

第10回計算機アーキテクチャ国際会議の紹介

On the 10th Annual International Symposium on Computer Architecture

ーデータフローアーキテクチャの最近の話題ー

Current Topics on Dataflow Architectures

高橋 直久

Naohisa TAKAHASHI

日本電信電話公社 武蔵野電気通信研究所

Musashino Electrical Communication Laboratory, NTT

1. はじめに

計算機アーキテクチャ国際会議に出席し、会議終了後英・仏のデータフローマシン等の研究プロジェクトを訪問する機会を得たので、その概要を報告する。

本会議はIEEE Computer SocietyとACMの共催で毎年開かれる計算機方式に関する代表的な国際会議である。今年で第10回目であり、6月13日から5日間ストックホルムの王立工科大学にて開催された。会議には、日本からの参加者22名を含め23ヶ国約300名が集まり熱心な討論が繰り広げられた。今回は日本からの発表が12件（内3件は招待講演）と例年になく多かった。また、最終日には第5世代計算機に焦点をあてた"Special Day"が設けられ

るなど、この分野での日本のアクティビティが非常に高まってきたと感じられた。なお、1986年には、初めて日本で本会議が開かれることになった。

本報告では、まず会議の概要を紹介する。次に、本会議の発表を中心にデータフローアーキテクチャに関する最近の話題を、実行制御機構、専用アーキテクチャ、及び、性能評価の観点から整理して紹介する。

2. 会議の概要

会議の概要を表1に示す。本会議は、Tutorial Day, Program Sessions, および、Special Dayで構成された。

表1. 会議の概要

	内 容	
主旨	計算機アーキテクチャに関する国際会議。	
主催者等	IEEE Computer Society, ACM 等。第10回目。	
参加者	23カ国 約300名。(スウェーデン130名, 米国68名, 日本22名, 等。)	
一般論文	総数	48件
	分野	データフローアーキテクチャ(7件), 結合網の構成法(6件), VLSI向きアーキテクチャ(4件), キャッシュメモリ(4件), 高級言語マシン(4件), 画像処理マシン(4件)等
	国別	米国32件, 日本9件, スウェーデン・西独3件, 等。
Special Day	項目	人工知能の応用とコンピュータアーキテクチャへの影響。
	論文	新世代計算機概論, データフローマシン, リダクションマシン(各1件), ICOT関連(総論, 逐次型推論マシン, データベースマシン各1件)。
Tutorial Day	項目	データベースマシン(ウィスコンシン大学 D. J. DeWitt), VLSI 化技法(インテル社 D. Hammerstrom)。

Tutorial Dayでは、インテル社の D. Hammerstrom 氏の VLSI に関する講演とウィスコンシン大学の D. J. DeWitt 氏のデータベースマシンに関する講演が並行して行われた。

Program Sessionsには、148件と例年になく多くの論文が投稿され、このうち48件が採録され17のセッションに分かれて発表された。発表分野は、データフローアーキテクチャ(7件)、ネットワークアーキテクチャ(6件)、VLSI 向きアーキテクチャ(4件)、キャッシュメモリ(4件)、高級言語マシン(4件)、画像処理マシン(4件)などである。日本からの発表では9件の内データフローアーキテクチャと高級言語マシンが各3件で、発表分野にやや偏りがみられた。

並列計算機のアークテクチャに関する発表が大きな割合(全体の約2/3)を占め、その基本技術となるネットワークアーキテクチャの発表件数が多いのは例年通りである。今回は、この中でデータフローアーキテクチャに関する発表が特に多いのと画像処理アーキテクチャの発表が多くなったのが目立つ。

Special Day では、人工知能の応用とコンピュータアーキテクチャへの影響と題してパネルディスカッションと招待講演が行われた。招待講演では、新世代計算機に関する概論、リダクションマシンとデータフローマシンに関する論文(各1件)、及び、日本の第5世代計算機プロジェクトに関する論文(3件)が発表された。ICOTに対しては各国の参加者が大きな関心を持ち沢山の質問が寄せられた。

以下に、本会議で発表された論文の中からいくつかの話題を紹介する。なお、データフローアーキテクチャに関しては3.で述べる。

(1) VLSI アークテクチャ

Tutorial day には、インテル社の Hammerstrom が、"The Migration of Function Into Silicon" と題して、VLSI 化の効果、VLSI 化すべき機能の分類、及び、VLSI 化技法について解説した。ここでは、インテル社のチップなどの実例を多く用いながら VLSI に関して非常に体系的で整理された講演が行われた。

