

## 可変構造型並列計算機の構想

村上和彰<sup>‡</sup> 福田 晃<sup>†</sup> 末吉敏則<sup>†</sup> 富田眞治<sup>†</sup>

† : 九州大学大学院 総合理工学研究科 情報システム学専攻

‡ : 九州大学 工学部 情報工学科

『汎用』の高並列処理環境を構築するために、相互結合網を可変構造にした256台規模のマルチマイクロプロセッサ・システムを開発している。可変構造の相互結合網とは256×256のクロスバー網を基にしたもので、様々なトポロジーに成り得る。本システムはこの相互結合網を中心とした仮想計算機システムであり、各種のプロセッサ-プロセッサ結合形態およびプロセッサ-メモリ結合形態を精確にシミュレーションできる。これにより、高並列処理計算機アーキテクチャの一般性のある評価、ならびに、並列処理分野の一層の拡大が可能となる。本稿では、そのシステム構想について述べている。

### SYSTEM PHILOSOPHY OF A RECONFIGURABLE PARALLEL-PROCESSOR (in Japanese)

Kazuaki MURAKAMI<sup>‡</sup>, Akira FUKUDA<sup>†</sup>, Toshinori SUEYOSHI<sup>†</sup> and Shinji TOMITA<sup>‡</sup>

† : Dept. of Information Systems

‡ : Dept. of Computer Science and Communication Eng.

Kyushu University

Kasuga, Fukuoka, 816 Japan

We are now developing a reconfigurable parallel-processor ; a multi-microprocessor system with a reconfigurable interconnection network. The reconfigurable interconnection network is a 256×256 crossbar-switch network with a capability to simulate topology of other interconnection networks. The reconfigurable parallel-processor is a kind of virtual-machine system, which can virtualize any networks for processor-processor and/or processor-memory interconnection. Our objects are to provide a general-purpose parallel-processing environment and to research well-developed parallel architectures.

## 1. はじめに

マイクロエレクトロニクス技術の著しい発達にともない、その特長を最大限に活用し得る新しい計算機システムアーキテクチャとして、マルチマイクロプロセッサ構成による並列処理計算機システムが注目を集めている。マルチマイクロプロセッサのシステム構成にあたっては、構成要素であるマイクロプロセッサおよびメモリ周辺の設計もさることながら、これらをどのように結合するかがシステム成功の鍵を握っている。実際これまでに、バス、木状網、格子網、超立方体網、オメガ網などの種々の相互結合網を有した特徴のあるシステムが研究開発されている。

しかしながら、これらの結合形態は処理形態すなわちプロセス間通信パターンや記憶参照パターンと表裏一体の関係にあり、相互結合網単体の評価が難しい。したがって、従来の並列処理計算機の研究では、実装した相互結合網の形態に依存した枠組み内での、個別的な評価および検討を行っているものが多い。しかも、相互結合網が特定の処理形態を強く反映しており、システムとしての応用分野が限定されがちである。また、プログラムを作成する際にも、物理的な結合形態を常に意識しなければならないという弊害がある。

将来における処理の高並列化を促進するには、応用分野の裾野を広げ、かつ、種々の並列アルゴリズムを容易に実装できる『汎用』の高並列処理計算機の実現が強く望まれる。また、結合形態も議論の対象として、高並列処理の諸方式をより一般性のある評価軸で総合的に比較および評価することが必要である。

そこで、筆者らは、相互結合網を可変構造にして、多様な結合形態を精確にシミュレートできるとともに、各種の応用ならびに並列アルゴリズムに柔軟に適應できる可変構造型並列計算機の開発を進めている。<sup>1)~3)</sup> 本稿では、そのシステム構想について述べる。

## 2. 開発目的

本システムの開発目的は、以下の課題に対する研究を大きく推進し、高並列処理環境のための総合的な評価・設計システムを構築することである。

### (1) 応用分野の拡大と並列アルゴリズムの開発

科学技術計算や知識情報処理などの計算量あるいはデータ量が膨大な処理分野には、かなりの並列性が内在していると言われる。並列性には自明なものもあれば潜在的なものもあって、これらを引き出すためには並列処理を前提としたアルゴリズム、つまり、並列計算アルゴリズムの開発が必須である。しかし、せっかく開発した並列計算アルゴリズムも、実際の計算機に実装するときにはその計算機構造に合った(並列または逐次)処理プログラムに変換しなければならない。あるいは、並列計算アルゴリズム自体を変更しなければならない。さもないと、処理性能が犠

