マルチPSIにおけるデバッグ機能・メンテナンス機能

杉野栄二†, 古市昌一‡, 稲村雄†, 瀧和男†
†：(財) 新世代コンピュータ技術開発機構
‡：三菱電機 (株) 情報電子研究所

1 はじめに

新世代コンピュータ技術開発機構(ICOT)は、並列ソフトウエアの研究開発環境として、並列推論マシン『マルチPSI』を開発した。

『マルチPSI』は、昭和61年度に試作したマルチPSI第1版(6台PE構成)に始まる一連の開発計画の成果であり([1]、[2]、[3]、[4])、今後開発される並列推論マシンPIM上で動作するソフトウエアの研究開発に用いられる([5])。

現在マルチPSIは、64台プロセッサで稼働しており、オペレーティングシステム PIMOSおよび、いくつかのアプリケーションプログラムが稼働している([6]、[7]、[8])。

本稿では、マルチPSIのメンテナンス機能および、これらファームウエア、ソフトウエアの開発支援のために用意された低レベルデバッグ機能について発表する。

2 マルチPSIのハードウエア構成

2.1 概要

マルチPSIは、逐次型推論マシンPSI-IIのCPUを要素プロセッサ(PE)として([9]、[10])、これらを高速ネットワークで2次元メッシュ状に接続したマシンである。PEは、1筐体あたり8台納められ、筐体単位で8台PEのシステムから最大64台PEのシステムにまで構成できる。

入出力機能は、フロントエンドプロセッサ(FEP)として接続されたPSI-IIにより提供される。FEPは、最大4台が接続可能であり、このうち1台がメンテナンス機能を有するマスタFEPとなる。マスタFEPとPEは、上記ネットワークの他にメンテナンス・バスと呼ぶ低速バスで接続されている。

2.2 PE

PEは、基本的にPSI-IIのCPUであるが、以下の停止割り込みに関するレジスタが追加された。

- 停止割り込みフラグ

  PEが停止割り込みを発行したことを示す。

- 停止割り込みフラグ集合レジスタ

  筐体内の8台のPEの停止割り込みフラグを集めたレジスタである。レジスタのビット情報で、筐体内のどのPEが停止割り込みを発行したか知ることができる。

2.3 FEP

FEP(フロントエンド・プロセッサ)は、PSI-IIにネットワークのインタフェース(FENWC：フロントエンド・ネットワーク制御装置)を付加したものであり、マスタFEPは、さらにメンテナンス・バスのインタフェース(MPC：メンテナンス・バス制御装置)を付加している。FENWC、MPCは、ともにPSI-IIの入出力装置として接続されている。

マスタFEP以外の3台の(スレーブ)FEPは、FENWCを経由してマルチPSIのネットワークに接続されており、メンテナンス・バスは接続されていない。スレーブFEPとマスタFEPは、PSI-IIの汎用ネットワークで接続されており、この汎用ネットワークは、スレーブFEPのFENWC初期化等の立ち上げ処理に利用される。

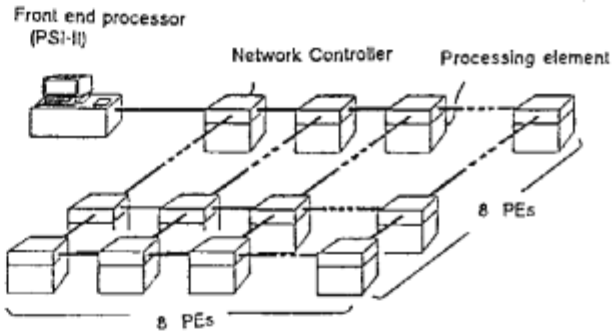

図1：マルチPSIの構成

2.4 接続ネットワーク機構

マルチPSIの接続ネットワークは、61年度に試作されたマルチPSI第1版に基づいて作られており([4]、[11])、自動ルーティング機能を持ち、隣接する4PEと双方向1バイト幅のチャネルでメッセージ通信する。なお、転送は200nsの同期転送である。

メッセージの先頭には、行き先PEアドレスがあり、各PEのネットワーク制御部(NWC)は行き先PEアドレスと受け取りチャネルから、転送チャネル選択用のテーブル(パス・テーブル)を引いて、転送チャネルを決めている。ネットワークは、

