

マルチ PSI における接続ネットワークの試作と評価†

益田 嘉直^{††} 石塚 裕一^{†††} 岩山 洋明^{††}
 瀧 和男^{††††} 杉野 栄二^{††††}

第五世代コンピュータ・プロジェクトの並列推論マシンの研究開発は、主にマシン・アーキテクチャの研究に重点が置かれてきたが、その過程で論理型プログラミングの並列実行に関するソフトウェア面での重要性が確認されるようになってきている。そのため、ICOT では並列ソフトウェアの研究開発環境を構築するために、逐次型推論マシン PSI を複数台接続したマルチ PSI システムの開発を進めている。並列ソフトウェアの研究課題としては、並列言語、並列 OS、負荷分散方式などがあり、これらはいずれも並列推論マシン PIM を研究開発する上で重要なものである。マルチ PSI システムは現行 PSI を専用のネットワークにより格子状に6~8台接続した第1版と、現在開発中である小形化 PSI を64台接続した第2版の二段階に分けて開発を進めている。本稿ではマルチ PSI 第1版の接続ネットワークの構成およびその評価について述べる。第1版では並列論理型言語 KL1 の処理系には PSI のシステム記述言語である ESP で記述したものを実装し、その上に並列 OS である PIMOS の核部分を試作中である。

1. ま え が き

述語論理をベースとした並列論理型言語を効率よく実行する並列推論マシンの研究が、第五世代コンピュータ・プロジェクトを契機として各所で行われるようになってきている¹⁾。これまでの並列推論マシンの研究では、主にマシン・アーキテクチャの研究に重点が置かれてきたが、その過程で論理型プログラミングの並列実行に関するソフトウェア面での研究の重要性が認識されるようになってきた²⁾。そのため、ICOT では並列ソフトウェアの開発環境を構築するために逐次型推論マシン PSI を複数台接続したマルチ PSI システムを第1版と第2版の二段階に分けて開発を進めている³⁾。

マルチ PSI 第1版は並列ソフトウェア研究の早期着手に重点を置いて短期間で完成可能な構成をとるとともに、より大規模で高性能なマルチ PSI 第2版を設計するための実験評価用システムとしての役割も持ち、既に稼働を開始している。具体的には開発済みの PSI マシンを6台から8台使用し、接続ネットワーク・ハードウェア部分を新規に開発して小規模のマル

チプロセッサ・システムを早期に完成させたものであり、並列論理型言語 KL1 (Kernel Language Version 1) の処理系には PSI のシステム記述言語である ESP で記述したものを実装し、その上に並列オペレーティングシステム PIMOS の核部分を試作中である^{4),5)}。また、簡単な評価用応用プログラムを実行して並列推論ソフトウェアに関する基本的な実験評価も実施中である。

マルチ PSI 第2版では、CPU 台数の拡張と性能向上を図るため、要素プロセッサとなる PSI 自体の改良小型化を行っており、接続ネットワークにも改良を加え、最終的には要素プロセッサを最大で64台程度接続できるシステムを目指している^{6),7)}。高い性能を得るため KL1 処理系はすべてファームウェアで記述したものを実装する。その上で PIMOS の改良拡張や大規模応用プログラムの実験を行い詳細な評価を加えていく予定である。

本稿では、ICOT において稼働を開始しているマルチ PSI 第1版の接続ネットワークの構成およびその評価について述べるが、マルチ PSI 第1版の構成は図1に示すように、現行 PSI マシンを入出力装置も含めて6台接続したシステムである。また、図2のようなシステム階層構成をとり、図中の下線部分が新たに製作または製作中の部分である。

2. 接続ネットワークの構成

2.1 試作の目的

今回のマルチ PSI 第1版接続ネットワーク試作の目的を以下に列挙する。

† Network Hardware and Evaluation of the Multi-PSI System by KANA E MASUDA (Computer Works, Mitsubishi Electric Corporation), HIROKAZU ISHIZUKA (Information Systems & Electronics Development Laboratory, Mitsubishi Electric Corporation), HIROAKI IWAYAMA (Computer Works, Mitsubishi Electric Corporation), KAZUO TAKI and EIJI SUGINO (Fourth Research Laboratory, Institute for New Generation Computer Technology).

†† 三菱電機(株)コンピュータ製作所

††† 三菱電機(株)情報電子研究所

†††† (財)新世代コンピュータ技術開発機構第4研究室

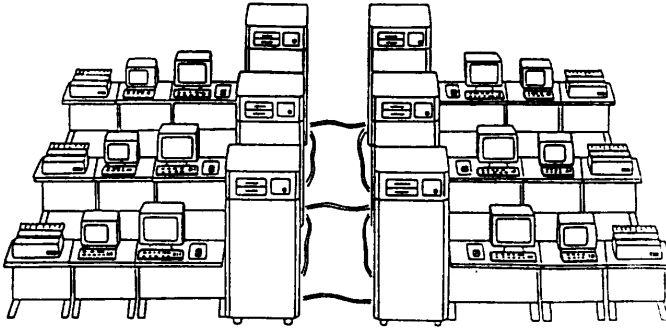


図1 マルチ PSI 第1版の構成図
Fig. 1 Multi-PSI-V1 system.

