

2T-7

データフローマルチプロセッサのための シミュレータDFG/DD

寺田 秀文
静岡大学 工業短期大学部

1. はじめに

ノイマン型計算機やデータフロー計算機などを基礎とした超並列・超多重処理計算機の研究が各所で行われている。近年、VLSI技術などの進展により、この超並列・超多重処理計算機は実現性のあるものとなり、次世代計算機としての期待が大きくなっている。

さて、筆者も超並列・超多重処理計算機を実現すべく多数の処理要素PMCを相互結合させたデータフローマルチプロセッサを研究中であり、そのシミュレータDFG/DDの開発をほぼ終えている。

本報告ではDFG/DDを中心に、目的としているデータフローマルチプロセッサの構想とDFG/DDにおけるデータ構造実現について述べる。

2. DFG/DDの背景

2.1 データフローマルチプロセッサ

本研究が目的としているデータフローマルチプロセッサの構想を示すために、結合形態をトラス形とした例を図1に示す。

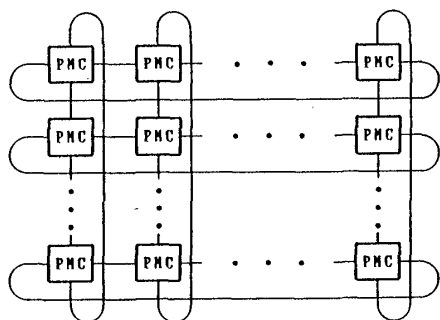


図1. データフローマルチプロセッサ

図において、PMCはプロセッサ、メモリ、通信機構などから成る処理要素で、その要素数は比較的大きな数を想定している。また、本データフローマルチプロセッサ

ではグローバルなメモリは考えておらず、アクタ（命令、オペランド）やデータ構造などのすべてはPMCのメモリ上に分散されることになる。従って、本システムは疎結合のデータフローマルチプロセッサと言えよう。

詳細（処理要素PMCの数、機能、結合など）については、次回の報告とする。

2.2 データ構造

データフローグラフの実行制御はデータ伝搬により行われる。従って、純粋なデータフローモデルの実行時におけるデータ構造伝搬は、一つのデータとしてデータ構造全体が伝搬される。しかし、これではいたずらに通信量増加を招くだけであり現実的でないため、データ構造伝搬はデータ構造のポインタ伝搬により代行される。さらに、データ構造の更新におけるデータ構造全体の複写にはデータ構造の部分共有方式などが採られている。

さて、これらの方式は構造メモリ（structure memory）により実現されるのだが、一般的に構造メモリはグローバルな位置にあり、本データフローマルチプロセッサのような構成には適さない。故に、DFG/DDでは分散化を考慮し、データ構造を実現した。

DFG/DDでのデータ構造はポインタ伝搬・データ共有方式を基本にし、データ構造に特有なアクタ（S-Actor、I-Actor）で実現される。このS-ActorとI-Actorはデータ構造の構造と具現をそれぞれ表現する。図2は一次元整数配列の実現例を示している。

データ構造はそれを定義する関数の呼出し時に生成され、終了時に削除される。S-Actorは具現（データ書込み）要求を受け取るとI-Actorを生成し、データと具現要求内の具現情報をI-Actorに与え、リストの先頭にリンクする。また、参照要求を受け取るとその要求をリンクされているI-Actorに伝える。I-Actorは参照要求に対して具現情報のマッチングを行い、マッチングに成功したならば応答のトークンを送り、そうでないならば次にリンクされたI-Actorに要求を伝える。

