

データフロー計算機による数値計算と演算過程の画像化

著者	幾志 新吉
雑誌名	国立防災科学技術センター研究速報
号	91
ページ	27-34
発行年	1990-03-30
URL	http://id.nii.ac.jp/1625/00002831/

データフロー計算機による数値計算と 演算過程の画像化

幾志新吉

国立防災科学技術センター

Application of Dataflow Computer to Numerical Simulation

By

Shinkichi Kishi

National Research Center for Disaster Prevention, Japan

Abstract

Numerical calculation for the simulation of natural phenomena were conducted using the dataflow type parallel processing computer. In the beginning chapter system configuration and programming of the computer are described and in the following chapter an example of dataflow computation for the typical simulation model of heat conduction is described.

In the case computing speed of the dataflow computer estimated from operation steps and actually measured computing time was equivalent to about 30 MFLOPS.

On the other hand real-time visualization of computing process was realized by using the image display directly connected to the memory of the dataflow computer.

Key words: Dataflow computer, Ring-bus structure, Dataflow graph, Parallel processing, Dynamic image display.

キーワード: データフロー計算機, リングバス構造, データフローグラフ, 並列処理, 動的画像表示

1. はじめに

大規模な科学技術計算における高速化の要求は増大の一途をたどり、高性能の計算機の普及が新しい応用分野を開拓する形でますますこの傾向を加速している。計算機の高速度の手段は素子の性能向上と並列処理であり、前者に限界がある以上、並列処理に期待がかかる。その背景には、素子の高速高集積技術の発達で同種の回路を多数並べることが容易となってきたことがある。

しかしながら、ノイマン型という方式に分類されるこれら従来の計算機では、命令の実行

順序をすべてプログラムで指示しなければならないため、処理が複雑になるにつれてソフトウェアの手間は幾何級数的に増大する。また、現在のスーパーコンピュータやノイマン型の並列計算機では、高性能を引き出すために並列実行制御を明示するなど特殊なプログラミング技法を必要とする。

一方、ノイマン型の計算機が、つぎに実行する命令をプログラムにより直接指示する（プログラム駆動）のに対し、一対のデータの到着によって実行可能となった命令から実行して行く（データ駆動）のがデータフロー計算機である。これは、可能なものはすべて並列に実行するという意味から並列処理の基本モデルであって、問題に内在する並列性をハードウェアが極力自動検出できるという能力をもっている。

このように有利な性質をもつデータフロー方式であるが、従来の計算機の方式と全く異なっているため、その普及は遅れている。しかし問題点の多くは解決され、商用機も日本、米国で登場し、1970年代後半から盛んになったデータフロー研究がそろそろ収穫期に入る。今後はより実用性を指向した研究が期待されている。

2. システム構成

NEDIPSのシステム全体の構成を図1に示す。データフロープロセッサは制御ユニットを介して、プログラムの起動や結果の受取りなどを受け持つホスト計算機と接続されている。また、本システムでは画像表示部がメモリユニットに接続されており、これが演算過程の動的画像表示を可能とするゆえんである。

データフロープロセッサは、図2のように、アドレッシングユニットとアリスメティックユニットから成る。アドレッシングユニットは、ビット演算を含む固定小数点演算を行う演算器とメモリインタフェースで構成されており、アリスメティックユニットは、各々 150 ns

