

## 意味記憶システム IX (イククス) の プロトタイプについて

樋口哲也 古谷立美 楠本博之 半田剣一 国分明男  
(電子技術総合研究所)

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。

IX システムは、意味ネットワークに基づく知識表現言語 IXL と、これで記述した応用プログラムを並列処理する IXM から成り、意味ネットワーク処理をトータルな形でサポートする。

IXM のプロトタイプは、32 台のトランスピュータと大容量連想メモリを備えることによって約 2~3 万リンクまでの大規模意味ネットワークを並列処理することができる。また PE 間の接続形態をプログラムで変更することができ、最適なアーキテクチャの検証や実験が行いやすくなっている。

### 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, Sakura-mura, Niihari-gun, Ibaraki 305

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 enable the parallel processing of a large semantic network of 20000 to 30000 links. In order to evaluate various connection schemes, the prototype has reconfigurable structure.

## 1. はじめに

意味ネットワークは代表的な知識表現形式の一つとして知識ベース、自然言語処理、記憶モデルなど、これまでに広く用いられてきた。これは、グラフとノードによる図的表現のわかりやすさや、階層的知識の表現能力によるところが大きい。しかし、意味ネットワークの提案以来二十年近くを経ても、その利用形態はいまだ実験的な領域を出ず、これまでに大規模な意味ネットワークにもとづくシステムが作られた例はない。

この理由として第一に挙げられるのは、意味ネットワーク処理の並列性を生かすことのできる計算機アーキテクチャが十分にないことである。つまり、たとえ大きな意味ネットワークを構築して処理しようとしても多大な時間がかかり、とても実用にはならないことが多い。意味ネットワーク処理でよく行われるのはパターンマッチであるが、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し、多少計算機単体の速度をあげたところでそれに追従できない。つまり逐次的な探索アルゴリズムでは大規模意味ネットワークの処理に限界があり、もっと抜本的な解決策として、並列探索が行える計算機アーキテクチャが必要となる。

一方、意味ネットワークの応用は多岐にわたっており、これまで提案された種々の意味ネットワークはそれぞれのニーズに特化したものとなっている。たとえばリンクの意味付けやノードの表わす概念の抽象度にしても、実に様々である。したがってそれら各種の応用を柔軟に記述できる知識表現言語を提案し、その言語で記述したプログラムを上述の並列探索が行える意味ネットワークマシンが実行できる形態が望ましい。

このような実用規模での意味ネットワーク処理を実現するために、筆者らは意味記憶システムIX（イクス）を開発している。IXは、(1) 意味ネットワーク処理用並列マシン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）から成り、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。ここでは、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、そのIXLプログラムをIXMが高並列で実行する。IXはこのような処理の全階層をトータルな形でサポートすることを目指している。

筆者らはすでに意味ネットワークに基づく知識表現言語IXL（イクスル）を開発し[半田85]、これを用いて知識ベースや知識獲得システムを作成しているが、IXLが逐次計算機(micro VAX II)上のprolog（インタプリタ）で実装されているために速度が遅く、数千リンク以上の意味ネットワークになると実験効率が

著しく低下する状況にある。

このため、すでに提案している意味ネットワークマシンIXMのアーキテクチャ[樋口86]の検証と実験をソフトウェアシミュレーションで行うのがむずかしくなっている。また性能評価をおこなうにしても、実機に近いプロトタイプの上で運用ソフトウェアを開発し、それら実際の種々のオーバーヘッドを含めた上でのデータで評価したい。

このため筆者らは意味ネットワークマシンIXMのプロトタイプとして32台のトランスピュータと連想メモリから成るシステムを試作している。

本稿では、IXシステムの概要を示すとともに、試作中のプロトタイプについて述べる。

## 2. IXシステムの概要

### 2.1 意味ネットワーク

意味ネットワークによる知識表現では、ノードが「概念」を、リンクが二つの概念間に成り立つ「関係」を表わす。

意味ネットワークに対する処理は、意味ネットワークの各ノードにマーカビットと呼ぶ1ビットのフラグを複数個用意することによって効率良く扱える。マーカを用いる意味ネットワーク処理の基本演算として次の3つがある。