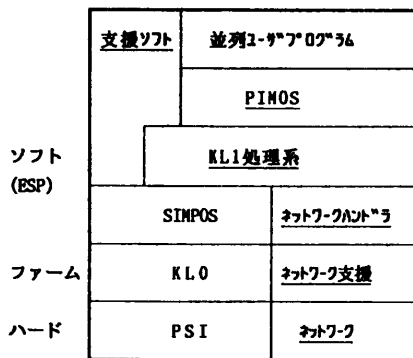


図2 マルチ PSI 第1版のシステム階層構成
Fig. 2 Configuration of the Multi-PSI-V1.

- (1) PSI をマルチ化するに当たり接続ネットワークの機能、制御方式を明確にすること。
- (2) 接続ネットワークのハードウェア構成を明確化し、第2版に向けて LSI 化の検討を行うこと。
- (3) 早期に並列ソフトウェアの実験、評価環境を実現すること。

2.2 基本構成

接続ネットワーク・ハードウェアは PSI マシンを格子状に接続するためのハードウェアで、CPU の内部バスのオプション・スロットに実装され、隣接の4台のプロセッサと接続するための4本のケーブルが引き出される。これらの引出し口をそれぞれチャンネルと呼び、データはパリティビットを含め10ビット単位で並列転送されチャンネルの一方当りの転送能力は約500KB/secである⁸⁾。各チャンネルは、それぞれ10ビットの送信データ線と受信データ線を独立に持っており、チャンネルの入り口には受信インタフェース、送信インタフェースがあり受信、送信の転送制御を行っている。また、チャンネル信号線上のデータはハンドシェイク制御で10ビットごとに非同期並列転送される。接続ネットワーク・ハードウェアとCPUのイ

ンタフェース部分には4KBの容量を持つ受信信用、送信用FIFO型のバッファ(CPU ReadバッファおよびCPU Writeバッファ)を設置しており、また各チャンネルの送出部分にも256BのFIFO型のバッファ・メモリを設けた。これらのFIFO型バッファのサイズは性能評価その他のために変更可能としている。エラー検出機能としては、パリティエラー、パケットエラーのチェック機能を備えており、動作状態でエラーが検出されると、接続ネットワーク・ハードウェアは停止

状態になる。また、モード制御レジスタやシーケンス制御レジスタ等各種レジスタが搭載されているが、大部分のレジスタはファームウェアによりアクセスが可能である。

マルチ PSI 第1版のネットワークの形状は格子型とし、プロセッサ間通信は共有メモリは置かず、メッセージ・パケットの交換による方式を採った。パケット・データは可変長であり、各パケットの先頭には行先CPU番号の情報を持つ。10ビットのデータのうち最上位ビット(ビット9)はパリティで、ビット8はパケットの先頭/終了か、途中データかを示し、先頭/終了データの場合にはビット7で先頭/終了の区別を行う(図3)。ネットワークの構成は図4に示すように4本のチャンネル(CH0~CH3)とCPUインタフェースの5対5の接続形態となっており、行先CPU番号から送出先チャンネルを選択する操作、いわゆるルーティングの制御はバス制御メモリを用いて行われる。すなわち、各々のチャンネルから到着したパケッ

表1 接続ネットワーク・ハードウェアの主な仕様
Table 1 Basic specification of the Network Hardware.

項目	内容
接続方式	格子型(メッシュ構造)
交換方式	メッセージ・パケット交換
データ転送方式	ハンドシェイク制御 (10ビットごとの非同期並列転送)
転送能力	約500KBytes/sec(各チャンネル)
最大ノード 交換転送能力	2.5MBytes/sec
データ転送幅	10ビット(パリティビットを含む)
パケット形式	可変長(先頭に行先CPU番号の情報)
エラー検出機能	パリティエラー、パケットエラー (強制エラー機能有)

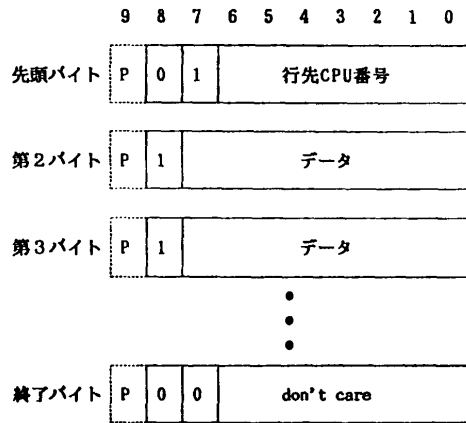
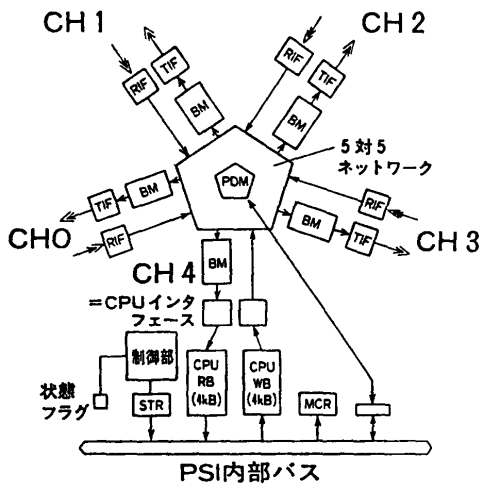


図3 パケット形式
Fig. 3 Packet format.



