する手法を提案した。<sup>1)</sup>

本稿では複数のプロセッサによって大量の図形データを処理するシステムをEWS上でシミュレーションした結果について述べる。本システムは設計マスクパターンデータをホストプロセッサへ入力し、前処理として、ホスト上でチップ全体を複数の領域に分割してから、各プロセッサで小領域ごとのパターンデータを並列に処理し、結果をホストへ出力する。システムの特長は、従来の逐次的処理ではデータ量の増加に伴い、処理時間が膨大になる種々の図形処理に対し、アルゴリズムを変えずに並列に処理すること、従来の図形処理アルゴリズムの変更、追加が簡単にできること、並列プロセッサは分散メモリ方式で、各プロセッサでの処理は全てメモリ上で行われることである。

2. システム概要

図1にシステムの概要を示す。

- ①階層構造を展開されたパターンデータを入力する。
- ②図2に示すように、可変領域分割法により、パターンの分布状況から、各プロセッサが担当する小領域の最大値・最小値(左下頂点座標・右上頂点座標)を決定、テーブルを作成する。
- ③小領域の最大値・最小値テーブルをホストプロセッサから各プロセッサへ送信する。
- ④各プロセッサは自分の担当する領域の最大値・最小値を受信する。
- ⑤実際のパターンデータを転送する。
- ⑥各プロセッサはあらかじめ送られた小領域の最大値・最小値と各パターンの最大値・最小値を比較し、自分の担当領域内のパターンを読み込む。ここで図3の境界パターン参照法を使用する。
- ⑦各プロセッサ単位で独立に図形処理(DRC)を行う。
- ⑧結果をホストプロセッサへ送信する。
- ⑨領域数がプロセッサ台数を上回る場合は③に戻る。
- ⑩ホストプロセッサは処理結果を受信してチップ全体の処理結果に統合し、出力する。

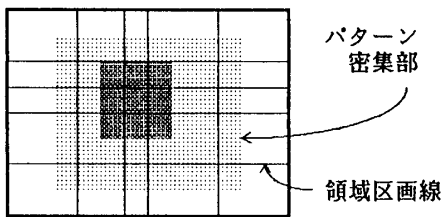


図2 可変領域分割法

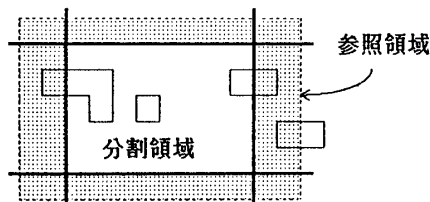


図3 境界パターン参照法

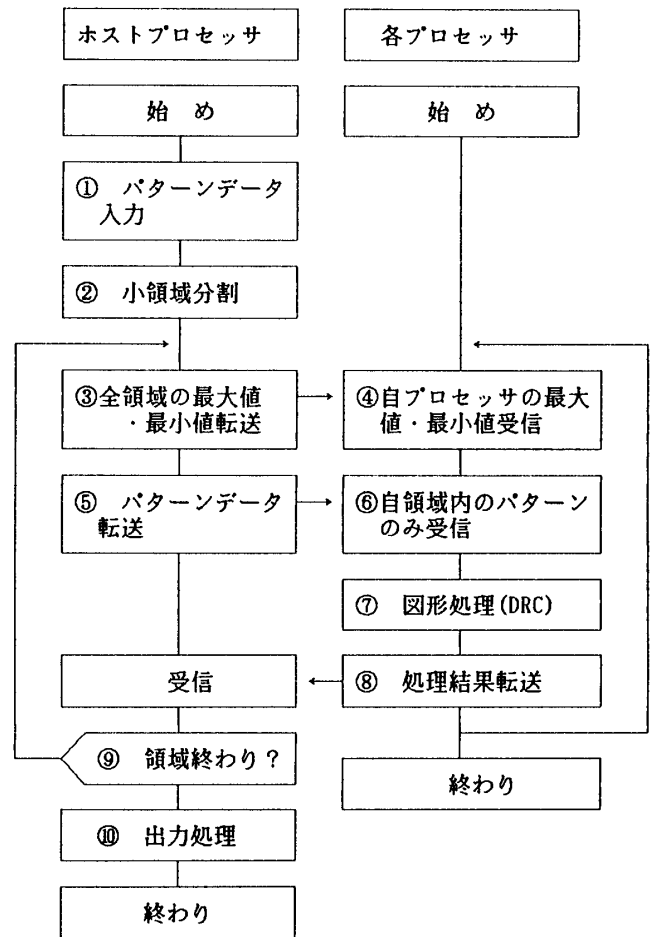


図1 システム概要

3. 実験結果

今回、DRC を図形処理の一例として選び、2MIPS のEWS 上でデータの分割処理と分割データ単位のDRC を行って、その結果を評価した。

(1) 負荷分散

下の式によれば、可変領域分割法を使用したことにより60%~80%の負荷分散率が得られた。使用しない場合の負荷分散率は20%程度であり、本領域設定法は領域分割による並列処理に有用、不可欠であると考えられる。

$$\text{負荷分散率} = \sum P Z_P / Z_{MAX} \times P$$

$Z_P$  : プロセッサPで処理されるデータ量

$Z_{MAX}$  : P台のプロセッサのうち処理するデータ量が最も多いプロセッサのデータ量

(2) 並列効果

実験結果によると、3千パターンのデータではプロセッサ台数4台で80%以上の並列効果が得られたがその後は下降し、17万パターンのデータでは、プロセッサ台数64台で80%の並列効果を得たが、その後144台では60%程度にとどまった。データ量がさらに大きくなった場合、200台以上のプロセッサ台数で高い並列効果を示すと予想される。

