計算機アーキテクチャ　66－2
（１９８７．７．１０）

# マルチマイクロプロセッサシステムの大容量共有メモリの一構成法

加納　尚之*・山根　一博・井上　倫夫・小林　康浩
* 米子高専・鳥取大学工学部

本報告では共有メモリを階層化して，全プロセッサで共同利用するデータを貯えるメインメモリと特定のプロセッサごとの共通ワーキングエリアとしてのサテライトメモリに機能分化したシステムにおけるサテライトメモリの構成とバス結合方式について述べる．

具体的には複数のサテライトメモリユニットをマルチリードワンライト方式とし，サテライトメモリユニットへのアクセスバスを書込み用と読み出し用とに分離し，かつ，書込み用バスをオメガネットワークに接続することにより，（１）グループ内いずれかのプロセッサからの全サテライトメモリユニットへの同一内容の書込み，（２）グループ内プロセッサがそれぞれにグループ別に各サテライトメモリユニットへの読み書き，（３）グループ単位で行うパイプライン処理を効果的に実行できるようにしている．

# Design of Satellite Memories of The Multimicroprocessor System

Naoyuki KANO* ・Kazuhiro YAMANE ・Michio INOUE・Yasuhiro KOBAYASHI
Department of Electrical Engineering, Faculty of Engineering, Tottori University
* Yonago National College of Technology
4-101 Koyama-Minami, Tottori, Tottori, 680 Japan

Our multimicroprocessor system, called "Sakyu", furnishes two kind of common memory: i.e., a main memory for storaging prime data globally used by all processors and satellite memories for offering common working areas by every local group of processors. The access buses to these satellite memories are specialized to implement an exclusive use : i.e., either read-only or write-only operations. The write-only buses are connected to the processors via the omega network. Thereby, it becomes possible to implement (1) simultaneous broadcasting from a specified processor to all satellite memories, (2) private communication between processors in every local group, and (3) pipe-lining at every local groups of processors.

This paper is concerned with the design of the satellite memories and evaluation of its usefulness.

## 1. はじめに

市販の1チップマイクロプロセッサ（以下μPと記す）を数多く使用しても経済的にはそれほど重荷ではなくなってきた．このμPを複数結合して，演算処理能力の向上を図るマルチμPシステムが各種提案されている[1],[2]．筆者らは，複数のμPを接続する方法として資源共有型（共有バス方式）によるマルチμPシステムの開発を行っている[5]～[7],[12],[13]．一般に，この種のシステムは接続可能なμPの台数に物理的制限があり，処理能力の飛躍的な向上は困難であるが，既存のハードウェア技術を活用して，システムの処理能力を十数倍に向上させることは比較的容易である[6],[7]．また，種々の処理アルゴリズムに柔軟に適用できるような利便性のよいシステムの構築が可能である．

マルチμPシステムで能率よく並列処理を行うには，ジョブを複数の並列タスクに分割して複数台のμPで均等に処理することが望まれる．しかし，ジョブレベルで見れば，必ずしもジョブ全体を並列処理できるわけではなく，単一のμPにより処理せざるを得ない部分も存在する．このため，システムの処理速度は，必ずしもμPの構成台数には比例しない．これは，システムの構成規模を決定する上で重要な因子であり，システムの基本設計時に十分に考慮しなければならない点である[3],[4]．

また，共有資源利用時の各μPのアクセス競合による待機時間の増加を抑制できなければ，これによっても，システムの処理能力が著しく低下し，期待されるシステムの能力を出し得なくなる．このアクセス競合に対する効果的な対策をも実施しなければならない．

この対策としてはメモリ空間の利用頻度別階層化，アクセスバスの多重化や読み出し・書込みバスの専用化，バス選択切換機構の効率化などが講じられなければならない．また，アクセス競合が生じたときには速やかにこれを調停する機構も必要である．

本報告では，まずシステムの構成規模を決定する上で重要な，並列処理の特性（並列化率とμPの台数の関係）について述べる．次に，現在開発中の並列計算機“砂丘”[12],[13]のシステム構成を示し，大容量共有メモリとしてマルチリード・ワンライトメモリ方式を採用した場合のシステムの性能特性について述べる[8],[9]．