直接PE番号を指定する

- 物理PEアドレス

プロセッサが分布する論理的平面の、座標で指定する

- 論理PEアドレス

の二系統をサポートしているが([11])、現在は物理PEアドレスだけ使用している。

Debug and mentenance functions on the Multi-PSI.
by Eiji Sugino†, Masakazu Furuichi‡,
Yu Inamura†, and Kazuo Taki†
†: Institute for New Generation Computer Technology (ICOT)
‡: Information Systems & Electronics Development Lab.,
Mitsubishi Electric Corpolation

2.5 メンテナンス機構

メンテナンス・バスは、マスタFEPをマスタ、全てのPEをスレーブとしたシングル・マスタ/マルチ・スレーブのバスである。8ビットのデータ・アドレス共用線と若干の信号線から成り、複数PEへのブロードキャスト機能を持つ。

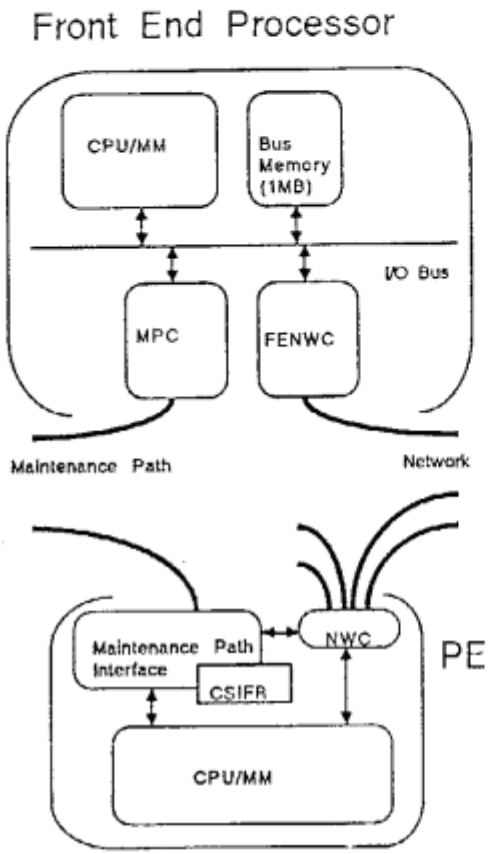

図2：マルチPSIのFEPとPE

メンテナンス・バスのPE側には、メンテナンス・バス・インタフェースとして、CSIFR(Console Interface Register)なるレジスタ群が接続されている。CSIFRには、

- PEのCPUの起動・停止を行うレジスタ、
- マイクロ命令を格納するレジスタ、
- PEの実行を、上で設定したマイクロ命令に切り替えるスイッチ、

などがある。これらによって、外部から任意のマイクロ命令を実行させることができ、PE上でマイクロプログラムが行える全ての操作が行える。

一方メンテナンス・バスのマスタFEP側には、MPC(メンテナンス・バス制御装置)が接続されている。マスタFEPのCPU、MPC、それにバス・メモリは、I/Oバスで接続されており、MPCは、バス・メモリに置かれたコマンド列を解釈、実行してCSIFRとバス・メモリの間のデータ転送を行う。

3 マルチPSIのソフトウエア

マルチPSIは、ユーザ言語として並列論理型言語KL1を採用している([12])。

KL1は、AND並列の論理型言語GHCをもとにして、以下のような機能拡張を行った言語であり([12])、KL1によってオペレーティングシステムPIMOSも記述されている。

- 実行制御機能「荘園」
- 負荷分散、優先順位指定機能「プラグマ」

KL1は、抽象機械語命令KL1-Bにコンパイルされ([13])、メンテナンス機能によりPEにロードされる。KL1-Bの各命令は、マイクロプログラムにより実現されており、マイクロプログラムもメンテナンス機能を用いて全PEにロードされる。

なお、現在KL1処理系は、一命令53ビットのマイクロ命令で16Kステップ、およそ0.1MB程度の量になっている。

4 FEP上のソフトウエア

FEPには、論理型言語ESPとオペレーティングシステムSIMPOSが提供されている。この上で、メンテナンス機能を提供するCSP(コンソールプロセッサ機能)ソフトウェア、および入出力インタフェースを提供するFEP(フロントエンド機能)ソフトウエアを開発した。

