

解説



超並列マシンとその応用

2. 超並列マシンの実現技術†

小柳 滋† 田邊 昇†

1. はじめに

デバイス技術の目ざましい進歩により、並列計算機の研究開発は国内、国外を問わず活発となり、最近では超並列と呼ばれるシステムが登場し、話題となっている^{1)~5)}。

超並列マシンはスーパーコンピュータでも達成できない高速の演算速度の実現が期待されると同時に、ニューラルネットワークに代表される並列分散的な情報の表現と処理により、新しい情報処理のメカニズムの実現が期待されている。

本解説では、超並列マシンのハードウェアやソフトウェアに関する基本技術の特徴や、実際に開発されている超並列マシンの具体例を紹介することにより、実現技術について解説する。

2. 超並列モデル

超並列マシンが有効な応用分野は、多くの並列性を内在し、それが自然に表現できる分野である。超並列マシンの登場とともに、新しいモデルが提唱され、それに基づいたソフトウェアが整備されつつある。このような超並列モデルの代表的なものについて説明する。

2.1 ニューラルネットワーク⁶⁾

多数のニューロンの相互作用により、連想や認識などの人間の右脳に相当する機能の実現をめざしたモデルである。各ニューロンの演算は非常に単純であるが、それらを多数個組み合わせることにより複雑な情報処理機能を実現する。したがってプログラミングに相当するのはネットワークの構造決定や学習データの準備であり、いわゆるプログラムレスなモデルである。超並列マシン上で

のニューラルネットの実現については、3.2 (曾根原, 平山) で述べられている。

2.2 *並列オブジェクト指向モデル

データとそれに対する手続きとを一体化したものをオブジェクトと考え、オブジェクト間のメッセージのやり取りにより計算を行うモデルである。自然現象や社会現象のシミュレーションにおいては多数のオブジェクトが並列に動作するようなモデル化が自然であり、超並列処理の枠組みとして本質的に適している。

並列オブジェクト指向モデルとして *Actor モデル⁷⁾や ABCM モデル⁸⁾が知られている。Actor モデルに基づくソフトウェアとしては、Dally による Concurrent Smalltalk⁹⁾ があり、ABCM モデルに基づくものとして ABCL/1⁸⁾ がある。なお並列オブジェクト指向では、問題を構成する要素をオブジェクトとして直接的にモデル化する上では強力であるが、要素間の制約条件や間接的な相互作用をモデル化することが今後の大きな研究課題となっている⁵⁾。今後、より強力なモデル化と、それを実行するソフトウェアが超並列マシンの上に実現されれば、超並列マシン向けの基本ソフトとして重要な位置を占め、応用分野の一層の拡大が期待される。

2.3 *データ並列モデル¹⁰⁾

データの単位ごとにプロセッサを割り当て、全データに対して均一な処理を施すモデルである。従来の連想メモリの考え方を発展させたようなものである。

コネクションマシンではこの考え方を応用して AI やデータベースなど種々の新しい分野を開拓している。コネクションマシンの種々の応用分野については 3.3 (喜連川, 湯浅) に述べられている。

2.4 シストリックモデル

演算器を規則正しくアレイ状に配列し、隣接す

† Realization Technologies for Massively Parallel Machines by Shigeru OYANAGI and Noboru TANABE (TOSHIBA R & D Center).

† 東京総合研究所

*: 用語解説にあることを示す記号

る演算器間でパイプライン的にデータを流しながら並列処理する考え方に基づくモデルで、H. T. Kung らにより提案された¹¹⁾。シストリックとは心臓の収縮を意味する形容詞で、プロセッサの規則的な動作と心臓収縮とのアナログをとっている。シストリックモデルは、VLSI 技術と相性がよく高密度化が可能で、行列計算、信号処理、画像処理などへの応用に適しているが、一般にシストリックアルゴリズムを設計することは、発見的要素が強くかなり困難である。シストリックアルゴリズムについては、文献 12) に詳しく解説されている。

3. 超並列ソフトウェア

超並列マシンの基本ソフトウェアとして、以下のような機能が必要である。

3.1 言語処理系

現状では、商用超並列マシン上でのプログラミング言語としては、C や FORTRAN などの標準言語にメッセージ通信機能を拡張したものや、データ並列用の並列データ型を導入したものが実用化されている。しかし、これだけではユーザの負荷が重く、あるいは適用範囲が限定される。このため、新しい言語処理系の研究開発が進められている。

超並列マシンの言語処理系としては新しい並列言語を用いるアプローチと、従来の逐次処理言語の中から並列性を自動抽出するアプローチとがある。前者については多くの研究がなされているが、2.2 に述べたような並列オブジェクト指向言語は強力な記述能力をもち、並列性が自然に記述できる点から今後の実用化が期待される。

後者については、主に FORTRAN の並列コンパイラの研究が盛んであり¹³⁾、イリノイ大の Parafrese¹⁴⁾ や、IBM¹⁵⁾、早大¹⁶⁾などで行われている。並列コンパイラはプログラム中のデータの流れや制御の流れを解析し、並列性を抽出してそれを各プロセッサに割り当て、実行順序を決定する機能を果たす。並列性のレベルとして、命令レベルの細粒度からサブルーチンやブロックレベルの粗粒度まで種々のレベルでの並列性を抽出する技法が研究されている。また、データフローマシン用の言語処理系については文献 17) に解説されている。

3.2 負荷分散・スケジューリング

超並列マシンを効率よく動かすためには、どのタスクをどのプロセッサでどのような順序で実行させるかという空間時間的なスケジューリングが重要である¹⁸⁾。

スケジューリングの主な手法としては、静的手法と動的手法の二つがある。あらかじめタスクの性質が分かっている場合は、それを利用した静的な手法が有効である。各タスクの実行時間とタスク間の先行関係が与えられている場合は、タスクをノードとする有効グラフで問題を表現できる。このグラフをプロセッサに割り当て、実行順序を決める問題は一般に NP 完全であるが、効率よい近似解法が提案されている¹⁹⁾。また、オブジェクト指向のようにタスク間の実行順序は与えられないがタスク間の関連情報が与えられる場合は、処理コストと通信コストの和の最小化の形で取り扱うことができる。この解法として、グラフ理論²⁰⁾、整数計画法²¹⁾、ニューラルネットワークによる手法²²⁾などが提案されている。

一方、動的な手法としては、負荷を監視し、負荷が少ないプロセッサに新たな負荷を割り当てる方式が提案されている。このため、プロセッサ間結合ネットワークに負荷分散のためのメカニズムを実現する研究が行われている。

