

TM-0098

並列推論マシン・アーキテクチャの調査

ICOT Working Group WG1
並列推論マシン・サブグループ

March, 1985

©1985, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

— 目 次 —

はじめに	．．． 1
1. 電々公社 通研データフローモデル	．．． 2
2. ICOT データフロー方式並列推論マシン(PIH-D)	．．． 6
3. 神戸大 K-PROLOGモデル	．．． 10
4. ICOT (富士通) 節単位処理モデル	．．． 14
5. 東大 PIH:Parallel Inference Engine	．．． 18
6. ICOT リダクション方式並列推論マシン(PIH-R)	．．． 26
7. 電総研 プロセスグラフ・モデル	．．． 30
8. 京大 並列リダクション・モデル	．．． 34
LIST OF CONTRIBUTORS	．．． 38

はじめに

ICOTのワーキンググループWG1 は、並列処理アーキテクチャをテーマに活動を続けて来たが、活動内容は並列推論マシン、知識ベースマシン、VLSIの3つに分けられ、それぞれのサブグループが存在する。本テクニカルメモは、その内、並列推論マシンサブグループでまとめたものである。

並列推論マシンは、KL1 等の並列論理型言語を効率良く支援するハードウェアであり、第五世代コンピュータの推論サブシステムを構成する基本ハードウェアであると考えられる。

本メモは、従来までに各所で行われて来た並列推論マシン研究をまとめたものである。この方面の研究は未だ始まってから3～4年にしかないのでその数は余り多くない。ここで取り上げたものは日本国内の研究に限られているが、論理型言語を強く志向したこの方面の研究は、世界的に見ても現在の所これ以外そう多くない。従って、このメモは、現時点で存在する代表的な推論マシンの研究であると言ってもよからう。

勿論、この他に、スウェーデンのOR並列トークンマシンがある他、関数型言語マシンとしてはインペリアルカレッジのALICE、ノースカロライナ大学のMagoマシン等多く存在するし、意味ネットワークのハードウェア支援を考えたConnection Machine等もある。

従って、広い観点からすればほんの一部ではあるが、それでもこのメモ内の研究は、並列推論マシンの代表的なものであって、このメモが現時点におけるそれらの状況を伝える良い資料となるであろう。

元々このメモは、WG1 内の並列推論マシンサブグループの59年度活動に於いて、諸並列推論処理のモデルと、マシンアーキテクチャを整理し、各モデルの位置付けや特徴を明らかにしようということから生まれたものである。実際の活動は1984年 9月より1985年 1月まで行い、各マシンについてその研究者と密な議論を行ったが、このメモ内の記述はその時用いた資料を基にしたものである。

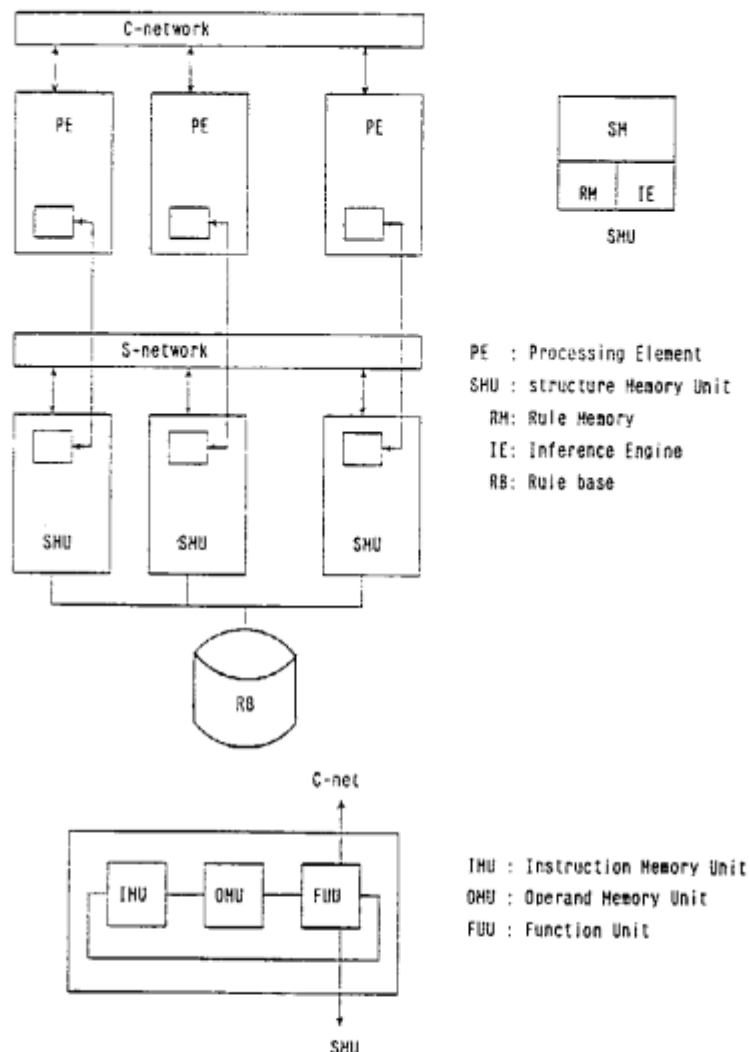
もとよりこのメモは、各研究のすべてを含んでいない。各研究のポイントをつかむ意味から、各マシンの処理モデル、特徴、マシンアーキテクチャの概要等に内容をしぼり、定型形式で各マシンの研究者に書いていただいたものを集めたものである。より詳しい内容等は、各研究の末尾に付けられた参考文献リストを参照されたい。このメモが、今後の並列推論マシンの研究に役立てば幸いである。

WG1 主査 田中 英彦
(東京大学)

(1/4) 並列マシン・アーキテクチャの研究 (昭和59年 8月現在)	
並列処理 モデル名 or マシン名	通研データフローモデル (名称は未定)
研究機関	日本電信電話公社 武蔵野通研
主研究者	雨宮 真人、長谷川 隆三 他
研究経過 (開始時期、現状、予定等)	
1981年	直接実行モデル (clauseを双方向リンクのデータフローグラフへ変換)
1982年	DFM-1 の設計 証明木探索モデル (goal木の導出をデータフロー制御で行う)
1983年	DFM-2 (リスト処理用) の設計 Valid 言語処理系の作成
1984年	DFM-2 試作 Valid による並列Prologインタプリタ作成
1984年11月	DFM-2 搬入予定
並列処理モデルの主な特徴	
1. データフロー制御によるOR並列、AND パイプライン処理モデル、インタプリタ実行形式。 2. 先行評価 (Eager evaluation) 機構を使ってAND パイプライン処理を達成 (結果が求まる度に次のatom処理に引渡す)。 ORプロセスの爆発を防ぐため、遅延評価 (lazy evaluation) 機構とカウンタを用いて、起動プロセス数を制御。 3. 変数管理、プロセスの生成・消滅管理を容易にするため環境木を生成 (配下のプロセスから得られる結果をconsによりまとめ、木構造化する)。	

(2/4) 並列処理モデルの概略	
想定規模	1000台規模 現在規模 (DFM-2) PE 8台/SH 8台
対象言語 及び 対象分野	Valid (ボタン照合、モジュール、集合演算機能をサポート) 現時点: 並列Prolog
目標性能	現在、1PE あたり2MOPS (car/cdr 等のリスト演算が2M回/秒) 20K × 1000 LIPS ?
(概略説明) 1. 証明木探索モデルのリダクション型実行過程を関数実行の枠組の中で、データフロー制御により実現する。(clause body 中の1つのgoalアトムを解くORプロセス、head照合に成功したclause body 全体を解くAND プロセスを関数で記述) 2. 先行/遅延評価機構をサポートする構造体メモリを備えた、リスト処理用データフローマシンでインプリメントする。 Valid で記述した並列Prologインタプリタ(OPLOG) を、データフローマシン上で実行させる。 3. OR並列、AND 逐次実行を基本とするが、先行評価機構を用いて、AND のパイプライン処理を実現する。ORプロセスの爆発を抑えるため、遅延評価機構及びカウンタを用いて、ORプロセスの生成数を制御する。 4. カウンタの指定法を変更することにより、depth-first/breath-firstの探索ストラテジーを適宜使い分ける。 5. 変数管理を容易にするため、OR-connectivesを解くことにより得られる結果の環境を木構造化し、次のgoalアトムの処理に引渡す(1次元ストリーム化及びストリームのマージ操作は行わない)。	

(概略図)



(主な特徴)