図2では配列要素a[1], a[3], a[N]などに具現が行われている。

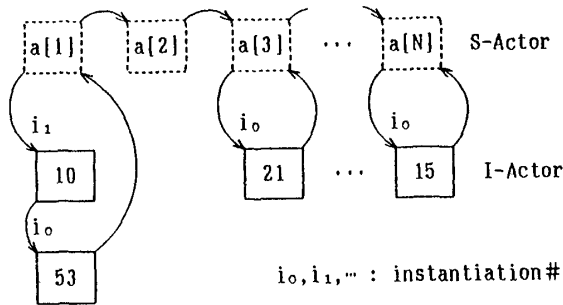


図2. 一次元配列の実現例

3. シミュレータDFG/DD

3.1 概観

独自のウィンドウシステム上に構築されたDFG/DDは、データフローグラフ開発のためのエディタ、ステップ単位で同期実行をするシミュレータなどから構成される。また、DFG/DDのすべてはC (VAX C) 言語で記述し、グラフィックはGKSに準拠とした。

データフローグラフ編集や実行などのすべてのコマンドは、キーボード (コンソールウィンドウ)、マウス (コマンドメニュー) のどちらの入力デバイスでもほぼ同等に実行できるように実現されている。

3.2 シミュレーション

DFG/DDでは実行過程の視覚化を図っており、実行時のトークンはアーク上に表示される。図3の実行ウィンドウ例を参照されたい。

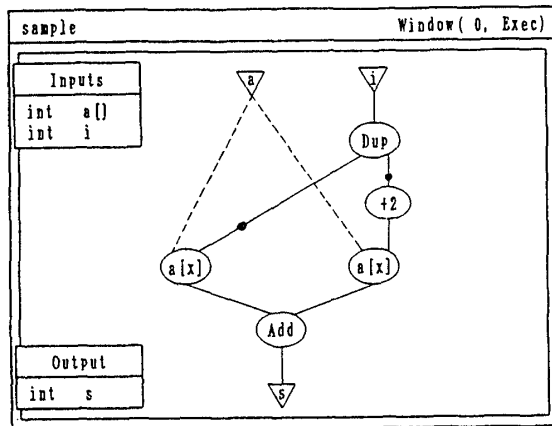


図3. 実行ウィンドウ例

DFG/DDでは、▽で入出力ポートを表現する。従って、図3の関数は2入力関数である。また、アクタa[x]は整数配列への参照アクタであり、点線アークが配列へのポインタを、実線アークが要素指定子を伝える。

トークン受け渡しは、入力ポートへはキーボード (トークン入力ウィンドウ) か外部ファイルから、出力ポートからはトークン出力ウィンドウか外部ファイルに行われる。また、実行はステップ単位に同期的であり、トリガによるステップ実行や発火可能なアクタがなくなるまでの実行などがサポートされている。

現在、DFG/DDは Micro VAX I に実現されているため、実行速度の関係上、シミュレーション時には「通信遅延は一定」などの簡略化が行われている。

4. おわりに

本報告ではシミュレータDFG/DDを中心に、超並列・超多重処理計算機を目指したデータフローマルチプロセッサの基本構想について述べた。さらに、本データフローマルチプロセッサが採用する疎結合の構成を考慮したデータ構造実現 (データ構造に特有なアクタ、S-ActorとI-Actorによるデータ構造実現) について述べた。

DFG/DD開発は本研究計画における初期段階のものであり、データフローマルチプロセッサ実現のためには解決すべき問題が多くある。現在、処理要素PMC機能 (命令セット、メモリ量、通信容量) やPMC結合形態などに対する設計指針を獲得するために、より詳細なシミュレーションを進めている。

【謝辞】

本シミュレータ開発にあたり、グラフィック機器に対して御配慮を賜ったヤマハ株式会社電子事業本部の皆様 に深く感謝いたします。

【参考文献】

[1] J.Rumbaugh, "A Dataflow Multiprocessor", IEEE Trans. Computers, vol.C-26, no.2, Feb.1977
 [2] A.L.Davis, R.M.Keller, "Data Flow Program Graphs", Computer, vol.15, no.2, Feb.1982
 [3] J.-L.Gaudiot, "Structure Handling in Data-Flow Systems", IEEE Trans. Computers, vol.C-35, no.6, June 1986