

並列形のデータフローマシンとノイマンマシンの科学技術計算への適用性に関する検討

Adaptability of data flow and von Neumann type computer
to the parallel processing of scientific program

藤原祥隆
Yoshitaka Fujiwara

田中 忠
Tadashi Tanaka

(日本電信電話公社 武蔵野電気通信研究所)

Musashino Electrical Communication Laboratory, NTT

1. まえがき

データフローマシンは、問題に内在する並列性を自然に引出し処理を高速化する、ソフトウェアの生産性向上を目的とする関数形言語を効率よく実行する、等を狙いに近年盛んに研究が行われている⁽¹⁾。

特にM. I. T. のDennis等により原子炉解析、気象予測等にみられる大規模計算を対象に並列処理による高速化を狙いとしたデータフローマシンの研究が進められている⁽²⁾。

一方、並列処理にも基づく高速化の手法は、データフローマシンに限らずILLIAC-IV以来ノイマンマシンの拡張形においても採用されている考えかたである。しかし並列処理に関するこれらの研究はノイマンマシンおよびデータフローマシンのそれぞれの分野で独立に行われており、両マシンの得失については定量的な議論がなされていないのが実情である。

本稿は以上のような背景の下で、並列処理において、データフローマシンとノイマンマシンとが、性能、ソフトウェア作成等に関しいかなる利点、欠点を有するかを明確にすることを狙いに以下の検討を行った。すなわち、内外の論文等において並列形ノイマンマシンと並列形データフローマシン間の得失を反映する技術項目として指摘されている、①メモリ使用特性（アクセス頻度、記憶容量）、②処理時間、③プログラム作成の難易度、等に注目し、大規模科学技術計算の具体問題である気象予測及び半導体デバイス解析の処理内容を例にとって両マシンの相互比較を行った。

以下、2章に比較方法、3章に得られた評価結果を示す。

2. 比較方法

2.1 並列形データフローマシンのモデル

データフローマシンには多くの方式が提案されているが、本稿では幾つかの研究機関で採用され、実現性が高いとみなさされている方式⁽³⁾⁽⁴⁾⁽⁵⁾を参考に基本構成を設定し使用する。

図2.1に構成モデルを示す。PE内処理、PE間通信処理

の概念を以下に示す。

2.1.1 PE内処理

1) 命令メモリには予め命令コード(OPコード)とリンクアドレスが格納されている。またオペランドメモリには処理の進行に応じてオペランドがオペランドの存在を示すタグ情報と組にして格納される。

命令発火機構にオペランドパケットが到着すると指定されたデスティネーションアドレスに対応する命令メモリとオペランドメモリの内容が同時に読みだされる。オペランドメモリに当該オペランドと組をなすもう一方のオペランドが存在している場合は命令パケットが作られ演算ユニットへ送られる。

なお起動時にはオペランドパケットとして起動パケットがデータフローマシンの外部より与えられる。

2) 演算ユニットは指定された演算を実行し、演算結果と命令パケット中のリンクアドレスを組にしてリザルトパケット

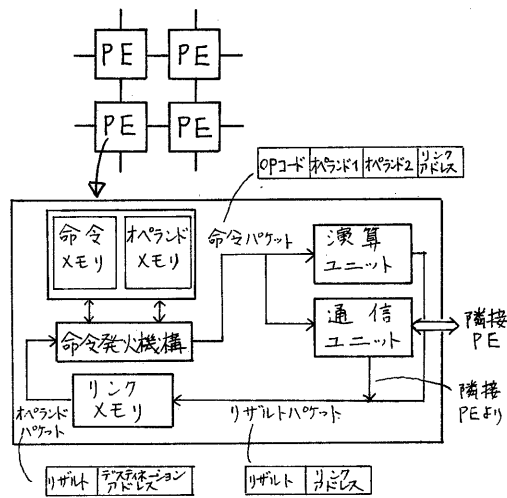


図2.1 並列形データフローマシンのモデル

を作成しリンクメモリへ転送する。

3) リンクメモリは演算結果を使用する複数の命令に対し複数のオペランドバケットを提供する機能を有しており、指定されたリンクアドレスから始まる一連のアドレスに予め格納されているデスティネーションアドレス群を読みだし、所要のオペランドバケット群を作る。

2. 1. 2 PE間通信処理

1) 演算結果を他PEへ送る場合、PE間通信を実行させる命令バケットが作られ、このバケットは演算ユニットではなく通信ユニットへ送られる。

2) 通信ユニットでは命令バケットのリンクアドレスで示される行き先情報に基づき、隣接する4つのPEの中の1つに対して命令バケットを送出する。

3) 通信ユニットはまた隣接する4つのPEから送られる命令バケットを受信する。受信したバケットが自PEで処理すべきものであれば自PEのリンクメモリへ送り、その他のものは所定の隣接PEへ送る。

