

データ駆動計算機 EDDEN のアーキテクチャ†

三浦 宏喜^{††} 川口 正樹^{††} 田中 一行^{††}
大橋 秀紀^{††} 清水 雅久^{††}

我々は、実用的な並列処理計算機の実現に向け、要素プロセッサ数 1,000 台規模の高並列データ駆動計算機 EDDEN (Enhanced Data Driven ENgine) を開発中である。本研究では、データ駆動方式の演算処理機能とプロセッサ結合網の要素としての通信制御機能を兼ね備えた要素プロセッサを 1 チップの LSI 上に実現し、これを多数台結合した大規模データ駆動計算機の稼働を目指している。本論文では、まず EDDEN におけるプロセッサ結合網である自己ルーティング方式のトラス結合網、次に EDDEN の要素プロセッサ LSI の概要を述べた後、各プロセッサにおける通信制御、セルフルーティング方式の詳細、実行制御方式の詳細を述べ、EDDEN の特長を明らかにする。要素プロセッサ LSI は、当社の CMOS 1.2 μm ルールのプロセスを用い、循環パイプライン式の演算ユニット、トラス結合網に基づく通信制御ユニットなどを 1 チップに集積している。回路規模は約 32 万トランジスタである。また、浮動小数点 ALU、乗算器を内蔵しており、単一プロセッサの最大性能は 10 MFLOPS である。実行制御方式としては、オペランド待ち合わせ方式の改良による待ち合わせメモリの小容量化、循環パイプライン上の演算と並列的に実行可能なベクトル演算機構の導入などの特長をもつ。

1. はじめに

データ駆動アーキテクチャは、①問題に内在する並列性を自然に抽出することができる、②各命令の実行が関数的となるため、各プロセッサ内部における命令実行のパイプライン化が容易である、③プロセッサ間の通信と同期のメカニズムを含んだアーキテクチャである、などの点で大規模並列処理に適したアーキテクチャであると考えられる^{1),2)}。我々は既に、ストリーム処理を導入し画像処理を指向したデータ駆動計算機 SPM、および C 言語をベースにした言語処理系ソフトウェアを開発し^{4),5)}、それらの性能評価^{6),7)}を通じてデータ駆動方式の有効性を確認した。しかし、一方でいくつかの問題点も明らかになった。それは、(a)配列等の定型的な構造体に対して単純な計算を繰り返す定型的演算の際の性能低下、(b)オペランドデータ対の供給能力の限界による演算パイプライン充足率の低下、(c)特にオペランド待ち合わせのための発火制御機構のハードウェア規模が大きくなるため要素プロセッサの 1 チップ LSI 化が困難であることなどである⁸⁾。また、高並列計算機の実現のためには、主としてプロセッサアーキテクチャに起因するこれらの問題点の解決に加え、優れたプロセッサ結合網の実現が必

須である。これらの課題を克服すべく我々は、データ駆動計算機 EDDEN (Enhanced Data Driven ENgine) の開発を行っている。EDDEN では、各要素プロセッサの演算部にベクトル演算機構を導入することによって上記 (a)、(b) の解決を図り¹¹⁾、発火制御機構に大幅な改良を加えることにより (c) の解決を図っている⁹⁾。さらにプロセッサ結合網として、自己ルーティング方式のトラス結合網を採用し、高い通信能力の実現を目指している¹⁰⁾。本論文では、まずプロセッサ結合網の概要と 1 チップの CMOS-LSI に実現する要素プロセッサの概要について述べる。次に通信制御方式の詳細を述べ、さらに実行制御方式として特に発火制御機構の改良、ベクトル演算機構の導入などを中心にその有効性について論じる。

2. プロセッサ結合網

2.1 プロセッサ結合網の条件

プロセッサ結合網の条件として以下を考慮した。

① 通信ポート数 4 以下

EDDEN では、通信機構と演算機構を 1 チップの要素プロセッサ LSI に実現するとともに、各要素プロセッサ (PE) チップにデータメモリを外付けして分散メモリ方式をとる。LSI の総ピン数とデータメモリの結合に要するピン数を考慮し、かつプロセッサ間の通信性能を高くするためには、通信ポート数を 4 以下とすることが適当であると考えられる。

② プロセッサ間距離が小さい

プロセッサ間距離の議論においては、プロセッサ数

† Architecture of a Data Driven Computer "EDDEN" by HIROKI MIURA, MASAKI KAWAGUCHI, KAZUYUKI TANAKA, HIDEKI OHASHI and MASAHISA SHIMIZU (Computer Department, Information & Communication Systems Research Center, SANYO Electric Co., Ltd.).

†† 三洋電機(株)情報通信システム研究所コンピュータ研究部

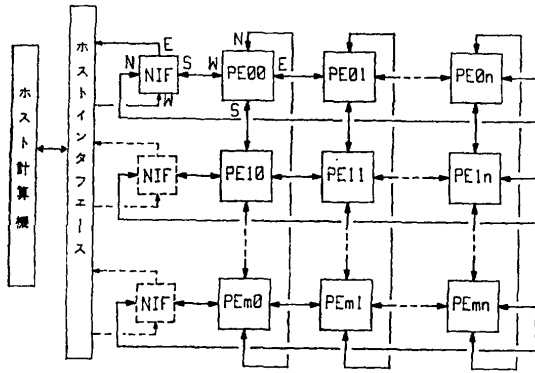


図 1 EDDEN の基本構成
Fig. 1 Basic configuration of EDDEN.