(1) 連想処理：ノード名や、特定のマーカビットがセットされているノードを連想によってさがす。

(2) 集合演算：セットされたマーカビットをもとに、ANDやORなどの集合演算を行う。

(3) マーカ伝搬：特定の属性をみたす一連のノード群をマークするための操作であり、あるノードから別のノードへと、マーカを次々に転送していく。

これらの基本操作を用いて、意味ネットワーク処理がどのように行われるかを示す。

いま図1の意味ネットワークがあり、AグループとBグループに属する人間について表わしているとする。これに対してAとBの両グループに属している人を求めよという命令がでた場合、次の処理がなされる。まず連想処理によって、Aグループのノードを見つけ、そのノードのマーカビット1番をセットする。次にAグループに属する人間を求めるためにマーカ伝搬を実行する。これは、マーカビット1番が立つノードから順番にisaリンクをたどっていき、途中で通過するノードのマーカビット1番をセットすることに対応する。同様にして、Bグループのメンバーも

求めるが、この場合は各メンバーのノードのマーカビット2番をセットする。

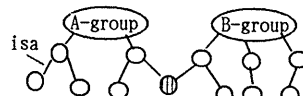


図1 意味ネットワークの例

A, B 両グループに属するメンバーは、1つのノード内でマーカビット1番と2番が両方とも立っているからマーカビット1番と2番のANDをとる集合演算を行うと答えが出る。

これらの処理のうち、マーカ伝搬は、意味ネット処理に特有のものである。図1の意味ネットならネットの深さである2ステップで集合のメンバーを求められ、たとえ大規模意味ネットワークの場合でも、ネットワークの深さに比例したステップ数で集合の要素を求められる。

## 2. 2 IXL

IXLは図2に示すように、ロジックの述語に似た形式のコマンド言語であり、知識ベースの記述、検索、修正の機能を持つ。例えば、知識ベースを記述するためのコマンド、link(is\_a, penguin, bird)は、「ペンギンは鳥である。」を示す意味ネットワークを生成する。このあと、検索コマンドとしてisa(penguin, X)を与えれば、解としてbirdが得られる。

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。

To construct a relation:      To connect nodes by a link:  
 assertion(R, X, Y).            link(is\_a, X, Y).  
 property(R, X, Y).            link(not\_isa, X, Y).  
                                      link(instance\_of, X, Y).

To inquire about a link:  
 isa(X, Y).                        link(not\_instanceof, X, Y)  
 instance(X, Y).                 link(a\_kind\_of, [X, Y, ...], Z).  
 ako(X, Y).                        link(source, R, X).  
 source(R, X).                    link(destination, R, Y).  
 destination(R, Y).              link(rule, X, ((

To inquire about a relation:  
 asst(R, X, Y).                    prop(R, X, Y):-....  
 prop(R, X, Y).                    isa(X, Y):-....  
                                      instance(X, Y):-....)).

図2 IXLコマンド

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と、旅行費用が、交通費と滞在費の合計であると考えるならば、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。

$$\begin{aligned} \text{asst}(\text{旅行費用}, X, Z) : - \\ \text{asst}(\text{交通費}, X, Y), \\ \text{asst}(\text{滞在費}, X, W), \\ Z \text{ is } Y + W. \end{aligned} \quad \text{式 1}$$

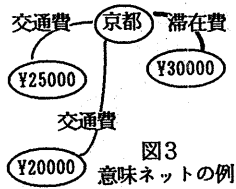


図3 意味ネットの例

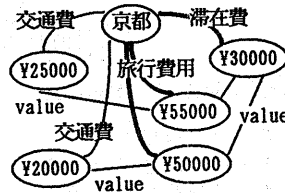


図4 数値演算の実行結果

交通費が交通手段によってこの場合2通りあるので、旅行費用も2通りとなり、この手続き的知識の実行結果として2つの旅行費用リンクと各費用のノードが図4のように

新たに意味ネットワークに追加される。このような手続き的知識の実行結果としての新ノード、リンクの生成は、実際のハードウェア上でそれらのノードやリンクが新たに割り付けられることに対応している。

