

## 意味ネットワークマシン (IXM) プロトタイプの開発

樋口哲也、古谷立美、半田剣一、楠本博之、国分明男  
(電総研)

本報告では、意味ネットワークに基づく知識情報処理システムIXの概要と、その中核をなす意味ネットワーク処理用高並列計算機IXM(イクスム)のプロトタイプについて述べている。

IXMのプロトタイプは、32台のCPU、大容量連想メモリ、プログラム可能な相互接続網を主な構成要素とする。プロトタイプとはいっても、CPUと連想メモリの並列性を合わせると100000以上の並列性を持つ高並列計算機であり、大規模意味ネットワークを効率的に処理できる。

IXMは、汎用的な意味ネットワーク言語IXLで書いたプログラムを実行するが、IXLはPrologの組み込み述語としてPrologプログラム内から呼び出すことができる。従って、見かけ上、Prolog処理系に意味ネットワークの全探索機能を持つバックエンドプロセッサが付加された形となる。

## The Development of the Prototype of a Semantic Network Machine IXM

Tetsuya Higuchi, Tatsumi Furuya, Hiroyuki Kusumoto,  
Ken-ichi Handa, Akio Kokubu  
Electrotechnical Laboratory  
1-1-4 Umezono, Tsukuba, Ibaraki 305, Japan

This report describes an overview of a semantic network knowledge information system IX [iks], and a prototype of semantic network machine, IXM. The IX system comprises knowledge representation language IXL and a semantic network machine IXM. Once the application program such as knowledge base and natural language processing is written in IXL, IXM can execute the program with high parallelism. The prototype of IXM machine consists of 32 transputers and large associative memories which involve massive parallelism over 100000. In order to evaluate various connection schemes, the prototype has reconfigurable structure. IXL procedures can be called within usual Prolog programs as they are defined as built-in predicates.

## 1. はじめに

近年、人工知能（AI）が実用化の領域に入り、種々のシステムが商品化されるようになってきた。しかし、AIには扱うデータ数の増大に伴い計算量が爆発するアルゴリズムが多々ある。これらのアルゴリズムはたとえ有用であることがわかっていても、現状では計算パワーの制約からまったく手つかずといっても過言ではない。確かにAIワークステーション等に進歩は見られるものの、科学技術計算におけるスーパーコンピュータのような存在が、まだAI分野に確立されてはいない。今後、AI関連技術の進展によって、莫大な計算量を要する領域の比重が増すことは明らかであり、その際、超並列計算機の導入が必須のものであると思われる。

このような状況を背景として電総研では意味記憶システムIX（イックス）の研究開発を行っている。IXでは、述語論理やフレームと共に代表的な知識表現技法として知られる意味ネットワーク（Semantic Network）に注目し、(1)高度な知識を効率良く表現する方法を確立し、(2)最近のVLSI技術に基づく知識情報処理向き並列処理マシンの構築を目指しており、(3)自然言語処理、知識ベース、学習、記憶モデル等、意味ネットワークの広汎な応用を対象としている。

意味記憶システムIXは、具体的には、(1)意味ネットワーク処理向き並列マシンIXM（イクスム）と、(2)意味ネットワークに基づく知識表現言語IXL（イクスル）を核として成り立っており、意味ネットワーク処理を統一的にサポートすることを目的としている。すなわち、種々の応用を意味ネットワークでモデル化してこれをIXLで記述し、それをIXMが高並列で実行する統一的な環境の実現を図っている。

意味ネットワークはこれまでに広く用いられているものの、大規模なシステムがほとんど作られていないのは、意味ネットワークが大きくなって来ると、探索すべき空間が爆発的に増大するためである。この問題の緩和のためには高並列化が必須である。

意味ネットワーク処理向き並列マシンIXMは、大容量連想メモリを備えたマルチプロセッサシステムであり、現在構築中のプロトタイプでも、PEと連想メモリの並列性をあわせると100000以上の並列性を有する高並列計算機である。IXMは、意味ネットワーク言語IXLで書かれたプログラムを高並列処理する。

