計算機アーキテクチャ　66－8
（１９８７．７．１１）

マルチＰＳＩにおける接続ネットワークの構成とその評価

益田嘉直、岩山洋明、石塚裕一、末原義人
（三菱電機株式会社）
瀧和男、杉野栄二
（ＩＣＯＴ）

現在、我々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及びその評価について述べる。第１版では並列推論言語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。

Network hardware and its evaluation of the Multi-PSI system

Kanae MASUDA,Hiroaki IWAYAMA,Hirokazu ISHIZUKA,Yoshihito SUEHARA
Mitsubishi Electric Corporation
325 Kamimachiya,Kamakura,Kanagawa 247,Japan

Kazuo TAKI,Eiji SUGINO
Institute for New Generation Computer Technology
Mita Kokusai Bldg. 21F, 1-¼-28 Mita,Minato-ku,Tokyo 108,Japan

We are currently developing the Multi-PSI system which supports a real parallel execution environment for parallel software research and development. The parallel software research themes are such issues as languages, operating system, program development system, basic research like load balancing and algorithms, and application programs. All of these are important for constructing the Parallel Inference Machine (PIM) system.

The Multi-PSI contains several PSI machine CPUs (6 for Multi-PSI-V1 and 64 for V2) and a dedicated lattice network.

This paper describes the network hardware and its evaluation of Multi-PSI-V1. The parallel logic programming language KL1 is based on GHC and implemented in ESP,the sequential language of the PSI machine. The basic part of the system is currently being tested.

1．はじめに

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研究の重要性が確認される様になって来ている〔1〕。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２段階に分けて開発を進めている〔2〕,〔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，既に稼動を開始している。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，並列推論言語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行く予定である。

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及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。

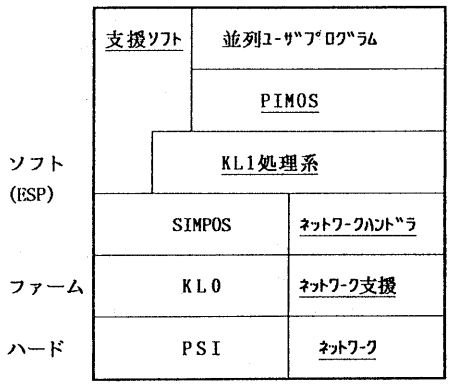


図２．マルチPSI第１版のシステム階層構成

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。

(1) PSIをマルチ化するに当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，マルチPSI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。

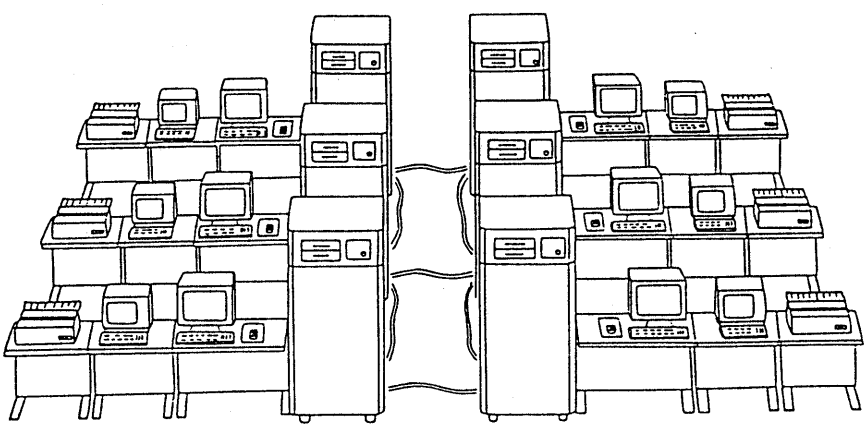


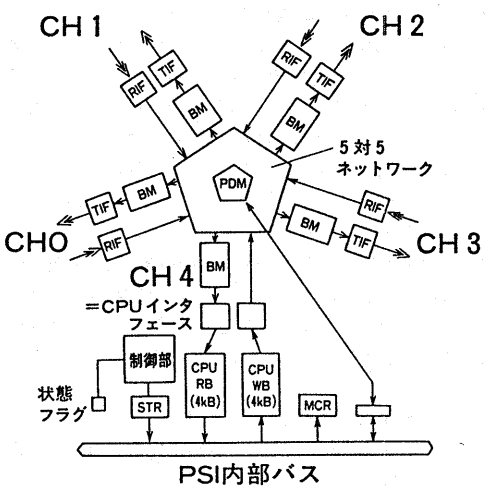
図１．マルチPSI第１版の構成図

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本のケーブルが引出される。これらの引出し口をそれぞれチャネルと呼び，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ネルの一方向あたりの転送能力は約500KB/secである〔8〕。

