マルチプロセッサ型並列処理機械の有用性に関する2つの仮説

On two hypotheses relating to the usefullness of multi-processor systems

有田 五次郎
Itsujiro ARITA

九州大学 工学部
Faculty of Engineering, Kyushu University

1. まえがき

並列処理の研究は計算機の出現と同時に始まった．ビット並列回路や先行制御，チャネルによる並行動作，パイプライン演算器，多重プロセッサシステム，さらにはILLIAC Ⅳのようなアレイ計算機，STARANのような連想計算機，CRAY-Iのようなパイプライン計算機など，多くの並列処理方式・並列処理機械が研究され開発されてきた．[1),2)] しかし，最も汎用性の高い並列処理方式であるMIMD[3](多命令多データ流)型並列処理にはプログラムの並列化の困難さ，メモリアクセス競合による効率低下，同期のオーバヘッド等の問題があり，効率のよい高多重化は困難であるとされてきた．

MIMD型並列処理であるマルチ・マイクロコンピュータシステムについても多くの実験・実用システムが実現されているが，成功例はいずれも多重度が小さいものか応用を限ったものであり，汎用性を持った高多重並列処理システムの研究・開発は今後の問題と言えよう．

VLSI技術の発達により，従来の汎用中型機並みの性能を持った1チップのLSIプロセッサの作製が可能になった現在，計算機の性能向上の手段としての複合化の技術は，益々その重要性を増してくるものと思われる．

我々は既にプロセッサ間の結合方式として階層ループイングバスを，同期手段としてFIFOキューを持つメモリ共有型並列計算機を提案し[4)]，その実験・評価システムとして回線結合の並列計算機HYPHEN C-16を製作した．[5)]

HYPHENシステムは応用を意識しない汎用のメモリ共有型高多重並列計算機である．本稿では，並列処理システムを研究する上での1つの観点 ― 我々の立場 ― を明らかにし，我々の提案しているシステムを概説して，このようなシステムが有用であることを示す2つの仮説を提出する．

2. 並列処理システムに関する一つの観点

現在までに研究され製作されてきた並列処理システムの例でも明らかなように，並列処理システムの研究はアーキテクチャ・ハードウェア・ソフトウェア・応用の各側面から総合的に行われなければならない．

アーキテクチャ・ハードウェアの面からは，並列化の方式や並列処理に適したデータ構造・制御構造，高多重化・LSI化可能な回路構成等の研究が必要であり，ソフトウェアの面からは並列プログラミング言語や並列コンパイラ，並列プログラムを効率よく実行するための並列OS等の研究が重要である．さらに応用の面からは，並列処理に適した問題解法や並列アルゴリズムの理論的研究が必要になる．これらの間には密接な関係があり，1つの並列処理システムを構築するためには，ハードウェアから応用まで，すべての分野にまたがる研究が必要になる．

それでは我々はどこから研究を始めるのがよいであろうか？ また並列処理に関する研究の目的をどこに置くべきであろうか？ ここでは

これらの点に関する我々の見解を示す．

A．専用システムか汎用システムか？

第1点は専用システムか汎用システムかという問題である．

従来の並列処理システム・複合計算機システムは，各応用ごと・各方式ごとに個々に研究されてきた．このように特別の目的を持った専用システムを指向する理由は，並列処理における種々の問題を一般的に解決することは非常に困難であるが，問題の特質に応じた処理方式を採用することによって効率のよい並列処理が実現できると考えられるからである．

しかしこのような方向は，並列処理における共通的な問題を統一的に解決することを困難にし，並列処理システムを特別なものに止めてきた．例えばソフトウェアの分野では，前述のように並列プログラミング言語やその処理系としての並列プログラミングシステムの研究，並列アルゴリズムの研究等が重要であるが，現状では特定の分野ごとに，特定のハードウェアについて研究が行われているにすぎない．