VLSI アークテクチャのセッションでは、4件の発表があった。内2件は、CMU の Kung のグループからの発表で、プログラマブルシストリックチップ(PSC)と大規模プロセッサアレイの同期の問題に関するものである。PSC は、nMOS の1チップシストリックプロセッサであり、マイクロプログラム制御により多様なシストリックアルゴリズムの構成要素として使用できる[1]。

他の2件は、ビットシリアル演算のプロセッサを CCC (Cube Connected Cycles) ネットワークで結合したマルチプロセッサ、及び、関係データベース演算用の木構造マルチプロセッサに関するものである。

(2) 画像処理アーキテクチャ

画像処理用アーキテクチャに関して4件の論文発表があった。各アーキテクチャを分類すると、SIMD (Single Instruction stream and Multiple Data stream) 型プロセッサアレイが2件、MIMD (Multiple Instruction stream and Multiple Data stream) 型マルチプロセッサが2件である。

前者の内の1件は、SIMD 型プロセッサアレイをピラミッド状に多層に接続したピラミッドマシンに関するものである。このマシンは、矩形格子面上の結合と4進木状の垂直方向の結合を組合せたネットワーク構成で、画像データのローカルな解析とグローバルな解析とを同時に行えるようにしている[2]。

我国からは、阪大の画像生成用マルチマイクロプロセッサシステム LINKS-1が発表された。LINKS-1は、64台の処理ユニットからなり、三次元画像生成に必要なプロセスをパイプライン処理すると同時に各プロセスで画像データを分割して並列処理することにより高速画像生成を実現している[3]。

(3) リダクションマシン

招待講演で、インベリアルカレッジの M. Reeve よりリダクションマシン ALICE (Applicative Language Idealized Computing Engine) に関する発表が行われた[11]。ALICE は、グラフィダクションモデルに基づき関数型・論理型プログラムを並列実行するマシンである。

ALICE プロジェクトでは、図1のように Software Oriented な研究を進めて、高級言語と親和性の高いアーキテクチャの開発を目指している。まず、ALICE のオブジェクトコードに相当する CTL (Compiler Target Language)

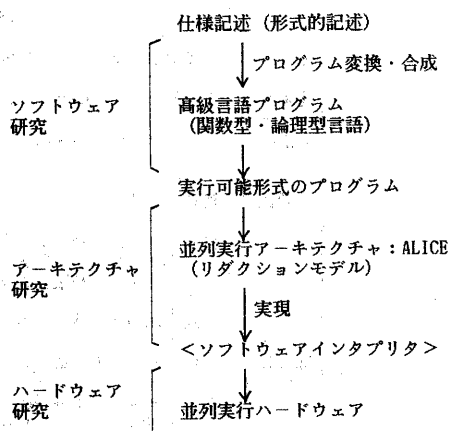


図1. ALICE プロジェクト (インベリアルカレッジ)

を設定し、言語処理系の開発とアーキテクチャの詳細設計を並列に進めている。これまでに、Lisp、関数型言語 HOPE、並列論理型言語 Parlog から CTL へのトランスレータを試作している。アーキテクチャに関しては、CTL プログラムを実行する ALICE シミュレータを開発し、それを用いて詳細な特性評価を行っている。また、現在、16 台の要素プロセッサから成るプロトタイプシステムを開発中である。

3. データフローアーキテクチャの最近の話題

今回の会議での発表件数を見ても分かるように、データフローアーキテクチャの研究が現在活発に行われている。データフローアーキテクチャ研究の全般的なサーベイは文献 [12, 13] に詳しく述べられているので、ここでは、本会議での発表を中心に最近の話題を紹介する。

3.1 実行制御機構

データフロー原理は非常にシンプルで拡張性に富んでいるので、Dennis の提案以来、様々なアーキテクチャが検討されている。特に、プログラムの持つ並列性を最大限に引き出すことを狙って、各種の試みがなされてきた。例えば、関数、或いはループの処理を展開して並列実行するために、データフローグラフのコピー機構 [16] やデータに色 (タグ) を付けて識別する機構 (色付きトークンの概念) [6, 14, 15, 25] を組み込むことが提案されている。