## 2. 3 IXM

### 2. 3. 1 全体構成

IXMマシンは、ホストコンピュータの制御下でIMD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を置く。ネッ

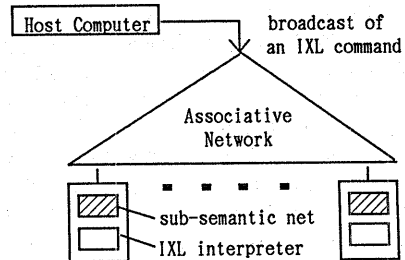


図5 IXMの全体構成

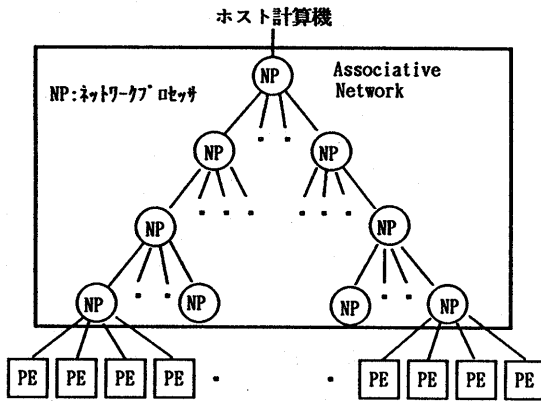


図6 木構造の場合のIXMの構成

ワークプロセッサも連想メモリを備えており、プロセッサ間通信を行うほかに、連想メモリを用いて後述するマーカー伝搬の並列化処理を行う。

IXMマシンにブロードキャストされるIXL命令には主に3つのタイプがある。

第一のタイプは、1つのIXL命令が唯一つのIXL命令に変換されるもので、これは与えられたIXL命令が直接的に事実を問い合わせたり、宣言的な知識を記述するといった場合に対応する。

第2のタイプは、手続き的知識の実行に伴って、新しいノードやリンクを生成するための命令群である。たとえば式1の実行結果として図4では新しく「旅行費用リンク」を張る必要があり、その際リンク接続先の決定や実際のリンク生成のための命令が必要となる。

第3のタイプは手続き的知識の記述の中に現れる算術論理演算である。なおこの場合にもリンク生成命令が必要となる。

### 2.3.2 命令階層

ここではIXL命令がどのように実行されるかについてのべる。図7はIXシステム内の命令階層を示している。

まずユーザからIXL命令がIXシステムに対して与えられると、これをIXMのホスト計算機上のIXMコントローラが受け付ける。IXMコントローラはOSであり、IXL命令を「IXL命令」に変換して、それを連想ネットワーク経由で各PEへとブロードキャストする。また手続き的知識が定義されている場合には、そのボディを構成するIXL命令を順番に一つずつIXL命令に変換してIXMにブロードキャストする。たとえば前ページ式1の旅行費用の手続きの場合、 $asst(X, Y)$ 、 $asst(X, W)$ の二つのIXL命令に対応するIXL命令、そして「 $Z$  is  $Y + W$ 」に対応する数値演算用のIXL命令に変換してIXMへブロードキャストする。

IXL命令は、IXMの各PEの持つIXLインタプリタ内でそれぞれ一つの命令処理ルーチンに対応しており、各PEによってインタプリタされる。この各PEの実行する種々のIXL命令処理ルーチンは、「IXM機械語命令」で記述されている。

このIXM機械語命令は意味ネットワーク処理の並列性を活かす目的から、その命令実行の開始が非同期的になっている（マーカーの到着によって実行が開始される）。IXM機械語命令の実行は図7にもある通り、さらに「IXMマイクロ命令」の形でエミュレートされて実行される。ただし今回試作するプロトタイプシステムではIXL機械語命令をトランスペュータでエミュレートする。