に対するオーダが使われる場合が多いが、結合網に結合できるプロセッサ数を考慮した実際の値をも重視する必要がある。

③ 自己ルーティング可能

大規模な並列処理システムにおいては、PE 間で通信される膨大なパケットを集中的に管理するのは極めて困難である。したがって、通信されるパケットが保持する行き先と現在の PE の位置を用いてパケットを自動的にルーティングできなければならない。

④ デッドロック回避が可能

通信効率が多少落ちることがあっても、結合網がデッドロックする事態は回避できなければならない。

⑤ 一様構造

画像処理、科学技術計算などのいわゆるデータパラレル型の処理に適応するためには、結合網上の位置によって通信効率に差が生じない方がよい。また、各 PE に要求される通信制御機能が同一である方が LSI 化に有利である。

2.2 プロセッサ結合網の概要

上述の条件に加え、PE 間あるいはボード間の配線などの実装面での容易性をも考慮した結果、これらすべての条件を満たす結合網として図 1 に示すトラス結合網を採用した。すなわち、各 PE に東 (E)、西 (W)、南 (S)、北 (N) の 4 系統の通信ポートを設け、PE を $m \times n$ 台行列配置して各行方向、列方向を循環的に結合する。m, n は 2 のべき乗に限り、それぞれ 1~32 の範囲で可変にできる。結合網の一様性を保ち、PE 間の平均通信距離を小さくするために PE 間の通信リンクはすべて双方向とした。さらに、PE チップのピン数削減のために

同一の通信リンクを時分割的に双方向で共用する方式とした。また、各 PE チップに内蔵する通信制御機構にパケットを行き先へ向けて最短距離で到達させるための交換機能をもたせることにより、自己ルーティング結合網を実現した。また、網の外部とのデータのやりとりのためにネットワークインタフェース (NIF) を通信リンクに挿入することとし、初期設定に応じて PE チップがこの NIF としても動作できるようにした。

3. 要素プロセッサ (PE) の概要

EDDEN では、色つきトークン方式の動的データ駆動アーキテクチャを採用し、関数共用によるプログラムメモリの有効利用と、関数の多重実行によるパイプライン充足率の向上を目指している。EDDEN の PE は、データ駆動方式の 32 ビットマイクロプロセッサであり基本的には、図 2 に示すように、通信制御部 (NC)、循環パイプライン構造のデータ駆動プロセッサ部分、および外部データメモリ (EDM) から構成される。

NC は東西南北 4 系統の通信ポートを保持してトラス結合網に基づくパケットのルーティング制御を行う。入力バッファ (IBUF)、出力バッファ (OBUF) はそれぞれ NC から循環パイプラインへの入力、循環パイプラインから NC への出力の際の 1 パケット分のバッファである。キューメモリ (Q) は循環パイプライン上のデータ流量の変動を吸収する FIFO メモリであり、32 パケット分の容量をもつ。プログラム記憶 (PS) はパケットの制御情報の付け替え、パケッ

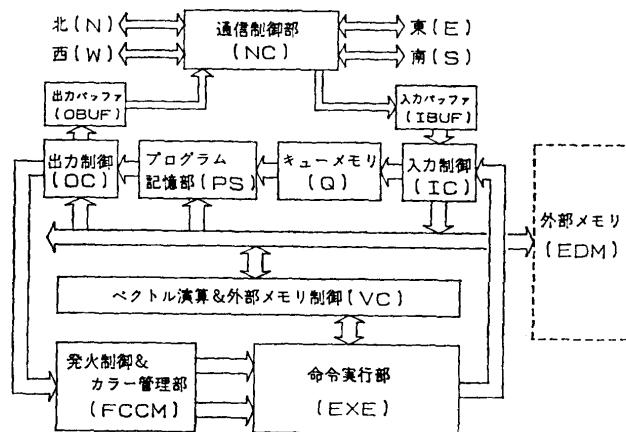


図 2 要素プロセッサの構成
Fig. 2 Configuration of a processor element.

トのコピー、定数の付与を行う。発火制御・カラー管理部 (FCCM) はオペランドの待ち合わせ、および関数共用時のカラーの獲得/返却の管理を行う。命令実行部 (EXE) には 32 ビット浮動小数点 ALU および乗算器を内蔵しており、命令の種類としては論理演算、整数/浮動小数点演算、バレルシフト、ベクトル演算、ベクトル通信、条件判定・分岐、セマフォのセトリセット、および関数共用関連の命令などがある。

EDM には、1 語 32 ビット構成で最大 1M 語のデータを格納できる。EDM には演算、通信の対象となるスカラデータやベクトルデータのほかにプログラムをも格納する。このように、プログラム領域を大容量の外部メモリ上に置くことによって、大規模なプログラムにも対応できる。さらに、Q がオーバフローした時には、EDM 上に一時的に外部キューを形成し、いわゆる並列度の爆発を防ぐことができる。

ベクトル演算 & 外部メモリ制御部 (VC) は、ベクトル演算命令、ベクトル通信命令の制御や、外部からの EDM に対するメモリアクセス要求の調停を行う。なお、ベクトル通信の際には、図 2 に示したように入力制御部 (IC) および出力制御部 (OC) を介して、EDM と NC の間でベクトルデータの各要素データを連続的に直接送受するようにし、循環パイプラインの動作に悪影響を及ぼさないようにしている。

また、EXE においては、循環パイプラインによる通常の 2 項演算や単項演算とベクトル演算とが同一の演算パイプラインを時分割で共用することによりパイプライン充足率を高めることができる。

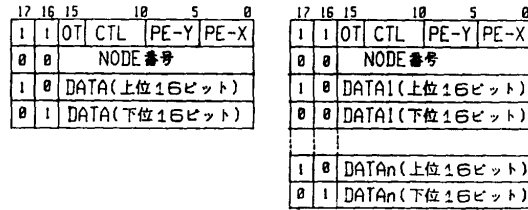
各部には 10 MHz の 2 相クロックを供給し、このクロックに従い通信制御部は自己同期式で、その他の部分は同期式で動作する。循環パイプライン段数は 11 段であり、パケットが 1 周するのに 11 クロックを要する。また、EDM は 20 MHz のアクセスレートで動作させる。

EDM を除く部分は、当社の CMOS 1.2 μm プロセスを用い、1 チップの LSI 上に実現する。その回路規模は、約 32 万トランジスタである。

4. 通信制御方式

4.1 通信パケットの構成

EDDEN におけるパケット構成は、循環パイプライン上ではパイプライン処理効率を上げるために 1 語構成としている。一方、プロセッサ結合網における



(a) スカラパケット (b) 構造体パケット
(a) Scaler data packet. (b) Structured data packet.

