超並列配線マシンのアーキテクチャ

4P−3

澁谷利行* 河村薫 進藤達也 三渡秀樹 大木由江
株式会社富士通研究所

1．はじめに

制約緩和迷路法は従来の配線アルゴリズムと比べて，非常に高い配線率を得ることができるが，実用規模のデータに適用するには時間がかかりすぎるという欠点があった．

我々は制約緩和迷路法の実用化をはかるために，アルゴリズムの並列性に注目して超並列配線マシンMAPLEを開発した．以下ではMAPLEの基本的なアーキテクチャとその考え方を示す．

2．設計方針

アーキテクチャ設計にあたっては，制約緩和迷路法の性質を十分に検討して，以下のような方針を立てた．

(1)格子構造の超並列計算機の実現：迷路法の高速化のためには，格子構造の並列計算機が適していることが知られている．また，実用規模の配線問題では，処理すべきデータが百万個以上存在する．したがって，"格子構造を持つ超並列計算機"を基本として制約緩和迷路法の高速化に必要な機能を検討する．

(2)高速なデータアクセス・隣接間通信の実現：制約緩和迷路法で最も処理時間の大きいラベリング処理の基本は，演算・通信の繰り返しである．したがって，全体のスループットを高めるためにはデータアクセス・隣接間通信を高速化することが最も重要である．

(3)全プロセッサデータに対する放送・収集機能の実現：制約緩和迷路法におけるラベリングでは，ラベルが一つ進む度にターゲットのコスト値を取り出し，全プロセッサの中でターゲットのコスト値よりも小さなコスト値を持つデータがあるかどうかを調べる終了判定が必要である．このためには，任意のプロセッサのデータを高速に収集し，全プロセッサに高速に放送する機能が必須になる．

(4)複数データ／プロセッサ処理の実現：実用規模の配線データを処理するためには，物理プロセッサ数よりも大量のデータを取り扱える必要がある．そのためには，各々の物理プロセッサが複数の仮想的なプロセッサをシミュレートすることにより，無限個のプロセッサが存在するように見せる機構が必要である(仮想プロセッサ空間)．しかも，この機構をソフトウェアで実現したのでは，制御が複雑になり，速度も遅くなるので，ハードウェアでサポートされていることが重要である．

3．実現方法

2．で示した考え方を基本にして超並列コンピュータMAPLEを設計した．MAPLEはSIMD型の超並列計算機で，最大64K個のプロセッサアレイとそれらを制御するコントローラから構成した．(図1)配線処理に必要なデータが様々なビット幅を持つこと，およびハードウェアとしての単純さから，プロセッサのALUは1ビットとした．

前章の設計方針は以下のように実現した．

(1)パイプライン処理

各プロセッサにおけるデータの読み込み，隣接通信，ALUによる演算，データの書き込みがパイプライン動作となるようにした．制御は，コントローラから送られるマイクロコードをデコードして行う．

パイプライン処理により，ALUは毎クロック毎に処理を行うので，データアクセスや隣接通信のオーバヘッドは見掛け上は0に見せることができる．

(2)放送収集演算回路

コントローラ内にグローバルメモリを設け，全プ

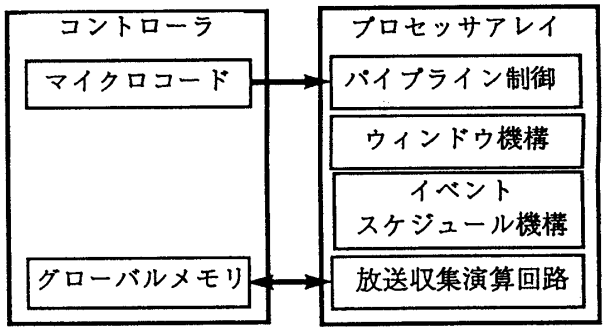

図1 全体構成

Architecture of the Massively Parallel Routing Machine
Toshiyuki Shibuya, Kaoru Kawamura, Tatsuya Shindo, Hideki Miwatari and Yoshie Ohki
FUJITSU LABORATORIES LTD., KAWASAKI

ロセッサのデータの収集演算結果を蓄える収集演算や，全プロセッサにグローバルメモリの内容を放送する放送処理を実現した．このための専用回路を放送収集演算回路と呼んでいる．収集演算回路では，論理和，論理積，和，最小値，最大値をとることができる．放送収集演算回路もパイプライン動作するように設計を行ない，グローバルデータの放送と収集演算のオーバヘッドは見掛け上0になる．

