動的負荷分散を行う相互結合網

Interconnection Network with
Dynamic Load Balancing Facility

坂井 修一□　小池 汎平□　田中 英彦□　元岡 達□
Shuichi Sakai　Hanpei Koike　Hidehiko Tanaka　Tohru Moto-oka

□ 東京大学 工学部
Faculty of Engineering, The University of Tokyo

1．まえがき

ＶＬＳＩ技術の発達とともに高並列計算機（数十～数千台）への期待が高まっている。高並列計算機の実現にあたっては、（１）解くべき問題に内在する並列性の抽出、（２）それぞれのプロセッサへの効率的な負荷の分配、（３）プロセッサ間の高速なデータ転送、などが問題である。本稿は、（３）を考慮した（２）の実現法、すなわち、通信オーバヘッドを低く抑えながら、効率良い負荷分散を実現する方法について述べる。

負荷分散は、コンパイラなどによってタスク実行以前にあらかじめプロセッサの割付けが決められる静的な方式と、タスク生成時に決められる動的な方式に大別される[1～3]。定型的な問題を処理するＳＩＭＤ型の計算機では前者で十分であるが、データフローマシンやリダクションマシンなど、非定型的な問題を処理するＭＩＭＤ型の計算機では、後者が必要とされる[4～10]。

現在までに提案されている並列処理計算機の動的負荷分散の方法として＊、

（１） 全システムに１つ、またはある範囲のプロセッサ群に１つ、負荷制御装置を置き、その時点での負荷の量が最少のプロセッサにタスクを割付ける（集中制御）[6]。

（２） 木状に配置された「通信および負荷分散ノード」が、葉ノードのプロセッサ群の負荷を監視し、不均衡が生じたときにタスクの再配置を行う[4]。

（３） リングバス上を移動しているタスクを、空きプロセッサが次々に獲得し、処理する[5]。

（４） タスクの転送を行う相互結合網に、行先の負荷の情報をフィードバックする信号線を設け、これを使って行先プロセッサを自動的に決定する[8,9]。

などがある。このうち、（１）は、負荷情報の管理のオーバヘッド（通信および更新のオーバヘッド）の点から、集中制御可能なプロセッサ台数に限界があると考えられる。負荷の制御を階層的に行う（２）では、再配置時に大量のデータの移動が必要になるという問題点、（３）では、リングバスの容量からタスク分配の速度が制限されるという問題点が予想される。

一方、（４）の方式は、負荷分散の管理のオーバヘッドが小さく、多段結合網などの局所性のない網構成を採った場合、各プロセッサがグローバルな負荷情報を得ることができ、同時に高い転送スループットを保証することができる。

本稿では、（４）の負荷分散を実現する相互結合網（負荷分散適応型網と呼ぶ）の提案と、シミュレーション評価による検証に関して述べる。最初に、負荷分散適応型網の構成要素となるスイッチング・ユニット（ＳＵ）の設計を２種示し、動作速度・ハードウェア量の評価を行う（２章）。次に、これらのＳＵを用いた相互結合網の特徴を述べ、具体的な網（オメガ網）を想定した負荷分散のシミュレーション評価を行う（３章）。さらに、提案したいくつかの方式を比較・検討し、また、実際のマシンへの適用を考察する（４章）。最後に、今後の課題を列挙する（５章）。

2．動的負荷分散制御を行うＳＵ

2．1 設計方針

負荷分散適応型網のＳＵには、タスクの転送と逆の向きに行先プロセッサの負荷の情報を伝送し、負荷の量が最少の行先を選択してそこにタスクを送りつける機能が必要である。ＳＵの設計方針を以下に列挙する。