IXLは、知識ベースや辞書など、ネットワーク型の知識の記述を柔軟に行える汎用的な言語である。I

XLはPrologプログラムの中から呼ぶことができ、これをIXMが並列処理する。つまり、見かけ上、Prolog処理系に意味ネットワークの全探索機能を持つバックエンドプロセッサが付加された形となる。

本稿では現在開発中の意味記憶システムIX（イックス）の概要と意味ネットワークマシンIXMのプロトタイプについて述べる。

## 2. 意味ネットワーク

意味ネットワークは、ノードとリンクから成る図式的な知識表現形式であり、ノードは概念、リンクは概念間に成り立つ関係を表わす。たとえば図1は動物についての知識を意味ネットワークを用いて階層的に表現したものである。図からわかるように意味ネットワークでは人間の直観にあった知識表現が可能であり、述語論理など他の知識表現形式に比べて、(1)階層的な知識表現をわかりやすく行える。(2)上位概念の持つ属性を下位概念に継承できる、などの利点を持つ。上位概念の継承をインヘリタンスと呼ぶ。たとえば図1において動物のノードには属性として、「えさを食べる」とあり、いま仮りに「鳥はえさを食べるか」という質問がきたとする。鳥のノードには何も書かれていないが、鳥の上位概念である動物のノードには「えさを食べる」とあるため、この属性を継承して、答えはイエスとなる。

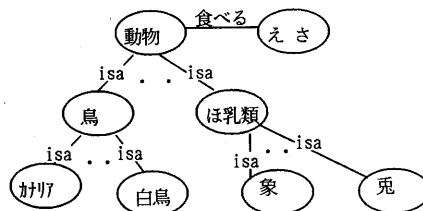


図1 意味ネットワークによる階層的知識の例

## 3. 知識表現言語IXLと意味ネットワークエディタIXG

IXLは、意味ネットワークに基づいて知識ベースの記述、検索、修正を行うことができる<sup>1)</sup>。例えば、「カナリアは鳥である。」という知識を言うためにIXLでは「～である」を表わすisaというリンクを使って、link(isa, canary, bird)というコマンドでこの知識を記述する。IXLでは、新たな知識を知識ベ

To construct a relation:  
 assertion(R, X, Y).  
 property(R, X, Y).

To inquire about a link:  
 isa(X, Y).  
 instance(X, Y).  
 ako(X, Y).  
 source(R, X).  
 destination(R, Y).

To inquire about a relation:  
 asst(R, X, Y).  
 prop(R, X, Y). 図2 IXLコマンド

To connect nodes by a link:

```
link(is_a, X, Y).
link(not_isa, X, Y).
link(instance_of, X, Y).
link(not_instanceof, X, Y).
link(a_kind_of, [X, Y, ...], Z).
link(source, R, X).
link(destination, R, Y).
link(rule, X, ((
  asst(R, X, Y):-....
  prop(R, X, Y):-....
  isa(X, Y):-....
  instance(X, Y):-....))).
```

図2 IXLコマンド

ースに追加する際、それまでに知識ベースにある知識と矛盾しないかどうかのチェックを行う。この知識がシステムに矛盾なく格納されたとすると、このあと「カナリアは何の一種か?」を知るために検索コマンドとしてisa(canary, X)を与えれば、図1では解としてbirdとanimalが得られる。

IXLのコマンドはPrologに似ているが、その実行は逐次的に解を返すのではなく、IXMマシンによる全解探索が基本である。IXLコマンドを図2に示す。すでに述べたように、IXLコマンドは、Prologプログラムの中に組み込んで実行することができる。たとえば、上のカナリアの上位概念を求める場合なら、

```
?-isa(canary, X), write(X), fail.
```

と入力することによって、すべての解を得ることができる。このようにIXLコマンドは、Prolog(現在はQuintus Prolog)の組み込み述語として組み込まれており、通常のPrologプログラムと、なら変わらない形で利用できる。IXLコマンドの部分だけが、IXMマシンの方に渡り、全解探索の結果として複数の解がProlog側に戻される。そのIXLコマンドにバックトラックするたびに、解が一つずつバインドされる。