以上の動作はすべて通信ユニット内で行う。PE間通信の動作概念を図2.2に示す。

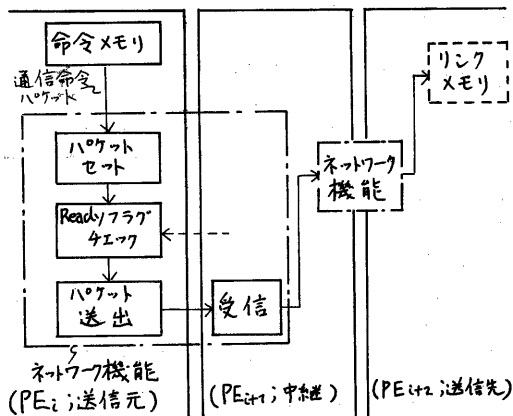


図2.2 データフローマシンのPE間通信の概念

2. 2 並列形ノイマンマシンのモデル

モデルを図2.3に示す。PE内処理、PE間通信処理の概念を以下に示す。

2. 2. 1 PE内処理

図2.1のデータフローマシンのモデルに対応して、命令メモリとオペランドメモリがオーバーラップ動作するメモリ構成とする (Harvard Architecture)。また演算ユニットは上記のメモリとオーバーラップして動作する。ただし演算ユニットは1時期に1つの演算を実行する one at a timeモードで動くものとする。

メモリ、演算ユニットの動作速度および演算ユニットの動作モードは図2.1で示したデータフローマシンと図2.3のノイマンマシンで同一とする。ただしノイマンマシンでは

中間結果、定数等を保持するレジスタは演算ユニット中に含まれるものとする。

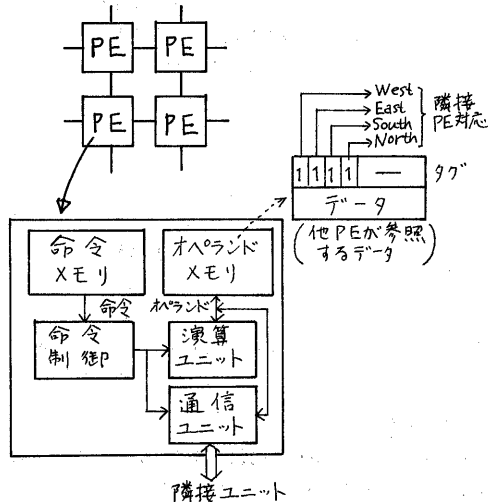


図2.3 並列形ノイマンマシンのモデル

2. 2. 2 PE間通信処理

PE間にまたがって参照されるデータについてはデータフローマシンと同様にそのデータを使用するPEに直接、送り付ける方法も可能である。しかしここでは通信時間が多少かかるがノイマンマシンの制御構造となじみやすいロックイン方式、すなわち必要なデータはその都度、そのデータが存在するPEから取り出す方式をとることとする。

PE間にまたがるデータは図2.3に示すようにタグ情報と組にしてオペランドメモリ中に格納する。このタグ情報により他PEはデータ利用の可否を知る。PE間通信処理の動作概念を図2.4に示す。

2. 3 評価項目

2. 3. 1 メモリ使用特性

2節で述べたように、データフローマシンはオペランドの複製を行う。また動作原理上、データの再利用を意識的に制御できないこともあって中間結果を保持するためにレジスタを用いるという試みはこれまで提案されているデータフローマシンには見当たらない。上記の理由等から、データフローマシンはノイマンマシンと比較してメモリアクセス頻度、記憶容量がともに増加することになる (表2.1)。オペランドメモリ、命令メモリに着目して、2.4項に示す計算問題の処理を例にとって、アクセス頻度と記憶容量に関し相互比較を行う。

2. 3. 2 処理時間

PE内処理時間についてはデータフローマシンはノイマンマシンと異なり図2.1の構成に示したように演算結果に基づき次の演算が開始されるまでに、リンク情報作成、命令バケット作成、等の動作が直列的になされる。ただし、並列性

が十分大であるときは、それらのオーバーラップ動作が可能であるが、前項で述べたオペランドメモリのアクセス頻度が多いことを考慮する必要がある。

PE間通信処理時間についてはノイマンマシンは他PEにあるデータの読みだしや他PEが利用するデータの格納に際し図2.3に示したようなタグ情報のチェック動作を伴う。しかしデータフローマシンは上記のような動作は命令発火機構により代替される。

本稿では2.4項に示す計算問題を対象に、両マシンの上記相違点に影響を与えるメモリアクセス時間と演算時間の相対関係、内部処理量と通信処理量の関係を変化させ、上述した個々の動作のオーバーラップを考慮した総合的な処理について両マシンの実行マシンサイクル数を比較する。