また応用の面から見ると，並列処理の方式が問題の解法に密着しているため，問題の解法が定まらなければシステムを構成することができず，解法ごとに異なるハードウェアを必要とすることになる．問題の解法が確立してあり，処理速度のみが問題になっている応用分野では，システムが高価であることを除いて特に問題はないが，パターン認識や言語処理，人工知能の研究で代表されるような問題の解法を研究している分野では，効果的な並列処理の手段が得られないため問題の解決が困難になり，問題が解決されないから効率のよい並列処理システムが構成できないという矛盾に突き当っているように思われる．

同様のことがシステムの価格についても言える．専用システムは特殊なハードウェア構成を持っているため高価であり，高価であることがそのシステムを他の分野で利用する研究を阻害して普及を妨げ，システムの価格を高価なままにしている．

並列処理の問題は多くの分野で，種々の立場から研究されなければならない．そのためには多くの応用分野に適応できる汎用の並列計算機が必要となる．ここで言う「汎用」は汎用計算機と言う場合の汎用とは多少異なる．並列処理を要求する問題は1つのシステムを占有して処理を行うことが必要なほど高速・大量のデータ処理を含む．従ってこのようなシステムは専用にならざるを得ない．ここで言う「汎用」は，適当な周辺装置を接続し，適当なソフトウェアを開発することによって多くの応用分野で使用可能な，「専用化可能な汎用」を意味する．

B．高性能化か低価格化か？

第2点は高性能化か低価格化かという問題である．

現在の汎用大型計算機で処理可能な問題は汎用の計算機で処理し，その能力を超える問題に対してのみ並列処理を取り入れるべきであるという議論がある．この議論の根底には，1つの並列処理システムを構築するためには多くの経費と労力がかかり，汎用の大型計算機で処理する方が安価であるという考えがある．しかし，汎用大型計算機で処理可能であるといって問題が解決されたと言えるであろうか？

例えば，A/D変換された1時間分の音声データを計算センターに持って行き，大型計算機を1時間使用して認識処理ができたとする．時間遅れを無視すればこれでほぼ実時間の音声認識ができたことになる．時間遅れについては，光通信など高速のデータ通信技術を用いてオンライン化すれば問題はない．実時間の音声認識の問題はこれで解決されたのであろうか？

大型計算機を含む音声認識システムは実用的にはほとんど無意味であろう．物理的にも小さく安価な処理装置が開発され，その上で実時間の認識処理ができて始めて問題が解決されたことになろう．

現在の汎用計算機はその設計思想の上から，バランスのとれたシステムを想定している．従って，演算能力のみが必要だからといって複数台のCPUだけでシステムを構成することはできない．システムを動作させるだけのために，高価なチャネルや高速・大容量のファイル，大量のメモリ，厖大なソフトウェアを必要とする．このことが専用システムとして使用する上で決定的な不利となる．

並列処理システムは現在の計算機で達成でき

ない高性能・高速処理を実現する手段として重要である．それと同時に，より安く・より小さいシステムを要求する分野に対しても有効であろう．この両面を考えることにより並列処理の適用分野は広がり，並列処理に関する研究が進んで，よりよい並列処理システムを実現することができよう．

C．ノイマン型か非ノイマン型か？

次ぎ哭はノイマン型か非ノイマン型かという問題である．

並列処理は本来，アドレスアクセス・逐次処理のノイマン型計算機の能力を超えるものであろう．しかし，パイプライン計算機や複合計算機のようにノイマン型計算機の拡張として構成された並列計算機が存在する．これらは逐次処理の組合せによって並列処理を実現しているもので，疑似並列処理といえる．このような型の並列処理はノイマン型並列処理と呼ぶことができる．

これに対してセルラ論理やデータフロー計算機のように，アーキテクチャからソフトウェアまでノイマン型計算機と異なった並列処理システムがある．これらは本質的な並列処理機械とされ，問題の中に並列性が含まれていればそれを陽に意識しなくても並列に処理される．この型の並列処理は非ノイマン型並列処理と呼ぶことができる．

