

AIワークステーションPSI/UXのアーキテクチャ

益田嘉直 松本 明 池田守宏

三菱電機株式会社

第5世代コンピュータ・プロジェクトの一環として、これまでに論理型言語ESPを実行するPSI及びPSI IIを開発してきた。しかし、最近の知識情報処理システムは既存の情報処理機能に知識処理の機能を融合させて構築される傾向にある。これに対応するため、PSI IIの後継機として第5世代コンピュータ・プロジェクトの研究成果と世界標準OSであるUNIXの環境を統合した新しい概念のAIワークステーションPSI/UXを開発した。PSI/UXは、同プロジェクトにおける並列推論マシンPIM/mの要素プロセッサとして開発した推論VLSIをCPUに使用し、最大推論性能1.4MLIPSを実現している。

本稿では、論理型言語の特長を考慮したアーキテクチャの大幅な改良と、半導体技術の進歩による遅延時間短縮などの相乗効果により、従来のPSI IIに比べて2~3倍以上の高性能を得たPSI/UXの設計思想、アーキテクチャならびに性能評価について述べる。

Architecture of AI Workstation PSI/UX

Kanae MASUDA Akira MATSUMOTO Morihiro IKEDA

Mitsubishi Electric Corporation

325 Kamimachiya, Kamakura, Kanagawa 247 JAPAN

Knowledge information systems are gradually being implemented and developed. We have developed the new AI workstation PSI/UX which takes advantage of research results from Japan's Fifth Generation Computer Project. The inference unit of PSI/UX employs VLSIs developed in the project. Continuing the tradition of previous PSI series, PSI/UX offers very high inference performance 1.4 MLIPS while incorporating the world-standard UNIX operating system.

This paper describes the design concept of PSI/UX and its architecture for logic programming execution. We have evaluated the inference performance. This results show that PSI/UX is two to three times as fast as PSI II.

1. はじめに

知識情報処理は第5世代コンピュータ・プロジェクトを契機として急速に進展し、最近では各分野で応用され、人間の知的活動を支援する技術として注目されつつある。このような中で、同プロジェクトの一環としてこれまでにProlog系の論理型言語ESPを実行するPSI及びPSI IIを開発してきた[1],[2],[3]。しかし、高度な知識情報処理システムを実現するためには、より高速な推論性能や高機能化に加えて、ネットワーク機能の充実、ユーザインタフェースの標準化など既存情報処理システムの標準化動向にそった機能・高性能化が一層求められている。今回、このようなニーズを満たすためにPSI IIの後継機として第5世代コンピュータ・プロジェクトの研究成果と、世界標準OSであるUNIX（注）の環境を統合した新しい概念のAIワークステーションPSI/UXを開発した。

本稿では、PSI/UXの設計思想、概要について述べた後、PSIのOSであるSIMPOSとUNIXとの結合方式、推論性能の高速化を実現したアーキテクチャ、及びその性能評価について述べる。

2. PSI/UXの概要

2.1 設計方針

PSI/UXの開発に当たっては、最近の知識情報処理をとりまく環境の変化を考慮し、以下のような基本方針で設計を行った[5]。

(1) 知識情報処理の進展に伴い増大するCPU需要に対応できるよう、PSI IIの2倍以上の実効推論性能を実現する。

(2) 高度な知識情報処理システムは既存の情報処理機能に知識処理の機能を融合させて構築される傾向にある。これに対応すべく、PSIの提供する知識処理機能と、業界標準OSであるUNIXの既存情報処理機能との統合化を図る。

