計算機アーキテクチャ 59−2
オペレーティング・システム 29−2
(1985・10・24)

ＢＣプロセッサアレイの実現

IMPLEMENTING BC PROCESSOR ARRAY

小畑正貴[+] 金田悠紀夫[++] 宮垣嘉也[+]
Masaki KOHATA Yukio KANEDA Yoshiya MIYAGAKI

[+] 岡山理科大学
Okayama University of Science

[++] 神戸大学
Kobe University

[1] はじめに

Kungらによって提案されたシストリックアレイ[1]は、単純で規則的なデータの流れを受け、それにパイプライン制御されたシストリックアルゴリズムを適用していくことにより目的の計算を高並列に実行することができる。われわれは2)において、隣接プロセッサとの結合端子に加えてバスによるデータの放送端子を持つ演算プロセッサ(BCプロセッサ)を提案し、BCプロセッサから構成される1次元または2次元のプロセッサアレイが種々のマトリクス計算を約 nステップで実行できることを示した。

本論文では、現在われわれが試作を行っている1次元BCプロセッサアレイについて述べる。以下、プロセッサアレイの構成、BCプロセッサのハードウェア、基本動作、各種計算アルゴリズムについて述べる。

[2] １次元ＢＣプロセッサアレイ

BCプロセッサアレイの構成を図1に示す。各BCプロセッサエレメント(以下PEと呼ぶ)は RAMとMPU(micro processing unit)で構成され、 2つの入力ポート(BIとLI)と2つの出力ポート(BOとRO)とを持つ。

BI(Bus Input) ポートはホストの出力データバス(0 バス)に接続され、バスからのデータを入力する。また、BO(Bus Output)ポートはホストの入力データバス(I バス)に接続され、ホストに対してデータを出力する。LI(Left Input)ポートは左の PE からのデータを入力し、 RO(Right Output)ポートは右のPEにデータを出力する。

BIおよびBOポートに対するバスアクセスには2つのモード(ブロードキャストモードとダイレクトモード)がある。ダイレクトモードではホストコンピュータはアドレスによって特定のPEを指定し、そのPEに対してデータの書き込みと読み出しを行う。一方ブロードキャストモードでは、ホストからの出力データは 0バスを通して全PEに同時に書き込まれる(放送)。また、ブロードキャストモードでの読み出し時には、プログラムで指定された条件を満足する複数 P E(場合によっては0台のときもある)からデータがオープンコレクタ出力され、 Iバス上でワイヤードオアがとられてホストコンピュータに入力される。

CPU にザイログ社のワンチップマイクロコンピュータ Z-8を用いたBCプロセッサエレメントの詳細を図2 に示す。入出力端子の機能は前節で説明しているが、 BOポートだけはZ-8が内蔵する入出力ポート数の制限からシリアル信号にした。こうすることによってハードウェア量を少なくすることができた。また各ポートはハンドシェイクのための信号(AcknowledgeとStrobe)をそれぞれ持っている。

Z-8は8ビットのマイクロプロセッサで、次のような特徴を持つ。

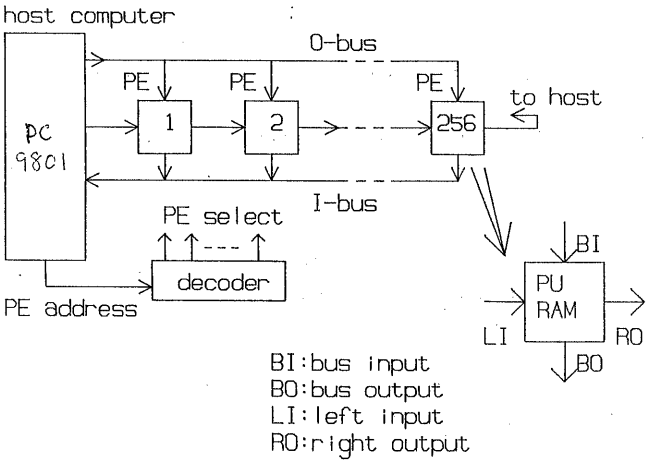


Fig. 1 One-dimensional BC processor array

(1) ROM(2K)、レジスタ(124)、並列ポート(ハンドシェイク機能付き)、直列ポートを内蔵しており、外部回路をほとんど必要としない。
(2) 命令の先読みを行うことにより、最小命令実行時間を1.5us (4MHzクロック時)と高速にしている。
(3) 内部レジスタはすべてアキュムレータとして使えるので、データ転送命令を大幅に削減できる。