1. データフロー制御により多重プロセス制御 (colored token 方式)
2. 各PEにはインタプリタの核がロードされる。
リスト演算はSH内で実行。Lenient cons, Lazy consなどの先行・遅延評価機構はSHでサポートする。
3. 将来、推論処理用に命令を高機能化する予定 (Unifier などの機能をハードウェア化しSHに付加)。
4. 各PEはローカルSHU を有しSHへの高速アクセスを可能にする。但し、SHアドレスはグローバルにし、他のSHU へもアクセスできるようにする (SH全体で1つのアドレス空間を形成)。
5. C-network は論理的には木構造を実現する。物理構成としては、アレイ、ルーチング・ネットワーク、マルチリングバスを検討中。

(4/4)

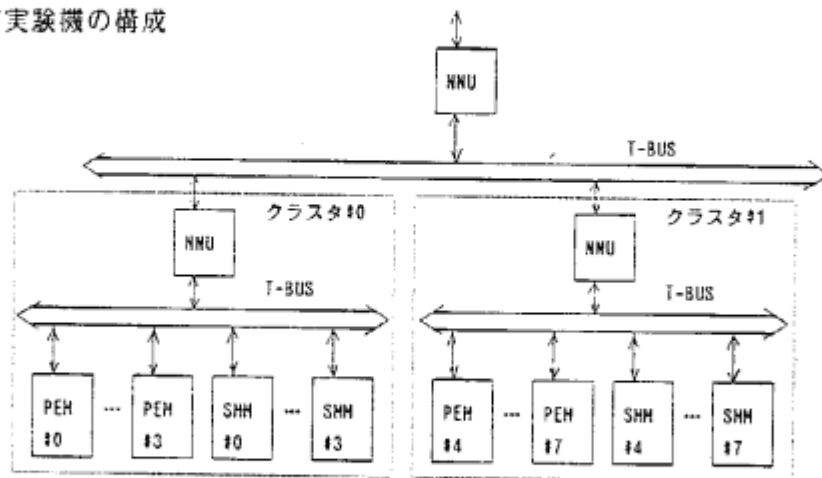
〈参考文献〉

- [1] 雨宮、長谷川、橋爪、“データフロー概念に基づく推論実行方式の検討”、
信学技法 EC82-30, 1982.
- [2] 雨宮、長谷川、“データフロー制御による論理型プログラム実行機構”、
Proc. of Logic Programming Conference'83, Tokyo, ICOT, 1983.
- [3] M. Amamiya, R.Hasegawa, O.Nakamura and H.Mikami, "A List-Processing-Oriented Data
Flow Machine Architecture," Proc. of the 1982 National Computer Conference, AFIPS,
1982, pp-143-151.
- [4] M. Amamiya, R.Hasegawa and H.Mikami, "A List Processing with Data Flow Machine,"
Lecture Notes in Computer Science, No.147, Springer-Verlag, 1983, pp.165-190
- [5] R.Hasegawa and M. Amamiya, "Parallel List Processing using Data Flow Machine,"
Trans. of IECE (D), J66-D,12,pp.1400-1407, Japan 1983.
- [6] M. Amamiya, R.Hasegawa and Y. Kiyoki, "Eager and lazy Evaluation Mechanism in Data
Flow Architecture and Its Application to Parallel Inference Machine, "Proc. of
Work Meeting for Computers, IECE, Japan, Nov., 1983.
- [7] M. Amamiya and R.Hasegawa, "Dataflow Computing and Eager and Lazy Evaluation," New
Generation Computing, vol. 2, No.2, 1984.
- [8] R.Hasegawa and M. Amamiya, "On the Implementation of Lazy Evaluation with Data Flow
Machine, "Proc. Ann. Conf. IPSJ, Japan, 1982.
- [9] M. Amamiya, R.Hasegawa and Y. Ono, "Valid, A High-Level Functional Programming
Language for Data Flow Machines," To appear in Review of the E.C.L. Vol.32, No.5,
1984.

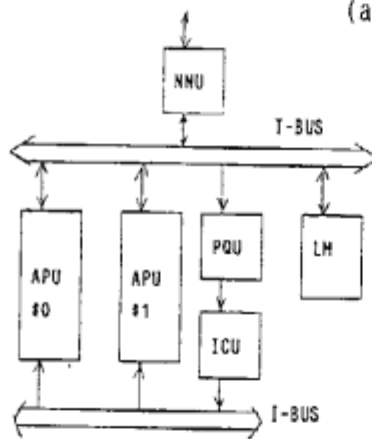
(1/4) 並列マシン・アーキテクチャの研究 (昭和59年10月現在)	
並列処理 モデル名 or マシン名	データフロー方式並列推論マシン (PIM-D)
研究機関	ICOT (沖電気工業)
主研究者	伊藤 徳義、清水 肇、来住 晶介、久野 英治 他
研究経過 (開始時期、現状、予定等)	
1982年夏 処理モデル検討 1983年 マシンの詳細シミュレーション開始 1984年春 実験機の設計開始 1985年春 実験機試作完了予定	
並列処理モデルの主な特徴	
1. データフローモデルに基づく論理型言語の並列実行。 OR並列、AND 並列、及び引数間の並列性を実現。 2. OR並列はストリームを使用して制御。 3. AND 並列を実現するために、変数を共有変数と非共有変数に区別する。無矛盾性検査は共有変数に対する結合情報の間で行う。 4. 構造データはプロセス間で共有され、必要になった時にコピーされる (遅延コピー)。	

(2/4) 並列処理モデルの概略	
想定規模	中期末に 100台規模
対象言語 及び 対象分野	核言語第1版 (KL1)
目標性能	中期末で1MHUPS(Head Unification Per Second) 程度か？
(概略説明) 1. 論理型プログラムをデータフローグラフにコンパイルし、実行する。このため、統一化や非決定性制御のためのノード（プリミティブ）を命令セットとして用意する。グラフは処理要素内の命令メモリ（IH）に格納される。 2. ゴールが与えられるとそのリテラル引数（群）は対応する定義のグラフに与えられ、定義節群を起動する。起動される節を実行するプロセスの割当ては動的に行われ、プロセスが他処理要素に割当てられたとき、ゴール引数は処理要素間で通信される。処理要素の割当ては個々の処理要素のファームウェアで制御される。 3. 各処理要素は互いに独立動作するPQU(Packet Queue Unit)、ICU(Instruction Control Unit) 、及び、2台のAPU(Atomic Processing Unit) から構成され、これらがパイプライン動作しながらグラフを並列インタプリートする。 4. 構造データは構造メモリに格納され、処理要素にはそのポインタが渡される。処理要素は、構造データの内容を参照する必要がある際、メモリ側に要求を転送する。	

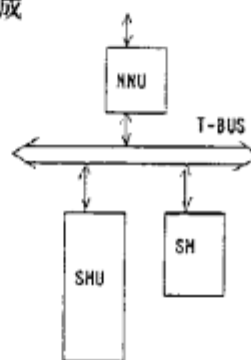
(概略図) 試作実験機の構成



(a) 実験機全体構成



(b) PEH の構成



(c) SHM の構成

(主な特徴)

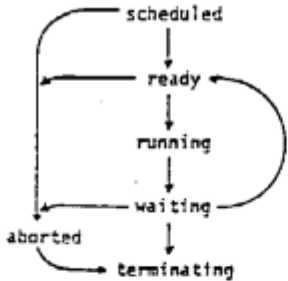
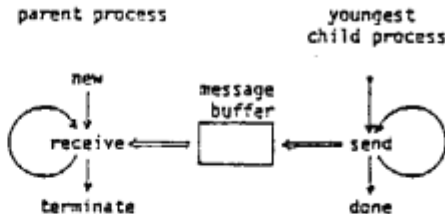
1. モジュール内通信及びモジュール間通信網を階層型バス構成で実現。
各モジュール (PEH 又は SHM) 内の局所性、クラスタ (PEH $\times$ 4、SHM $\times$ 4) 内の局所性が生かせる。
2. プログラム (データフローグラフ) は PEH に格納し、構造データは SHM に分散して格納する。
3. 構造データアクセスの必要が生じた際、PEH から SHM へ指令を送出。PEH はアクセス結果の到着を待たずに他の処理を続行する (低レベルの並列性)。
4. 構造データやプロセスの割当ては、各 PEH の LM 内の管理表により管理。

(4/4)

