

流れ処理概念に基づく 高度並列連想処理方式の一検討

許 炎, 柳 純一郎, 西川 博昭, 寺田 浩昭
大阪大学工学部

筆者らは、均質な多数のVLSIプロセッサによって構成される、極めて高性能な知的情報処理システムの実現法の確立をめざしている。

本稿は、この検討の一環として、人間の記憶モデルに基づく階層的なシステム構造上に、筆者らが提案している流れ形処理概念を導入すれば、高度並列連想処理システムが原理的に構成可能となることを示している。本稿ではまず、人間の知識処理過程に合致した階層的なシステム構造とその上での流れ形連想処理の考え方を簡単に述べる。続いて、流れ形連想処理方式の原理と、組織的な拡張性を持つ階層構造を採用したシステム構成法にふれる。最後に、関係モデルを対象とした、流れ形連想処理機構の一実現法として、相互結合網と、その上での流れ制御アルゴリズムを具体的に示している。

A Highly-Parallel Associative Processing Scheme Based on the "Flow-Thru Processing" Concept

Yan XU, Jun'ichirou YANAGI, Hiroaki NISHIKAWA and Hiroaki TERADA

Faculty of Engineering
Osaka University, Suita 565, Japan

This paper describes an architecture of a VLSI-oriented high-performance associative processing system based on the "flow-thru processing" concept which was proposed earlier by the authors.

A hierarchical system structure which is derived from a human storage model is first introduced. Subsequently, this paper reports a highly-parallel associative processing scheme based on the "flow-thru processing" concept. This paper also deals with a organization of hierarchical system which has extremely repeatable and expandable structure. Finally, an implementation of a highly-parallel associative processing mechanism and its execution control are briefly described with special emphasis on data-flow in the system structure proposed.

1. まえがき

計算機システムの高速度化は、従来から主としてVLSIに代表される、ハードウェア技術の進展によって、支えられてきた。最近では、従来形の逐次処理アーキテクチャを脱却し、高度な並列処理の実現によって飛躍的な性能の向上をはかる研究も活発に行われている。

一方、その性能向上に従って、応用分野が拡大されつつあることも疑いを容れない事実である。近頃では、人間の持つ高次な処理を対象とした、いわゆる知識処理システムが提唱され、基礎的研究からエキスパートシステムの構築まで、幅広い研究がなされると共に、従来形の逐次処理アーキテクチャの原理的な欠陥がしばしば指摘されるに到っている。例えば、i) 従来形システムで記号を処理する場合、ほとんどの処理時間はアドレス処理の実行に消費されること、或いは、ii) 記号処理では一般に、膨大なデータを取り扱う為、メモリとプロセッサが分離したノイマンボトルネックがシステムの性能アップの一つの大きな障害になること、などが挙げられている。

本研究は、これらの問題点を原理的に解決しうる高度並列連想処理方式の検討の為、人間の記憶モデル¹⁾に基づき、連想機構としての短期記憶(STS:Short-Term Store)及び分類構造化を持つ長期記憶(LTS:Long-Term Store)の機能を極めて自然に実現できるシステム構成法の確立を当面の目標に掲げている。

本稿では、多数のデータ駆動形VLSIプロセッサによって構成される、極めて組織的で且つ高性能な高度並列処理システム上での実現を前提にして、筆者らが提案している流れ形処理概念²⁾に基づく、高度並列連想処理の一実現法について述べる。まず、知識処理の構造に合致したシステムの階層構成と、その上での流れ形処理による連想処理の概念を述べる。続いて、組織的な拡張性を持つ階層構造を採用した高度並列連想処理システムの構成法を示す。最後に、関係モデルを対象とした流れ形連想処理の一実現法として、相互結合網と流れ制御のアルゴリズムを簡単に述べる。

2. 高度並列連想処理システムの構成原理

人間の知的情報処理の核機能は、極めて高度な並列性を持つ連想処理と考えられる。本章では、人間の連想処理を工学の立場から検討し、知識の処理構造、人間の記憶モデルの階層概念に基づく、システム構成の考え方と共に、システムの全体に一貫して採用した流れ形連想処理の概念について簡単に述べる。

2.1 記憶モデルに基づくシステムの構成法

既に筆者らは、情報処理システムの構築の際の、言語体系の重要性を指摘し、構造了解性に優れた図的な手法を導入した非手続き形超高位言語、ハードウェアから最低限独立したデータ駆動形図的中間言語³⁾、及び、中間言語のVLSI実現であるデータ駆動形プロセッサに対する実行形式、の3階層からなる、図的言語処理体系を提案している^{4) 5)}。