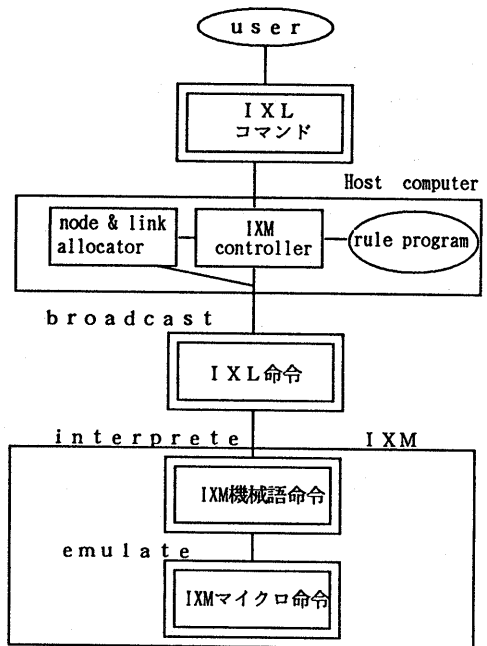


図7 IXMの命令階層

### 2.3.3 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。

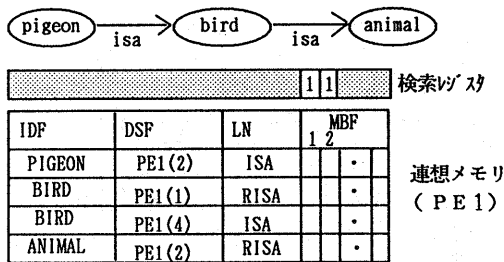
#### (1) 意味ネットワーク記憶・処理用連想メモリ

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になる。

意味ネットワークを連想メモリに格納した様子を図8に示す。連想メモリへの格納はリンクを単位としている。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ービット領域から成る。一リンクは両端にノードを持つので、一リンクにつき連想メモリの2語が占められる(図8参照)。

マーカー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1の意味ネットワークではAグループのメンバーをマ



LEGEND IDF: identifier field LN: link name  
DSF: destination field -> PE No. (displacement)  
MBF: marker bit field RISA: Reverse ISA

図8 連想メモリへの意味ネットワークの格納

ーカービット1番、Bグループのメンバーをマーカービット2番で識別したが、両者の共通要素をもとめる積集合の演算は、連想メモリの検索データレジスタを図内のそのようにセットすれば、ただ一回の連想メモリアクセスで実行することが可能である。言い替えれば、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・集合演算は常に一定時間内で実行できることになる。

(2) 連想ネットワーク

図5で示した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間のマーカー伝搬、(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、またマーカー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、これはマーカー伝搬における並列性を生かす目的を持つ。

マーカー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ーを流すと、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の場合にはマーカー伝搬の並列性を生かすメカニズムが必要である。equivalenceノードと呼ぶのが、IXMに導入した方式であり、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ーを流し、大学に属する全学生をマーカー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にわりつ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ー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、マーカー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ノードにマーカーが送られてきたとき、そのマーカーは複製されて、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、それが物理的につながっているstudentノードのマーカービットがセットされたならば、他のstudentノードのマーカー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ー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からのマーカー伝搬の並列化の様子を示す。これは、PE