## 2. 並列化率とμPの台数

あるジョブを単一のμPで処理させた場合の処理時間がTj（1）［s］であったとする．このジョブのp割（0≦p≦1）を，上記μPと同等の機能を有するn台のμPを用いて並列処理できたとすれば，このジョブの全処理時間は次式で表される．

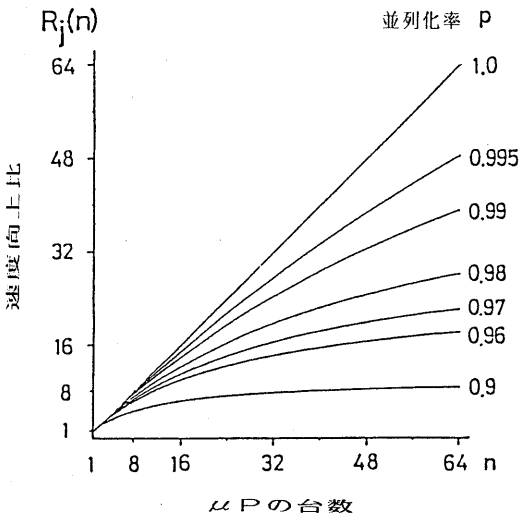



図1 μPの台数と速度向上比

$$T_j(n) = (1-p)T_j(0) + \frac{p}{n}T_j(0) \qquad (1)$$

ただし，ここでは並列処理のためのオーバーヘッド等のロス時間は無視して考える．並列処理による速度向上比をRj（n）とすれば次式で与えられる．

$$R_j(n) = \frac{1}{1 - \frac{n-1}{n}p} \qquad (2)$$

図1に，並列処理の割合p（以下，並列化率と定義する）に対するμPの台数nと速度向上比Rj（n）を（2）式に従って計算した結果を示す．Rj（n）の値はpの値に鋭敏に反応し，並列化率が僅かに低下しても台数効果を著しく阻害することがわかる．したがって並列化率が0．9程度しか期待できないようなジョブの処理に対しては，そのまま全てのμPで並列処理を行うのではなく，小グループでの並列処理，分散処理，ジョブレベルでのタスクのパイプライン処理等を課題に即して柔軟に対応できるようにすべきである．

以上のことから，筆者らは実験室レベルで特定ユーザが利用するデータ処理システムとして“安く”，比較的“速く”，しかも“扱いやすい”システムを目標に，数台～数十台のμPで構成される資源共有型の並列計算機“砂丘”を製作中である[12],[13]．

## 3. 並列計算機“砂丘”のシステム構成

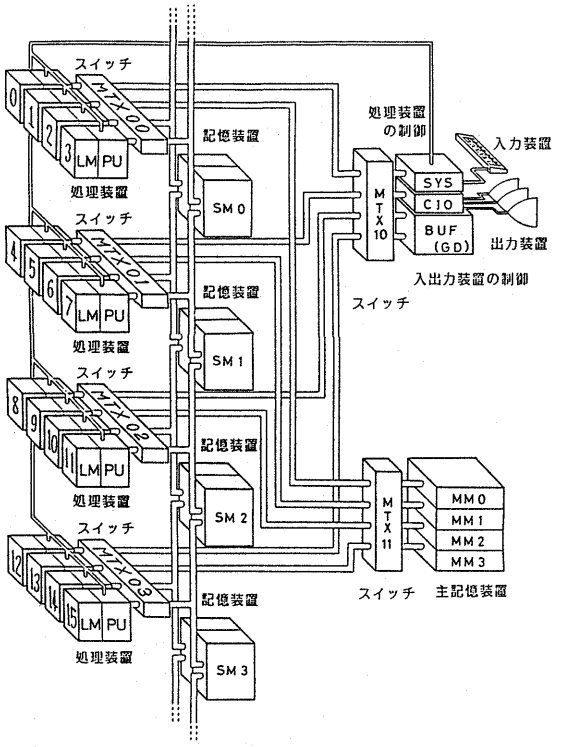
現在製作中の並列計算機“砂丘”のシステム構成を図２に示す．本システムは，マトリックススイッチを２段構成とし，物理メモリの構成を，ローカルメモリ，サテライトメモリ，メインメモリおよびシステムメモリの３階層にしている．ここで，ローカルメモリはプログラム格納エリアとして，サテライトメモリは特定のプロセッサごとの共通のワーキングエリアとして，メインメモリおよびシステムメモリは全プロセッサで共同利用するデータエリアとして機能分化している．各マトリックススイッチＭＴＸは，ほぼ同一の機能を有し，４×４のパススイッチとして用いる．μＰを４台一組として４組１６台をそれぞれのマトリックススイッチ　ＭＴＸ０ｉに接続する．システムの共有資源（メインメモリ，システムメモリ，Ｉ／Ｏ装置等）を，後段のマトリックススイッチＭＴＸ１ｉに接続し，すべてのμＰより平等に利用できるようにした．

