

高並列リングアーキテクチャ

吉沢英樹 加藤英樹 市來宏基 浅川和雄
(株) 富士通研究所

本研究の目的は、ニューロコンピューティング研究ツールとして、高速シミュレーションが可能な計算機を開発することである。我々は独自の高速マルチプロセッサ型アーキテクチャを開発し、2プロセッサの試作機による実験結果から64プロセッサでNETtalkの学習を102MCUPS、各層に256ニューロンを持つ3層ネットワーク学習を256プロセッサでは、583MCUPSの速度で処理できることがわかった。

A Highly Parallel Architecture Using Ring-Register Data Way

Hideki Yoshizawa, Hideki Kato, Hiroki Iciki and Kazuo Asakawa

Computer-Based Systems Laboratory

Fujitsu Laboratories Ltd., Kawasaki

1015 Kamikodanaka, Nakahara-ku, Kawasaki 211, Japan

The purpose of our research was to develop a highly parallel neurocomputer for fast neural computation.

One of the most important results of our work was that we could get 583 million connection updates per second (MCUPS) using our architecture when the system consists of 256 processors elements and learns for a large, 3-layer neural network with 256 neurons in each layer.

The system's performance was estimated from experimental results of a pilot system consisting of two processors which execute exclusive-OR learning using an incremental error back-propagation algorithm.

1 はじめに

近年、生体の情報処理メカニズムを取り入れたニューロコンピューティングシステムの開発が盛んである。この研究分野では、コンピュータシミュレーション技術に負うところが大きく、非常に大きな計算処理能力を必要としている。また、ニューロコンピューティングの応用範囲が拡大され、場合によっては実時間処理を要求されることもある。

ここに報告するアーキテクチャは、リングレジスタと呼ぶ高速データ伝送機構を持ち、層構造ネットワークモデルの学習則として広く用いられている、バックプロパゲーションアルゴリズムを極めて高速に処理を行うことができるものである。これによって、ニューロコンピュータを研究する上でボトルネックとなっている学習処理のTAT（ターンアラウンド時間）を短縮することが可能となる。

本稿ではアーキテクチャの特長、試作機(2PE版)の構成、及び同システムによる2入力排他的論理和(XOR)ニューラルネットワークの学習処理を行った場合について述べる。

2 基本アーキテクチャ

本研究で開発したシステムは、3つの双方向バスを持つ、トレイと呼ぶレジスタブロックとプロセッサの対を、多数リング状に配置したSIMD型計算機である(図1)。トレイは、プロセッサの外部アクセス用ポートの一方に接続され、隣り合う他のトレイと共にリング状の32ビットバスを形成する。このリングは、論理機能的には、双方向のサイクリックシフトレジスタであり、システムレベルでデータフローを考えると、プロセッサ外部に設けられたパイプラインレジスタの役割を果たす。故に、プロセッサ間の直接通信は行わない。

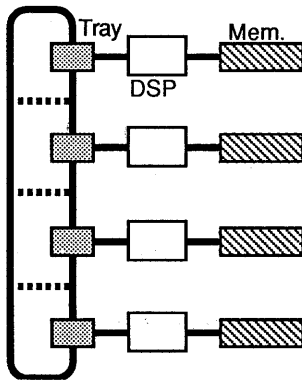


図1 リングレジスタアーキテクチャ

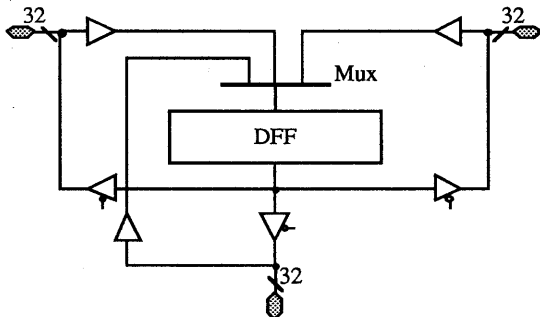


図2 トレイの構成

また、その構造はプロセッサとのインターフェイス回路とマルチプレクサ付き32ビットDFFの2つで構成される極めて簡単な回路である(図2)。

トレイに対する命令は、DSPのアドレスバス下位2ビット、及びプロセッサのアクセスモード信号(XR_W)を用いて与える。トレイの持つ命令は、NOP、RSR（リード&シフトライト）、RSL（リード&シフトレフト）、WSR（ライト&シフトライト）、WSL（ライト&シフトレフト）、RD（リード）、WR（ライト）の7種である。図3に、これら命令によるデータフロー制御の様子を示す。