このPE 1台を構成するICを以下に挙げる。

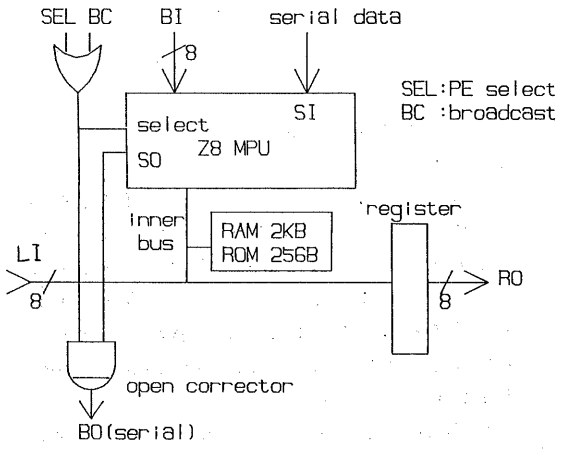


Fig. 2 The BC processor hardware

・Z8-81(CPU)
・6116(RAM)、2732(ROM)
・TTL(74LS00,74LS02,74LS139,74LS374*2)

1台のPEは1枚のプリント基板に実装される。そしてこれは40本の信号線を持つ。われわれは1枚のマザーボードに32台のPEを載せ、さらにこのマザーボードを8枚結合することにより256PEのシステムを構成する。→最終ページ写真1,2参照

[3] 基本動作

3.1 起動

各PEには、直列入力ポートからプログラムをRAMに入力し実行を開始するための簡単なモニタプログラムを持たせている。これにより、ホストから各PE内のRAMを直接アクセスする場合に通常必要なアドレスバスラインやアドレスマルチプレクサ等のハードウェアをなくすることができた。

ホストコンピュータからのリセットにより各PEはプログラム入力状態となり、プログラムロード後の実行開始指令により実行を開始する。プログラムロード時には、全PEに対して同一プログラムが同時にロードされる。

3．2　同期

隣接PE間での同期はハンドシェイクにより自動的にとられる。全PE間での一斉同期はブロードキャスト転送時のハンドシェイクにより次の手順で行う。

step1. ホストがダミーデータをブロードキャスト転送する。
step2. 処理を終えたプロセッサはこのデータを入力する。
step3. 全 PE が入力を終えた時点でバス上のAcknowledge信号が1になるので、各PEはこれを待って次の動作を開始する。

[4] 応用

4．1　ベクトル・行列積

行列$A=(a_{ij})$とベクトル$x=(x_1,x_2\ldots x_n)^T$の乗算問題(y=Ax)をとりあげる。

図3において、左端からは常に0が入力される。帯幅がwでPE数がこれより大きい場合、PE_wはデータをROポートのかわりにBOポートから 0バスに出力する。計算はk=1～nに対して以下の手順で進められる。各BCプロセッサは 1ステップごとに次のサブステップを行う。

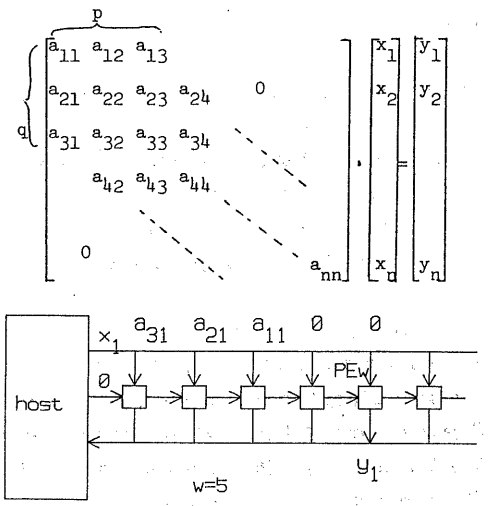


Fig. 3 Vector-matrix multiplication

step1. x_kをブロードキャスト転送後、ダイレクトモードにてAの第k列を各PEに分配する。
step2. 各PEは LI+x_k*a_{ik} を計算して ROに出力する(ただしPE_wはBOに出力する)。

同期は各データ転送時のハンドシェイクにより自動的にとられるので、ステップごとでの一斉同期の必要はない。全計算は (n+p-1)ステップで終了する。

4．2　行列・行列積

計算は以下のサブステップにより進められる。

step1. ブロードキャストモードにてa_k列の各要素を順に出力し、全PEに記憶させる。次にb_k行をダイレクトモードにて各PEに分配する。
step2. 以下のサブ・サブステップを実行
(i)1ステップ目は全PEがc_{ij}の中間結果(前回