また、配列などの構造体データの処理で並列性を高めるために、構造体データの作成側プログラムと読み出し側プログラムとの間で構造体データの各要素毎に非同期制御を行う方式が各種提案されている。具体的には、構造体データを分解して扱い各要素にアクセスする演算をデータ駆動で行う方式 [6, 14]、及び、要素毎に読み出し・書き込みの同期制御機能を備えた構造体データ格納用共有メモリを設ける方式 [15, 21, 25] がある。これらの方式により、読み出し側プログラムでは、構造体データの値が全て定まるのを待たずに、値の定まった要素から順に処理開始可能となる。

上記のように論理的な並列性を最大限に引き出すことを狙った方式の検討に加え、有限なハードウェア資源を有効に利用するようにプログラムの持つ並列性を効果的に引き出す方式、或いは、部分計算結果を共有可能にして不要な計算を減少させる方式等が提案され始めている。また、実行制御のオーバーヘッドを減少させる検討もなされてきている。これらのアプローチでは、プログラムの構造を実行前に解析し、その結果を実行制御機構で利用する点が共通している。具体例を次に紹介する。

(1) ワーキングセットの概念

データフローマシンでは、逐次型計算機のようにプログラムの実行の局所性を利用した効率的な記憶階層システムを作るのが困難である。この理由のひとつは、オペランドデータの揃った全ての命令が並列に実行可能になるので、プログラムの動作 (実行命令の動き) を予測し近い将来実行可能になる命令を選び出すのが難しい点にある。

慶応大学では、この問題を解決するためにデータフロープログラムの実行の同時性と局所性に着目してワーキングセットの概念をデータフローマシンに導入することを提案している [4]。

この方式では、実行前に、データフローグラフを解析して各命令に D, E, L レベルと呼ぶ 3 つのパラメータを付与する。実行時に、このパラメータの値に従って、近い将来に実行が予想される命令を選び主記憶に置く。ここで、D, E, L レベルは、それぞれ、データフローグラフの入力ノードからの最短経路長、同じく最長経路長、および、出力ノードからの最長経路長をそのデータフローグラフ全体の最長経路長からひいたものである。シミュレーション結果により、分岐・ループ・関数呼出を含まない静的なデータフローグラフでは L, E レベルの小さな命令から順に主記憶にロードする方策を用いると高い性能が得られることを示している。

(2) コントロールフローの導入

Lawrence Livermore National Laboratory (LLNL) の Piecewise data flow architecture (PDF) では、プログラムの各ブロックの処理をデータフローで実行し各ブロック間の制御をコントロールフローで行っている [5]。

コンパイラのフロー解析でプログラムをサイクリックグラフを含まないブロックに分割し、ブロック内の各命令間のデータ依存関係に従った命令の番号付けとブロック間の実行シーケンス制御情報の付与を行う。このような命令の番号付けを行うと比較的小容量のレジスタで発火制御用マッチングストアを実現できる。また、ブロック間の実行シーケンス制御情報に従うと、ハードウェア資源の使用率が高く、かつブロック間の実行が最大限オーバーラップするように各ブロックの実行順序を制御できる。

(3) 準動的な資源割付

通研のデータフロープロセッサアレイ実験システム Eddy では、実行時の適応的な動的資源割付と実行前の静的資源割付とを組合せた準動的な資源割付を行っている [6]。この割付方式の狙いは、実行制御のオーバーヘッドを減らすこと、及び、プロセッサ間の通信の局所性を活かすようにプログラムを割りつけることにある。具体的には、次のような割付を行っている。

(i) 配列処理プログラムでは、実行前に配列要素を分解し要素プロセッサ (P E) に静的に割付、実行時にはデータ駆動原理に従い個々の演算命令に各 P E の演算器群の中から一つの演算器を動的に割りつける。

(ii) 実行前に関数間の呼出関係を解析して、実際の関数呼出時に引数に付与すべき実行環境識別用タグ (色) の割付方法を決定しておく。実行時には、この決定に従ってタグに簡単な演算を施すことにより、システム内でユニークなタグを割りつけることができる。

