可変構造型並列計算機の構想

村上, 和彰
九州大学大学院総合理工学研究科情報システム学専攻

森, 眞一郎
九州大学大学院総合理工学研究科情報システム学専攻

福田, 晃
九州大学大学院総合理工学研究科情報システム学専攻

末吉, 敏則
九州大学大学院総合理工学研究科情報システム学専攻

他

https://doi.org/10.15017/17738

出版情報：九州大学大学院総合理工学報告. 10 (3), pp.337-343, 1988-12-31. 九州大学大学院総合理工学研究科
バージョン：
権利関係：

九州大学総合理工学研究科報告
第10巻 第3号 337-343頁 昭和63年12月

Engineering Sciences Reports, Kyushu University
(KYUSHU DAIGAKU SOGORIKOGAKU KENKYUKA HOKOKU)
Vol. 10, No. 3 pp. 337-343 December 1988

可変構造型並列計算機の構想

村上和彰*・森　眞一郎**・福田　晃*
末吉敏則*・富田眞治*

(昭和63年8月31日　受理)

System Philosophy of a Reconfigurable Parallel-Processor

Kazuaki MURAKAMI, Shin-ichiro MORI, Akira FUKUDA
Toshinori SUEYOSHI and Shinji TOMITA

We are now developing a reconfigurable parallel-processor; a multi-microprocessor system with a reconfigurable interconnection network. The reconfigurable interconnection network is a 128×128 corssbar-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[1]．

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，結合形態を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，著者らは，相互結合網を可変構造にして，多様な結合形態を精確にエ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応できる可変構造型並列計算機の開発を進めている[2),3)]．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

*情報システム学専攻
**情報システム学専攻修士課程

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，プロセッサープロセッサ間およびプロセッサー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在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エ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．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[4]．

本システムでは，可変構造の相互結合網に加えて，メモリ形態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，結合形態やメモリ形態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エ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多く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，

i）プロセスのプロセッサへの割付け
ii）データのメモリへの割付け
iii）プロセス実行順序のスケジューリング
iv）プロセス間の同期・通信
v）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，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[1]．これらの相互結合網のエミュレーションを行って，プロセッサープロセッサ間およびプロセッサーメモリ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的挙動を明らかにする．

さらに，上記評価に基づいて次世代の高並列処理計算機の設計に活用する．

3.　システム概観

本システムは，**Fig. 1** に示すように次の2つの並列処理環境を提供する．

i）適応型計算機システム（adaptive machine system）環境：解くべき並列処理問題の構造に最も適した結合形態および物理的メモリ・イメージをシステムが提供する（**Fig. 1 (a)**）．

ii）仮想計算機システム（virtual machine system）環境：本計算機システムをユニバーサル・ホスト計算機として用いることで，種々の並列計算機をエミュレートする（**Fig. 1 (b)**）．したがって，**Fig. 2** に示すように階層構成をとり，システムに存在する2つの計算機をゲストマシン（仮想計算機である並列処理計算機），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，

- 相互結合網：プロセッサープロセッサ間およびプロセッサー—メモリ間の結合形態
- メモリ構成：密結合（共有メモリがある）／疎結

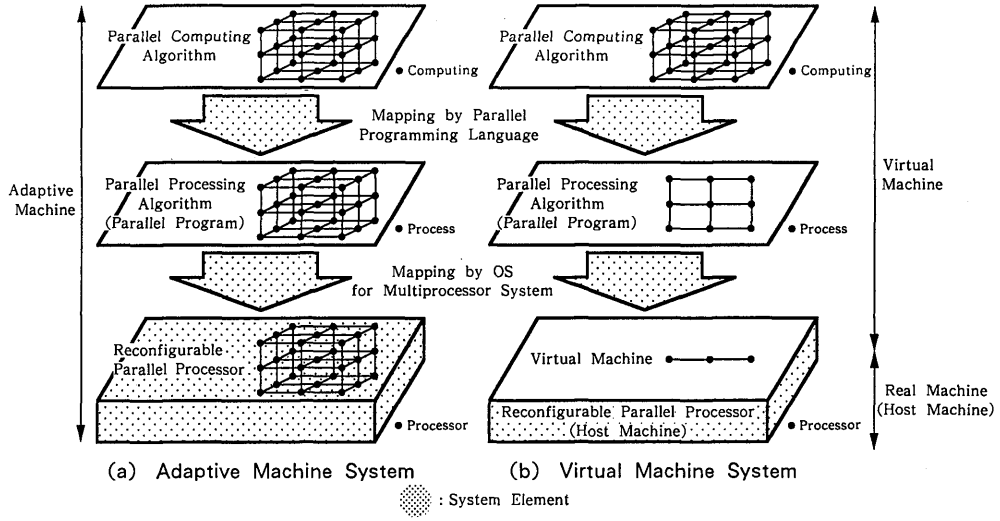


Fig. 1　Parallel Processing Environment.