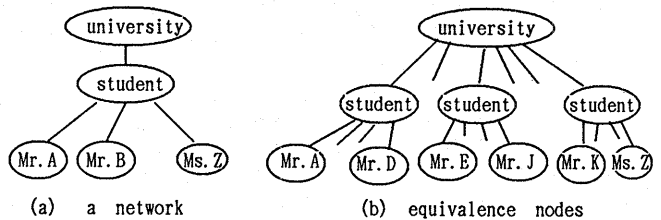
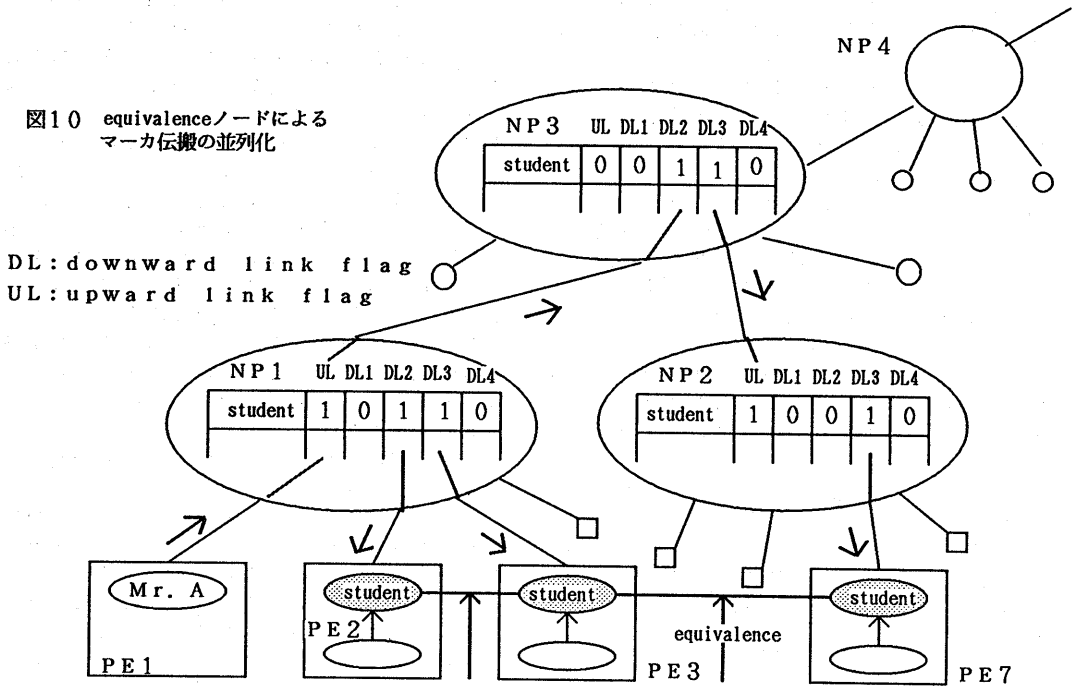


図9 equivalenceノード

図10 equivalenceノードによる  
マーカ伝搬の並列化



1にあるMr. AのノードからPE 2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PE 3、およびPE 7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。

たとえば1600リンクの意味ネットワークに対する質問処理が約2分であるのに対し、意味ネットワークが約3倍の4500リンクになると約15分かかり、探索空間の増大のために実行時間が著しく増加する。

試作にあたっては次のような方針をとる。

### 3. プロトタイプシステムの概要

32台のトランスピュータと連想メモリから成る意味ネットワークマシンIXMのプロトタイプの概要について述べる。

#### 3.1 試作の目的と方針

試作の第一の目的は、提案しているアーキテクチャに近い性能を確保しつつ、これらのアーキテクチャの検証や実験を柔軟に行えるシステムを作ることである。特に連想ネットワークの接続形態についての実験を重視しており、これまでに提案した木構造の有効性の確認、あるいは木構造と同じく通信の局所性を生かすことのできるハイパーキューブ等との比較検討を行っている。

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(一万リンク以上)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。現在の環境はすでに述べたようにIXLが逐次計算機(micro VAX II)上のprolog(インタプリタ)で実装されているために速度が遅く、数千リンク以上の意味ネットワークになると実験効率が著しく低下する状況にある。