図 3 通信パケットの構成

Fig. 3 Formats of communication data packets.

通信パケットは PE チップのピン数削減を図るために図 2 に示すように、通常のスカラパケットで 4 語構成とし、ベクトルデータなどを保持する構造体パケットでは可変長構成を許している。

図 3 の PE-X, PE-Y はそれぞれ行き先 PE の列番号、行番号である。CTL はカラー、あるいはロード/ダンプのコードなどの制御情報を保持する。OT は、トラス結合網の外部行きのパケットを識別するフラグである。さらに、上位 2 ビットのフラグによりパケットのヘッダ、テイル、およびそれ以外の語を識別できる。また最上位のフラグは 1 語ごとに値が反転することにより、語の存在を示す転送制御ビットの役割をも果たす。

4.2 基本的なデータ転送方式

プロセッサ結合網上のデータ転送路はカスケード接続された複数のラッチによって構成されており、各段では現在保持しているデータが次段に転送されるまで新たなラッチ動作を許可しない。すなわち、2 相式クロックによって 1 語単位で自己同期式のデータ転送を行うエラスティックなデータ転送路となっている。これによって、固定長のスカラパケットのみならず、構造体などの可変長パケットの通信をも可能にしている^{10), 12)}。

4.3 双方向通信制御方式

前述のように、プロセッサ結合網における PE 間の通信はすべて双方向通信を基本としており、同一の通信リンクを共用する方式としている。このような双方向通信の優先度制御は、次のように行う。

- ① 双方 2 つのパケットが時差をもって衝突した時は、先に到着したパケットの転送を優先する。
- ② 2 つのパケットが通信リンクに全く同時に到着して衝突した場合は、固定的に片方の転送を優先する。
- ③ 時差衝突、同時衝突によって待たされたパケット

は次のサイクルでは優先して転送する。

④ 上記①～③によって1度優先権が与えられても、優先権が与えられたパケットが次段に進めない時（転送路の停止時）は他方のパケットに優先権を与える。

以上のような方式により、同一通信リンクを共用しながらも、結合網の一様性を保つことができる。

4.4 自己ルーティング方式

(1) ルーティングの概要

EDDEN のプロセッサ結合網は、図1に示したように $m \times n$ 台の PE と複数の NIF (NIF モードに設定された PE チップ) から構成される。すなわち、PE チップの通信制御機構は PE モードまたは NIF モードで動作する。基本的には、PE モードでは自身の番号と等しい番号をもつパケットのみを循環パイプラインに入力しそれ以外のパケットは、パケットが行き先に最短距離で到達するように東西南北いずれかの通信ポートに出力する。NIF モードでは結合網外部からのパケットを網に取り込み、網の外部行きのパケットを選択的に網外部へ出力する。

ただし、各 PE チップにリモートでプロセッサ番号を設定するために、電源を投入してハードリセットを行った状態 (COLD ステート) では、通信制御機構に到着するパケットをすべて循環パイプラインに入力する。すなわち、COLD ステートにおいてプロセッサ番号レジスタへのロードパケットによってプロセッサ番号を設定し、設定の終了と同時に通信制御機構は NORMAL モードに移行し、以後は PE モードまたは NIF モードに応じて通常のルーティング動作を行う。

また、トラス結合網の外部行きのパケットを NIF に向けてルーティングするために、OT=1 のパケットには、NIF の直近の PE の番号をもたせる。各 PE においては、パケットの番号と自身の番号が等しくてもパケットの OT=1 である時は、循環パイプラインに入力せずにパケットを東(E)ポートに出力する。これによって、網外部行きのパケットのために循環パイプラインの処理効率の低下するのを防ぐ。

(2) 通信制御機構

図4に通信制御部 (NC) の構成を示す。図において R1~R5 は分岐回路 (ルータ) であり、後述のルー

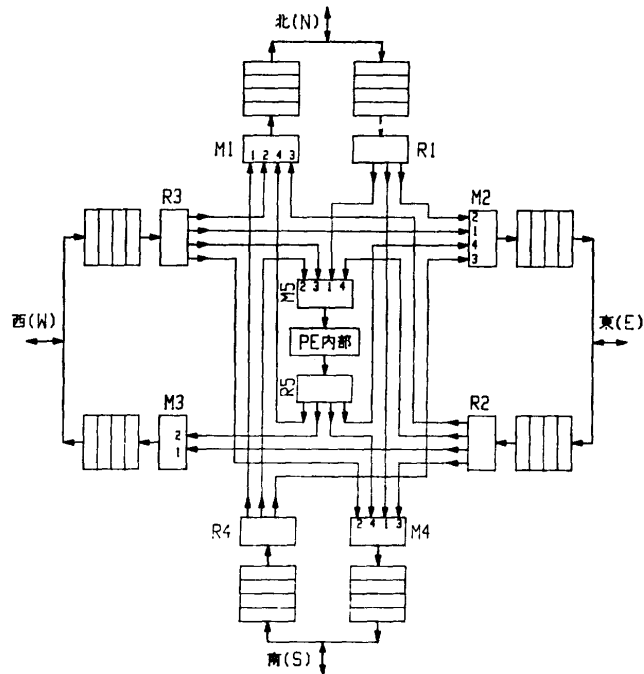


図4 通信制御部 (NC) の構成

Fig. 4 Configuration of network control unit (NC).