知識処理においてもまず、知識を表現し、モデル化できる言語体系が重要となる。本研究は、処理対象となる知識に対しても、要素間の関連、即ち、知識の構造の把握を最重要視し、構造了解性の高い超高位水準の図的言語として状態遷移概念⁶⁾或いはE-Rモデル⁷⁾に基づく記法を用いた、知識表現体系の検討を出発点としている。更に、このような図的表現による知識の処理には、その構造を保存しうる知識処理体系を確立する必要がある。これについては、図的表現を用いたデータ構造体を中間水準とする、統一的な処理系を導入し、その有効性を実験的に検討している⁶⁾。

本研究は、極めて高い性能を発揮する知識処理システムの検討の一環として、このような図的言語体系に基づき、生成された実行水準の記述を効果的に実行するシステムアーキテクチャとその実行制御方式を探索することを目的としている。

一方、人間の知識処理過程においては、ある刺激に対して、既存の知識構造を再編成し、あるいは、刺激を新しい知識として統合する、短期記憶における処理と共に、得られた知識

構造における情報間の関連性を整理し、保存する、長期記憶に対する処理が行われることが明らかにされている¹⁾。

この人間の記憶モデルに基づく、知識に対する立体的な処理構造を反映する為、知識処理を行うシステムの実行水準の処理機構に、この構造を保存した階層的構成を想定した。

従って、高度並列連想処理システムとして、i) 活性化された情報に対する記憶と連想処理の機能を兼ね備え、且つ階層的構造を持つ短期記憶(STS)、及び、ii) 知識の分類などの再構成機能を有する長期記憶(LTS)、からなる組織的な構成を採用した(図1)。

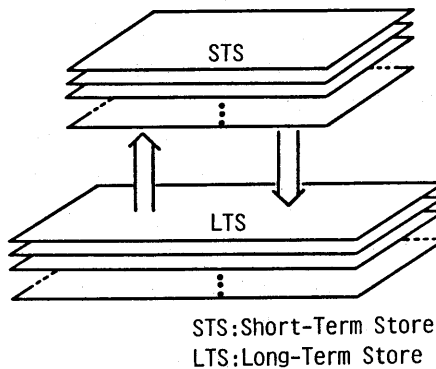


図1 階層的構造を持つ記憶モデル

2.2 流れ形連想処理概念の導入

従来形の情報処理システムは、記憶と処理を分離したシステム構成を採用した為、①大量のデータを、記憶から処理機構へ転送する際の転送路が隘路となり、更に、②記憶上のデータをアクセスする際に、記憶のアドレスの概念を強く意識することを必要とし、アドレスの計算に大量の時間を消費する。これらの問題が、システムの性能向上の障害になることがしばしば指摘されている。

本研究では、この問題を原理的に解決する為、データを取り扱う為の記憶と処理の機能を一体化した流れ形処理機構を想定した。更に、知識を表現するデータの処理は、基本的

には、特定のデータに対して、その内容に関連したデータ集合を求める、あるいは、これらのデータを統合する処理、即ち、連想処理にあると意識するに至った。従って、この連想処理を自然に行う処理として、処理機構上で、データ集合を転送すると同時に連想処理を行う、流れ形連想処理の概念を導入した。

又、この流れ形処理方式は、既に、現在想定している図的言語処理体系における中間言語D³LのVLSI実現法として、データ駆動形プロセッサQ-pに採用され、高度並列処理方式に関する有効性が実験的に確認されている²⁾。

2.3 階層的システム構成における層間処理

以上のような構成を採用した、連想処理システムでは、各階層内でデータの処理が行われるが、与えられた刺激に対する知識は長期記憶に蓄えられており、対象となるデータは短期記憶に転送され、処理される。従って、対象となるデータを抽出する層間処理が連想処理に極めて重要となることは明らかである。

しかしながら、長期記憶に蓄えられた大量のデータを、そのまま短期記憶に持ち込むことは得策とはいえない。更に、短期記憶で処理されたデータを分類し、長期記憶に統合して蓄えることは、効率の良い階層的システムの構築に対する必須の条件となる。また、短期記憶内の階層に対しても同様である。

従って、データ流の階層間移動の際にも、必要となるデータを制約し、あるいは、分類する処理を行う為、層間のデータ転送機構にも、流れ形連想処理機構を想定した。

3. 流れ形処理概念に基づく連想処理システムの構成法

本章では、データに対する統合的処理を可能とする流れ形連想処理を全面的に採用し、知識の階層構造に適合した、連想処理システムについて、まず流れ形連想処理方式の原理を示した後、短期記憶と長期記憶の階層的構成による知識処理システムの構成法を述べる。