- ・接続形態の変更を柔軟に行えることを中心に考える。すなわちトランスピュータ間の接続をスイッチ基板を介して行い、このスイッチ基板をプログラマブルにすることで、任意の接続形態を実現する。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。
- ・PEの台数とNPの台数が接続形態に応じて変わるため、PEにもNPにも共通に使えるPE基板を一種類作成する。
- ・ソフトウェア開発環境としての目的があるため、高速性よりも安定動作を中心に考える。このため、提案しているマーカ駆動型命令制御やパイプライン方式はハードウェア化せずに、トランスピュータでエミュレートする。またトランスピュータ間のデータ転送も、並列転送のための回路を新たに設けずに、トランスピュータ自体が持つ4本のシリアルリンクを利用する。ただしIXMマシンのポイントである全解探索機能を実現し、提案しているアーキテクチャに近い性能を得るためには、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。
- ・システム記述言語としてoccam2を用い、並列

同期もこれで記述できる範囲で行う。このため、occam2のシリアルリンクプロトコルを超える同期機構はハードウェアでサポートしない。ただし、IXLインタプリタのカーネル部分やトランスビュータ間通信プログラムなど、ボトルネックになる可能性がある箇所については、トランスビュータの機械語、もしくはCを用いてプログラムする。

### 3.2 プロトタイプの概要

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4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。

スイッチ基板とPE基板は、ともにダブルハイトのVMEサイズであり、これら36枚の基板を20スロットずつ、縦2段の構成で19インチラックに収める。

#### 3.2.1 ホスト計算機

ホストとなるIBM PCにはトランスビュータ(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、ホストのトランスビュータに割り込みをかける。

#### 3.2.2 スイッチ基板

IXMマシン内の32台のトランスビュータの持つシリアルリンクの総数は128本である。このほかに、8枚のPE基板についてはさらに4本のシリアルリンクが増設されている。これらを4枚のスイッチ基板に分けて接続する。スイッチ基板には、一枚につき2個のリンクスイッチLSIを実装する。一個のリンクスイッチ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ビ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

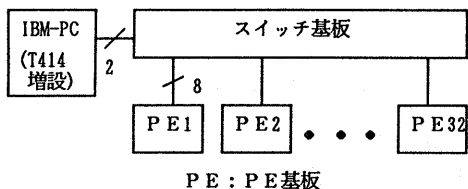


図11 プロトタイプの概要

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、あるいは図12に示すようなハイパーキューブ構成も実現することができる。図12は1ネットワークプロセッサ当たり3台のPEが接続される場合を示している。

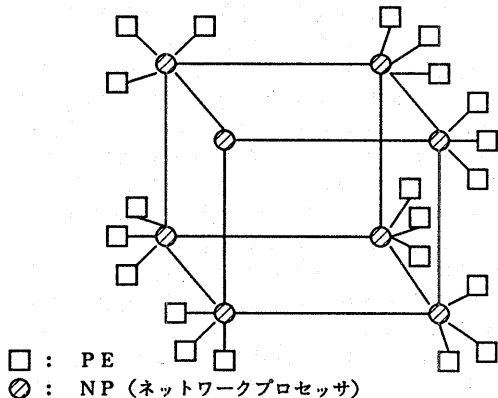


図12 3次元ハイパーキューブ接続時の構成

#### 3.2.3 PE基板

PE基板は、図13に示すようにT414トランスビュータ、連想メモリ、SRAM、リンクアダプタ、および周辺回路から成る。

トランスビュータはプロセッササイクルが17.5MHzで、およその性能は約8.5MIPSである。内部にはサイクルタイム59nsの2KByteオンチップRAMを備えており、ここにIXLインタプリタ等、使用頻度の高いプログラムを置く。外部メモリは32K語 X 32ビットのSRAMで、サイクルタイムは230nsである。トランスビュータの4本のシリアルリンクは、5Mbit/s、10Mbit/s、20M