＊ 乱数分散方式、巡回分散方式[1]は、特別の制御を行っていないことから除外してある。

（１）　負荷分散向けの転送モードと、行先を指定する通常の通信向けの転送モードをもつ。モードの切り替えはポート単位で行う。

（２）　どちらのモードでも、マルチキャストができる。

（３）　多段結合網（マルチパスの網を含む）を対象とする。

（４）　網内のクロック・スキューを考慮して、ＳＵ内は同期制御、ＳＵ間は非同期制御とする。

（５）　回線交換方式を採る。データ転送は、同期でも非同期でも行える。

以下の節で、ＳＵの設計例を２種示す。

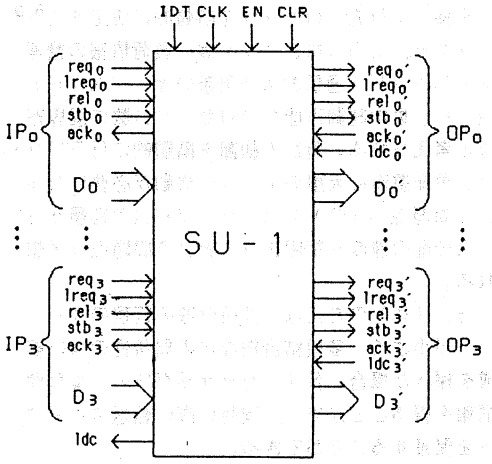

図１　ＳＵ－１の全体構成

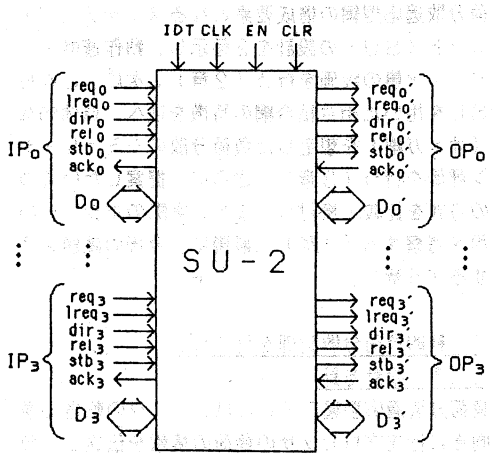

図２　ＳＵ－２の全体構成

２．２　ＳＵ－１　：　タスク要求線方式（図１）

４つの入力ポート（IP_0 ～IP_3 ）と、４つの出力ポート（OP_0 ～OP_3 ）より成る、単方向のクロスバスイッチ（データ線は$D_i^{(l)}$：８ビット幅）である。制御線としては通常見られるreq,ack,stb,rel の他に、lreq（モード指定）とIdc （負荷受入れ可）が設けられている。前者は転送モードの指定（２．１節）を行い、後者は行先に負荷を受入れることができるプロセッサが存在するかどうかを示す。

ＩＰ側にはルーティング・コントローラがあり、モードに応じてＯＰに転送要求を出す。これを受けたＯＰ側は、アービトレーションを行って、適当なゲート制御信号を立て、次段のＳＵに転送要求を出す。

行先を指定するモードでは、ソース側はreq を立てると同時にデータ線に行先プロセッサの番地を出力し、各段のＳＵはこの情報をもとにルーティングを行う。負荷分散モードの場合、各段のＳＵは Idc' の立っている任意の空きＯＰを選択して経路を設定する。

ＳＵ－１では、各プロセッサがIdc を立てる負荷量の閾値設定が問題となる。一般に、セントラル・コントローラによる負荷分散の監視（厳密に行う必要はない）が必要な場合があると考えられる。

２．３　ＳＵ－２　：　最少負荷量方式（図２）

ＳＵ－１同様の４入力４出力クロスバ・スイッチであるが、双方向（半二重）の通信を行う点、Idc の線がない点が異なっている。通信の向きは、制御線dir の信号で決まる。使用中でないデータ線には、その時点で閉塞なしに行けるプロセッサ群のうち、最も負荷の軽いものの負荷量が、逆方向に流れている。これは、各ＳＵが４入力の比較器をもち、後段よりの負荷の情報のうちで最も小さい値のものを前段に伝送することで実現される。