ティングアルゴリズムに従ってパケットをルーティングする。M1~M5 は優先度付きマルチプレクサであり、図の数字で示される優先度 (1が最優先) でパケットを合流させる。東西南北各ポートが保持する入力バッファおよび出力バッファは、前述のように1語単位でエラスティックになっている。隣接プロセッサとの通信、および分岐・合流の方向制御はパケット単位で行い、ヘッダの到着で方向を選択し、テイルの転送を終了するまで同一のパケットを選択し続ける。

東西南北の通信ポート、および循環パイプラインへの入力ポート、出力ポートにおける通信性能は各々理想的には 20 MB/秒である。

(3) ルーティング・アルゴリズム

図4の R1~R5 は、以下に述べるアルゴリズムに従ってパケットを所定方向に分岐させる。

図1に示したような $m \times n$ (m 行 n 列) の結合網において、 y 行 x 列にある PE に (y, x) と番号付けする。各プロセッサにおいて、自身の PE 番号が (q, p) 、到着したパケットの行き先が (y, x) である時、

$$\Delta x = (x - p) \bmod n \quad (|\Delta x| \leq n/2)$$

$$\Delta y = (y - q) \bmod m \quad (|\Delta y| \leq m/2)$$

を算出する。

R1~R5の分岐制御アルゴリズムは、

① COLD ステートの時

W, E, N, S のいずれから来たパケットも PE 内部(P)へ出力.

② NORMAL ステートかつ NIF モードの時

[R 1]: OT=0 ならば S へ, OT=1 ならば E へ出力.

[R 2]: OT=0 ならば N へ, OT=1 ならば E へ出力.

[R 3]: OT=0 ならば S へ, OT=1 ならば E へ出力.

[R 4]: OT=0 ならば N へ, OT=1 ならば E へ出力.

③ NORMAL ステートかつ PE モードの時

[R 1]: $\Delta y=0$ かつ OT=0 ならば P へ,
 $\Delta y=0$ かつ OT=1 ならば E へ,
それ以外は S へ出力.

[R 2]: $\Delta x=0$ かつ $\Delta y>0$ ならば S へ,
 $\Delta x=0$ かつ $\Delta y<0$ ならば N へ,
 $\Delta x=0$ かつ $\Delta y=0$ かつ OT=0 ならば P へ,
それ以外は W へ出力.

[R 3]: $\Delta x=0$ かつ $\Delta y>0$ ならば S へ,
 $\Delta x=0$ かつ $\Delta y<0$ ならば N へ,
 $\Delta x=0$ かつ $\Delta y=0$ かつ OT=0 ならば P へ,
それ以外は E へ出力.

[R 4]: $\Delta y=0$ かつ OT=0 ならば P へ,
 $\Delta y=0$ かつ OT=1 ならば E へ,
それ以外は N へ出力.

[R 5]: $\Delta x=0$ かつ $\Delta y>0$ ならば S へ,
 $\Delta x=0$ かつ $\Delta y<0$ ならば N へ,
 $\Delta x<0$ ならば W へ,
それ以外は E へ出力する.

以上のアルゴリズムから明らかなように, NORMAL ステートにおいて, パケットは列番号が一致するまで東西方向のリング上を転送され, 列番号が一致した PE において南北方向のリングに合流して, 行き先に向けて最短距離で転送される.

4.5 デッドロック回避方式

EDDEN で採用したようなトーラス結合網におけるデッドロックとしては, ①網上で複数のパケットがループを形成し, このループに新たなパケットを合流させようとした時に生じるデッドロック, ②双方向通信リンク上を転送されているパケットが, その転送の

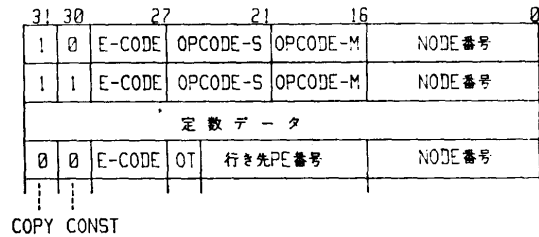


図5 プログラム記憶形式

Fig. 5 Configuration of instruction memory.

終了を待って停止している反対方向のパケットが原因で, 転送の途中で停止してしまった時に生じるデッドロックが考えられる.

①については, 東西南北の各通信ポート間にそれぞれ2系統の通信路を設けて結合網をループフリーにすれば防ぐことができる. しかし, トーラス網の場合, この方法ではハードウェア規模が大きくなりすぎる.

そこで, EDDEN のルーティングアルゴリズムでは PE に南北方向から入力されたパケットが東西方向に分岐してルーティングされるケースをなくし, パケットによるループを1行または1列すべての PE を巡るもののみに限定している. さらに, 図5の M1~M4 において, 東西方向または南北方向に通過するパケットを, 他の方向から合流しようとするパケットよりも優先して転送することによって①を防止している.

また, ②のデッドロックを防ぐためには, 当然ながらパケットの途中の語の転送中に転送が停止しないことを保証すればよい. そこで, 図5に示したように東西南北の各通信ポートに1パケット(4語)の入力バッファおよび出力バッファを設けて②を防止している.

以上により, 固定長のスカラパケットに関しては, デッドロックを完全に回避している. また, 構造体パケットの通信に起因するデッドロックについては, デッドロック防止のためのセマフォを導入している. すなわち, 各 PE に, 通信制御部の使用を排他制御するセマフォを設けており, 遠く離れた PE 間で構造体通信を行う際には, 事前にセマフォチェック&セット命令により通信経路のセマフォを獲得するものとする.