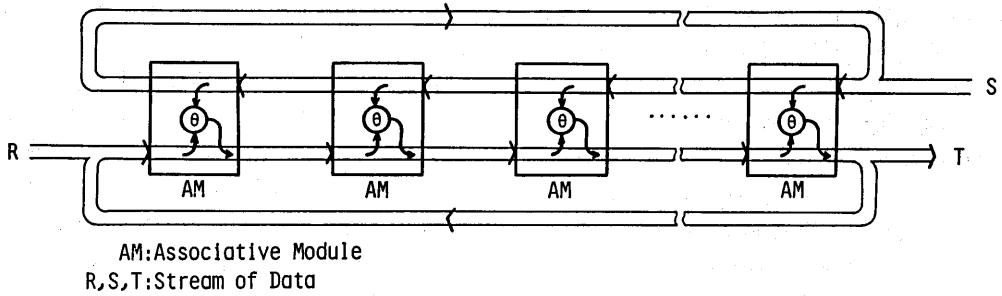


図2 対向流れ形並列連想処理方式の原理

3.1 流れ形処理概念に基づく連想処理方式

流れ形連想処理方式において、高度並列処理を実現するには、効果的なデータ対の生成法の検討が必須となる。これに対して、対向した流れを用いた流れ形処理概念が、非常に効果的な対検出機構を実現しうることを実験的に確認している²⁾。この処理機構は、二つの流れの相対的な速度により、与えられたデータ対に関する照合機会を実効的に2倍にすると共に、流れに沿った高度並列対検出処理を実現している。本稿に述べる連想処理方式は、この機構の原理を活用したものである。

図2は、有限個の単位連想機構によって構成する、原理的な対向流れ形連想処理方式を示している。ここで、四角は単位連想機構を表す。この方式に対し、①2つのデータの流れR, Sを、パイプラインの両端から流し、②単位連想機構上でデータ対が照合する時に、連想処理を行ない、更に、③結果として得ら

れたデータを、オペランドデータ流に合流し、データ流Tとして、処理が実行される。

本研究は、この対向流れ形連想処理方式を、次に述べる基本処理平面における処理機構として想定している。更に、1つのデータに対して多数のデータを連想するような処理における、データ対生成の重複を回避する為の手法を重点的に検討している。

一方、層間のデータの移動にその典型が見られる、データの一方が他方に比べて流量が少ない、特別な場合に対しては、処理の効率は一方向の流れ形程度であるが、データ対生成の重複の回避が原理的に保証された、以下のような流れ形処理方式の検討も有用になる。

即ち、図3のような構成の直交流れ形処理方式を検討した。この有限個の単位連想機構により構成したパイプラインに対して、流量の大きなデータ流Rは、垂直方向に分散して、並列に流れる。また、流量の少ないデータ流Sは、一つのループになり、先頭を識別子パ

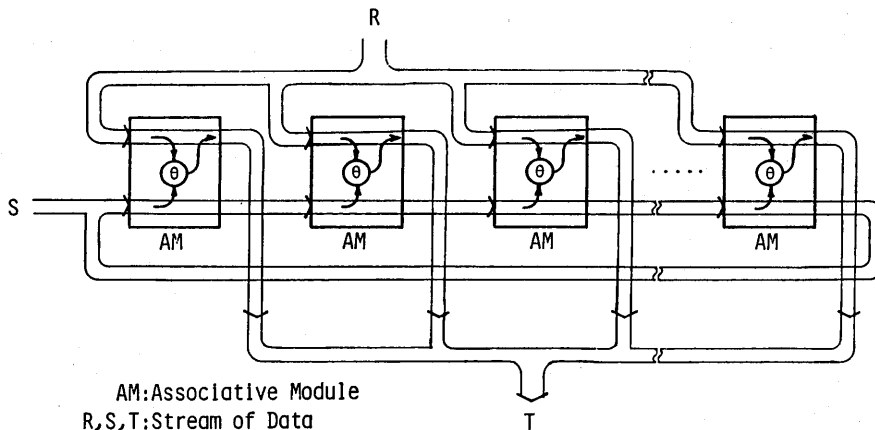
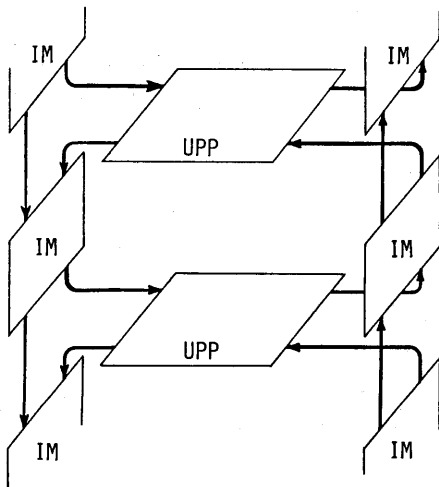