非ノイマン型並列処理機械は将来の情報処理システムとして非常に重要であろう．特に，パターン認識や言語理解を行う知能モデルはこの型の並列処理システムとして実現されると思われる．しかし，何をハードウェアで実現すべきか，何をソフトウェアと呼ぶべきかも判然としない現状では，問題解決の道具としての並列計算機としてこの型の機械を考えることは適当でない．

一方ノイマン型並列処理は，ハードウェア・ソフトウェア共に従来の計算機の延長上にあるため，現在確立されている技術及びその応用で実現できる．さらにソフトウェアが従来のソフトウェアの拡張になっていることにより，現在使用されている計算機から自然に並列計算機に移行でき，並列計算機の応用分野を広げることが容易である．

D．我々はどこから研究を始めるべきか？

最後に，我々が並列処理システムを研究するとき，アーキテクチャ・ハードウェア・ソフトウェア・応用のどの立場に立つかという問題がある．

勿論，並列処理システムはこれらすべての立場から研究されなければならない．しかしどこから手をつけるのがよいのであろうか．

従来の並列処理システムのほとんどは，まず応用を考え，それに合ったアーキテクチャを定めてハードウェアを作製し，ソフトウェアを開発している．問題を解くために並列処理を考えるのであるからこれは当然である．しかし，このことは効率のよい専用並列処理システムを構築する上では役立つが，並列処理システムを普及させ，並列処理の研究を広げていく上では障害となる．

現在の計算機がこれだけ発達し・普及した大きな要因として，アーキテクチャ・ハードウェア・ソフトウェア・応用の分離があると思われる．基本的ないくつかのアーキテクチャに従ってハードウェアが作製され，そのハードウェアの上で多くの研究者によってソフトウェアの研究・開発が行われ，さらに多くの人々によって種々の応用プログラム・利用法が開発された．応用分野での研究はソフトウェアにフィードバックされ，ソフトウェアの分野での研究成果はアーキテクチャ・ハードウェアに反映されて，よりよい計算機システムが構成されてきた．

並列処理システムの分野においてもこのような研究の分離・分担が必要であり，また可能であると思われる．それによって多くの研究者が種々の立場で並列処理の研究を行うことが可能になる．そのためにはまず，安価で応用範囲の広い並列計算機が必要となる．従って，応用にとらわれない，純粋にアーキテクチャの立場から汎用性のある高多重化可能な並列計算機を構築することが肝要である．

以上の議論から得られる我々のシステムは，次章に示すような，LSIマイクロプロセッサから構成されたモジュール構造のメモリ共有型マルチプロセッサシステムである．このシステムにどのような周辺装置を接続しどのようなソフトウェアを開発するかは今後の問題である．

3. HYPHENシステム

本章では我々が提案している高多重並列計算機HYPHEN システム（highly parallel processing system on hierarchical exchanging network）の特徴について概説する．

3.1 HYPHEN システムの目標

我々の目標は，ソフトウェアが現在のソフトウェアの延長上にある「規模の大小」を問わぬ「専用化可能な汎用」の高多重並列計算機を構築することにある．

図1に「専用化可能な汎用」システムの適応範囲を示す．横軸に規模（性能），縦軸に応用分野をとると，従来の汎用システムは縦方向の線で表され，専用システムは点又は横方向の線で表される．これに対して「専用化可能な汎用」システムはこの平面上の面として表され，その適応範囲は非常に広い．

このようなシステムを構成することは可能であろうか？　また得策であろうか？

もし複合化に要する経費が小さければ得策であることは明らかである．ハードウェアの経費は大量生産によって低減されるので，本当に多くの分野で使用可能であればこの経費は小さくなるであろう．

並列処理システムの基本問題に以下のものがある．

(1) 並列化問題
(2) 同期問題
(3) アクセス競合問題
(4) 高密度実装問題

(1)は並列アルゴリズム等，主として問題解法の理論とソフトウェアに関係するもので，(2)及び(3)はソフトウェアとアーキテクチャに，(4)はハードウェアとアーキテクチャに関係する問題である．アーキテクチャに関係する(2)～(4)の問題についてはすべての並列処理システムに共通の問題であり，応用を限定した専用システムでも応用を意識しない汎用システムでも，解決の困難さは変らないと思われる．