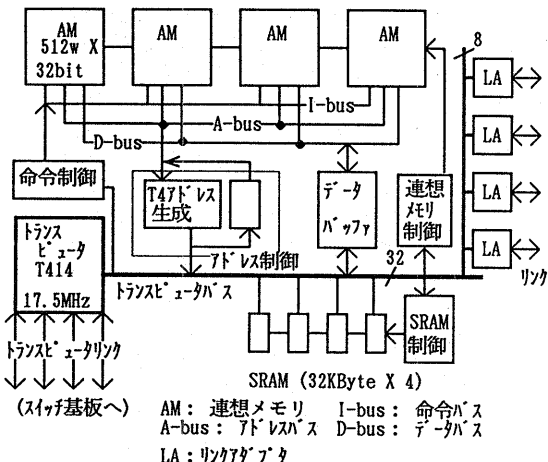


図13 PE基板のブロックダイアグラム

bit/sの3種類の速度がえらべるが、今回は10M bit/sを標準に設定し、96bitのDINコネクタ経由でスイッチ基板へ接続する。

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。バイト単位の転送ごとにトランスピュータに割り込みがかかるが、トランスピュータの割り込み処理時間が約1マイクロと短いこと、増設したリンクについてもoccamでプログラミングできる利点があることから、この方式を採用した。

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と同じにアクセスできる。サイクルタイムもSRAMと同じ230nsである。図14にoccamのアドレス空間と連想メモリの割り付けを示す。

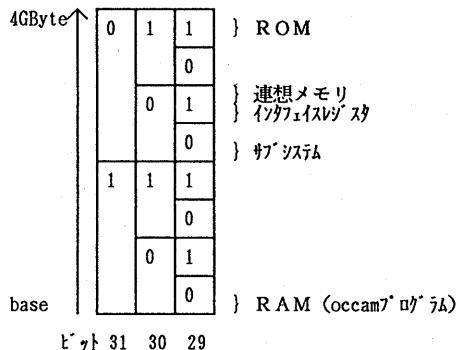


図14 occamアドレス空間と連想メモリの割り当て

連想メモリには意味ネットワークを展開して格納し、更にこの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32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、マーカビットとして用いることができる。残りの24ビットはバイト単位でマスクできる。IXMのワード構成はこの連想メモリの2ワードを用いて図15のようになる。すなわち最初のワードの第一、第二バイトをあわせてidentifier領域とし、第三バイトをリンク領域とする。残りの8ビットのうち、7ビットはマーカビット領域として用い、1ビットは最初のワードであるのか、2番目のワードであるのかの識別に用いる。2番目のワードの第一、第二バイトはdestination領域とし、第三バイトはリテラル保持のための呼び、残りの8ビット

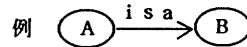
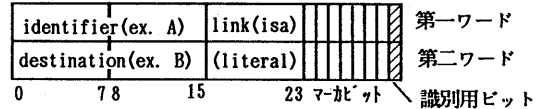


図15 連想メモリ上のリンク表現

は識別用1ビットを含むマーカビット領域として用いる。

このように連想メモリの設定を行うことで、一つの意味ネットワークのリンクにつき、identifier領域、destination領域として各16ビット、マーカビット領域として14ビットとなる。つまり、この1リンクあたり2ワードの構成では3万2千リンクの意味ネットワークを扱うことができる。より大規模の意味ネットワークを処理する場合や、こみいった手続き的知識を扱うためにマーカビット領域が不足する場合には順次ワード構成をふやせばよい。

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。つまり連想メモリをoccamのアドレス空間内の複数の別々のアドレス領域に重複して割り付けており、動作モードごとに別々のアドレス領域をアクセスすることで、そのとき出るアドレスの一部を動作モードの設定に利用する。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2k語を実装する予定であり、IXMマシン全体で64k語となる。周辺回路は連想メモリの制御が中心である。トランスピュータが高速であるため、74F、AS-TTLとPALで構成する。

#### 4. 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トランスピュータの採用により、処理時間の大幅な向上も可能である。現在はブレッドボードを試作している。末筆ながら本研究の機会を与えられた柏木寛電子計算機部長と貴重な助言を頂いた内堀義信技官に感謝する。

#### 参考文献

[半田85] 半田、他：“意味記憶システムIX - IXLによる知識表現”、本研究会、3月、1985。

[樋口86] 樋口、古谷、国分、楠本、半田：“並列連想記憶を用いた意味ネットワークマシン”、信学技報、EC85-55、1986。