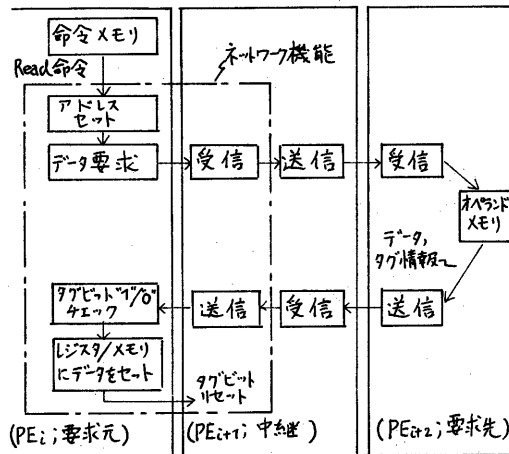


図2.4 ノイマンマシンのPE間通信の概念

表2.1 データフローマシンにおけるメモリアクセス頻度、記憶容量増加の理由

項目	理由	記事
メモリアクセス頻度	<ul style="list-style-type: none"> オペランドを複製する 中間結果を保持するレジスタがない 命令発火機構によりオペランド利用可否をチェクする 	<ul style="list-style-type: none"> ノイマンマシンでは オペランドを共有する レジスタを持つ オペランド利用可否のチェクはしない
記憶容量	<ul style="list-style-type: none"> オペランドを複製する 中間結果を保持するレジスタがない 	<ul style="list-style-type: none"> ノイマンマシンでは オペランドを共有する レジスタを持つ

2.3.3 プログラム作成の難易度

並列処理においては、上述のメモリアクセス、演算、PE間通信のオーバーラップ動作が性能を左右すると考えられる。データフローマシンは命令発火機構を備えているため、命令ストリームの切り換えを自動的に実行できる。このためユーザプログラムないしはコンパイラにおいて命令実行順序のスケジューリングが不要となる。ここでは上記特長がコンパイラ

ラ負担をどの程度軽減するかを考察する。なお、複数PEに対する処理のマッピングに関する効果についてはここでは検討対象から除いている。

2.4 具体的計算問題

大規模科学技術計算の具体例としては半導体デバイス解析問題、気象予測問題等とりあげる。これらの並列処理に関する代表的なアルゴリズムの一つに差分方程式の反復解法 (Odd・Even SOR法) があり、これを対象として検討を行う。

2.4.1 Odd・Even SOR法の基本パラメータ

2次元空間におけるポアソン方程式の解析問題を例にとって本方法における基本パラメータ、求解手順ならびに並列性について述べる。

1) 問題 (2次元矩形領域のポアソン方程式)

① 原方程式

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\varphi + S(x, y) = 0$$

② 差分表現した近似式 (Odd・Even SOR法向きの離散化)

$$\varphi_{i,j}^{(n+1)} = (1-\omega)\varphi_{i,j}^{(n)} + \frac{\omega}{4}(\varphi_{i+1,j}^{(n)} + \varphi_{i-1,j}^{(n)} + \varphi_{i,j+1}^{(n)} + \varphi_{i,j-1}^{(n)}) + \omega \cdot S_{i,j} \quad (1)$$

$\varphi_{i,j}^{(n)}$; 格子点 (i, j) における反復計算 n 回後の変数値

ω ; 加速因子

2) 求解手順

①問題空間を図2.5のように格子点に分割し、EvenグループとOddグループに分ける。

②各格子点に変数の初期値を与える。

③Even (Odd) グループの各格子点において(1)式の計算を行う。すなわち隣接4点と自分自身の値から次の状態の自分自身の値を求める (Even (Odd) ステージの計算)

④次にOdd (Even) グループの各格子点において、③と同一の計算を実行し次の状態の自分自身の値を求める
以下、各々の格子点における変数値が、一定の条件に収束するまで、③、④の計算ステップを交互に繰り返す。

ノイマンマシン、データフローマシンにおける上記③、④の計算の実行フローをそれぞれ図3.1、図3.2に示す。

3) 並列性

上記の求解法はEvenステージ或いはOddステージにおいて、各々の格子点で独立に同一の計算を行うため、各計算ステージ当たり、格子点数/2の高い並列性を得ることが

できる。また各格子点において行う計算においても演算レベルで部分的に並列性を得ることはできる。しかし上記に比べて並列性は小さく、さらに演算ステージのすべてにわたる均一な並列性は得られない (図3.2 参照)。