3.3 並列デバッグ²³⁾

並列処理のデバッグは逐次処理に比べて、一貫性のある共通データ(状態)がない、関連する情報が多くかつ分散している、並列処理のもつ非決定性により再現性がない、多数のプロセッサを同時に停止させることが困難である、などの理由により、困難である。

並列処理のデバッグに関しては、低レベルデバッグと高レベルデバッグに分類できる。低レベルデバッグ手法としては、再実行性を保証するメカニズム²⁴⁾や、トレース情報のデータベース化²⁵⁾などが提案されている。高レベルのデバッグ手法としては、Assertion を用いる手法²⁶⁾が提案されている。一般に、低レベルのデバッグは、逐次処理マシン上でシミュレートすることにより、モジュール単位のデバッグがある程度可能である。高レベルデバッグに関しては、モジュール間のメッセージの流れを視覚的に表示するアプローチが重要と考えられる²⁷⁾。視覚化の手法としては

time-process diagram やアニメーションがあげられる。特に超並列システムにおいては、データが膨大となるため、システム全体の挙動を分かりやすく表示するために、モデルに応じて抽象化することが重要である。

4. 超並列アーキテクチャ

4.1 アーキテクチャの分類

超並列マシンのアーキテクチャを分類すると、SIMD 型、MIMD 共有メモリ型、MIMD 分散メモリ型、シストリック型、データフロー型の5つに大別できる。以下、それぞれについて特徴を述べる。

(1) *SIMD 型

SIMD 型は、現在、商用超並列計算機で最も成功しているアーキテクチャといえる。このタイプは一つのコントローラから出される共通の制御信号により、すべてのプロセッサが同一または若干の修飾を加えた演算を別々のデータに対して一斉に行う方式である。このためプロセッサを単純化することができ、最もプロセッサ数を多くとれるアーキテクチャといえる。

(2) *MIMD 共有メモリ型

MIMD 型はプロセッサごとに別の命令を実行することができる。このため SIMD と比べて汎用性が高い。MIMD 型は SIMD 型に比べてプロセッサが複雑になるが、既存の汎用マイクロプロセッサを使用することができるという利点もある。

共有メモリの実現法としては、単一バス結合では十数台程度が限界とされており、超並列マシンには向かない。超並列計算機における共有メモリの構成法は次の三つが提案されている。

① 集中型共有メモリ (図-1)

多段結合網のように均一な結合網でプロセッサノードとメモリノードを結合する。Monarch (BBN)²⁹⁾がこれに属する。

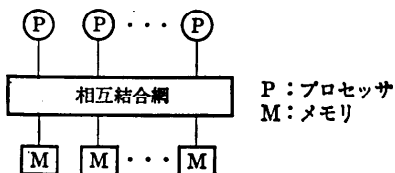


図-1 集中型共有メモリ

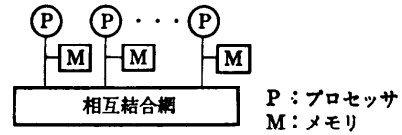


図-2 分散型共有メモリ

② 分散型共有メモリ (図-2)

メモリをもつプロセッサ間を均一な結合網で結合し分散型共有メモリを構築する。このタイプでは通常自分のノード内のメモリアクセスはほかのノードのメモリアクセスより高速である。Butterfly (BBN)²⁹⁾、や RP 3 (IBM)³⁰⁾などがこれに属する。

③ 階層型共有メモリ

クラスタ内部を単一バスやクロスバ網で密に結合し、クラスタ間を疎結合する。このタイプではアプリケーション上の通信の局所性を積極的に利用する必要がある。Ceder (イリノイ大)³¹⁾や PIM (ICOT)³²⁾がこれに属する。

(3) MIMD 分散メモリ型

分散メモリ型ではすべてのプロセッサ間情報伝達はメッセージパッシング方式による。分散共有メモリと単なる分散メモリの違いは、プログラマが自分のノードのメモリとほかのノードのメモリを区別しないでプログラムできる環境が整っているか否かの違いである。

(4) シストリック型

シストリックは隣接間の通信しか基本的には用いないので集積回路向きであり、集積度の飛躍的向上や*ウェファスケールインテグレーション (WSI) 技術の確立によりその有効性が実証されていく可能性をもつ。具体例として SPA (岡山理科大)³³⁾、Systolic/Cellular System (Hughes 研究所)³⁴⁾などがあげられる。

このほかに Intel がカーネギーメロン大学の Warp を LSI 化した iWarp のチップを完成させている³⁵⁾。iWarp はシストリック型の通信以外にメッセージパッシング式の通信もサポートしているため汎用性は高い。

(5) データフロー型

データフロー計算機は演算に必要なデータが揃った時点で演算を行う方式である。プログラムカウンタが不要であるので非ノイマン型アーキテクチャである。

データフロー型の超並列計算機としては、

SIGMA-1³⁶⁾, EM 4 (電総研)³⁷⁾, EDDEN (三洋)³⁸⁾ などがある。データフロー型は命令レベルの並列性が自然に引き出せるという利点もあるが、そのほかのアーキテクチャに比べて結合網の負荷が重いなどの欠点もある。最近では EM 4 などのようにノイマン型との融合を目指す研究が盛んになってきている。

4.2 プロセッサ間結合網

超並列マシンに用いられているプロセッサ間結合網には次の4種類がある。

① キューブ結合

*ハイパキューブ結合網は非局所的な通信に対しても強力な汎用の結合網であり、NCUBE³⁹⁾, iPSC⁴⁰⁾, Connection Machine⁴¹⁾などの商用機で採用されている。しかし実装が必ずしも容易ではないためこれらのマシンではシリアル通信路を用いている。このほかハイパキューブではないが変形キューブ結合のマシンとして、R 256 (NTT)⁴²⁾, ADENA (松下)⁴³⁾がある。これらは変形によりハードウェアを単純化している。

Prodigy (東芝)⁴⁴⁾では、ハイパキューブを拡張した base- m n -cube を提案している。これは、 m 個のプロセッサをクロスバ結合し、これを n 次元構成とする方式である。Base-8 3-Cube の例を図-3に示す。これにより、一般性を失わずに、プロセッサ間距離とポート数を減少させることができる。