(注) UNIXシステム ラボラトリーズ社が開発し、ライセンスしている。

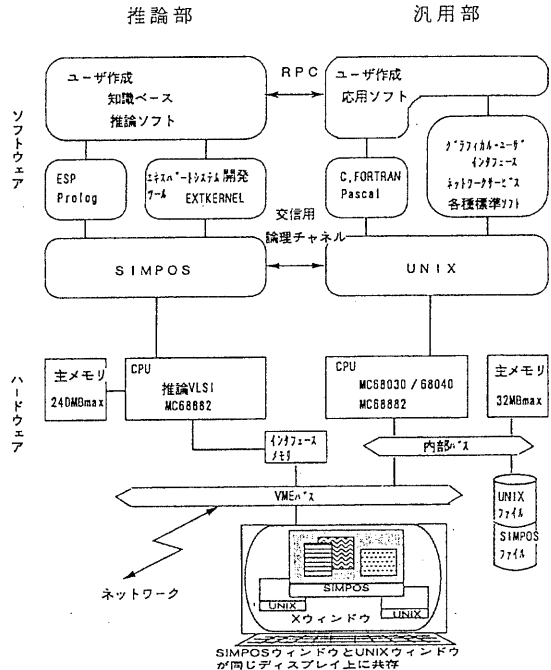


図1 PSI/UXのシステム構成

(3) PSI, おび PSI II の開発で培ったハードウェア、ソフトウェアの財産を最大限利用して効率よく開発する。

上記の目標達成のため、PSIのCPUをVLSI化により1ボードにして小型化、高性能化し、この1ボード化PSIとUNIXワークステーション相当のハードウェアを組合せてPSI/UXを実現した。ソフトウェアはPSI IIの上方互換とし、ハードウェアは第5世代コンピュータ・プロジェクトにて開発した並列推論マシンPIM/mの要素プロセッサの成果を利用している[4]。

2.2 システム構成

PSI/UXのシステム構成を図1に示す。PSI/UXは推論部と汎用部（UNIXワークステーション相当）から構成され、PSIのOSであるSIMPOSとUNIXの両システムの機能を同時に利用できる。

推論部はPSIとしての機能を提供する部分

で、推論CPUとメモリからなり、SIMPOS及びそのもとで動作するプログラムをここで実行する。PIM/mの要素プロセッサとして開発した推論VLSIをCPUに使用し最大推論性能1.4MLIPS(140万回推論/秒)を実現している。また、メモリには4MDRAM素子を使用しており、最大240Mバイトまで実装できる。

汎用部はUNIXワークステーション相当のハードウェアで実現しており、CPUとメモリ、及び入出力装置群からなる。CPUには汎用のマイクロプロセッサ(MC68030またはMC68040)を使用している。UNIX機能を提供すると共に、推論部のプログラムの入出力も実行する。

PSI/UXでは、図1に示すようにSIMPOSウィンドウを、UNIXの標準ウィンドウであるXウィンドウシステム上に実現した。PSIIのディスプレイ画面に相当するウィンドウがXウィンドウでの一つのウィンドウとして表示され、これがSIMPOS画面となる。その中に、SIMPOSの複数のウィンドウが表示される。SIMPOS画面の中のウィンドウ制御や、マウス、キーボードの処理は、SIMPOSで処理される。これにより、PSIIと完全な操作互換を実現した。

3. PSI/UXのアーキテクチャ

3.1 実現方式

PSI/UXの汎用部(UNIX)と推論部(SIMPOS)の通信は、交信用論理チャンネルと呼ばれる通信路を介して行われる。この交信用論理チャンネルを実現するために、汎用部の標準入出力バスであるVMEバスを使用し、推論部のインタフェースボードを介してこのVMEバスに接続する方式を採用している。インタフェースボード上には交信用のインタフェースメモリとその制御回路が搭載されている。インタフェースメモリは、物理的にはVMEアドレス空間上に設けた128KBの共有メモリであり汎用部と推論部のいずれからも読み書きが行えるようになっている。

通信を開始する側は、制御情報とデータをインタフェースメモリ上に書き込むと同時に割込み信号を送る。受取り側は、インタフェースメモリから制御情報とデータを読み取り制御情報に基づいた処理を行う。この基本機構の上に、交信用論理チャンネルと呼ぶ上位プロトコルをソフトウェアで実現し汎用部(UNIX)と推論部(SIMPOS)の通信を行う方式とした。

