

データ駆動計算機EM-4プロトタイプの開発・動作環境

岡本一晃* 児玉祐悦** 坂井修一** 山口喜教**

*三洋電機 情報通信システム研究所

**電子技術総合研究所

データ駆動計算機EM-4プロトタイプの開発過程，ならびに現在の動作環境について述べる．開発過程においては，メンテナンスアーキテクチャを採用しテストベッドを開発することにより調整およびデバッグ作業を効率化した．また，段階的な基本動作試験法がハードウェア・ソフトウェアの両面において調整，デバッグ作業に対して有効であることを確認した．現在EM-4プロトタイプの試験プログラム開発環境として，アセンブラ，シミュレータ等を開発したほか，ステップ動作機能，パケットトレース機能をサポートしており，プログラムデバッグ作業を容易にしている．また実行時の性能測定も可能であり，テストプログラムでは999MIPSを実測した．

A PROTOTYPE SYSTEM AND ITS ENVIRONMENTS OF THE DATAFLOW MACHINE EM-4

Kazuaki OKAMOTO* Yuetsu KODAMA** Shuichi SAKAI** Yoshinori YAMAGUCHI**

*SANYO Electric Co., Ltd.

**Electrotechnical Laboratory

This paper describes the process of developing a prototype of the dataflow machine EM-4 and its operating environments. The EM-4 has the maintenance architecture which makes our debugging works much easier by monitoring the condition of each processing element (PE). In addition, we developed a testbed system which is used for individually testing a PE board taken out of the prototype system. Currently, the testing environments of the prototype system consists of an assembler, a software simulator, and debugging environments - a stepper and a tracer. It also has the function to measure its run-time performance, and the system has recorded 999MIPS on a test program.

1. はじめに

データ駆動型の計算機アーキテクチャは、問題の並列性を自然な形で抽出でき、かつ同期や通信による遅延をハードウェアで吸収できるなどの利点から、高並列計算機アーキテクチャとして特に有望であると考えられる^{[1][2][3]}。しかし一方で、従来のデータ駆動アーキテクチャには、命令の先行制御ができない、レジスタの使用が困難であるなどの理由から、並列度の小さい処理に対しては十分なスループットが得られないことが多かった。また、柔軟な資源管理が困難である、パケット転送量が莫大である、などの問題点が指摘された。これを解決する手段として我々は強連結枝モデルを提案し^{[4][5]}、現在この強連結枝モデルに基づく要素プロセッサ数1000規模の高並列データ駆動計算機EM-4を開発中である。

EM-4の開発は1986年より進められているが、その第一段階として80台のシングルチップ化された要素プロセッサEMC-Rにより構成されるEM-4プロトタイプを開発した^[10]。そして現在このEM-4プロトタイプの上で強連結枝モデルの有効性の検証や自動負荷分散方式^[9]の検討、高級言語処理系の構築などを行っている。本稿ではまずEM-4プロトタイプの基本構成を紹介する(第2節)。次にEM-4プロトタイプの開発過程について述べ、ハードウェアならびにソフトウェアの開発を効率化するための手法、さらに基本動作の試験法について論ずる(第3節)。そして、EM-4プロトタイプの現在の動作環境について述べ、試験プログラム開発環境や性能測定法などを紹介する(第4節)。最後に今後の予定を述べる(第5節)。なお、EM-4の動作原理の詳細は、文献[6][7][10]を参照されたい。

2. EM-4プロトタイプの基本構成

EM-4プロトタイプは、全体として80台の要素プロセッサ(PE)がプロセッサ結合型オメガ網で相互結合された形態を持つ。データ駆動型シングルチッププロセッサEMC-Rは結合網の

要素としてのデータ交換機能を内蔵しており、各プロセッサはEMC-Rと外部メモリのみによって構成される。EM-4プロトタイプのPE群は結合網の循環路^[8]によって5台ずつ16のグループに分割されており、各グループは関数割り付けの基本単位となっている。1グループに属するPE群はそれらをモニタリングするためのメンテナンスプロセッサ回路とともに、1枚のプロセッサ基板を構成する。したがってEM-4プロトタイプは、16枚のプロセッサ基板より構成される。