転送の手順はＳＵ－１とほぼ同じである。負荷分散モードの経路設定は、各段のＳＵで、最少負荷量を示す空きＯＰを選択することで行われる。

ＳＵ－２によるデータ転送のスナップショットを図３に示す。

ＳＵ－２では、負荷量の閾値設定を行う必要がなく、セントラル・コントローラを設けなくてよい。

本ＳＵでは、半二重通信を行うため、内部の接続線数が多くなる難点がある。そこで、これを４ビットスライス化し並列接続する方式を考え、改良設計を行った。以下、これをＳＵ－２b と呼ぶ。並列接続されるＳＵ－２b は、１つが制御専用、他は単なるマルチプレクサとして動作する（ハードウェア構

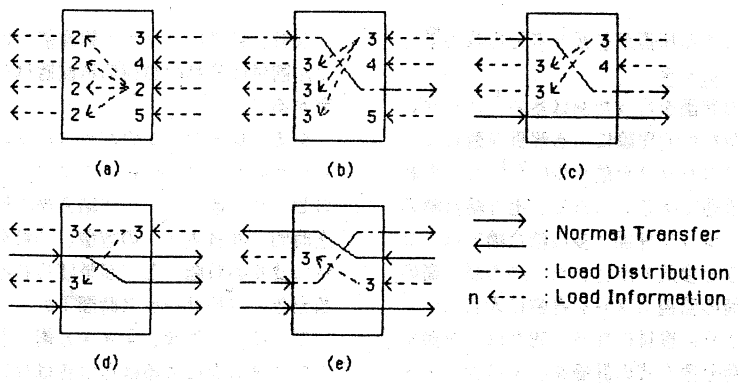

図3　データ転送のスナップショット（SU－2）

成は同じ）。

2．4　各SUのハードウェア量

STTLでSU－1、SU－2、SU－2b を設計したときの総ハードウェア量を、表1に記す。内部接続線数の項から、LSI化を考えた場合、SU－2はビットスライス化が有利であると推察される。

2．5　各SUの動作速度

各SUの動作速度（STTLを用いて設計した場合の最大遅延）を、表2に示す。

3．負荷分散網の構成とシミュレーション評価

3．1　網構成

前章のSUより成る負荷分散適応型網を用いることの利点を以下に列挙する。

（1）　均等な負荷の分配が実現される。

（2）　負荷分散制御のための、ハードウェア的及び、時間的なオーバヘッドが小さい。特に、セントラル・コントローラによる監視が、大雑把で良い（SU－1）か、不必要になる（SU－2）。

（3）　網内の閉塞を避けつつタスクの分配を行うことによって、通信時間が短縮される。これは、タスクの転送時間と処理時間が同程度であるとき、特に有効である。

網トポロジーは、セルフ・ルーティングの可能な多段結合網を対象とする。オメガ網（図4）[12]と位相的に等しい網[11,13,14]、その冗長構成[15〜17]、ガンマ網[18]などがこれである。超立方体網・CCC網・格子型網などへの適用には、行先指定モードのルーティン

表1　各SUのハードウェア量

	SU－1	SU－2	SU－2b
総ゲート数	1225	1740	1544
入出力線数	115	118	94
内部バス本数	108	140	108
$IP_{0\sim3}$内部バス接続線数	120	284	188
$OP_{0\sim3}$内部バス接続線数	228	260	180

表2　各SUの動作速度

	SU－1	SU－2	SU－2b
行先指定の経路設定	159	189	195
負荷分散の経路設定	157	152	152
リリース	115	115	115
ストローブ	12	12	12
アクノリッジ	12	12	12
負荷情報の更新	11	177	141
データ転送（順方向）	19	24	24
データ転送（逆方向）		31	31

