

《解説》

ポリプロセッサ・システム PPS-1

元岡 達* 山室 良夫**

1. まえがき

複数台のプロセッサが、主記憶を共有する形式 (Memory Share) で結合されているシステムを、一般に、多重プロセッサ・システム (Multiprocessor System) と呼んでいる¹⁾。従来の多重プロセッサ・システムには、本来、1台のプロセッサからなるシステムが基本にあつて、システムを拡張する目的で、同種のプロセッサか同族のプロセッサを、2~3台接続するシステムが多かつた。

多数のプロセッサが、主記憶を共有する形で結合されたシステムを基本システムとし、各プロセッサは、一定の基本条件を満たしていればどんなものでもよいシステムを、従来の多重プロセッサ・システムと区別する意味で、ポリプロセッサ・システムと呼ぶことにする²⁾。

ポリプロセッサ・システムの1例である PPS-1 の試作目標は、以下に示すようなものである。

(1) 拡張性の高い、成長可能なシステムを提供すること。

プロセッサを含めたすべてのリソースが、目的に応じ必要量に応じて適当数利用でき、小規模システムから大規模システムまでを、連続的に拡張可能にする。従来のシステムでは、記憶と入出力機器に関しては、ある程度この条件を満たしていたが、プロセッサの処理能力については、不十分であつた。

(2) 目的指向形の、柔軟性のあるシステムにすること。

プロセッサを、従来の汎用プロセッサとせず、目的指向形プロセッサとすることによって、この目標を達成しようとしている。このため、マイクロプログラム・プロセッサとし、しかも、制御記憶に RAM を用い、同一システムを、多目的に順応できるようにした。

マイクロプログラムによって、各種のプリミティブを用意し、高級言語の処理、プロセッサ間並列処理、ファイル管理、入出力処理、リスト処理、波形処理などの高性能専用プロセッサを作ること、今後に残された研究課題であり、この目的がどこまで達成できるかは、未知数である。

専用プロセッサ化しても、プロセッサの核になる部分は共通化しておかないと、ポリプロセッサ・システムとして働かせることはできない。核となる部分に何が必要かをしらべ、確認することも、本システムの目的の1つである。

(3) 性能/価格比のすぐれたシステムにすること。

従来機種にくらべて、性能/価格比が一般に劣るのでは、特殊用途にしか用いられないことになる。このため、本プロセッサでは、従来機種に対するエミュレータ性能に、十分考慮をはらったアーキテクチャを採用している。量産効果を考えれば、汎用システムとしても十分対抗できる性能/価格比を、確保できる予定である。また、将来、LSI 化することも考慮した設計になっている。

(4) 信頼性の高いシステムにすること。

リトライなどの回復機能を、マイクロプログラム・レベルで組み込むことを考慮したアーキテクチャになっており、また、プロセッサ相互間通信のインターフェースを用いて、相互診断を可能にしている。手法としては、マイクロプログラム診断であるが、診断専用のハードウェアをほとんど用いない点で、ポリプロセッサの特色が生かされた方式といえる。

(5) 並列処理に適したシステムにすること。

同一ジョブを、多数のプロセッサが分担処理する方式の並列処理システムは、高速化による性能向上に限界がでてきたことから、各方面で関心をもちられている。本システムも、このような目的に対する道具を提供することにある。どの程度の効果を発揮するかは、管理プログラム、処理言語など、ソフトウェアとしてどのようなものが準備できるかに大きく依存するので、その評価は、今後の研究にまつ必要がある。

* 東京大学工学部電気工学科

** 富士通(株)第2開発技術部第3開発課

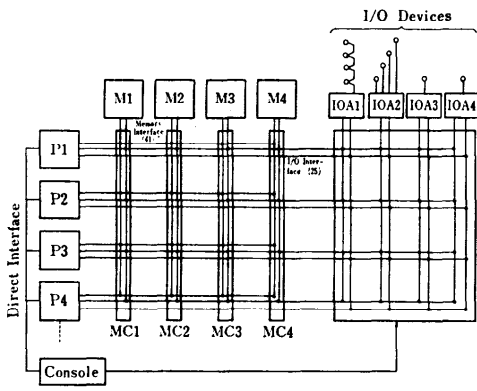
2. システムの構成

プロセッサ、記憶、入出力装置といったリソースを、できるだけ自由に希望数だけ接続でき、しかも、小規模システムにおいても、高価にならない結合方式が理想である。自由な接続法を可能にすること、安価な結合方式とは、一般に矛盾する要求であり、従来、主として価格の点から、接続法に制限が加えられていた²⁾。