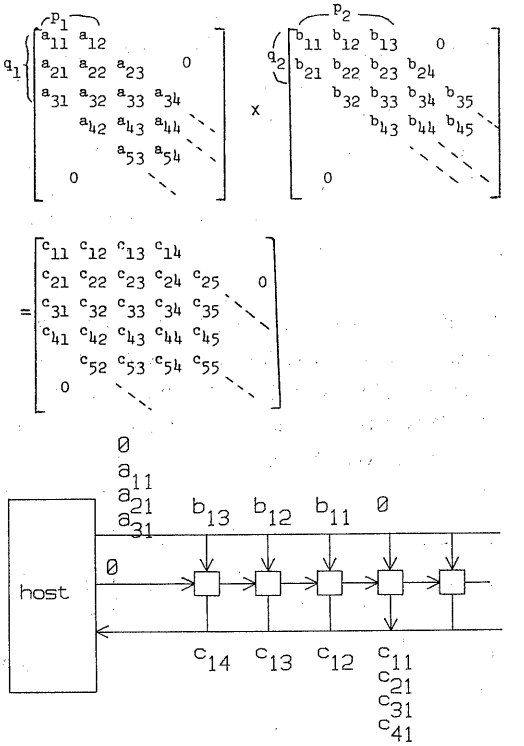


Fig. 4 Matrix-matrix multiplication

ステップでLIより入力し内部に保持)に$a_{i,k}$*$b_{k,j}$を加えBOに出力、ホストが順に読む。
(ii)2-wステップでは同様の計算をするがPE_wのみがBOに出力し、他はROに出力する。各PEはこれをLIから入力し内部に保持する。

サブ・サブステップの実行にw時間必要なので、全計算時間は約nw時間となる。

4．3　その他の行列計算

BCプロセッサアレイではこのほかに次のような行列計算が可能である。

- ＬＵ分解　　約ｎｗ時間
- ガウス消去（前進）　　約ｎｗ時間
- 三角方程式　　約ｎ時間

4．4　ソーティング

a_1-a_{k-1}を整列されたデータ($a_1<a_2<...<a_{k-1}$)とすると、各PEはa_kに対して以下のサブステップを実行すればよい。

step1. a_kをブロードキャスト転送する。$a_i>a_k$となるPE(PE_i～PE_{k-1})はROへa_iを出力する。
step2. 一斉同期をとった後、PE_i～PE_{k-1}のうちでLIからのデータ入力のあるPEはこれを入力し、LIにデータのないPE(PE_i)はa_kを自分のデータとする。

n個のデータに対してnステップで整列は終了する。またすでに整列されているデータに対して追加を行う場合でも、1データにつき1ステップで処理が終了するので単発的なデータに対しても早い応答が得られる。

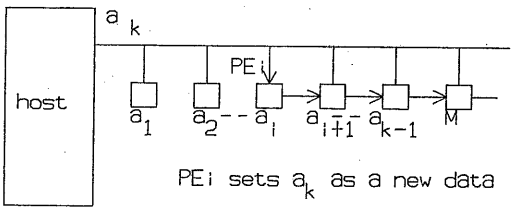


Fig.5 Sorting

4．5　サーチング

図6 において各PEの持つデータはすべて異なるものとする。データの検索は次の手順で行われる。

step1. 検索データ xをブロードキャスト転送する。
step2. 一致するデータを持つPEが自分のPE番号をBOに出力したのち同期をとる。ホストはこのPE番号と関連データを読み出す。一致データがない場合はホストプロセッサは Acknowledge信号を得られず、一致データがないことを知る。

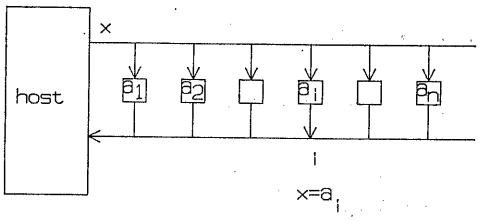


Fig.6 Searching

4．6　最小値(最大値)

複数データa_i(i=1..n)の中の最小値を得るには、データの最上位ビットから順にビット数だけ以下の操作を繰り返す。

step1. 全PEはOバスにデータを出力(オープンコレクタ出力)する。
step2. ホストは第kビットを調べ、その値をIバスにブロードキャスト転送する。
step3. 自分の第kビットがstep2の値より大きいPEはOバスへの出力をやめる。

以上の操作により mビットのデータに対してmステップで最小値が求められる。これはデータの個数に関係しない。mがバス幅より大きい場合は上の操作を上から mビットずつ順に行えばよい。また、最大値を得るには、最初にデータのビット反転を行っておけばよい。