② 格子結合

格子結合は実装が容易であるため超並列計算機

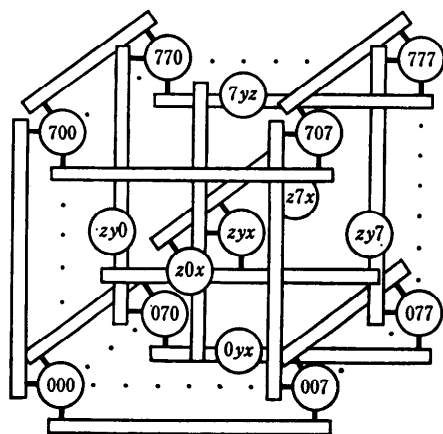


図-3 Base-8 3-Cube⁴⁴⁾

(○はプロセッサ、数字はプロセッサ番号の8進表示)
(□は8×8のクロスバ・スイッチ)

向きである。このタイプに属するマシンとしては古くは DAP (AMT)⁴⁵⁾, MPP (Goodyear)⁴⁶⁾などの主に画像処理をターゲットとしたビットシリアル型の格子結合の商用機がある。国内では AAP (NTT)⁴⁷⁾, MAPLE (富士通)⁴⁸⁾, CAP (三菱)⁴⁹⁾, QCDPAX (筑波大)⁵⁰⁾が格子結合である。

最近では通信部の LSI 化と中継時のバイパスなどを使って遠隔通信を格子結合で高速化しようとする動きがあり、CAP-II (富士通)⁵¹⁾, J-machine (MIT)⁵²⁾はその例である。J-machine ではさらに基板積み重ね実装法を用いて三次元格子結合による通信距離短縮もはかっている。

③ *多段結合網

多段結合網を用いるとプロセッサ間の距離は一定になる。このタイプに属するマシンとして、MasPar 社の MP-1⁵³⁾ は4ビットプロセッサ16384台を格子および3ステージの多段結合網で結合している。このほかに GF 11 (IBM)⁵⁴⁾も多段結合である。

④ 木結合

探索処理や分割統治法は木状に処理が進むので木結合型との整合性が良い。しかし離れた枝の間では通信経路が長くなり、根の部分にアクセスが集中するので適用できる問題は限られる。この型に属するマシンには DADO 2 (コロンビア大)⁵⁵⁾などがある。

5. 超並列マシンの具体例

5.1 国内の具体例

国内で開発されている超並列マシンの主要なものを図-1に示す。表に示すように国内での超並列マシンの開発は活発に行われており、試作レベルから実用レベルのものまで種々のものが含まれている。ここでは紙面の関係上、詳細については各文献を参照されることとし、全体の概要のみについて述べる。

プロセッサ規模の大きいものは、NTTのAAP-2⁴⁷⁾, 富士通のMAPLE⁴⁸⁾, 三菱電機のCAP⁴⁹⁾の3種類があり、いずれもSIMDで2次元格子結合のアーキテクチャである。これらは、画像処理やLSICADなど、大量のデータに対する均質な処理を必要とする分野を狙ったものであり、いずれも1チップ内に16-64個のプロセッサを内蔵することによりシステム全体としてプロセッサ規模

表-1 国内の具体例

名称	開発機関	プロセッサ数	方式	ネットワーク	主な用途	文献
AAP-2	NTT	65536	SIMD	2D格子	画像処理, CAD	47)
MAPLE	富士通	65536	SIMD	2D格子	配線	48)
CAP	三菱電機	4096	SIMD	2D格子	画像処理	49)
PIM/p	ICOT, 富士通	512	MIMD 共有メモリ	バス+キューブ	推論	32)
R256	NTT	256	MIMD 分散メモリ	変形キューブ	デバイスシミュレーション	42)
ADENA	松下電器	256	MIMD 分散メモリ	変形キューブ	偏微分方程式	43)
Prodigy	東芝	512	MIMD 分散メモリ	8進キューブ	AI	44)
可変構造型	九大	128	MIMD 分散メモリ	クロスバ	並列マシンテストベッド	56)
QCDPAX	筑波大	480	MIMD 分散メモリ	2D格子	量子色力学シミュレーション	50)
CAP-II	富士通	1024	MIMD 分散メモリ	2D格子	グラフィクス	51)
SPA	岡山理科大	256	シストリック	リング	数値計算	33)
SIGMA-1	電総研	256	データフロー	クロスバ+オメガ	科学技術計算	36)
EM4	電総研	1000	データフロー	オメガ	記号処理	37)
EDDEN	三洋電機	1000	データフロー	2D格子	画像処理	38)

の大きいアーキテクチャをとり、高速化をはかっている。

共有メモリ型に属するものとしては ICOT と富士通の PIM/p³²⁾があげられる。PIM/p ではクラスタ構成であり、クラスタ内の8個のプロセッサはバス結合の共有メモリ方式で、クラスタ間はキューブ結合されている。

MIMD 分散メモリ方式に属するものとしては、NTT の R256⁴²⁾、松下電器の ADENA⁴³⁾、東芝の Prodigy⁴⁴⁾、九大の可変構造型⁵⁶⁾、筑波大の QCDPAX⁵⁰⁾、富士通の CAP-II⁵¹⁾など多数の例がある。現状では、プロセッサ台数が数百から千台程度であり、ネットワークとしてはキューブ系と格子系に分かれる。応用分野は各システムごとに異なり、数値計算から AI まで多彩に広がっている。特に、CAP-II や QCDPAX では高性能のプロセッサや浮動小数点演算器を用いることにより、システム全体として 10 GIPS, 10 GFLOPS 以上の性能を目指している。なお、ADENA については、3.1(野木)で詳しく説明されているので参照されたい。

シストリック型に属するものとして岡山理科大の SPA³³⁾がある。SPAは256台の1次元アレイ構造をとり、各プロセッサには DSP を用いている。

データフロー型に属するものとしては電総研の SIGMA-1³⁶⁾、EM 4³⁷⁾と三洋の EDDEN³⁸⁾があげられる。特に SIGMA-1 は、世界初の本格的なデータフローマシンとして評価されている。EM4 ではノイマン型とデータフローの融合モデルとして強連結枝モデルを考案し、RISC アーキテク

チャの利点をデータフローマシンに取り入れることにより高速化し、80台のプロセッサで1GIPSの性能を達成している。

5.2 海外の具体例

(1) Monarch (BBN)²⁸⁾(図-4)

Monarch は BBN 社の Butterfly や TC 2000 の後継機種であり、現在開発中である。Monarch はプロセッサ、スイッチ、コンセントレータ、メモリコントローラの4種類のカスタム CMOS LSI とメモリチップにより構成される。

Monarch は多段結合網により最大 65536 台のプロセッサと 32768 台のメモリモジュールを接続する。メモリノードはプロセッサノードと完全にわかれており、どのプロセッサからも等距離に見える。結合網は1本あたり 125 Mbit/s のシリアル通信路により構成される。高速信号を扱うのでデジタル制御のオンチップ終端抵抗・データ遅延補正回路を用いている。