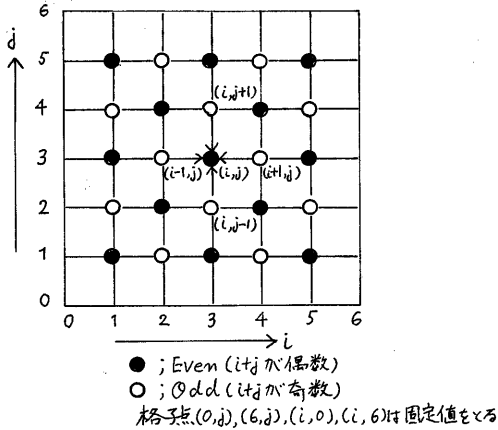


図2.5 Odd-Even SOR法の概念

2. 4. 2 半導体デバイス解析, 気象予測問題におけるパラメータ値

上述したOdd-Even SOR法により解析する半導体デバイス解析, 気象予測問題の具体的なパラメータ値の例を表2.2に示す。

3. 評価結果

3. 1 メモリ使用特性

命令メモリとオペランドメモリの両方に着目し2.4.2の具体例についてメモリアクセス頻度と記憶容量に関し評価した結果を図3.3に示す。なおメモリアクセス頻度は総メモリアクセス回数/総演算回数により求めている。

表2.2 半導体デバイス解析, 気象予測問題のパラメータ値例(1格子点当り)

	変数		定数		計算式の個数	
	自PE内の個数	他PEからの個数	共有度	共有度		
半導体デバイス解析	4	14	2.2	2.5	1.5	6
気象予測	7	15	2.6	14	2.1	7

(注) 共有度; 定数, 変数が考慮される平均回数

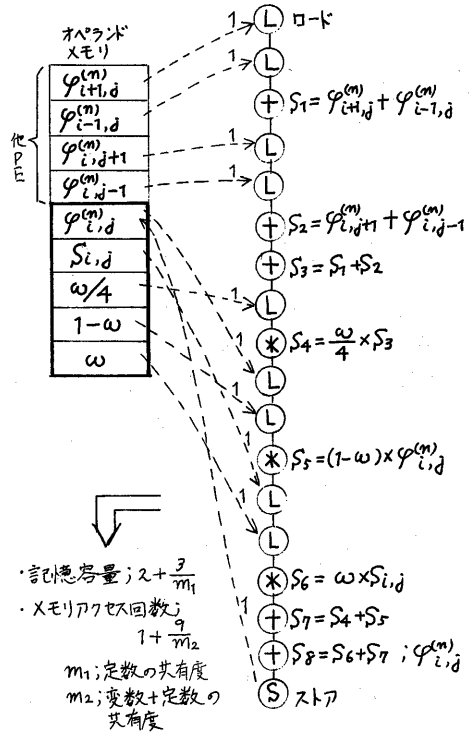


図3.1 パイプラインにおけるメモリ使用の概念

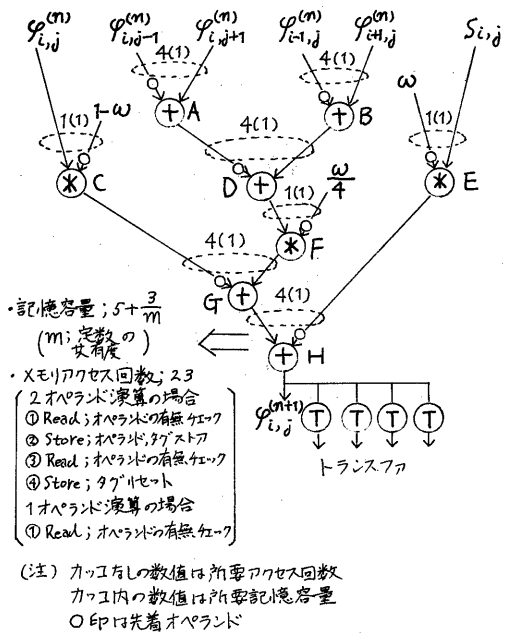
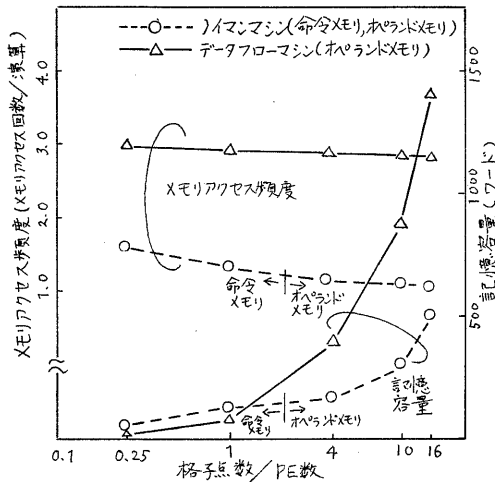


図3.2 データフローマシンにおけるメモリ使用の概念



(注) メモリアクセス頻度、記憶容量の値は半導体デバイス解析、負荷予測で求めた値の単純平均値を示す
図3.3 メモリアクセス頻度、記憶容量の比較例

評価結果を要約すると次のようになる。

- 1) アクセス頻度については、データフローマシンのオペランドメモリはノイマンマシンのオペランドメモリおよび命令メモリに比べて2, 3倍大である。
- 2) 記憶容量については、格子点数/PE数 ≤ 1 の条件では命令メモリが支配項となり、データフローマシンとノイマンマシンは同程度である。また、格子点数/PE数 ≥ 1 の条件ではオペランドメモリが支配項となり、データフローマシンはノイマンマシンと比べて3倍程度多くなる。