なおCSP・FEPソフトウエアのソースコードは、現在両方合わせて3MB以上である。

4.1 CSPソフトウエア

CSPソフトウエアは、ESPで記述され、SIMPOSの豊富な機能を利用してユーザインタフェースを実現している。

ESPは、オブジェクト指向言語であり、CSPソフトウエアの開発にあたっても PE、FEP、MPCなどをオブジェクトとしている。

(1) CSPコマンド

CSPソフトウエアの提供する機能は、CSPコマンドウインドウからのコマンド入力により起動される。

CSPコマンドは、

PE番号 @ コマンド[コマンド引数]

の形をしており、任意のPEに対する操作を行うことができる。ここで、先頭の'PE番号 @'を省略すると、デフォルトで指定してあるPEに対して実行される。デフォルト指定のPEは、複数設定することができ、これによって複数PEへの一括操作を容易にしている。

(2) 出力メッセージ

コマンドの結果表示は、CSPウインドウに出力される。どのPEに関する出力であるか示すために、出力メッセージには一般にPE番号を付けている。複数PEからの出力が混在して出力されるような場合には、出力先ウインドウをPEごとに別にすることもできる。

(3) コマンドファイル、ユーザ定義コマンド

一連のコマンド列を実行させるために、コマンド・ファイル機能も用意されており、直線的な実行ならば入れ子構造も可能である。

また、ESPの述語レベルで、CSPソフトウエアの機能を解放しており、ユーザがESPプログラムで作った機能をCSPのコマンドとして新たに追加できるようになっている。これにより、さらに複雑なコントロールや、機能拡張もできるようになっている。実際、この機能を利用してハードウエアの診断プログラムも開発された。

4.2 FEP ソフトウエア

FEP ソフトウエアは、マルチPSIにおけるI/O機能を提供するものである。FEP ソフトウエアは、並列オペレーティングシステムPIMOSの一部に位置付けられており([6]、[14])、ネットワークを介して、PEとFEP上のI/Oデバイス(ウインドウ、メニュー、ファイル等)の間でデータを転送する。

5 メンテナンス機能

CSP ソフトウエアは、メンテナンス機構を利用して以下のような機能を提供している。

5.1 立ち上げ、初期化

CSP ソフトウエアは、マスタFEP上のアプリケーションである。従ってマルチPSIを利用するためには、まずマスタFEPの電源投入をして、CSP ソフトウエアを立ち上げる必要がある。

マルチPSI本体の電源は、CSP ソフトウエアを利用してマスタFEPから投入/切断される。スレーブFEPは、メンテナンス系から独立しているため、個々に電源投入しなければならない。スレーブ上のFEP ソフトウエアについては、汎用ネットワークを利用して起動できるようにしている。

PEの初期化は、マスタFEPから次のような手順で行う。

- 初期化用のマイクロプログラムを全PEにブロードキャストし、
- PE上で同時実行させる。
- 全PE同じ初期化情報(内部で用いるテーブルのサイズ等)をブロードキャストする。
- PEごとに異なる初期化情報(PE番号、パス・テーブル)を個々にロードする。
- KL1処理系マイクロプログラム・コードを全PEにブロードキャストする。
- KL1プログラム(KL1-Bコード)をブートPEに(だけ)ロードする。
- スレーブFEPがある場合には、上記処理と同期して、スレーブFEPのネットワークの初期化を行う。

KL1プログラムは、ブートPEとして指定されたPEにしかロードしない。その他のPEについては、必要に応じてPE間でロードを行う。

次に、マルチPSIのCPUを起動するとともに、KL1の初期ゴールを転送する。

初期ゴールは、ブートPE上に初期プロセスを作ると同時に、FEP ソフトウエアと初期プロセスの間にストリームを張る。このストリームを利用して、KL1プログラムは、FEP ソフトウエアの機能を利用することができる。

CSP ソフトウエアは、初期ゴールの転送と同時にスレーブFEPへの起動メッセージを送る。スレーブFEPの起動メッセージは、ネットワークを通じて受け取られ、マスタFEPが初期ゴールで作った初期プロセスと、スレーブFEP上のFEP ソフトウエアの間のストリームを張る。

5.2 構成の変更、縮体運転

マルチPSIのPEは、8台単位で1筐体に格納されており、筐体単位で8台版システムから64台版システムまで構成できるようになっている。

また、最大4台のFEPは、方形に接続されたネットワーク外周の向かい合う二辺であれば、いずれのPEにも接続可能である。

現在は、64PE・4FEP(8×8)版、32PE・2FEP版(8×4)、16PE・1FEP版(4×4)、16PE・2FEP版などで運用されている。

このようにマルチPSIは、ハードウエア構成が柔軟であるので、PEがハードウエア障害を起こした際も、筐体を切り離して運転することが可能である。しかし、このような切り離しはユーザが簡単に行えるものではない。