STTLを用いた最大遅延　単位はナノセカンド［ns］

グを表引き方式にする拡張が必要と考えられる[19]。

3．2　シミュレータ

負荷分散方式の評価を行った例は多い[1]。しかし、動的負荷分散は待ち行列理論による解析が難しく、解析のときは、プロセッサ台数が少い[2,3]などの制約をつけるのが普通である。一方で、相互結合網の解析・シミュレーション評価（多段結合網では、文献13）14）17）など参照）が行われているが、網の通信コスト・負荷の監視コストを考慮に入れれば、負荷分散問題はさらに複雑になる。我々は、前節までで提案した負荷分散方式の評価を、イベント・ドリブン・シミュレーションによって行った。シミュレータは、プロセス管理機構を付加したC言語で書かれている。

シミュレーション・モデルを図5に示す。本稿では、系がクローズド・システムで、定常的な動作をする場合を扱う。求める値は、次の3種である。

（1）　正規化スループット　：　平均タスク処理時間のうちに、1つのプロセッサが処理し終えるタスク数の平均値。全プロセッサがいつも稼動中であれば1となり、平均プロセッサ稼動率とみなすことができる。

（2）　タスクの平均系内時間　：　タスクが生成されてから、プロセッサで処理が終わるまでの時間の平均値。

（3）　キュー長の標準偏差　：　負荷量のばらつきの指標。一定時間ごとに測定し、時間平均をとる。ただし、ここで「キュー長」とは、入力キューの中のタスク数と出力キューの中のタスク数の和のことである。

なお、各プロセッサの処理能力は同じとし、いったん割付けられた負荷の再配置は行わないことを仮定する。

シミュレーションのパラメータとしては、①プロセッサのサービス分布、②タスクの生成分布、③プロセッサ台数（N）、④結合網の形状、⑤タスク転送時間、⑥各キューの容量、⑦負荷情報の伝達・更新に要する時間、などが挙げられる。今回は、①指数分布（平均サービス時間Tp）、②1タスクの終了ごとに1つ生成、④オメガ網、⑤SU1段あたりTt（タスクによるばらつきはないとする）、⑥無限大、⑦0*としたものに関して報告する。