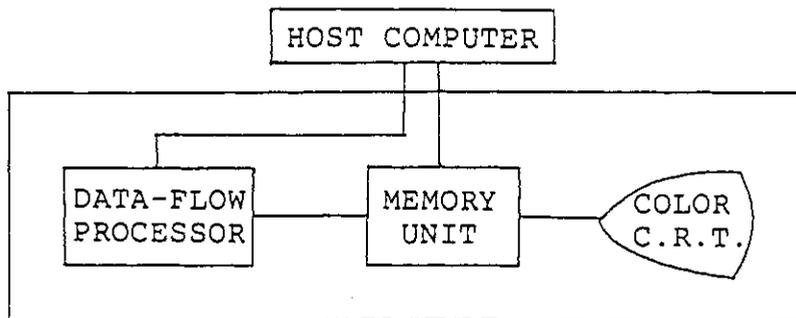
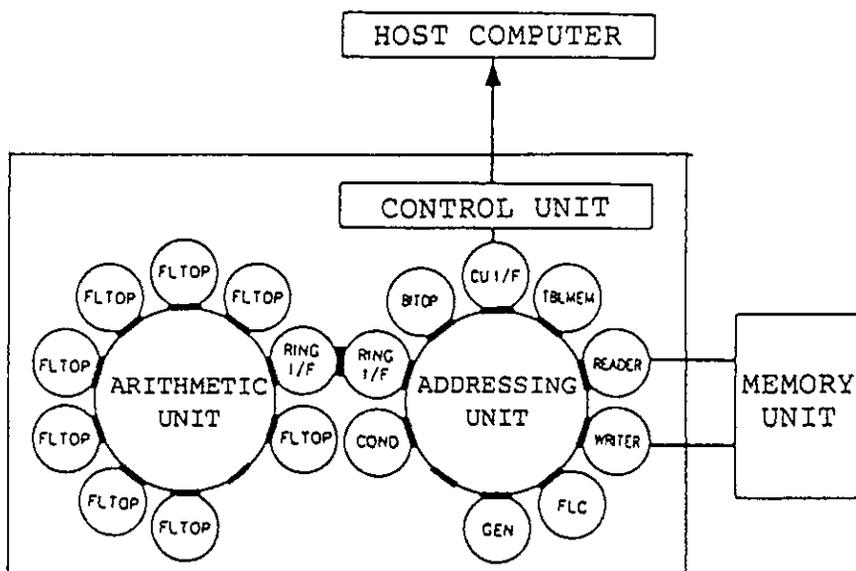


図1 システムの全体構成

Fig. 1 Outline of the System Configuration



FLTOP : Floating Operator
 FLC : Flow Controller
 GEN : Generator
 COND : Conditioner
 BITOP : Bit Operator

図2 データフロープロセッサの構成
 Fig. 2 Configuration of the dataflow processor

(ナノ秒)の演算速度をもつ8個の浮動小数点演算器で構成されている。各ユニットとも演算器をリングバスで接続した構造で、両者はリングインタフェースを介して結合されている。演算の実行は、メモリから適当な量のデータをリングバス上へ流すことにより、予めホストからプログラムがロードされている各演算器で並列に行われる。

このように、NEDIPSにおける演算処理は、プログラムではなく、データの流れによって制御される。すなわち、必要なデータが揃いしだい演算が実行される。このことは、ハードウェアの能力が最大限に引き出され、並列処理が自動的に行われることを意味する。

加算、乗算のような二項演算は一对のデータの到着を待って処理が実行される。このような場合、先に到着したデータは各演算器のキュー(queue)領域に貯えられ、相手のデータが到着ししだい処理が実行される。

さらに、データが演算器間を流れている間、ノイマン型計算機のボトルネックとなっている演算器とメモリ間の入出力動作がなく、したがって演算処理能力が入出力動作に影響されないという利点がある。

3. プログラミング技法

データフロー演算のためのプログラムは、データフローグラフで表わされる(図3参照)。フローグラフで、アクタは加算、乗算などの演算を表わし、アークはデータの入(出)力線を表わす。データフローグラフにおける計算は入力アークにデータを表わすトークンを置くことによって始まる。アクタの発火(演算の実行)時期と演算結果は入力データのみで定まり、他のアクタとは全く独立である。すなわちアクタ同士は互いに独立に並行実行することができる。

データフローグラフに基づいて、アセンブラ言語でコーディングされたNEDIPSのプログラムはテンプレートと呼ばれる。テンプレートは、各演算器へ入ってくるデータの処理の仕方と、処理結果の行先を規定する。処理の順序は規定されず、データは順序や記憶場所に関係なく、付加された名前によってのみ識別される。