図3 直交流れ形並列連想処理方式の原理

ケットとした流れとして循環する。先頭識別子が到着した単位連想機構は、照合を行うデータ流Rのデータを要求し、連想処理を行う。データ流Rのデータを全部消費した所で、すべてのデータ対に対する照合が行われており、処理が終了する。

本方式では、データ流Sのデータは単一の循環した流れとなる為、データバケットの流量が制限される。更に、データ流Rの流量が、データ流Sのデータ数に依存する。その為、データ流SがRのデータ数に依存しない、知識に対する一般的な連想処理には、対向流れ形連想方式を用いる。しかしながら、他階層によって要求されたデータの制約を行う層間データ転送処理などには、適用可能であると考えている。

本方式は、単位連想機構をデータ流Sの方向へ一次的に結合した場合を考えると、単位連想機構の数が増加するにつれ、データ流Rの方向の流路の幅は増加するが、データ流Sの循環時間も長くなる為、結局、Rの流量は制限される。従って、Rの流量を保存する為、通路の幅と長さのバランスをとり、単位連想機構を相互結合網により結合した、パイプライン状の並列処理機構が有望となる。この相互結合網と、その結合網上で流れ制御方式については、次章で述べる。



UPP:Unit Processing Plane
IM:Interconnecting Module

図4 短期記憶の階層構成

3.2 短期記憶の階層的構成

短期記憶は、①構造的知識の表現と処理に合致した構造的なシステムの構成と、②心理学的検討より明らかにされたように、単一つの部位の機能ではなく、幾つかの部位による統合的な情報処理を行う構成、を想定した。更に、処理の対象によって、システムのリソースを柔軟かつ有効に活用することが要求される。その為、短期記憶自身にも、組織的な拡張を可能とする階層的な構成を想定した。

この階層的な短期記憶の構成は図4に示すように、単純かつ反復した構造を有する。ここで、データに対する連想処理を行う、対向流れ形処理概念に基づく並列処理機構を、基本処理平面 (Unit Processing Plane) と呼び、この基本処理平面と垂直に接続した並列処理機構を、層間接続処理機構 (Interconnecting Module) と呼ぶ。層間接続処理機構は、自分の階層の基本処理平面の入出力と接続し、処理要求によって、データを下位階層から選択的に転送し、或いは、基本処理平面で得られた結果を分類して、下位階層に転送する。

このような階層構成全体は、対向形の流れ処理機構とみなすことができ、対象となるデータ集合どうしを非常に効率良く与える短期記憶が構築可能になると考えている。

3.3 長期記憶と短期記憶との結合

高速に処理を実行できる短期記憶と大量のデータを保持する長期記憶の間の結合部に生じる隘路の問題は、大規模な知的情報処理システムを構築する際の、重要な問題となる。

本研究では、長期記憶と短期記憶との間のデータ転送のボトルネックを避ける為、図5に示すように、長期記憶を直交流れ形処理機構を通じて短期記憶の基本処理平面と結合する構成法を採用した。

更に、先程述べた、データの分類処理をも層間接続処理機構で行い、分類構造的性が高く、データのアクセス効率の良い、長期記憶を構築する。

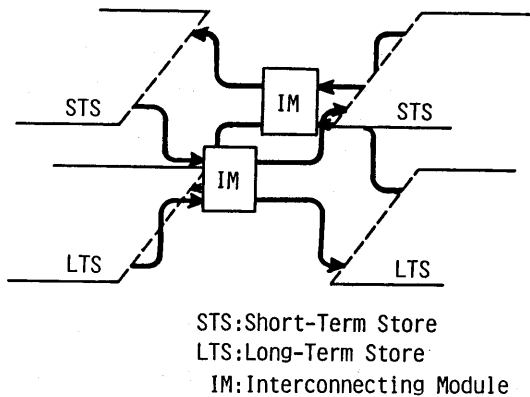


図5 長期記憶と短期記憶との結合

4. 流れ形連想処理機構の一実現法

本研究は既に、図的データ構造体の処理に対して、①並列処理に適したデータ独立性、②規則的なデータ構造、更に、③演算完備性が付与された、関係データモデル⁹⁾を図的データ構造体の実行水準の表現形式とする処理系の実現手法を提案している³⁾。

本章では、関係データモデルを内部表現形式とした知識処理システムにおける、層間処理の一実現法として、直交流れ形連想処理機構の構成法を示すと共に、この機構上の相互結合網により結合された単位処理機構に対する流れ制御アルゴリズムについて述べる。