この結合網の特徴はコンセントレータと呼ばれる一種のマルチプレクサによりスイッチステージの間で配線を絞り込み、実装を容易にしている点にある。このように数万台以上規模の超並列計算機では配線を減らす工夫が必須となる。

メモリアクセスは 1 μ s のフレームとよばれる期間にアドレス送出、方向反転、データ返送の三つをプロセッサが一斉に行う。メモリは2バンク構成で読み出しのみ独立に二つの要求を同時に処理できる。

プロセッサは64ビット、ピーク性能は6MIPS, 2MFLOPS (各64ビット)である。メモリバンド

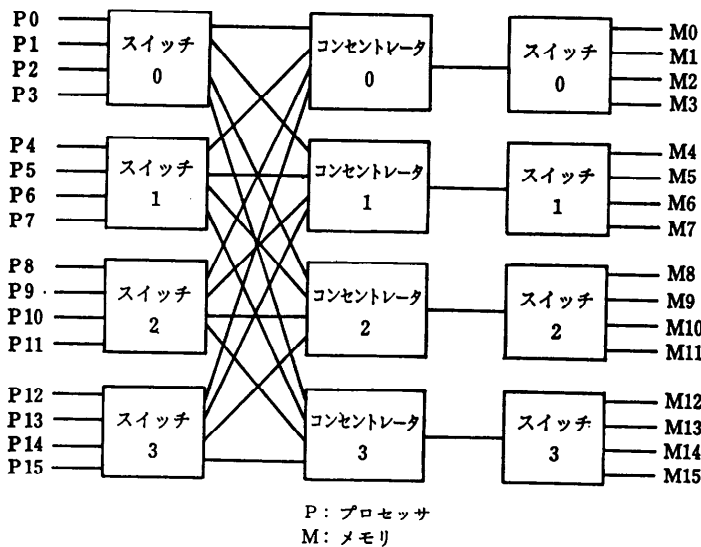


図-4 Monarch の結合網 (16プロセッサの場合)⁴⁾

幅は 8 MB/s と少ないが、64本のレジスタによりメモリアクセスを減らしている。

(2) CM 2 (Thinking Machine)⁴⁾(図-5)

CM 2 は MIT で提案され、シンキングマシン社が開発したコネクションマシンの二号機である。1チップの中に16個の1ビットプロセッサを集積し、4096個のチップ単位でハイパキューブ(binary 12-cube)結合した65536プロセッサ構成の超並列マシンである。プロセッサ数のわり

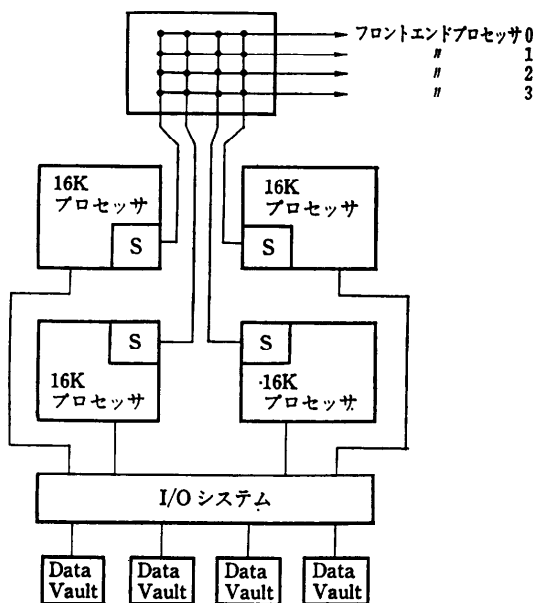


図-5 コネクションマシンのシステム構成⁴⁾

に筐体の大きさは 1.5 m 角と小さい。

コネクションマシンは 16384 プロセッサごとに一つのシーケンサにより独立に制御され、SIMD 動作をする。各プロセッサは同一演算を実行するが、内部状態によりその実行を抑止できる。4個のシーケンサは4×4クロス・パススイッチにより1～4台のフロントエンドプロセッサと接続される。

各チップは16個のプロセッサのほかにルータも内蔵する。このチップの外にメモリチップが付き、プロセッサ当たりのメモリ容量は 256 Kbit である。

CM 2 は、32プロセッサごとに一つのトランスポザ (32プロセッサからのビット列を32ワードに変換する) と浮動小数点演算器が付加されている。

CM 2 では n 次元の NEWS グリッド (格子結合) 通信と一般的なメッセージアドレスによる通信のほかに直接ハイパキューブ通信という通信形態がある。これはチップ内の16個のプロセッサのうち12個を12本の通信線に直接接続して通信の効率化をはかる機能である。基本的な NEWS 通信では平均2本しか通信線が利用されないが、この直接ハイパキューブ通信では12本をすべて使うことができる。

コネクションマシンではプロセッサ数や結合トポロジを仮想化でき、物理的なプロセッサ数より多くのプロセッサがあると思ってプログラムすることができる。プログラミング言語としては、標準言語にデータ並列機能を拡張した *c, *Lisp が提供されている。

(3) MP-1 (MasPar)⁵⁾(図-6)

MP-1 は 1990年1月に米国の MasPar 社により開発された。1チップに32個の4ビットプロセッサを内蔵する CMOS LSI (1.6 μ m, 45万トランジスタ) を512個使用し、8隣接の二次元格子結合 (X-Net) に加え、64×64クロスバ・スイッチ LSI を用いた三段の多段結合網により 16384 個 (128×128) の並列度を実現した超並列計算機で

ある。多段結合網のほうは同時には最大 1024 本のシリアル・リンクしか動作できない。プロセッサ間でランダムに通信させる場合は 300~400 本が並列に動作することになるという。

MP-1 ではプロセッサあたり 16 Kbit のローカルメモリをもち、プロセッサ用 LSI を 32 個とメモリチップを 1 枚の基板に搭載し、基板 16 枚で 16384 プロセッサを実装する。動作周波数は 12.5 MHz であり、4 ビットの加算や、1 ビットの転送を 1 クロック (80 ns) で行う。各プロセッサは隣接プロセッサや多段結合網との通信インタフェースと 48 本の 32 ビット・レジスタ、および簡単な浮動小数点演算器 (32 bit 乗算、加算 80 KFLOPS) をもつ。

プロセッサのアレイはアレイコントロールユニットにより制御され、120 ビットのマイクロコードと 4 ビットのデータをクロックに同期してアレイに放送することにより SIMD 動作する。