牲になる可能性があるからである。

本システムでは、プロセッサ-プロセッサ間およびプロセッサ-メモリ間の相互結合網が可変構造になっているので、並列計算アルゴリズムを素直な形で並列処理プログラムに変換できる。これにより、計算機の物理的構造に依存しない並列計算アルゴリズムの開発が可能となる。

また、1つの並列計算アルゴリズムに対して各種の相互結合網をシミュレートすることで、両者の間の親和性を定量的に比較および評価し、応用を指向した高並列処理計算機のアーキテクチャを模索する。

### (2) 並列プログラミング言語および計算モデルの開発

先の並列計算アルゴリズムを実際に計算機上で実行可能とするためには、なんらかの並列プログラミング言語を用いて並列処理プログラムを記述する。このプログラムで表現されるアルゴリズムが並列処理アルゴリズムである。並列計算アルゴリズムをうまく並列処理アルゴリズムに写像するには、記述能力の高い並列プログラミング言語が不可欠である。一般に、並列プログラミング言語はある計算モデルに基づいて設計される。たとえば、表1に示すような計算モデルとそれに基づく言語がある。さらに、計算モデルのいくつかは、特定の計算機モデルを想定している場合がある。たとえば、プロシージャ指向モデルは密結合型モデルを、また、メッセージ指向モデルおよびオペレーション指向モデルでは疎結合型モデルを想定している。<sup>4)</sup>

本システムでは、可変構造の相互結合網に加えて、メモリ構成として密結合および疎結合のいずれをも装備している。これにより、結合形態やメモリ構成にとらわれることなく、新しい並列プログラミング言語あるいは計算モデルの試作および試験が容易にできる。

また、既存の言語を実装して、それに対する各種の相互結合網やメモリ構成のシミュレーションを行い、並列プログラミング言語指向計算機アーキテクチャを検討する。

### (3) 並列処理オペレーティング・システムの開発

並列処理プログラムにいくらかの並列性が含まれていても、実行時にそれらを引き出す環境、つまり並列処理オペレーティング・システム(OS)がないと意味がない。並列処理OSにおいては、逐次処理OSに比べて、

#### ① プロセスのプロセッサへの割付け

表1. 並列プログラミング言語の例

計算モデル	並列プログラミング言語
プロシージャ指向モデル	Concurrent PASCAL, Modula, Mesa, Linda, Nano-2, etc.
メッセージ指向モデル	CSP, Gypsy, PLITS, Lynx, etc.
オペレーション指向モデル	Distributed Process, StarMod, Ada, SR, etc.
データフローモデル	VAL, Id, Lucid, etc.
述語論理モデル	Prolog, GHC, etc.

- ② データのメモリへの割付け
- ③ プロセス実行順序のスケジューリング
- ④ プロセス間の同期・通信
- ⑤ プロセッサ間の負荷分散

などの機能が特に重要となる。

本システムでは、新OSを開発する過程で、上記機能を実現するための各種方式について検討する。

さらに、もし可能ならば既存のOSを本システム上に移植し、その動特性を把握する。

#### (4) 並列処理計算機アーキテクチャの評価

現在、表2に示すように、種々の相互結合網を有する商用ないし研究用の並列処理計算機が開発されている。<sup>9)</sup> これらの相互結合網のシミュレーションを行って、プロセッサ-プロセッサ間およびプロセッサ-メモリアン間における通信レベルでの詳細な測定データを収集し、並列処理計算機のシステムの挙動を明らかにする。

さらに、本システムをプロセッサ-メモリアンレベルのハードウェア・シミュレータとみなし、上記評価に基づいて次世代の高並列処理計算機的设计に活用する。

表 2. 並列処理計算機

単一バス結合	ACE (電総研), MICS-11 (NEC), EPOS (東芝), SMC-201 (シーメンス社), LINKS-1 (阪大), MACSYM (京大), Multimax (アンコール社), DSP9000 (アポロコンピュータ社), Balance2100 (シークエント社), RDBM (ブラウンシュバイク大), SABRE (INRIA), EVLIS (阪大)
階層結合	Cm* (カーネギー・メロン大), EXPERTS (京大), HPRG (沖電気), (SM) <sup>2</sup> -11 (慶大)
リング結合	HMMS (日立), Cyberplus (CDC)
木結合	HYPHEN-16 (九大), CORAL (徳島大), DADO (コロムビア大), DBC/1012 (テラデータ社), TOP (ピサ大)
格子結合	PAX-128 (筑波大), CAP (富士通), FAIM-1 (シュランベルジュ社), Multi SIM (ICOT)
2進n-キューブ結合	Cosmic Cube (カリフォルニア工科大), NCUBE (NCUBE社), FPS-T (フローティング・ポイント・システムズ社), iPSC (インテル社)
クロスバー結合	C.mmp (カーネギー・メロン大), S1 (スタンフォード大), DIRECT (ウィスコンシン大), YSE (IBM)
多段結合	Cedar (イリノイ大), HAL (NEC), Butterfly (BBN), RP-3 (IBM)