- PDM: パス制御メモリ
- TIF: 送信インタフェース
- RIF: 受信インタフェース
- BM: バッファメモリ
- CPUWB: CPU ライト・バッファ
- CPURB: CPU リード・バッファ
- MCR: モード制御レジスタ
- STR: ステータス・レジスタ

図4 接続ネットワーク・ハードウェアの構成
Fig. 4 Block diagram of the Network Hardware.

トの行先 CPU 番号を、CPU 番号と送出先チャンネルの対応表を記憶しているパス制御メモリを用いて認識し、それぞれ対応した送出先へ再送出する。この時、パケットがそれぞれ異なる送出先へ再送出され、行先チャンネルが競合しない場合は転送は同時並列的実行され、各接続ネットワーク・ハードウェア（ノードに対応）当りの最大ノード交換転送能力は 2.5 MB/sec となる。この方式では横方向優先などの簡単なルーチング戦略を固定的に用いるだけでデッドロックが回避さ

れる。接続ネットワーク・ハードウェアの各チャンネルの送出部分にはネットワーク上を流れるデータのよどみの解消や、転送効率の向上のために FIFO 型のバッファ・メモリが設けられているが、実際には1つのメモリを5つに分け、各々のチャンネル用として時分割で用いている。

2.3 動作

接続ネットワークのパス制御メモリ、シーケンス制御等の動作を以下に示す。

(1) パス制御メモリ

パス制御メモリは、パケット・データの先頭バイトの〈6:0〉で表される行先 CPU 番号を入力とし、そのデータをどの方向へ送出するかを決める3ビットのデータ（東西南北と自分の5種類）を出力するメモリである。通常、パケット・データが入ってくると、パス制御メモリを引いて転送先チャンネルを求める。

(2) シーケンス制御

各々のチャンネルから入力されたデータがどのような手順で転送されるかを以下に示す。

- ① 接続ネットワーク・ハードウェアは動作状態に入ると CH0 から CH3 および CH4 (自 CPU) とのインタフェースの順にデータ到着の有無をチェックし、次に CH0 から CH4 の順にバッファ・メモリ (BM) のデータが出力可能かどうかチェックする。
- ② データが到着し、かつそれが最初のデータの時はパス制御メモリを引いて転送先チャンネルを求める。そして、そのチャンネルがビジーでなければ、チャンネルの使用権を得てデータ到着チャンネルからの転送経路を開設し、Busy フラグをセットする。既にビジーであればビジーが解けるまで待たされる。
- ③ バッファ・メモリへのデータの書込みおよびバッファ・メモリからのデータ送出は 10 ビット単位で時分割で行われ、送出先ネットワークがビジーでないか、または行先が自 CPU で CPU Read バッファが full でない場合データ送出は積極的に行われる。

このほか、CPU へパケットを読み込む際には、完全なパケットのみを CPU Read バッファから高速に読み込む必要があるため、未完パケットカウンタやパケット到着フラグを設けている。

(3) パケットの送受信

本接続ネットワークには通常使用モードとハード

ウェアの初期化, エラー発生時の処理, 保守時に使われる非通常使用モードがある. 通常使用モードにおけるパケットの送出, 受取りの処理はネットワーク支援ファームウェアによって行い, 非通常使用モードにおけるレジスタ・アクセスの処理は標準の組込述語 (PSI システムの中に標準に用意されている述語) で行っている. 通常使用モードでネットワーク支援ファームウェアによって行われるパケットの送出, 受取りの処理について述べる.

① パケットの送出処理

CPU からパケットをネットワークへ送出するためには CPU 内でパケットの形に成形されたデータをファームウェアにより CPU Write バッファに書き込むことにより行う. この時, ファームウェアにより CPU Write バッファの空き領域を調べてから実際の書き込み処理を行う. また, パケットの先頭/終了マークバイトの付加などもファームウェアで行う.

② パケットの受取り処理

ネットワークから CPU にパケットを受け取るには CPU Read バッファに転送されて来たパケットをファームウェアで CPU 内に読み込むことにより行う. この時, CPU Read バッファ内には複数のパケットが入っていることが一般的であり, ネットワークから転送途中のパケット (未完パケットと呼ぶ) が入る可能性もある. したがって, CPU にパケットを読み込む際には, まず CPU Read バッファに1個以上の完全なパケットが到着したことを知らせるパケット到着フラグをファームウェアで調べる. そして, このフラグがセットされている時のみ実際の読み込み処理を行う. ここで未完パケットカウンタの内容を読み込むことにより, CPU Read バッファ内の完全なパケットのバイト数を計算する処理もファームウェアで行う.

上記の処理を実現するために, 各 PSI マシン上にはネットワーク支援ファームウェアが組込述語追加の形で実装されている.

3. 接続ネットワークの評価

3.1 目的

マルチ PSI 第1版は KL1 処理系と接続ネットワークから構成されている. さらに接続ネットワークは図5のような階層構造をしている.