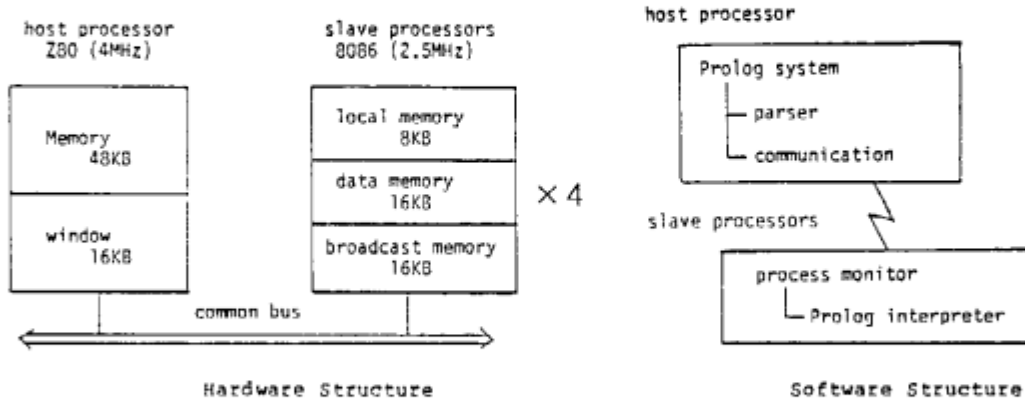
〈参考文献〉

- [1] 伊藤、尾内、益田、清水、“データフロー方式の並列Prologマシン”
Proc. of Logic Programming Conference '83, ICOT, 1983.
- [2] 伊藤、益田、清水、“データフロー方式Prologマシンにおける非決定的制御機構”
情報処理学会第27回全国大会、1983.
- [3] 益田、伊藤、清水、“データフロー方式Prologマシンのシミュレーションによる評価”
情報処理学会第27回全国大会、1983.
- [4] 伊藤、来住、“データフローマシン上でのConcurrent Prolog 機能の実現”
情報処理学会第28回全国大会、1983.
- [5] 伊藤、久野、“並列Prologマシンのシミュレーションによる評価”
情報処理学会第28回全国大会、1983.
- [6] 伊藤、益田、清水、来住、久野、“データフロー方式並列推論マシンのアーキテクチャ”
Proc. of Logic Programming Conference '84, ICOT, 1984.
- [7] Ito, N., and K. Masuda, "Parallel Inference Machine Based on the Data Flow Model",
Int'l Workshop on High Level Computer Architecture, Los Angels, May, 1984.

(1/4) 並列マシン・アーキテクチャの研究 (昭和59年11月現在)	
並列処理 モデル名 or マシン名	K-Prolog
研究機関	神戸大学工学部システム工学科
主研究者	松田秀雄、田村直之、金田悠紀夫、前川禎男 他
研究経過 (開始時期、現状、予定等)	
1982年春	モデルの設計
1982年10月	ブロードキャストメモリ結合形並列計算機(KUPS)上にシステム第1版を作成 (8086× 4台)
1983年夏	KUPS-Ⅱ完成(8086× 8台)
1983年冬	KUPS-Ⅱ上にシステムを作成 OR並列を導入
1984年夏～	MC68000 による並列Prologシステムの検討
並列処理モデルの主な特徴	
1. AND リテラル間でのパイプライン並列とOR並列 2. AND-OR木に基づいたプロセス単位の処理 3. プロセス間にメッセージバッファを設け、send、receive で通信 柔軟な処理が可能、not やcut の処理も比較的容易 4. 共有メモリによるストラクチャ・シェアリング	

(2/4)	並列処理モデルの概略
想定規模	4～64台程度
対象言語 及び 対象分野	Prolog+cut
目標性能	(10K～20K)LIPS×(4～64) 台
(概略説明) 1. 各プロセッサ上にプロセスモニタを設け、マルチプロセス処理を行う。ready、running、waiting、terminating などの状態がある。 2. プロセス制御あるいは通信用のコマンドにはnew、send、receive、terminate、done、abort があり、Prologインタプリタはこれらのコマンドを用いて記述されている。  Transition diagram of states  Communication diagram 3. 構造体はBoyer & Moore によるストラクチャシェアリング方式を使用する（連想リスト表現）。したがって、メッセージはbinding environment へのポインタである。 4. 1台のみの実行でも比較的パフォーマンスが高い。 第1版で30LIPS。 5. パイプライン並列のみの場合、システム資源を浪費せず、むしろ使用量が減る傾向にある。	

(概略図)



KUPS	thread	:link for the process list or free process list
KUPS・Ⅱでは8086 8台	locked	:flag for mutual exclusion
メモリは各64KB (1台192KB)	status	:status of the process
(1) Local Memory:	parent	:pointer to the parent
Prolog interpreter	brother	:pointer to the elder brother
Input Clauses	ychild	:pointer to the youngest child
(2) Data Memory:	pc	:program counter
Process area	mbtop	:top of the message buffer
(3) Broadcast Memory:	mblen	:length of the message buffer
New Process List	mbuff[0..3]	:message buffer
Binding Environment	arg[0..2]	:arguments
	var[0..2]	:local variables
	<process Control Block>	

(主な特徴)

1. バス結合によるマルチプロセッサ方式。
2. 各プロセッサには、ローカルメモリと、二種類の共有メモリが結合されている。
 - ・データメモリ：プロセッサごと。他のプロセッサからアクセスするときはコモンバスを使用する。
 - ・ブロードキャストメモリ：すべてのプロセッサで同一内容。書き込みのみコモンバスを通じてブロードキャスト転送する。
3. 多数のプロセッサを結合する場合は、複数バスの使用、あるいは階層的な結合方式を取り入れる必要がある。

(4/4)

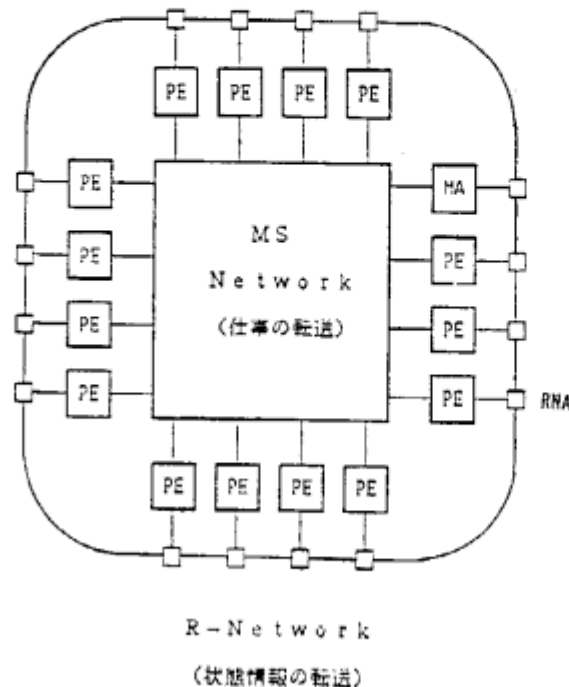
〈参考文献〉

- [1] 松田秀雄、田村直之、小畑正貴、金田悠紀夫、前川禎男
“並列Prolog処理系‘K-Prolog’の実現”
情報処理学会論文誌（採録予定）
- [2] Naoyuki TAMURA and Yukio KANEDA
“Implementing Parallel Prolog on a Multi-processor Machine ”
Proc. of 1984 International Symposium on Logic Programming ,
pp.42-48, Atlantic City, Feb. 1984
- [3] 田村直之、松田秀雄、金田悠紀夫、前川禎男
“K-Prolog（並列Prolog）の実現方法について”
Proc. of the Logic Programming Conference '83, 10.3
Tokyo, Mar. 1983
- [4] 田村直之、有尾隆一、松田秀雄、金田悠紀夫、前川禎男
“K-Prolog：並列マシン上でのPrologの実現”
情報処理学会記号処理研究会資料20-1、1982年10月
- [5] 田村直之、金田悠紀夫、前川禎男
“並列計算機上でのPrologの実現”
情報処理学会第26回大会、6D-3、1983年 3月
- [6] 松田秀雄、田村直之、小畑正貴、金田悠紀夫、前川禎男
“K-Prologの並列処理方式とその評価”
情報処理学会計算機アーキテクチャ研究会資料54-4、1984年 7月
- [7] 松田秀雄、田村直之、小畑正貴、金田悠紀夫、前川禎男
“K-Prolog（並列Prolog）の並列処理方式とその評価”
情報処理学会第29回全国大会、7B-2、1984年 9月
- [8] 小畑正貴、金田悠紀夫、前川禎男
“ブロードキャストメモリ結合形マイクロプロセッサシステムの試作”
情報処理学会論文誌Vol.24, No.3, pp.351-356, 1983

(1/4) 並列マシン・アーキテクチャの研究 (昭和59年11月現在)	
並列処理 モデル名 or マシン名	節単位処理モデル
研究機関	ICOT (富士通株式会社)
主研究者	相馬 行雄、増沢 秀穂 他
研究経過 (開始時期, 現状, 予定等)	
1983年	・ OR並列に基づく節単位処理方式の提案 ・ ソフトウェア・シミュレーション及び小規模実験機 (5台のSUN Workstation) による検討
1984年	・ 株分け方式の提案 ・ 株分け方式に基づく処理系試作 ・ 小規模実験機 (5台のSUN Workstation)によりデータ収集、評価 ・ 実験機 (16台PE) の試作
並列処理モデルの主な特徴	
1. OR並列、AND 逐次処理 2. 要素プロセッサ(PE)内では、通常の逐次型処理系の処理方式 (スタックを使用した Structure Sharing 方式及びdepth-first search方式) を基本として処理する。それに仕事の分割、転送する仕事の作成、及び転送機能を探り込んだ。 3. busy PE では、idle PE から仕事の要求があったときに限り、自分が処理している仕事を分割して分配する。 4. 均等に仕事を分散できるようにする為に、各PEの状態情報を転送し合う手段を設けた。	

