

マルチポートメモリ結合を用いる並列プロセッサ MC1 の試作例
— ハイパーキューブ型と de Bruijn 型 —

酒居 敬一 園田 幸生 佐伯 嘉崇 阿江 忠

広島大学 工学部

本稿では、当研究室で製作された並列プロセッサ MC1 の試作例について述べる。並列プロセッサ MC1 の基本ボードはプロセッサ間通信のためにマルチポート共有メモリを使用しており、プロセッサエレメントはこのマルチポート共有メモリを介して合計4つのプロセッサに結合できる。

一つめの試作例としてハイパーキューブ型プロセッサを紹介し、もう一つの試作例として de Bruijn 型を紹介する。それぞれのプロセッサエレメントはグラフの辺に配置され、結合用のマルチポート共有メモリはグラフの頂点に配置される。

Examples of multi-port memory connected parallel processor MC1
— Hypercube type and de Bruijn type —

Keiichi Sakai Yukio Sonoda Yoshitaka Saeki Tadasu Ae

Faculty of Engineering, Hiroshima University

Kagamiyama 1-4-1 Higashihiroshima City Hiroshima Prefecture 724 JAPAN

In this paper, we describe two examples of parallel processor MC1 made in our laboratory. Parallel processor MC1 is using multi-port memory connection for inter-processor communication, total four processor elements can be connected via multi-port memory.

An example of Hypercube type parallel processor realize three or four dimensional Hypercube graph. The other is de Bruijn type parallel processor that realizes an arbitrary dimensional de Bruijn graph. A processor element is mapped onto an edge, a multi-port memory used for inter-processor connection mapped onto a vertex.

1 はじめに

我々はすでにマルチポートメモリ結合型並列プロセッサ MC1 の開発を始めている [16]。この並列プロセッサは、図 1 のようにマルチポート共有メモリの各ポートにプロセッサが接続される型のプロセッサである。ただし、ここで使用するマルチポート共有メモリは完全なマルチポート共有メモリではないが、機能的には複数のプロセッサから同時に read/write できるように設計されている。なお、複数のポートを持つ完全なマルチポート共有メモリは提案あるいは試作にとどまっている [3][8][9][10] (光結合 3 次元集積回路回路技術によるマルチポート共有メモリの実現も期待されている [2][3][4][5][6][7])。

本稿ではこのような結合をまとめてマルチポートメモリ結合 [1] と呼ぶことにし、マルチポートメモリ結合を用いた並列プロセッサ MC1 を試作したので、その試作例を述べる。本稿で使用したマルチポートメモリ結合型並列プロセッサ MC1 は、トランスピュータなどと同様、結合アーキテクチャは

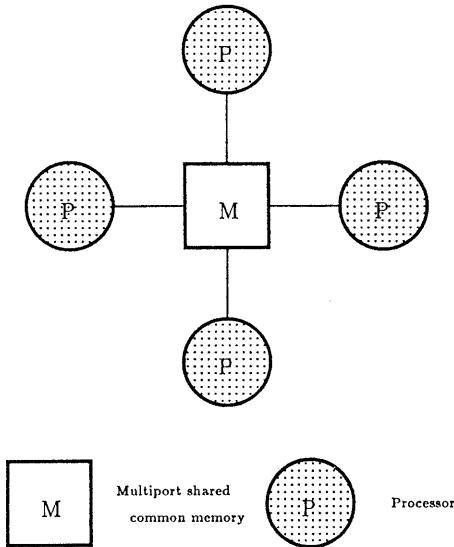


図 1: マルチポート共有メモリ結合 (概念)

自由を選べるという特徴を有する。さらに、通常プロセッサエレメントはグラフの頂点に配置するが、MC1 においては結合に使用したマルチポート共有メモリをグラフの頂点に配置し、プロセッサエレメントはグラフの辺に配置することもできる、ということから結合の自由度が増している。