偏微分方程式の反復解析プログラムの実行結果では、プロセッサ間のデータ転送遅延がある程度大きくなってもプログラムの実行時間が増大せず、プロセッサ間通信の局所性を活かすように割付を行った効果が示されている [17]。

(4) 世代差演算子モデル

色付きトークンの概念を導入すると、異なる色 (タグ) を持つトークン間では相互のインタラクションがないので、データフローグラフを複数の実行環境で共有できる。例えば、図2では、ノード "f" は、"a", "b", "c" の環境で共有されている (図で、 $\boxed{x}_a$ は値 x, 色 a を持つトークンを表す)。

しかし、異なる色をもつトークン間で相互のインタラクションがないことは、ある環境で計算された結果を別の複数の環境で共用することができないことを意味し、次のような問題を引き起こす。

(i) 高階関数の実現で基本となる閉包 (関数とその関数が評価される環境の対) [19] を扱えない。

(ii) 部分計算 [20] 機構を実現できない。

これらの問題は関数型プログラムを効率良く実行するデータフローマシンを実現する上で解決すべき重要な問題である。

通研の提案した世代差演算子モデルでは、色付きトークンの概念を拡張した演算子を導入することにより上記問題を解決しようとしている [18]。

このモデルの特徴は、割りつける色の集合に図3のような木状の半順序関係を導入し、特定の関係にある2つの色をそれぞれ持つようなトークン間で演算を行わせる点にある。以下に具体例を使って世代差演算子の実行開始規則を示す。

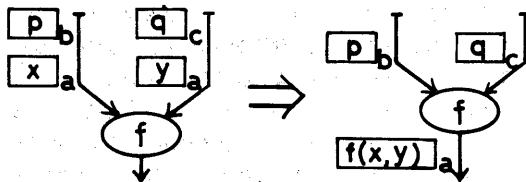


図2. 色付きトークンモデルの演算の発火例

世代差演算子は、図4に示すように入力アーク間に特別のアークと整数 n が付けられる。この特別のアークをブリッジ、 n を世代差と呼ぶ。今、演算 f の両オペランドに到着したトークンの色 a と b の間に半順序関係が存在し a から b への組成列の長さが n に等しい時、演算 f が実行可能となる。例えば、色の関係が図3を満たす場合には、 $a = a_1$, $b = b_1$, $n = 1$ ならば演算 f は実行可能である。

文献 [18] では、このモデルを実現するための色の操作命令とプログラム中の各演算子の世代差を決定する規則を示し、これを使った閉包の処理法と部分計算機構について述べている。

3. 2 専用アーキテクチャ

データフローアーキテクチャは柔軟性の高い並列計算機方式として各種応用に適用することが検討されている。ここでは、数値計算、記号処理、信号処理の分野での最近の話題を紹介する。

(1) 数値計算

前述の LLNL の PDF, 通研の Eddy, 及び、ポーランドの Marczyński らのシステムが数値計算用データフローマシンとして本会議で発表された。この他に、MIT の Dennis のグループと電総研の Sigma-1 のグループが大規模実験シス

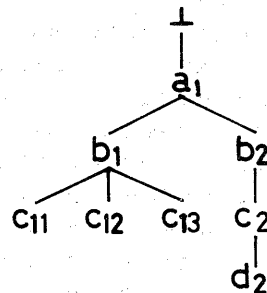


図3. 木状構造の色の集合の例

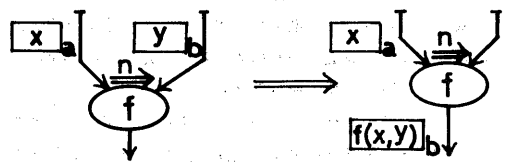


図4. 世代差演算子の発火例

テムの開発を含む活発な研究を進めている。

Dennis らのシステムは、ルーティングネットワークで結合された8台のプロセッサ (PU) からなる [22]。各PUではマイクロプログラムによりデータフロープロセッサがシミュレートされるので、このシステムで各種アーキテクチャの評価が可能である。

Sigma-1のアーキテクチャの主な特徴は、マッチングストアに並列連鎖ハッシュハードウェアを導入した点、及び、B構造と呼ぶ非同期アクセス制御を行う構造体メモリを備えた点にある [21]。1 MFLOPS の PE を100台程度接続した大規模な超高速システムを目指して、現在、PEのプロトタイプの詳細設計を進めている。