(i) KL1 処理系

これは KL1 プログラムの実行制御および実行を行う言語処理プログラムであり, ESP で記述されて

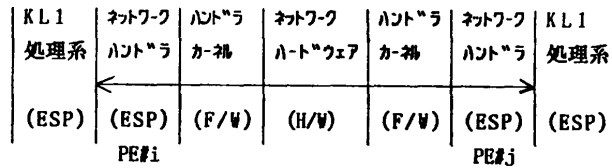


図5 ネットワークの階層構造

Fig. 5 Layered structure of the Network.

いる.

(ii) ネットワーク・ハンドラ

これは処理系から渡されたメッセージ (通信オーダー) をパケット (バイト・ストリング) に成形しハンドラ・カーネルに渡し, またその逆の変換処理を行ったり, さらに KL1 に関するゴール送受の管理やアドレス変換処理を行うものである. これも ESP で記述されている.

(iii) ハンドラ・カーネル (支援ファーム)

これはネットワーク・ハンドラから渡されたパケット (バイト・ストリング) をバイト・シリアルなデータに変換し, またその逆のデータ変換を行うものである. このハンドラ・カーネルは KL0 の組込述語としてファームウェアで実現されている.

今回の接続ネットワークの評価にあたっては次の2項目を主目的として各種データを測定, 評価を行った⁹⁾.

(1) PE 内処理性能と PE 間通信処理性能の比 (1リダクション当りの処理時間比)

1リダクションを PE 単体で実行するのに要する時間と2台の PE で実行するのに要する時間を求めて, 接続ネットワークの性能をマクロに論ずるものである.

(2) 接続ネットワークの各階層別動特性の分析

これは, (1)で測定したデータをさらに細かく分析して, ネットワークの各階層における通信コスト (時間) の比較を定量的に行おうとするものである.

3.2 測定方法

(1) 測定に当たりの考え方

第一の目的として掲げた「PE 内処理性能と PE 間通信処理性能」と言った場合, まず何をもってして比較するかを決めておく必要がある. そこで今回は2台版マルチ PSI を用いて1リダクションに要する実行時間をその指標とした. すなわち, 自 PE 内で1個のゴールをリダクションするのに要する時間と, 同じゴールを送出してその受信先の PE でそのゴールをリダクションして応答が戻るまでの実行時間の比を

```
test3_la(0):-true | true.
test3_la(N):-N 1:=N-1 | wait 3_la(R, N1),
              call (bench_mark @t 3_la, R, _).
wait3_la (success, N) :- true | test 3_la(N).
t3_la :- true | alloc(5) @@tt 3a(x).
tt 3a(X) :- true | X=atom.
```

図 6 ベンチマーク・プログラム (unify テスト) の例
Fig. 6 An example of the bench-mark program (unify test).

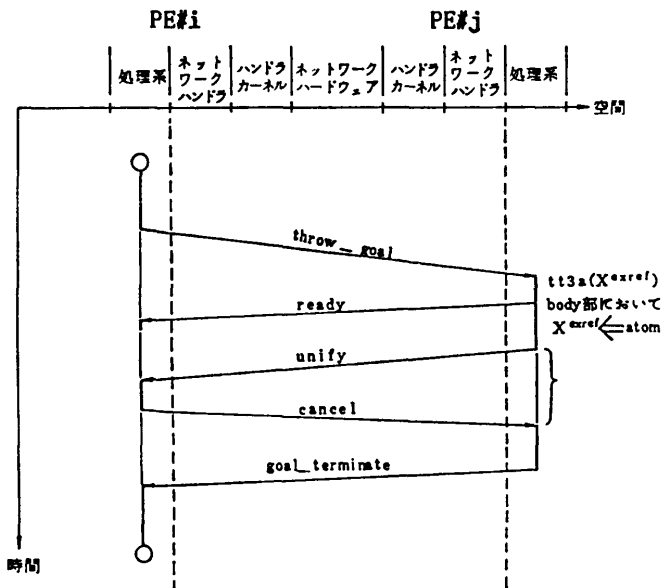


図 7 メッセージ (通信オーダ) 交換の概念図
Fig. 7 An example of the message exchange.

「PE 内処理性能と PE 間通信処理性能」の比と定義した。この比の値は KL1 プログラムを記述する場合に PE 内にどの程度処理をとり込み、PE 間通信を減少させる必要があるかを検討する場合に重要な値となる。測定のために図 6 のようなベンチマーク・プログラムを数種類用意し、実行時間を測定した。すなわち、throw_goal, unify, read といった PE 間通信オーダ (全部で 13 種類有り) のコストを引数タイプおよび引数個数を変化させ計測した。図 7 は図 6 に示したベンチマーク・プログラムを実行させた時のメッセージ交換の様相である (正確には tt3a(x) というゴールを 1 回投げた時の通信記録)。測定に当ってはそれぞれ 100 回同じプログラムをループして実行し、その平均値を算出することで OS のサポートするタイマの誤差を軽減した。

(2) 各階層における通信コスト

(i) ハンドラの通信コスト (Ha)