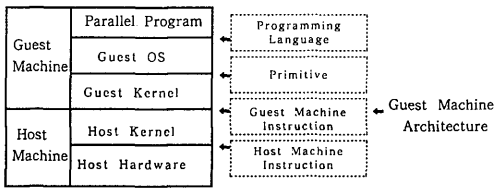


Fig. 2　Layer Structure.

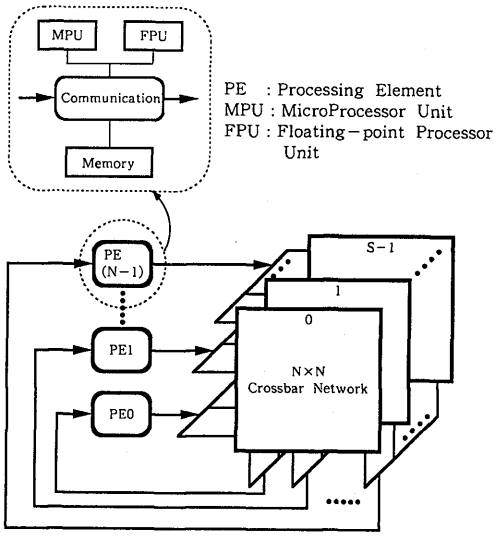


Fig. 3　Configuration of Hardware System.

合（共有メモリがない）構成

- 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
- 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．

以上の2つの環境を提供するためには，本システムのハードウェアとしては次の要件を満たす必要がある．

i）結合形態を可変とするためには，本システムの相互結合網は特殊な構造を持ってはいけない．しかも，多様な結合形態をエミュレーション可能とするため，各種結合網のスーパセット的な相互結合網である必要がある．

ii）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．

iii）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．

iv）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，**Fig. 3** 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．

i）相互結合網に動的網のスーパ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／動的網をエミュレーション可能としている．

ii）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．プロセッサーメモリ接続により，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，プロセッサー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．

iii）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．

iv）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，128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128×128のクロスバー網で接続する計画で開発を進めている．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．

4.1 クロスバー網

128×128のクロスバー網は，**Fig. 4** に示すように，これを8×8のクロスバー網256個（16×16）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・レベルで効率よくエミュレーションできるように考慮してある．

エミュレーションの対象となる項目には，

i）トポロジー

ii）交換方式：パケット交換方式／回線交換方式

iii）制御方式：集中制御方式／分散制御方式

などがある．以下，トポロジーのエミュレーション方法について，簡単に述べる．

(1) 静的網のエミュレーション

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，静的網のエ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，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，現実の静的網は次数が2以上であるからである．

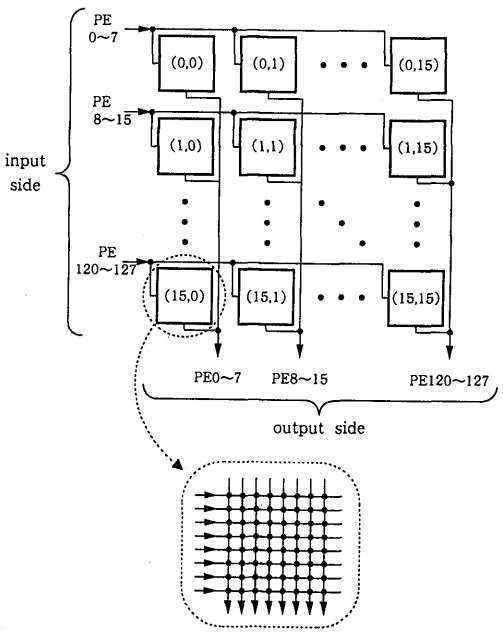


Fig. 4 Configuration of 128×128 Crossbar Network.