3.2 処理時間

Odd・Even SOR法の反復計算1ステージの処理に要する総合的な処理時間について比較を行う。処理時間の算出に当たっては、命令フェッチ、演算、PE間通信等に要するマシンサイクル数をオーバーラップ動作を考慮して加算する。マシンサイクル数算出ならびにオーバーラップ動作に関する主要な条件を表3.1に示す。

2. 4項に示す計算問題を用いた評価結果を図3.4から図3.7に示す。得られた主な結果は次のようになる。

1) 総合的な処理時間

①メモリアクセス時間=ALU動作時間の場合(図3.6) データフローマシンの処理時間はノイマンマシンの処理時間の1.5倍程度である。これはデータフローマシンにおいてはオペランドメモリ・アクセス回数が多いことによりPE内処理時間が長いためである。

②メモリアクセス時間 \leq ALU動作時間/2(図3.4, 図3.5, 図3.7); データフローマシンの処理時間とノイマンマシンの処理時間は同程度である。これはノイマン

マシンがデータフローマシンに比べて通信処理時間が長い、一方、データフローマシンがノイマンマシンに比べてオペランドアクセス回数が多いことおよび逐次性処理のためPE内処理時間が長いためである。

なお、格子点数/PE数 > 1 の場合、メモリアクセス時間=ALU動作時間の条件ではPE間通信時間の影響がほとんどないためさらにノイマンマシンが有利となりデータフローマシンの処理時間はノイマンマシンの処理時間の2倍程度となる。またメモリアクセス時間 \leq ALU動作時間/4の条件では両マシンともに演算時間が支配要因となるため両マシンの性能は同程度となる。

2) 並列展開による処理時間短縮効果

PE個数が格子点数と同等以上に多くなるとデータフローマシンとノイマンマシンの両者において並列展開効果が鈍化し始める。(上記の評価結果によるとPE数が1から4に増しても総合性能は2.5倍程度しか向上しない。)

この理由はPE数増加によるPE当たりの処理量の削減はどPE間通信量が減らないことおよび隣接しないPE間の通信が増すことによるデータ転送距離の増大のためである。

なお、上記の検討を補足する目的で、PE内処理の並列度の変化に対するPE内処理時間の変化、PE間通信量の変化に対するPE間通信処理時間の変化、およびノイマンマシンにとって厳しい条件における総合的な処理時間について付録に示す。

3.3 プログラム作成の難易度

データフローマシンに対するソフトウェアの命令実行順序スケジューリングが不要であるという特長がコンパイラ負担をどの程度軽減するかを検討した。ノイマンマシンに対するスケジューリングの一方法として、命令のオーバーラップ動作の可能性を利用し、命令実行順序を組み替える方法を想定する。その内容は付録に示す。

1) 検討の結果、2.4項に述べた問題においては、スケジューリング問題は通信処理量が多い場合、例えば1PEに1格子点を割り当てて最大性能を引き出すようなケースに生じる。この場合、演算と通信のオーバーラップ処理がスケジューリング問題として生じるが、静的なスケジューリングで対処可能であることからスケジューリングの負担は小さい。しかもスケジューリングの工夫による性能向上効果は高々1.5倍程度である。

静的でしかも全PEが一樣であることから、このような側面は、Poisson方程式の求解問題に限らず、偏微分方程式で記述される連続体のシミュレーション、同一形態の関係式の集合で表現される離散系の問題等、数多くの科学技術計算問題に共通的にみられる。

2) スケジューリングの難易の観点から、データフローマシンの特質を活かした並列処理を科学技術計算の枠にとらわれず考えてみると、データフローマシンの特質を自動発火のハードウェア機構による自動スケジューリング機能に見いだす

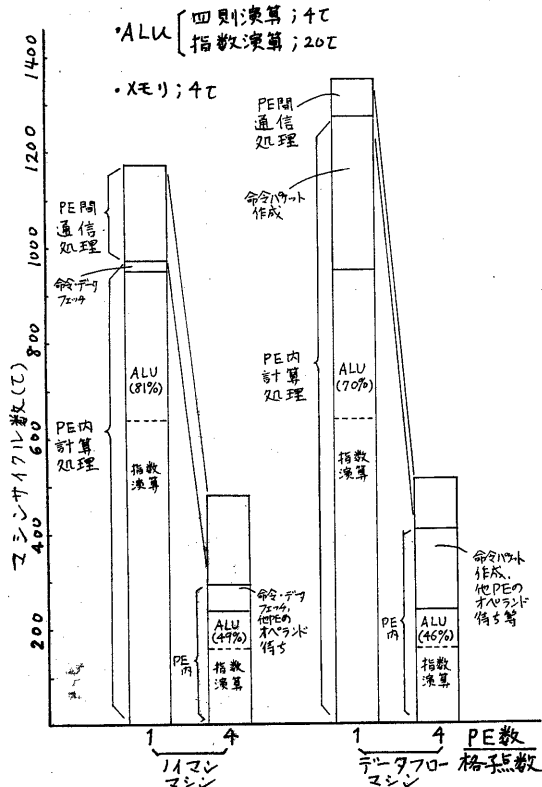


図3.4 半導体デバイス角解析問題における総合性能算出例(1)