従来の一段構成のマトリックススイッチ方式[6] では困難であったμＰ台数の増加が比較的容易となり，システムに拡張性ができた．また，メモリを階層化することによって，各グループでの並列処理および分散処理，数グループによるタスクレベルでのジョブのパイプライン処理等，処理アルゴリズムにあわせてシステムを柔軟に運用できるようになった．

## 4. 大容量共有メモリ

一般に，メモリ等資源を共有する密結合システムでは，バス結合の制御が頻繁に行われる．したがって，各μＰの共有資源アクセス要求を効率よく制御し，アクセス競合による待機時間の累積をいかに抑制するかがシステムを能率よく運用する上で重要な課題である．

本システムでは，メモリ空間を３階層に分割し，共有メモリをメインメモリとサテライトメモリの２タイプで構成している．従来よりこのサテライトメモリの運用は，処理アルゴリズムに対応してある程度変更できるようにしている．開発当初は，主に数値計算を効率よく処理すべく，メモリ空間の配分が考えられていたが，非数値演算に対する要望も多く，より大容量のメモリ空間が必要とされている．そこで，このシステムをさらに能率よく運用するために，サテライトメモリをマルチリード・ワンライトメモリ方式[8],[9] に変更することを検討した．具体的には，サテライトメモリへのリードアクセスパスとライトアクセスパスとを分け，ライトアクセスパスは，オメガネットワークで接続する[10],[11] ．図３に各μＰとサテライトメモリとの接続図を示す．




PUi ：処理ユニット　μＰ（HD68000-8）
LM ：ローカルメモリ　２５６KB
SMi ：サテライトメモリ　２×２５６KB
MM ：メインメモリ　１～４MB

MTX0i：前段マトリックススイッチ
MTX1i：後段マトリックススイッチ
SYS ：PU制御，システムＩ／Ｏ
ＣＩＯ ：ユーザ用共有Ｉ／Ｏ
BUF ：システムバッファメモリ
（GD）：濃淡画像表示メモリ

図２　並列計算機“砂丘”のシステム構成

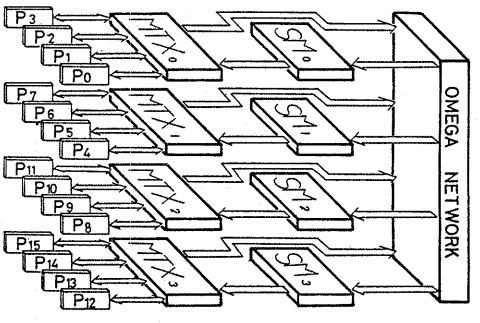



図３　複数μＰとサテライトメモリとの接続

## 4．1　マルチリード・ワンライトメモリ方式とオメガネットワーク

サテライトメモリへのリードアクセスパスを図4に，ライトアクセスパスを図5に示す．

（1）　リードアクセスパス

一般に，μPのメモリアクセスは，リード動作の方がライト動作より多く行われる．したがって複数のμPをグループ化し，特定のμP数台で一つのメモリユニットを共有させ，それぞれのグループでリードアクセスをそのメモリユニットに限定するようにすれば，アクセス競合はそのグループ内で配慮すればよい．

（2）　ライトアクセスパス

複数に分散配置されたメモリユニットへのライトアクセスパスをオメガネットワークで接続する．これによって，サテライトメモリの利用形態を処理アルゴリズムに対応してダイナミックに変更することが可能となる．ただし，ライトアクセスは，全μPの並列動作によるアクセス競合を考慮しなければならない．