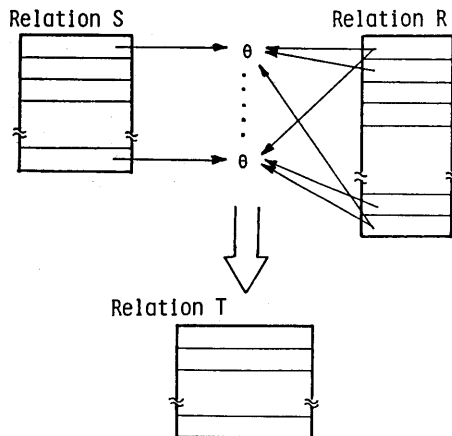


図6 結合演算におけるタプル対の連想処理

4.1 関係データモデルの流れ形連想処理

関係データモデルでは、情報は表の形式で表現される。この表を関係と言い、表の行と列はそれぞれタプル、属性と呼ばれる。更に、タプルの各属性に対応する値は属性値と呼ばれる。関係データモデルにおける処理の基本単位は、幾つかの属性値の組であるタプルの集合、即ち、関係である。

更に、関係に対する演算としては、単項関係演算の射影 (projection)、選択 (selection)、制約 (restriction)、及び、2項関係演算の結合 (join)、積集合 (intersection)、和集合 (union)、差集合 (difference) などがあるが、その全部は、関係における特定の属性に基づく処理である。

即ち、関係演算は、この属性の値を連想処理のキーワードとした連想処理となる。更に、射影演算におけるタプル重複の除去処理や結合演算などの2項関係演算は、すべてのタプル対を照合する為、処理負荷の高い連想処理となる。

ここで、結合演算 $T = R [X \theta Y] S$ を例とすると、その流れ形連想処理の手順は次のようになる (図6)。即ち、

- 1) 関係Rのすべてのタプルを、順序と無関係に1回かつ高々1回、関係Sのあらゆるタプルと照合して、
- 2) 照合した際、関係Rの属性Xの値と、関係Sの属性Yの値との間に、 θ 比較演算を行い、更に、
- 3) 比較演算子 θ を満足するタプル対を検出して結合し、関係Tのタプルを生成する。

この手順において、関係Rと関係Sのタプルの照合、即ち、連想の順序は、演算の結果と全く関係がなく、この性質を関係データモデルのタプル独立性と呼ぶ。

4.2 層間処理機構の一実現法

本研究では、知識の内部表現形式として、関係データモデルを採用している為、関係演算における処理の効率がシステムの性能向上

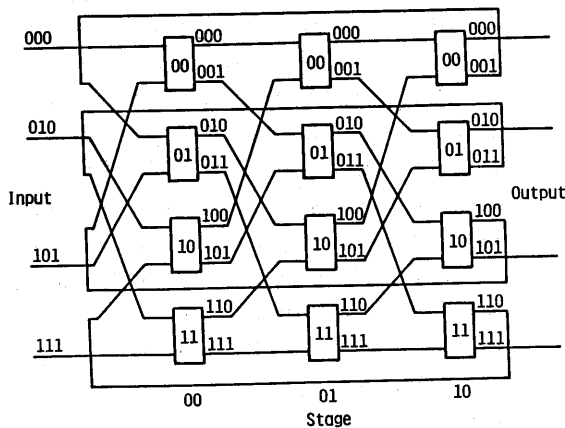


図7 8×8オメガネットに基づく相互結合網

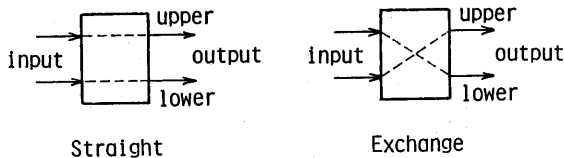


図8 単位処理機構のデータ転送

の重要な鍵となる。既に、大規模な関係データベースを効率的に処理できるシステムアーキテクチャが幾つか提案されている¹¹⁾¹²⁾が、研究の主流は、専用ハードウェアの構成にあった。従って、システムの拡張性に問題点が残ることがしばしば指摘されている。ここでは、汎用のデータ駆動形並列処理システム上に構築される、関係データベースに対する直交流れ形連想処理方式を用いた、層間接続処理機構の実現法を述べる。

本処理方式は、

- i) 流れ形処理概念に基づく為、並列処理機構はデータの流れを自然に反映するパイプライン構造を形成し、更に、データ流量の保存を可能とする充分な経路を提供し、
- ii) 専用ハードウェアの構成を避けて、並列処理のデータ供給に必須な処理である転送、分岐さらに合流の処理をも行なえる、少なくとも2入力2出力の構造を有する汎用の単位処理機構を持ち、