(3)仮想プロセッサ機構

ラベリング処理では，有効なラベリングを実行しているプロセッサ(ウェーブフロント)は，仮想プロセッサ空間全体の一部である．これを物理プロセッサがひとまとまりで，仮想空間を分割して担当する(図2 (a))よりも、物理的なプロセッサをいくつかのグループに分割して，小さい領域の単位でウェーブフロント上にマッピングさせた方が，有効なラベリングを実行しているプロセッサの割合は高くなる．(図2 (b))この機構をハードウェアで実現したのが仮想プロセッサ機構である．

MAPLEでは，物理的なプロセッサの256(16×16)個ごとのまとまりをグループ，一つのグループが処理する仮想プロセッサ上の領域をウィンドウと呼ぶ．図3 (a)に於て，グループAが担当する5番ウィンドウから，グループBが担当する4番ウィンドウへラベルが伝わった時には，グループAからグループBに対して意味のあるデータを伝えたことを示す信号(イベント)を発生させる．イベントを受けたグループBは，現在6番ウィンドウを実行中なので，グループAのウィンドウとは不連続である．そこで，次にラベリングするウィンドウとして4番ウィンドウを，グループBはスケジュールする．

ここで，次にグループBが4番のウィンドウの処理を行なおうとした時，既にAは別のウィンドウを処理しているかもしれない．したがって，各ウィンドウの境界では，お互いの処理中のウィンドウに依存せずに，自分と連続なウィンドウのデータを受け取れる機構が必要となる．MAPLEでは，この機構をウィンドウ機構と呼び，各グループの東西南北の境界プロセッサに対応して，通信専用プロセッサを設けることで実現した．図3 (b)において，PA，PBを，それぞれグループA，Bの東端，西端の境界プロセッサとする．CA，CBはPA，PBに対応する通信専用プロセッサである．CA，CBのメモリの内容は，PAとPBと同一になるようにする．PAとPBが不連続なウィンドウを処理している場合には，PAは本来PBから直接受けるべきデータをCBから受ける．

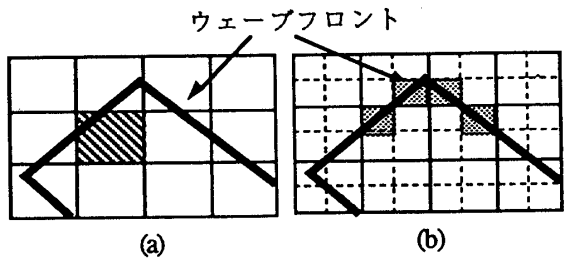

図2 ラベリング

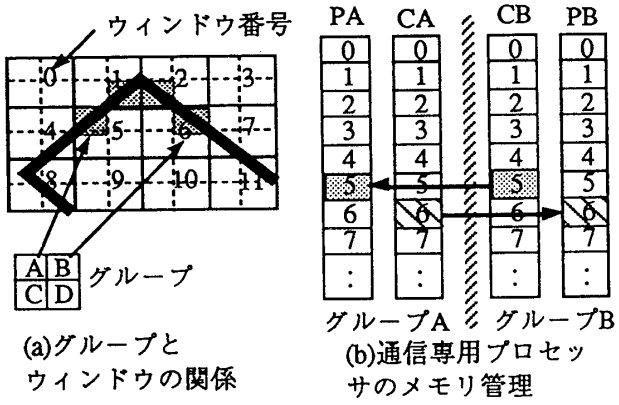

図3 仮想プロセッサ機構によるラベリング

PBも同様である．つまり，CA，CBのメモリアドレスを，それぞれ隣接するグループB，Aのプロセッサと同一にすることによって実現できる．

全グループにおいて，現在担当しているウィンドウ内で有効なラベリングが行なわれなくなったら，スケジュールの中から次に実行すべきウィンドウのデータに切り替えて，ラベリングを続けるという処理を繰り返す．このようにイベント信号を基本とした動作メカニズムをイベントスケジュール機構と呼んでいる．

これらの機構により，各グループにおいて，常にウェーブフロント上の領域が実行されて，ソースから始められたラベリングは，仮想空間の中を効率良く広がって行くことが実現できる．また，プログラマは物理プロセッサ数やグループの構成を意識しなくても良いので，ソフトウェアの開発も容易になる．

4. まとめ

本稿では，超並列配線マシンMAPLE の開発方針と実現方法について述べた．現在，実用規模の配線データを用いて本アーキテクチャの有効性を検証中である．