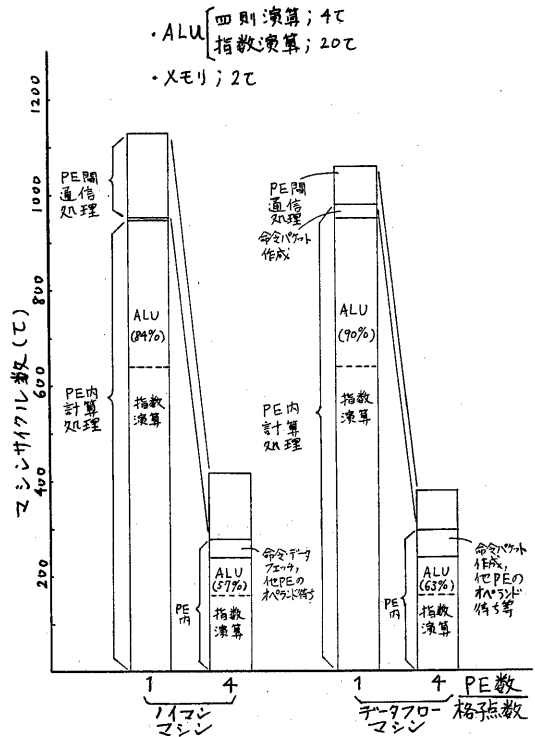


図3.5 半導体デバイス角解析問題における総合性能算出例(2)

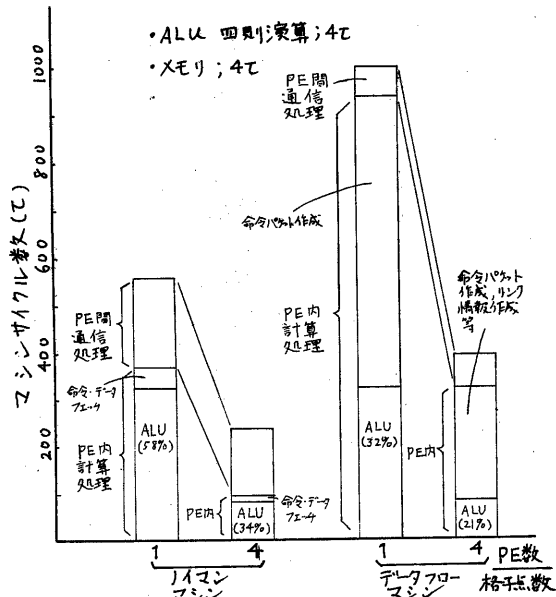


図3.6 気象予測問題における総合性能算出例(1)

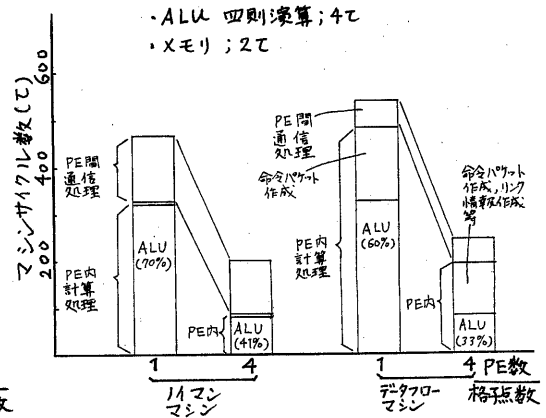


図3.7 気象予測問題における総合性能算出例(2)

表3.1 演算, PE間通信, 命令フェッチ等のマシンサイクル数
算出とオーバーラップ動作の主要条件

分類	ノイマン / データフロー	内 容	言 事
演算	ノイマン, データフロー	<ul style="list-style-type: none"> 四則演算; 4T 指数演算; 20T 	
PE間通信	ノイマン	<ul style="list-style-type: none"> オペランド入力 データ転送制御; 14T (12T) 中継; $(l-1) \times 6T \times 1$ オペランドセット; 20T (10T) 	l : PE間距離 (隣接; $l=1$) $X1$: データの要求, データ受信の回復を含む $X2$: ネットワーク制御キューを含む
	データフロー	<ul style="list-style-type: none"> オペランド出力 データ転送制御; 11T (9T) 中継; $(l-1) \times 5T \times 2$ 	
命令フェッチ等	ノイマン	<ul style="list-style-type: none"> 命令, データフェッチ; 4T (2T) 	$Q1$: 命令メモリ・オペランドメモリ・キュー $Q2$: ALUキュー $Q3$: リンクメモリキュー $Q4$: ネットワークキュー
	データフロー	<ul style="list-style-type: none"> 命令パケット作成 1オペランド命令; 4T (2T) 2オペランド命令; 16T (8T) リンク情報作成; 5T (3T) キュー $Q1$; 2T, $Q2$; 2T $Q3$; 2T, $Q4$; 2T 	
オーバーラップ動作	ノイマン	演算, 命令フェッチ, データフェッチ, PE間通信処理をオーバーラップ	
	データフロー	演算, PE間通信, 命令パケット作成, リンク情報作成をオーバーラップ	