EM-4プロトタイプの全体システムは、図1に示すとおり16枚のプロセッサ基板と1枚のパケットインタフェーススイッチ基板とが大域結合網で結合された形で構成され、これにパケットインタフェースプロセッサとホスト計算機が接続されている。パケットインタフェーススイッチ基板はシステムと外部との間のパケットの入出力を司る基板であり、同時にシステムクロックの供給元でもある。これはパケットインタフェースプロセッサに接続されており、ここからパケットの入出

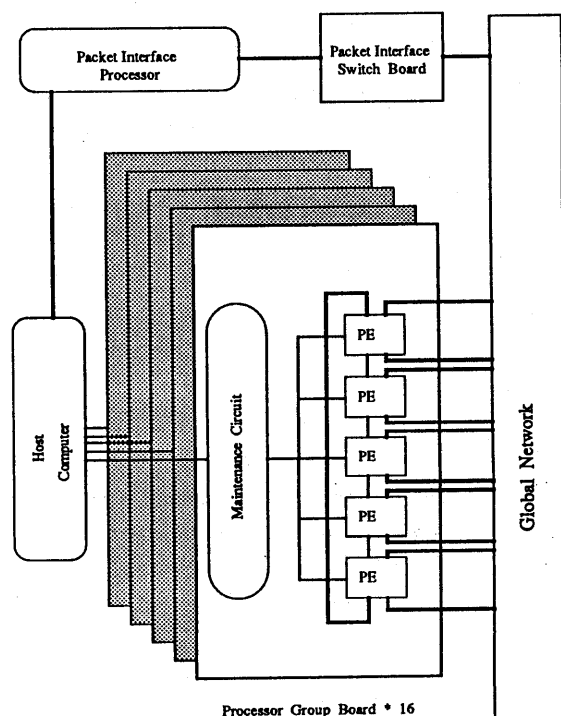


図1 EM-4プロトタイプのブロック図

力およびシステムクロックに関する制御を受ける。一方プロセッサ基板はPE群と1組のメンテナンス回路から構成され、このメンテナンス回路はシステムとは別のクロックで独立に動作して各PEの内部状態をモニタリングする。各グループのメンテナンス回路はそれぞれがホスト計算機と直接接続されていて、ホスト計算機から制御を受けている。

3. EM-4 プロトタイプの開発

3.1 開発法

本節では、EM-4 プロトタイプの開発過程について述べる。

一般に並列処理計算機を開発するにあたり、次に示す点が重要である。

- (1) 部分もしくは全体の基本的な動作試験が、容易にかつ効果的に行えること。
- (2) 誤動作が確認された時、その要因の究明が効率的に進められること。

これら2つの事項は互いに密接に関連している。すなわち動作試験中に誤動作が確認された時には直ちにその要因を究明しなければならないし、またその要因を究明するためには新たな基本動作試験が必要になってくる。そこで、並列処理計算機の開発を効率よく進めるためには、これら2つの事項が満足される様、①ハードウェア構成、②開発環境の両面からの支援が不可欠である。

EM-4 プロトタイプでは、①ハードウェア構成の面ではメンテナンスアーキテクチャ⁽¹⁾を導入し、②開発環境の面では各基板の調整専用のテストベッドシステムを開発することにより、上記の事項を実現している。

〔メンテナンスアーキテクチャ〕

EM-4 ではメンテナンスおよびデバッグ処理の効率化、動作状況のモニタリングなどを実現するために、プログラム実行を担当するプログラムアーキテクチャとは別にメンテナンスアーキテクチャを採用している。メンテナンスアーキテクチャは、①プログラム実行時の動作状況を実時間でモニタリングする、②プログラムアーキテクチャ

の異常時やシステムクロックの停止時にも各PEの内部状況を読み出す、ことを可能とするため、システムクロックとは別のクロックで独立に動作するよう設計されている。さらにEM-4のメンテナンスアーキテクチャは、4段に階層化された構造を持っており、各PEにおけるメンテナンス処理の並列実行を可能にしている。