これは(1)で述べた考え方に沿って実測した結果 (L) からハンドラ・カーネル (F) およびネットワー

ク・ハード (H) のコストを差し引いて求めることができる。

$$Ha = L - F - H$$

ただし F および H に関しては次の (ii), (iii) を参照。

(ii) ハンドラ・カーネルの通信コスト (F)

1 バイトずつデータをインタフェース用レジスタに書いたり、読み込んだりする mpsi_write-buffer, mpsi_read-buffer というような組込述語の実行時間。

(iii) ネットワーク・ハードウェアの通信コスト (H)

ハードウェアの通信コストはファームウェア (F) との境界があいまいで、環境により変動するが、ハードウェアの動作解析により求まる。

3.3 測定結果と考察

第一の目的である「PE 内処理性能と PE 間通信処理性能」の比について考察する。まず、PE 間通信オーダの中から特に重要な throw_goal, unify, read を選択し、それらの通信コストを調べる。そのために引数タイプ (atom, integer, undefined variable, structure, list) および引数個数 (1, 2, 3, 4, 5) の異なるベンチマーク・プログラムを実行し得られた結果が図 8 である。この図の縦軸の実行時間は 100 回ループの 1 回当たりの平均実行時間であり、ゴー

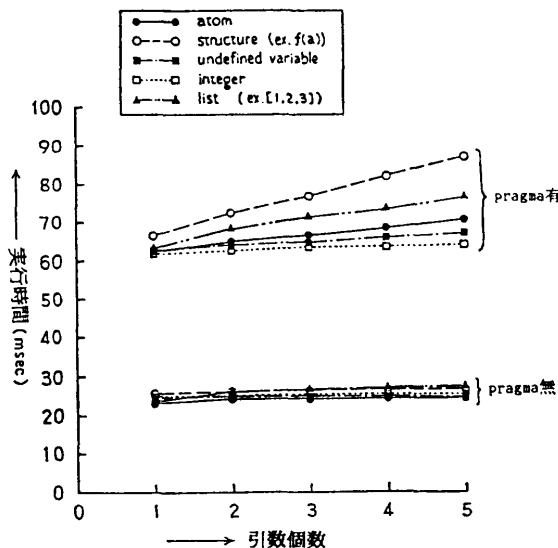


図 8 引数個数と実行時間の関係 (throw_goal)
Fig. 8 Execution time for predicate argument numbers (throw_goal).

ルを他 PE に投げる場合の実行時間 (pragma 有) と、そうではなく同じゴールを自 PE 内で処理する場合 (pragma 無) の実行時間が示されている。この場合 pragma 有の実行時間 A と、pragma 無の実行時間 B の差がおおむね PE 間通信コストと言える。ここで、リダクションには最適化の適用されるものとそうでないものの2種類あることに注意する必要がある。このベンチマーク・プログラムにおいて、pragma 無の場合は最適化が適用されるが、pragma 有の場合は適用されない (図9のリダクション R_{op} とリダクション R_n)。したがって、通信コストを算出する場合、単に実行時間の差を求めるだけではなく、このリダクション時間の差も考慮しなければならない。よって実際には、次式により通信コスト L を算出した。

$$L = \text{pragma 有の実行時間 } A - \text{pragma 無の実行時間 } B - (\text{リダクション時間 } R_n - \text{リダクション時間 } R_{op})$$

ただし最適化の適用されるリダクション時間 R_{op} (1.12 msec) と適用されないリダクション時間 R_n (2.25 msec) は、別のベンチマーク・プログラムにより測定したものを使用した。

さらに、「PE 間通信処理 対 PE 内処理」の処理時間比は以下の式より求められる。

$$\text{処理時間比} = (L + \text{リダクション時間 } R_n) / (\text{リダクション時間 } R_n)$$

引数タイプ、および引数個数を変えて測定した pragma 有、無の実行時間 (A, B) よりこの処理時間比を求めると表2の1リダクション当りの処理時間比 R のようになる。

この表から明らかなように、条件によって処理時間比 R はばらつきがあるが、おおよそ20対1程度であり、自 PE 内でゴールをリダクションするのに要する

時間と、同じゴールを他 PE でリダクションして応答が戻るまでの時間との間には約 20 倍の開きがある。

さて第二の目的である各階層での通信コストについて考察する。各階層における通信コストに関しては、throw_goal, unify, read 等のパケットのサイズが引数のタイプにより固定しているので、ファームウェア (ハンドラ・カーネル) およびハードウェアの通信コストの算出が可能であり、その結果ハンドラの通信コストも算出できる (表2)。まずファームウェアのコストであるが、これは GEVC カウンタによる測定の結果、次に示すような実験結果を得ることができた。

mpsi-write-buffer という組込述語の場合 L バイトのパケット・データの処理に

$$S = \underbrace{24}_{\text{オーバーヘッド}} + L \times \underbrace{5}_{\substack{1 \text{ データ処理} \\ \text{に必要なステップ数}}} + \underbrace{\text{INT}\{(L-1)/4\} \times 2}_{\text{補正項}}$$