### 3. システム概観

本システムが提供する並列処理環境の概念図を図1に示す。大まかな処理の流れは、

- ① 並列処理プログラム記述：並列プログラミング言語を用いて、並列計算アルゴリズムを並列処理アルゴリズムに写像する。
- ② 並列処理プログラム実行：並列処理OSは、並列処理プログラムを並列処理計算機で実行する。OS自身も、この計算機上で動作する。

となる。並列処理計算機の構成として多様な方式を可能とするため、これを仮想計算機として実現することにする。

したがって、本システムは図2に示すような仮想計算機システムの階層構成をとる。システムに存在する2つの計算機を

- ① ゲストマシン：仮想計算機である並列処理計算機
- ② ホストマシン：実計算機である可変構造型並列計算機

と区別して呼ぶ。ホストマシンのハードウェアおよびカーネルは、与えられたゲストマシン・アーキテクチャの記述に従い、

- ① 相互結合網：プロセッサ-プロセッサ間およびプロセッサ-メモリアン間の結合形態 (図3参照)
- ② メモリ構成：密結合 (共有メモリがある) / 疎結合 (共有メモリがない) 構成 (図3参照)
- ③ プロセッサ：実際に存在するプロセッサの台数以上のプロセッサ
- ④ マルチプロセッサ用命令：同期・通信のための命令を仮想化する。これにより、ゲストマシン上のソフトウェア (ゲストカーネル, ゲストOS および並列処理プログラム)

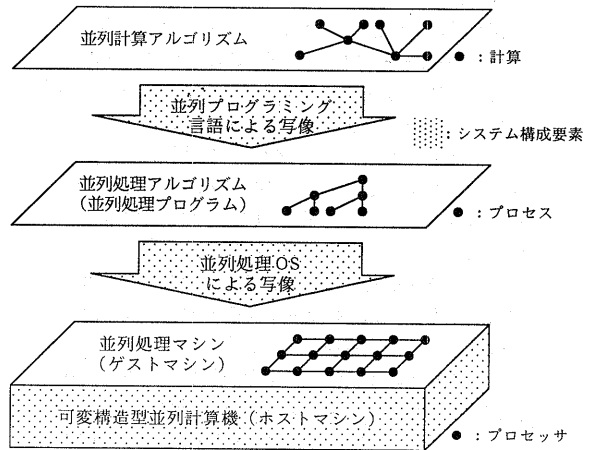


図 1. 並列処理環境

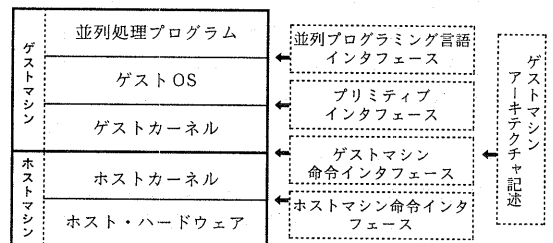


図 2. 本システムの階層構成

は、あたかもゲストマシンそのものが存在しているかのように動作可能である。

以上のことから、ホストマシンのハードウェアとしては次の要件を満たす必要がある。

- ① 結合形態を可変とするためには、ホストの相互結合網は特殊な構造を持ってはいけない。しかも、多様な結合形態をハードウェア・レベルでシミュレーション可能とするため、各種結合網のスーパーセット的な相互結合網である必要がある(図4参照)。
- ② 密結合/疎結合構成のメモリをハードウェアとして備える。密結合とは、複数のプロセッサからアドレスにより直接アクセス可能な共有メモリを有するものをいう。疎結合とは、そのような共有メモリを一切持たないものを指す。
- ③ 1台のプロセッサが複数台分のプロセッサの働きをする。
- ④ 共有メモリ・アクセス用命令およびメッセージ転送用命令を提供する。