〔テストベッドシステム〕

EM-4 テストベッドは、要素プロセッサ単位もしくはグループ単位の調整・デバッグを効率よく行うことを目的に開発されたもので、

- ①プロセッサ基板1枚または2枚
- ②パケットインタフェーススイッチ基板1枚
- ③結合網を構成するマザーボード1枚
- ④パケットインタフェースプロセッサとコントロールプロセッサを兼務する計算機1台

から構成されている。プロセッサ基板が1枚の時のテストベッドのブロック図を図2に示す。テストベッドを用いることにより、以下の諸点が実現される。

- (1) 要素プロセッサ単位、グループ単位の調整・デバッグを、結合網から切り離して容易に行うことができる。

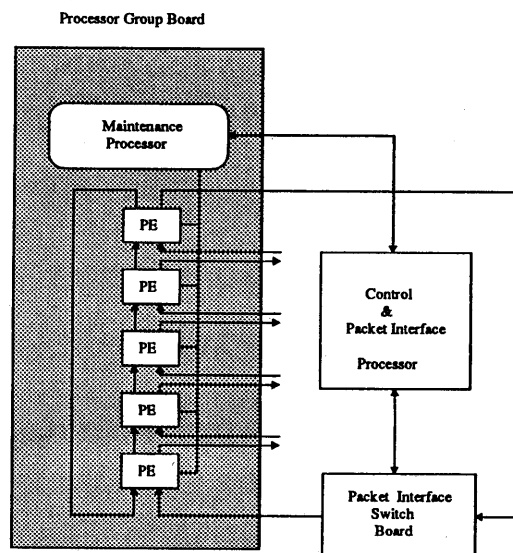


図2 テストベッドシステムのブロック図

(2) パケットインタフェーススイッチ基板の調整をここで行うことができる。

(3) プロセッサ基板2枚からなる構成とすることでグループ間の簡単な通信テストが可能であり、通信の基本的な調整をここで行って、全体システムでの結合網調整の負担を軽減できる。

3.2 基本動作の試験法

前述のとおり、開発中のシステムが誤動作を起こしている時、誤動作の要因の特定が容易に行えるような動作試験法を実現する必要がある。ところが、高並列計算機において誤動作の要因を特定することは、システムの複雑さと並列要素間の順序性の検証の点から、実際にはかなり困難である。さらにデータ駆動計算機の場合、

①副作用がなく、処理の痕跡が残らない。

②複数の文脈が命令レベルで混在する。

ため、状況把握がより困難になることが多い。

そこで、このような複雑なシステムの基本動作を試験する時には、以下に示すような段階的な動作試験法が有効になる。

- (1) 最も簡単に正常かつ安定に動作していることが確認できる部分を、動作試験の核とする。
- (2) 核の部分を用いて、その周辺部の動作を試験する。
- (3) 正常かつ安定に動作していることが確認できたら、それを用いてさらにその周辺部の未確認部分を試験する。
- (4) 以上の繰り返しにより、確認部分の範囲を少しずつ広げていく。

このように段階的に試験を行うことで少しずつ確定要素を拡大していく方法では、絞り込んだ範囲で誤動作の要因が追求されるため、効率よく要因を究明することができる。

ところで、このような段階的な動作試験を行うには、動作試験の核となる部分の安定動作が容易に確認できる必要がある。したがって、動作試験の核となる部分は、実装上できる限り単純な構造をとることが望ましい。EM-4プロトタイプでは以上の見地より動作試験の核として、

①メンテナンス回路

②システムクロック制御回路

の2つを持っている。メンテナンス回路は前述のメンテナンスアーキテクチャを実装したものであり、システムクロックとは別クロックで独立に動作して各要素プロセッサのモニタリングを行う。本回路は低機能ではあるが、単純な構造を有している。また、システムクロック制御回路はシステム全体に1組装備されていて、クロックを停止したり任意の数だけのクロックを供給したりする。本回路も簡単な構造の回路である。この2つの回路を組み合わせることで、段階的な基本動作試験を行った。