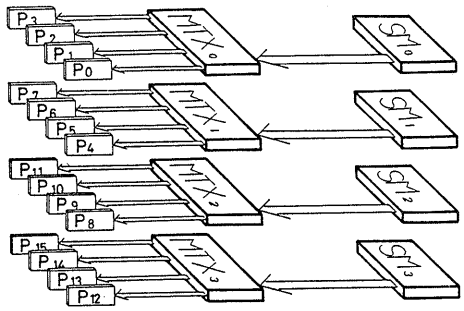



図4　リードアクセスパス

## 4．2　サテライトメモリのダイナミックアロケーション

サテライトメモリ（複数のメモリユニット）のライトアクセスパスを，オメガネットワークで接続することによって，基本的には以下に示す3種類の利用形態を実現できる．オメガネットワークで利用するバススイッチの接続モードは図6に示すように，並行接続a，クロス接続b，ブロードキャスト接続c，dの4タイプが可能である．ジョブまたは，タスクレベルで接続モードを固定することも，各ライトアクセスサイクルごとにそのつど変更することも可能である．

（1）一斉放送モード（図7－a）

バススイッチの接続モードを，ブロードキャスト接続c，dにし，ライトアクセスサイクルごとに変更できるようにすれば，特定のμPのライトアクセスを全メモリユニットに対して同時に行うことができる．これによって全メモリユニットは，全て同一の内容を保持できる．

（2）個別利用モード（図7－b）

バススイッチの接続モードを並行接続aに固定すれば，各グループのライトアクセスパスはリードアクセスパスで利用するメモリユニットと同一のユニットに接続される．各グループのμPが独立にそれぞれのメモリユニットを利用することができる．

（3）パイプライン接続（図7－c）

バススイッチの接続モードを並行接続aとクロス接続bとの両モードを併用して利用すれば，任意のメモリユニットへライトアクセスパスを接続することができる．たとえば，各グループがライトアクセスパスを隣接するメモリユニットへ接続すれば，タスクのパイプライン処理が可能である．これは，現システムのサテライトメモリの利用形態とほぼ同様の方法にあたる．