5. 実行制御方式

5.1 プログラム記憶方式

EDDEN のプログラム記憶形式を図5に示す. プログラム記憶部 (PS) において, パケットのノード番号によりプログラムが参照され, パケットの制御情報

の付け替えが行われる。

図の NODE 番号は付け替えるべき新しいノード番号である、OPCODE-M はメイン命令コード、OPCODE-S はサブ命令コードであり、これらによって命令実行部 (EXE) における命令の種類が規定される。E-CODE は2項演算の右パケット、左パケット、単項演算パケット、多入力同期処理パケット、PE 外部への出力パケット、カラー獲得パケット、およびカラー解放パケットを区別する。PE 外部への出力パケットの場合は、命令コードフィールドには OT フラグと行き先の PE 番号が格納される。

PS は、COPY フラグ、CONST フラグによりパケットのコピー・定数付与を行う。すなわち、COPY=1 である限りプログラムを連続的に読み出してパケットをコピーする。さらに CONST=1 の時は、到着したパケットが保持していたデータを次番地の定数データに置き換えてパケットを出力する。パケットコピー時はQからの読み出しが禁止され、後続のパケットはQ内部で待つ。

PS におけるパケット生成レートは、最大 10 MOPS であるが、定数付与時はその 1/2 になる。

5.2 発火制御、カラー管理方式

(1) 発火制御、カラー管理における問題点

一般に動的データ駆動方式では、パケットのノード番号とカラーをタグとして、2つのパケットのタグの一致を検出し、オペランド対を生成して命令実行部に供給する発火制御機構が必須である。タグ空間と等しいアドレス空間をもつ大容量のオペランド待ち合わせメモリを用いればこの機構は実現できる。しかし、この場合待ち合わせメモリの使用効率が極めて悪いとともに、メモリが大規模になり PE の1チップ LSI 化が不可能となる。そこで我々は、ハッシュビット方式を用いた発火制御機構を用いてきた^{4),6)}。しかしこの場合、ハッシュ衝突したパケットを循環パイプライン上で周回させながら待たせるため、PE の処理効率を低下させるとともに発火制御機構のデッドロックを生じる可能性があった。また、色付きトークン方式を採用した場合には空きカラーの管理を高速に行う特殊なハードウェアが必要であり、PE のハードウェア規模がさらに増大してしまう。

(2) 発火制御・カラー管理機構の概要

これらを解決するために EDDEN では、「ハッシュ & チェイン方式」を採用した。これは、図6に示すように、待ち合わせメモリを、ハッシュアドレスで直接

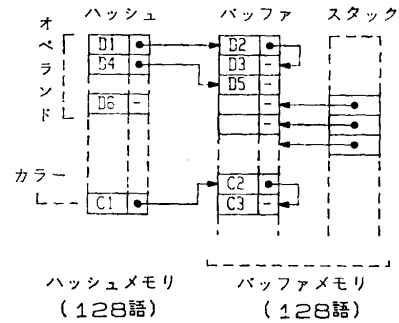


図6 オペランド待ち合わせとカラー管理
Fig. 6 Matching and color management scheme.

アクセスされるハッシュメモリ (HM) と、ハッシュ衝突時に衝突したオペランドをリンクリスト形式でチェイニング (連鎖) して格納するバッファメモリ (BM) の2段階構成としたものである、これによって、ハッシュ衝突時の通過パケットをなくするとともに、小容量の待ち合わせメモリによる効率的な発火制御機構の実現を図っている。さらに、BM の空き番地をスタック方式で高速に管理することにより、チェイニング操作の高速化を図っている。

また、カラー管理に関しては、カラーの獲得 (GET-COLOR)、および解放 (FREE-COLOR) がそれぞれ、オペランドの発火、および待ち合わせに類似していることに着目し、特殊なハードウェアを必要とせずに、カラー管理を発火制御機構を用いて実現する方式とした⁹⁾。

(3) 発火制御・カラー管理機構の処理方式

図7に待ち合わせメモリ (HM, BM) の記憶形式を示す。HM, BM の容量は各 128 語であり、ハッシュアドレスとしては、タグ 20 ビットのうちの7ビットを使用する。図の比較タグは残りの 13 ビットである。また、チェイニングのためのポインタを7ビット保持する。FULL フラグはオペランドの存在を示し、NEXT フラグはチェイニングの有無を示す。

具体的な動作としては、まず初期状態で、

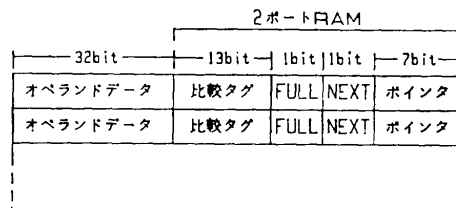


図7 オペランド待ち合わせメモリの記憶形式
Fig. 7 Configuration of matching memory.

- ① HM の FULL フラグをすべてクリアする。
- ② BM 上に、BM の空き番地 (すなわち BM 自身の全番地) をチェイニングし、スタックを形成する。
- ③ さらに、図 6 に示したように、HM の特定番地を先頭にして使用するカラーをすべて BM 上にチェイニングしてカラーリストを形成する。この時、カラーが占有した BM の番地はスタックからポップしておく。

そして、プログラム実行時には、

- ④ 到着したパケットのタグから生成したハッシュアドレスによって HM をアクセスし、そこに相手のオペランドがあればオペランド対を生成し、そこが空いていればオペランドをそこに格納する。
- ⑤ ハッシュ衝突が生じた場合には、スタックをポップして BM の空き番地を獲得し、オペランドを BM 上にチェイニングする。
- ⑥ オペランドがチェイニングされている場合は、リンクリストをたどりながら、相手のオペランドを検索する。この時、リストの先頭、すなわち HM において相手が見つかったら、リストの 2 番目のオペランドを HM に移動し、新たに生じた BM の番地をスタックにプッシュする。また、リストの途中で相手が見つかったら、その番地を空き番地としてスタックにプッシュし、発火したオペランドがリストから削除されるようにポインタ操作を行う。
- ⑦ GET-COLOR パケットが到着すると、カラーリストの先頭、すなわち HM の特定番地にあるカラーをパケットに格納し、カラーリストの 2 番目のカラーを HM に移動させ、新たに生じた BM の空き番地をスタックにプッシュする。
- ⑧ FREE-COLOR パケットが到着すると、スタックをポップして BM の空き番地を獲得し、パケットが保持しているカラーを BM に格納し、カラーリストの 2 番目に挿入する。