これらの要件を鑑みて、本システムのハードウェアは、図5に示すような構成となっている。その特長としては、次の点が挙げられる。

- ① 相互結合網に動的網のスーパーセット的結合網であるクロスバー網を採用している。このクロスバー網を用いることにより後述するように、各種の静的網/動的網をシミュレーション可能としている。
- ② プロセッサは、このクロスバー網を経由して、任意のプロセッサないしメモリと接続される。プロセッサ・メモリ接続により、あるプロセッサは任意のメモリ(共有メモリ)に対してアドレスを指定して直にアクセスできる(密結合)。また、プロセッサ・プロセッサ接続により、あるプロセッサは任意のプロセッサに対してメッセージ転送を行える。
- ③ 1台の実プロセッサ上で複数の仮想プロセッサを時分割により動作させる。プロセッサ間通信におけるプロセッサ指定は、仮想プロセッサを指定するものとする。
- ④ 共有メモリへのアクセスは、マイクロプロセッサの通常のLOAD/STORE命令により行える。また、メッセージ転送も通常の入出力動作として実現する。

以上の特長を有することから、本システムを『可変構造型並列計算機』と呼ぶ。

#### 4. ハードウェアの概要

現在、256台のプロセッシング・エレメント(マイクロプロセッサおよび周辺装置からなる処理装置)を256×256のクロスバー網で接続する計画で開発を進めている。<sup>2)</sup>以下に、クロスバー網およびプロセッシング・エレメントの概要について述べる。

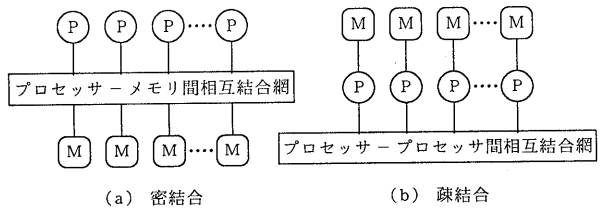
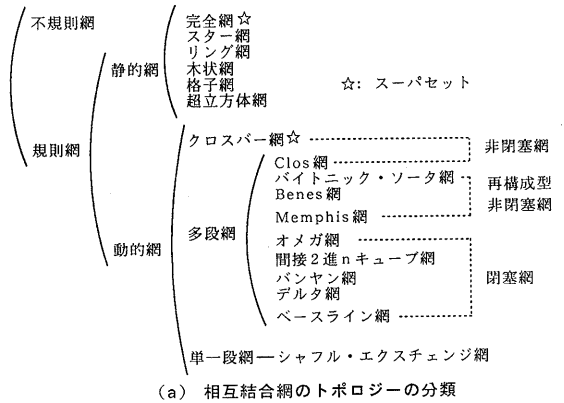


図3. ゲストマシン構成



(a) 相互結合網のトポロジーの分類

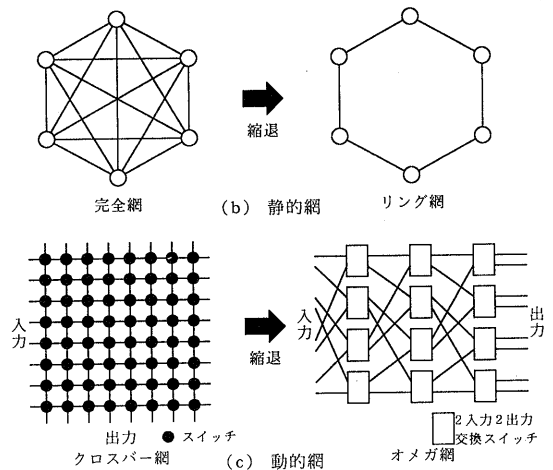


図4. 相互結合網

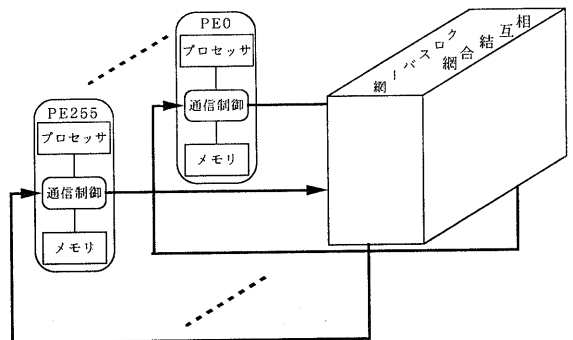


図5. 可変構造型並列計算機の構成

#### 4.1 クロスバー網

256×256のクロスバー網は、図6に示すように、これを8×8のクロスバー網1024個(32×32)に平面分割することで実現する。このクロスバー網は、クロスバー・スイッチ本来の動作に加えて、各種の相互結合網をハードウェア・レベルで効率よくシミュレーションできるように考慮してある。シミュレーションの対象となる項目には、

- ① トポロジー：図4(a)参照。
- ② 交換方式：
  - (i) パケット交換方式
  - (ii) 回線交換方式
- ③ 制御方式：
  - (i) 集中制御方式
  - (ii) 分散制御方式