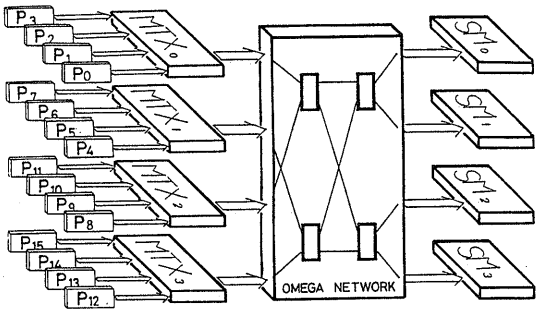



図5　ライトアクセスパス

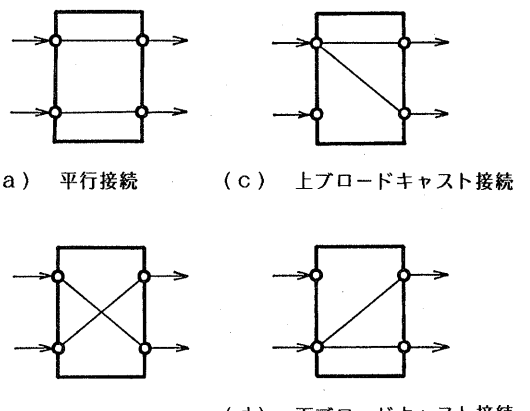



図6　バススイッチの接続モード

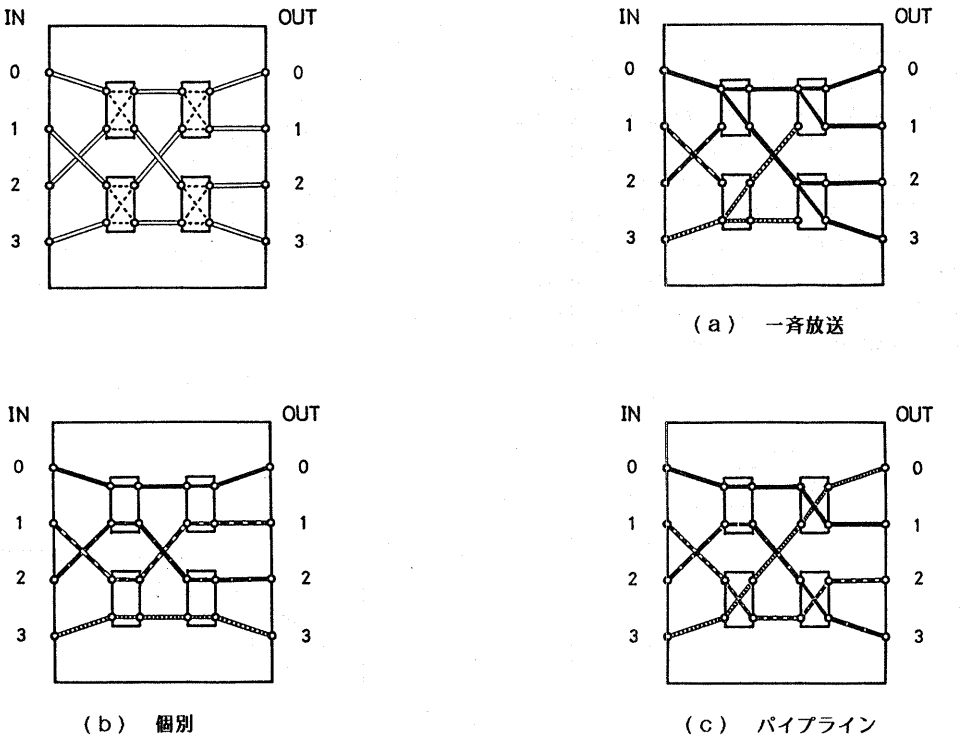



図7　サテライトメモリへのライトアクセスパス

## 5．　検討

### 5．1　並列動作時の各μPの稼働率

（1）稼働率の定義とその実測法

ここでは，μPとメモリシステムとの接続を図8のようにモデル化する．サテライトメモリ，メインメモリ，システムメモリ等の共有メモリを外部メモリとする．これらのメモリをμPがアクセスするとき同期のために挿入されるウェイト時間によってμPの持てる能力がどれくらい低下するかの指標としてμPの稼働率を定義する．

具体的には，ある処理プログラムを用いて下記の条件下での処理時間を計測することによって行う．

a）処理プログラムとデータの両方がローカルメモリに存在するとき　・・・・・・・　T（0）［s］

b）処理プログラムがローカルメモリに，データが外部メモリに存在するとき　・・・　T（k）［s］

ここで，上記処理プログラムのデータのアクセス回数をk回とすれば，稼働率P，シンクタイムThを以下のように定義することができる．

$$P = \frac{T(0)}{T(k)} \times 100 \quad [\%]$$

$$Th = \frac{T(0)}{k} \quad [s] \qquad (3)$$

ここで，シンクタイムThは，μPが外部メモリをアクセスする間隔に相当する．μPの動作に注目すれば，アクセス間隔Th［s］ごとに，ウェイト時間tw［s］が挿入されることになる．したがって，μPの稼働率をこのシンクタイムとウェイト時間とをもとに次式で表すことができる．

$$P = \frac{100}{1 + \frac{t_W}{Th}} \quad [\%] \qquad (4)$$

図9に，現システムでの，ウェイト時間に対するシンクタイムThとμPの稼働率Pの関係を示す．図中で，○印はサテライトメモリ，△印はシステムメモリをそれぞれ利用する場合の稼働率の実測値を表している．

（２）稼働率の予測

筆者らは，“複数のμＰが同一のタスクを実行しているとき最もアクセス競合が頻繁に発生して各μＰの待機時間が増加する”ことに着目し，この待機時間の累積を定量することによってタスクレベル（ほぼ同一の処理ルーチンを繰返し実行している状態）でのシステムの性能を陽的に解析できることを示した[7]．これより，システムのハードウェア特性（共有資源のアクセス時間，サイクル時間）および，ソフトウェア特性（タスクの平均シンクタイム）とを用いて，並列動作時の各プロセッサの平均稼働率を次のように表すことができる．

$$P(n,Th) = \frac{100}{1 + \dfrac{t_{AC} + m_n t_S}{Th}} \quad [\%] \quad (5)$$

ここで，

$$m_n = \begin{cases} 0 & , n \leqq i_B n_0 \\ \dfrac{n}{i_B} - n_0 & , n > i_B n_0 \end{cases} \quad (6)$$