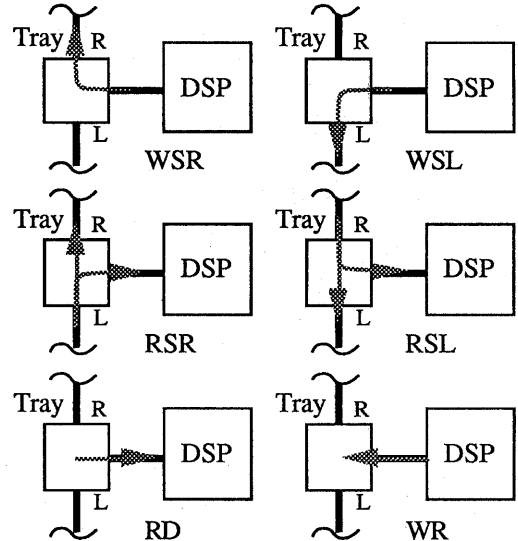


図3 データフロー制御

また、各々のプロセッサは浮動小数点ALUや乗算器を内蔵し、重みベクトルやプログラムはローカルメモリ内に保持され、もう一方の外部アクセスポートを介してアクセスされる。さらにプロセッサの演算処理サイクルとトレイの転送サイクルを同一とすることによって、プロセッサの演算時間とデータ通信時間を重ねることができ、見かけ上データ転送に関するオーバーヘッドを省略できる。しかもSIMD計算機であるため、すべてのプロセッサは互いに緩衝せず、独立にデータ収集を行い、処理することができる。

したがって、本アーキテクチャは台数効果を有するとともに、プロセッサ単体の演算処理速度がそのままシステム性能に反映される特長を有する。層構造ネットワークモデルに対しては層方向に時分割で逐次実行する。これによってネットワークの層数をハードウェアと無関係に自由に変えることができる。

3 処理アルゴリズム

図4に示すような3層ニューラルネットワークモデルについて、処理アルゴリズムを説明する。まず時刻0で、同図のように重みベクトルが配置され、四つの入力データがトレイ上にロードされていると仮定する。中間層の出力値Y1, Y2, Y3は同図の矢印方向にデータシフトを行ないながら、メモリ内の重みベクトルを順次読みだし、積和処理をデータがリング上を一巡するまで繰り返すことによって、各プロセッサ内のAccに生成される。

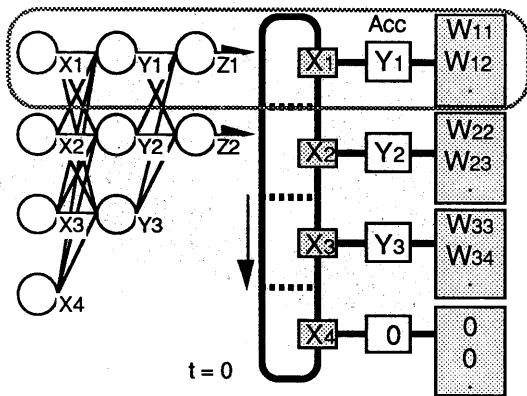


図4 前向き積和处理

これはちょうど(式1)に示すマトリックス演算を実行することにはかならない。但し、演算はマトリックスを斜めに横切る方向を同時刻処理の対象として進められることに注意していただきたい。

次に、出力層の値Z1, Z2を計算するには、いま得られたY1, Y2, Y3それぞれの値をAccからトレイに書き出した後、先程と同様にデータを一巡させた後Acc内を得ることができる(式2)。

このとき、注意すべきことはリング長が変化し、4番目のプロセッサはシステムのデータフローから自動的に切り離され、特にIDLE時の処理がプログラムされている場合を除き、プロセッサは再びリングがつかがるまで無意味な演算を繰り返すことになる。

次にバックプロパゲーション処理について説明す

$$\begin{matrix} \text{時間} \\ \rightarrow \\ 1 \quad 2 \quad 3 \quad 4 \end{matrix} \begin{matrix} Y_1 \\ Y_2 \\ Y_3 \\ 0 \end{matrix} = \begin{matrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \\ 0 & 0 & 0 & 0 \end{matrix} \times \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{matrix} \quad (1)$$

$$\begin{matrix} 1 \quad 2 \quad 3 \\ Z_1 \\ Z_2 \\ 0 \end{matrix} = \begin{matrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ 0 & 0 & 0 \end{matrix} \times \begin{matrix} Y_1 \\ Y_2 \\ Y_3 \end{matrix} \quad (2)$$