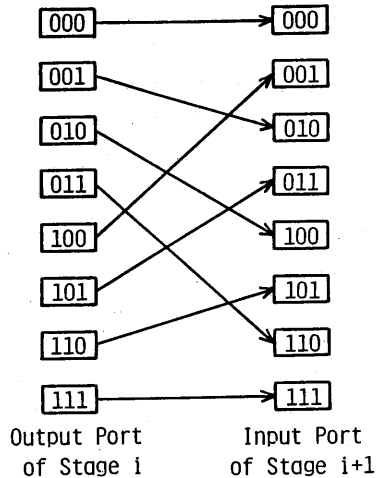


図9 Perfect Shuffle リンク

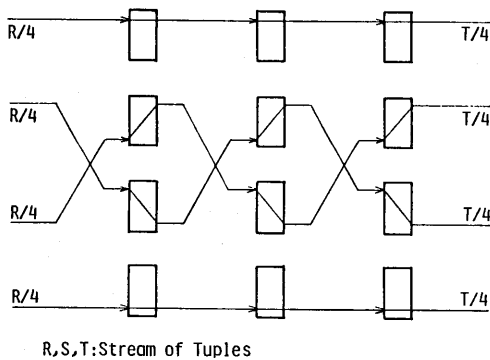
- iii) 自由度の高い分散制御方式の採用と、データ流の容易な制御を可能とする為、単独かつ重複した接続構造を持つ、並列処理システムとして構成される。

並列処理機構における相互結合網は数多く提案されているが¹²⁾。その中でオメガネット¹³⁾は多入力・多出力を持ち、且つ、全ての段に同じ接続構造を持つ為、このネットに基づいて結合した並列処理機構は、以上の要件を満足する。更に、この処理機構は、二次元的、且つ、各段に同様な接続構造を有する為、直交流れ形連想処理機構として採用すれば、前述したデータの流量を保存し、通路の幅と長さのバランスをとる問題は、流れ制御のアルゴリズムは不変のまま、単位処理機構の数と段数の調節によって、解決できる。しかし、単純なオメガネットは全体の単位処理機構を通るループ状通路が存在しない為、データ流Sの循環が実現できない。本研究では、このネットのトポロジに基づき、ループ状の通路となる入出力間の帰還路を与えた相互結合網上での並列処理機構の構築を試みた。図7は、8×8オメガネットに基づいて結合した並列処理機構を示している。単位処理機構は2入力と2出力を持ち、関係演算におけるタプル処理機能のみならず、ExchangeとStraightの転送機能(図8)を有する。また、

基本的な $N \times N$ オメガネットは N 入力と N 出力、および、 $k = \log_2 N$ 段より構成され、各段は $N/2$ 個の単位処理機構を持ち、 N 組の perfect shuffle リンク (図9) より連結される。更に、全ての単位処理機構に対する重複のないループを構成可能な帰還路を与える。この帰還路については、perfect shuffle リンクに補正を与えたものとなるが、 8×8 オメガネットに対しては、補正を必要としないことが、明らかになっている。

このような並列処理機構上で、関係 S のタプル数が関係 R のタプル数に比べて、非常に少ない特殊な場合であり、層間のデータ転送に対して、関係 S の要素により制約を加える処理となる。結合演算 $T = R (X \theta Y) S$ を考える。この場合、関係データモデルのタプル独立性を用いて、ネットワークの入力に、関係 R のタプル列と関係 S のタプル列を $N/2$ 本の流れに分けて与え、各流れを独自の通路を通じて流す。図10は、この流れの 8×8 オメガネットに基づく相互結合網上での実現例を示している。このような流れが以下の特徴を有することは、図10より明らかである。即ち、

- i) 流量の大きい関係 R のタプルの流れは、4 つの流れに任意に分割して入力され、各流れは3段を経て、最短経路で全部の単位処理機構へ分散できる (図10(a))。更に、関係演算によって生成した関係 T のタプルは最短経路で出力可能であり、システム内



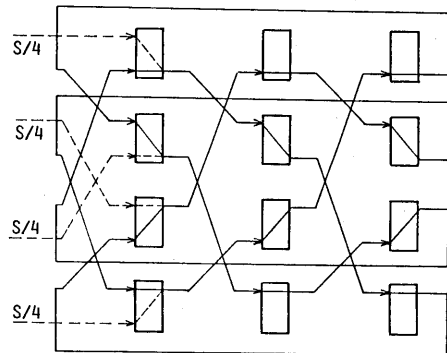
(a) 関係 R に対する通路

- のオーバーヘッドを減少させることが出来る。
- ii) 流量の少ない関係 S のタプルの流れが形成するループに対しても、入力のオーバーヘッドを減少させる為に、関係 S のタプル集合によって形成した流れを4つの入力から任意に分割して入力し、結合網へ流す。各流れは最短経路を経て、全ての単位処理機構を結ぶループに合流する (図10(b))。関係 S の各タプルは全部の単位処理機構を1回かつ高々1回経由して、関係 R のタプルと照合する際に θ 比較演算を行う。
- iii) 関係 R の流れと関係 S の流れは各単位処理機構で照合するが、リンク上では重ならない為、均質な流れが得られる。