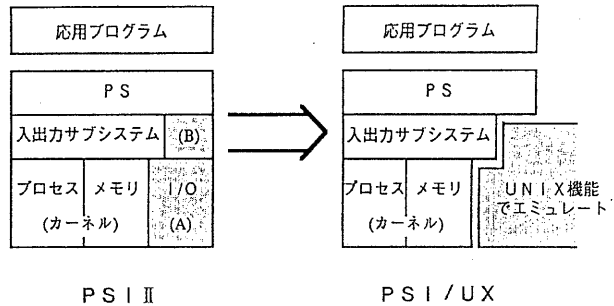


図2 SIMPOSの内部構造

PSI/UXでは、SIMPOSの固定ディスク、ウィンドウ、ネットワーク用ソケット等すべての入出力はUNIXの管理下にある入出力装置で実行される。このためSIMPOSの一部を改修している。図2はSIMPOSの内部構造を示しているが、図中影を付けた部分がSIMPOSをPSI/UX用に書き換えた部分である[7]。

但し、入出力サブシステムから下位の部分をUNIX機能でエミュレートするように書き換えたので、応用プログラムやプログラミングシステム(PS)では、オブジェクト・コード・レベルの互換を維持しつつ実現している。

次に、汎用部(UNIX)と推論部(SIMPOS)の通信を行うために設けた交信用論理チャンネルと各入出力サブシステムの実現方式について述べる。

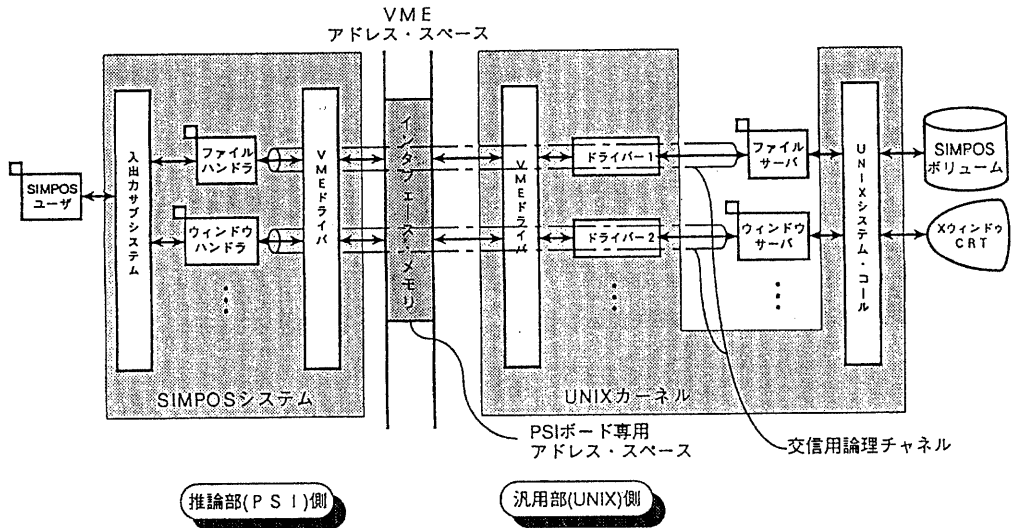


図3 交信用論理チャンネル

(1) 交信用論理チャンネル (図3)

交信用論理チャンネルは、SIMPOSとUNIX間を結ぶ論理的な通信路であり、この中を制御情報やデータが流れる。交信用論理チャンネルを実現するために、交信用のインタフェースメモリ(128KB)をアクセスするドライバをUNIXカーネルに追加した。

(2) SIMPOSファイル・システム

ディスク上にUNIXファイル領域とは別に、SIMPOSファイル領域を設けている。これは、両者のファイル・システムにはファイル名の文字数、バージョン管理方法、プロテクションの方式等で細かな相違があるためである。

UNIX側からSIMPOSファイルへの読み書きはSIMPOSを経由して行う。例えば書き込みデータは、制御情報とともにSIMPOS上のディスクハンドラから交信用論理チャンネル経由でUNIX上のディスクサーバ・プロセスに送られた後に、UNIXのシステムコールを用いて実際にSIMPOS用ディスク領域に書込む。PSI/UXではディスクキャッシュをSIMPOS上のメモリ空間に設けた。このためキャッシュ上にデータがある場合はUNIX側へのアクセスが発