プログラミング言語としては、C、FORTRAN-77 に、データ並列用の機能を拡張したものを提

供している。

(4) NCUBE/ten, NCUBE/2 (NCUBE)³⁹⁾

NCUBE/ten および NCUBE/2 は NCUBE 社によって開発された MIMD 型超並列計算機である。NCUBE/ten は 1985 年に開発され、11 組のシリアル・リンクを備える専用 32 ビットプロセッサ LSI により最大 1024 プロセッサを構成するシステムである。NCUBE/2 はその 2 号機にあたり 1989 年に開発された。シリアル・リンク 14 組を備える専用 64 ビットプロセッサ LSI により最大 8192 プロセッサを構成する。

これらの設計思想は一貫しており、メモリ以外の 1 台のプロセッサのすべてを 1 チップの LSI に集積し、各プロセッサをチップ付属のシリアル・リンクで直接ハイパキューブ結合し、各プロセッサはローカルメモリのみを参照して MIMD 動作する、という特徴をもつ。

NCUBE/2 のプロセッサ LSI は 50 万トランジスタ CMOS で 20 MHz で動作する。64 bit CPU (7.5 MIPS) のほかにスカラ型 64 bit 浮動小

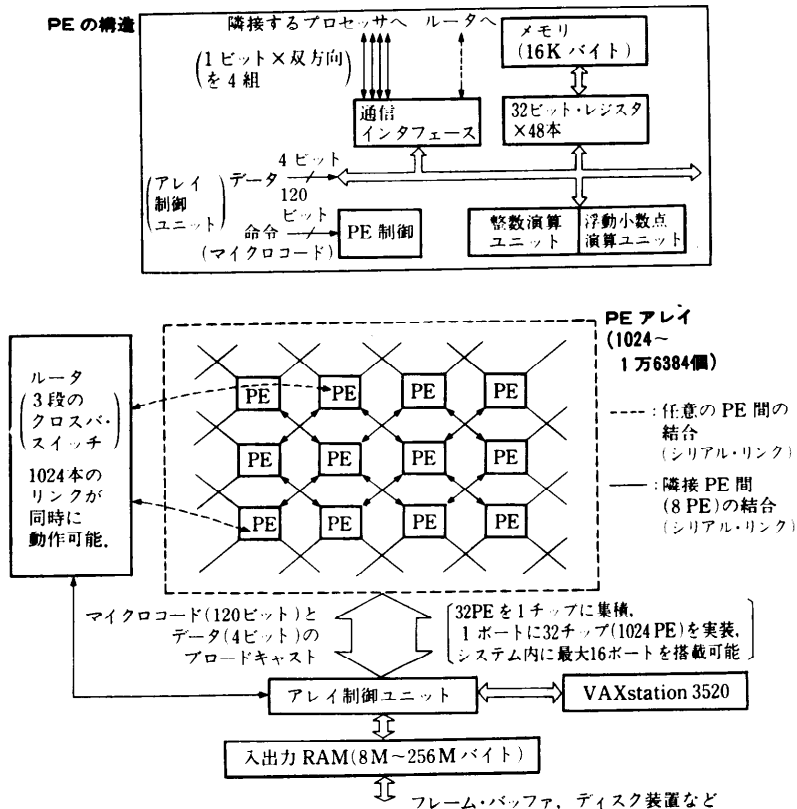


図-6 MP-1 のアーキテクチャ³⁹⁾

数点演算器 (単精度 3.3 MFLOPS, 倍精度 2.4 MFLOPS), 28 チャンネルの DMA コントローラ (2.2 MB/s), メッセージルーティングユニット, 39 bit 幅エラー訂正機能付きメモリ管理インタフェース, 8 ワードのデータキャッシュ, 128 バイトの命令キャッシュを備える。

NCUBE/ten では中継するメッセージはすべていったんメモリに格納 (ストア & フォワードルーティング) しなければならなかったが, NCUBE/2 では中継処理をハードウェアが行うのでメモリに格納する必要がないため, 通信オーバーヘッドは 1/10 になったという。

ローカルメモリは 8192 台の構成時に 4 MB (全体で 32 GB) まで, 4096 台の構成時に 16 MB (全体で 64 GB) まで持つことができる。

NCUBE/2 では以上のようなプロセッサを 1 枚の基板に 64 個搭載する。1 シャーシにプロセッサ基板を 16 枚, 入出力基板を 8 枚収納し, 8 個のシャーシで 8192 台のシステムを実装する。これらはすべてプリント基板を組み合わせてハイパキューブの配線を構成しており, ケーブルは使用しない。

入出力にも同じプロセッサ LSI を用いており, 通常のプロセッサがハイパキューブ結合に使わなかった 1 組のシリアル・リンクを通して入出力プロセッサに接続する。各入出力基板は 16 個のプロセッサ LSI を搭載する。入出力基板 1 枚あたり 569 MB/s の転送能力がありフルシステムでは 36 GB/s の入出力能力をもつ。

プロセッサの演算速度 60 GIPS, 27 GFLOPS, プロセッサ間通信速度 270 GB/s とシステム性能では群を抜いている。プログラミング言語としては, C, FORTRAN-77 にメッセージ通信機能を付加したものが提供されている。

(5) iPSC, iPSC/2, iPSC/860 (Intel)⁴⁰⁾

iPSC は 1985 年, iPSC/2 は 1987 年, iPSC/860 は 1989 年に Intel により開発された。これらはすべて, その時点で最高の性能をもつ Intel 製の汎用マイクロプロセッサ (i286→i386→i860) を用いて, 128 台のプロセッサをシリアル・リンクによりハイパキューブ結合したものである。通信部に関して初代 iPSC が汎用の LAN コントローラ (10 Mbit/s) を 8 個用いて構成し, 2 代目以降はプログラマブルゲートアレイ (LCA) を用いて構

成している。

初代 iPSC は NCUBE/ten 同様ストア & フォワードルーティングしなければならなかったが, 2 代目以降はダイレクトコネクとと呼ばれる中継処理をハードウェアが行うので通信が高速化した。LCA の使用とクロックの向上により最大転送速度も 2.8 MB/s と向上した。

iPSC/2 では最大 64 台のノードプロセッサを並列動作可能な浮動小数点加算器, 乗算器 (各 10 MFLOPS) をもったベクタノードに交換することで数値演算能力を 1.28 GFLOPS に向上させた。iPSC/860 では i860 に浮動小数点演算器 (単精度 80 MFLOPS, 倍精度 60 MFLOPS) がすでに内蔵されているため, 128 台すべてが数値演算能力が高く, 倍精度でも最大 7.68 GFLOPS の演算性能を実現している。現在 Intel は DARPA の援助を受け Touchstone プロジェクトを進めており, iPSC/860 はその第一ステップであり, 1991 年には 2000 プロセッサ構成の機種を開発すると言っている。さらに 1994 年には 1000 GFLOPS のマシンを開発することを目標にしている。