る。各々のプロセッサが出力層の値を計算した後、ローカルメモリ内に予め保持されていた教師データとの差を求め、シグモイド関数の微係数を掛け合わせ、誤差ベクトルE1, E2をAcc内を得る。この時点ですでに2層目と3層目間の重み修正処理は完了している。そこで1層目と2層目間の誤差ベクトル $\delta_1, \delta_2, \delta_3$ を計算するため、まず、トレイに0を書く。次に各々のAcc内の誤差ベクトルEiに重みベクトルを掛け、トレイ上のデータに加算し、その結果を再びトレイに書き込む(図5)。これはちょうど(式3)に示すマトリックス演算を実行することと等価であ

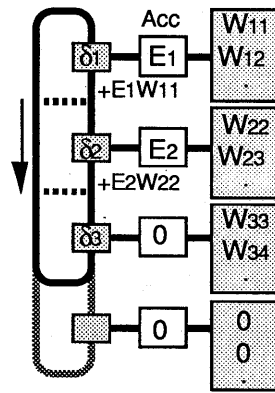


図5 後ろ向き積和处理

これら一連の処理を時空間チャートとして図6に示す。

この演算に用いられるマトリックスは、先に示した式1中の重みベクトルマトリックスを転置したものである。即ち、本アーキテクチャにおいては、メモリ内のデータを再配置することなく、しかも新た

$$\begin{matrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{matrix} = \begin{matrix} 1 & 2 & 3 \\ W_{11} & W_{21} & 0 \\ W_{12} & W_{22} & 0 \\ W_{13} & W_{23} & 0 \end{matrix} \times \begin{matrix} E_1 \\ E_2 \\ 0 \end{matrix} \quad (3)$$

なオーバーヘッドを生じさせることなく、 W_x 及び、 W_x の演算を同一システムで行えることを示している。このことは、バックプロパゲーションなるアルゴリズムの特長であって、その学習則を実行できる本アーキテクチャが潜在的に有する特長の一つである。

3. 試作機のシステム構成

図7に試作機のシステム構成を示す。このシステムは、プロセッサボード(図8)2枚からなり、SUN-3/Eをホストコンピュータとし、VMEバスを介して接続されている。使用プロセッサはTI社製DSP(デジタルシグナルプロセッサ), TMX320C30GBH2(10MHz版)[11]である。リングはフロントパネル側に設けたコネクタでプロセッサボード間を接続することにより実現する。

トレイ部は、PLDを用いて構成され、約1600ゲート相当である。各プロセッサには2Kw(1wordは32bit)の内部メモリとパリティ付64Kwの外部メモリを持つ。このローカルメモリは、外部インターフェイス回路を介し、ホストプロセッサからアクセス可能である。この回路は、標準VMEバスインターフェイス仕様に従い、アクセスモードとして特権・非特権データアクセスをサポートし、/dev/VME24d32又は/dev/VME16d16を介してUNIXよりアクセスでき、ホストプロセッサより32ビットデータアクセス(VME24d32)された場合は、プロセッサのローカルメモリを、ショートアドレスモード(VME16d16)によりアクセスされた場合はコントロール/ステータスレジスタをアクセスするよう制御される。

本試作機では複数プロセッサの同期制御、及びホストプロセッサとのアクセス競合等の問題を解決す

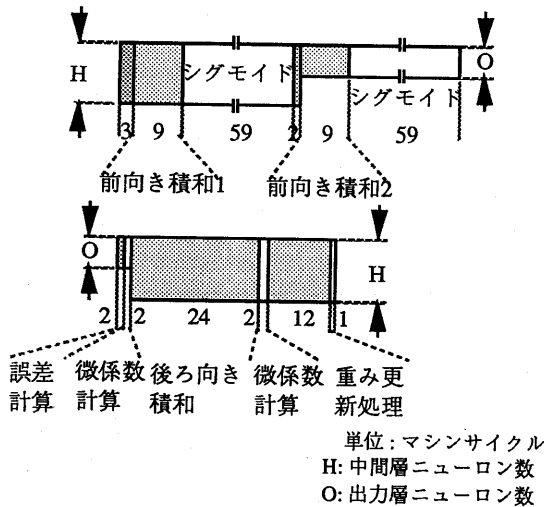


図6 時空間チャート

るため、同期制御回路を持っている。この回路は、本機内に一つだけ存在し、電源投入直後の同期操作の制御、ホストプロセッサからの非同期要求を同時に通知する処理、ローカルメモリのアクセス権の調停処理等を実行する。