生しないのでより高速にディスクアクセスできる。

(3) SIMPOSウィンドウシステム (図4)

SIMPOSウィンドウシステムは、大きく分けると次の3つの部分から構成される。

① SIMPOSウィンドウマネージャ

SIMPOSウィンドウシステムの論理的な管理を行う。具体的には、ユーザからの要求を解析し、UNIX側に対して描画コマンドを生成発行する。また、マウスやキーボードからの入力データを各SIMPOSウィンドウへ振り分ける。

② ウィンドウ交信ハンドラ

UNIX側とのデータ送受信処理を行う。

③ SIMPOSウィンドウサーバ

PSI側からのコマンドの受信・実行、Xサーバからのイベントと処理、およびPSI側へのデータの送信を行う。ウィンドウサーバはXlibレベルで実現している。

SIMPOSウィンドウサーバは、Xウィンドウをベースにして実現したため、ネットワーク上に接続されているXサーバ付きのワークステーションやX端末からPSI/UXのSIMPOSウィンドウを使用することが可能である。

(4) SIMPOSネットワークシステム

SIMPOSのソケット(ネットワーク通信の仮想的な端点)機能は、ソケット一つごとに、UNIX上のソケットサーバプロセスを一つずつ割り当てて実現している。これによって、UNIXでサポートしているソケット機能をSIMPOSからもそのまま利用できる。また、このソケットを利用してUNIX上の標準的プログラム間通信機能であるRPC(Remote Procedure Call)をSIMPOS上にも実現した。これによって、SIMPOS上においても、UNIXと同じようにRPC機能を用いたサーバクライアントのプログラムを実現することができる。

3.2 高速処理方式

PSI/UXは、ESP及びPrologを高速に実行するのに最適なハードウェア方式を採用している。ESPは、Prologにオブジェクト指向の機能を組込んだもので、記述性が高くシステムの開発効率を高めることができる反面、Prolog特有のユニフィケーション、バックトラック処理、及びオブジェクト指向のためのオブジェクト間メッセージ通信といった大きな処理を実行時に行う必要がある。このためESPの高速化を目標に各種のハードウェア方式を採用した[4],[6]。

以下にその主なものについて述べる。

(1) タグアーキテクチャ

パイプラインの上流にはオペランドのタグにより実行ステージのエントリアドレスを修飾する機能や、2方向、多方向の分岐条件を設定する機能がある。また、実行ステージにおいてはタグによる各種分岐機能が用意されている。これらの強力なタグ判定機構でデータタイプによる処理内容の変更を効率的に行うことができる。

(2) 5段のパイプライン制御

PSI/UXは、Warrenの提案した仮想マシン(WAM)の命令セットを拡張した機械語命令セットを持ち、ESPプログラムはこの機械語命令列にコンパイルされて実行される。機械語命令は、推論実行時に最適化された五つのパイプラインステージに分解され、複数の機械語命令が並列に実行される。5段のパイプラインの各ステージは順にD(デコード)、A(アドレス計算)、R(データ読出し)、S(データセットアップ)、E(実行)に分かれている。Eステージはマイクロプログラムによって制御される。パイプライン上流ステージはオートデリファレンス機能を有しており、Prolog特有のデリファレンス処理を自動的に実行できる。