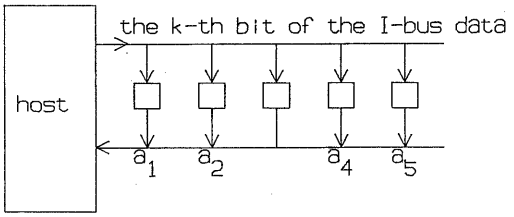


Fig. 7 Searching minimum data

4．7　画像処理

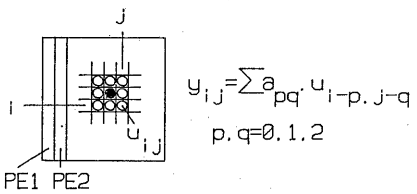
図8 のようなn×nの画像に対する3×3のテンプレートによるたたみ込みを考える。 PE_kに画像の第k列を割り当てる。

step1.　第i行を順に Iバスに出力する。 画素 u_{ij}の値をPE_{j-1},PE_j,PE_{j+1}の 3PEが同時に取り込む。（他のPEは入力値をすてる）

step2.　PE_{j-1} は過去に入力しておいた隣接画素値と step1での入力とから $\sum a_{pq} * u_{i+p,j+q}(p,q=-1,0,1)$を計算する。

step3.　ホストはこの値をOバスから順に読む。

全計算はnステップで終了する。



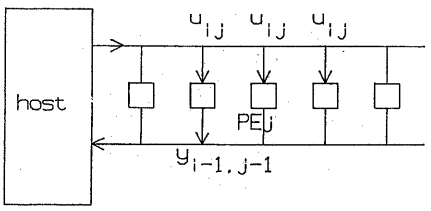


Fig. 8 2-D combolution

[5] まとめ

本論文で提案したBCプロセッサアレイは、データ線を共通バス接続する点で通常のメモリチップに演算機能、ブロードキャスト機能、隣接データ転送機能を付加したものと考えることができ、実装も容易で実現性が高い。 現在のLSI技術でも PE の1チップ化は容易に可能であり、VLSI技術により複数PEの 1チップ化も可能となる。

パイプライン処理は大量データに対する連続処理に対して平均的な意味での高速処理が実現できるが、単発データに対する応答は遅い。これにブロードキャスト機能を加えた本システムは少数データに対しても早い応答が得られる点で優れている。

参考文献

1)H.T.Kung and C.E.Leiserson, "Systolic Arrays (for VLSI)",Proc.Symp.Sparse Matrix Computations and Their Applications, Nov. 2-3,1978,pp.256-282

2)金田、小畑、前川、ＢＣプロセッサアレイと高並列マトリクス計算、情処論、Vol.24, No.2 pp.175-181 (1983)

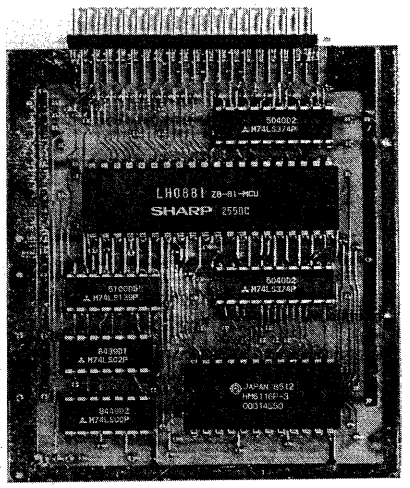


写真1　BCプロセッサ基板

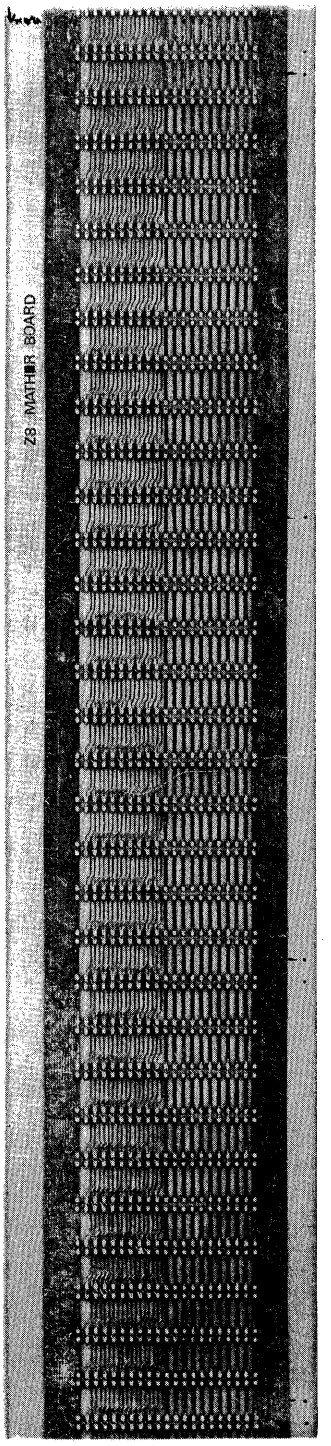


写真2　マザーボード（全長43cm）