計算機アーキテクチャ　101－13
（１９９３．　８．　２０）

実時間処理用共有メモリ

藤原 寛†，　藤田 聡†，　相原 玲二††，　阿江 忠†
†広島大学工学部第２類（電気系）
††広島大学集積化システム研究センター

ハードな実時間処理を要求されるシステムにおいては、マルチプロセッサシステムを利用して各々の処理を分散させることによって多くの利点が得られる。プロセッサ間の結合方法には、バス結合、共有メモリ結合、多段スイッチ結合などがあるが、リアルタイムシステムの構成においてはプログラミングの容易さ、データ転送速度の高速性などの点において、共有メモリ結合が適す。そこで本稿では、共有メモリ結合型マルチプロセッサを構成するための新しい、マルチポートメモリの構成方法を提案する。

Multiport Shared Memory for Real-time Processing

Yutaka Fujihara†, Satoshi Fujita†, Reiji Aibara††, Tadashi Ae†
†Electrical Engineering,Faculty of Engineering,Hiroshima University
†Research Center for Integrated Systems,Hiroshima University

In the high performance system for hard real-time processing, there are many advantages by using multiprocessor system and distributing many tasks on each processor.

There are many ways of connection between processors, e.g., Bus connection, Memory connection, and Multi stage switch connection. In real-time processing system, Memory connected multiprocessor is better than all, because it makes the programming easier, data tansfer faster, and so on.

In this report, we propose a new multiport shared memory architecture makes more processor connect to each other.

1 はじめに

ハードな実時間処理を要求されるシステムにおいては、数10〜数100程度の複数のプロセッサを結合して各プロセッサに処理を分散させることによりプログラミングを容易にしたり負荷を分散させることができるといった利点が生まれる。このようなことから実時間処理用のシステムにおいては複数のプロセッサを結合させることが有用であるが結合は密でなければならない。

この場合プロセッサ同士を結合するには数々の方法がある。例えば、代表的なものとしてバスを使ったバス結合方式、マルチポートメモリを使ったメモリ結合方式、多段スイッチを使った多段スイッチ方式などであるが、実時間処理の用途においては、比較的プログラミングが容易であり、かつ、他の方式と比べてデータ転送速度が高速なメモリ結合型が適当であると思われる。

そこで本稿では、より多くのプロセッサを共有メモリによって結合させるために既存のマルチポートメモリを組み合わせることによってさらに多くのポート数を持つような共有メモリのアーキテクチュアを提案し、評価、検討する。

2 共有メモリの実現方法

マルチプロセッサ間で共通して使用する共有メモリの実現形態としては、以下に示すようないくつかの方法がある。

バス結合型：もっとも簡単に各プロセッサと共有メモリの間を結合できるのが、図1に示すようなバス結合方法である。この方式は、シングルプロセッサの場合の自然な拡張になっているが、一時刻には一つのプロセッサしかバスを占有できないために、時分割して複数のプロセッサが使用する。このため通信量が増えるとバスが空くのを待つ時間が増えるので、プロセッサ数がある程度多くない場合に向いている。しかし一対多へ通信する放送 (broadcast) は簡単に行なうことができるという特徴を持っている。

マルチポートメモリ型：マルチポートメモリ型は、図2に示すようにプロセッサがマルチポートメモリに各々直接につながり、そのマルチポートメモリを共有メモリとする方法である。マルチポートメモリを使用するとメモリへの書き込みと読み出しの時に衝突が起こるので、これを回避させる手段を講じなければならない。多数同時書き込み/読み込みのものが望ましいが多数書き込みの場合には、同一アドレスへの書き込みは一時刻には一つのプロセッサに限られるのが普通である。

クロスポイント型：図3に示すように、$n \times n$のスイッチマトリクスを使ってn対nのプロセッサ、共有メモリ間結合ネットワークを構成する方式である。この場合、n対nの結合ネットワークはそれぞれの入力から所望の出力へ直接結合できるように、交点 (crosspoint) に接続のためのスイッチが設けられる。図3では、各入力に対して一つの出力が選択される例を示している。

多段ステージネットワーク型：クロスポイント型の発展形として多段ステージネットワーク型がある。これは、図4に示すようにクロスポイント型結合ネットワークを多段ステージに分けて実現する方法である。