各チャネルは，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，チャネルの出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，チャネル信号線上のデータはハンドシェーク制御で10ビット毎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の容量を持つ送信用，受信用FIFO型のバッファ（CPU Readバッファ及びCPU Writeバッファ）を設置しており，また各チャ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。



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3．接続ネットワーク・ハードウェアの構成

表1．接続ネットワーク・ハードウェアの主な仕様

項　目	内　容
接続方式	格子型(ﾒｯｼｭ構造)
交換方式	ﾒｯｾｰｼﾞ・ﾊﾟｹｯﾄ交換
ﾃﾞｰﾀ転送方式	ﾊﾝﾄﾞｼｪｰｸ制御 (10ﾋﾞｯﾄ 毎の非同期並列転送)
転送能力	約500KBytes/sec(各ﾁｬﾈﾙ)
最大ﾉｰﾄﾞ交換転送能力	2.5MBytes/sec
ﾃﾞｰﾀ転送幅	10ﾋﾞｯﾄ(ﾊﾟﾘﾃｨﾋﾞｯﾄを含む)
ﾊﾟｹｯﾄ形式	可変長（先頭に行先CPU番号の情報）
ｴﾗｰ検出機能	ﾊﾟﾘﾃｨｴﾗｰ,ﾊﾟｹｯﾄｴﾗｰ (強制ｴﾗｰ機能有)

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4）。

ネットワークの構成は図3に示すように4本のチャネル(CHO～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ネルを選択する操作，いわゆるルーチングの制御はパス制御メモリを用いて行われる。すなわち，各々のチャネルから到着したパケットの行先CPU番号を，CPU番号と送出先チャ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2.5MB/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)で表わされる行先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ネルを求める。このパス制御メモリの出力はシーケンス制御部への入力となり，終了バイトが入力されるまで転送先チャネルはこのパケット・データの転送用に占有される。

(2) シーケンス制御

各々のチャ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及び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ネルを求める。そして，そのチャネルがビジーでなければ，チャネルの使用権を得てデータ到着チャネルからの転送経路を開設し，Busyフラグをセットする。すで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及びバッファ・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。

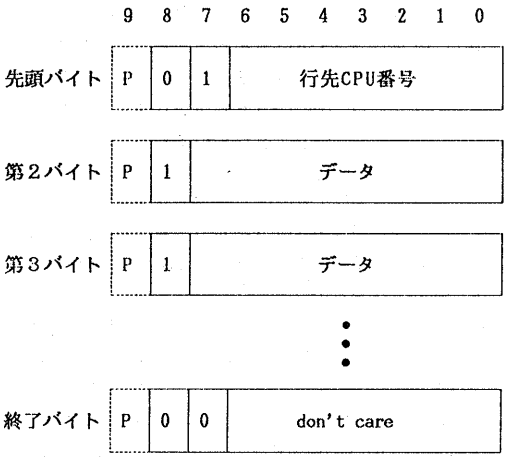


図４．パケット形式

この他，CPUへパケットを読込む際には，完全なパケットのみをCPU Readバッファから高速に読込む必要があるので，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。従って，CPUにパケットを読込む際には，まずCPU Readバッファに１ケ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。

(4) エラー処理

本接続ネットワーク・ハードウェアでは，エラー検出機能として，パリティエラー，パケットエラーのチェックを行っている。動作状態で，パリティエラー，パケットエラーが検出されると，検出された次のサイクルでハードウェアが停止状態となるとともに，モード制御レジスタのエラービットが１となる。ファームウェア側ではパケットの送受の際にこのエラービットをテストし，１を検出するとエラーレジスタを読み出して詳細情報を得，ソフトウェア側でエラーに対する処理を行う。

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。更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及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）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，更にKL1に関するゴール送受の管理やアドレス変換処理を行なうものである。これもESPで記述されている。

(iii) ハンドラ・カーネル（ネットワーク支援ファーム）

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なうものである。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，評価を行なった〔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更に細かく分析して，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なおうとするものである。

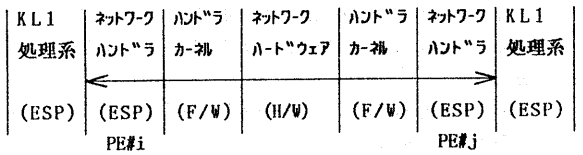


図５．ネットワークの階層構造

3.2　測定方法

(1) 測定にあたっての考え方

第一の目的として揚げた「PE内処理性能とPE間通信処理性能」と言った場合，まず何を持ってして比較するの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。即ち，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「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じ込め，PE間通信を減少させる必要性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の様なベンチマーク・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及び引数個数を変化させ計測した。測定に当たってはそれぞれ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。

```
test3_1a(0) :- true | true.
test3_1a(N) :- N1 := N-1 | wait3_1a(R,N1),
                call(bench_mark@t3_1a,R,_).
wait3_1a(success,N) :- true | test3_1a(N).
t3_1a :- true | alloc(5)@@tt3a(x).
tt3a(X) :- true | X=atom.
```