などがある。以下、トポロジーのシミュレーション方法について、簡単に述べる。

##### (1) 静的網のシミュレーション

静的網では、ターミナル(プロセッサやメモリなど)間の接続形態が静的に固定されている。よって、静的網のシミュレーションにあたっては、これらの接続形態を単にクロスバー網に写像するだけでよい。しかしこのとき、ターミナル間の最大接続数(次数: degree)が問題となる。なぜなら、

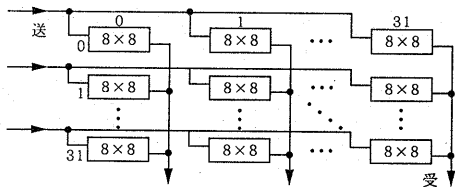


図6. 256×256のクロスバー網

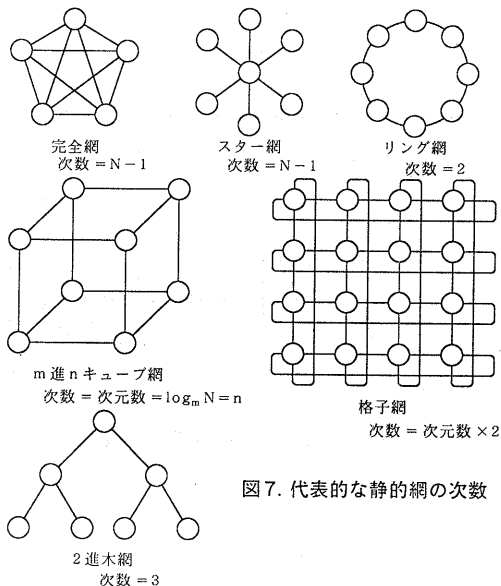


図7. 代表的な静的網の次数

クロスバー網の一時における次数は高々1であるが、一方、現実の静的網は図7に示すように次数が2以上であるからである。

したがって、クロスバー網を多重化する必要性が生じる。これには、

- ① 空間多重化：S個のクロスバー網を物理的に用意して、空間多重度Sとする。
- ② 時間多重化：1個のクロスバー網の使用時間をT分割して、時間多重度Tとする。

の2方法があり、現在、両方法とも取り入れた形で開発を進めている。いずれの方法で多重化したにしろ、1つ1つのクロスバー網をここではプレーン(plane)と呼ぶことにする。つまり、空間多重度Sで時間多重度Tの場合、S×T枚のプレーンが利用できることになる。そして、次数nの静的網のシミュレーションに、n枚のプレーンを用いることにする。図8に、静的網のシミュレーション例を示す。

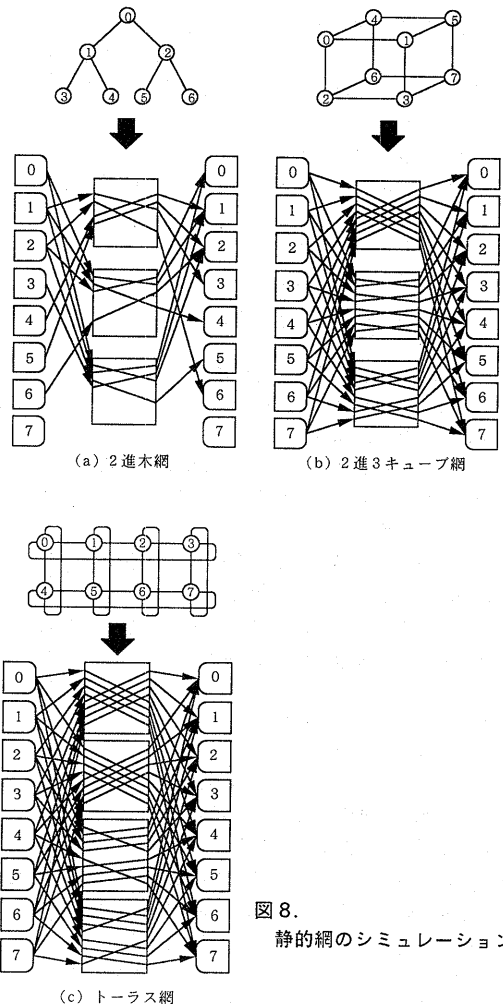


図8. 静的網のシミュレーション例

## (2) 動的網のシミュレーション

動的網では、ターミナル間にスイッチが存在し、これらのスイッチを制御することで動的にターミナル間の接続形態が定まる。したがって、動的網のシミュレーションにあたっては、スイッチをどのように実現するかが課題となる。これには、次の2方法がある。