一般に、このようなマルチプロセッサに対する同期制御や、非同期イベントの同時刻通知といった機能を実現することは難しい。特に高速で動作している複数のプロセッサに対し同時刻通知を行う場合、信号の到着時刻に関して厳しい条件が課せられる。筆者らは、一般に、割り込み入力信号等の非同期要求信号が、プロセッサへ供給されるクロックとの時間差（セットアップ/ホールド時間）によって条件付けられていることに着目し、図9に示すようににプ

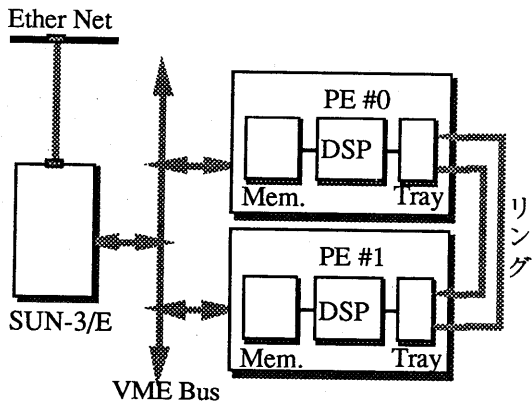


図7 試作システム構成

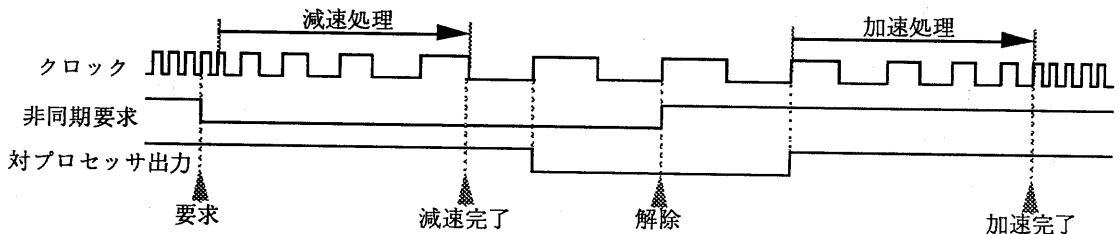


図9 同期制御シーケンス

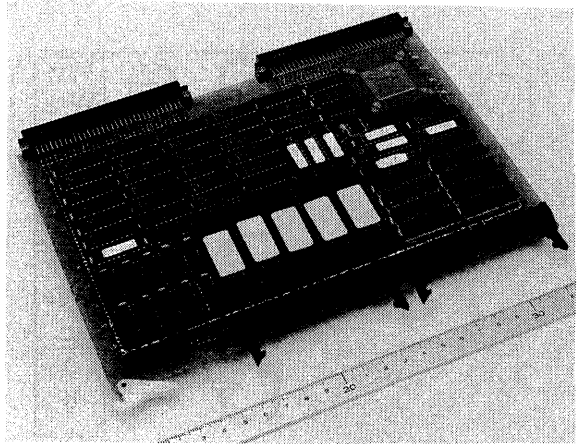


図8 プロセッサボード

ロセッサへの供給クロックの周波数を動的に変更可能とすることで、すべてのプロセッサに対して時間差条件を満たすことを可能とした。すべてのプロセッサに対して同時に要求を出す場合、ホストプロセッサからの要求をマスタ側の回路で受けた後、クロック周波数を一端適当な周波数まで下げ、プロセッサに対し要求を通知し、プロセッサからのアクセプト通知を確認後、クロック周波数を元の値まで引き上げる。このような処理を行うことで、すべてのプロセッサに対し、時間差条件を満たし、論理的な同時刻通知を行うことが可能となる。

4 実験結果

処理速度の評価として2入力排他的論理和 (XOR) のバックプロパゲーション学習を試作機で行なった。尚、バックプロパゲーションアルゴリズムのインプリメンテーションの詳細については、文献[10]を参照されたい。DSPのアセンブラプログラムから推定される、一回の学習に要するマシンサイクル（2クロック）は、前向き処理が18、シグモイド関数計算で118、後ろ向き処理で36、その他12の計184サイクルで、クロック入力8MHzの場合、推定処理時間は46 μ sであるのに対し、実測値は約53 μ sであった。推定値と実測値の差7 μ s (+15%)はDSP内部のバス・コンフリクトによって生じるオーバーヘッドである。図10は、XOR学習中の4種の入力パターン(00, 01, 10, 11)に対応する4通りの誤差曲線である。重みをパターン毎に更新する、インクリメンタル方式を用いている。