3.3 ハードウェア開発

EM-4プロトタイプのハードウェア開発を進めるための調整手順について以下に述べる。まず、インタフェーススイッチ基板は、クロック供給およびパケットインタフェース機能を持ち、プロセッサ基板より先に単独で調整される必要がある。インタフェーススイッチ基板を調整する上で確認しなければならない事項は、

①システムクロックが正常に供給されること。

②パケットの入出力が正常に行えること。

の2点である。①のクロックはオシロスコープなどで実測することにより確認した。また②のパケット入出力は、インタフェーススイッチの入力部と出力部をケーブルで短絡することにより、単独で試験できる様に設計されている。

プロセッサ基板の調整は以下に示す2段階に分けて行った。

(1) システムクロックを停止して、メンテナンス回路よりテスト、調整を行う。

(2) システムクロックを供給して、パケット処理に関するテスト、調整を行う。

このうち(1)についてはプロセッサ基板とホスト計算機のみで調整できるが、(2)についてはインタフェーススイッチ基板とパケットインタフェースプロセッサが必要である。(1)(2)の調整とも、要素プロセッサ単位から始め、調整が進むにした

がってグループ単位、システム全体へと範囲を広げていくのが効果的である(図3)。要素プロセッサ単位、グループ単位の調整にはテストベッドの利用が有効であった。

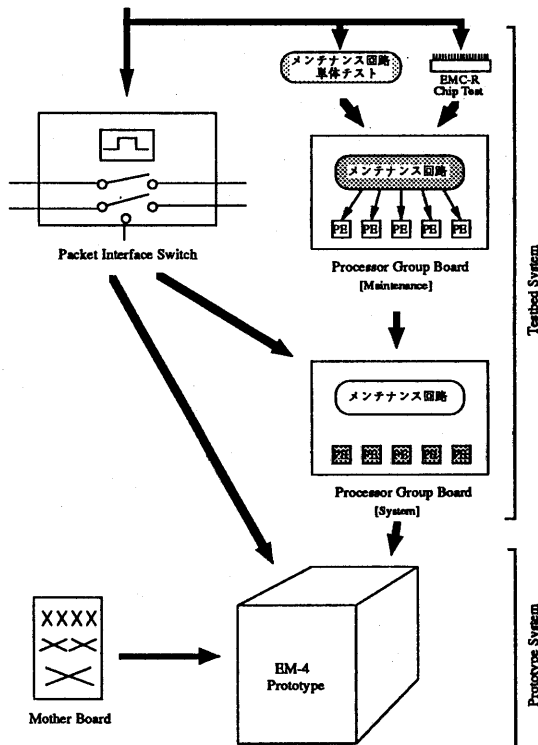


図3 ハードウェアの調整手順

3.4 基礎ソフトウェアの開発

ハードウェアの開発とともに、それを制御するための低位のソフトウェアが必要になる。EM-4では次の3種類の低位基礎ソフトウェアを開発した。

(1) パケットインタフェース制御プログラム

パケットインタフェースプロセッサ上においてパケットの入力および出力を制御する。さらにその上位で、起動パケットの生成や結果パケットの表示などを行う。またインタフェーススイッチ基板の制御も行い、システムクロック制御回路のコントロールを行う。

(2) メンテナンス制御プログラム

ホスト計算機上において、各グループのメンテナンス回路との通信を行い、制御を行う。

(3) メンテナンスモニタプログラム

各グループのメンテナンス回路上に常駐し、ホスト計算機からのコマンドを受けて命令を実行する。具体的にはグループ内の各要素プロセッサに対し、メンテナンスレジスタの読み出しまたは書き込み命令を発行する。読み出したデータはホスト計算機に送出する。