- ① クロスバー網そのもので動的網全体をシミュレートする(図9(a)参照)。これは、クロスバー網が動的網のスーパーセット的結合網であることから可能であり、集中制御方式かつ回線交換方式の動的網に対しては充分有効である。先の静的網シミュレーションにおける時間多重化の手法がそのまま適用可能である。
- ② スwitchの動作をプロセッシング・エレメントがシミュレートする(図9(b)参照)。これは、分散制御方式あるいはパケット交換方式などを用いた動的網において、スwitchの機能が高度な場合に有効である。ターミナルスswitch間およびスswitchスswitch間の接続形態は静的に固定されているので、これらの接続形態は静的網としてクロスバー網に写像できる。

### 4.2 プロセッシング・エレメント

プロセッシング・エレメント(PE)は図10に示すように、

- ① プロセッサ・ユニット
- ② 共有アクセス・ユニット
- ③ メッセージ通信ユニット
- ④ メモリ・ユニット

の4つのユニットから構成される。

#### (1) プロセッサ・ユニット

プロセッサ・ユニットは、

- ① マイクロプロセッサ (MC68020を使用予定)
- ② コプロセッサ (浮動小数点演算用にMC68811を使用予定)

③ メモリ管理プロセッサ (MC68851を使用予定)の3個のプロセッサを主とするユニットである。メモリ管理プロセッサでは論理アドレス変換を行って、論理アドレス空間を図11に示す物理アドレス空間に写像する。

#### (2) 共有アクセス・ユニット

共有アクセス・ユニットは、

- ① 共有アクセス・リクエスト機構
- ② 共有アクセス・サービス機構

の2つの機構から成る。共有アクセス・ユニットは、プロセッサ・ユニットから共有メモリへの直接アクセスを可能とし、密結合構成のメモリ・イメージを提供するものである。

共有アクセス・リクエスト機構は、物理アドレスとしてグローバル・アドレス ( $2^{31} \sim 2^{32} - 1$ ) が指定された際に起動される。対応するPEへの『共有アクセス』メッセージを作成し、メッセージ送信機構経由で送信する。

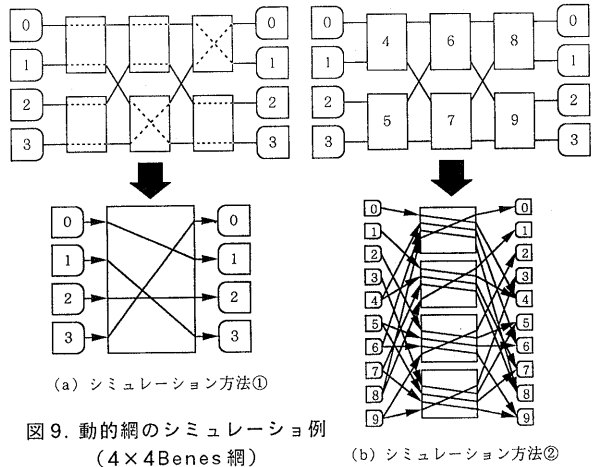


図9. 動的網のシミュレーション例 (4x4 Benes 網)