また、数値計算用データフロー実験機 LAU System-1 [16]を試作したフランスの ONERA-CERT では、最近、データフローとコントロールフローの概念を融合させたスーパーコンピュータの研究開発を進めている。これは、コントロールフローで動作する従来のアレイプロセッサやパイプラインプロセッサをネットワークで結合し全体をデータフローで制御する構成である。これまでに、マシンの概念設計、言語設計、及び、シミュレーション評価を行っている [23]。

(2) 記号処理

記号処理の分野でもデータフローアーキテクチャの研究が活発であり、電総研、通研などが大規模なハードウェアプロトタイプを開発中である [7, 25]。また、マンチェスタ大学では、関数型言語の処理系や構造体メモリの検討を進め、同大学のプロトタイプシステム [14] 上で記号処理も行おうとしている。

(3) 信号処理

ヘルシンキ工科大学では、データ駆動型の信号処理プロセッサを検討している [8]。アーキテクチャの特徴は、色付きトークンの概念を用いてリエントラントなプログラムとストリームを実現している点、及び、演算の機能レベルを高く設定しベクトル演算のように構造体データをオペランドで使え、かつ、オペランドの数が任意である点にある。これまでに、アーキテクチャの設計、シミュレーション評価、制御部の VLSI 化の検討などを行っている。

4. 3 性能評価

(1) DDDP の評価

沖電気では、データフローハードウェアシステム DDDP (Distributed Data Driven Processor) を試作し、性能評価を行っている [9]。DDDP は、マイクロプログラム制御のデータフロープロセッサが4台リングバスで接続された構

成である。ハッシングハードウェアでマッチングストアを実現し、演算部にビットスライスマイクロプロセッサを用いて、0.73 MIPS (Million Instruction Per Second) の性能を得ている。

4クイーン問題などの例題プログラムの実行結果により、4台程度の小規模マルチプロセッサ構成でも命令の割付方法がシステムの稼働率に大きな影響を与えることを示している。

(2) MIT Arvind の評価法

MIT Arvind は、本会議の招待講演で、次の2つの観点から従来のマルチプロセッサシステムの問題点を定性的に解析し、データフローアーキテクチャの有効性を主張した [10]。

(i) メモリアクセス要求が出されてから結果を得るまでに要する時間遅れを許容できるか。

(ii) 並列度を損なわずにデータの共有ができるか。

彼は、講演の中で、大規模な実験システム MEF (Multi-processor Emulation Facility) の開発構想を発表した。

MEF は、64台程度の Lisp マシンを7次元超格子ネットワークで結合した構成で、大規模データフロープログラムの実行環境を与える。MEF での実験を通して上記2つの観点からデータフローアーキテクチャの有効性を証明することを狙っている。

(3) マンチェスタ大学のプロトタイプマシン

マンチェスタ大学では、1981年にプロトタイプマシンを試作し、性能評価を行っている。このマシンは、1リングの巡回パイプライン型データフローアーキテクチャであり、複数の演算器を並列に動作させることができる。現在、12個の演算器を実装し、小規模プログラムの実行結果では1~1.8 MIPSの性能を得ている [24]。FFT、偏微分方程式の反復解析、論理シミュレーションなど小規模ではあるが広範囲の応用プログラムの実行特性を解析している。その結果、これらのプログラムは、データ依存関係で決まる論理的な並列度がそれぞれ時間的に様々に変化するが、いずれも演算器の個数にほぼ比例して実行速度が向上し並列性が十分にあることを示している。

4. おわりに

第10回計算機アーキテクチャ国際会議の概要とデータフローアーキテクチャの最近の話題を紹介した。

本会議では、ほとんど全ての発表で、約1.0分間の質問時間を一杯に使った活発な討論が行われた。また、昼食後のひとときやコーヒブレイクの時も討論や情報交換の場として議論の輪がいくつもできて非常に盛況であった。この時期のストックホルムは夜11時過ぎまで明るく、連日遅くまでパーティが催され各国の参加者と交流を深めた。なお、今回は昭和59年6月4～7日に米国の Ann Arbor で開催される予定である。