マルチポート共有メモリの役割は、図 2 に示した図のように、図 1 のなかのメモリがその周囲に接続されたプロセッサ間を完全に結合しているというものである。そこで、ハイパーキューブ型と de Bruijn 型の結合アーキテクチャを採用した試作例について述べることにする。

2 並列プロセッサ MC1 について

並列プロセッサ MC1 [16] は 2 つの部品ともいうべき 2 種の基板から構成される。

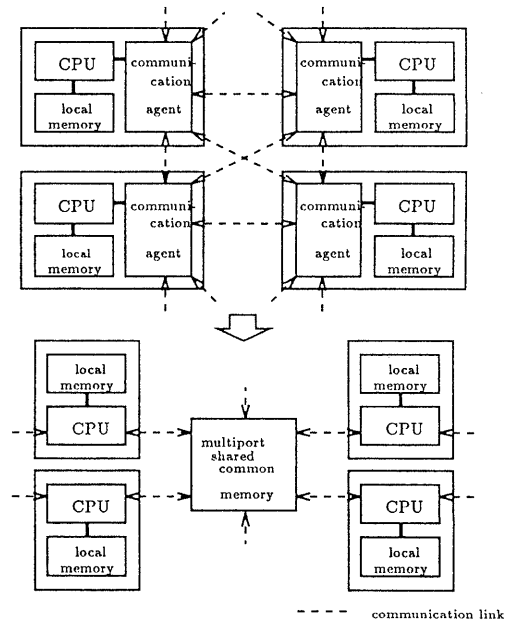


図 2: マルチポート共有メモリ結合 (機能)

Memory PORT to shared common memory

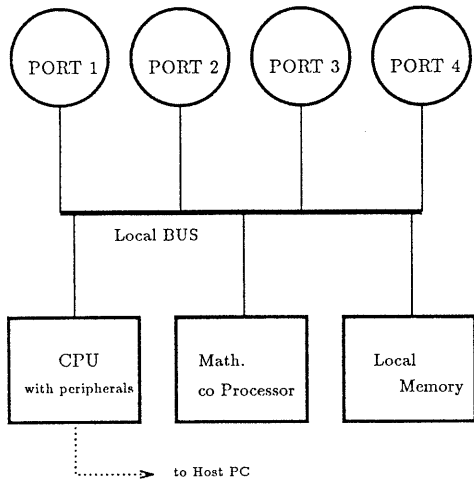


図 3: プロセッサエレメント

ひとつは、図3のプロセッサエレメントである。プロセッサエレメントはプロセッサ、マルチポート共有メモリへ結合するためのポート、ホストと通信するライン、プロセッサ間で同期をとる機構を装備してある。マルチポート共有メモリへ結合するためのポートはプロセッサのメモリ空間にマッピングされ、通常のメモリアクセスによりメッセージの送出や受信が可能である。このため非常に高速な、つまりバンド幅の広いメッセージ転送が実現できる。ホストと通信するラインはプロセッサに対してプロセスの制御・ロード・実行に使われる。

もうひとつは図4のマルチポート共有メモリで、どのポートから見える内容も同じに見えるように製作されている [3]。ただし、同時に2より多い数のプロセッサがひとつのメモリに対してアクセスすることは現状では不可能であるので、ひとつのマルチポート共有メモリの内部構成をいくつかのメモリブロックに分割し、不完全ながらも改善している。各メモリブロックは、クロスポイントスイッチにより切替えられ、アービタが各メモリブロックに対してメモリアクセスの衝突を避けるように制御