IXLは、従来の意味ネットワーク言語の弱点を補強し、次の3つの特徴をもっている。

- 1) 概念間の関係の意味をより正確に、かつ柔軟に記述する。
- 2) 手続き的知識をロジックの節形式で記述する。
- 3) 否定的な知識を陽に記述できる。

手続き的知識の記述が必要になるのは、意味ネットワークの表わす知識が各種の事実や階層関係など宣言的なものが主であり、これだけでは複雑な推論規則を扱えないからである。IXLではロジックの節形式で手続き的知識を表わしている。たとえば図3のよう

に旅行先とそこへの交通費、および滞在費についての意味ネットワークがあるとする。旅行費用が、交通費と滞在費の合計であると考

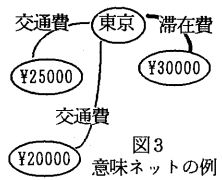


図3 意味ネットの例

```
えるならば、旅行費用を求めるための手続き的知識は次のようになる。
asst(旅行費用, X, Z):- asst(交通費, X, Y),
  asst(滞在費, X, W), Z is Y + W.
```

IXLのインタプリタは今回のプロトタイプとは別に、すでにPrologで書かれて、DEC2060、およびVAXで稼働しており、知識獲得システム、ワインの知識ベースなど、種々の応用システムが開発されている。

意味ネットワークの応用プログラムを開発する上で重要なのは、その開発環境である。すなわち意味ネットワークは、ノードとリンクから成る図的な表現であるため、意味ネットワークを文章型言語で書くのではなく、図的表現でそのまま書けた方がデバッグや理解の上で好ましい場合がある。このためIXではグラフィックディスプレイ上で意味ネットワークを図的に記述し、さらにそれらの修正や検査の行える意味ネットワークエディタIXGも開発した。実際的な応用に供される意味ネットワークは整った構造をとることはまずないので、IXGのようなグラフィックな支援は、特に大規模ネットにおいて不可欠である。

#### 4. 意味ネットワークマシンIXM

意味ネットワークマシンIXMの並列性の高さは大容量連想メモリによるものである。連想メモリというと、アドレスを用いない、メモリ内容による検索だけが想起されがちであり、その機能だけならスーパーコンピュータでハッシュした方が速いかもしれない。しかし、内容検索は連想メモリの一機能にすぎない。たとえば、検索して選ばれた複数の語に対して並列に書き込みを行うといった機能もあり、連想メモリを一種のSIMD計算機として利用することが十分に可能である。連想メモリはこれまで小規模なものしかなかったので、計算機アーキテクチャの分野でごく限られた状況でしか用いらなかったが、最近ではNTTの20Kbチップのような大規模なものも発表されており<sup>2)</sup>、超並列処理を実現するための有効なデバイスと成りうる。

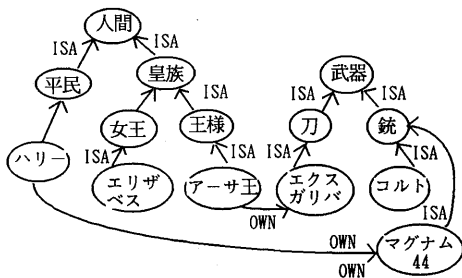


図4 意味ネットワークの例

(1) 意味ネットワーク処理

意味ネットワークに特徴的な処理は[1]連想処理、[2]マーカ伝搬、[3]集合演算である。これらは意味ネットワークの各ノードに、マーカと呼ぶ1ビットのフラグを複数個割り当てることで効率よく処理できる。たとえば図4の意味ネットワークに対し「武器を持つ皇族は誰か」という質問がきた場合を考える。まず、[1]皇族のノードを見つける(連想処理)。次に、[2]皇族より下位の概念(例えば女王や王様)を、皇族から出発して次々にたどり、それぞれのノード内のマーカ1番をセットする(マーカ伝搬)。  
 [3]武器のノードおよび、その下位概念を連想処理とマーカ伝搬により見つけ、各々のマーカ2番をセットする。  
 [4]OWNリンクの両端にマーカの1、2番がセットされているのが解となるので、すべてのOWNリンクに対して、両端のマーカ1、2番のAND演算を行わせる(集合演算)。この結果エクスガリバを持つアーサー王が解となる。