HYPHENシステムは(2)に対処するため，FIFOキューによる高速な同期手段を持ち，(3)，(4)に対処するため多重アクセス可能な階層ルーチングバス（以後H-Rバスと書く）で結合されたモジュラ型のメモリ共有マルチプロセッサシステムである．

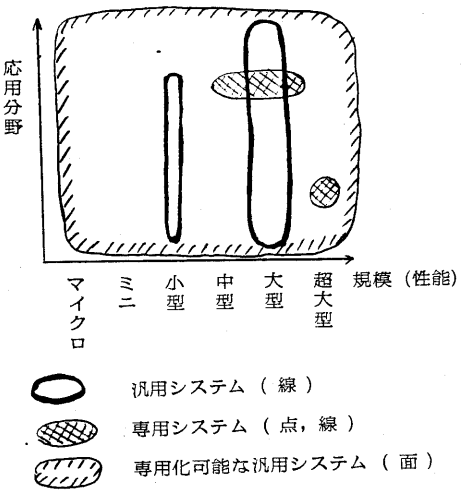
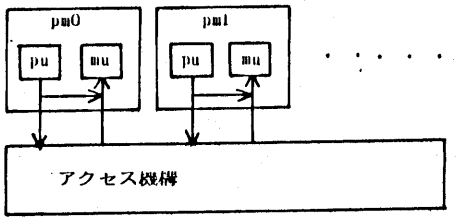


図1　システムの適応範囲



(a) 構造

CPU 0 FIFO	CPU 1 FIFO	CPU 2 FIFO	... ...	CPU F FIFO
00000 ∫ 0FFFF	10000 ∫ 1FFFF	20000 ∫ 2FFFF	... ...	F0000 ∫ FFFFF

(b) アドレス空間

図2　システムモデル

3.2 システムモデル

図2にHYPHENのシステムモデルを示す．このシステムは同図(a)に示すように，プロセッサユニット（以後puと書く．以下同じ）とメモリユニット（muと）とから成るプロセッサモジュール（pmと），アクセス機構によって結合したメモリ共有型並列計算機である．各モジュールはシステムアドレス（SAと）と呼ば

れるモジュール識別番号で識別される．システム内のアドレス空間は図2(b)に示すように，SADとモジュール内のロケーション(LOC)との2次元アドレスで表され，単一のアドレス空間を構成する．各PUは全メモリ空間を参照することができるが，命令フェッチはモジュール内のMUからしか行われない．即ち，各MUは各PUのローカルメモリであると同時に，他のPUから参照可能なグローバルメモリでもある．又 FIFOは実行制御のためのキューであり，これについては次節に述べる．

3.3 待なし並列プログラム

図3(a)の逐次的プログラムはデータの依存関係を考慮して(b)のようなグラフで表現でき，適当な変換操作によって(b)から(c)を得ることができる．ここで矢はオペレーションを，枝は制御の移動を表し，同一列のオペレーションは同一のプロセッサで実行されることを意味する．従って，異なる列の矢に向う枝は異なるプロセッサで遂行されるオペレーションの起動，即ち並列分岐を表しており，(b)，(c)は並列プログラムを表現している．

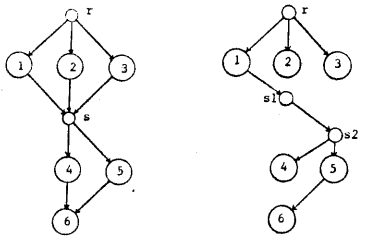
図3(c)のように表現されたプログラムの各列が異なるプロセッサのメモリに分散格納され，各プロセッサが先着順(first come first served，以後FCFSと書く)にそのプロセッサ上のオペレーションを実行しているとすると，(c)の実行結果は(a)の実行結果と等しくなる．このような木構造グラフで表現される並列プログラムを待なし並列プログラム(self synchronizing parallel program，以後SPPと書く)と呼ぶ．SPPは同期操作を持っておらず，同期はFCFSという逐次操作に還元されており高速な同期手段となる．