ただし，

$n$　：同時に動作しているμＰの台数

$Th$　：タスクの平均シンクタイム［ｓ］

$t_S$　：共有資源のサイクル時間

$t_{AC}$　：共有資源をアクセスするとき各μＰが要する平均アクセス時間［ｓ］

$$t_{AC} = t_B + t_A + \frac{1}{2} t_S \quad (7)$$

$t_A$　：共有資源のアクセス時間

$t_B$　：バスドライバー，ケーブル，マトリックススイッチバスアービタ等による信号遅延時間［ｓ］

$i_B$　：共有バスのインタリーブ数（多重度）

$n_0$　：同一の共有バスでアクセス競合による待機時間の累積なしに動作できるμＰの台数，

$$n_0 = 1 + \frac{Th}{t_S} \quad (8)$$

上式をもとにシステムのハードウェア特性の評価を行うことができる．具体的にはバスの多重度，高速メモリの評価等をシステムの基本設計時に行うことができる．

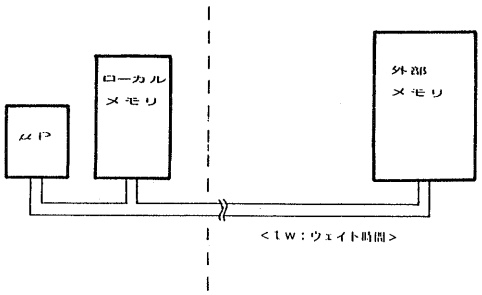



図８　μＰと外部メモリとの接続

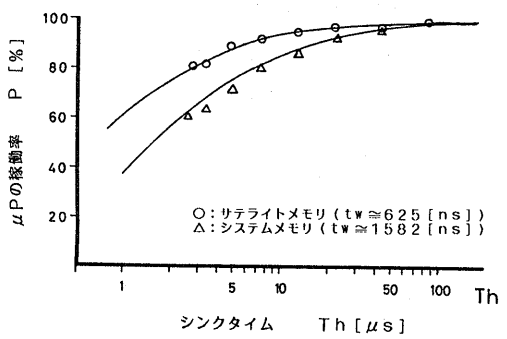



図９　平均シンクタイムとμＰの稼働率

○，△：μＰの処理時間より求めた実測値

――：理論値

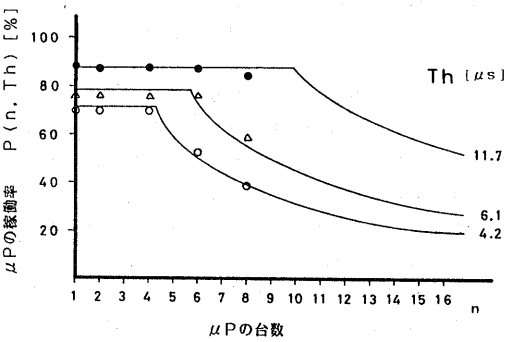



図１０　システムメモリ利用時のμＰの台数とその稼働率

●，△，○：μＰの処理時間より求めた実測値

――――：理論値

表１　ハードウェア特性

| | n | $i_B$ | $t_B$ [ns] | $t_A$ [ns] | $t_S$ [ns] |
|---|---|---|---|---|---|
| サテライトメモリ | 4(8) | 2 | 125 | 250 | 500 |
| メインメモリ | 16(32) | 4 | 250 | 330 | 660 |
| システムメモリ | 16(32) | 1 | 250 | 666 | 1332 |

（　）：２ポートメモリ方式による最大接続台数

現システムのハードウェア特性を表１に示す．メインメモリについては現在計画中であり設計値を示しておく．

図１０に，現在のシステムでシステムメモリ利用時の各シンクタイムＴｈに対するμＰの台数と稼働率の関係を示す．図中●△○印は，あるタスクの処理時間の計測により求めた実測値を示す．また，実線は，各シンクタイムでの稼働率を（５）式で求めた計算値である．以上の結果より，（５）式の稼働率の予測は，実測結果と良く一致していることがわかる．