ただし L: パケットサイズ, INT: 整数化の関数
ステップ必要であり、同様に mpsi-read-buffer という組込述語では

$$S = \underbrace{26}_{\text{オーバーヘッド}} + \text{INT}(L/4) \times \underbrace{32}_{\substack{4 \text{ データ処理} \\ \text{に必要なステップ数}}} + \underbrace{\alpha}_{\text{補正項}}$$

$$\text{ただし} \begin{cases} \alpha = 0 & (L \bmod 4 = 0) \\ \alpha = 12 & (L \bmod 4 = 1) \\ \alpha = 19 & (L \bmod 4 = 2) \\ \alpha = 26 & (L \bmod 4 = 3) \end{cases}$$

ステップ必要である。

その他の組込述語では

mpsi-check-read-buffer	11 step
mpsi-check-write-buffer	17 step
mpsi-sense-packet-arrival	2 step
mpsi-no-errors	2 step

となり、これを使って計算した結果が表2の F である。

次にハードウェア部分の動作時間を考察する。図10はマルチ PSI 第1版の2台接続時のネットワーク・ハードウェアの概念図と、その2台版におけるデータ処理に必要な最小の時間を求めるために描かれたタイミングチャートである。第1版のネットワーク・ハードウェアは5つのチャンネルを有し、その1つ1つをポーリングしてデータを送受信する方式をとっている (ただし、チャンネル4は CPU とのインタフェース)。第1版のハードウェアの通信コストを考える場合注意すべき点は次の3つである。

- (i) 送り側のポーリングコスト
- (ii) 受け側のハンドシェイクコスト
- (iii) 受け側のポーリングコスト

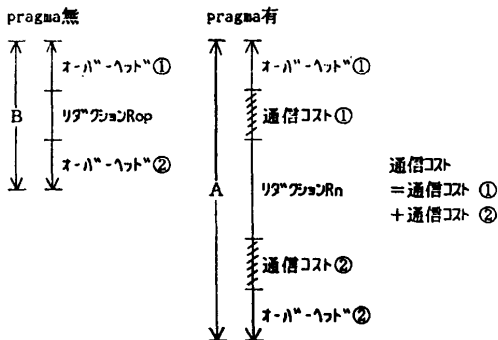


図9 pragma 有、無による実行時間の相違
Fig. 9 Execution time with pragma and without pragma.

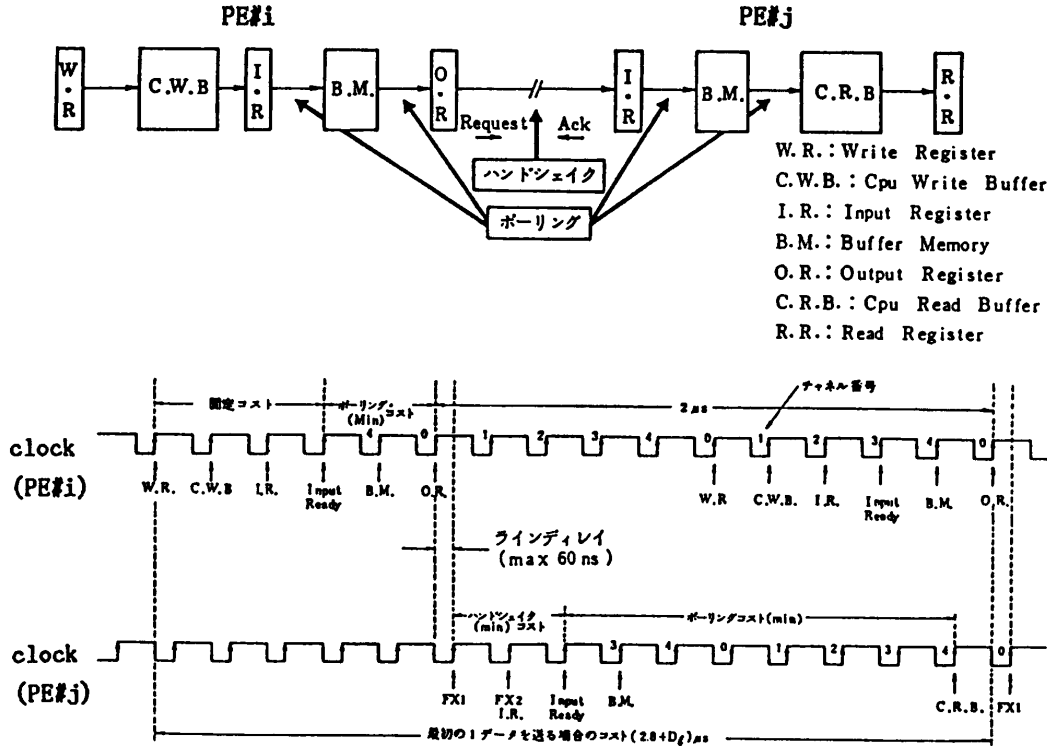
表2 測定結果一覧
Table 2 Results of execution time with pragma and without pragma. (単位 msec)