本処理方式は、十分なオペランドデータ対の供給を可能とする為、処理時間は、照合処理の個数と単位処理機構の個数の比に依存する。この場合に、単位処理機構における演算処理の単位処理時間を1として、一段の単位処理機構の数を $N/2$ とし、段数を h とし、更に、関係 R と関係 S のタプル数を $t(R)$ 、 $t(S)$ とすると、演算処理は、ほぼ $O(t(R) * t(S) / (h * N/2))$ に従う処理時間で実現できる。

4.3 相互結合網上での流れの制御

N 入力 ($N = 2^{k+1}$, $k = 0, 1, 2, \dots$) のオメガネットは、各段に $N/2$ ($= 2^k$) 個の2入力2出力を持つ単位処理機構を用いて、



(b) 関係 S に対する通路

図10 8×8 オメガネットに基づく相互結合網上での流れ制御の一例

構成する為、今、 s_i ($i = 0, 1, 2, \dots, k$) を 2 進数の 1 桁とすると、一段の中の単位処理機構の番号は、 $s_k s_{k-1} \dots s_1$ で、単位処理機構の入出力端子は s_0 で、識別可能である。従って、一段の入出力端子の番号を 2 進数 $s_k s_{k-1} \dots s_1 s_0$ で表わすとした。

各段の入出力端子間は、perfect shuffle 関数、

$$F_{ps}(s_k s_{k-1} \dots s_1 s_0) = s_{k-1} \dots s_1 s_0 s_k$$

に従って接続する為、 i 段の単位処理機構と $i+1$ 段の単位処理機構との接続関係を考えると、次の関数

$$F(s_k s_{k-1} \dots s_1) = s_{k-1} \dots s_1 s_0 \quad (s_k, s_0 = 0, 1)$$

で、この関係が示される (図11)。ここで、 $s_k s_{k-1} \dots s_1$ ($s_k = 0, 1$) は i 段の二つの単位処理機構を表し、 $s_{k-1} \dots s_1 s_0$ ($s_0 = 0, 1$) は $i+1$ 段の二つの単位処理機構を表す。

この関数は次のように解釈できる。即ち、

- 1). i 段の単位処理機構 $s_k s_{k-1} \dots s_1$ からの流れに対し、 s_0 が 0 であれば upper output を、 s_0 が 1 であれば、lower output を、出力端子として選択し、出力する。この制御によって、データの流れは $i+1$ 段の単位処理機構 $s_{k-1} \dots s_1 0$ 、または、 $s_{k-1} \dots s_1 1$ に選択的に分岐する。
- 2). i 段の単位処理機構 $0 s_{k-1} \dots s_1$ と、 $1 s_{k-1} \dots s_1$ におけるデータの流れは、出力端子として、upper output ($s_0 = 0$) を選択することによって、 $i+1$ 段の単位処理機構 $s_{k-1} \dots s_1 0$ で合流する。

この解釈より、図11に示すような流れの通路を決定するアルゴリズムを導出する。図中の実線は関係 R の流れを表し、点線は関係 S の流れを表すとすると、以下の結果が得られる。

i 段の単位処理機構 $0 s_{k-1} \dots s_1$ に到達した関係 R の流れと、関係 S の流れは、1) によって、upper output ($s_0 = 0$) と lower output ($s_0 = 1$) から出力し、 $i+1$ 段の

異なる単位処理機構へ分岐して、新たなタプル対の生成を可能にする。

i 段の単位処理機構 $0 s_{k-1} \dots s_1$ に到達した関係 S の流れと、 $1 s_{k-1} \dots s_1$ に到達した関係 R の流れは、2) によって、双方ともに upper output ($s_0 = 0$) から出力して、 $i+1$ 段で合流して、照合を行う。

更に、この流れの制御と反対に、 i 段の単位処理機構 $0 s_{k-1} \dots s_1$ に到達した関係 R の流れを upper output ($s_0 = 0$) から出力し、 $1 s_{k-1} \dots s_1$ に到達した関係 R の流れを lower output ($s_0 = 1$) から出力すると、 $i+1$ 段で異なる単位処理機構に到達でき、関係 R の流れは並行して流れる。

従って、転送路上のすべての単位処理機構 $s_k s_{k-1} \dots s_1$ において、関係 R と関係 S の流れがすべてのリンク上で衝突することなく、且つ、全てのタプル対の照合を実現する為、例えば、関係 R の流れに対して、

$$s_0 = s_k$$

で、更に、関係 S の流れに対して、

$$s_0 = \neg s_k \quad (\neg: \text{NOT})$$

で、出力を制御する。

しかしながら、最後尾の段における帰還路の選択については、帰還路に対する perfect shuffle リンクの補正が関係するため、その定式化を現在検討している。