以上の実験結果を基にNETalkの学習速度を推定する。NETalkのネットワークは203(入力層) x 60(中間層) x (出力層) 26の3層ネットワークとして計算した。この場合、前向き処理795、シグモイド関数計算118、後ろ向き処理1304、その他12の計

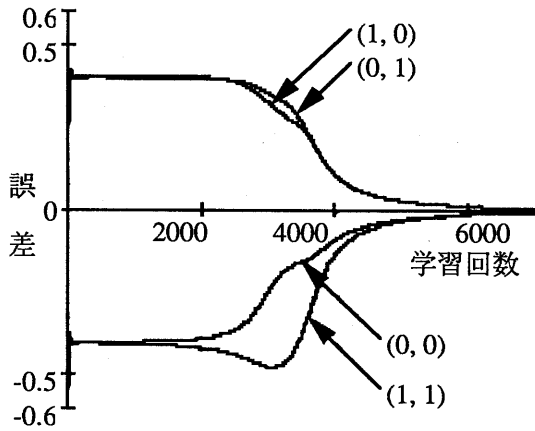


図10 XOR学習中の誤差曲線
2229サイクルとなる。DSPのクロックを33MHzとすると、本アーキテクチャを適用した64PE版の処理時間は134 μ s (=103.4 MCUPS)となる。表1は、この結果と1 MCUPS以上の性能が報告されたニューロコンピュータ[1-7]との処理速度比較である。

表1 処理速度比較

マシン名	NETtalk	最大処理速度
2PE	-	0.9
ANZApplus	-	1.5
Neuro Turbo	-	2
Delta	-	2.7
NeuMan	-	8.5
CM-1	2.8	13
Warp	17	17
64PE*	103	107
SX-2	72	180
256PE*	103	583

*: 予測性能, 単位: MCUPS

5 まとめ

リングレジスタを用いた新しいマルチプロセッサアーキテクチャと、試作機によるバックプロパゲーション学習の実験結果について述べた。また、実用機レベルの機種として64PE版、256PE版の性能を推定した。今後は、高並列コンピュータとして256PE版を実現し、本アーキテクチャの有用性を検証する予定である。

謝辞

高並列コンピュータの研究に対して御助言を下された、情報処理研究部門 棚橋部門長、ならびに林人工知能研究部長に感謝致します。

参考文献

- [1] R. Hecht-Nielsen, (1988). Neurocomputing: picking the human brain. *IEEE SPECTRUM*, 25(3), 36-41.
- [2] 佐藤, 岩田, 鈴木, 松田, 吉田, (1988). 汎用浮動小数点DSPによるニューラルネットワークアクセラレータ. 電子情報通信学会MEとバイオサイバネティクス研究会資料, MBE88-134, 83-88.
- [3] G. A. Works, (1988). THE CREATION OF DELTA: A NEW CONCEPT IN ANS PROCESSING. Proceedings of IEEE ICNN88, II, 159-164.
- [4] N. Kajihara, S. Matsushita, T. Nakata, & N. Koike (1988). Parallel Neural Network Simulation Machine: NeuMan. Abstract of the first annual INNS meeting, 544.
- [5] G. Bletloch, C. R. Rosenberg (1987). Network Learning on the Connection Machine. Proceedings of IJCAI87, 323-326.
- [6] D. A. Pomerleau, G. L. Guscoira, D. S. Touretzky, & H. T. Kung (1988) Neural Network Simulation at Warp Speed: How We Got 17 Million Connections Per Second. Proceedings of IEEE ICNN88, II, 143-150.
- [7] 麻生川, 西, 妹尾, (1988). スーパーコンピュータによるニューラルネットワーク学習. 情報処理学会第36回全国大会講演論文集, 2321-2322.
- [8] D. E. Rumelhart, G. E. Hinton, & R. J. Williams (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing* (Vol. 1, pp. 318-362). Cambridge, MA: MIT Press.
- [9] T. J. Sejnowski, C. R. Rosenberg (1986). NETtalk: A Parallel Network that Learns to Read Aloud. JHU/EECS-86/01, The Johns Hopkins University.
- [10] H. Kato, H. Yoshizawa, H. Iciki, & K. Asakawa (1990) A Parallel Neurocomputer Architecture towards Billion Connection Updates Per Second. Proceedings of IJCNN-90-WASH-DC, Vol. 2, pp. 47-50.
- [11] Texas Instruments Inc. (1988). Third-Generation TMS320 User's Guide.