(3) 通信時間

本システムでは境界パターン参照法により、プロセッサ間通信をなくすようにしている。ホスト-プロセッサ間通信は50MB/secと仮定し、テストしたデータは17万パターンで約7MBであったので、往復の通信時間は約0.3秒であり、これはELAPS タイムが30秒程度であるので十分小さいと言える。

(4) 処理時間

図4に17万パターンデータにおいてDRCに要した時間(システム概要⑦)と、前処理(システム概要②~⑥)に要した時間の比較を示す。今回の実験では、DRCの機能は最小間隔と幅チェックであり、他の機能を追加した場合、DRCの時間が増加するので前処理は小さいものになる。また、前処理は最も時間のかかるパターンの受信(システム概要⑥)が、プロセッサ台数の増加に従って減少するため、全体としても減少する。前処理の高速化は今後の検討課題である。

図5に前処理時間とセル1台分のDRCの処理時間を加えた結果を示す。ただしDRCの処理時間は、スリット法による図形論理演算と、総当たり法による最小間隔、幅チェックに用いた時間のみである。プロセッサは処理するパターン数が最も多いものについてのみDRCを行った。この結果、P台のプロセッサでDRCを行った場合、従来、 $O(N^1.5)$ であった図形論理演算及びチェックは本手法によれば $O(N^{1.3}/P^{0.7})$ であり、また前処理+作成したDRCのプログラムのELAPSは2MIPSのEWS上で17万パターンについて、同性能のプロセッサを144台で並列処理した場合をシミュレーションした結果、約40MIPSの汎用計算機の約10倍の処理速度が得られ、並列効果は80%であったので、十分な高速化が得られると評価された。

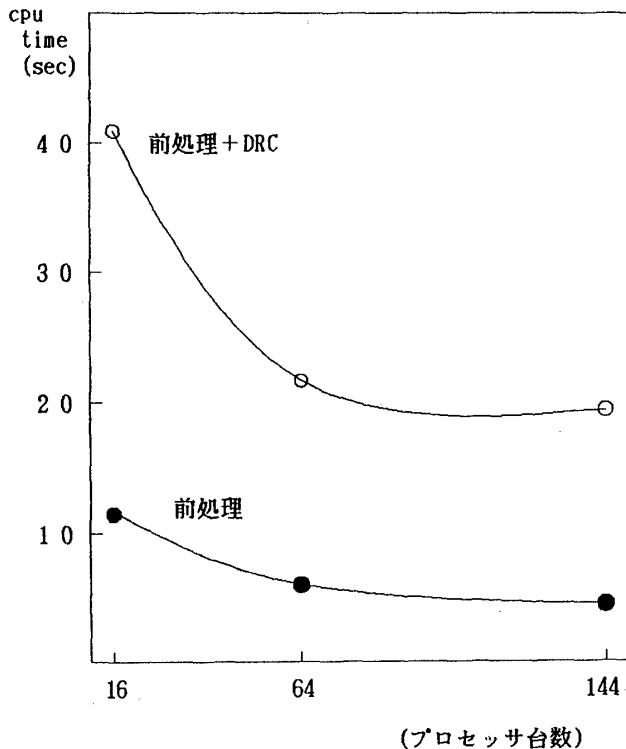


図4 前処理とDRCのCPUタイム

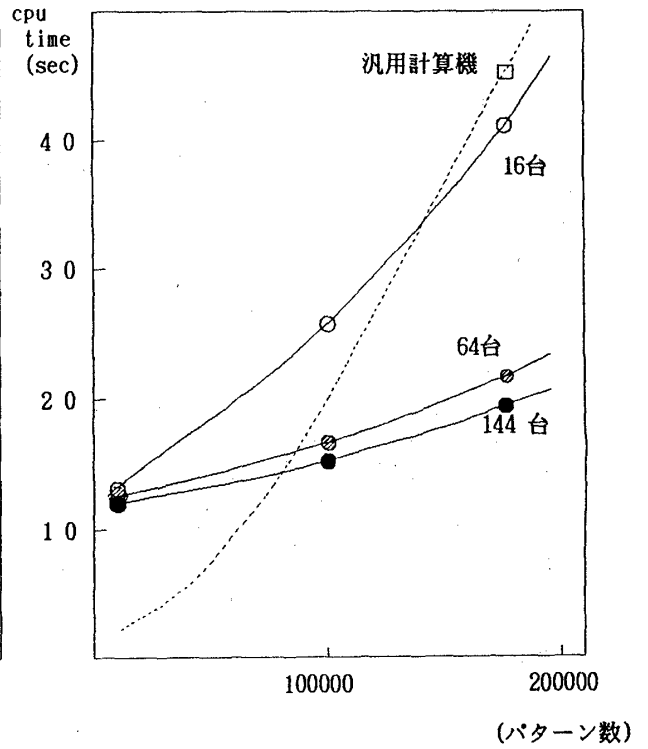


図5 パターン数と処理時間

4. まとめ

設計マスクパターンデータを並列に処理するシステムを提案し、ワークステーション上でシミュレーションした結果を評価した。その結果、大容量データほど、高い速度向上率が得られ、汎用計算機との処理速度の差が広がるので、今後のLSIデータ処理には並列処理化が不可欠であることを示した。マスクパターンは一般に階層構造を意識して設計される。本手法では階層を展開した状態のデータを入力としたが、階層展開も含めた並列化を検討中である。