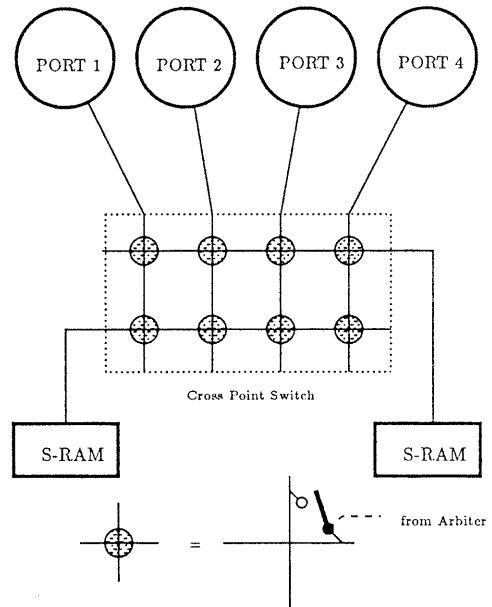


図 4: マルチポート共有メモリ

する。この方式では、read/write ともに平等の扱いをするので、read/write ともに平均的なアクセスタイムは同じである。各プロセッサからは1ワードおきに2組のメモリブロックが交互にアクセスされる構成になっており、プロセッサからの連続アクセスによる1プロセッサの専有状態を避けるように製作されている。ひとつのマルチポート共有メモリをアクセスしようとする時にそれぞれのプロセッサがアクセスするメモリブロックが異なっている限りでは実際に同時アクセスが可能である。実際の使用におけるプロセッサからマルチポート共有メモリへのデータ転送は、先にも述べたが、プロセッサのメモリアクセスによって行なわれるため、メッセージの送出や受信は無手順で高速であるという特徴を持たせることができる。

さらに、並列プロセッサ MC1 の構成要素であるプロセッサエレメントボードとメモリボードは、それぞれ図3と図4で示すように、互いに結合を行

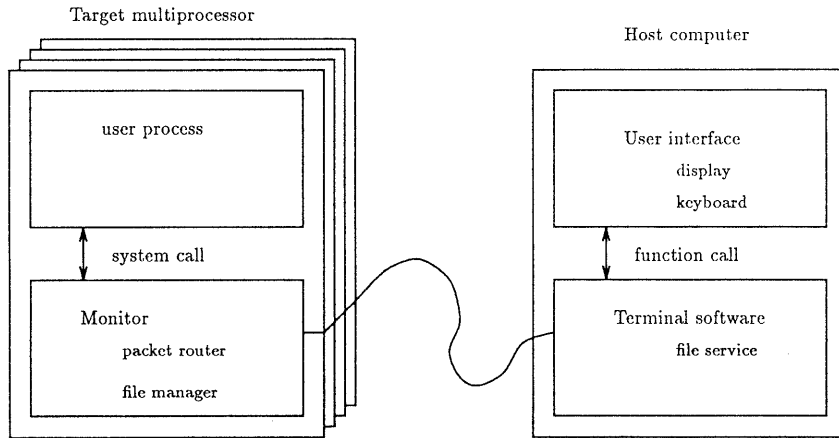


図 5: 並列プロセッサ MC1

なうことのできるポートを4つずつ装備しているため、自由な結合アーキテクチャを採用することができる。これらをラックマウントに納めホストと接続することによって並列プロセッサとして動作させることができる。全体の図を図5に示す。ホストコンピュータのキーボードとディスプレイを使用して各プロセッサの制御ができるようになっていいる。並列プロセッサ MC1 はプロトタイプとして製作されているため、つまりこの並列プロセッサをハードな応用に供する目的で製作されていないため、遅いのを我慢すればファイル転送も行なえなくはない。

3 ハイパーキューブ型プロセッサ

3.1 ハイパーキューブ型プロセッサの構成

ハイパーキューブ型プロセッサはプロセッサどうしの結合形態としてハイパーキューブグラフを採用している。通常、プロセッサ間を間接的に結合する場合、グラフの頂点にプロセッサを配置し辺はプロセッサ間通信のための線を配置する。しかし、我々の研究室で製作された並列プロセッサ MC1 はその構成要素であるプロセッサエレメントどうしが

マルチポート共有メモリを介して結合されているため、結合に用いるマルチポート共有メモリをグラフの頂点に配置することが可能である。これを、逆ハイパーキューブアーキテクチャとよぶことにする。