(4) 発火制御・カラー管理機構の性能

本方式では、上述のようなスタックの導入に加え、HM および BM のデータフィールド以外を 2 ポート RAM で構成し、アクセスを多重化することにより処理の高速化を図っている。

本方式の発火制御の処理時間 (クロック数: CLK) は、

[A] HM にオペランドが存在しない時: 1 CLK

[B] HM にオペランドが存在する時:

[B-1] チェイニング (連鎖) がない時:

タグ一致 (発火) の時 : 1 CLK

タグ不一致 (未発火) の時: 2 CLK

[B-2] チェイニングがある時:

HM 上で発火の時 : 3 CLK

HM 上で未発火の時: $(n+2)$ CLK

(n はポインタをたどった回数)

である。

したがって本方式の発火制御機構におけるオペレーションパケット (オペランド対) の生成レートは、理想的な場合 (上記の [A] および [B-1]) に最大 5 MOPS である。実効性能は、平均的なリンクリスト長、クロック増加率によって決定できるが、アドレス空間を一定とした場合には、これらはオペランドの平均的滞在時間 (寿命) に依存すると考えられる。また、BM の最大使用量も寿命に依存すると考えられ、本方式の性能はオペランドの寿命によって左右される。これに関しては、数種類の計算例においてオペランドの寿命は極めて小さいという報告³⁾があり、平均寿命の増加に起因するデメリットよりも、メモリの小容量化によるメリットのほうが大きいと考える。

5.3 命令実行方式

(1) 命令実行機構の概要

命令実行部 (EXE) は①特殊命令実行ブロック、② ALU ブロック (ALU, 乗算器, パレルシフトなど)、③条件判定・分岐制御ブロックがカスケードに接続された構成をとる。

①では外部データメモリの直接および間接のリード・ライト (スカラデータアクセス)、簡易定数の発生、関数共用関連の命令 (CHANGE-COLOR, RETURN) などを実行する。簡易定数の発生は、パケットの保持するサブ命令コードにより、 $-8 \sim 7$ までの定数を発生して右オペランドとするものである。

②では、パイプライン型の 32 ビット整数/浮動小数点 ALU および乗算器が並列に接続されており、メイン命令コードに従って演算を実行する。演算器の性能は、理想的には 10 MFLOPS である。

③は OPCODE-S で指定される条件に従って真/偽などのフラグを出力するか、パケットのノード番号をインクリメントして分岐制御を行う。

以上①～③における処理の組み合わせにより種々の複合命令が実行可能である。

(2) ベクトル演算機構の導入

循環パイプラインリング方式のデータ駆動計算機では 2 項演算実行時のパイプライン充足率が最大でも

50% となってしまいます。すなわち、演算器の能力が 10 MFLOPS であるにもかかわらず、実際の最大性能は 5 MFLOPS となってしまいます。また、特に配列などの定型的構造体に対して単純な計算を繰り返す際には、問題のいわゆる横方向の並列度の不足とループオーバーヘッドにより、演算パイプラインの充足率がさらに低下するという問題がある。

EDDEN では、この問題に対処するために演算部にベクトル演算機構を導入した。すなわち、ベクトル演算命令が発火すると外部メモリに格納されたベクトルデータに対して局所的に繰り返し演算を施す¹¹⁾。

図 8 にベクトル演算機構の構成を示す。演算ベクトル長レジスタ (N) はベクトル演算を施すベクトル長を指定するデクリメント機能付きレジスタ、通信ベクトル長レジスタ (M) は、同様に他の PE と通信するベクトルデータの長さを指定するデクリメント機能付きレジスタである。また、左オペランドアドレスレジスタ (LA), 右オペランドアドレスレジスタ (RA), 結果アドレスレジスタ (DA) および通信アドレスレジスタ (CA) によってベクトルの各要素が参照/更新される。

さらに、差分レジスタ (LI, RI, DI, CI) によって、1 要素の処理が行われるごとにアドレスレジスタが更新される。ビット長は M, N が 12 ビット、アドレスレジスタが 20 ビット、差分レジスタが 10 ビットである。

(3) ベクトル命令の種類と性能

ベクトル命令は以下の 8 種類である。

- ① $op\ V \Rightarrow V$ (ベクトルの各要素の 2 乗など)
- ② $V\ op\ V \Rightarrow V$ (ベクトル間の演算)
- ③ $S \Rightarrow V$ (ベクトルの初期化など)
- ④ $op\ V \Rightarrow S$ (ベクトルの総和など)
- ⑤ $V\ op\ S \Rightarrow V$ (各要素のスカラ倍など)
- ⑥ $V \Rightarrow [V]$ (ベクトルの分配など)
- ⑦ $[V] \Rightarrow V$ (ベクトルの収集など)
- ⑧ VSEND (ベクトルの送信)

なお、ベクトルの受信は構造体パッケージが入力バッファ (IBUF) に到着すると自動的に起動される。

EDDEN では、外部データメモリのアクセスレートをクロックの 2 倍の 20 MHz としており、ベクトル演算時の性能は③が 10 MFLOPS, ④が 2.5 M