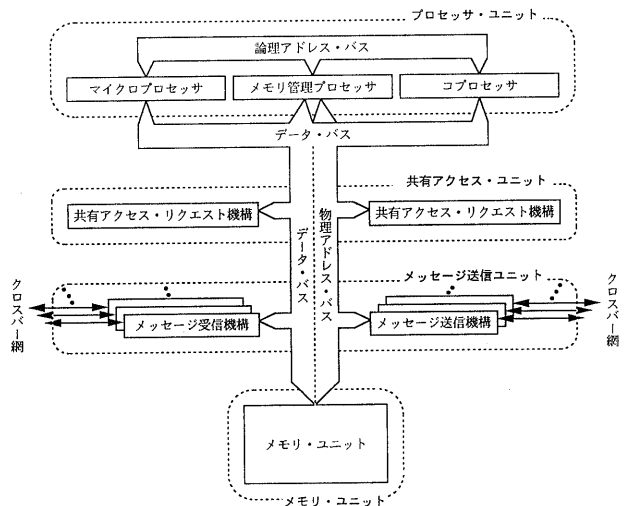


図10. プロセッシング・エレメントの構成

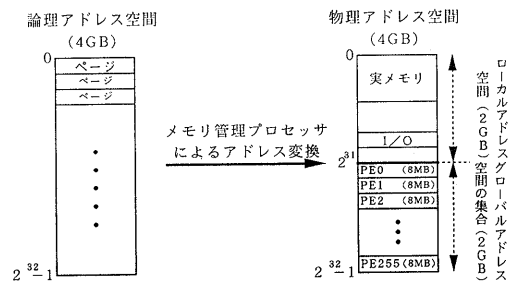


図11. 論理アドレス空間の写像

一方、共有アクセス・サービス機構は、メッセージ受信機構が『共有アクセス』メッセージを受信した際に起動される。メッセージ中に指定されている物理アドレスを図12に示すように再変換した後、メモリへのアクセス要求を出す。このアドレス再変換により、アクセス側のページング管理および被アクセス側の実メモリ管理の単純化を図っている。アクセスのタイプとして読出ししか指示されている場合、読出しデータをメッセージ受信機構経由で返信する。

(3) メッセージ通信ユニット

メッセージ通信ユニットは、

- ① メッセージ送信機構
- ② メッセージ受信機構

の2種類の機構から成る。メッセージ通信ユニットは、PEとクロスバー網とを接続するためのもので、他ユニットからは通常の入出力デバイスと等価に見える。

メッセージ送信および受信機構がそれぞれ1個ずつで対を成し、空間多重化された(物理的に存在する)クロスバー網の数(S個)だけ実装される。また、各々のメッセージ送信および受信機構は、時間多重化された分(T個)のバッファを有する。よって、全体として、プレーン数(S×T枚)に等しい数の送信および受信バッファを有することになる。

共有メモリからのデータ読出しのため、クロスバー網では双方向転送を可能としている。したがって、メッセージ送信および受信機構では、読出しデータの返信の際には逆方向転送を行うようにしている。

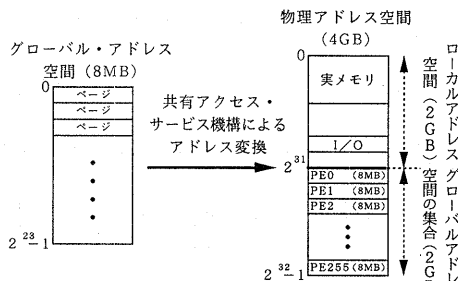


図12. グローバル・アドレス空間の写像

表3. 同期・通信方法

計算機モデル	密結合モデル	疎結合モデル
インタフェース		
並列プログラミング言語	共有変数 セマフォ モニタ	メッセージ通信 手続き呼出し
プリミティブ	LOCK/UNLOCK	SEND/RECEIVE MESSAGE
ゲストマシン・アーキテクチャ	TS (Test&Set) 命令 CS (Compare&Swap) 命令 RM (Read&Modify) 命令	SIGP (Signal Processor) 命令

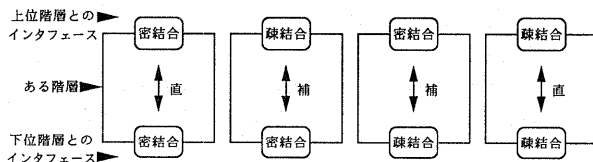
5. ソフトウェアの概要

先に図2で示したように、本システムのソフトウェアは次の4つの階層に大別される。<sup>3)</sup>

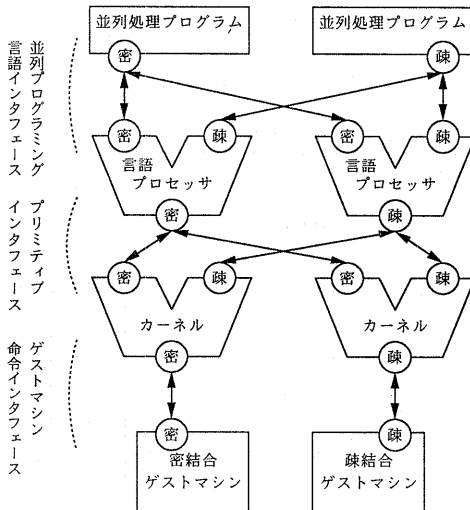
- ① 並列処理プログラム：ゲストマシンで実行される並列処理プログラム
- ② ゲストOS：ゲストマシン用の並列処理OS
- ③ ゲストカーネル：並列処理OSの核部
- ④ ホストカーネル：ゲストマシンを仮想計算機として実現するための仮想計算機モニタ