逆ハイパーキューブアーキテクチャにおいて問題であると考えられている点は、ハイパーキューブグラフの頂点の次数がハイパーキューブの次数と同じになるため、より多くのプロセッサを結合させるにはそれに見合った数の結合ポートを実装したマルチポート共有メモリを用意する必要がある、ということである。必要であればより多くのポートを持つ結合用マルチポート共有メモリを製作すれば良いと考えているので、ここでは触れないことにする。

実際に我々の研究室で製作された並列プロセッサ MC1 を構成する要素の一つである結合用マルチポート共有メモリには、プロセッサへ結合するためのポートを4つ実装している。ここでは、3次元ハイパーキューブを基本とする3次元逆ハイパーキューブ型プロセッサを試作例として紹介する。図6に示すのが3次元逆ハイパーキューブアーキテクチャによるプロセッサとマルチポート共有メモリの結合状態である。図7にラックにマウントされた状態の3次元逆ハイパーキューブ型プロセッサの写真を示す。ハイパーキューブグラフの次元数を N と

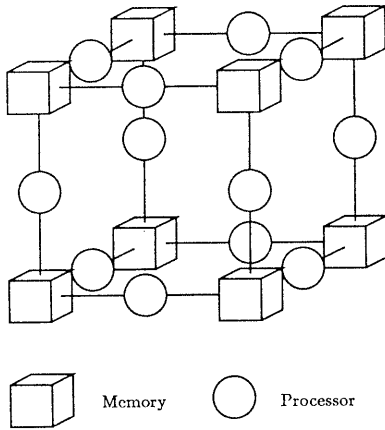


図 6: 3次元逆ハイパーキューブアーキテクチャ

すると、辺の数は $N \cdot 2^{N-1}$ 、頂点の数は 2^N となるので、3次元逆ハイパーキューブ型プロセッサのプロセッサ数は 12、結合に使用するマルチポート共有メモリの数は 8 となる。

3.2 ハイパーキューブ型プロセッサにおけるプロセッサ間通信

プロセッサ間の通信にはパケットルーティングを採用している。パケットは固定長であるが、パケットルータより上位に位置するソフトウェアの働きにより一つのデータを分割して送るようにすれば問題はないと思われる。プロセッサ間の通信路にマルチポート共有メモリがあるため、その内部をFIFO方式にてパケットを送出/受信することができる。

まず、FIFO方式を実現するためにマルチポート共有メモリの内部をパケット送出手続きのあるプロセッサ数と同数の領域に分けて管理する。各領域に対して1プロセッサのみが書き込めるように割り振る。パケットを送出したいプロセッサは、随時パケットを自プロセッサに割り振られた領域へパケットを書き込むことによって、パケットを送出できる。どこまで書き込まれたかはその領域に書き込

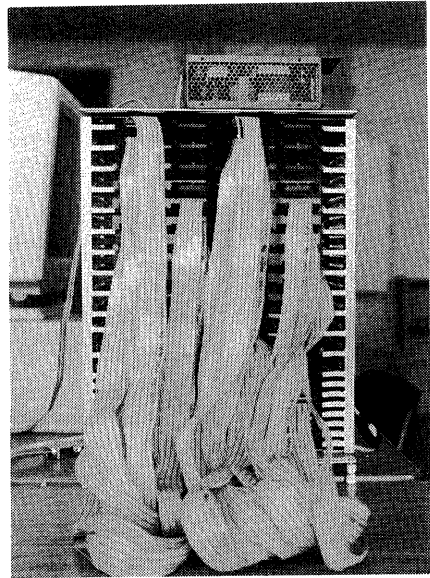


図 7: 3次元逆ハイパーキューブ型プロセッサ

みを許可されたプロセッサが行なう。パケットが書き込まれたことを知らせるために、パケットを受け取る可能性のあるプロセッサ数と同数のフラグを用意し、それをセットする。パケットを送り出したプロセッサはそのフラグがすべてクリアされたという事象を観測することにより認識することが可能である。パケットを受け取るプロセッサはフラグがセットされたという事象を観測することによりパケットの到着を認識できる。なお、パケットを受け取る側からみてどの位置まで読み出したかは受け取る側のみ管理し、パケットを読み出したという事象はフラグのクリアにより送出側のプロセッサに伝わる。