プログラミング及びその実行例として、画像データのエンハンスをとりあげる。エンハンスとは原画像の濃度値に線形変換を施すことにより、階調を増し見易い画像を生成する操作である。

画像の大きさは、1024×1024画素とし、各画素はチャンネルあたり1バイトを占める4チャンネルのデータから成る。変換式は、

$$Y_{ij} = a_j \cdot X_{ij} + b_j$$

$$(i = 1, \dots, 1024^2 ; j = 1, 2, 3, 4)$$

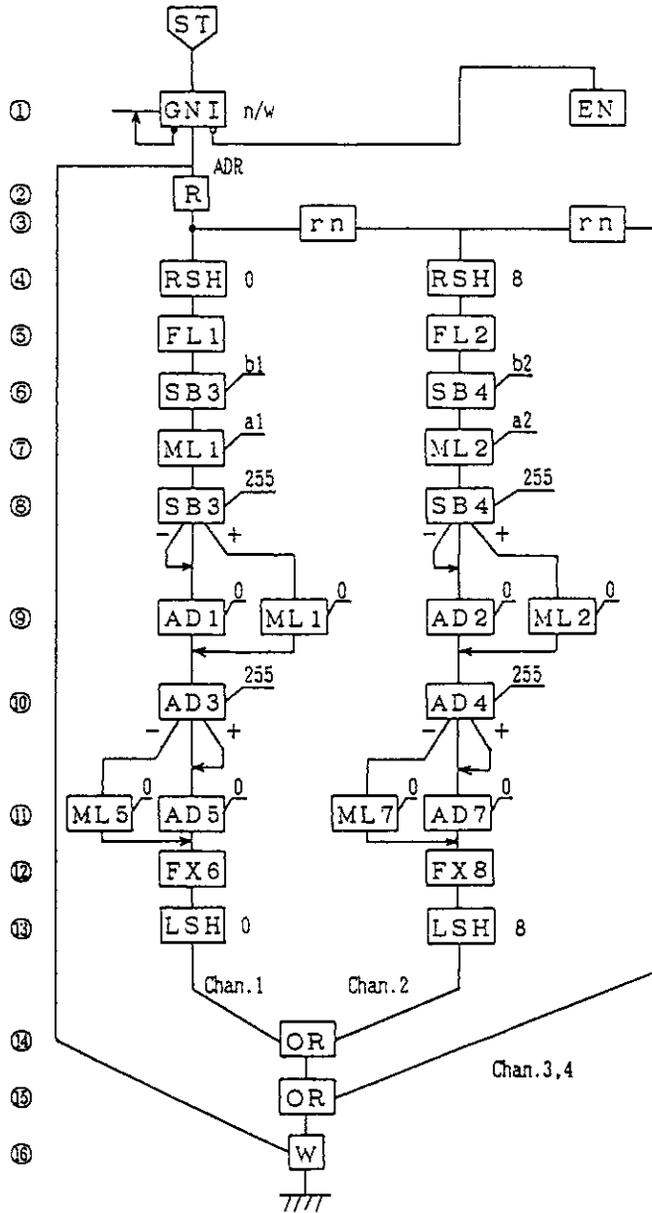
ここに、 X 、 Y はそれぞれ原画像および生成された画像の濃度値を表わし、 a 、 b はチャンネルごとに予め与える定数である。

これは、画像処理における基本的な変換であるが、ここではデータフロー計算機による高速並列処理の例として示す。

図3は、画像のエンハンスに関するデータフローグラフである。フローグラフで縦方向の列が各チャンネルごとの処理に対応している(図3ではチャンネル3と4は省略している)。また、縦方向に配置されている演算器は、概ねパイプライン並列に作動する。

演算実行の結果、実測の演算時間は、1024×1024画素に対して、1チャンネルの場合約0.51秒、4チャンネルの場合約1.17秒であった。このことから、演算のステップ数が4倍になっても、演算時間はほぼ2倍にしかならないことがわかる。これは4つのデータの流れる、ある程度並列に処理されていることを意味する。