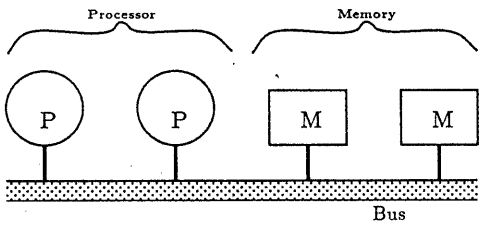


図 1: Bus connection

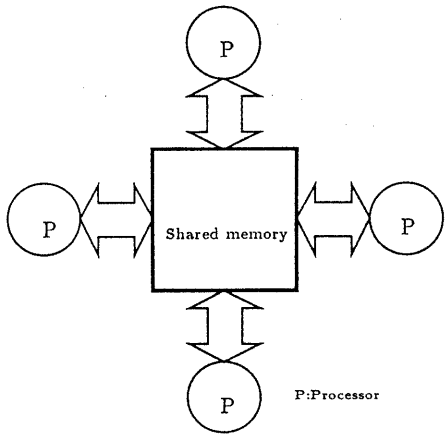


図 2: Shared memory connection

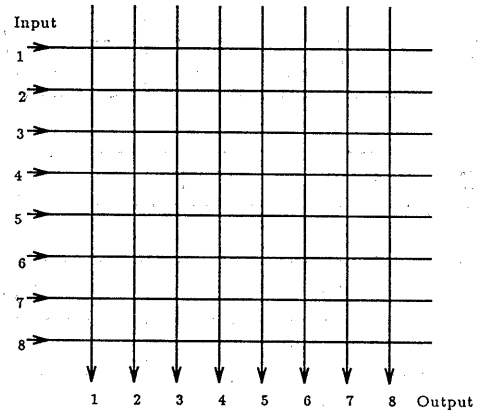


図 3: Cross point connection

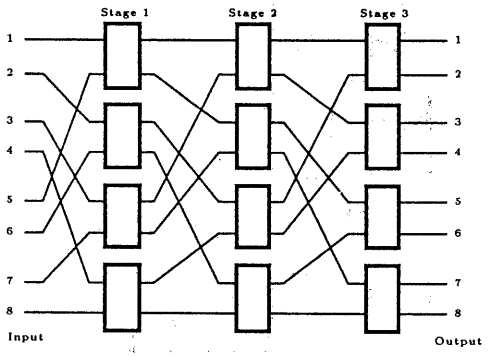


図 4: Multistage Network

3 kポート共有メモリからk^2ポート共有メモリへの拡張

3.1 構成方法

本稿では、共有メモリ結合型のマルチプロセッサを構成する場合に使用する新しいマルチポートメモリのアーキテクチャを提案する。光結合三次元ＩＣ技術によるマルチポートメモリ (以下光三次元ＩＣとする) を使用したものと、既存のマルチポートメモリを使用したものを二種類を考察するが、既存のマルチポートメモリを使用した方は光三次元ＩＣが実現するまでの間の既存の技術を使った代替方法である。以下にその構成方法を示す。

3.1.1 光結合三次元ＩＣによる方法

本稿で提案する共有メモリの構成方法のひとつは光結合三次元ＩＣ技術を使用したマルチポートメモリを使う方法である。この方法は図5に示すようにそれぞれのマルチポートメモリを列に見立てて配置し、各々のメモリのポート間を光配線技術などによって結合する。このような方法で複数のマルチポートメモリを結合することにより基となる光三次元ＩＣが持つポート数の２乗のポート数を持つマルチポート共有メモリが新たに構成できる。

3.1.2 既存のマルチポートメモリによる方法

ふたつめの構成方法は、現在光三次元ＩＣはまだ開発中であり入手できないので、かわりに既存のマルチポートメモリを使用する方法である。ここで既存のマルチポートメモリとは、現在あるデュアルポートメモリ、またはその他のメモリを組み合わせることによって疑似的に構成されたマルチポートメモリのことを指す。この方法は図6に示すようにそれぞれのマルチポートメモリを行と列に見立てて交差状に組み合わせ、各々の交差点を図7に示すようなインターフェース (プロセッサも同時に結合する) によって結合する。そしてそれぞれの交差点を一つのポートと見ることによりマルチポートメモリを構成する。このような方法で複数のマルチポートメモリを結合することにより基となるマルチポートメモリが持つポート数の２乗のポート数を持つマルチポート共有メモリが新たに構成できる。なおこの場合は各行および列単位では常に同じ内容を持つバンクメモリとして使用する。このように構成した場合には行・列はそれぞれ対象なものとみなすことができる。