５．２　マルチリード・ワンライトメモリ方式とμＰの稼働率

以上の結果をもとに，現システムのハードウェア特性を用いて前節で述べたマルチリード・ワンライトメモリ方式とオメガネットワークの利用について，（５）式を用いて各μＰの稼働率を求めてみる．ただし，ネットワーク内のバススイッチの信号遅延時間を現システムと同等の値と考える．また，並列に動作させるプログラムの平均シンクタイムＴｈを５［μｓ］以上であると考える．

（１）リードアクセスパス

同時に動作するμＰは，グループ内の台数にのみ注目すればよい．グループ内のμＰ台数をｎとし，各シンクタイムＴｈに対するμＰの稼働率を図１１に示す．グループ内のμＰ台数が８台ぐらいであれば，インタリーブ数ｉｂは１でよいことが分る．ただし，８台以上の接続の場合は，ｉｂ＝２にすべきである．

（２）ライトアクセスパス

ライトアクセスの競合はオメガネットワーク内のバススイッチの接続モードによって異なる．

ａ）オメガネットワーク内での競合がない場合

バススイッチの接続モードが並行ａ，クロスｂであり，比較的長時間接続モードの変更を行わないとき，オメガネットワーク内ではアクセス競合は起らない．リードアクセスパスのときと同様にマトリックススイッチでの同一グループ内での競合だけを考慮すればよい．

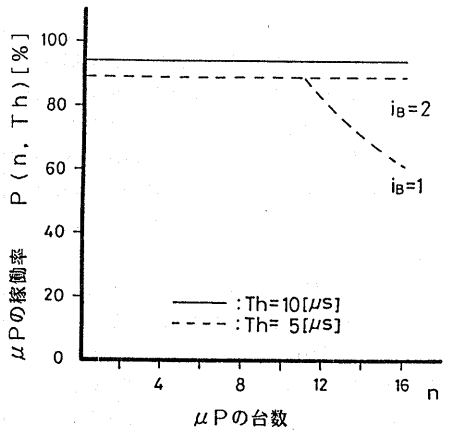



図１１　μＰの台数とその稼働率（リードアクセスパス）

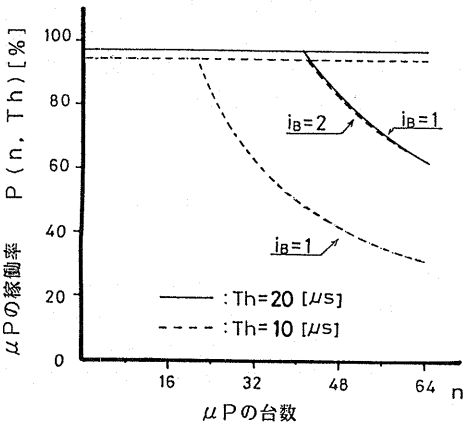



図１２　μＰの台数とその稼働率（ライトアクセスパス）

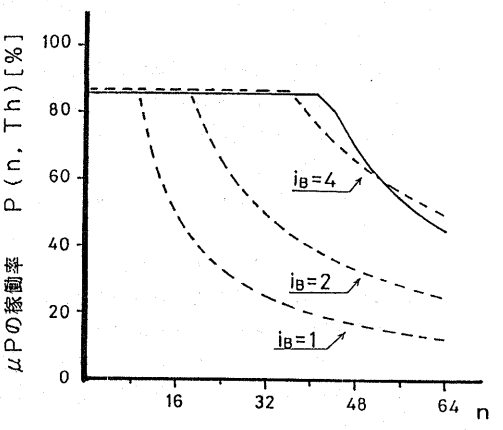



図１３　μＰの台数とその稼働率

－－－：従来のサテライトメモリ方式
（平均シンクタイムＴｈ＝４［μｓ］）
――：マルチリード・ワンライトメモリ方式
（平均リード間隔　＝　５［μｓ］，
平均ライト間隔　＝２０［μｓ］）

b）オメガネットワーク内で競合がある場合

ライトアクセスサイクルごとに頻繁にバススイッチの接続モードの変更を行うとき，特に接続モードがブロードキャスト接続c，dのときオメガネットワーク内で全μPの並列動作によるアクセス競合が頻繁に起こり得る．ライトアクセスパスでのμPの台数と稼働率について図12に示す．