(注) メモリアクセス所要サイクル数が4Tの場合と2Tの場合について示す。() 内がメモリアクセスが2Tの場合の値

ことができる。この場合、走行状態を判断しながら処理を進めることが可能となり、計算手順が走行結果によって動的に大きく変化する処理においてデータフローマシンの効果が発揮されたと考える。

4. むすび

並列処理において、データフローマシンとノイマンマシンが性能、ソフトウェア作成等に関していかなる利点、欠点を有するかを狙いに以下の検討を行った。即ち、論文等において並列形ノイマンマシンと並列形データフローマシン間の得失を反映する技術項目として指摘されている、①メモリ使用特性(アクセス頻度、記憶容量)、②処理時間、③プログラム作成の難易度、等に着目し、大規模科学技術計算の具体問題である半導体デバイス解析及び気象予測の処理内容を例にとって両マシンの相互比較を行った。

その結果、データフローマシンはノイマンマシンと比べて①メモリアクセス頻度、記憶容量はそれぞれ2、3倍多い、②メモリアクセス時間=ALU動作時間の条件のとき、PE内処理時間が2.5倍程度かかると、PE間通信処理時間が1/3倍程度に少なく、総合的な処理時間が2倍程度かかる。またメモリアクセス時間 \leq ALU動作時間/4の条件のとき、PE内処理時間が同程度、PE間通信処理時間が1/2倍程度に少なく、総合的な処理時間が同程度である。等を把握

した。

また命令実行順序のスケジューリングが自動的に行われるというデータフローマシンの特長は、計算手順が走行結果によって動的に大きく変化するような非定形的な処理において効果が発揮される可能性があることを把握した。

謝辞

本検討にあたりデータフローマシンに関する各種の情報を提供していただいた当研究所の情報基礎研究部第二研究室 両宮室長、長谷川調査員に御礼申し上げます。

文 献

- 1) 宇都宮公訓, "データフロー計算機", bit, vol 12, no. 3~no. 9
- 2) J. B. Dennis, et al, "Workshop on Data Flow Computer and Program Organization", ACM Computer Architecture News, vol. 6, no. 4, 1977
- 3) 高橋, 両宮, "超高速科学技術計算向きデータフロープロセッサレイ計算機のアーキテクチャ", 信学論 (D) vol. J67-D, NO. 1, 1984
- 4) I. Watson and J. Gurd, "A prototype data flow computer with token labelling, NCC, 1979
- 5) Arvind and V. Kathall, "A Multiple Processor Data Flow Machine That Support Generalized Procedures", Proc. of 8th Symposium on Computer Architecture, 1981

付 録

1. PE内処理時間, PE間通信処理時間

1) PE内処理時間

PE内処理において総ての命令を逐次に行う必要がある場合(並列性なし)と、相互に独立な命令群が十分に存在している場合(並列性大)について、1命令当たりの実行時間を試算した結果を付表3.1に示す。これらの結果をまとめると次のようになる。

i) 並列性大の場合

① メモリアクセス時間=ALU動作時間の場合; データフローマシンの処理時間はノイマンマシンの処理時間の約

2.5倍である。これは、データフローマシンはノイマンマ

シンに比較してALUの使用率が低いことによる。

② メモリアクセス時間 \leq ALU時間/4の場合；データフローマシンの処理時間とノイマンマシンの処理時間は同じである。これは両マシンにおいてALUが性能の支配要因となるためである。

ii) 並列性なしの場合

データフローマシンの処理時間はノイマンマシンの処理時間の約2.5倍である。これはデータフローマシンの命令ターンアラウンドタイムがノイマンマシンにくらべて長いことによる。

2) PE間通信処理時間

PE間通信処理がPE全体の処理の支配要因となる場合を想定してPE間通信量の変化に対するPE間通信処理時間の変化を試算した。1つの変数値の計算に要する他PEのデータ利用回数と所要通信時間の関係を付表3.2に示す。これよりノイマンマシンはデータフローマシンの2.3倍の処理時間がかかることがわかった。なおノイマンマシンではデータフローマシンと比べて他PEデータの利用回数に依存しない固定分が大きいいため他PEデータの利用回数増加に対するPE間通信処理時間増加の割合がデータフローマシンに比べて緩慢である。またノイマンマシンではデータフローマシンと比べてメモリアクセス時間の影響が大きい。