これら3つのプログラムが、EM-4の制御ソフトウェアの核となる部分である。さらにこれらの上位に位置するソフトウェアとして、

(4) EM-4 アセンブラ、ローダ

(5) 動作試験を行うための各種テストプログラムなどを開発した。また、その他の基礎ソフトウェアとして、

(6) パケットインタフェースプロセッサとホスト計算機の通信を司る通信制御プログラムを開発した。

これら基礎ソフトウェアの構成を図4に示す。

4. EM-4 プロトタイプの動作環境

本節では、EM-4プロトタイプの現在の動作環境について述べる。

4.1 試験プログラム開発環境

EM-4プロトタイプの試験プログラム開発環境としては、以下に示す3つを提供している。

(1) 言語処理系

EM-4プロトタイプの言語処理系としては現在のところまだアセンブラしか開発されていない。EM-4のアセンブリ言語はデータフローグラフを忠実に表現できるだけでなくレジスタの指定など詳細な動作を記述できる。

高級言語としては、現在C言語準拠の高級言語DFCIIの処理系を開発中である。具体的にはDFCIIのコンパイラが生成する中間言語SASGAをEM-4アセンブリ言語に変換するトランスレータを作成している。DFCIIは科学技術計算用データ駆動計算機SIGMA-1用に開発さ

れた高級言語である^[12]。ここでは特殊なブロック構造を定義でき、その構造が中間言語SASGAに反映されるため、強連結ブロックの指定、およびその最適化が陽に行える。また先述したトランスレータによって、強連結ブロックを自動生成することができる。

さらにDFCIIとは別に、EM-4上に実現する新言語を検討中である。

(2) ソフトウェアシミュレータ

EM-4ではプログラム開発環境の1つとして、ソフトウェアシミュレータを開発している。これはシングルチッププロセッサEMC-Rの機能をレジスタトランスファレベルでシミュレートするものであり、このシミュレータを用いることによって、EM-4のプログラムの初期的なテストおよびデバッグを行うことができる。

(3) 実機上におけるステッパ・トレーサ

EM-4では、メンテナンス回路とシステムクロック制御回路を組み合わせて用いることによ

り、実機上でプログラムのデバッグ処理を行うことができる。例えば断続的にシステムクロックを投入することによりステップ動作を行い、各時間におけるそれぞれのPEの内部状態をメンテナンス回路から読み出すことによって、誤り個所を見つけ出すことが可能である。

またEM-4は、PE間で転送される任意のバケットの内容をホスト計算機に取り込むバケットトレース機能を有している。このトレース機能を用いることで、最初に誤りを起こしたPEを特定できることが多い。

実際にはステッパ、トレーサの両者を併用して、次に示す手順でデバッグ処理を行うのが効率的である。

- ①プログラム実行時に生成されるバケットをトレースする。
- ②バケットの内容を順に解析し、誤りを起こしたPEを見つける。
- ③再度プログラムを実行、ステップ動作を行う。
- ④前記のPEについて各時間における内部状態をメンテナンス回路より読み出し、動作状況を追跡する。
- ⑤誤り個所を見つける。

4.2 故障診断法

EM-4プロトタイプに誤動作が認められたとき、その要因として考えられるものに、

- ①ハードウェアの論理的欠陥
- ②ハードウェアの物理的欠陥
- ③低位ソフトウェアの誤り
- ④高位ソフトウェアの誤り

がある。このうち④の高位ソフトウェアの誤りは前述のソフトウェアシミュレータを用いることにより、単独で試験することができるので比較的発見が容易であるが、①～③の要因については互いに関連し合っているので、症状を特定することが困難である。したがってEM-4プロトタイプの故障診断法としては、開発過程で行ったような段階的な基本動作試験を適用して解析していくことになる。