これら3つの基本的処理を効率よく実行するのが意味ネットワークマシンの要件であり、これらの実現をIXMマシンでは連想メモリの効果的利用により図っている。

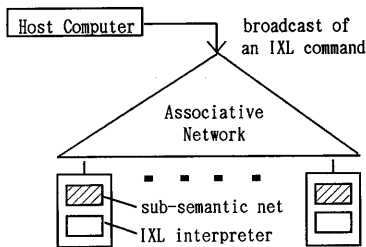


図5 IXMの全体構成

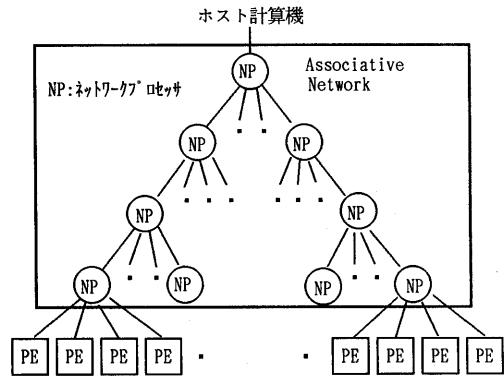


図6 木構造の場合のIXMの構成

(2) IXMマシンの全体構成

IXMマシンは、ホスト計算機の制御下でMIMD的に動作するマルチプロセッサであり、図5に示すように連想メモリを備えたPEと、これらを接続し、かつ意味ネットワーク処理の一部も担う連想ネットワークから構成される。

各PEには二種類の連想メモリがあり、全体を分割した“部分”意味ネットワークと、知識表現言語IXL(イクスル)のインタプリタをそれぞれ置く。PEの実行はプログラムカウンタに基づく制御でなく、意味ネットワーク内を流れるマーカによって起動される非同期型制御に基づいており、並列性を生かす目的を持つ。連想ネットワークは図6に示すような木構造のネットを提案しており、各接点にネットワークプロセッサ(NP)を置く。ネットワークプロセッサも連想メモリを備えており、プロセッサ間交信を行うほかに、連想メモリを用いて後述するマーカ伝搬の並列化処理を行う。

IXMマシンの実行は、一つのIXLコマンドを単位としている。たとえば、カナリアは何の下位概念かを調べるのに、isa(canary, X)というIXLコマンドを発するが、この処理が終わるまでIXMマシンは他のコマンドを受け付けない。但しその一つのコマンドの実行中にIXMマシン内の各PEはMIMDで動作する。

IXLコマンドはホスト計算機上で実行されるPrologプログラムの中から発せられ、IXMマシンにブロードキャストされる。

## 5. IXMの特徴

IXMのアーキテクチャの主な特徴を以下に列挙する<sup>3)</sup>。

### (1) 連想プロセッサ

連想処理は、意味ネットワーク処理の最も基本的な演算の1つである。IXMでは、各PEに連想メモリを設け、そこに大規模な意味ネットワークを分割したサブネットワーク、つまり複数のノードを格納し、連想処理、そして集合演算を一斉に施す。つまり連想メモリのワード数分の並列性が1台のPEの中に存在することとなる。

図7の意味ネットワークがあり、二つの集合と、その積集合の様子を表現しているとする。そしてその一部を連想メモリに格納した様子を図8(b)に示す。連想メモリへの格納はリンクを単位としている。図8(a)に示すように、連想メモリの一語は、そのリンクによって接続されるノードの属性、そのリンクのもう一つの接続先、そのリンク名、及び意味ネットワーク処理の演算結果を保持するマーカビット領域から成る。たとえば集合AのメンバーであるA1のノードは、Aのノードをisaリンクで指しているので図8(b)のようになる。