測定項目 (通信オーダー)	引数		測定結果		通信コスト ($Ha+F+H$) L	F/W のコスト F	H/W のコスト H	Handler のコスト Ha	1リダクション 当りの処理 時間比 R
	個数	タイプ	**A	B					
throw_goal (含 ready, goal_terminates)	1	atom	62.72	23.90	37.69	0.23	0.07	37.39	17.8
		undef	62.61	24.36	**37.12	0.23	0.07	36.82	17.5
		integer	61.96	24.76	36.07	0.22	0.07	35.78	17.0
		structure	66.62	25.41	40.08	0.24	0.07	39.77	18.8
		*list(1)	63.10	23.40	38.57	0.24	0.07	38.26	18.1
		list(2)	64.33	24.11	39.09	0.24	0.07	38.78	18.4
		list(5)	66.80	25.39	40.28	0.26	0.08	39.94	18.9
	2	atom	64.90	24.17	39.60	0.24	0.08	39.28	18.6
		undef	63.61	24.60	37.88	0.25	0.08	37.55	17.8
		integer	62.51	25.11	36.27	0.23	0.07	35.97	17.1
		structure	72.42	26.06	45.23	0.26	0.08	44.89	21.1
		list(1)	68.49	26.44	40.92	0.26	0.08	40.58	19.2
	3	atom	66.62	24.11	41.38	0.26	0.08	41.04	19.4
		undef	64.78	24.57	39.08	0.27	0.08	38.73	18.4
		integer	63.27	25.09	37.05	0.23	0.07	36.75	17.5
		structure	76.92	26.24	49.55	0.28	0.09	49.18	23.0
		list(1)	71.53	26.66	43.74	0.28	0.09	43.37	20.4
	4	atom	68.64	24.19	43.32	0.27	0.09	42.96	20.3
		undef	66.12	24.68	40.31	0.28	0.09	39.94	18.9
		integer	63.42	25.16	37.13	0.24	0.07	36.82	17.5
structure		82.01	26.54	54.34	0.30	0.10	53.94	25.2	
list(1)		73.83	27.01	45.69	0.30	0.10	45.29	21.3	
5	atom	70.62	24.13	45.36	0.29	0.09	44.98	21.2	
	undef	67.11	24.69	41.29	0.30	0.09	40.90	19.4	
	integer	64.05	25.13	37.79	0.24	0.08	37.47	17.8	
	structure	87.13	26.73	59.27	0.33	0.10	58.84	27.3	
	list(1)	76.70	27.18	48.39	0.33	0.10	47.96	22.5	
**unify (含 cancel)	1	atom	87.99	28.34	22.89	0.14	0.04	22.71	—
		undef	88.48	28.78	22.94	0.14	0.04	22.76	
		integer	88.99	29.21	23.02	0.13	0.03	22.86	
		structure	89.46	29.66	23.22	0.15	0.04	23.03	
		list(1)	87.18	27.53	22.89	0.15	0.04	22.70	
**read (含 read_answer)	1	atom	101.44	70.45	30.99	0.12	0.04	30.83	—
		undef	—	—	—	—	—	—	
		integer	100.31	71.10	29.21	0.11	0.03	29.07	
		structure	106.13	72.03	34.10	0.13	0.04	33.93	
		list(1)	101.11	69.15	31.96	0.13	0.04	31.79	

注) A: pragma 有
B: pragma 無
L: $A-B+1.12-2.25$
F: ファームウェアのコスト(計算値)
H: ハードウェアのコスト
(最小と最大の中間の値で計算)

$Ha: Ha=L-F-H$
 $R: (L+2.25)/2.25$
*1: list(1), (2), (5)は要素数が
1, 2, 5 であり, データタイプ
は integer
*2: 標準偏差は1データにつき
0.42 msec

*4: unify に関しては
 $L=A-B-(*)3$
*5: read に関しては A, B 両方と
も pragma 有,
 $L=A-B$ となるようにプログ
ラムを工夫.



◎ 2データ目以後 2μs 必要だから
 n データあれば $2.8 + D_l + 2(n-1) = (2n + 0.8 + D_l) \mu s$
 ◎ 1 clock = 0.2 μs
 図 10 2台接続時の通信のタイミングチャート (コスト最小の場合)
 Fig. 10 Timing chart for connected adjacent two PEs.

これらのコストはすべて変動する可能性を持っている。したがってハードウェアの通信コストは上・下限の極値を持ちマシン環境によってその間を揺れ動く。
 図 10 により解析した結果、2台版におけるハードウェア全体のコストは

- (a) 通信コスト最小の場合 (n データ)
 $C_{min} = (2n + 0.8 + D_l) \mu s$
- (b) 通信コスト最大の場合 (n データ)
 $C_{max} = (2n + 5.8 + D_l) \mu s$
- (c) 通信コストの平均値 (n データ)
 $C_{av} = (2n + 3.3 + D_l) \mu s$

今回表 2 に掲げたハードウェアのコストはこの (c) 式に沿って計算したものである (ただし、 D_l : ラインディレイ)。

今回の評価で測定を行ったデータの誤差は単体でおよそ 0.3 msec もあり、これとファームのコストならびにハードのコストを比較するとこれらは誤差範囲であることが言える。このことから明らかなように、マルチ PSI 第 1 版のネットワークまわりでは、ハンドラのコストがファームウェアやハードウェアのコス