(2/4) 並列処理モデルの概略	
想定規模	16～64台 現在規模 16台
対象言語 及び 対象分野	Pure Prolog + cut
目標性能	試作機：1 KLIPS/PE 16台×1 KLIPS
(概略説明) 1. 各PEは、初期ゴールと同様な形式のゴールを受けとり、逐次処理を基本として、そのゴールを最後まで処理するつもりで実行する。 2. 処理中のbusy PE は、idle PE からの仕事の要求を切っ掛けとして、仕事を分割できるか否かを調べる。(そのPEが、処理してきた探索木を最初から見て行き、まだ処理がなされていないalternative のruleが存在するかを調べる。) 3. 分割する仕事が見つかったら、その仕事を初期ゴールと同じ形式に作り上げる。その場合、定義された変数を未定義にする必要が生じるが、それを高速にする為に、各変数の定義時期を覚える手段を設けた。	

(概略図)



MA : Manager
 PE : Processor Element
 MS-Network : Multi-Stage Network
 R-Network : Ring Network
 RNA : Ring Network Adapter

(主な特徴)

1. MAは、処理に先立って、全PEのメモリに推論ベースを、MS-Networkを通してブロードキャスト転送を行う。
2. R-Network 上には、MA及び全PE数と同数のslotが存在し、MA及び各PEはそれぞれ自分のslotを持っている。各PEは、自分のslotに状態情報を乗せて他に知らせる。
3. 各PEは、推論処理中で自分の都合の良い時に、その旨をRNA に通知する。RNA はそのPEの状態とidle PE からの仕事の要求とにより、PEにidle PE からの仕事の要求を知らせる。

(4/4)

〈参考文献〉

[1] 佐藤、板敷、増沢、

“節単位処理モデルの提案”、第28回情報処理全国大会5H-8, 1984.

[2] 板敷、佐藤、増沢、

“PROLOGの並列処理システム”、第29回情報処理全国大会7B-4, 1984.

[3] 久門、板敷、佐藤、増沢、相馬、

“並列推論処理システム－改良型節単位処理方式－”、第30回情報処理全国大会, 1985.

[4] 板敷、久門、佐藤、増沢、相馬、

“並列推論処理システム－改良型節単位処理方式の実験－”、第30回情報処理全国大会,
1985.

(1/8) 並列マシン・アーキテクチャの研究 (昭和59年 8月現在)	
並列処理 モデル名 or マシン名	PIE: Parallel Inference Engine
研究機関	東京大学工学部
主研究者	元岡 達、田中 英彦 他
研究経過(開始時期、現状、予定等)	
1981年夏 ゴールフレームによる処理モデル 1981年冬 Parallog 第1版(TOPSTAR を用いた実験システム) 1982年夏 基本アーキテクチャ(PIE-I)の設計 1982年冬～ 基本アーキテクチャのシミュレーション 1983年夏～ 構造メモリの検討、単一化プロセッサ(UP)の中核部の試作 1984年春 PIE-II(構造メモリを用いた階層的構成法) 1984年夏～ 推論ユニット(IU)の試作	
並列処理モデルの主な特徴	
1. 論理型言語の実行過程を書き換えとしてとらえる。 2. 処理の単位となる中間ゴール節は必要な情報をすべて含んだゴールフレームというデータ構造でやりとりするので並列処理要素間での独立性が高い。 3. ゴールフレームの書き換えは単一化プロセッサで、探索ストラテジやプログラムの制御構造はアクティビティコントローラ(AC)で実現し、Logic とControl が分離されているので柔軟な制御が可能。	

(2/8) 並列処理モデルの概略	
想定規模	IU 1024 台程度
対象言語 及び 対象分野	Prolog+ 否定+ 多重世界+ 時相論理 (?)
目標性能	(20K ×1024) LIPS?
(概略説明) 1. ゴールフレームを単位とした書き換えモデルに基づく。 2. UPにおいてゴールフレームとそれに対応するすべての定義節の単一化を行い、複数のゴールフレームを導出する。 3. 2. のようにして生成されたゴールフレームは複数のメモリモジュールから成るゴールプールに蓄えられるが、その負荷分散は原則として各モジュール個別の判断で行い、極端に偏った場合のみアクティビティ・マネージャ(AH)が指示する。 4. UPにおけるゴールフレームの書き換えにおいて、縮退とよぶ操作によりゴールフレームから不要な情報をとりのぞき、大きさを最小限におさえる。 5. 各ゴールフレーム間の関係は、推論木とよばれる木構造により、AC内に設けられたノードメモリ内に記憶される。 6. 5. の推論木上の各ノードに種々のノード属性を導入し、ノード間でノード制御コマンドを送受することにより、探索ストラテジや言語上の拡張機能を実現する 7. ゴールフレームの大きさを小さくおさえるため、ゴールフレーム間で共有可能なデータは構造メモリに格納し、そこへのポインタのみを持ち歩く。	

(概略図)

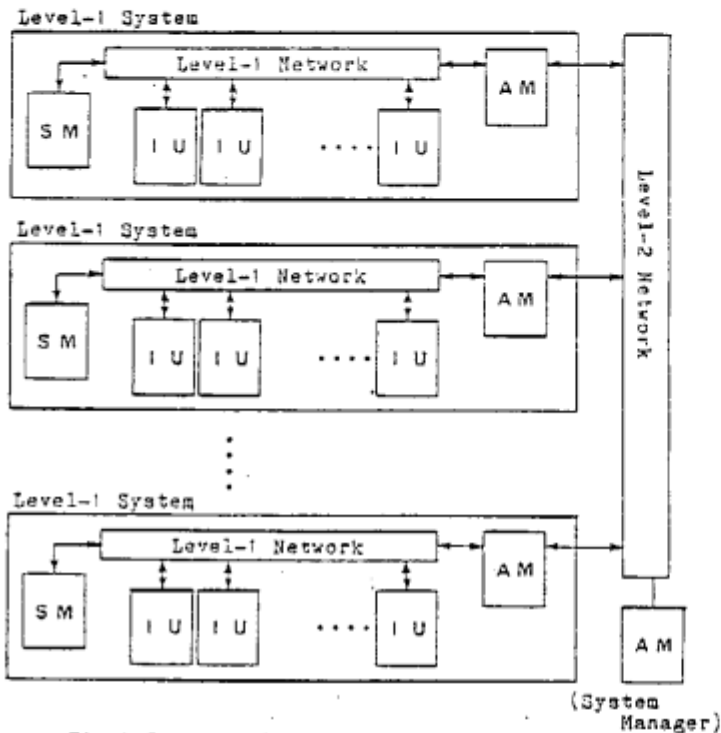


Fig.1 The Global Architecture of PIE

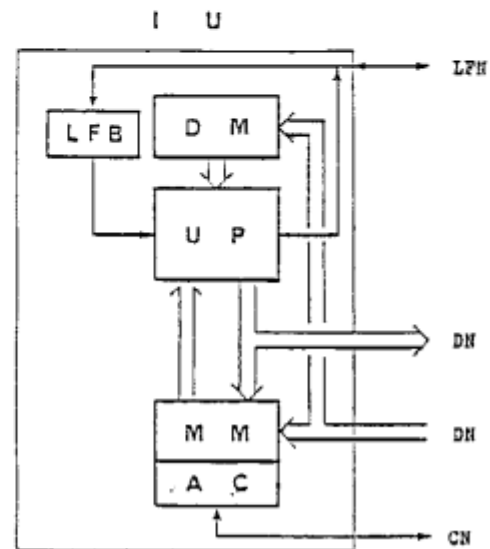


Fig.2 Inference Unit (IU)

(主な特徴)

- ・構造メモリのアクセスタイムを短くおさえるため、プロセッサ16台に対して 1台の構造メモリを設けてこれをlevel-1 システムとし、level-2 システム間では全構造データをひきつれた形でゴールフレームを転送する。
- ・モジュール間をつなぐネットワークとしては、level-1 システム内にはゴールフレームおよび定義節の分配のためのDN、AC間のコマンド転送のためのCN、構造メモリにアクセスするためのLFB の3種がある。level-2 より上ではDNとCNの2種。
- ・AMはlevel-1 システムの全てのACを監視して全体的な負荷均衡をはかるとともに、ゴールフレームをlevel-2 ネットワークに送出する際に構造データのよせあつめを行ってlevel-1 ネットワークとlevel-2 ネットワークのインタフェースをつかさどる。

A. 学会誌等

相田, 田中, 元岡,

“並列Prolog 処理システム“Paralog”について”,

情報処理学会論文誌 Vol. 24, 6, 1984.