SPPにおいて，1度制御が渡ると中断されずに実行される同一プロセッサ上のオペレーションの集合をタスクと呼ぶ．SPPの実行単位はタスクで，FCFSによるSPPの実行管理機構は以下のように構成される．

各プロセッサはそのプロセッサ上のタスクの実行順序を制御するため，実行すべきタスクの入口番地を保持するFIFOキューとその上の2つの操作を持つ．

```
①   SX=SIN(X)
②   SY=SIN(Y)
③   CXY=COS(X+Y)
④   Z1=CXY*(SX+SY)
⑤   Z2=SQRT(SX*SX+SY*SY)
⑥   Z=Z1/Z2
```

(a) 逐次的プログラム



(b) 並列プログラム　(c) 待なし並列プログラム

図3　プログラム例

(1) タスク切換(exchange task，EXT)

1つのタスクの実行が完了するとプロセッサは，自分のキューから次に実行すべきタスクを取り出し，その入口番地に制御を渡す．このときキューが空であればプロセッサはアイドル状態となり，次に実行すべきタスクがキューに登録されるのを待つ．

(2) 並列分岐(parallel branch，PB)

他のプロセッサに割当てられたタスクの起動は，そのタスクが割当てられているプロセッサのキューにそのタスクの入口番地を登録することによって実現される．タスクを起動した側のプロセッサは引続き実行を続けることができるので，この操作は並列分岐となる．

SPPにおいては各タスクは他のタスクの発行するPB命令によって実行権を得，自分の発行するEXT命令によって実行権を放棄する．

図3(c)のプログラムの各タスク間の制御の移動の様子とを図4に示す．このプログラムの実行はpm1のr:から始まる．なお各PUはアドレスSAD+LOCによって全メモリを参照することができる．従ってデータはどのMU上にあってもよいが，アクセス競合を少なくするためには参照回数の最も多いプロセッサ上に置くのがよい．