3.2 構成方法の発展

前節のような構成方法を考えた場合、その発展として以下のようにポート数を増やしていくことが考えられる。前節のようにして構成したマルチポートメモリを一つのコンポーネントとして考える。そしてそのコンポーネントを新たに一つのマルチポートメモリと見て、さらに前節と同じ構成法を使って再帰的に構成することが可能である。このような再帰的構成を繰り返す回数を次元と呼ぶことにする。以上のようにして構成した場合、もととなるマルチポートメモリのポート数をk、次元dとした時の総ポート数を表１に示す。ただし実時間処理への応用を考えた場合には、データ転送時間の遅れなどを考慮に入れると次元数２の時の構成を行なうのが適当で

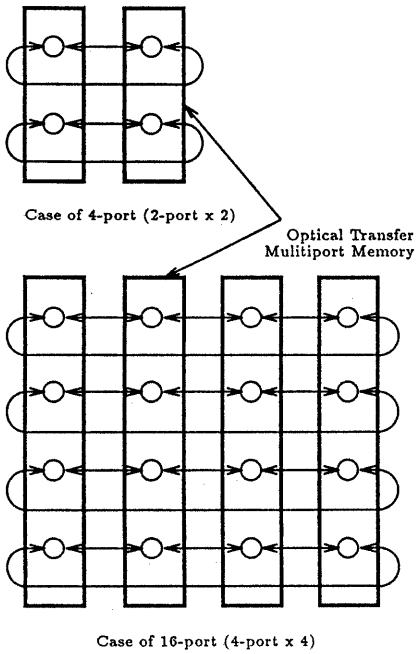


図 5: Optical transfer Multiport Memory

あると思われる。そこで本稿では次元数 2 の場合についてのみの評価を行なった。

表 1: ポート数 k, 次元数 d のときの総ポート数

k \ d	2	3	4
2	4	16	256
4	16	256	65,536
8	64	4,096	16,777,216
16	256	65,536	2^{32}

4 転送方法

前節のようにして、マルチポート共有メモリを構成したとき、共有メモリとして機能させるためには光三次元ＩＣを用いた場合には各列ごと、既存のマルチポートメモリを用いた場合には各バンクごと (行、列ごと) のメモリに蓄えられているデータが各々一致しなければならない。そのための手段として本稿のマルチポートメモリの構成では以下のようにして各々のメモリの内容を転送することによって、その中にあるデータを一致させるこ

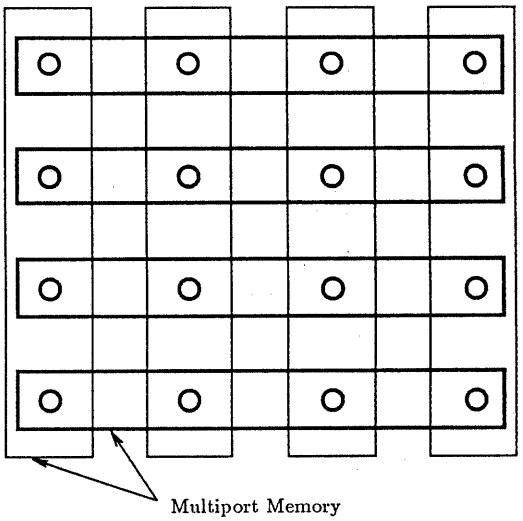


図 6: Composition of Multiport Memory (Memory transfer)

とにする。

4.1 光結合三次元ＩＣによる方法

1. まず書き込みが起こると同じ光三次元ＩＣの中では光技術によって、書き込まれた内容が自動的にすべての層へ転送される。
2. 次に各々の光三次元ＩＣの間でメモリの内容を一致させるために横一行のポートごとに転送を行なう。

以上のような手順で転送を完了する。

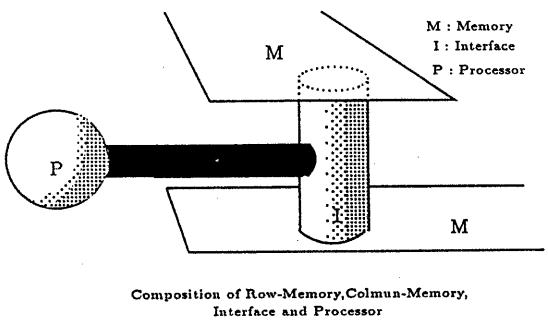


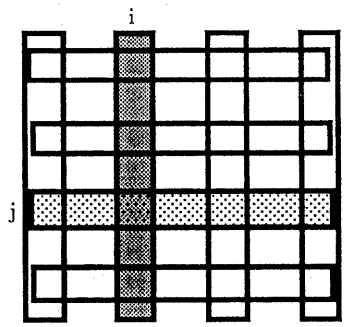
図 7: Interface between Memory and Memory