付表.1 1命令当りの実行時間

処理の内容 マシン	並列性なし	並列性あり	記 事
ノイマン マシン	10 τ (5.5 τ)	4 τ (4 τ)	R-R命令とR-X命令を均等に実行する条件で算出
データフロー マシン	24 τ (13.5 τ)	10 τ (4 τ)	1オペランド演算と2オペランド演算を均等に実行する条件で算出

(注) ALUの動作時間を4 τ とする。メモリアクセス時間を4 τ 、或は τ とする。()内は τ のときの値を示す。

付表.2 1つの変数値の計算に要するPE間通信処理時間試算例

他PEのデータの 利用回数 マシン	1	4	10
ノイマン マシン	34 τ (16 τ)	64 τ (46 τ)	124 τ (106 τ)
データフロー マシン	11 τ (9 τ)	29 τ (26 τ)	65 τ (62 τ)

(注) ALU動作時間を4 τ とする。メモリアクセス時間を4 τ 、 τ とする。()内は τ のときの値を示す。

2. 特異な処理内容による評価

ノイマンマシンにとって厳しい条件、すなわち1演算毎に他PEとのあいだでデータをやりとりする付図.1の処理モデルについてALU動作時間を固定し(4 τ)、メモリアク

セス時間を変化させて処理に要する総合的な処理時間を相互比較した。

この結果、メモリアクセス時間 \geq ALU動作時間/2の条件では並列性が十分ある条件においてもデータフローマシンはノイマンマシンとほとんど同じ性能であった。これはデータフローマシンにおいてオペランドメモリアクセス回数が多いことが性能を制限していることによる。

メモリアクセス時間をさらに短縮していくとノイマンマシンはPE間通信処理が制限要因となり性能向上が得られない反面、データフローマシンはメモリアクセス時間がALU動作時間/4程度までは性能が向上する。究極的にはデータフローマシンもPE間通信処理により性能が制限され、この場合データフローマシンの性能はノイマンマシンの約2.5倍となる。この差はPE間通信処理時間の差による。

3. ノイマンマシンに対するスケジューリングの一方

1) 前提条件

- ・スケジューリングの対象は命令実行順序と命令フェッチタイミングとする。
- ・命令間のフェッチ間隔の最小値を予め命令シーケンスの可能な組合せについて与えておく(命令フェッチ間隔情報)。

2) スケジューリングの手順

スケジューリングの概念を付図.1に示す。

i) データフローグラフ表現した処理内容について、以下のよう命令を実行順序の優先順位に関しグループ分けする。

- ・定数、外部入力を使用する命令を最も優先順位の高いグループとする(#1グループ)
- ・定数、外部入力、#1グループの出力を使用する命令を次に優先順位の高いグループとする(#2グループ)

以下同様にして最下位優先順位グループ(#nグループ)まで作る。

ii) #1グループから順番に#nグループまで以下の手順により実行順序をスケジュールする。なお同一グループ内においては以下の手順を各演算ノードについて順次行う。

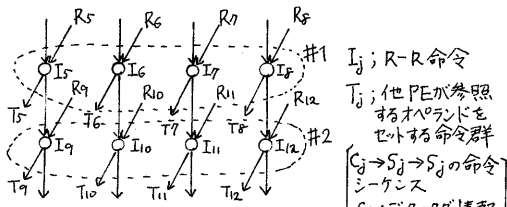
- ①着目する演算ノードに關係する処理を機械語命令のシーケンスに展開する。
- ②各命令に次の命令との命令フェッチ間隔情報を付加する
- ③先頭の命令を命令フェッチ間隔に関する規則を満足する範囲で先行する命令群の中に移動する。移動後、各命令の命令フェッチ間隔情報を修正する。

3) スケジューリングによる性能向上効果

付図.1の処理に上記のスケジューリングの手順を適用した場合の性能向上効果は1.5倍から2倍程度である。

(算出条件)

- メモリアクセス時間；2 τ 、演算時間；4 τ
- PE間通信処理時間；10 τ 、並列度；4以上



スケジュール順序

()内の数値は命令フェッチ間隔情報

・) -ド I_5 :

$R_5(4.5) I_5(0) C_5(1) S_5(0) S_5()$ (手順①, ②を適用)

・) -ド I_6 :

$R_5(4.5) I_5(0) C_5(1) S_5(0) S_5(0) R_6(4.5) I_6(0) C_6(1) S_6(0) S_6()$
 (手順①, ②を適用)

$R_5(3.5) R_6(0) I_5(0) C_5(1) S_5(0) S_5(3.5) I_6(0) C_6(1) S_6(0) S_6()$
 (手順③を適用; R_6 に適用)

付図.1 命令実行順序スケジューリングの概念