後藤, 相田, 田中, 元岡,

“ゴール書き換えモデルに基づく論理型プログラムの並列処理方式”,

情報処理学会論文誌 Vol. 25, 3, 1984.

Aida, Tanaka and Moto-oka,

“A Prolog Extension for Handling Negative Knowledge”,

New Generation Computing, Vol. 1, No.1, July 1983.

A. Goto, H. Tanaka, T. Moto-oka,

“Highly Parallel Inference Engine PIE

— Goal-Rewriting Model and Machine Architecture —”

New Generation Computing, Vol. 2, No.1, Feb. 1984.

B. 国際会議等

Moto-oka, Tanaka, Aida, Hirata and Maruyama,

“The Architecture of A Parallel Inference Engine -PIE-”

FGCS '84 (発表予定).

相田,

“並列Prolog システム“Paralog”について”,

Proc. of Prolog Conference, Tukuba, March, 1982.

Aida and Moto-oka,

“Performance Measurement of

Parallel Logic Programming System “Paralog””,

presented in Prolog Environment Workshop, Sweden, March, 1983.

後藤, 相田, 丸山, 湯原, 田中, 元岡,

“高並列推論エンジンPIEについて”,

The Logic Programming Conference '83, ICOT, March, 1983.

M. Yuhara, H. Koike, H. Tanaka and T. Moto-oka

“A Unify Processor Pilot Machine for PIE”,

The Logic Programmig Conference '84, ICOT, March, 1984.

- T. Maruyama, H. Aida, H. Tanaka and T. Moto-oka,
“A Preliminary Evaluation of the Activity Control Mechanism
in PIE”,
The Logic Programmig Conference '84, ICOT, March, 1984.
松原, 相田, 後藤, 田中, 元岡,
“論理型プログラムのOR並列処理における冗長計算の除去方式”,
The Logic Programming Conference '84, ICOT, March, 1984.

C. 大会発表

- 相田, 松方, 鈴木, 田中, 元岡,
“データフローマシン向きPROLOG型言語に関する一考察”,
22回 情報処理全国大会 5B-3, 1981.
相田, 田中, 元岡,
“データフローマシン“TOPSTAR-II”を用いた
PROLOG並列処理試作システムの検討”,
23回 情報処理全国大会 4E-3, 1981.
後藤, 田中, 元岡,
“ユニフィケーション向き計算機に関する一検討”,
24回 情報処理全国大会 5D-4, 1982.
相田, 田中, 元岡,
“並列PROLOGシステム“Paralog”の性能測定”,
24回 情報処理全国大会 5D-5, 1982.
相田, 田中, 元岡,
“並列処理向き拡張論理プログラミング言語”,
25回 情報処理全国大会 1D-2, 1982.
後藤, 田中, 元岡,
“推論向きデータフローマシンとそのユニフィケーション機能”,
25回 情報処理全国大会 2F-6, 1982.
後藤, 相田, 田中, 元岡,
“推論向き高並列計算機システムのアーキテクチャ”,
26回 情報処理全国大会 4N-4, 1983.
丸山, 相田, 後藤, 田中, 元岡,
“推論向き高並列計算機システムのアクティビティ制御機構”,
26回 情報処理全国大会 4N-5, 1983.

湯原, 相田, 後藤, 田中, 元岡,

“推論向き高並列計算機システムのユニフィケーション機構”,

26回 情報処理全国大会 4N-6, 1983.

相田, 後藤, 田中, 元岡,

“推論向き高並列計算機システムの基本言語機能”,

26回 情報処理全国大会 4N-7, 1983.

後藤, 相田, 田中, 元岡

“高並列推論エンジンP I E〜ゴール書き換えモデルとアーキテクチャ〜”

27回 情報処理全国大会 4P-9, 1983.

相田, 後藤, 田中, 元岡

“高並列推論エンジンP I Eにおける処理の効率化について”

27回 情報処理全国大会 4P-10, 1983.

丸山, 相田, 後藤, 田中, 元岡

“高並列推論エンジンP I Eにおけるアクティビティ制御機構のシミュレーション”

27回 情報処理全国大会 4P-11, 1983.

湯原, 相田, 後藤, 田中, 元岡

“高並列推論エンジンP I E〜単一化プロセッサの構成〜”

27回 情報処理全国大会 4P-12, 1983.

平田, 相田, 後藤, 田中, 元岡

“高並列推論エンジンP I Eにおけるゴールメモリ構成法に関する一考察”

27回 情報処理全国大会 4P-13, 1983.

相田, 後藤, 田中, 元岡

“論理型プログラムの書き換えモデルにおける read only annotation の活用について”

28回 情報処理全国大会 4H-5, 1984.

松原, 相田, 後藤, 田中, 元岡

“論理型プログラムのOR並列処理における冗長計算除去の効果”

28回 情報処理全国大会 4H-6, 1984.

後藤, 田中, 元岡

“P I Eにおけるアクティビティ制御〜メタ述語の実現方式”

28回 情報処理全国大会 6F-3, 1984.

丸山, 相田, 後藤, 田中, 元岡

“P I Eにおけるアクティビティ制御〜コマンドトラヒックのシミュレーション評価”

28回 情報処理全国大会 6F-4, 1984.

湯原, 相田, 後藤, 田中, 元岡

“P I Eの試作単一化プロセッサ〜マイクロプログラム”

28回 情報処理全国大会 6F-5, 1984.

小池，湯原，田中，元岡

“PIEの試作単一化プロセッサへシステム構成”

28回 情報処理全国大会 6F-6, 1984.

坂井，田中，元岡

“PIEのゴールフレーム分配網とコマンド通信網”

28回 情報処理全国大会 6F-7, 1984.

内山，平田，田中，元岡

“PIEの構造データ共有方式におけるガーベジコレクション”

28回 情報処理全国大会 6F-8, 1984.

平田，相田，後藤，田中，元岡

“PIEにおける構造データ処理方式の効率化～構造データ共有方式～”

28回 情報処理全国大会 6F-9, 1984.

濱中，丸山，相田，後藤，田中，元岡

“PIEの並列度評価”

28回 情報処理全国大会 6F-10, 1984.

相田，松原，田中，元岡

“PIEにおける集合および多重世界の実現方式”

29回 情報処理全国大会 2B-1, 1984.

濱中，北野，田中，元岡

“PIEのACの設計概容”

29回 情報処理全国大会 2B-2, 1984.

平田，田中，元岡

“PIEにおける構造化メモリの構成法”

29回 情報処理全国大会 2B-3, 1984.

相田，平田，丸山，田中，元岡

“高並列推論エンジンPIEの階層的構成法”

29回 情報処理全国大会 2B-4, 1984.

丸山，相田，田中，元岡

“PIEの階層的構成についてのシミュレーション”

29回 情報処理全国大会 2B-5, 1984.

小池，相田，田中，元岡

“PIEの試作UPの性能評価”

29回 情報処理全国大会 2B-6, 1984.

D. 学会研究会等

後藤，相田，田中，元岡，

“推論向き高並列計算機システムの基本アーキテクチャ”，

信学技報 EC 82-43, 1982.

後藤，相田，山崎，丸山，湯原，田中，元岡，

“高並列推論エンジンP I Eにおける並列処理の効率化手法について”，

信学技報 EC 83- 9, 1983.

湯原，相田，後藤，田中，元岡，

“高並列推論エンジンP I Eの単一化プロセッサと縮退アルゴリズム”，

信学技報 EC 83-30, 1983.

平田，相田，後藤，田中，元岡，

“高並列推論エンジンP I Eにおける構造データの効率的な処理方式について”，

信学技報 EC 83-38, 1983.

丸山，湯原，相田，後藤，田中，元岡，

“高並列推論エンジンP I E～並列度のシミュレーションとその評価～”，

信学技報 EC 83-39, 1983.