プログラミング言語としては, C, FORTRAN などの標準言語にメッセージ通信機能が拡張されたものが提供されている。

(6) J-machine (MIT)⁵²⁾

J-machine は MIT の W. J. Dally らのグループによって現在開発中の並列計算機である。1990 年中ごろには 4096 台 (16×16×16) のプロセッサが 3 次元格子結合されるプロトタイプが完成する予定といわれている。

各プロセッサは 36 bit 幅 (32 bit: データ, 4 bit: タグ) 4 K 語のメモリと 6 本の通信チャンネルを内蔵する。このチップを並べるだけでシステムが構築できる。各チャンネルは 9 bit 幅で 1 語を 4 回に分けて転送し, チャンネルあたりの転送速度は 450 Mbit/s と高速である。転送に用いられるクロックは 50 MHz と高速であり, Monarch と同様に J-machine でもプログラマブルディレイラインを用いてクロックスキューを抑える。

3 次元格子結合は 2 次元の場合と異なり 2 次元格子平面間を接続する多数の配線が必要であり, J-machine のように単方向のチャンネルあたり 15 本もの配線を使用して 1 枚の基板に多数のプロセッサを搭載するならば, 従来の技術では実装はきわ

めて困難になる。そこで J-machine では上下方向の基板間接続に細い導線を多数通したゴム製の板を用い、基板で挟みこみ圧着させて接続するという。

メッセージは8語長の FIFO に書き込むと自動的に送信されるようになっている。FIFO が空の状態では4096台のシステムで6語(216ビット)のメッセージを結合網の半径($D=24$)離れたプロセッサに送るのにわずか960 ns の遅延時間しかかからない。メッセージを配送し、メッセージ到着に応じてタスクを処理するまでの時間が3 μ s と少なく、オーバヘッドを極力抑えた設計になっており fine grain 向きである。

(7) XLNT (Cogent Research)⁵⁾(図-7)

XLNT (excellent) は1988年9月発表のマシンで、デスクトップスーパーコンピュータと称している。各プロセッサはトランスピュータ T800 (10 MIPS, 1.5 MFLOPS), ハイブリッド通信インタフェース, 4 MB の DRAM からなり, 10 cm \times 30 cm 程度の基板(コンピュータカード)に2プロセッサを搭載する。この基板を38 \times 48 \times 16 cm³ の小さな筐体(リソースサーバ)に15枚のコンピュータカードと1枚のコミュニケーションカードを収納する。リソースサーバ内にはクロスバ・スイッチ LSI 4個からなる128ウェイクロスバ・スイッチと, 32ビットバスの2種類の結合網がある。筐体間は37 MB/s の転送速度をもつ光ファイバリンクで結合し最大1900ノードまで拡張できるという。

ソフトウェアとしては, UNIX 互換の OS 上で Linda サポートによる C, C++, FORTRAN 77 の並列環境がサポートされている。

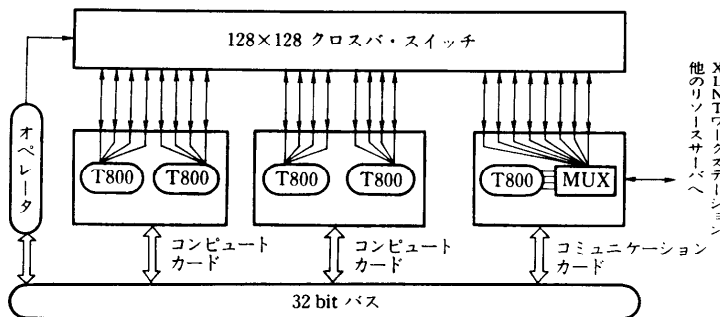


図-7 XLNT の全体構成⁵⁾

(8) iWarp (Intel, CMU)³⁵⁾

iWarp は CMU で開発されたプログラマブルシストリックアレイである Warp を Intel が LSI 化したものであり, 1989年12月にチップが完成した。1990年秋にはプロセッサ64台(8 \times 8)が2次元トラス結合された3台のデモンストレーションシステムを開発する予定である。1024台までは容易に拡張でき, そのときの性能は20 GFLOPS, 160 GB/s という。

iWarp のチップは65万トランジスタを集積し, 20 MIPS, 20 MFLOPS (単精度) の性能をもつ。チップは20 MHz で動作するが転送周波数は40 MHz である。4本の入力バスと4本の出力バスを備え, 合計320 MB/s の転送速度をもつ。1バイト幅の通信路, 40 MHz のクロックで40 MB/s を実現するために, iWarp では単純なハンドシェイクは行わず, 受信側の空きバッファ量を示すカウンタを操作しながら転送を制御する。

iWarp チップにメモリを外付けすることによりセルを構成することができ, これらをアレイ状(線形アレイ, 2次元メッシュ, 斜めが単方向の六角格子など)に結合するだけでシステムを構築できる。ローカルメモリのバンド幅は160 MB/s であり, 通信制御部と計算部の間も40 MB/s の入力2本出力2本(計160 MB/s)で接続される。

iWarp はシストリックアレイとして動かせることは当然として, メッセージパッシング方式でも動作させることができるため汎用性が高い。

iWarp の論理チャンネル機能は, 物理的なチャンネル数よりも多くのチャンネルをアレイ上にマッピングすることを可能にする。チャンネルは時分割されているので, メッセージのクラスに応じたバンド幅が保証される。この性質がシストリックで重要なのは自明であるが, メッセージ交換型で動かす場合でもシステムメッセージをタイムリに転送できるなどの効果がある。

6. おわりに

超並列マシンの実現技術として, 関連する幅広い技術的課題と事例の紹介を行った。紙面の関係および, 著者の浅学のため表面的な解説にとどまった。詳

細は、各参考文献を参照されたい。

最後に、今後の超並列マシンに対する私見を述べたい。超並列アーキテクチャとして、MIMD分散メモリのように高機能プロセッサを用いて構成する道と、SIMDのようにプロセッサアーキテクチャを簡略化してさらに大規模化する道の二つに分化すると考える。半導体技術の進歩により、前者では数万台、後者では数百万台のプロセッサをもつ超並列マシンが数年で実現可能となるであろう。さらに大規模化するためには WSI (Wafer Scale Integration) や光技術など新しいデバイスや実装技術をアーキテクチャとしてどう取り込むかがキーとなる。すなわち、現在の実装法による種々の限界をいかに乗り越えるかが重要な問題である。また、現状規模ではあまり問題とはなっていないが、さらに大規模化する場合は高信頼化対策、消費電力や冷却の問題が深刻となる。