ここで，処理の最少単位をリードアクセス4回とライトアクセス1回として，平均処理時間を20［μs］とする．これを十分な回数繰返し実行している並列タスクを考える．マルチリード・ワンライトメモリ方式と従来のメインメモリまたは，システムメモリを使用した場合のμPの稼働率を図13に示す．実線はマルチリード・ワンライトメモリ方式を，破線は従来のメモリ方式の場合を示している．共有バスのインタリーブ数（多重度）ibを増加させないでも，マルチリード・ワンライトメモリ方式によれば，シンクタイムの短い処理をμPの稼働率を著しく低下させないで実行させ得ることがわかった．

## 6．　おわりに

以上，大容量共有メモリを能率よく利用する方法として，マルチリード・ワンライトメモリ方式と，ライトアクセスパスにオメガネットワークを導入した場合について，各アクセスパスを利用したときのμPの稼働率の低下について述べた．

現在のサテライトメモリ方式において，メモリユニットへのリードアクセスパスとライトアクセスパスとを分離し，ライトアクセスパスをオメガネットワークで接続するようにシステムを変更すれば

（1）各グループによる並列処理結果を隣接グループへ転送するパイプライン処理が効率よく実現できる．

（2）一斉放送モードを利用すれば，全メモリユニットを同一内容に保持できる．

（3）各グループ別にメモリユニットを利用する．グループごとの単独処理ができる．

（4）上記（1），（2），（3）を，アクセスサイクルごとにダイナミックに変更する事もできる．

このとき従来の共有メモリ方式およびメモリユニットを多重化したインタリーブ方式のシステムよりも，μPのアクセス競合を極力おさえ大容量共有メモリを能率よく利用できることがわかった．

今後は，オメガネットワークで使用するバススイッチを具体化することにあり，現在作業を進めている．

## 参考文献


［1］　白川　他：“並列計算機PAX－128”通信学論，Vol.J67-D，No.8，pp.853-860，Aug.(1984)

［2］　出口　他：“コンピュータグラフィックスシステムLINKS－1における画像生成の高速化手法”情報処理論文誌，Vol.25，No.6，pp.944-952，Nov.（1984）

［3］　Rodrigue，G.：“Prallel Computations”Academic Press (1982)

［4］　Paker，Y.：“Multi-microprocessor systems”Academic Press (1983)

［5］　井上・小林：“マイクロプロセッサを用いた並列処理システムα－16”　シミュレーション第2回研究会資料，pp.19-24，March (1982)

［6］　井上・小林：“マルチマイクロプロセッサシステムα－16のアーキテクチャ”情報処理論文誌，Vol.25，No.4，pp.632-639，July (1984)

［7］　井上・小林：“α－16マルチマイクロプロセッサシステムの性能評価”情報処理論文誌，Vol.25，No.4，pp.640-646，July (1984)

［8］　阿江・相原：“三次元集積回路を想定した並列処理方式の一実現”情報処理論文誌，Vol.26，No.6，pp.1145-1148，Nov.（1985）

［9］　中川　他：“多重アクセス形仮想記憶を備えた汎用並列計算機の一構成法”情報処理論文誌，Vol.28，No.5，pp.525-533，May (1987)

[10]　Stone，H. S.：“Parallel Processing with the Perfect Shuffle”IEEE，Trans.Comput.，Vol.c-20，No.2，pp.153-161，Feb.（1971）

[11]　Lawrie，D. H.：“Access and Alignment of Data in an Array Processor”IEEE，Trans.Comput.，Vol.c-24，No.12，pp.1145-1155，Dec.（1975）

[12]　井上・小林：“資源共有型マルチマイクロプロセッサシステムにおけるアクセス競合の調停について”電子情報通信，回路とシステム研究会資料CAS84-206，pp.9-16，（1985）

[13]　山根　他：“並列計算機“砂丘”のハードウェアアーキテクチャ”電子情報通信，コンピュータシステム研究会資料，（1987）