4.2 既存のマルチポートメモリによる方法

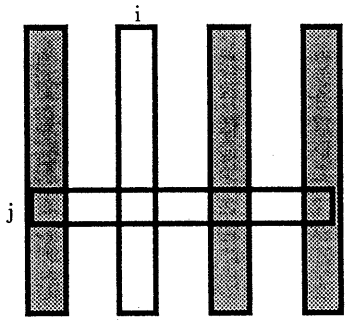
1. まず書き込み要求が起こるとそのデータはインターフェースによって実際に書き込む方のバンク及び同じ交差点上にあるもう一方のバンクのワークエリアにも同時に書き込まれる。図8では (a) の手順ににあたる。

2. 次に書き込みの起こった交差点とおなじ行または列に接続しているインターフェースはワーキングエリアに書き込みのあった方のバンクからもう一方のバンクへそのワーキングエリアの内容を転送する。図8では (b) の手順にあたる。

以上のような手順で転送を行なうことにより、それぞれの行にあたるバンクどうし、列にあたるバンクどうしは同様の内容を持つようになる。



(a) Processor (j,i) writes through on Column i at t=1



(b) All Interface Units(except Unit on Column i)on Row j write the transfer data on all Columns except Column i at t=2

図 8: Transfer Sequence(Memory Tranfer)

5 シミュレーション

5.1 シミュレーション方法

前章で提案した実時間処理用マルチポート共有メモリでは、書き込み時の衝突がある程度発生すると思われる。この衝突が発生することによって、全く衝突が生じないようなシングルポートのメモリなどより性能が低下すると思われる。そこで本稿では、この性能低下がどの程度のものか以下のような評価実験を行なった。評価実験はワークステーション上でC言語によるシミュレータを作成して行なった。シミュレータは、ある一定のサイクルの中でどの程度共有メモリへのアクセスが起こり、そのアクセスが読み込みであるか書き込みであるかなどを決定してそれぞれのサイクルにおいてアクセスの衝突が発生するか否かをシミュレートしている。評価基準としては、全く衝突の起こらない場合を1とした時の性能の低下割合を採用した。

5.2 評価シミュレーション

- 光転送の場合

  1. ノード数 4〜256、Read : Write 比 15:1
     アクセス確率 1/2〜1/128
  2. ノード数 16、Read : Write 比 1:1 〜15:1
     アクセス確率 1/2〜1/128

- メモリ転送の場合

  1. ノード数 4〜256、Read : Write 比 15:1
     アクセス確率 1/2〜1/128
  2. ノード数 16、Read : Write 比 1:1 〜15:1
     アクセス確率 1/2〜1/128

以上4つの場合を想定してシミュレーションを行なった。なお1回の繰り返しごとに各プロセッサごとの平均値をとり、それをそれぞれ100回繰り返し、その1回ごとの平均値の100回での平均をとった値を最終的なスピード値とした。

5.3 シミュレーション結果

5.3.1 光結合三次元ICによる構成の場合

シミュレーションの結果以下の図9,10に示すような値が得られた。

5.3.2 既存のマルチポートメモリーによる構成の場合

シミュレーションの結果以下の図11,12に示すような値が得られた。

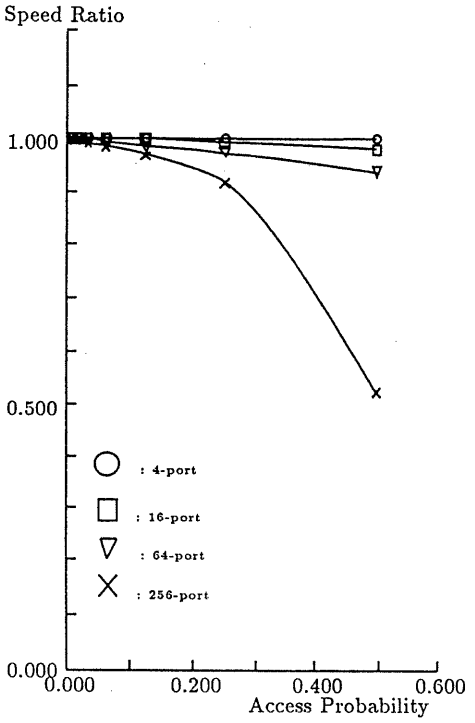


図 9: Optical Transfer (Read:Write = 15:1)