トと比べ桁はずれに大きく、これらの比はほぼ 500:3:1 の割合である。

最後にマルチ PSI 第 2 版に言及する。表 2 の最上段のケースを例にとり、仮に処理系およびハンドラが 100 倍、ネットワーク・ハードウェアが 5 倍高速化されると、処理時間比は

$$\begin{aligned} \text{処理時間比} &= (\text{通信コスト} + 1 \text{リダクションのコスト}) / (1 \text{リダクションのコスト}) \\ &\text{で与えられていたから、これは次式にとってかわる。} \\ \text{処理時間比} &= (\text{ハンドラのコスト} / 100 + \text{ファームウェアのコスト} + \text{ハードウェアのコスト} / 5 + 1 \text{リダクションのコスト} / 100) / (1 \text{リダクションのコスト} / 100) \end{aligned}$$

これを計算すると 17.8 倍が 28 倍になる。すなわち第 2 版では「PE 間通信処理 対 PE 内処理」の処理時間比は第 1 版と比べ多少増加すると予想され、このことは第 2 版において PE 間通信量をそれだけ減少させる必要があることを意味する。

4. む す び

第五世代コンピュータ・プロジェクトの並列推論マシンの研究開発においては、常にその並列ソフトウェア開発と一体となって研究開発が進むべきものであるという認識が高まってきている。

現在 ICOT においてはマルチ PSI 第 1 版には ESP で記述した KL 1 処理系を実装し、並列プログラムの動特性の計測評価を行っている。今後も評価を継続しその結果をもとにマルチ PSI 第 2 版のネットワークならびにシステムの開発を進める予定である。

謝辞 本研究開発は、第五世代コンピュータ・プロジェクトの一環として実施している。ご指導ご鞭撻をいただいた ICOT 第 4 研究室内田俊一室長ならびに関係各位に深謝する。

参 考 文 献

- 1) 田中(英): 並列推論マシン, 情報処理学会研究会資料, Vol. CA-57, No. 3, p. 8 (1985).
- 2) 後藤, 杉江, 服部, 伊藤, 内田: 並列推論マシン PIM—中期構想—, 第 33 回情報処理学会全国大会論文集, 3B-5 (1986).
- 3) Taki, K.: The Parallel Software Research and Development Tool: Multi-PSI System, *France-Japan Artificial Intelligence and Computer Science Symposium 86*, pp. 365-381 (1986).
- 4) Ueda, K.: Guarded Horn Clauses, *Logic Programming Conference '85*, No. 9.3, pp. 225-236 (1986).
- 5) 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, pp. 83-92 (1986).
- 6) 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).
- 7) 中島(克), 瀧, 中島(浩), 京, 江原, 山本, 横田: 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).
- 8) 益田, 瀧, 木村: マルチ PSI のネットワーク・ハードウェア構成, 第 33 回情報処理学会全国大会論文集, 7B-2 (1986).
- 9) 益田, 岩山, 石塚, 末原, 瀧: マルチ PSI における接続ネットワークの評価, 第 34 回情報処理学会全国大会論文集, 4P-7 (1987).

(昭和 62 年 8 月 19 日受付)

(昭和 63 年 7 月 15 日採録)



益田 嘉直 (正会員)

昭和 22 年生。昭和 45 年早稲田大学理工学部電気工学科卒業。同年三菱電機(株)入社。昭和 51 年南カリフォルニア大学大学院電気工学科修士課程修了(MSEE)。以来、汎用計算機の開発、並列処理、推論マシンの研究開発に従事。この間、昭和 57 年～60 年(財)新世代コンピュータ技術開発機構(ICOT)に出向。現在、コンピュータ製作所 EWS 部主幹。本会 ARC 研究連絡会委員。IEEE 会員。



石塚 裕一 (正会員)

昭和 37 年生。昭和 61 年東京理科大学物理学科卒業。同年三菱電機(株)入社。情報電子研究所において、並列推論計算機の研究・開発に従事。



岩山 洋明 (正会員)

昭和 36 年生。昭和 59 年東京工業大学工学部情報工学科卒業。昭和 61 年同大学院修士課程修了。同年三菱電機(株)入社。コンピュータ製作所において、並列推論計算機のハードウェア開発に従事。



瀧 和男 (正会員)

昭和 27 年生。昭和 51 年神戸大学工学部電子工学科卒業。昭和 54 年同大学院修士課程システム工学修了。工学博士。同年(株)日立製作所入社。同社大みか工場にて制御用計算機システムの設計に従事。昭和 57 年(財)新世代コンピュータ技術開発機構に出向。以来逐次型および並列型推論マシンの研究開発に従事。並列マシンのアーキテクチャ、並列プログラミングなどに興味を持つ。電子情報通信学会、IEEE 各会員。



杉野 栄二 (正会員)

昭和 35 年生。昭和 58 年東京都立大学理学部数学科卒業。同年(株)富士通ソーシャル・サイエンス・ラボラトリ入社。Prolog コンパイラ等の開発に従事。昭和 61 年(財)新世代コンピュータ技術開発機構に出向。現在同機構研究所において、並列推論マシン、並列ソフトウェアの研究・開発に従事。