受け取ったパケットが自プロセッサ宛ではない場合、ルーティングを行なう必要がある。プロセッサの識別子はその両側に位置するマルチポート共有メモリのハイパーキューブグラフ上での頂点の値が容易にわかるように付けられる。そのため自プロセッサがパケットを受け取った方向と反対側のマルチポート共有メモリへパケットを送出すべきかどうかは、パケットの発信元と屈先の間の経路に自プロセッサが含まれるかどうかによって決定され

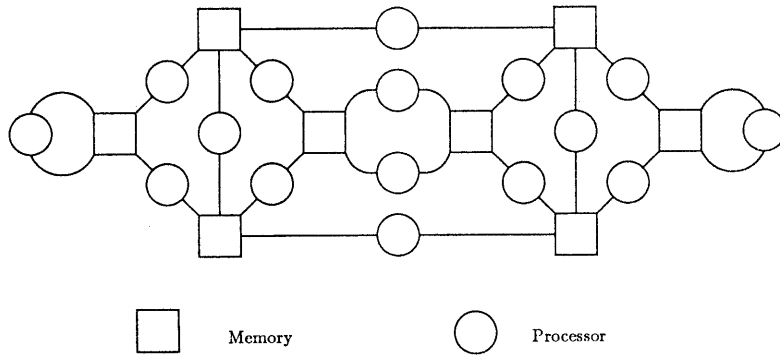


図 8: 3次元逆 de Bruijn アーキテクチャ

る。パケットを頂点に位置するマルチポート共有メモリに送出すれば既存のハイパーキューブグラフ上における頂点から頂点へのルーティングアルゴリズム [17] が利用可能であるので、ここでは触れないことにする。問題点としては、辺に位置するプロセッサがどちらのマルチポート共有メモリにパケットを送出するかである。届先がどちらから受け取るかを考慮に入れても場合の数は 4 であるので、通過する辺の数の少ないものが決定される場合は良いが、そうでない場合はハイパーキューブグラフの頂点の値のパリティによって振り分けている。パケットが偏って通過するという欠点があるが簡単なため現時点ではこのような便法を用いている。

4 de Bruijn 型プロセッサ

4.1 de Bruijn 型プロセッサの構成

de Bruijn 型プロセッサでは有向 de Bruijn グラフを基本としている。有向 de Bruijn グラフでは次元数を N としグラフの頂点の値を 2 進数の並び $X_{N-1}X_{N-2}\cdots X_1X_0$ で表すとすると、その頂点からは $X_{N-2}X_{N-3}\cdots X_00$ と $X_{N-2}X_{N-3}\cdots X_01$ という値を持つ頂点へ有向辺が伸びている。すべての頂点について 2 つの有向辺が出ていて、2 つの有向辺が入っている。つまり頂点の次数が 4 に固定されている。

これは、プロセッサエレメントとマルチポート共有メモリとも接続用のポート数が 4 であることから、物理的に並列プロセッサ全体が組み立てることが可能であればいくらかでもプロセッサ数が増えるという大きな利点があるということの意味する。

有向 de Bruijn グラフにおいて有向辺の数は 2^{N+1} であり、頂点の数は 2^N であることから、有向辺の数は頂点の数の 2 倍である。そこで、プロセッサ数は多いほうが良いという立場から、逆 de Bruijn アーキテクチャなるものを採用する。前の節でも述べたが、並列プロセッサ MC1 はその構成要素であるプロセッサエレメントどうしの結合をマルチポート共有メモリを介して行っていることから、結合用のマルチポート共有メモリをグラフの頂点に配置することが可能である。逆 de Bruijn アーキテクチャとは有向 de Bruijn グラフの頂点に結合用マルチポート共有メモリを配置し、グラフの辺にプロセッサエレメントを配置するというものである。2 つの頂点間で有向辺が 2 本存在する可能性があるがその場合においても、プロセッサを各辺に配置することとし、その時の有向辺の方向はルーティングの際に意味を持ってくる。