マーカビット領域が連想メモリ上で実現されていることにより、集合演算や連想処理がIXMでは高並列で実行できる。

たとえば図8の意味ネットワークにおいて二つの集

合の積集合をもとめる場合を挙げる。まず、集合Aの各メンバーにマーカビット1番をセットするために、連想メモリの検索データレジスタを図8(c)のようにセットして検索を行う。これにより、A1からABZまでのノードが同時に選ばれる。これは、ノードAからのマーカ伝搬処理が、リンク数に拘わらず、わずかに連想メモリサイクルで行われることを意味する(注: この例ではそうだが、すべてのマーカ伝搬が1サイクルで出来るわけではない)。続いて、選ばれたこれらのノードに対してマーカビット1番の箇所と同時に1を書き込みに行く。ここまでの処理は、メンバーの数がいくつあろうと、マスク設定を含めて、つねに連想メモリの4サイクルで済む。

集合Aのメンバーにマーカビット1番、集合Bのメンバーにマーカビット2番がセットされたとすると、両者の共通要素を求めるには、連想メモリの検索データレジスタを図8(d)のようにセットして検索を行えばよく、一斉に共通要素のワードにヒットフラグが立つ。つまり積集合の演算を、共通要素がいくつあっても、ただ一回の連想メモリアクセスで実現することが可能である。言い替えば、大規模意味ネットワークでもそれが連想メモリのワード数内に収まるなら、連想・集合演算(および、特定のマーカ伝搬)は常に一定時間内で実行できることになり、意味ネットワークが大きくなったときの計算量の爆発に対して有効な手段となる。

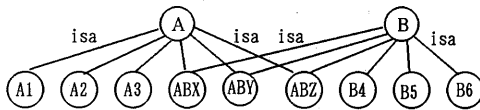


図7 二つの集合を表わす意味ネットワーク

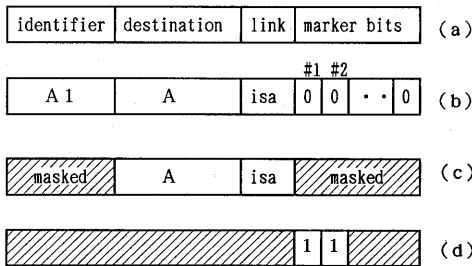


図8 連想メモリ上での意味ネットワークの表現

### (2) 連想ネットワーク

連想ネットワークには大別して次の4つの機能がある: (1) IXL命令のブロードキャスト、(2) PE間のマーカ伝搬、(3) 結果の回収、(4) equivalenceノード。

連想ネットワークの形状は、ピラミッド型の木構造を提案しているが、これはブロードキャストや結果の回収の上で都合がよく、またマーカ伝搬に要するステップ数も削減できる。ただし、ハイパーキューブ構成も同様な利点が予想される。

これら4つの機能のうちもっとも特徴的なのはequivalenceノードであり、これはマーカ伝搬における並列性を生かす目的を持つ。

マーカ伝搬は意味ネットワーク処理に特有なものであり、並列性が多く内在している。例えばNノードから成る2進木の意味ネットワークがあり、全ノードをマークするためにルートノードからマーカを流すと、logNステップでマークを終了することができる。も

し100万のノードなら20ステップである。但し、これは意味ネットワークの構造と全く同じにPEが接続された場合であって、連想メモリにサブネットワークを格納するIXMの場合にはマーカ伝搬の並列性を生かすメカニズムが必要である。先に集合のメンバーを求めするために、isaリンクを一段だけマーカ伝搬でたどり、わずか1サイクルで済む例を示したが、このやり方は、isa階層が多段になると効率が悪くなる欠点がある。図7の例を挙げるとノードAからマーカ伝搬を行う際に、ノードAからisaリンクを一つづつたどって、A1ノードからABZノードまでを順番に求めて行くやり方のほうが多段isa階層の場合には効率がよい。しかし、これだと、ノードAがマーカ伝搬を接続リンクの数だけ繰り返すので、もしリンク数の多い場合には処理のボトルネックとなる。意味ネットワークは、階層的な知識を扱うことが多いため、上位概念か