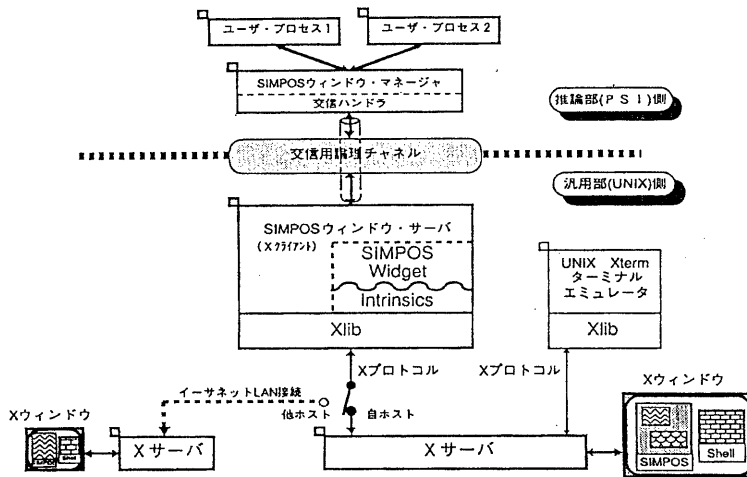


図4 SIMPOSウィンドウの実現方式

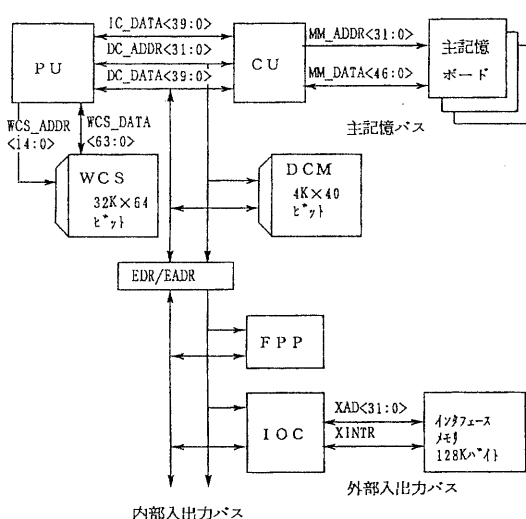


図5 推論部のハードウェア構成

(3) メモリアーキテクチャ

キャッシュメモリを命令用とデータ用に分離する、いわゆるハーバードアーキテクチャを採用することによってパイプラインステージ間のメモリアクセス競合による性能低下を抑制し、性能向上を図っている[4],[9]。

また、バックトラック処理を高速に実行するneck-cut最適化のために新規にundo情報を保持するトレイルバッファを導入した。

3.3 ハードウェア構成

P S I / U X のハードウェアは推論部と汎用部から構成されている。図5に推論部のハードウェア構成を示す。推論部のハードウェアはV L S I や高速S R A M を使用し、C P U ボードを1枚で実現し、推論機能の向上とともに小型化・高信頼性化を図った。

C P U の中心となるのがP U とC U の2種類の推論V L S I であり、最先端のセルベース方式で開発したV L S I である。図6にP U 、図7にC U の構成を示す[4]。

(1) P U (Processing Unit)

P U は5段のパイプラインで制御され、E S P 、P r o l o g を高速に実行することができる。

(2) C U (Cache Unit)

C U は命令キャッシュ(5Kバイト)、データキャッシュ(20Kバイト)および主記憶インターフェースで構成され、P U と主記憶間のデータ転送を高速に行うことができる。

(3) I O C (Input Output Control)

内部入出力バスを外部入出力バスに変換するための回路であり、外部入出力バスにはインタフェースボードが接続される。I O C は、上記の機能のほかにシステムタイマ、割込み制御、デバッグ機能も提供している。

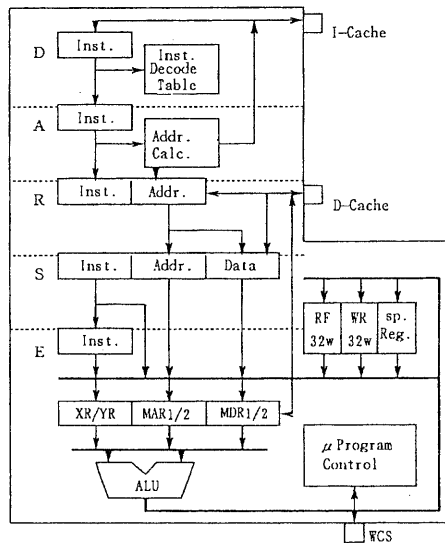


図6 P U (Processing Unit) の構成

(4) インタフェースボード

交信用インタフェースメモリ(128Kバイト)を搭載し、汎用部のV M E バスに接続される。推論部と汎用部のいずれからも高速にアクセスできる2ポートメモリを実現している。