一方ソフトウェア的には超並列向けのプログラミング環境を整備することが急務である。超並列マシンの応用分野としては 3. に示されているように、科学技術計算、ニューラルネットワーク、シミュレーション、データベースなどの大規模計算問題が有望であり、これらの分野でスーパーコンを大幅に上回る性能が実証されつつある。超並列マシンの実用化にとって、これらの分野のユーザがスーパーコン程度に超並列マシンを使いこなせるような環境を整備することが当面の課題である。さらに、超並列マシンの開発を促進するために、より広範な情報処理分野への適用を考える必要がある。そのためには、新しい並列モデルやパラダイムの研究、複数のパラダイムの統合化の研究、および大規模な計算能力を生かした新しい機能を開拓するアプローチが必要である。

現在、通産省では、超並列マシンを一つの柱とした新情報処理技術に関する調査研究が行われている⁵⁷⁾。この調査研究では、超並列超分散処理に基づいて人間に近い柔らかな情報処理能力を実現することを目指している。このように超並列マシンが単に性能向上だけでなく、膨大な処理能力を生かして新しい情報処理へと発展することを期待したい。

参考文献

- 1) 小池：超並列マシン，情報処理，Vol. 28, No. 1, pp. 94-105 (Jan. 1987).
- 2) 富田：並列計算機構成論，昭晃堂 (1986).
- 3) 富田，末吉：並列処理マシン，オーム社 (1989).
- 4) 並列コンピュータ・アーキテクチャ，bit 臨時増刊，Vol. 21, No. 4 (1989).
- 5) 日経エレクトロニクス：10年後の主役をめざす超並列マシン，No. 478, pp. 123-146 (July 1989).
- 6) 小池：ニューロコンピューティングのためのハードウェア，信学論，Vol. J 73-D 2, No. 8, pp. 1132-1145 (Aug. 1990).
- 7) Agha, G.: Actors: A Model of Concurrent Computation in Distributed Systems, MIT Press (1986).
- 8) Yonezawa, A.: ABCL: An Object-Oriented Concurrent System—Theory, Language, Implementation and Application—, MIT Press (1990).
- 9) Dally, W. J. and Chien, A. A.: Object Oriented Concurrent Programming in CST, SIGPLAN notices, Vol. 24, No. 4 (1989).
- 10) Hillis, W. D. and Steele, S. L. Jr.: Data Parallel Algorithms, Comm, ACM, Vol. 29, No. 12, pp. 1170-1183 (Dec. 1986).
- 11) Kung, H. T.: Why Systolic Architecture?, IEEE Computer, Vol. 15, No. 1, pp. 37-46 (Jan. 1982).
- 12) 梅尾：シストリック・アレイ，情報処理，Vol. 30, No. 1, pp. 15-28 (Jan. 1989).
- 13) 本多：マルチプロセッサシステムのためのコンパイラ技術，情報処理，Vol. 31, No. 6, pp. 744-752 (June 1990).
- 14) Polychronopoulos, C. D. et al.: Parafraze-2, Proc. ICPP, pp. 1139-1148 (Aug. 1989).
- 15) Allen, F. et al.: A Framework for Determining Useful Parallelism, Proc. 2nd ACM Int. Conf. on Supercomputing (1988).
- 16) 本多，岩田，笠原：階層型マルチプロセッサシステム OSCAR 上での Fortran の並列処理手法，並列処理シンポジウム JSPP-89, pp. 251-258 (Feb. 1989).
- 17) 関口，山口：データ駆動計算機用的高级言語と処理系，情報処理，Vol. 31, No. 6, pp. 753-762 (June 1990).
- 18) 坂井：並列計算機におけるスケジューリングと負荷分散，情報処理，Vol. 27, No. 9, pp. 1031-1038 (Sep. 1986).
- 19) 笠原，成田：マルチプロセッサ・スケジューリング問題に対する実用的な最適及び近似アルゴリズム，信学論，Vol. J 67-D, No. 7, pp. 792-799 (July 1984).
- 20) Bokhari, S. H.: Assignment Problems in Parallel and Distributed Computing, Kluwer Academic Publishers (1987).
- 21) Chu, W. W. et al.: Task Allocation in Distributed Data Processing, IEEE Computer, pp. 57-69 (Nov. 1980).