最後に、報告者の興味からデータフローアーキテクチャに関する話題に偏ってしまい、本来の会議報告からずれてしまったことをお詫びしたい。会議出席と調査の機会を与えて下さり数々の御助言・御配慮を頂いた畔柳功芳基礎研究部長、塚本克治第八研究室長はじめ関係各位に深く感謝致します。

参考文献

- (1) Fisher H.L., Kung L., Monier L.M. and Dohi Y., "Architecture of the PSC : a Programmable Systolic Chip," Proceedings of the 10th Annual Symposium on Computer Architecture, June 1983, pp. 48-53.
- (2) Tanimoto S.L., "A pyramidal approach to parallel processing," *ibid.* pp. 372-378.
- (3) Nishimura H. et al., "LINKS-1 : A parallel pipelined multimicrocomputer system for image creation," *ibid.* pp. 387-394.
- (4) Tokoro M. and Jagannathan J.R., "On the working set concept for dataflow machines," *ibid.* pp. 90-97.
- (5) Requa J.E., "The piecewise data flow architecture : control flow and register management," *ibid.* pp. 84-89.
- (6) Takahashi N. and Amamiya M., "A data flow processor array system : design and analysis," *ibid.* pp. 243-250.
- (7) Yamaguchi Y., Toda K. and Yuba T., "A performance evaluation of the data driven machine EM-3," *ibid.* pp. 363-370.
- (8) Kronlöf K., "Execution control and memory management of a dataflow signal processor," *ibid.* pp. 230-235.
- (9) Kishi M., Yasuhara H. and Kawamura Y., "DDDP : A Distributed Data Driven Processor," *ibid.* pp. 236-242.
- (10) Arvind and Iannucci, "A Critique of Multiprocessing von Neumann Style," *ibid.* pp. 426-436.
- (11) Darlington J. and Reeve M., "ALICE and the parallel evaluation of logic programs," (Preliminary draft).
- (12) 日本情報処理開発協会, 日本電子工業振興協会, "第5世代のコンピュータ. データフローマシン/データベースマシンの調査研究," 1982.
- (13) Treleaven P., Brounbridge D.R. and Hopkins R.P., "Data-Driven and Demand-Driven Computer Architecture," ACM Computing Surveys, Vol. 14, No. 1, March 1982, pp. 94-143.
- (14) Gurd J. and Watson I., "Data Driven System for High Speed Computing," Computer Design, Vol. 9, No. 6 & 7, pp. 91-100 & 97-106.
- (15) Arvind and Kathail V., "A Multiple Processor Dataflow Machine That Supports Generalized Procedures," Proceedings of the 8th Annual Symposium on Computer Architecture, 1981, pp. 291-302.
- (16) Plas A. et al., "LAU system architecture : a parallel data driven processor based on single assignment," Proceedings of International Conference on Parallel Processing, 1976.
- (17) 高橋, 両宮, 大淵, "シミュレーションによるデータフロープロセッサアレイ計算機の評価," 情報学会計算機アーキテクチャ研究会48-4, 1983.
- (18) Ono S., Takahashi N. and Amamiya M., "Partial Computation with a Dataflow Machine," R.I.M.S. Symposium on Mathematical Methods in Software Science and Engineering, 5th Conference, Kyoto University, 1983.
- (19) Allen J., "Anatomy of LISP," McGraw-Hill, 1978.
- (20) Ershov A.P., "Mixed Computation : Potential Application and Problem for Study," Theoretical Computer Science 18, 1982, pp. 41-67.
- (21) 島田, 平木, 西田, "科学技術計算用データ駆動型計算機 Sigma-1 のアーキテクチャ," 信学技報 Vol. 83, No. 78, 1983, pp. 47-53.
- (22) Dennis J.B., Lim W.Y.P. and Ackerman W.B., "The MIT Data Flow Engineering Model," Proceedings of the IFIP 83 Congress, 1983, pp. 553-560.
- (23) Apelantado M., Conte D., Siron P. and Syre J.C., "A MIMD supercomputer system for large scale numerical application," *ibid.*
- (24) Gurd J. and Watson I., "Preliminary Evaluation of a Prototype Dataflow Computer," *ibid.* pp. 545-551.
- (25) Amamiya M., Hasegawa R. and Mikami H., "List Processing with a Data Flow Machine," Lecture Notes in Computer Science, R.I.M.S. Symposia on Software Science and Engineering, Springer-Verlag, 1982.