逆 de Bruijn アーキテクチャの 3 次元の例を図 8 に示す。この例では、プロセッサ数:16、結合用マルチポート共有メモリの数:8 となる。3 次元の逆 de Bruijn アーキテクチャを採用した de Bruijn 型プロセッサの写真を図 9 にしめす。

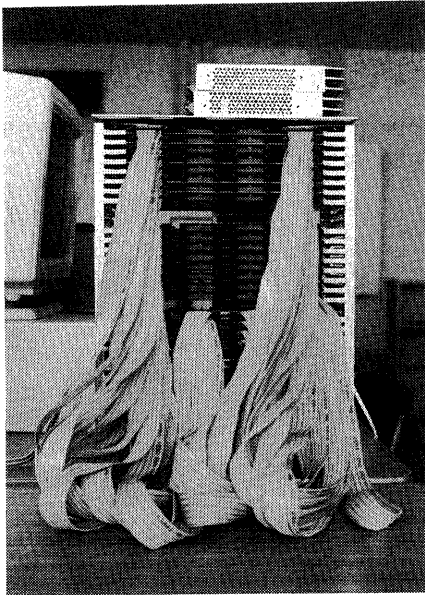


図 9: 3次元逆 de Bruijn 型プロセッサ

4.2 de Bruijn 型プロセッサにおけるプロセッサ間通信

de Bruijn 型プロセッサでもパケットルーティングを採用した。パケットは固定長であるが前の節で述べたように、ルーティングするソフトウェアより上位に位置するソフトウェアがデータを複数のパケットに分割して送り受け取った時に集約すれば問題ないと考えている。

de Bruijn 型プロセッサでは基本となった de Bruijn グラフの辺の方向にパケットを流すことにする。これは、ソフトウェアによってパケットルーティングを行わねばならないために、逆方向にパケットをルーティングするかどうかを判定するのにコストがかかると判断したためである。その理由として、逆方向へルーティングした場合が良いと思える場合がそう多くなかったことと、その場合順方向にルーティングしても通過するプロセッサの数は高々 N 程度であったことがあげられる。さらに有向辺の方向にしかパケットを流さないためマルチポート共有メモリを2つに分けるだけでよく、それぞれ

を頂点から出る辺に配置されたプロセッサに管理させることができ各プロセッサは1領域の書き込み管理だけで FIFO としてマルチポート共有メモリが使用できるという理由もあげることができる。パケットルーティングの方法についても発信元のプロセッサの配置されている有向辺が入る頂点、パケットの届先のプロセッサがある有向辺に出ていく頂点、この2つの頂点の間の問題に置き換えることができ既知のアルゴリズム [17] の適用によって実現が可能であることも大きな利点である。

5 おわりに

並列プロセッサ MC1 は通信速度や扱いやすさから実用的であるとは言いがたく、そういった意味ではプロトタイプ機である。プロセッサ間通信はすべてソフトウェアが実現しており、メモリに対するアクセス速度と比較して通信速度が低くなってしまった。そのため、今後並列プロセッサを製作する機会があれば、専用通信プロセッサと演算プロセッサを組み合わせ、各プロセッサに専用のディスクと大容量の主記憶を持たせた並列プロセッサを製作したいと思っている。

我々の研究室では、1978年以來、一貫してマルチプロセッサ製作にあたっては、1号機 AKOVST[8][9][13]、2号機 UNIP[9][11]ともメモリ結合を用いてきている。今回の3号機 MC1[16]においてもそれを踏襲したわけである。メモリ結合に関していえば、デュアルポートでは古くからもちいられているが [14]、マルチポートとしてはごく最近である [15]。キャッシュ全盛の現在であるが、デバイスサイドの支援とともにマルチポート共有メモリの発展に努力したいと考えている [18]。