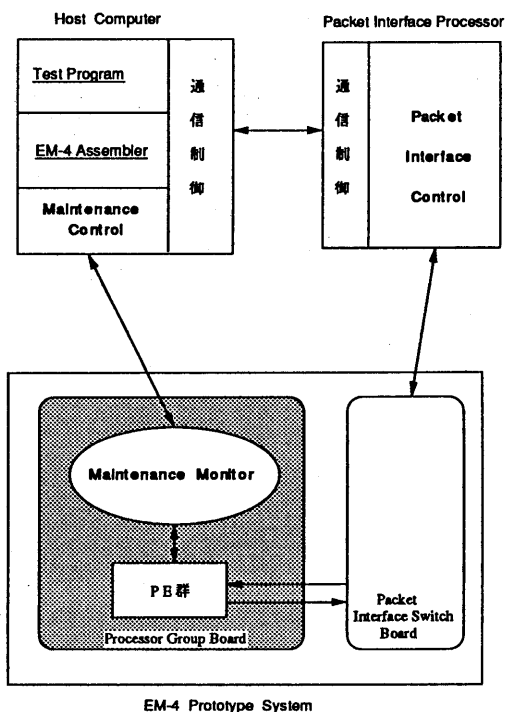


図4 EM-4の低位ソフトウェア

ここでは例として、プロセッサ基板上の結合網の一部に欠陥を生じた場合について考える。プロセッサ基板上の結合網が断線している時、ここを通るパケットはカラーが変えられてしまったり誤ったノードに送られたりするために、プロセッサ間の通信が失敗する。しかしこのことに関してユーザが確認できる症状としては、パケットが返ってこない、あるいは誤ったパケットが返ってくるということだけなので、この誤動作が故障によるものかどうかさえもユーザ側で認識することは難しい。そこで次に示す手順にしたがって動作試験を行う。

(1) メンテナンス回路の動作確認

メンテナンス回路は段階的な動作試験の核となる部分なので、最初に正常動作を確認しなければならない。実際には各メンテナンス回路上に常駐するメンテナンスモニタプログラム内に、簡単な自己診断ルーチンを設けており、ホスト計算機との間の簡単な通信だけで確認できるようになっている。

また、同様にインタフェーススイッチ基板上のシステムクロック制御回路も動作確認しなければならない。

(2) P E の動作解析

システムクロックを制御することによりステップ動作を行い、各要素プロセッサの動作を解析する。具体的には、プログラムが正しくロードされているか、命令が正しく実行されているか、送受信パケットの内容などを解析する。特に特定の基板上のP Eの動作に不審が感じられるときは、その基板単体をテストベッド上で試験する。

(3) 基板上の結合網の試験

テストベッドでの試験において、P Eが受信するパケットの内容が不審な時には、基板上の結合網が正しいかをチェックする。具体的にはいくつかのパターンのパケットを流し、受信パケットの内容を調べる。このようにしてパケットが正しく伝達されないことにより、結合網上の断線を発見することができる。

4. 3 性能測定法

E M-4 プロトタイプでは、メンテナンス回路とシステムクロック制御回路を組み合わせることで、基本的な性能測定を行うことができる。一例として実行性能の測定手順を以下に示す。

- (1) 所望の実行プログラムを各P Eにダウンロードし、実行可能状態にする。
- (2) メンテナンス回路より、各要素プロセッサチップ内に設けられている命令数カウンタをクリアする。
- (3) パケットインタフェースプロセッサより起動パケットを投入し、システムクロック制御回路を制御して、クロックを計数しながらシステムに供給する。
- (4) パケットインタフェースプロセッサがE M-4からの終了パケット（結果パケット）を受け取ると同時にクロックの供給を停止する。
- (5) 各P Eの命令数カウンタには、実際にそのP Eが実行した命令の数が保持されている。したがってこれら命令数カウンタの値をメンテナンス回路より読み出して集計することにより、システムに供給したクロック数との関係から実行性能が算出できる。

この方式でいくつかのサンプルプログラムによりE M-4 プロトタイプの実行性能を測定した。たとえば1からnまでの対数の和を計算するプログラムでは、実測999MIPSを達成している。

またE M-4 プロトタイプでは、プログラム実行時の任意の時間にシステムクロックの供給を止め、メンテナンス回路より各P Eの内部状態を読み出すことで、各々のP Eについて