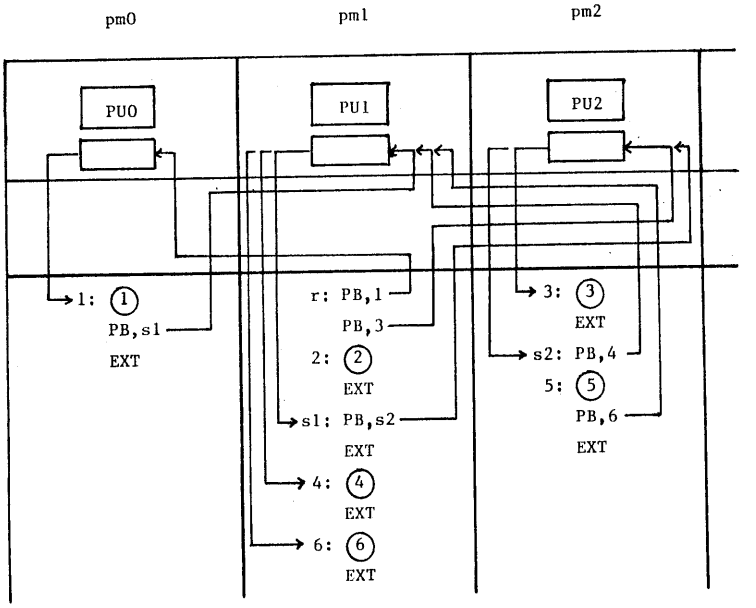


図4
メスクの配置と制御の移動

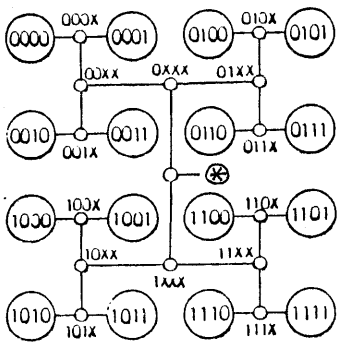


図5 H-Rバス

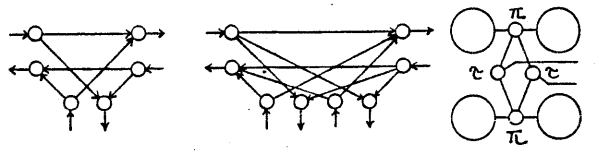


図6 バススイッチ

3.4 H-Rバスの構造

バス上にアドレス情報を持ち，これを利用して行先を切換える情報転送路をルーチングバスと呼ぶ．H-Rバスはバススイッチであるノードを階層的に結合したルーチングバスで，システムを構成するモジュール及びノードはその物理的位置によって図5に示すように2進でアドレス付けされている．ルーチングはこのアドレスに基づいて実行される．ノードは図6(a)のような構造をしており，これを⊂スイッチと呼ぶ．

階層構造は多重化は容易であるが隘路があるため，アクセス競合が大きくまた信頼性にも問題があるとされる．このシステムでは図6(b)に示すπスイッチを用いて，同図(C)のように任意の場所でバスを多重化することができる．

HYPHENシステムはH-Rバス結合のシステムであり，システムの中には，nをモジュールの個数として $2n-2$ 本のバスが存在する．これらのバスは完全に均質であり，システムを任意の場所で切ると互に相手側がメモリに見える．例えば図5で㊀印の所から見ると単なるメモリシステムである．即ち見方を変えれば，このシステムはプロセッサ付きメモリシステムであり，適当にプログラムすることにより任意の機能モジュールとして動作させることができる．しかもその能力は機能モジュールに含まれるプロセッサ数によって自由に増減できる．

表1　バス結合方式の比較

方　式	Smax	Smin	Dmax	Dmin	H
m多重マルチバス	m	m	1	1	m×n
クロスバ・スイッチ	n	n	1	1	n^2
H-Rバス(C結合)	n	2	2(k−1)	1	6(n−1)
H-Rバス(C+2C結合)	n	4	2(k−1)	1	6n−2

$k=\log_2 n$

3.5　H－Rバスの性質

アクセス機構において同時接続が可能なアクセスパスの数をアクセス多重度，アクセスパスの中のバススイッチの数をアクセス距離，全バススイッチの数をハードウェア量と定義し，これらをそれぞれS，D，Hと書く．アクセス機構の良さはこれらによって計ることができる．主要なバス結合方式の簡単な比較を表1に示す．H－Rバス結合はアクセス多重度の点からもハードウェア量の点からもかなり良い性能を持っている．

H－Rバス結合は非常に柔軟な結合方式であり種々の結合形態を実現することができる．例えば4個のH－Rバス結合のシステムを，4個のモジュールを共有させてCスイッチで結合する場合を図7に示す．このような方法でアドレスビットを増加させずにシステム内のプロセッサ数を増すことも可能である．

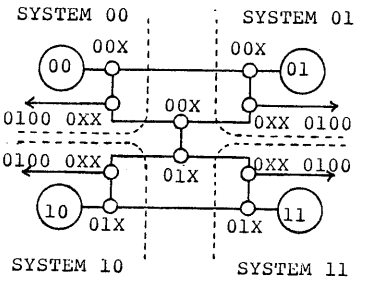


図7　複数モジュールの結合

4. 階層構造マルチプロセッサシステムの有用性を示す2つの仮説

本章では階層構造のマルチプロセッサシステムが汎用性を持ち，有用であることを示す2つの仮説を提出する．

A．分散アルゴリズムの仮説

問題解決の手段について次のことが成立する．

(A1) 我々が現在及び近い将来手に入れることができる機械的な問題解決の手法はアルゴリズムしかない．

(A2) すべての（並列）処理はアルゴリズムとその間の通信によって実現される．即ち，すべての並列処理は分散アルゴリズム――空間分割されたアルゴリズムである．

我々が現在持っている，又近い将来に持ち得る最も効果的な情報処理の手段は，電気・磁気現象を利用したディジタルシステムである．論理回路の組合せであるディジタルシステムは有限状態機械とメモリの直並列接続であり，アルゴリズムを実行する機械の集合体である．