ここで、図3から数えられる演算回数と、上述した実測の演算時間から、演算速度を算出してみる。演算回数は1チャンネルの場合13ステップ、4チャンネルの場合49ステップである。



GNI : Generation of address
 rn : rename
 RSH : Right Shift
 LSH : Left Shift
 (the number on the left encircled signifies the operation step)

図3 画像エンハンスに関するデータフローグラフ
 Fig.3 Dataflow graph of the image enhance program

したがって演算速度は、それぞれ、 $13/0.51 \div 25$ メガ回/秒、 $49/1.17 \div 43$ メガ回/秒となる。ここに、計数の対象となる演算が浮動小数点演算に限られていないのでこのような単位を導入したが、前述したようにNEDIPSの演算器の演算速度が、浮動小数点演算を含めて一律に150 nsであることから、これは通常用いられている演算速度の単位MFLOPS (Million Floating-point Operations per Second) 相当と考えてさしつかえない。

4. 数値計算と演算過程の画像化

数値シミュレーションの演算過程を動的画像として可視化することにより、シミュレーションモデルの視覚的な検証が容易となる。この効果を目標に、ここでは典型的な微分方程式である二次元非定常熱伝導方程式の陽解法による差分法について、演算時間の見積りと演算過程の画像化を行った。

基礎方程式は、

$$\frac{\partial u}{\partial t} = k \cdot \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

$\Delta x = \Delta y = \Delta s$ として差分化すると、

$$u_{i,j,t+\Delta t} = u_{i,j,t} + k \cdot \frac{\Delta t}{(\Delta s)^2} \cdot \left(-4 u_{i,j,t} + u_{i-1,j,t} + u_{i+1,j,t} + u_{i,j-1,t} + u_{i,j+1,t} \right)$$

収束のための有効条件は、

$$k \cdot \frac{\Delta t}{(\Delta s)^2} \leq 0.25$$

ここでは簡単のために、 $\Delta t = \Delta s = 1$ 、 $k = 0.25$ とし、境界条件として長方形領域の周囲の4辺に熱源として正の一定値を与え、内領域の初期値は0とする。

データフローグラフを図4に示す。ある格子点の温度が Δt だけ前の周囲の値から計算されると、直ちに擬似カラー化(値の増加に対応して青から緑をへて赤の配色を多くする)され、カラーモニタに表示される。図4では擬似カラー表示の内容は省略しているが、これを含め時刻 t における格子点ごとの演算回数は33ステップである。