ら、その下の下位概念に多数のリンクがのびることは常におこる。このため、equivalenceノードと呼ぶ方式をIXMに導入した。図9を用いてequivalenceノードの概念を説明する。

いまuniversityのノードからマーカを流し、大学に属する全学生をマーカ伝搬によってマークするとする。もしstudentのノードが1台のPEに割り付けられているとすると、そのPEは全学生に対して逐次的にマーカを流していかなくてはならず、あい路となる。これはPEが1台しかないのだから当然である。しかし、もしstudentのノードを複数化して図9(b)のように分散すれば、マーカの伝搬を並列に行わせることができる。このstudentノードのようなノード、つまり論理的属性は唯一つであるが物理的にはPE間に分散されているノードをIXMではequivalenceノードと呼ぶ。

但し、あるequivalenceノードにマーカが送られてきたとき、そのマーカは複製されて、そのequivalenceノードを含む他のPEに対しても転送されなければならない。たとえばMr.Aによって、それが物理的

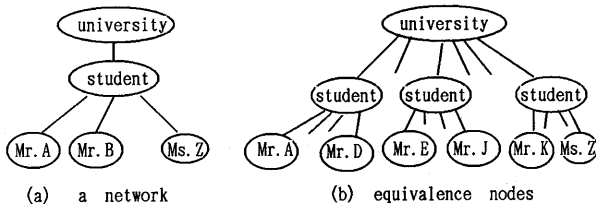


図9 equivalenceノード

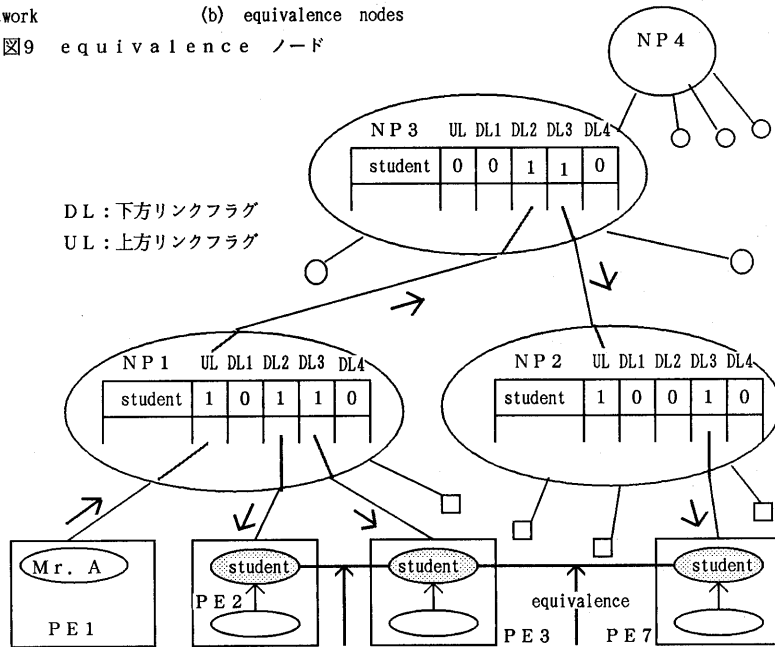


図10 equivalenceノードによるマーカ伝搬の並列化

につながっているstudentノードのマーカビットがセットされたならば、他のstudentノードのマーカビットもセットされなければならない。このため、連想ネットワークを形成する各ネットワークプロセッサの中にはequivalenceノードを登録している表があり、さらにその表は連想メモリに格納されてマーカ伝搬の効率化が図られている。一例として図10に上述のMr. Aからのマーカ伝搬の並列化の様子を示す。これは、PE1にあるMr. AのノードからPE2にあるstudentノードへマーカを送ったときに、マーカが複製されてPE3、およびPE7にもあるstudentノードへもマーカが送られる様子を示している。