アナログ的並列処理は将来の情報処理方式として免くべからざるものであろう．しかしアナログ的処理は自然現象，或は物質の性質として機能が実現されるものであり，現在このような処理を行う素子又は装置を作製，或は制御することは不可能である．例えば，レンズや空間フィルタを用いた光情報処理が実用的に意味を持つためには，光情報処理に関係する屈折率や透過率を，簡単な手段――例えば電気・磁気的に精密に制御できる物質が作られる必要があろう．従って我々が現在できることは「対応する計算」をいかに速く実行するかということだけになる．

人工知能の分野で多く用いられる「経験的手法」や「学習」はどうであろうか？　これはアルゴリズムとは異なる問題解決の手法である．しかし我々がこのような機械を作製するためには，我々が機械に戦略を与え，機械はその戦略に従って処理をしなければならない．戦略を受理しそれを実行するメカニズムはアルゴリズムとしてしか記述できない．結局我々が何かを論理的に理解できる，或は何かを作ることができるということは，それに対するアルゴリズムを持っているということである．

(A1)における「機械的な」には意味がある．「パターン認識システムを作ることは簡単である．子供をつくればよい．5年もすればすばらしいパターン認識を行うようになる．」という話がある．これは単なる笑話ではなく，生体

情報処理機構の利用，或は生化学的情報処理の利用を示唆している．しかし現状ではこのような情報処理システムを機械的に作る手段はなく，もしそのような手段が得られたとしても実現できることは機能モジュール（器管レベル）のものであり，それらを統合制御する部分はディジタルシステムであろう．

我々が現在できることは，現象をモデル化しアルゴリズムの形でディジタルシステムの上で実行させ，自然界をシミュレートすることだけである．ハードウェアで行うかソフトウェアで行うかは，実行速度と柔軟性とのトレードオフにしかすぎない．

ノイマン型計算機は万能のアルゴリズム実行機械であるという．もしこの仮説を認めるならばノイマン型計算機の複合体であるHYPHENシステムは万能並列処理機械である．勿論この仮説は専用ハードウェアの必要性を否定するものではない．万能アルゴリズム実行機械であるノイマン型計算機に対して並列計算機が要求されると同様に，分散アルゴリズムの実行を専用ハードウェアで行うことが必要になる．この場合にしても，分散アルゴリズムの正しさを検証し，専用ハードウェアの効果を予測するためにはやはり並列計算が必要となろう．またこのような専用ハードウェアは「システム」として構築されるべきものではなく，「機能モジュール」として製作されるべきであろう．

階層構造における隘路は分散アルゴリズムの実行の障害にならないのであろうか？　この疑問に対して次の仮説を用意する．

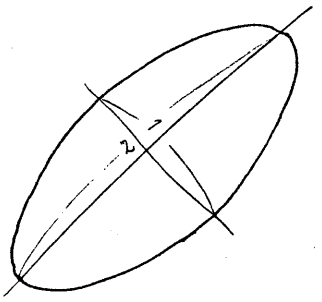
B．モジュール分割の仮説

ある程度大きなシステムに関して次の2つが成立する．

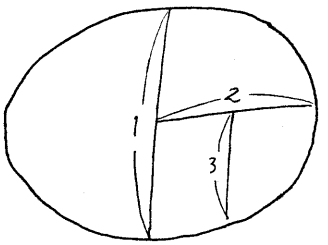
（B1）システムを2分割するときインタフェース巾を最小にする分割が存在する．

（B2）このときのインタフェースの巾はシステムの大きさによらず一定におさえられ，システムが大きくなったからといって無限に大きくはならない．

この仮説の概念を図8に示す．この仮説は自然界における近接作用と遠隔作用との関係から



1のインタフェースは2のインタフェースより巾が狭い．



システムが大きくなると1，2，3のインタフェース巾は同程度となり，無限に大きくなることはない．

図8　モジュール分割

生ずるものであり，すべての自然的或は人工的システムに成立すると考えられるが，問題解法及びプログラムに関しては次のように表現することができる．

B′．局所性の仮説

プログラムが大きくなったときメモリアクセスがシステム全体に均等に広がることはない．必ず局所性，即ち近いメモリにアクセスする率が高くなるような問題解法（プログラム）が存在する．しかも遠くをアクセスする率は問題が大きくなるほど減少し，結局一定値を超えることはない．