そこで、マルチPSIでは、このようなハードウエア的な接続換えを行わない縮体運転を可能した。

例えば、図のような4×4の16台構成のマルチPSIでプロセッサAがハード障害を起こしたとする。この場合は、右端の1列を(ソフトウエア的に)外して、3×4の12台構成として運転することができる。ここで、1列外さなければならないのは、ネットワークのルーティングのためであり、メッセージがAを通ることがないようにするためである。

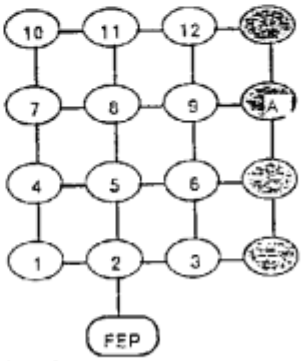

図3:縮体運転(A)

内側のプロセッサBに障害が起きた場合には、Bを中心とする十字方向のPEを外し3×3の9台構成の運用が可能である。但しこの場合、B以外のPEについては、ネットワークだけ動作させ、メッセージを素通りさせることが必要である。

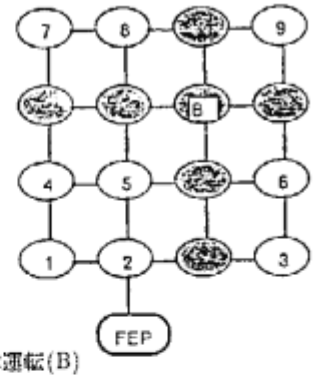

図4:縮体運転(B)

また、マルチPSIでは、上の図のように任意の配置でPE番号を割り当てることが可能である。

5.3 コントロール、エラー処理

PEが停止すると、MPCからマスタFEPへI/O割り込みが起こる。

割り込みを受けてCSP ソフトウエアは、前述の停止割り込みフラグをたよりに、停止しているPEの検索を行う。停止PEを確認して、割り込み原因に応じた操作を行う。

PEの停止には、エラーによる停止の他に、デバッグによる停止もある。デバッグ時では、1PEが停止することで実行の進みぐあいが著しくアンバランスするので、こ

れを抑えるために全PEを停止させている。全PEを停止させないモードも用意してあり、初期化処理に用いられる。

また、PEの停止/稼働状態は、マスタFEP上の『状態監視パネル』に表示される。図5の状態監視パネルは、16台構成のマルチPSIでのものである。ここで、白丸のPEが稼働しており、黒丸のPEsが停止していることが分かる。

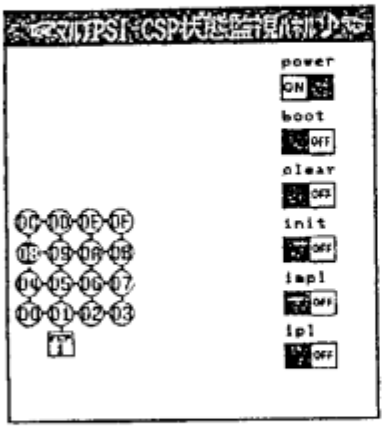

図5：状態監視パネル

6 デバッグ機能

マルチPSIのデバッグ機能は、

- マイクロプログラム
- 抽象機械命令KL1-B
- 並列論理型言語KL1

の3つの言語レベルに応じたものがある。

6.1 マイクロプログラムに対するデバッグ機能

マイクロプログラムについては、以下のようなものが用意されている。

- ステップ実行
- アドレス指定による、特定マイクロ命令での中断
- 直前まで実行されたマイクロ命令のログ(1KW程度)
- 逆アセンブラ
- タグニーモニック表示