3．3　オメガ網での負荷分散シミュレーション

オメガ網（図4）を用いたシミュレーションに関して述べる。対象とした負荷分散方式は、以下の5種類である。

（1）　タスクの行先を乱数で与える乱数分散方式。**

（2）　負荷情報を集中管理し、現時点で最もキュー長の短いプロセッサをタスクの行先とする方式。

（3）　現時点で、網上の閉塞なしに到達することのできる任意のプロセッサをタスクの行先とする方式。

（4）　SU－1を用いる方式。負荷量の閾値は、

---

*今回は、1タスクの転送時間と比べて十分に小さい場合を想定した。

**ただし、タスクを生成したプロセッサのキューが空の場合、自分の入力キューに戻すことにする（EmptySelf[7]）。（2）～（5）も同様。

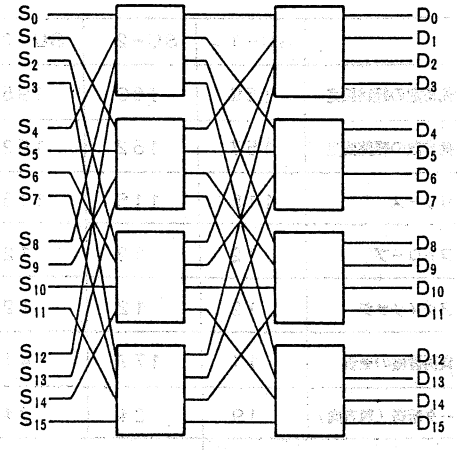

図4　オメガ網

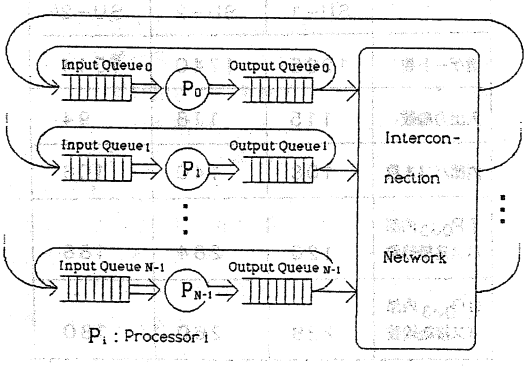

図5　シミュレーション・モデル

1プロセッサあたりのタスク数の平均にある値（ΔL）を加えたものとする。簡単な集中制御を仮定する。

（5） SU－2を用いる方式。集中制御をしない。

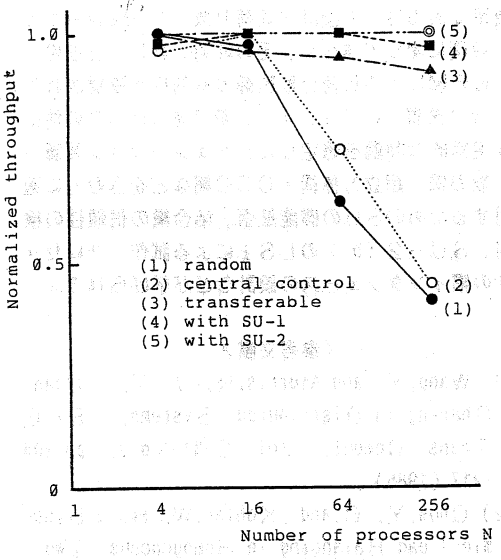

図6 オメガ網を用いた場合の正規化スループット

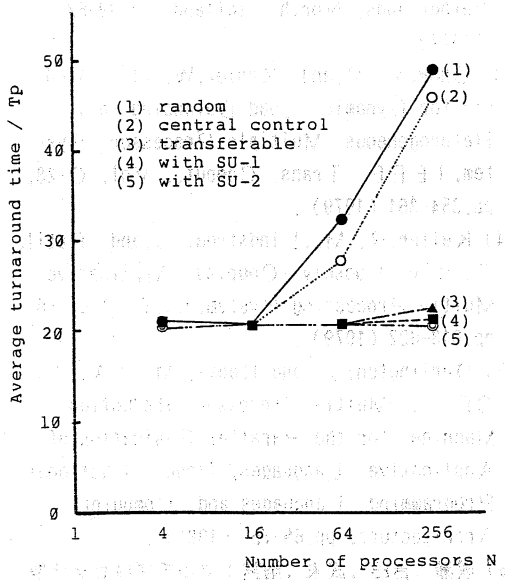

図7 オメガ網を用いた場合のタスクの系内時間

初期状態として、各プロセッサの出力キューに21個のタスクがあるとし、Tt／Tp = 0.25 、ΔL＝5と仮定した。

シミュレーションの結果を図6（正規化スループット）、図7（タスクの系内時間）、図8（キュー長の標準偏差）に示す。

4．考察

4．1 負荷分散方式の比較・検討

3．3節で得られたシミュレーション結果にもとづいて、動的負荷分散の方式を比較検討する。

計算機網の場合[1~3]と異なり、並列処理計算機では、個々のタスクの処理時間より全体のスループットが重視される。したがって、各プロセッサがほとんどいつも稼動状態となる方式（4）（5）は、方式（1）（2）より優れているといえる。この傾向は、プロセッサ数Nが大きくなるほど顕著である。スループットおよびタスクの平均系内時間に関しては、負荷を集中管理する方式（2）より、結合網の閉塞のみを考慮する方式（3）の方が良い結果を示している*（図6、図7）。