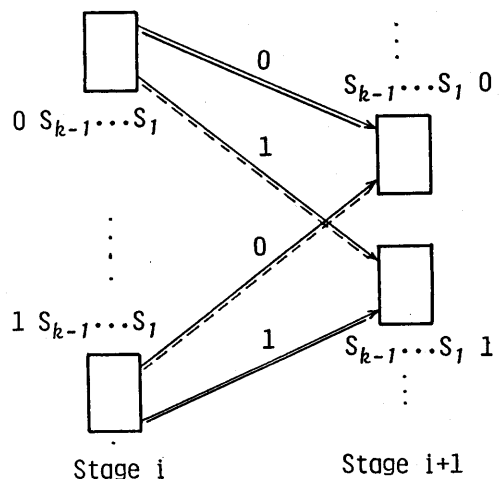


図11 衝突が起こらない分散制御

5. むすび

本稿では、知識処理の構造に合致したシステム構成の一実現法と、知識表現の内部表現形式となる関係データ構造の流れ形高度並列連想処理の一方式を提案した。

本方式は、流れ形処理の概念に基づき、システム上に隘路のない流れ形処理を導入し、高度な並列連想処理の実現をめざしている。

更に、本稿で提案したシステム構成法は、均一の処理機構による反復性の高い構成を採用した為、システムの拡張が容易に行なえる。又、長期記憶は、流れ形連想処理機構を介して短期記憶に結合される為、この間のデータ転送のボトルネックが原理的に回避される。従って、大規模な高度並列連想処理システムも原理的に構成可能な方式となっている。

今後は、i) 対向形流れ処理方式に基づく、高度並列連想処理機構の検討、ii) 関係データベースの処理の効率を向上させる為、演算結果に寄与する可能性のないタプル対の照合処理を回避できる、オペランド関係のタプルに対する前処理を含む実行方式、に検討を加える予定である。現在、本方式に基づくプロトタイプシステムの実験的検討を進めている。これについては、本稿の始めにふれた図的言語処理体系を含めて稿を改めて、論じたいと考えている。

謝辞 本研究に関し討論および種々の援助を頂いた大阪大学工学部寺田研究室の各位に感謝する。なお、本研究の一部は、文部省科学研究費（一般B,61460137, 試験2,61850063, 奨励A,61750334）の援助を受けて行ったものである。

参考文献

- (1) Atkinson, R.C. and Shiffrin, R.M.: "Human Memory: A Proposed System and its Control Processes", Reprinted in Bower, G. (ed.): "Human Memory: Basic Processes", Academic Press, pp.7-113 (1977).
- (2) 寺田, 中村, 八木, 森, 白井: "データ駆動形プロセッサ Q-x の設計思想", 第32回情処全大, 5R-1他 8件 (5R-2 ~9) (1986-3).
- (3) 西川, 寺田, 浅田: "履歴依存性を許すデータ駆動図式", 信学論(D), J66-D, 10, pp.1169-1176 (1983-10).
- (4) 寺田, 西川, 浅田: "D²L による図的言語処理体系とその一実現法について", 信学会データフローワークショップ予稿集, pp.119-126 (1986-5).
- (5) 西川, 浅田, 寺田: "超高位図的言語処理システムの設計思想", 第33回情処全大, 4F-9他 4件 (4F-10 ~13) (1987-3).
- (6) 金倉, 姜, 西川, 浅田, 寺田: "超高位図的言語処理システムにおける知的インタフェース機能の一実現法", 第34回情処全大, 4V-1他 2件 (4V-2,3) (1987-3).
- (7) Chen, P.P.-S.: "The Entity-Relation Model - Toward a Unified View of Data", ACM Trans. on Database System, Vol.1, No.1, pp.9-36 (Mar. 1976)
- (8) Codd, E.F.: "A Relational Model of Data for Large Shared Data Banks", Comm. ACM., Vol.18, No.6, pp.377-387 (Jun. 1970).
- (9) Schuster, S.A., Nguyen, H.B. and Ozkarahan, E.A.: "RAP.2--An Associative Processor for Databases and Its Applications", IEEE Trans. Comput., Vol. C-28, No.6, pp.446-458 (1979).
- (10) Baum, R.I. and Hsiao, D.K.: "Database Computers-A Step Towards Data Utilities", IEEE Trans. Comput., Vol.C-25, No.12, pp.1254-1259 (1976).
- (11) Berra, P.B. and Oliver, E.: "The Role of Associative Array Processors in Database Machine Architecture", IEEE Comput. Vol.12, No.3, pp.53-60 (1979).
- (12) Feng, T.: "A Survey of Interconnection Networks", IEEE Comput., Vol.14, No.12, pp.12-27 (Dec. 1981).
- (13) 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).