参考文献

- [1] 阿江 忠, 藤田 聡, 相原 玲二, 山中 太市郎, 酒居 敬一: “光インターコネクション向きメモリ結合型超並列プロセッサアーキテクチャ”, 信学技報, CPSY 92-25, pp.41-46, (August 1992).
- [2] 林 巖雄, 阿江 忠, 小柳 光正: “光インターコネクション”, 通信学会誌, 75, 9, (1992).

- [3] 阿江 忠：新しい計算機アーキテクチャ (飯塚 肇編), 第4章, 丸善, (1990).
- [4] M.Koyanagi, H.Tanaka, H.Mori and J.Iba : "Design of 4kbit \times 4Layer Optically Coupled Three-Dimensional Common Memory for Parallel Processor System", IEEE J.Solid State Circuits, 25, 1, pp.109-116, (1990).
- [5] 小柳 光正, 広瀬 全孝, 阿江 忠 : "3次元光結合共有メモリを用いた並列処理コンピュータシステム", オプトロニクス, No.126, pp.76-82, (June 1992).
- [6] 阿江 忠, 相原 玲二 : "マルチプロセッサシステムのための光結合共有メモリ", 第28回情報処理学会全国大会, 3C-7, (1984).
- [7] 阿江 忠 : *VLSI* コンピュータ, 第5章, 電子情報通信学会, (1988).
- [8] 阿江 忠, 高橋 浩一, 松本 健治 : "共有メモリ結合によるマルチマイクロプロセッサの並列動作について", 電子通信学会論文誌, Vol.J65-D, No.3, (1982).
- [9] T.Ae and R.Aibara : "Experimentation and Analysis of Multiprocessor Systems", Proc.IEEE Real-Time Systems Symposium, L.A., pp.69-80, (1982).
- [10] T.Ae, S.Tenma, H.Yamasaki and M.Kitagawa : "A Distributed Real-Time Processing Language on Multimicroprocessor System", Proc.IEEE Real-Time Systems Symposium, Washington D.C., pp.20-29, (1983).
- [11] 相原 玲二, 阿江 忠 : "マルチマイクロプロセッサによるソート/サーチエンジンの試作", 情報処理学会論文誌, Vol.26, No.2, pp.349-355, (March 1985).
- [12] T.Ae et al. : "Hypercube is better than De Bruijn for Connectionist", 5th ISMM Int.Conf. on Parallel and Distributed Computing and Systems, Pittsburgh, Oct.1-3, (1992).
- [13] 阿江, 大崎, Vuong : "小規模マルチマイクロプロセッサシステムの一方式", 電子通信学会電子計算機研究会資料 EC78-35, (1978).
- [14] 星野 力 : *PAX* コンピュータ—高並列処理と科学計算—, オーム社, (昭和 60).
- [15] Daniel Litaize, Abdelaziz Mzoughi, Christine Rochange, Pascal Sainrat : "Towards a Shared-memory Massively Parallel Multiprocessor", 19th. annual ISCA, pp.70-79, (1992).
- [16] 酒居 敬一, 藤田 聡, 相原 玲二, 阿江 忠 : "メモリ結合型マルチプロセッサ MC1 について", IPSJ 93-ARC-101-14, (August 1993).
- [17] F.Thomson Leighton : *Introduction to Algorithms and Architectures : Arrays · Trees · Hypercubes*, Morgan Kaufmann publishers, chapter 3, (1992).
- [18] T.Ae, S.Fujita, T.Ando, K.Sakai and Y.Fujihara : "Optical Interconnection and Massively Parallel Processing", Proc. 1st Inter. Workshop on Massively Parallel Processing Using Optical Interconnections, Cancun, pp.118-132, (1994).