E. その他

元岡，田中，後藤，相田

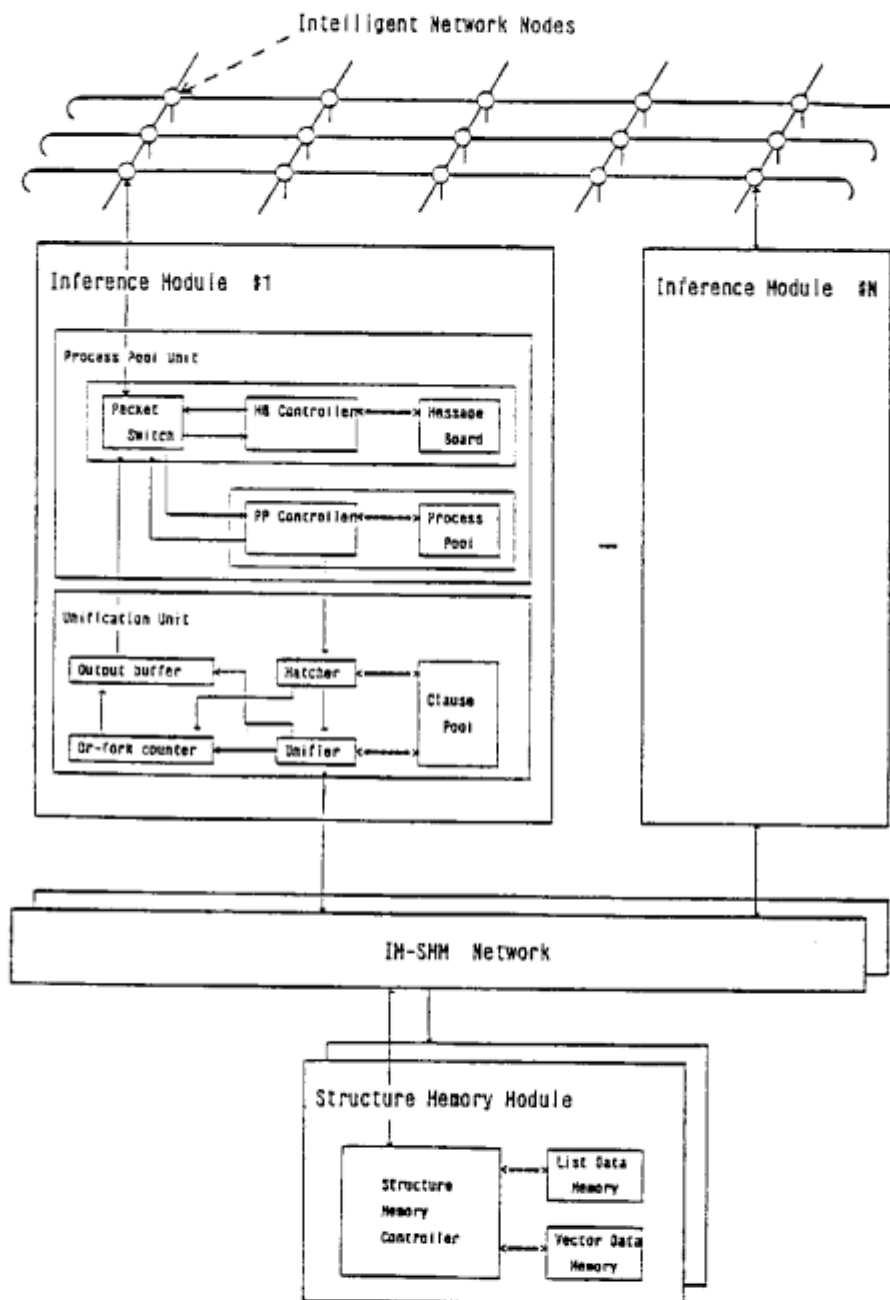
“推論マシンシステムに関する研究”，

東京大学工学部総合試験所 年報 Vol. 42, 1983.

(1/4) 並列マシン・アーキテクチャの研究 (昭和59年12月現在)	
並列処理 モデル名 or マシン名	P I M - R (リダクション方式並列推論マシン)
研究機関	I C O T (日立製作所)
主研究者	尾内 理紀夫、麻生 盛敏、清水 肇、益田 嘉直、松本 明 他
研究経過 (開始時期, 現状, 予定等)	
57年 6月 研究開始 59年春 PrologプログラムのためのSoftware Simulator (in Prolog) 59年秋 Concurrent Prolog プログラムのためのSoftware Simulator (in Prolog) 60年春 (予定) ・ 68000 8台からなるSimulation専用装置 ・ Software Simulator version 2 (in occam)) 完成	
並列処理モデルの主な特徴	
①PrologをOR並列に、Concurrent Prolog をAND 並列に実行する。 ②structure-copy採用。 ③Network 通過packet数とpacket幅の低減、copy量とそれに伴う処理量の低減のため、以下の各方式を採用している。 ・ only-reducible-goal copy方式 ・ reverse compaction方式 ・ 独特のプロセス構成 (ex. プロセス内へのProcess Life Block導入etc.)	

(2/4) 並列処理モデルの概略	
想定規模	64年春 100台規模
対象言語 及び 対象分野	Pure Prolog + 制限付き Concurrent Prolog 知識情報処理
目標性能	100台規模で1M~10MLIPS
(概略説明) 1. 並列処理の単位としてgoal(literal) を考える。各UU内Clause Pool は同一のclause群を格納し、1つのUU内で1つのgoalとunifiable なすべてのclauseとのunify を行う。 2. PPU からUUへはreducible なgoalのみを送る。(only reducible goal copy) PrologはAND 逐次、OR並列実行なので、goal列の最左goalがreducible、CPでは、並列AND operatorで区切られたgoals はAND forkし、分配ストラテジーに従い自IMあるいは異なるIM内Process Poolに分配されreducible となる。 3. OR fork 時のnew resolvent の各IMへの分配、AND fork時のgoals の各IMへの分配に際してはIntelligent Network Nodeが関与することが可能である。 4. reverse compactionによりProcess Poolのメモリ有効利用、copyに伴う処理量の低減をはかる。 5. プロセス内にProcess Life Blockを導入し、Network 通過packet個数の低減をはかる。子プロセスから親プロセスへはポインタが張られ、これにより情報が親に返される。 6. Process Pool内メモリに余裕のある間は、PLB 内return数check、PCB 内return数check、fork down packet処理を遅延し、解を求める処理を優先できる。 7. PCB 内reduction level をもとにした、ready queue 内でのPCB の入れかえ操作により探索ストラテジーの変更(IM内擬似depth-first、擬似breadth-first) をすることができる。	

(概略図)



(主な特徴)

- Network packet低減のためPPU とUUを別ModuleではなくIM内に収容。
- CPのチャネルのための分散化共有メモリMessage Board の導入。
- Intelligent Network Nodeの導入。
- groundにおちた長い構造体データの格納のためのSHM の導入。

(4/4)

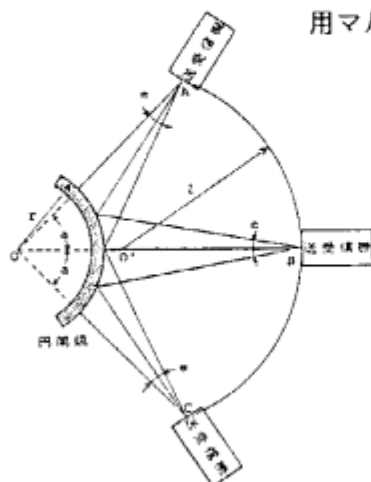
〈参考文献〉

- [1] 尾内他：“リダクション機構に基づくPrologマシンの一構成法”
情報処理第26回全国大会4N-3, 1983.3.
- [2] 尾内、麻生：“並列推論マシンにおけるGuard と入力annotationの制御機構”
情報処理第27回全国大会4P-5, 1983.10.
- [3] 尾内、益田、麻生：“Prologプログラムの静的解析について”
情報処理第28回全国大会4H-13, 1984.3.
- [4] 尾内、清水、益田、麻生：“逐次型Prologプログラムの解析”
Logic Programming Conference'84, Tokyo, 1984.3.
- [5] R.Onai, H.Shimizu, K.Masuda and H.Aso：“Analysis of Sequential Prolog Programs”
ICOT Technical Report TR-048, May 1984.
- [6] 尾内、麻生：“並列環境におけるConcurrent Prolog 実現法”
情報処理第29回全国大会7B-5, 1984.9.
- [7] 麻生、尾内：“並列PROLOGマシンのソフトウェアシミュレーションによる評価”
情報処理第29回全国大会、7B-6, 1984.9.
- [8] 尾内、麻生、清水、益田、松本：“並列推論マシンPIH-R のアーキテクチャとソフトウェア・シミュレーション”、ICOT Technical Report TR-077, 1985.1.
- [9] 尾内、麻生、清水、益田、松本：“並列推論マシンPIH-R のアーキテクチャ”
情報処理第30回全国大会、6C-6, 1985.3
- [10] 尾内、麻生、清水：“並列推論マシンPIH-R のソフトウェアシミュレーション”
情報処理第30回全国大会、6C-9, 1985.3
- [11] 清水、麻生、益田、尾内：“並列推論マシンPIH-R におけるプロセス内部表現”
情報処理第30回全国大会、6C-7, 1985.3
- [12] 益田、清水、麻生、尾内：“並列推論マシンPIH-R の構造体メモリの一構成法”
情報処理第30回全国大会、6C-5, 1985.3