また、それぞれの階層間のインタフェースを次のように呼ぶ。

- ① 並列プログラミング言語：並列処理プログラムにゲストOSの言語プロセッサ(コンパイラなど)が提供するインタフェースである。
- ② プリミティブ：ゲストOSの各種プログラムにゲストカーネルが提供するインタフェースである。
- ③ ゲストマシン命令：ゲストマシン上のすべてのソフトウェアにゲストマシンが提供する機械命令レベルのインタフェースである。大部分はホストマシンの機械命令と一致するが、残りはホストカーネルによりシミュレートされる。



(a) 計算機モデルの4つの組合せ



(b) インタフェースの可能な組合せ

図13. 計算モデルとインタフェース

現在、以下のように、インタフェースの設計およびソフトウェアの開発を進めている。

- ① 並列プログラミング言語およびその言語プロセッサ
- ② プリミティブとゲストカーネル
- ③ ゲストマシン・アーキテクチャ記述とホストカーネル

ここでは、各種インタフェースの概要およびホストカーネルの処理内容について簡単に述べる。

### 5.1 インタフェース

並列プログラミング言語、プリミティブおよびゲストマシン命令の3種類のインタフェースには、

- ① 密結合型モデル：共有メモリがある。
- ② 疎結合型モデル：共有メモリがない。

という計算機モデルの分類が共通に当てはまる。この計算機モデルの違いは表3に示すように、並行に動作するアクティビティ(タスク、プロセス、プロセッサなど)間の同期・通信方法に影響を与える。

しかも、この2つのモデルは互いに補の関係にあり一方のモデルを基にして他方のモデルを実現できる。よって、図13(a)に示すように、下位インタフェースと上位インタフェースとの間で4通りの計算機モデルの組合せが可能となり、1つのモデルに対して少なくとも2通りの実現方法が存在することになる。そこで、各インタフェースの設計にあたっては、図13(b)に示すように、モデルの実現方法に依らず1つの計算機モデルに基づくインタフェースはすべて共通となるよう考慮している。

### 5.2 ホストカーネル

本システムのホストカーネルは、一般の仮想計算機システムの仮想計算機モニタに比べて、以下の処理に特長がある。

- ① 相互結合網の仮想化
- ② メモリ構成(密結合/疎結合)の仮想化
- ③ プロセッサの仮想化
- ④ マルチプロセッサ用命令の仮想化

表4. ゲストマシンの相互結合網のシミュレーション方法

相互結合網の分類			シミュレーション分担	
トポロジー	交換方式	制御方式	クロスバー網	ホストカーネル
静的網	回線 / パケット	集中	◎	-
		分散		
動的網	回線	集中	◎	-
		分散	○	○
	パケット	集中 / 分散	○	○

◎ : 単独シミュレーション可  
○ : 連携シミュレーション  
- : 不要

このうち、②~④は4.2節で述べたようにPEのハードウェアにより大部分がサポートされており、ホストカーネルは制御テーブル等の設定を行う程度でよい。

しかし、①に関しては、ゲストマシンの相互結合網にかなりの自由度があり、ゲストマシンのクロスバー網のみで正確なシミュレーションを行うには困難な場合がある。したがって、表4に示すように、必要に応じてホストカーネルとクロスバー網とが連携してシミュレーションを行う。

### 6. おわりに

以上、筆者らが開発を進めている可変構造型並列計算機について、そのシステム構想を述べた。現在、ハードウェア設計においては、

- ① 8×8クロスバーLSI
- ② プロセッシング・エレメント(PE)

の詳細を詰めている。また、ソフトウェアとしては、

- ① メッセージ指向の並列処理プログラミング言語の設計、およびSUN-3上での処理系の作成
- ② 密結合および疎結合イメージのプリミティブ・インタフェースの設計
- ③ ホストカーネルの設計

を鋭意進めている。

本システムの実現により、高並列処理方式の研究を多様な側面から推進できるものと考えられる。

### 謝辞

我々とともに、設計・開発を行っている 田中幸二、安富伸浩、濱口一正、石田淳子、尾上琢哉、蒲池恒彦の各氏、および、日頃ご討論いただく富田研究室の皆様へ感謝いたします。

### 参考文献

- 1) 福田ほか：可変構造型並列計算機の概要，昭和62年電気関係学会九州支部連合大会，No.1039
- 2) 安富ほか：可変構造型並列計算機のハードウェア，昭和62年電気関係学会九州支部連合大会，No.1040
- 3) 田中ほか：可変構造型並列計算機の基本ソフトウェア，昭和62年電気関係学会九州支部連合大会，No.1041
- 4) Andrews, G.R. and Schneider, F.B.: Concepts and Notions for Concurrent Programming, ACM Computing Surveys, Vol.15, No.1, 1983, pp.3-43
- 5) 富田眞治：並列計算機構成論，昭晃堂，1986