  ファームウエアでは59種類のタグを使用している。

6.2 KL1-B に対するデバッグ機能

KL1-Bについては、以下に示すようにステップ実行に加えて、種々の条件による条件停止が用意されている。

- ステップ実行
- アドレスBreak

  特定アドレスのKL1-B命令で中断する。

- オペレーションコードBreak

  特定タイプのKL1-B命令で中断する。

- レジスタBreak

  指定レジスタの値が指定条件を満たした時に中断する。

- メモリBreak

  指定アドレスのメモリの値が指定条件を満たした時に中断する。

これらの設定は、メニュー選択により容易に行える。

7 KL1 に対するデバッグ機能

7.1 簡易入出力機能

通常KL1ソフトウエアの入出力は、ネットワーク経由でFEPソフトウエアを利用して行う。この他に、メンテナンスバスを利用したデバッグ用の入出力が用意されている。

これは、PE上にあらかじめ決められた領域を、CSPソフトウエアとのインタフェース領域として、PEと、CSPソフトウエアの間でデータ転送するものである。

例えばPEから出力する際には、インタフェース領域へ出力データと停止原因コードを書き込み停止する。停止割り込みにより、CSPソフトウエアは、停止PEと停止原因を調べ、インタフェース領域からデータを読み取ってCSPウインドウに表示する。

なお、簡易入出力機能として、次がサポートされている。

- ストリングの表示
- 任意のデータ(タグ部、データ部)入出力
- 文字コード入出力

7.2 トレーサ

KL1の実行では、ゴールはすべてAND関係にあり実行順序にあまり意味がない。連続して実行されるゴールの間には、一般に因果関係があるとは限らず、単にゴールをトレースしたのでは、不必要なゴールをトレースしてしまうことになる。そこで本トレーサでは、ゴールの子孫だけに注目して、実行を追えるようにした。

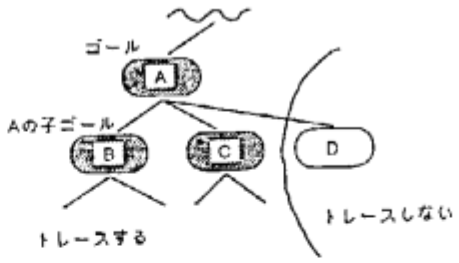

図6：ゴールの実行木

図6のようなゴールAの実行のうち、ゴールB、Cの実行を見たいとする。ゴールAは、あらかじめトレースを指定しているものとする。

図7のようにゴールスタックは一般に、トレースゴール(黒)、非トレースゴール(白)が混在している。ゴールスタックから取り出したゴールAがトレースゴールならば、トレーサが起動され実行過程を見ることができる。実行結果として、子ゴールB、C、Dが生成されたことがわかる。ここで、さらに実行を追いたい子ゴールB、Cについて、トレースを設定し、トレースの不要な子ゴールDについては、非トレースを設定してゴールスタックに入れる。

トレースの開始は、KL1プログラム中に記述することができる。また、以下に述べるスパイ機能、インスペクト機能によってもトレースを設定することができる。

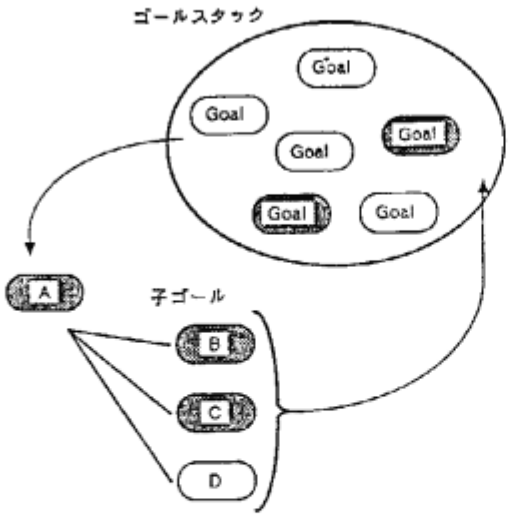

図7：トレースゴールの実行

7.3 スパイ機能

トレースしたいゴールを指定するために、スパイ機能が設けてある。スパイは、モジュール名、述語名、引数個数で指定することができる。

7.4 アトムの印字表現化

アトムは、その内部表現に印字名を持っていない。従って、デバッグにおいては、コンパイル時に作られたリンク情報をもとに、CSPソフトウエアが印字表現に変換している。

7.5 インスペクト機能

実行途中のPEの内部状態を調べるために、インスペクト機能が用意されている。インスペクタは、内部形式のデータを視覚化したり、データの探索をしたりする機能を持つ。

ゴールスタック・インスペクタは、優先度別にスタックされているゴールを検索することができ、内部にあるゴールに直接トレースをかける機能も持つ。

また、荘園インスペクタは、複数プロセッサに展開された荘園の木構造を視覚化する機能を持つ。

例として、荘園のインスペクタのメニューウインドウを示す。ウインドウで木構造に表示されている数字が、それぞれ荘園に対応する。それぞれの数字(先頭2文字)により、荘園がどの、PE上にあるかも判別できる。また、それぞれの数字をマウス選択することにより、その荘園の詳細な情報を表示させることもできる。