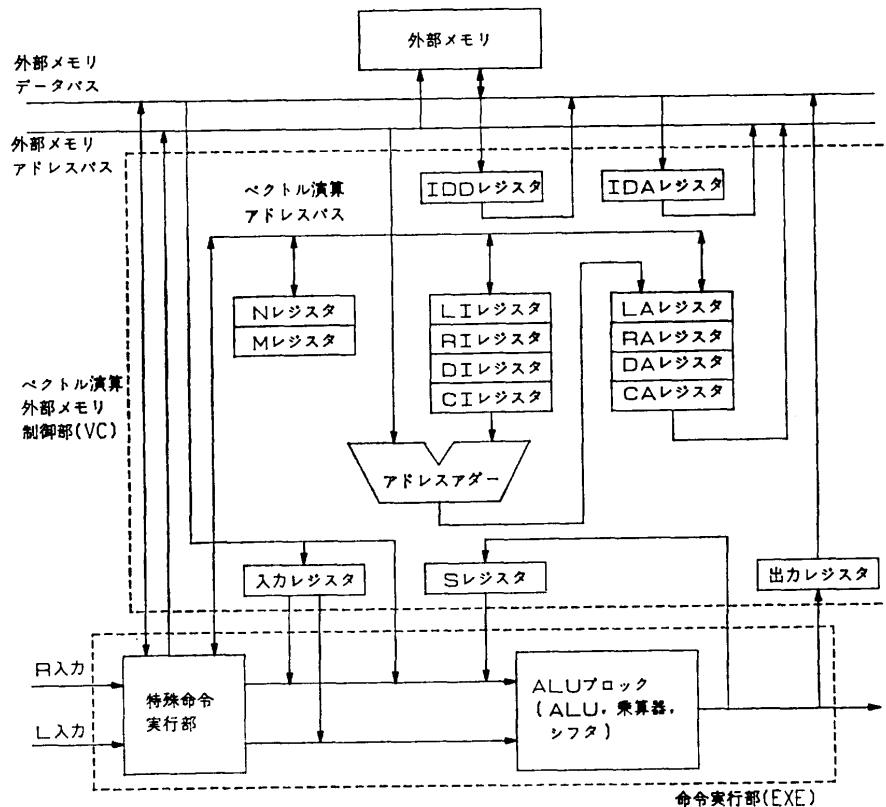


図 8 ベクトル演算機構の構成
Fig. 8 Configuration of vector-operation unit.

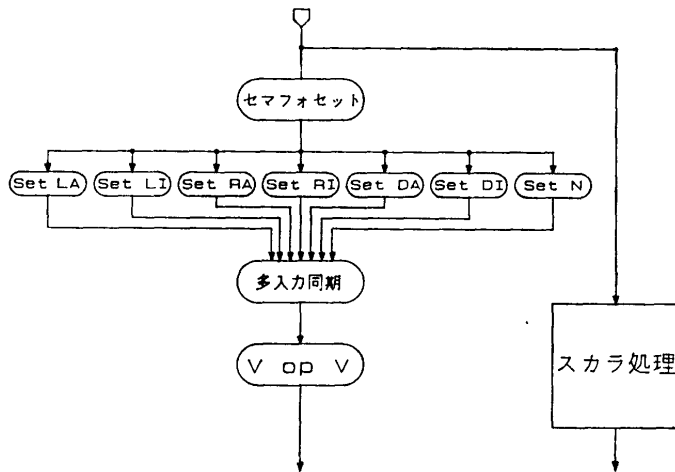


図9 ベクトル演算とスカラ演算の並列実行

Fig. 9 Parallel execution of vector-operation and scaler-operation.

FLOPS, それ以外は 5 MFLOPS である。

(4) ベクトル演算とスカラ演算の並列実行

さらに EDDEN ではベクトル演算と循環パイプラインによる通常のスカラ演算が、同一のパイプライン演算器を 50% ずつ時分割で共用する方式をとっている。これにより、両者の並列実行を可能にするとともに、演算パイプラインの充足率を最大限まで高め、理想的には 10 MFLOPS/PE の性能を得ることが可能である。

すなわち、図9に示すような、ベクトル演算とスカラ処理とが並行的に生じるような処理を実行した場合に、FCCM を経たスカラデータと、前述のアドレスレジスタによって読み出されたベクトルデータとが、図8の ALU ブロックに平均的に交互に投入され、ALU ブロックの各パイプラインステージは、休みなく稼働できる。

5.4 セマフォの導入

EDDEN の各 PE は、次の4種類のセマフォフラグを保持しており、種々の共有資源を保護するための排他制御に使用することができる。

① ネットワークセマフォ

各 PE の通信制御部 (NC) を保護するためのセマフォ。

② 構造体通信セマフォ

構造体通信用レジスタ (CA, CI, M) を保護する。

③ ベクトル演算セマフォ

ベクトル演算用レジスタを保護する。

④ 汎用セマフォ

その他の共有資源を保護するために汎用的に使用。

各セマフォは、セマフォチェック&セット命令によってセットされ、セマフォクリア命令によってクリアできる。また、①は構造体パケットのテイルが NC を通過した時に、②は構造体送信の終了時に、③はベクトル命令実行の終了時に、それぞれ自動的にクリアされる。

5.5 メモリアクセス制御方式

EDDEN では、外部データメモリ (EDM) に対しての各部からのアクセス要求が競合するため、これらを適正に調停することが重要となる。

メモリアクセス制御の基本方針は、

① メモリアクセスタイミングを、 $\Phi 1$, $\Phi 2$

の2種類のタイムスロットに分割し、各部

のアクセスタイミングをいずれかに割り当てる。(各タイムスロットはクロックサイクルの 1/2 である。)

② プログラムアクセスと他のメモリアクセスができる限り競合しないようにする。

③ 緊急度の高いアクセス要求ほど優先的にアクセス権を与える、などである。

具体的なアクセスタイミングの割り当てと優先度を以下に示す。(それぞれ優先度の高い順)