## 6. プロトタイプシステムの概要

以上に述べたアーキテクチャの検証を行うために、現在32台のCPUと連想メモリから成る意味ネットワークマシンIXMのプロトタイプを製作している。

第一の目的は、連想ネットワークの接続形態についての実験で、これまでに提案した木構造とハイパーキューブ等との比較検討を行う。従って接続形態の変更を柔軟に行える必要があるため、トランスピュータ間の接続を、プログラムで接続を制御できるスイッチ基板を介して行う。具体的にはトランスピュータ用のリンクスイッチLSIを利用する。これに伴いPEの台数とNPの台数が接続形態に応じて変わるので、PEにもNPにも共通に使えるPE基板を一種類作成する。

試作の第二の目的は、これまでに開発した意味ネットワークの応用プログラムを高速に実行できるようにし、より大規模(数万リンク以上)の意味ネットワークを扱える環境を整備することである。IXMマシンのポイントである全探索機能を実現するためには、連想メモリの導入が必須であるため、各PEに連想メモリを実装する。この結果、プロトタイプの試作といっても連想メモリをすべて実装すると、ビット数の上では連想プロセッサSTARANの2倍となる。

プロトタイプは図11に示すように、1)ホスト計算機のIBM PC、2)任意の接続形態を実現するための5枚のスイッチ基板、3)PEまたはNPの機能を実現する32枚のPE基板の三つの部分から成っている。但しIBM PCは、ハードウェアのデバッグ、およびシステムプログラム開発用が主である。IXMの一般ユーザはSUN3上のPrologプログラムからIXMを利用する。

### ・ホスト計算機

IBM PCにはトランスピュータ(2MBRAM)を増設してある。ここでoccamプログラムの開発を行うほか、IXMマシンの初期化と実行時の制御を行う。

初期化では、まずスイッチ基板内のリンクスイッチLSIへ接続情報を送ることでIXMマシンの接続形態を決定し、続いてシリアルリンクを通じて各PE基板にプログラムをロードする。

実行時には、IXLコマンド命令をブロードキャストして各PEを一斉に起動すると共に、実行結果の回収、全PEの終了検知等を行う。全PEの終了はハードウェアで検知し、ホスト計算機につながるルートのトランスピュータに割り込みをかける。

一般ユーザ用のSUN3にはT414を含むボードを増設し、そこからIXMに接続する。

### ・スイッチ基板

32台のトランスピュータの持つシリアルリンクの総数は128本であるが、後述のようにさらに各PE基板について4本のシリアルリンクを増設している。これらを5枚のスイッチ基板に分けて接続し、32PE基板間の任意の接続形態を実現できるようにしている。スイッチ基板内のリンクスイッチLSIは、入力、出力を各32本持っており、入出力間の対応関係、つまりPE基板間の接続形態は、ホスト計算機内のトランスピュータのシリアルリンクから送られる情報に基づいて決定する。この可変な接続機構により、提案している4進木構成のほかに、2進木や3進木、あるいはハイパーキューブ構成も実現することができる。

### ・PE基板

PE基板の写真を図12に示す。PE基板は、図13に示すようにT800トランスピュータ、連想メモリ、SRAM、リンクアダプタ、および周辺回路から成る。

トランスピュータはプロセッササイクルが17.5MHzである。内部にはサイクルタイム59nsの4KByteオンチップRAMを備えており、ここにIXLインタプリタ等のプログラムを置く。外部メモリは32K語 X 32ビット

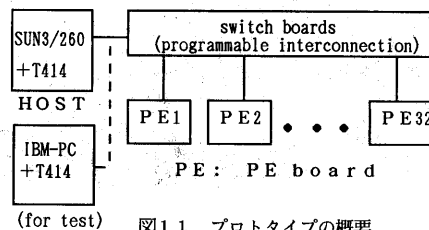


図11 プロトタイプの概要

のSRAMで、サイクルタイムは230nsである。トランスピュータの4本のシリアルリンクは、20Mbit/sに設定する予定であり、接続する。

しかし、4本のシリアルリンクだけでは4進木やハイパーキューブ接続ができないため、リンクアダプタと呼ぶ直並列変換用のチップを用いて、シリアルリンク数を8つに増やしている。すなわち、トランスピュータのバスに4つのリンクアダプタを接続し、外部のトランスピュータのシリアルリンクから送られるデータをバイトに変換してバスにのせる。あるいは逆にバイトデータをシリアルに送り出す。