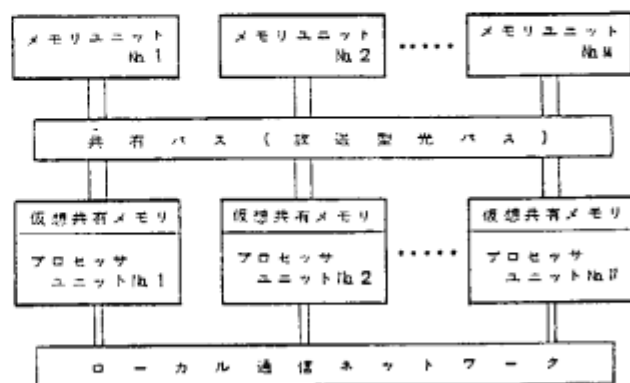
(1/4) 並列マシン・アーキテクチャの研究 (昭和59年12月現在)	
並列処理 モデル名 or マシン名	プロセスグラフモデル
研究機関	電子技術総合研究所
主研究者	梅山 伸二 他
研究経過（開始時期、現状、予定等）	
1981年 光バスの提案 Dialog.Hシステム（プロセスグラフモデルのインプリメントを想定しているマルチプロセスシステム）の開発開始 1983年 プロセスグラフモデルの提案 1984年 Dialog.Hプロトタイプシステム作成	
並列処理モデルの主な特徴	
1. 粗いレベル（リテラル）でのデータフロー制御によるOR並列処理モデル。 2. OR並列処理に必要な機能をいくつかの機能ユニットで実現しており、データはリテラルを単位（トークン）として、これら機能ユニット間を流れる。 3. 様々な探索ストラテジについても、特殊なトークンを付加したり、あるいはトークン自身にプライオリティを設定することにより実現できる。	

(2/4) 並列処理モデルの概略	
想定規模	～1000台
対象言語 及び 対象分野	Prolog
目標性能	?
(概略説明) <u>プロセスグラフモデルは</u> 1. リテラルを単位としたデータフローモデルに基づく。 2. Prologの各節に対応してプロセス（ある入力ゴールに対する解答を出力する）を考え、そのようなプロセスの動作をプロセスグラフと呼ぶ一種のデータフローグラフを用いて表現する。プロセスグラフは5種の機能ユニットから構成される。 3. データはリテラルを単位としてこのプロセス中を流れる。 4. PU内のインタプリタにより各機能ユニットの動作を実現する。 Prologプログラム自身はイニシャルトークンの形で、最初に各PU上に分散ロードされる。 5. 新しいタイプのトークンを導入し、またそれに伴って各機能ユニットの機能を拡張することにより、様々な実行制御を実現することができる。またトークン自身にプライオリティを与えることにより様々な探索ストラテジを実現できる。 <u>プロセスグラフモデルをDialogシステム上で実現する際には</u> 6. 各機能ユニットのインタプリタ、トークンなどは全てDialogシステムの仮想共有メモリ上に置かれる。仮想共有メモリの機能により、インタプリタ等常時使用される部分は、各PU上に常駐することになる。 7. トークンのヘッダー（タグ）はローカル通信ネットワークを通して、トークンのデータ自身は仮想共有メモリの機能を用いて通信される。	

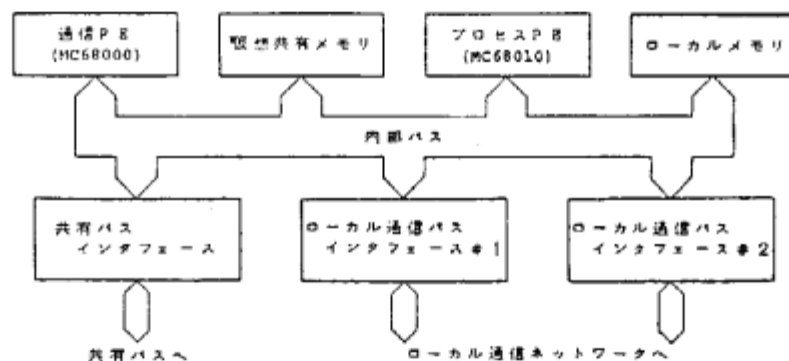
(概略図) Dialog.Hシステム プロセッサモデルのインプリメントを想定している光バス利用マルチプロセッサシステム



円筒鏡を用いた光バスの原理



Dialog.Hのシステム概念図



プロセッサユニットの構成

(主な特徴)

- ・レーザダイオードを用いた光バスを使用するマルチプロセッサシステム
光バス……・高速 ・スキューが少ない ・バス速度が接続ポート数に依存しない
・構造が簡単
- ・大規模な仮想共有メモリを持つ。
- ・仮想共有メモリのための円筒鏡方式の光共有バスと、ローカル通信用のホログラム方式の光バスを用いたローカル通信ネットワークの2種の通信手段を持つ。
- ・各プロセッサユニットは通信制御用、演算用の2個のプロセッサを持つ。

(4/4)

〈参考文献〉

プロセスグラフモデル関連

- [1] 梅山 “論理プログラムの並列実行について”
人工知能と対話技法研究会、1982
- [2] 梅山 “論理プログラムのためのOR並列モデル”
ロジックプログラミングコンファレンス'83、1983
- [3] S. Umeyama, K. Tamura. "A parallel Execution Model of Logic Programs" Proc. of the
10th Annual Int. Symp. on Computer Architecture, 1983.
- [4] 梅山 “論理プログラムのOR並列実行制御”
情報処理学会、第26回全国大会、1983.