[$\Phi 1$] ① 外部キューのリード/ライト

② 構造体通信のためのリード/ライト

③ スカラアクセス命令によるリード/ライト

④ ベクトル演算の結果オペランドのライト

⑤ ベクトル演算の左オペランドのリード

[$\Phi 2$] ① 外部キューのリード/ライト

② ベクトル演算の右オペランドのリード

③ プログラムアクセスのためのリード

このように、プログラムアクセスを他の大部分のアクセスとは別のタイムスロットに割り当てているために、プログラムフェッチと他のメモリアクセスとの競合はほとんど生じない。これによって、プログラムとデータを同一メモリ空間に置きながらも、これによる性能低下を最小限に押さえることができると考える。

6. おわりに

以上、EDDEN のアーキテクチャについて概要と特徴を述べ、その有効性について論じた。本アーキテクチャは①実現性、②ハードウェア資源の有効利用、③デッドロック回避を重視したものであり、並列処理

計算機の実用化のための一手法であると考えている。

現在、要素プロセッサ LSI の開発をほぼ終了し、演算結果の即時表示機能としての高速のグラフィックボード、大容量の画像メモリ等を接続した EDDEN システム第1版を開発中である。また、EDDEN のプログラミング環境としては、アセンブラ第1版、コンパイラ第1版、およびソフトウェアシミュレータの開発を終了している。

今後は、本アーキテクチャの種々の特性をさらに詳細に評価することが第1課題である。特に本論中で述べたオペランド待ち合わせ時の寿命、さらには並列計算時の通信コスト、構造体通信時のオーバーヘッドなどについて、実用的プログラムを用いシミュレーションおよび実機上での評価を行っていききたい。また、1,024台のプロセッサを結合した高並列データ駆動計算機の実現に向け、システムの実装方式、調整手法に関する詳細な検討を行うことが不可欠である。さらに、言語処理系ソフトウェアへの自動ベクトル化/並列化機能の組み込み、ソフトウェアデバッグ環境の高度化も大きな課題であると考えている。

末筆ながら、本研究を御指導、御支援してくださった関係各位に厚く感謝の意を表します。

参 考 文 献

- 1) Sharp, J. A.: *Data Flow Computing*, Ellis Horwood (1985).
- 2) 長谷川ほか: 記号処理用データフローマシン DFM, 電子通信学会データフローワークショップ予稿集, pp. 65-69 (1986).
- 3) 平木ほか: シングルチップデータ駆動計算機的设计思想, 第32回情報処理学会全国大会論文集, 6R-6 (1986).
- 4) 田中ほか: データ駆動計算機 SPM の試作, 第36回情報処理学会全国大会論文集, 7B-5 (1988).
- 5) 西川ほか: データ駆動計算機 SPM のコンパイラ, 第36回情報処理学会全国大会論文集, 7B-6 (1988).
- 6) 田中ほか: データ駆動計算機 SPM の性能評価(1), 第37回情報処理学会全国大会論文集, 1N-4 (1988).
- 7) 岡本ほか: データ駆動計算機 SPM の性能評価(2), 第37回情報処理学会全国大会論文集, 1N-5 (1988).
- 8) 三浦ほか: 並列データ駆動計算機 EDDEN の概要, 第38回情報処理学会全国大会論文集, 2T-2 (1989).
- 9) 田中ほか: データ駆動計算機 EDDEN における発火制御とカラー管理, 第39回情報処理学会全国大会論文集, 6W-2 (1989).

- 10) 三浦ほか: データ駆動計算機 EDDEN の通信制御機構, 第39回情報処理学会全国大会論文集, 6W-3 (1989).
- 11) 大橋ほか: データ駆動計算機 EDDEN の演算機構, 第40回情報処理学会全国大会論文集, 3L-6 (1990).
- 12) 三浦ほか: データ駆動計算機 EDDEN のアーキテクチャ, 並列処理シンポジウム JSPP '90, pp. 169-176 (1990).

(平成2年8月21日受付)

(平成3年2月12日採録)



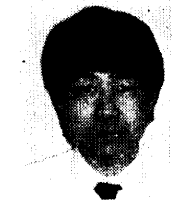
三浦 宏喜 (正会員)

1959年生。1982年京都大学工学部情報工学科卒業。同年三洋電機(株)に入社。デジタル信号処理、計算機アーキテクチャ、並列処理、データフローマシン等の研究に従事。現在、同社情報通信システム研究所コンピュータ研究部に所属。



川口 正樹 (正会員)

1961年生。1986年東京理科大学理工学部物理学科卒業。同年三洋電機(株)入社。現在三洋電機情報通信システム研究所コンピュータ研究部所属。並列プログラミング環境、数値シミュレーション手法等の研究開発に従事。



田中 一行 (正会員)

1962年生。1985年立命館大学理工学部電気工学科卒業。同年三洋電機(株)に入社。現在、同社情報通信システム研究所コンピュータ研究部に所属。データフローマシン、並列処理計算機等の研究開発に従事。



大橋 秀紀

1960年生。1983年立命館大学理工学部電気工学科卒業。同年三洋電機(株)に入社。現在、同社情報通信システム研究所所属。並列処理計算機の研究開発に従事。コンピュータアーキテクチャ、デジタル信号処理に興味を持つ。電子情報通信学会会員。

**清水 雅久**

昭和 27 年 9 月 13 日生。昭和 50 年大阪大学工学部通信工学科卒業。昭和 52 年同大学大学院工学研究科通信工学専攻修士課程修了。同年三洋電機(株)に入社。現在同社情報通信システム研究所コンピュータ研究部に所属。デジタル信号処理、並列処理システムの研究に従事。電子情報通信学会会員。