- ①動作中（b u s y）か否か
- ②負荷量
- ③カラー獲得数
- ④パケットの転送量

などの動作状況を観察することができる。これらの測定結果により、これまでに負荷分散方式や関数分散方式についての検討を行っている。

5. まとめ

EM-4プロトタイプの開発過程,ならびに現在の動作環境について述べた.開発においてはメンテナンスアーキテクチャの採用,テストベッドの開発などにより調整・デバッグ作業を効率化した.また段階的な基本動作試験法が,ハードウェア・ソフトウェアの両面において,調整・デバッグ作業に有効に作用することを確認した.さらにソフトウェアシミュレータ,メンテナンス回路,クロック制御回路などの組み合わせにより,充実した試験プログラム開発環境を開発した.

EM-4プロトタイプは1990年4月より稼働中であり,最大性能1GIPS,最大網転送性能14.63GB/Sを達成している.これまでに,いくつかのサンプルプログラムを用いて強連結枝モデルの有効性の検証,負荷分散問題の検討,関数分散方式についての検討などを行ってきた.今後は高級言語処理系を開発し,プログラム開発環境をさらに整備して,より実用的な応用プログラムによる評価を行っていく.また,強連結ブロックの自動抽出法,総合的なスケジューリング方式を検討し,新言語を開発するとともに,1000規模システムの開発を進めていく.本稿で述べた開発・デバッグ方式が,この超並列計算機において,より高機能化・高効率化される予定である.

謝辞

本研究を遂行するにあたり,御指導・御教授いただいた電子技術総合研究所の棟上情報アーキテクチャ部長,弓場知能システム部長,島田計算機方式研究室長,ならびに計算機方式研究室員の皆様に感謝いたします.

参 考 文 献

[1] Dennis, J.B. and Misnas, D.P.: A Preliminary Architecture for A Basic Dataflow Processor, Proc. 2nd Ann. Int. Symp. Computer Architecture, IEEE, pp. 126-132 (1975).

[2] Arvind, Dertouzos, M.L. and Iannucci, R.A.: A Multiprocessor Emulation Facility, MIT-LCS Technical Report 302 (1983).

[3] Gurd, J.R., Kirkham, C.C. and Watson, I.: The Manchester Prototype Dataflow Computer, CACM, Vol. 28, No. 1, pp. 34-52 (1985).

[4] 坂井, 山口, 児玉, 平木: データ駆動計算機の究極的高速化に関する検討, 並列処理シンポジウム JSPP'89, pp. 71-78 (1989).

[5] 坂井, 平木, 山口, 児玉, 弓場: データ駆動計算機のアーキテクチャ最適化に関する考察, 情報処理学会論文誌, pp. 1562-1572 (1989).

[6] Sakai, S., Yamaguchi, Y., Hiraki, K., Kodama, Y., and Yuba, T.: An Architecture of a Dataflow Single Chip Processor, pp. 46-53, Proc. of ISCA 89 (1989).

[7] Yamaguchi, Y., Sakai, S., Hiraki, K., Kodama, Y., and Yuba, T.: An Architectural Design of a Highly Parallel Dataflow Machine, Proc. of IFIP 89 (1989).

[8] 坂井, 児玉, 山口: プロセッサ結合型オメガ網を用いた並列計算機の構成, 信学技報, CPSY89-31 (1989).

[9] 坂井, 児玉, 山口: 高並列データ駆動計算機EM-4における負荷分散方式, 信学技報, CPSY90-47 (1990).

[10] Kodama, Y., Sakai, S., and Yamaguchi, Y.: A Prototype of a Highly Parallel Dataflow Machine EM-4 and its Preliminary Evaluation, Proc. of InfoJapan'90 (1990).

[11] 児玉, 坂井, 山口: データ駆動型シングルチッププロセッサにおけるメンテナンスアーキテクチャ, 第40回情処大全 (1989).

[12] 関口, 島田, 平木: 同期構造を埋め込んだSIGMA-1用高級言語DFCII, 情報処理学会論文誌, Vol. 30, No. 12, pp. 1639-1645 (1989).