8 その他

その他、以下のようなデバッグ機能がある。

8.1 ネットワーク・メッセージ・モニタ

FEPソフトウエアが送/受信するメッセージを、モニタすることができる。メッセージは、意味が読み取れる形に変換されて表示される。

8.2 擬似並列処理系 – Pseudo マルチ PSI –

マルチPSIの開発に平行して、PSI-II上でマルチPSIをシミュレートするPseudoマルチPSIも開発した。Pseudoマルチ PSIでは、擬似PEがソフトウエアで用意されており、PEの実行をスケジューリングすることで、擬似並列処理を行っている。

処理系は、マルチPSIと同じくファームウエアで実現されており、マルチPSIの1PEの性能とほぼ同じ処理性能が出ぜる。ファームウエアは、PSI-II本来のものとKL1用のものと二重に持っており、擬似PEが動作する際に、KL1のファームウエアがオーバレイされる。

またPseudoマルチPSIは、擬似並列実行のために再現性があり、CSPソフトウエア、ファームウエアなど、大部分が実際のマルチPSIと同じであるため、KL1ソフトウエアのみならず、CSPソフトウエア、ファームウエアのデバッグ、テストにも用いられた。再現性があるため、マルチPSIに比べてデバッグが容易であり、初期のデバッグは、ほとんどPseudoマルチPSIで行うことができる。

ただし、一定のルールで擬似PEのスケジュールをすると、一定のタイミングでしか動作せず、タイミングによるバグは取れ難い。そこで、Pseudoマルチ PSIでは、擬似PEのランダムスケジュール・モードもサポートしており、タイミングを変えた実行も可能にしている。

9 運用

実際に運用してみると、立ち上げ時間は、(スレーブFEPなしの場合で)

- 16台版マルチPSIで約2分35秒
- 32台版マルチPSIで約3分16秒
- 64台版マルチPSIで約5分36秒

となっている。
内訳を見ると、

- CSPソフトウエアの初期化に20秒
- 電源投入に約15秒から20秒(16台あたり)
- 初期化に

  約1分55秒(16台版)
  約2分15秒(32台版)
  約4分10秒(64台版)

であった。

初期化の大部分は、ブロードキャストによる全PEへのデータのロードであり、表1のような性能が出ている。

初期化の際にPEごとに異なる初期化情報として、パス・テーブル情報などがある。PEごとに異なる情報は、ブロードキャストすることができず、現在はこれを個々にロードしている。

パス・テーブルのロード時間は、

約7秒(16台版)
約20秒(32台版)
約1分15秒(64台版)

であった。

パス・テーブルのロード時間は、PE台数が多くなるにつれて、立ち上げ時間に占める割合が大きくなり、64台では無視できなくなってきている。

スレーブFEPが接続された場合は、

- 電源投入に約10秒
- 初期化に約30秒

余分に時間がかかる。これは、全系同期クロックで動作させるための、スレーブFEP接続ネットワークのクロック切り替え、スレーブFEP接続ネットワークの初期化等のためであると考えられる。