今回のシミュレーションでは、系全体の負荷の量

*これは、Tt／Tpが比較的大きいことからくる。ストリングリダクションマシン、手続きレベルのデータフローマシンなどが、これに相当する。

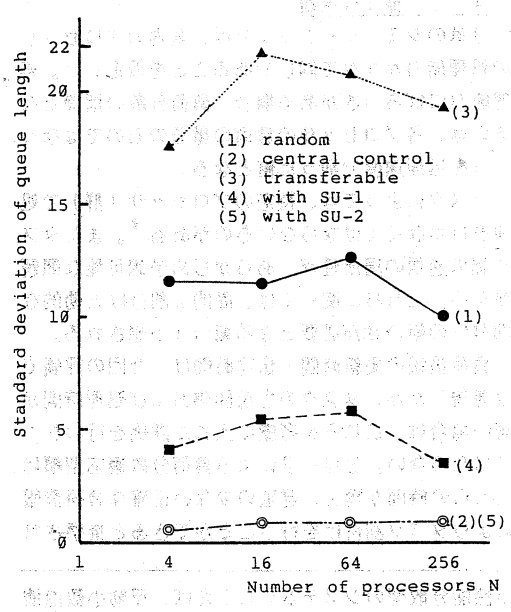

図8 オメガ網を用いた場合のキュー長の標準偏差

が一定の場合を扱った。現実に大きな並列度のある問題（ツリー状の探索問題など）では、①1つまたは少数の初期タスクから数多くのタスクが生成される段階、②生成される負荷の量が系の処理能力に対して飽和している段階、③タスクの生成率が減少し計算が収束していく段階、の順序で処理が進行すると考えられる。①では、全プロセッサをできる限りすばやく稼動状態にすること、②では各プロセッサの稼動率を高く保つこと、③では計算の収束にあたって各プロセッサの負荷の均等性を保持していくことが主たる問題点となる。図8より、方式（1）（3）では負荷の均等性が保持されず、③にかかる時間が大きくなる欠点があることが推察される。また、現実のプロセッサはキュー容量が有限であり、この点からも（1）（3）は不利である。なお、具体的な問題を想定した負荷分散の評価は、別途報告する予定である。

方式（4）（5）は、スループット、タスクの平均系内時間の2点で、他の方式より優れた値を示す。このうち、（4）は閾値設定の問題があり、負荷の量が変動するときに、これを更新する機構が必要となる。また、プロセッサ内の待ちタスク数のばらつきも、（5）のほうが小さい。

以上を総合して、SU－2によって構成される負荷分散適応型網を用いる動的負荷分散が、ここで採りあげた中でもっとも優れた方式であると結論づけられる。

4．2　様々な課題

3章のシミュレーションでは、系内のプロセッサの処理能力がすべて同じであることを仮定した。処理能力にばらつきがある場合、負荷分散の指標となるのは、各プロセッサの負荷の量そのものではなく、これを処理速度で割った値となる。

タスクによっては、特定のプロセッサ（群）で処理を行わなくてはならないものがある*。またタスク間の通信の局所性が、あらかじめ予想可能な問題がある。これらに関しては、静的な割付けと動的な割付けの組合せが必要となる場合が予想される。

負荷情報の更新時間・伝達時間は、今回の評価では無視したが、タスクの生起間隔および処理時間が短い場合は、これらを考慮に入れた評価を行わなくてはならない。SU－2による負荷分散適応型網は、これらの時間が短く、遅延の少ない正確な負荷情報によりタスク割付けを行うことができると推察される。

*機能分散型のシステム。たとえば、浮動小数点演算器を少数のプロセッサが持つ場合など。

SU－2を用いた相互結合網は、現在開発中の高並列推論エンジンPIE[7,8]の負荷分散網として用いる予定である。

5．むすび

動的負荷分散を行う相互結合網の提案、その構成要素であるSUの設計、負荷分散のシミュレーション評価に関して述べた。その結果、SU－2を用いた結合網による負荷分散制御の有効性が確認された。今後の課題としては、4．2節で述べたものの他に、非定常的な問題を想定したシミュレーション評価、一般の網（超立方体網・CCC網などを含む）に適用するためのSUの機能拡張、結合網の信頼性の検討、SU－2（b）のLSIによる試作、プロセッサの網インタフェースの設計などが挙げられる。

《参考文献》

1) Wang, Y. and Morris, R. J. T. : Load Sharing in Distributed Systems, IEEE Trans. Comput., Vol. C-34, No.3, pp.204-217 (1985).