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，

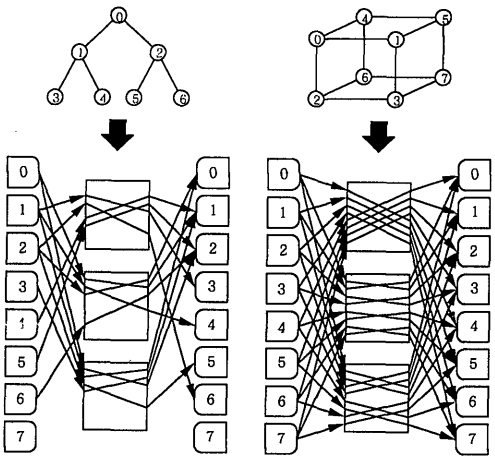
i）空間多重化：S個のクロスバー網を物理的に用意して，空間多重度Sとする．

ii）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の静的網のエミュレーションに，n枚のプレーンを用いることにする．**Fig. 5** に，静的網のエミュレーション例を示す．

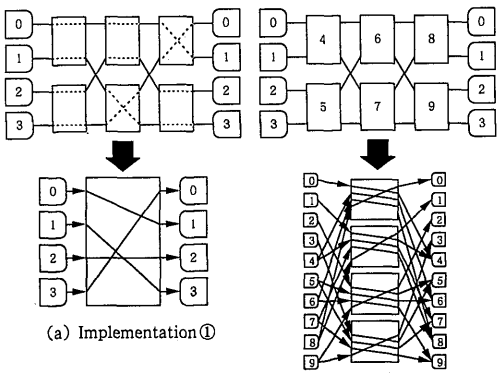
(2) 動的網のエミュレーション

動的網では，ターミナル間にスイッチが存在し，こ



(a) Binary Tree　　(b) Binary 3−Cube

Fig. 5 Implementain of Static Networks.



(a) Implementation①　　(b) Implementation②

Fig. 6 Implementain of Dynamic Networks (4×4 Benes Network).

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，動的網のエ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．

i）クロスバー網そのもので動的網全体をエミュレートする（**Fig. 6 (a)** 参照）．

これは，クロスバー網が動的網のスーパ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．先の静的網エ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．

ii）スイッチの動作をプロセッシング・エレメントがエミュレートする（**Fig. 6 (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，スイッチの機能が高度な場合に有効である．ターミナルースイッチ間およびスイッチースイッチ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．

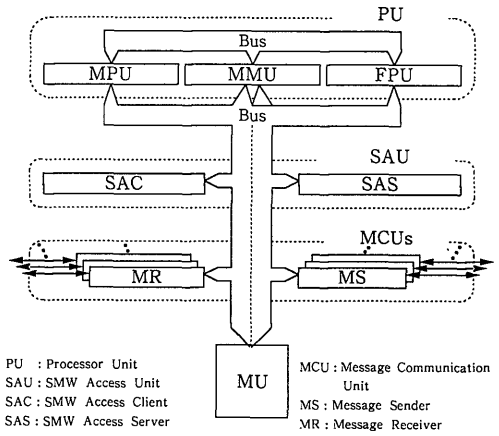


Fig. 7 PE Configuration.

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（以下，PE）は **Fig. 7** に示すように，

i）プロセッサ・ユニット（PU: Processor Unit）

ii）SMW アクセス・ユニット（SAU: Shared-memory-window Access Unit）

iii）メッセージ通信ユニット（MCUs: Message Communication Units）

iv）メモリ・ユニット（MU: Memory Unit）

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（PU）

プロセッサ・ユニットは，

i）マイクロプロセッサ・ユニット（MPU）

ii）浮動小数点演算ユニット（FPU）

iii）メモリ管理ユニット（MMU）

の3つのユニットから構成される．メモリ管理ユニットでは論理アドレス変換を行って，論理アドレス空間を **Fig. 8** に示す実アドレス空間に写像する．

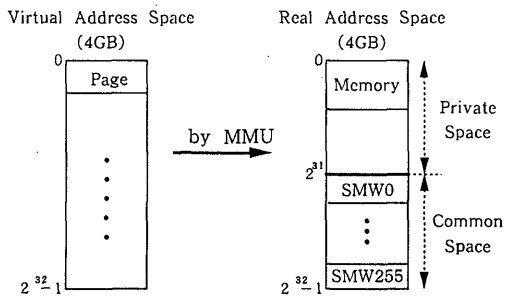


Fig. 8 Address Translation on PEm.