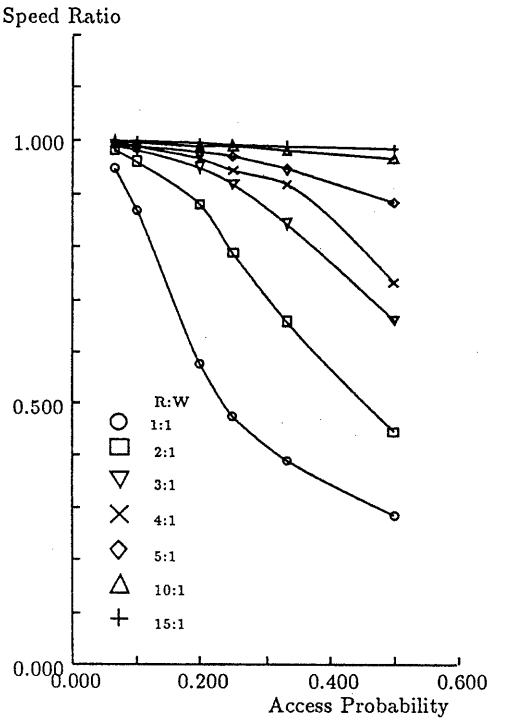


図 10: Optical Transfer (16-Port)

5.4 考察

シミュレーション結果を見ると、ポート数の観点からいえばメモリによる転送の場合にはポート数が 64 以上になると急速に速度が低下していることがわかる。このことからメモリによる転送の場合にはポート数 16 程度かそれ以下の構成での使用が望ましい。ただしある程度アクセスの頻度が低い場合、すなわちアクセス確率の小さい時 (例えばアクセス確率 0.1 以下程度の場合) には、64 ポートの構成でもある程度の速度が期待できるのでこのような場合には使用可能である。

光三次元ＩＣによる構成の場合は、4 ポートから 64 ポート程度までは目立った速度の低下が見られない。4 ポート、16 ポートの場合にはほとんど衝突の無い場合と大差ない。256 ポートの場合には多少速度の低下が見られるがメモリによる転送の場合ほど急激な低下は見られない。これらのことから、光三次元ＩＣを用いる場合には 256 ポートの構成でも十分実用に耐えうる。

読み込み対書き込み比の観点で見ると、メモリによる転送の場合には 15:1 のあたりからすでにそうとうのスピード低下が見られる。しかし実際の実時間処理応用においては読み出しに比べてあまり多く書き込みが起こることは少ないと思われるので、この程度のスピード低下でも十分実用に耐えうると思われる。

読み込み対書き込み比の観点で見ると、光三次元ＩＣによる構成の場合には、メモリによる転送の場合ほど急速な速度の低下は見られない。もっとも激しく書き込みが起こる、読み込み対書き込み= 1:1 の場合でもメモリによる転送の場合の 15:1 程度のスピードが見られる。この程度のスピードならば、激しく書き込みが起こるような場合の応用においても十分に使用に耐えうる。

全体としてみると、本稿で提案したマルチポート共有メモリの構成方法はメモリによる転送の場合の構成方法でも 16 ポート構成で、あまり激しいアクセスのない場合には非常に有効なマルチポートメモリの構成方法であるといえる。また、光三次元ＩＣを使った場合にはかなりの高性能が期待できるものと思われる。

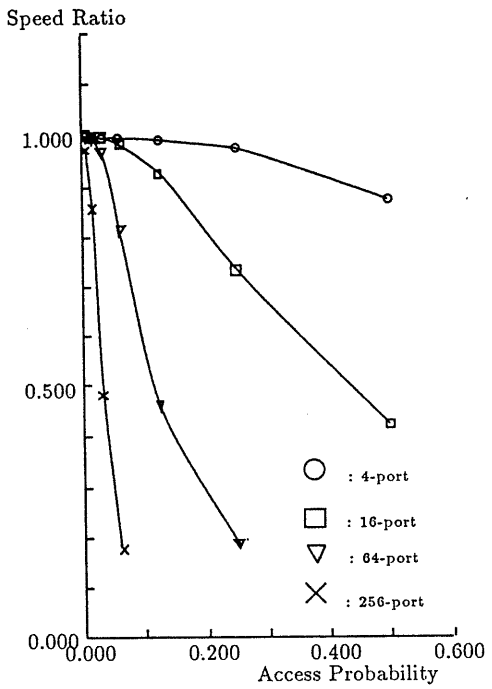


図 11: Memory Transfer (Read:Write = 15:1)