図６．ベンチマーク・プログラム（unifyテスト）の例

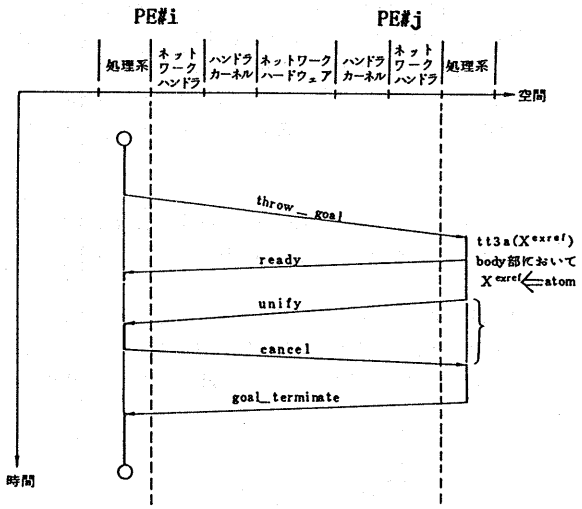


図７．メッセージ（通信オーダ）交換の概念図

(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)及びネットワーク・ハードウェア(H)のコストを差し引いて求めることができる。

Ha=L-F-H

但しF及びHに関しては次の(ii),(iii)を参照のこと。

（ii）ハンドラ・カーネルの通信コスト(F)

１バイトづ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。GEVCと呼ばれるカウンタにより測定。

（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)及び引数個数 (1,2,3,4,5)の異るベンチマーク・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回当りの平均実行時間であり，ゴールを他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RopとリダクションRn）。従って，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＝pragma有の実行時間A－pragma無の実行時間B
　－（リダクション時間Rn－リダクション時間Rop）

但し最適化の適用されるリダクション時間Rop(1.12msec)と適用されないリダクション時間Rn(2.25msec)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。

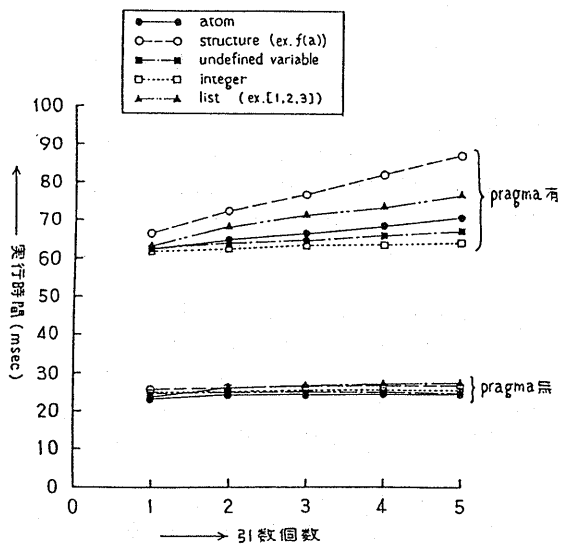


図８．引数個数と実行時間の関係（ throw_goal ）

更に，「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。

処理時間比＝（L＋リダクション時間Rn）
　　／（リダクション時間Rn）

引数タイプ，及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の様になる。

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。

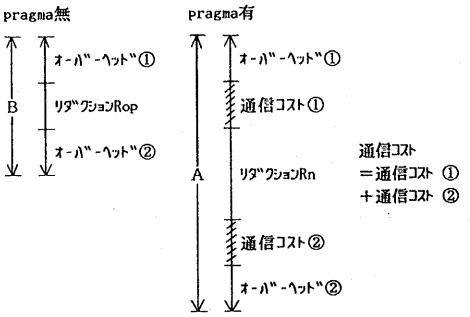


図９．pragma有，無による実行時間の相違

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）及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

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

但し　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

$$S = 26 + INT(L/4) \times 32 + \alpha$$

オーバーヘッド　　4データ処理に必要なステップ数　　補正項

但し { α=0 (L mod4=0)　α=12 (L mod4=1)
　　 { α=19 (L mod4=2)　α=26 (L mod4=3)

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であり，図11.はその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つのチャネルを有し，その一つ一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（但し，チャネル4はCPUとのインターフェース）。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)　送り側ポーリングコスト
(ii)　受け側ハンドシェイクのコスト
(iii)　受け側のポーリングコスト

これらのコストは全て変動する可能性を持っている。従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.図11.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）
　$C_{min}=(2n+0.8+Dl)\,\mu s$
(b) 通信コスト最大の場合（ｎデータ）
　$C_{max}=(2n+5.8+Dl)\,\mu s$
(c) 通信コストの平均値（ｎデータ）
　$\overline{C}=(2n+3.3+Dl)\,\mu s$