2) Chow, Y. C. and Kohler, W. H. : Dynamic Load Balancing in Homogeneous Two Processor Distributed Systems, in Computer Performance, K. M. Chandy and M. Reiser, Eds. Amsterdam, The Netherlands, North Holland, pp.39-52 (1977).

3) Chow, Y. C. and Kohler, W. H. : Models for Dynamic Load Balancing in a Heterogeneous Multiple Processor System, IEEE Trans. Comput., Vol. C-28, pp.354-361 (1979).

4) Keller, R. M., Lindstrom, G. and Patil, S. : A Loosely-Coupled Applicative Multi-Processing System, NCC, Vol.48, pp.613-622 (1979).

5) Darlington, J. and Reeve, M. : ALICE A Multi-Processor Reduction Machine for the Parallel Evaluation of Applicative Languages, Proc. Fuctional Programming Languages and Computer Architecture, pp.65-75 (1981).

6) 成瀬，吉田，武末，雨宮：マルチプロセッサ型データフロー計算機DFMの実行制御方式の検討 第28回情処全大, 4F-5 (1984).

7) Moto-oka,T., Tanaka,H., Aida,H., Hirata,K. and Maruyama,T. : The Architecture of a Parallel Inference Engine －PIE－ ,Proc. Int' l Conf. on FGCS '84, pp.479-488 (1984) .
8) 坂井，田中，元岡：高並列推論エンジンPIEにおける相互結合網の構成，信学技報，EC 84-46 (1984) .
9) 平木，関口，島田：科学技術計算用データ駆動計算機SIGMA－1における負荷分散機構，信学技報 ,EC85-6 (1985) .
10) 坂井，喜連川，田中，元岡：関係代数マシンGRACEにおけるバケット分配網，信学論 (D) Vol.68-D, No.6 掲載予定 (1985) .
11) Feng,T. : A Survey of Interconnection Networks,IEEE Comput., Vol.14, No.12, pp.12-27 (1981) .
12) Lawrie,D. H. : Access and Alignment of Data in an Array Processor, IEEE Trans. Comput., Vol. C-24,No.12, pp.1145-1155 (1975) .
13) Dias,D. and Jump., J. R. : Analysis and Simulation of Buffered Delta Networks, IEEE Trans. Comput., Vol. C-30, No.4, pp.273-282 (1981) .
14) Patel, J. H. : Performance of Processor Memory Interconnection for Multiprocessors, IEEE Trans. Comput., Vol. C-30,No.10, pp.771-780 (1981) .
15) Adams, G. B.,Ⅲ and Siegel,H. J. : The Extra Stage Cube : A Fault-Tolerant Interconnection Network for Supersystems, IEEE Trans. Comput., Vol. C-31,No.5, pp.443-454 (1982) .
16) Adams, G. B.,Ⅲ and Siegel,H. J. : Modifications to Improve the Fault Tolelance of the Extra Stage Cube Interconnection Network, Proc. of the 1984 Int' l Conf. on Parallel Processing, pp.169-173 (1984) .
17) Chin,C. Y. and Hwang, K. : Connection Principles for Multipath Packet Switching Networks,The 11th Annu.Symp. on Comput. Arch., pp.99-108 (1984) .
18) Parker,D. S. and Raghavenda,C. S. : The Gamma Network : A Multiprocessor Interconnection Network with Redundant Paths ,The 9th Ann. Symp. on Comput. Arch., pp.73-80 (1982) .
19) 坂井，計，田中，元岡：可変ルーティング機能を付加した相互結合網のスイッチング・ユニット，情処論，Vol.26,No.4,pp.1-7 (1985) .

(盛文社)