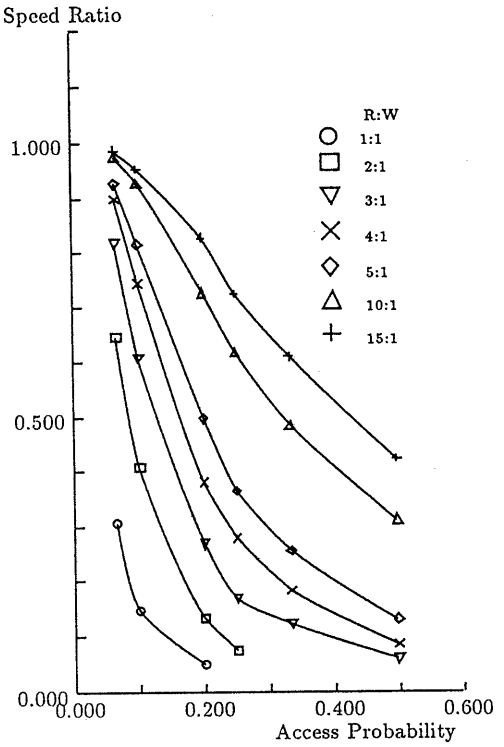


図 12: Memory Transfer (16-Port)

6 おわりに

光結合三次元ＩＣの使用によって、本稿で提案したマルチポート共有メモリの性能は飛躍的に向上するものと思われるので光三次元ＩＣが早期に実現することが望まれる。また、2次元以上の構成方法については評価を行なっていないが今後は2次元以上の場合の評価も行なって最適な構成方法を決定することが重要であると思われる。

参考文献

[1] 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”, 電子情報通信学会, J65-D, 3, 1982.

[2] T.Ae and R.Aibara, “Experimentation and Analysis of Multiprocessor Systems,” Proc. IEEE Real-Time Systems Symposium, L.A., pp.69-80, 1982.

[3] T.Ae, S.Tenma, H.Yamasaki and M.Kitagawa, “A Distributed Real-Time Processing Lanuage on Multimicroprocessor System,”Proc.IEEE Real-Time Systems Sympsium, Washington D.C., pp.20-29, 1983.

[4] 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.

[5] 広瀬 全孝,“電子光共存型集積システムの探求”, 通信学会誌,74,4,pp.407-413,1992.

[6] 林 巌雄, 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.

[7] M.Koyanagi, H.Tanaka, H.Mori and J.Iba, “Design of 4kbit × 4Layer Optically Coupled Three-Dimentional Common Memory for Parallel Processor System,”IEEE J.Solid State Circuits, 25, 1, pp.109-116, 1990.

[8] 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,126, pp.76-82, June 1992.

[9] J. A. Stankovic et al., "SpringNet : A Scalable Architecture For High Performance, Predictable, and Distributed Real-Time Computing,"IEEE Workshop on Real-Time Oparating System, 1991.

[10] D.Mosberger, "Memory Consistency Models," ACM Operating Systems Review, 27, 1, pp.18-26, Jan.1993

[11] 村岡洋一他, 超並列コンピュータ入門, オーム社, 1992.

[12] 阿江忠, ＶＬＳＩコンピュータ, 電子情報通信学会,1988.

[13] T.Ae et al., "Hypercube is better than De Bruijn for Connectionist,"Proc. 5th ISMM Int.Conf. on Parallel and Distributed Computing and Systems, Pittsburgh, pp.370-372, Oct. 1992.

[14] T.Ae, R.Aibara and S.Fujita, "Non-Protocol Shared Memory for Real-Time Systems,"Proc. International Workshop on Parallel and Distributed Real-Time Systems, Newport Beach, pp.159-163, April 1993.

[15] 阿江, 藤田, 相原, 山中, 酒居,"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", 電子情報通信学会技術報告, CPSY92-25, pp.41-46, 1992.

[16] 阿江 忠,"欧米における実時間システムの研究動向(招待講演)", 第 1回実時間システムワークショップ, CPSY91-71, 1992.