- 22) 田中, Canfield, 小柳, 玄地: ニューラルネットを用いた最適ジョブ割当て, 信学会研資, MBE 88-50 (Sep. 1988).
- 23) Mcdowell, C. E. and Helmbold, D. P.: Debugging Concurrent Programs, ACM Computing Surveys, Vol. 21, No. 4, pp. 593-622 (1989).
- 24) Leblanc, T. J. and Mellor-Crummey, J. M.: Debugging Parallel Programs with Instant Replay, IEEE Trans. on Comp, Vol. C-36, No. 4, pp. 471-482 (Apr. 1987).
- 25) Garcia-Molina, H., Germano, F. Jr. and Kohler, W. H.: Debugging a Distributed Computing System, IEEE Trans. Soft. Eng., Vol. SE-10, No. 2, pp. 210-219 (1984).
- 26) Bates, P. and Weleden, J. C.: An Approach to high-level Debugging of Distributed Systems, SIGPLAN notices, Vol. 18, No. 8, pp. 107-111 (Aug. 1983).
- 27) 小柳, 藤田, 鈴岡: 並列 AI マシン Prodigy の視覚的デバッグ (VDB), 第 38 回情報処理学会全国大会, pp. 1558 (1989).
- 28) Rettberg, R. D. et al.: The Monarch Parallel Processor Hardware Design, IEEE Computer, pp. 18-30 (Apr. 1990).
- 29) Crowther, H. et al.: Performance Measurements on a 128-node Butterfly Parallel Processor, Proc. ICPP, pp. 531-540 (1985).
- 30) Pfister, G. F. et al.: The IBM Research Parallel Processor Prototype (RP 3): Introduction and Architecture, Proc. ICPP, pp. 764-771 (1985).
- 31) 小原: 階層構造の MIMD 型スーパーコンピュータ情報処理, Vol. 25, No. 5, pp. 480-490 (May 1984).
- 32) 服部, 篠木, 久門, 後藤: 並列推論マシン PIM/p のアーキテクチャ, 情報処理学会論文誌, Vol. 30, No. 12, pp. 1584-1592 (Dec. 1989).
- 33) 小畑: シグナルプロセッサアレイ-SPA-情報処理学会計算機アーキテクチャ研究会資料 79-13 (Nov. 1989).
- 34) Przytula, K. W. and Nash, J. G.: Implementation of Systolic Aperture Radar Algorithms on a Systolic/Cellular Architecture, Int. Conf. Systolic Arrays, pp. 21-30 (1988).
- 35) Borkar, S. et al.: Supporting Systolic and Memory Communication in iWarp. Proc. ICCA 90, pp. 70-81 (1990).
- 36) Shimada, T., Hiraki, K., Nishida, K. and Sekiguchi, S.: Evaluation of a Prototype Data Flow Processor of the Sigma-1 for Scientific Computations, Proc. ISCA, pp. 226-234 (1986).
- 37) 児玉, 坂井, 山口: データ駆動型シングルチッププロセッサ EMC-R, 並列処理シンポジウム JSPP 90, pp. 161-168 (1990).
- 38) 三浦, 川口, 田中, 大橋, 清水, 森: データ駆動計算機 EDDEN のアーキテクチャ, 並列処理シンポジウム JSPP 90, pp. 169-176 (1990).
- 39) Hayes, J. P. and Mudge, T.: Hypercube Supercomputers, Proc. IEEE, Vol. 77, No. 12, pp. 1929-1841 (Dec. 1989).
- 40) 阿部: ハイパーキューブ・マルチプロセッサ, bit, Vol. 21, No. 5, pp. 640-651 (Apr. 1989).
- 41) ダニエル・ヒリス: コネクションマシン, パーソナルメディア (1990).
- 42) Fukazawa, T. et al.: R 256 a Research Parallel Processor for Scientific Computation, ISCA, pp. 344-361 (1989).
- 43) Kaneko, K. et al.: Processing Element Design for a Parallel Computer, IEEE Micro, pp. 26-38 (Apr. 1990).
- 44) 鈴岡, 田辺, 中村, 藤田, 小柳: 並列 AI マシン Prodigy の相互結合網の評価, 信学論, Vol. J 71-D, No. 8, pp. 1496-1501 (Aug. 1988).
- 45) 湯浅, 青柳: 超並列計算機 DAP, bit, Vol. 21, No. 10 (Sep. 1989).
- 46) Batcher, K. E.: Bit-Serial Parallel Processing System, IEEE Trans. Comp., Vol. C-31, No. 5, pp. 377-384 (May 1982).
- 47) 渡辺, 近藤, 北村, 杉山: 大規模セルラアレイプロセッサ: AAP-2, 情報処理学会研究会資料 76-3 (May 1989).
- 48) 河村, 進藤, 渋谷, 三渡, 大木: 超並列配線マシンの概要, 第 41 回情報処理学会全国大会, 6-117 (Sep. 1990).
- 49) 磯西, 宮田, 岩瀬: セルラアレイ型大規模並列プロセッサのアーキテクチャ, 情報処理学会計算機アーキテクチャ研究会資料 73-9 (Oct. 1988).
- 50) 小柳, 他: 並列計算機 QCDPAX による QCD シミュレーション, 並列処理シンポジウム JSPP 90, pp. 345-352 (1990).
- 51) 石畑, 稲野, 堀江, 清水, 加藤: 高並列計算機 CAP-II の構成とメモリスistem, 情報処理学会計算機アーキテクチャ研究会資料 83-37 (July 1990).
- 52) Dally, W. J. et al.: The J-machine: A Fine-Grain Concurrent Computer, Proc. IFIP 1989, pp. 1147-1153 (1989).
- 53) 日経エレクトロニクス, No. 503, pp. 102-103 (June 1990).
- 54) Beetem, J., Denneau, M. and Weingarten, D.: The GF 11 Supercomputer, Proc. 12th ISCA, pp. 108-115 (1985).
- 55) Stolfo, S. J.: Initial Performance of the DADO 2 Prototype, IEEE Computer, pp. 75-83 (Jan. 1987).
- 56) 村上, 福田, 末吉, 富田: 可変構造型並列計算機のアーキテクチャ, 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).
- 57) 日本情報処理開発協会: 新情報処理技術調査研究委員会報告書 (Mar. 1990).

(平成 2 年 12 月 20 日受付)

用語解説

Actor モデル

メッセージパッシングを基礎とする並列計算モデルであり、MIT の Carl Hewitt により提唱された。処理の構成要素すべてを Actor として統一的に表現する。

ウェファスケールインテグレーション WSI: Wafer Scale Integration

半導体ウェファをチップに切断せずに、ウェファのまま使用する技術。高密度の実装が可能となるが、通常ウェファ内に欠陥を含むため、高信頼化技術が課題である。

SIMD (Single Instruction stream/Multiple Data stream)

複数のプロセッサが1つの命令列により制御される方式の並列計算機アーキテクチャ。比較的大規模化しやすく、単純な数値処理や画像処理に適する。

MIMD (Multiple Instruction stream/Multiple Data stream)

各プロセッサが独自の命令列により制御される方式の並列計算機アーキテクチャ。汎用性が

高く、複雑な並列処理が可能な方式である。

多段結合網

小規模のスイッチを多段に並べて大規模なスイッチを実現する構成の結合ネットワーク。オメガ網、Clos 網、パンヤン網など各種の構成法がある。

データ並列

大規模なデータの処理において、データを多数のプロセッサに分散配置し、これらを並列に処理する並列処理モデル。マルチプロセッシング(制御並列)に対比する考え方として提唱された。

ハイパーキューブ

ノードに m 進数の番号を付け、1桁だけ番号が異なるノード間にリンクを設ける結合トポロジ。通常は z 進数 ($m=2$) の Binary n -Cube が用いられている。

並列オブジェクト指向モデル

データと手続きを統合したオブジェクトが、互いにメッセージ通信を行いながら並列に計算を進めるモデル。



小柳 滋 (正会員)

1949年生。1972年京都大学工学部数理工学科卒業。1977年同大学院数理工学専攻博士課程修了。工学博士。同年(株)東芝入社。現在、同社総合研究所情報システム研究所主任研究員。並列計算機アーキテクチャ、並列計算機の応用などに興味をもつ。電子情報通信学会、IEEE 各会員。



田邊 昇 (正会員)

1963年生。1985年横浜国立大学電気工学科卒業。1987年同大学院修士課程修了。この間ランダム疎行列処理の並列化アルゴリズムとその専用計算機の研究に従事。同年(株)東芝入社。以来高並列計算機の研究開発に従事。並列アーキテクチャに興味をもつ。電子情報通信学会会員。