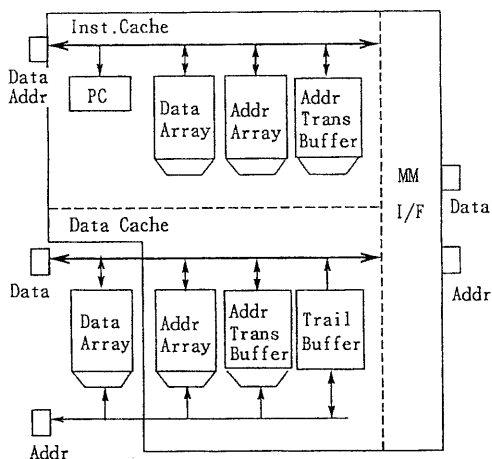


図7 CU (Cache Unit)の構成

4. 性能評価

4.1 ベンチマーク性能

PSI/UXの代表的なPrologのベンチマークプログラムの実行性能を表1に示す。参考までにPSI IIとの性能比も付記したが一般的にPSI IIの2~3倍以上の実行性能が得られている[8]。

これに関しては、特にパイプライン化されたデータタイプ判定/デレファレンス処理及び新たに導入したトレールバッファが、マシンサイクルとサイクル数の両面で性能向上に大きく寄与していることが明らかになっている[4],[9]。

表1 PSI/UXのベンチマーク性能

ベンチマーク	性能 (KLIPS)	処理時間 (msec)	対PSI II 性能比
APPEND	1393.3	0.00072	3.24
LISP-TARA1	0.183	306.40	2.61
LISP-FIB	191.3	38.79	2.66
LISP-NREV	232.2	15.06	2.88
QSORT50	422.7	1.44	2.24
NREV30	1074.7	0.47	2.96
8QUEEN(ONE)	769.7	7.48	2.80
8QUEEN(ALL)	822.7	121.55	2.80
RVSBL	230.5	5.64	2.46
SREV4	434.0	0.49	3.19
SREV5	332.8	2.57	2.42
SRVE6	281.3	12.13	2.23
TRAV1000	332.3	6.38	2.50

4.2 プログラム実行特性

表2に2種類ずつのベンチマークプログラム、応用プログラムについて、PSI/UXのプログラム実行特性及びPSI II性能比を示す。同表からベンチマークプログラムと応用プログラムの間にPSI IIとの性能比に差があることが分かるが、この差は命令キャッシュのヒット率、実行(E)ステージの稼働率(Eステージが動作していたクロック数/全実行クロック数)などに依存する。

上記の各プログラムのメモリアクセス特性(データキャッシュ、命令キャッシュのヒット率)、パイプライン動作特性(パイプラインキャンセルの頻度)をそれぞれ表2(b),(c)に示す。PSI/UXにおいてキャッシュがミスヒットした時のメンテナンスに要する時間はPSI IIと同じため、応用プログラムでそれほど性能が出ていないのはデータキャッシュのヒット率によると予想していたが、そうでないことが表2(b)よりわかる。アクセス頻度とも相関は見られなかった。また、命令キャッシュのヒット率はコンパイラ以外は比較的小規模なプログラムであるため、ほとんど100%となっている。コンパイラはOSのメソッド呼出しなどプログラムが広範囲に渡っているため、やや低い値になっている。コンパイラのような大規模プログラムではこのように命令キャッシュのヒット率が下がる傾向にあるので、このことが性能向上を抑制する一つの要因となっている。なお、アドレス変換バッファのヒット率は、すべて100%であった。

パイプラインが途切れる頻度は直接性能差に影響していないことが表2(c)よりわかる。表2(d)のEステージの稼働率は明らかに性能値と相関があり性能向上の高いプログラムほどEステージの稼働率も大きい傾向がある。