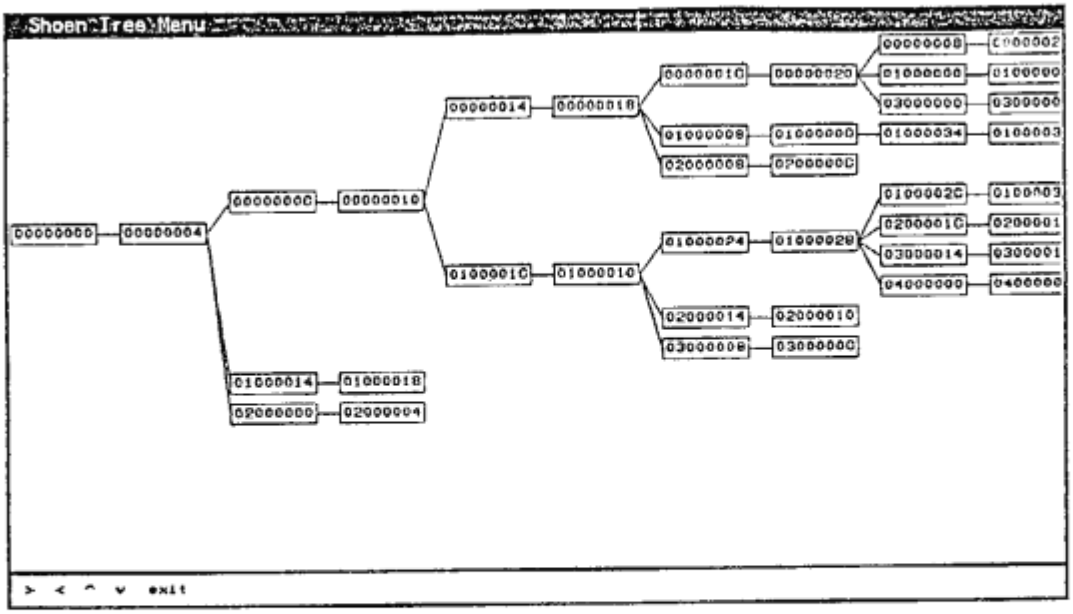

図8：並列インスペクタ・メニュー・ウインドウ

PE台数	性能	1台と比較した低下率
1	70KB/sec	− − −
8	63KB/sec	−10%
16	62KB/sec	−11%
24	59KB/sec	−16%
32	54KB/sec	−23%
40	51KB/sec	−27%
48	48KB/sec	−31%
56	46KB/sec	−34%
64	44KB/sec	−37%

表 1: ブロードキャスト・ローディングの性能

10 おわりに

このたび、並列推論マシン・マルチPSI上における、メンテナンス・デバッグ機能を開発した。現在、これらを用いて並列ソフトウエア、並列オペレーティングシステムPIMOSの開発が引き続き行われている。今後は、これらソフトウエアの評価のための測定機能の開発、および各機能の改良を進めて行く予定である。

11 謝辞

日頃ご指導を頂いております、ICOT第4研究室内田俊一室長ならびに第4研究室各位に感謝いたします。三菱電機(株)コンピュータ製作所の岩山洋明氏、他関係各位にも感謝いたします。

12 参考文献

[1] Taki, K. : "The Parallel Software Research and Development Tool : Multi-PSI System", France-Japan Artificial Intelligence and Computer Science Symposium 86, 1986

[2] 宮崎、瀧：『マルチPSIにおけるFlat GHCの実現方式』、Logic Programming Conference '86 No. 7.2、1986

[3] 木村、瀧、内田：『マルチPSIシステムとその接続方式』、第33回情報処理学会全国大会論文集 7B-1、1986

[4] 益田 他：『マルチPSIにおける接続ネットワークの試作と評価』、情報処理学会論文誌、第29巻第10号、1988

[5] 服部 他：『並列推論マシンPIM/pのアーキテクチャ』、JSPP'89、1989

[6] 佐藤 他：『PIMOSの資源管理方式』、JSPP'89、1989

[7] 斉崎 他：『マルチPSIにおける並列構文解析プログラムPAXの実現および評価』、JSPP'89、1989

[8] 神 他：『マルチPSIにおける並列版詰め碁プログラムの実現と評価』、JSPP'89、1989

[9] 中島(克) 他：『マルチPSI要素プロセッサPSI-IIのアーキテクチャ』、第33回情報処理学会全国大会論文集 7B-3、1986

[10] Nakashima, H. et al. : "Hardware Architecture of the Sequential Inference Machine : PSI-II", In Proceedings of 4th Symposium on Logic Programming、1987

[11] Takeda, Y. et al. : "A Load Balancing Mechanism for Large Scale Multiprocessor Systems and its Implementation", In Proceedings of International Conference on Fifth Generation Computer Systems 1988

[12] 宮崎：『並列論理型言語KL1の実現方式と並列OSの記述』、電子情報通信学会誌、'88.8 Vol.J71-D No.8、1988

[13] 木村他：『KL1のクローズインデキシング方式』、JSPP'89、1989

[14] Chikayama, T. et al. : "Overview of the Parallel Inference Machine Operating System (PIMOS)", In Proceedings (ICOT Research Topics) of International Conference on Fifth Generation Computer Systems 1988