連想メモリはSRAMと共にoccamのアドレス空間に割り付けられており、トランスピュータからはRAMと同じようにアクセスできる。サイクルタイムもSRAMと同じ230nsである。但し連想メモリは、トランスピュータと独立のクロックで動作し、トランスピュータからのアクセスが非同期であるため、平均的なアクセス時間は約450nsとなる。

連想メモリには意味ネットワークを展開して格納し、更なる上で集合演算や連想処理といった意味ネットワークの基本処理を実現する。連想メモリは1ワードが40ビットであり、このうち8ビットが1ビット単位でマスク可能であるため、マカビットとして用いることができる。残りの32ビットはバイト単位でマスクできる。

連想メモリの動作モード(命令)の設定は、バス上に出力されるアドレスを利用する。よく用いる動作モードには、検索を行ってヒットした複数の語に対する並列書き込みや選択分離等がある。

連想メモリはPE基板当たり4k語を実装する予定

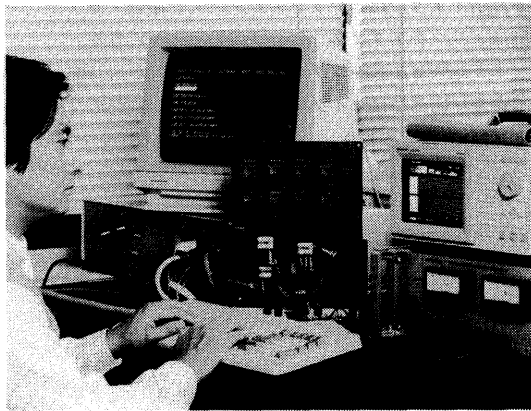


図12 PE基板(写真中央)

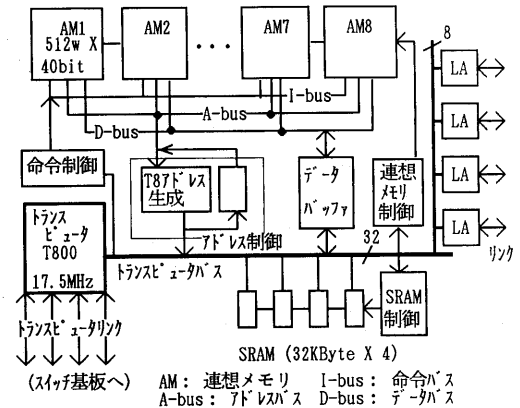


図13 PE基板のブロックダイアグラム

であり、IXMマシン全体で128k語となる。

周辺回路は連想メモリの制御および割り込みが中心である。トランスピュータが高速であるため、74F, AS-TTLとGALで構成する。

#### 7. おわりに

本稿では、意味ネットワークに基づくトータルな知識情報システムIX(イクス)と、その中核となる意味ネットワークマシンIXMのプロトタイプの概要について述べた。本プロトタイプは接続形態をプログラマブルにすることでアーキテクチャの実験・検証が行えるほか、大容量の連想メモリと高速なCPUの採用により、処理時間の大幅な向上が可能である。現在はPE基板のブレッドボード試作、そのプリント版化もおわり、32台版の製作に入っている。末筆ながら、本研究の機会を与えられた柏木電総研次長、日頃ご指導頂く棟上情報アーキテクチャ部長に感謝する。

#### 参考文献

- 1) Handa, K., et al.: "Flexible semantic net for knowledge representation", 情処欧文誌, 7, 1987.
- 2) 小倉, 他: "20Kb CAM(Content Addressable Memory) LSI", 信学技報, CPSY87-23, pp. 31-37, 1987.
- 3) Furuya, T., et al.: "Architectural evaluation of a semantic network machine," IWDM, 1987.
- 4) Higuchi, T. et al.: "The IX supercomputer for knowledge based systems", FJCC, 1987.