(2) SMW アクセス・ユニット

SMW アクセス・ユニットは，

i) SMW アクセス・クライアント (SAC: SMW Access Client)

ii) SMW アクセス・サーバ (SAS: SMW Access Server)

の2つの機構から成る．SMW 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・ウィンドウ (SMW: Shared-Memory-Window) 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．

SMW アクセス・クライアントは，実アドレスとしてコモン・アドレス (2^{31}～$2^{32}-1$) が指定された際に起動される．対応する PE への「SMW アクセス」メッセージを作成し，メッセージ・センダ (MS) 経由で送信する．

一方，SMW アクセス・サーバは，メッセージ・レシーバが「SMW 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実アドレスを再変換した後，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簡単化を図っている．アクセスのタイプとして読出しが指示されている場合，読出しデータをメッセージ・レシーバ経由で返信する．

(3) メッセージ通信ユニット (MCUs)

メッセージ通信ユニットは，

i) メッセージ・センダ (MS: Message Sender)

ii) メッセージ・レシーバ (MR: Message Receiver)

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・センダおよびレシーバがそれぞれ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・センダおよびレシーバは，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・センダおよびレシーバでは，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．

Table 1 に第一次ハードウェア・システムの諸元を示す．

Table 1 Specifications of Hardware System

Number of PEs (N)	128
Number of Crossbar Networks (S)	1
Switching Elements (N×N)	128×128
Inter-PE Communication	8-bit-width circuit switching
Bandwidth	10 Mbyte/sec. (for each PE pair) 1280 Mbyte/sec. (for total system)
Memory Organization	Distributed Memory Organization
Memory Capacity	4 Mbyte (for each PE) 512 Mbyte (for total system)
PE Configuration	MPU: SPARC MB 86900/10 (16.67 MHz) FPU: WTL 1164/65 (16.67 MHz) Memory: 4 Mbyte SMW Access Unit, Message Communication Unit, etc.
Single PE Performance	10 MIPS 1.6 MFLOPS (single precision LINPACK) 1.1 MFLOPS (double precision LINPACK)
Total System Performance	1.28 GIPS 205 MFLOPS (single precision LINPACK) 141 MFLOPS (double precision LINPACK)

5. ソフトウェアの概要

先に **Fig. 1** で示したように，本システムは適応型計算機システムと仮想計算機システムの2つの並列処

理環境を提供する．**Fig. 2** で示すゲストOSとゲストカーネルを交換することで2つの処理環境を提供する．

適応型計算機システム環境を提供するOSを設計するにあたり，以下の方針をとる．

i）システム資源を効率的に利用する．

システムの性能を最大限に引き出すためには，システムが提供する資源を効率よく利用する必要がある．その中でも特にMPUを効率よく利用することが重要であると考える．

ii）既存のアプリケーション・ソフトの互換性も重視する．

本システムの開発目的の1つは，研究者に並列処理研究の環境を与えることである．従って，既存の多くのアプリケーション・ソフトの互換性をもたせる必要がある．これを実現するため，UNIX† のシステム・コールの仕様を含んだシステム・コールをユーザに提供する．

iii）OSの実現，変更が容易な構造にする．

OSの開発段階で種々の変更が生じる可能性がある．そこで，これらに柔軟に対処できる構造にする必要がある．

現在行っている項目を以下に示す．

i）ホスト・カーネルが提供すべき機能の整理とその実現方法

ii）適応型計算機システム用OSの設計

iii）並列処理プログラミング言語およびその言語プロセッサの開発

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，

i）クロスバー網

ii）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，

i）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

ii）適応型並列処理環境を与える並列／分散OS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る．

謝　辞

我々と共に，設計・開発を行っている濱口，蒲池，廣谷，福澤の各氏，および，日頃ご討論いただく富田研究室の皆様に感謝いたします．

クロスバー・スイッチLSI開発においては㈱東芝・総合研究所・情報システム研究所の小柳主任研究員ならびに田邊氏，PE開発においては富士通㈱本体事業部・電算機第一技術部の内田部長ならびに高村課長，またシステム製作に関してはアジアエレクトロニクス㈱半導体計測事業部・第三開発部の作田部長に，ご助言ご協力をいただいている．慎んで感謝いたします．

なお，本研究は科研費一般研究による．

参考文献

1) 富田眞治：“並列計算機構成論”，昭晃堂（1986）．

2) 村上，福田，末吉，富田：“可変構造型並列計算機の構想” 情報処理学会第47回マイクロコンピュータ研究会資料 87-MC-47-2 (1987)．

3) 村上，田中，安富，福田，末吉，富田：“可変構造型並列計算機のシステム・アーキテクチャ”，情報処理学会「コンピュータアーキテクチャ」シンポジウム，pp. 165-174 (1988)．

4) Andrews, G. R. and Schneider, F. B.: “Concepts and Notations for Concurrent Programming”, ACM Computing Surveys, Vol. 15, No. 1, pp. 3-43 (1983)．

†UNIXは米国ベル研の登録商標である．