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（但し，Dl：ラインディレイ）。

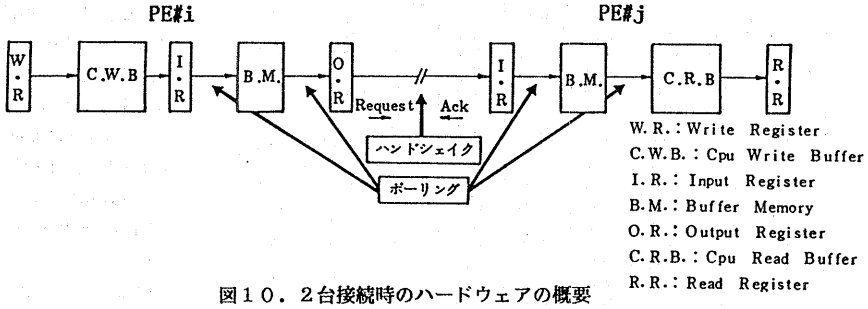
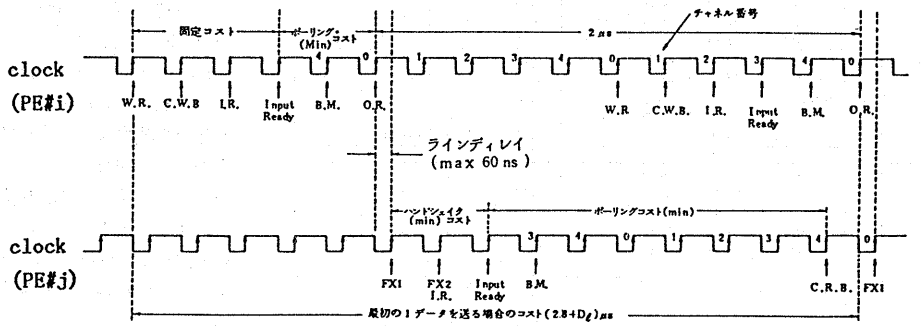


図１０．２台接続時のハードウェアの概要



◎　２データ目以後２μs 必要だから
　　ｎデータあれば　$2.8+D_\ell+2(n-1)=(2n+0.8+D_\ell)\,\mu s$

◎　１ clock＝0.2μs

図１１．２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）

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0.3msecもあり，これとファームのコスト並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。このことからも明らかなように，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版に言及しておく。表2.の最上段のケースを例にとり，仮りに処理系及びハンドラが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

処理時間比＝（通信コスト＋１リダクションのコスト）
／（１リダクションのコスト）

で与えられていたから，これは次式にとってかわる。

処理時間比＝（ハンドラのコスト/100＋ファームのコスト
＋ハードのコスト/5＋1リダクションのコスト/100）
／（１リダクションのコスト/100）

これを計算すると17.8倍が28倍になる。即ち第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．おわりに

現在ICOTにおいてマルチPSI第１版上で並列プログラムの動特性の計測評価を行っているが，今後も評価を継続しその結果をもとにマルチPSI第２版のネットワーク並びにシステムの開発を進める予定である。最後に，ご指導ご鞭達をいただいた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。

表２．測定結果一覧

（単位 msec）

測定項目（通信オーダ）	引数 個数	引数 タイプ	測定結果 *2 A	測定結果 B	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
throw_goal (含 ready, goal_terminates)	1	atom	62.72	23.90	*3 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
		*1 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
*4 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	
*5 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。

＜参考文献＞

〔1〕後藤，杉江，服部，伊藤，内田，“並列推論ﾏｼﾝPIM一中期構想一”，情報処理第33回全国大会，1986.

〔2〕瀧，木村，横田，近山，内田，“Multi-PSIｼｽﾃﾑ の概要”；情報処理第32回全国大会，1986.

〔3〕Taki,K. “The Parallel Software Research and Development Tool : Multi-PSI System” ,France-Japan Artifitial Intellingence and Computer Science Symposium 86, 1986.

〔4〕Ueda,K. “GUARDED HORN CLAUSES” Logic Programming Conference'85, JAPAN,1985.

〔5〕宮崎，瀧，“ﾏﾙﾁPSI におけるFlat GHCの実現方式” ,Logic Programming Conference'86,JAPAN,1986.

〔6〕木村，瀧，内田，“ﾏﾙﾁPSIｼｽﾃﾑとその接続方式”，情報処理第33回全国大会，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”，情報処理第33回全国大会，1986.

〔8〕益田，瀧，木村，“ﾏﾙﾁPSIのﾈｯﾄﾜｰｸ・ﾊｰﾄﾞｳｪｱ構成”，情報処理第33回全国大会，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情報処理第34回全国大会，1987.