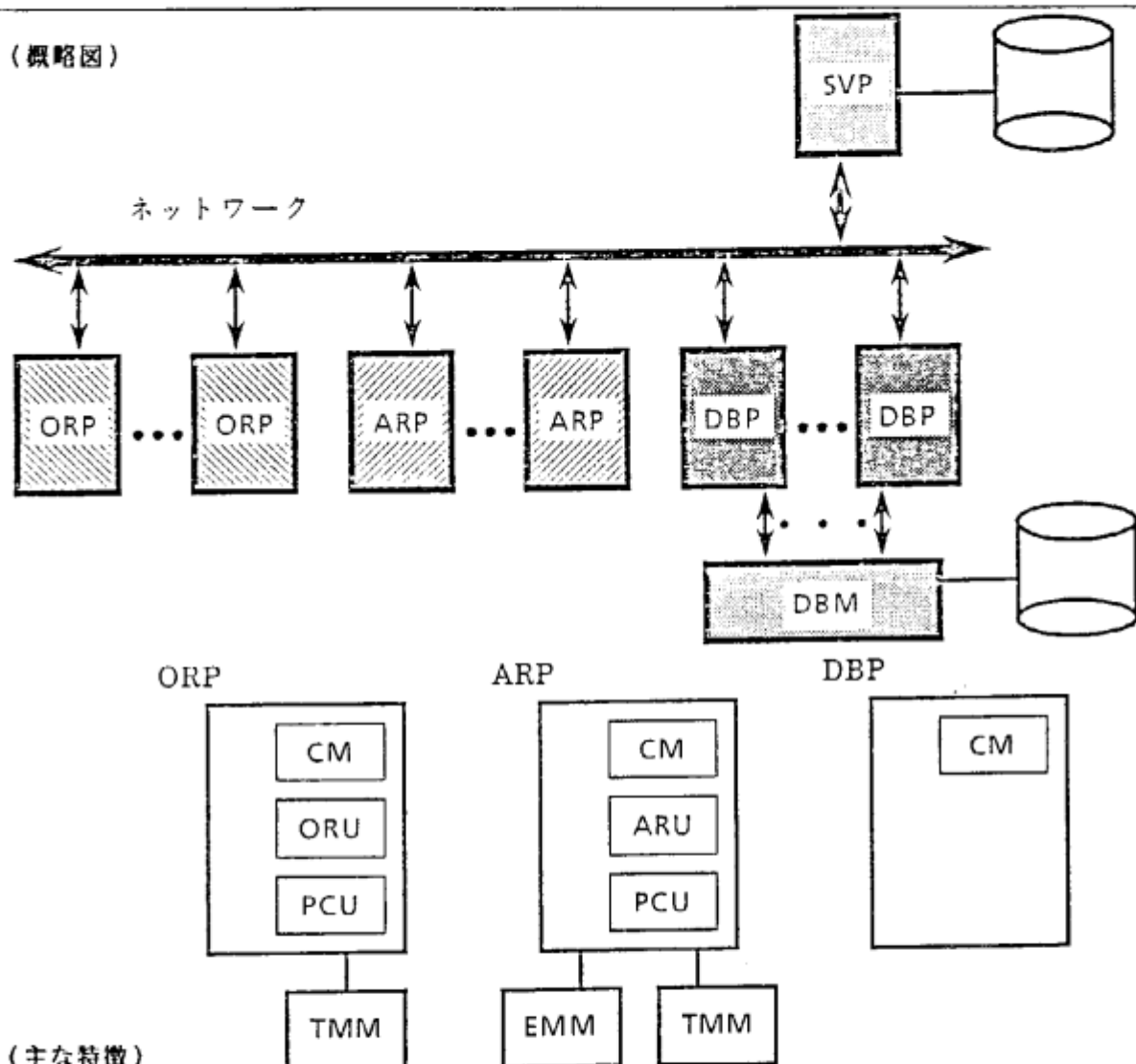
Dialog.Hシステム関連

- [1] 岡田他 “分散型アービタの一方法について”
情報処理学会第23回全国大会、1981.
- [2] 濱崎他 “5チャンネル100Mbit/s 光バスの試作について”
情報処理学会第25回全国大会、1982.
- [3] H. Tajima, et.al "A High Speed Optial Common Bus for a Multi-Processor System",
Trans. of the IECE of Japan, Vol. E66, No.1, 1983.
- [4] 田島他 “ホログラフィーを応用した高速光バスの考察”
情報処理学会第27回全国大会、1983.
- [5] Okada, et.al, "Dialog.H - a highly parallel processor based on optical bus"
Compcon Fall, 1983.
- [6] 濱崎他 "Dialog.H のプロトタイプシステム"
「アーキテクチャ・ワークショップ・インジャパン'84」シンポジウム、1984.

(1/4) 並列マシン・アーキテクチャの研究 (昭和 59 年 12 月現在)	
並列処理 モデル名 or マシン名	並列リダクション・モデルに基づく Prologマシン
研究機関	京都大学・工学部
主研究者	萩原 宏、富田真治、柴山 潔 その他
研究経過 (開始時期、現状、予定等)	
1982年秋	PL/Iによる Prolog処理システムの試作(文献[6])。
1983年春	並列処理方式による Prologマシン・アーキテクチャの検討開始。
1984年冬	ユニバーサル・ホスト 計算機 QA-2 による逐次型 Prologマシンの開発(文献[7]、約2KLIPS)。
1984年春	並列リダクション・モデルに基づく Prologマシンの第1版の提案(文献[1]~[4])。 第1版アーキテクチャの詳細化および改訂の開始。
1984年秋	QA-2による逐次型 Prologマシンの改良版の開発(文献[8]、約9KLIPS)。
現状	並列リダクション・モデルに基づく Prologマシンの第2版のプロトタイプ・アーキテクチャの設計中。マシンのシミュレータの作成中(文献[5])。
1985年末	プロトタイプ・ハードウェアの論理設計および製作。(予定)
並列処理モデルの主な特徴	
(1) Prologプログラムの実行過程を AND/ORプロセスグラフのリダクション操作とみなす実行モデルに基づく。 (2) 実行モデルとして、OR並列およびANDリテラル間のパイプライン処理(ストリーム並列)を行う。 (3) プロセス・ノードは各々その処理に向いた専用プロセッサ上で処理されるヘテロジニアス機能分散処理方式を採用。 (4) プロセス間通信はデマンド/イベントの授受による。	

(2/4)	並列処理モデルの概略
想定規模	100台程度(プロトタイプは数台)
対象言語 及び 対象分野	Pure Prolog (+ 並列型Prologの基本機能)
目標性能	1MLIPS程度(100台×10KLIPS) ◇◇◇SIM(数十～数百KLIPS)とPIM(1GLIPS以上)のつなぎ◇◇◇
(概略説明) (1) 入力されたプログラムはSVP(SuperVisor Processor)でテンプレート形式に変換して、各プロセッサにブロードキャスト転送する。 (2) 定義体の処理をO(Or) / A(Assertion)プロセスで、節本体の処理をS(Stream)プロセスで行う。O / S / Aの各プロセスはそれぞれ、対応するORP(Or Reduction Processor) / ARP(And Reduction Processor) / DBP(DataBase Processor)によって実行される。 (3) Oプロセスでは、定義体各部の頭部に対してユニフィケーションを試み、成功した場合で本体が有れば子プロセス(O / S)を起動し(invokeデマンド)、本体が無ければ親プロセスにsuccessイベントを送出する。 (4) Sプロセスでは、子プロセス(O)から得られた解メッセージ(success / failイベント)を基にして、引き続くゴールをパイプライン的に起動(先行評価)する。 (5) Aプロセスでは、インデキシングを用いることによりデータベース節(定義体がすべて事実より構成されている節)を高速に検索する。 (6) ハッシングを用いることにより、同一の定構造体(要素として変数を含まない)に対しては実体を一つしか作らない。 (7) success / failのイベントとそれに伴う処理を減らすために、イベントは不要なプロセスを飛び越して祖先のプロセスに直接送る。	

(概略図)



(主な特徴)

- ◎ ORP / ARPはそれぞれが、同一内容を複写したテンプレート・メモリ(TMM)を持つ。
- ◎ DBMはマルチポート構成で、複数のDBPにより共有される。
- ◎ 各プロセッサ(ORP / ARP / DBP)は、通信用メモリ(CM)を持つ。
- ◎ ARPはローカルな環境を保存するためのメモリ(EMM)に結合されている。
- ◎ プロセッサ間の結合方式については、検討中である。
- ◎ 組み込み述語は、リダクション過程に基づいて最適なプロセッサが選択され、それで処理される。
- ◎ ORP中のORUはユニフィケーションを主な仕事とする。
- ◎ ARP中のARUは子プロセスから返ってきた解(イベント)を基にして環境を書き換えたり、子(Or)プロセスの引数情報を作成したりする。
- ◎ PCUはプロセス識別子の割り当てなどのプロセス管理を行い、プロセス情報(子プロセスの数、イベントの送り先)を格納するメモリ(PM)を装備している。

(4/4)

< 参考文献 >

- [1] 柴山、村上、富田、萩原:「並列リダクション・モデルに基づくPrologマシン(1) -アーキテクチャ-」、情報処理学会第28回全国大会講演論文集、5F-6、1984年3月。
- [2] 村上、杉崎、柴山、富田、萩原:「並列リダクション・モデルに基づくPrologマシン(2) -実行メカニズム-」、情報処理学会第28回全国大会講演論文集、5F-7、1984年3月。
- [3] 八田、村上、柴山、富田、萩原:「並列リダクション・モデルに基づくPrologマシン(3) -プロセッサ構成-」、情報処理学会第28回全国大会講演論文集、5F-8、1984年3月。
- [4] 村上、柴山、富田、萩原:「並列リダクション・モデルに基づくPrologマシン(4) -メモリ構成-」、情報処理学会第28回全国大会講演論文集、5F-9、1984年3月。
- [5] 八田、柴山、萩原:「並列リダクション・モデルに基づくPrologマシンのハードウェア構成」、情報処理学会第30回全国大会講演論文集、6C-4、1985年3月。
- [6] 柴山、八田、富田、萩原:「PL/IによるPROLOG処理システムの試作」、情報処理学会第26回全国大会講演論文集、6D-2、1983年3月。
- [7] 小林、八田、柴山、富田、萩原:「ユニバーサル・ホスト計算機QA-2による逐次型Prologマシンのエミュレーション」、情報処理学会第28回全国大会講演論文集、6F-1、1984年3月。
- [8] 柴山、中田、富田、萩原:「ユニバーサル・ホスト計算機によるPrologの処理方式について」、情報処理学会第30回全国大会講演論文集、7C-4、1985年3月。

LIST OF CONTRIBUTORS

- | | |
|-------------------------------------|-------------------------|
| ①雨宮 真人、長谷川 隆三 他 | 電々公社 武蔵野通信研究所 |
| ②伊藤 徳義、清水 肇、
来住 晶介*、久野 英治* 他 | ICOT 第1研究室
(* 沖電気工業) |
| ③松田 秀雄、田村 直之、
金田 悠紀夫、前川 禎男 他 | 神戸大学 工学部 |
| ④相馬 行雄、増沢 秀穂 他 | 富士通株式会社 |
| ⑤元岡 達、田中 英彦 他 | 東京大学 工学部 |
| ⑥尾内 理紀夫、麻生 盛敏、
清水 肇、益田 嘉直、松本 明 他 | ICOT 第1研究室 |
| ⑦梅山伸二 他 | 電子技術総合研究所 |
| ⑧萩原 宏、富田 真治、柴山 潔 他 | 京都大学 工学部 |

編集担当 益田 嘉直
(ICOT第1研究室)