前述のように，計算或は情報処理が自然界をシミュレートするものであれば，Bを認めればB′は必然的に成立する．B′が成立するならば後は，一定値であるインタフェース巾をどれだけにとればよいかという問題だけが残る．

Bについて多方の考察を行う．

例えば惑星の運動を考えよう．惑星の運動はたぶん全宇宙の素粒子の振舞によって規定されるのであろう．しかしこのような方法で惑星の運動を説明することは不可能であるし，またその必要もない．これらの間には，素粒子論，量子力学，ニュートン力学などの階層があり，上の階層の現象は下の階層によって説明される．即ちある現象はいくつかの抽象化されたレベルに分けられ，上のレベルは下のレベルによって記述され，階層を構成する．このとき各レベル間のインタフェース（力学の場合は下のレベルで説明される各レベルの物理量）の巾はそれほど広くない．またレベルが上であるからといってインタフェース巾が広くなるわけでもない．

同様な例であるが計算機のプログラムの動作は何によって規定されるのであろうか．これもまた計算機のハードウェアを構成している物質中の電子の動きによって定まるものであろう．しかし我々はCPUとかメモリとかI/Oとかの機能モジュールを対象にしてプログラムを理解している．これらの機能モジュールの動作はマイクロプログラムで説明され，さらに論理回路で説明され，次に電子回路レベルで説明され，素子レベルで説明され，…，というようにこれもやはり階層構造をなしている．このとき，プログラマがハードウェア理解すること，ハードウェア技術者が論理回路を理解すること，IC技術者が電子回路を理解すること等々には，それほど難かしさに違いはないであろう．これは各レベルでのインタフェース巾がほとんど違わないことを意味している．

巨大システムの例として電話網を考えよう．いくつかの理由によってこのシステムはバイパスを持って網構造をしているが，本質的には階層的に構成されている．この場合，あるレベル以上の国内幹線はほとんど同じ容量を持っているであろうし，国際回線の容量もある一定量あれば十分機能するであろう．電話交換はれいのクロスバスイッチによって実現できる．しかしこのシステムだけでは加入者が増加したときその物理的重量に耐えられないであろう．このことは階層構造をとらないシステムはある程度以上大きくなり得ないことを示している．

これを示す他の例として巨大建造物，例えば艦船，航空機，巨大ビル等を考える．これらはすべてビルディングブロック方式で構築され，この方法以外では物理的にも設計上も対応できない．即ち，人智或は自然物が対応できる範囲には限界があり，それを超えるシステムを構築する場合はサブシステムに分割し，階層的にしか構築できない．

社会システムの例として国家を考えよう．これも明らかに階層的に構成されている．このとき，国家を代表し国家を機能させている内閣総理大臣は，国家の形成者である国民一人一人に対して直接のインタフェースを持っているわけではない．このインタフェースは役所の各レベルで統合・整理された形で存在するのであって，外界に対して総理大臣が持っているインタフェースの巾と，国民の1人が外界に対して持っているインタフェースの巾はほとんど違わない．

5．おわりに

以上，並列処理を考える上での我々の立場を明らかにし，階層構造の並列計算機を提案して，このようなシステムが有用であることを示す2つの仮説を提示した．この仮説が正しいかどうかは多くの事例について検証しなければならない．

参考文献

1）加藤，苗村：並列計算機，オーム社

2）特集：スーパコンピュータ，情報処理，Vol.22，No.12（1981）

3）Flynn，M.J.：Very high speed computing systems，proc.IEEE，Vol.54(1966)

4）有田，末吉：階層ルーチンバスとその応用，信学技報，EC81-40（1981）

5）末吉，有田，最所：回線型H-Rバス結合並列計算機 HYPHEN C-16について，本予稿集