一方、求解領域を 510×510 点として時刻 t における演算時間を実測したところ約0.277秒であった。したがって演算速度は、

$$33 \times 510^2 / 0.277 \div 30 \text{メガ回/秒}$$

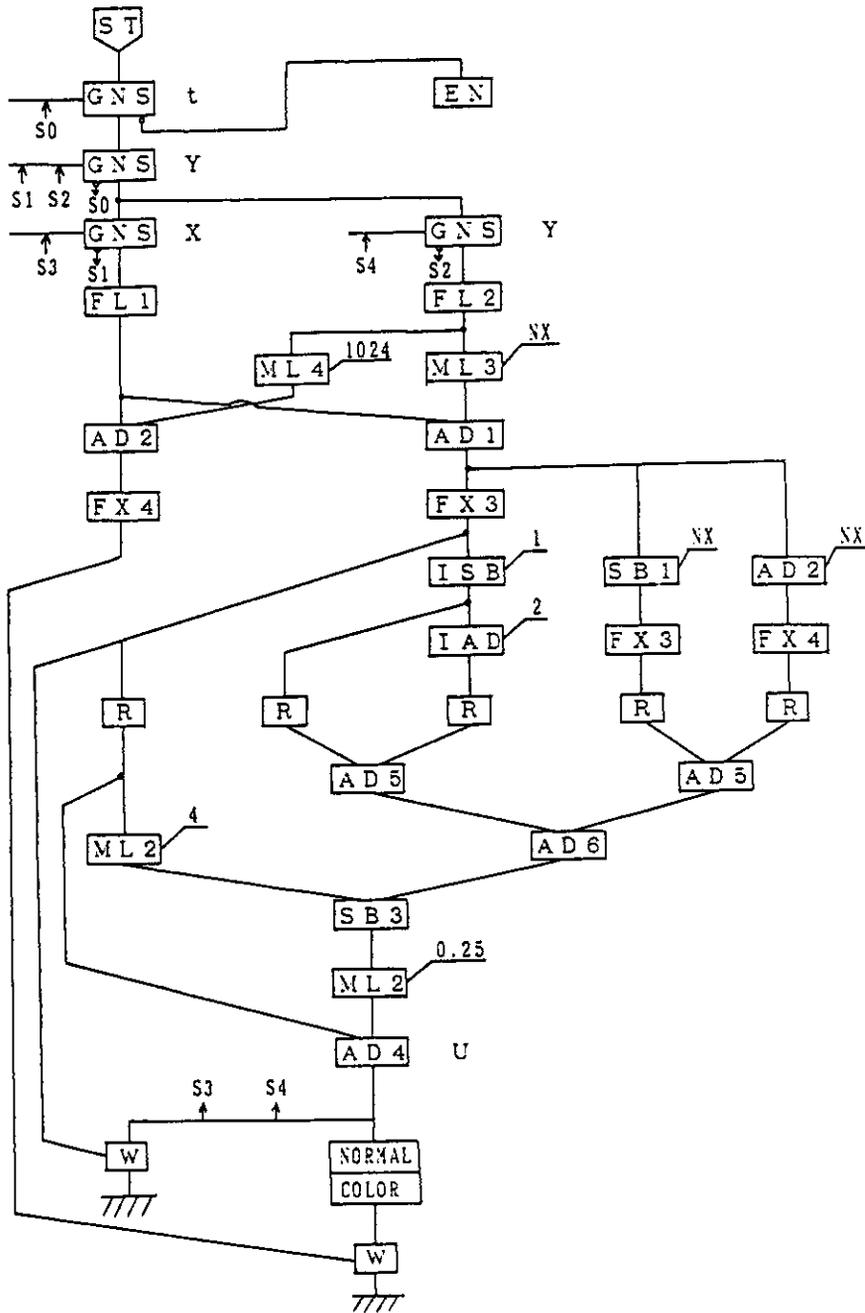


図4 熱伝導方程式の数値解析法に関するデータフローグラフ
 Fig. 4 Dataflow graph for numerical solution of the equation of heat conduction

と算出される。

この程度の大きさの領域に対してこの程度の演算速度が得られれば、対象領域の温度分布の擬似カラー表示により、時間経過とともに周囲の熱源から中央部へ向って熱が伝わって行く様子が動的画像として観察され、このような手法が一般的なシミュレーションモデルの視覚的検証に有効であるとの見通しが得られた。

一方、図4のフローグラフでは、ある格子点における演算のたびに、メモリ上の周囲4点の格子点値の入力にREADERが介在している。これに対し演算器のキュー領域を利用する遅延演算をとり入れることにより、各格子点値の入力に介在するREADERの回数を減らすことができ、この技法により全体の演算時間の短縮が見込まれている。

5. おわりに

基本的な数値計算2例について、データフロー計算機NEDIPSの性能を調べた。その結果、ほぼ30MFLOPS相当の演算速度が得られるとともに並列処理の効果が確認された。

また、この程度の演算速度のもとに、時間的に推移する演算過程をリアルタイムで画像化する方法が、シミュレーションモデルの視覚的検証に有効であるとの見通しを得た。

現在、形状を含む境界条件に関する制限を緩くした場合の前記熱伝導方程式の解法や、表層地盤への降雨浸透モデルに表われる擬似放物型偏微分方程式のS. O. R. (逐次過緩和法)による差分法などについてデータフローグラフを作成中である。

(1990年2月2日原稿受理)