PSI/UXのジャンプ命令は次のような種類に分類される。

- ① 無条件分岐 (a) ②無条件分岐 (b)
- ③ 2方向条件分岐 ④ 多方向分岐

①は分岐することが明らかで、分岐先もわかっているような命令である。この場合パイプライ

ン上流で指定することでパイプラインの乱れを最小限にしてジャンプできる。②はEステージに来るまで分岐先がわからず、パイプラインキャンセルを必ず伴うタイプである。③は与えられた条件を満たさなかった時のみ②のような動作となるが、条件を満たした場合はパイプラインは一切乱れない。④は動作としては②と同じである。各プログラム中で実行された命令を調べたところ、Eステージの稼働率は実行時まで次の命令実行アドレスが求まらず、かつ、プログラムカウンタ(PC)を正しい値に更新するとすぐに処理を終えるような命令の頻度に関係していた。表2(e)にこの様子を示す。このような命令は、パイプラインを乱すため効率的な処理を妨げることになる。

5. おわりに

PSIのSIMPOS開発・実行環境と、UNIXの環境を統合したPSI/UXの設計思想、アーキテクチャ、ならびに性能評価などについて述べた。

PSI/UXは、並列推論マシンPIM/mの要素プロセッサとして開発した推論VLSIを用いることにより、推論CPUを1ボード化することができた。また、論理型言語ESPの特長を考慮したアーキテクチャの大幅な改良と、半導体技

術の進歩による遅延時間短縮などの相乗効果により従来のPSI IIに比べて2~3倍以上の高性能を得ることができた。今後の課題としては、ウィンドウ表示の高速化と機能強化、及びRPCの機能強化等が挙げられる。

最後に、PSI/UXの開発に際して御指導を頂いたICOT研究部内田俊一郎部長をはじめとする関係各位に深謝する。

参考文献

[1] S.Uchida, et al.: Research and Development of the Parallel Inference System in the Intermediate Stage of the FGCS Project, Proc. of the Intl. Conf. on Fifth Generation Computer Systems 1988(1988).
 [2] H.Nakashima, et al.: Hardware Architecture of the Sequential Inference Machine: PSI II, 4th IEEE Symp. on Logic Programming(1987).
 [3] K.Masuda, et al.: Preliminary Evaluation of the Connection Network for the Multi-PSI system, Proc. 8th European Conf. on AI(1988)
 [4] 中島(浩), 武田, 中島(克): PIM/m要素プロセッサのアーキテクチャ, 並列処理シンポジウムJSP'90(1990).
 [5] 上田, 湯浅, 益田: AIワークステーションPSI/UXの概要, 情報処理学会第43回全国大会, 2Q-1(1991)
 [6] 田辺, 池田他: AIワークステーションPSI/UXのハードウェア, 同全国大会, 2Q-2(1991)

[7] 松本, 高橋他: AIワークステーションPSI/UXのソフトウェア構成, 同全国大会, 2Q-3(1991)

[8] 佐伯, 池田他: PIM/mフロントエンドプロセッサの速度性能評価, 情報処理学会第42回全国大会, H2-4(1991)

[9] 中島(浩), 武田: PSI IIのメモリアーキテクチャ評価, 情報処理学会計算機アーキテクチャ研究会, ARC-80-8(1990)

表2 PSI/UXのプログラム実行特性

項目	プログラム	ベンチマークプログラム		応用プログラム	
		QUEEN (ONE_SOL)	NAIVE_ REVERSE	ソシハイ	最短経路問題
(a)	対PSI II性能比	2.80	2.94	2.10	2.22
(b)	データキャッシュヒット率	0.980	0.974	0.983	0.976
	命令キャッシュヒット率	0.999	1.000	0.935	1.000
(c)	パイプラインキャンセル頻度 (回/全実行クロック)	0.021	0.007	0.015	0.015
	パイプラインキャンセル頻度 (回/全実行命令)	0.089	0.022	0.101	0.058
(d)	実行ステージ稼働率	0.868	0.927	0.707	0.779
(e)	効率の悪い パイプラインキャンセル頻度 (回/全実行クロック)	0.001	0.003	0.011	0.009