システムの構成法に対するもう1つの要求としては、システムの信頼性を高めうる構成法であることである。特定のリソースが故障することによって、システム全体がダウンしては困る。フェイルソフトなシステム構成がとれるよう、配慮する必要がある。

これらの要求を考慮して、図1に示すようなリソース結合方式を、基本結合方式として考えることにした。この結合方式は、あくまで基本であり、他の結合方式を否定するものではなく、インターフェースを変更することなく、他の結合方式もとれるものと考えている。

プロセッサと主記憶用の接続には、プロセッサ専用バス方式を用いる。データ伝送の要求発生源は、プロセッサであり、その限りにおいて、主記憶は受動リソースである、という立場からとった方式である。ただし、多数のプロセッサからの要求が、特定の主記憶に集中し、競争を生じたときの解決とその制御は、各主記憶に付属した制御回路が分担する。したがって、主記憶相互間の直接データ転送はできず、必ずプロセッサを介して行なうことが、原則である。



P: プロセッサ, M: 主記憶, MC: 主記憶制御回路, IOA: 入出力アダプタ

図1 PPS-1 のシステム構成

この方式では、情報転送の隘路が結合部に生じることはないが、切換え回路はマトリックス方式なみに必要になり、高価になる恐れがある。インターフェース線の数を減らし、切換え回路を簡単にする努力が必要である。

プロセッサと入出力装置間の接続については、当初プロセッサからみたインターフェースを、主記憶と入出力装置で同一なものにし、ユニバスの使用方法をとることを考えたが、データ転送線など一部のインターフェース線を論理的に共有させるだけで、実質的には、別のインターフェースをもうけることになった。プロセッサのデータ入出力用レジスタとしては、主記憶、入出力装置とも同一レジスタ(図3のYレジスタ)を用いており、インターフェースの共通化については、今後、さらに検討をすすめる予定である。

プロセッサと入出力装置間の接続は、マトリックス方式であり、切換えスイッチの制御は、コンソールを介して一括して行なう。プロセッサは、コンソールに頼み、ここが集中して行なうから安価になるが、切換えに時間を要すること、コンソールの事故時にすべての自動切換えが不能になることの欠点が生じる。切り換えられた状態は、つぎに切換え指令の来るまで保持される。1つの入出力アダプタが、ある時点で接続されているプロセッサは、1台以下である。

プロセッサ相互間、およびプロセッサとコンソール間のデータ転送用のバスがある。バスに接続されるリソースが、まったく平等な形で専有権を主張でき、バス制御回路を、すべてのリソースに平等に分散配置する方式もあるが³⁾、高価になるので、バス制御回路をコンソールに集中し、プロセッサは、必要に応じて要求を出し、コンソールに専有権を与えてもらう方式を

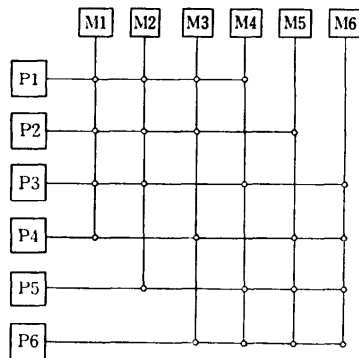


図2 プロセッサと記憶の結合例

採用した。制御は簡単になるが、コンソールが故障した場合、プロセッサ間のデータ転送が不能になる欠点がある。

論理的には、各種のリソースが、それぞれ 16 台まで接続可能であるが、実装されているスイッチでは、それぞれ 4 台までしか接続できない。リソース数が多くなると、スイッチが高価になるだけでなく、Fan-out Fan-in の増加にともなう遅れの問題があり、一方では、どの程度までこのような密結合をするのが得策か明らかでない。現在のスイッチでも、図 2 のような構成をとれば、接続台数を増すことができる。このことは、入出力装置とプロセッサの間についてもいえる。

3. プロセッサ・アーキテクチャ

3.1 設計目標

プロセッサの設計時に考慮した基本事項は、つぎのとおりである。ブロック図を、図 3 に示す。

(1) マイクロプログラム制御方式とし、制御記憶には、RAM を用いて書込み可能にする。マイクロ命令のビット長は、経済性を考慮して、24 ビットに制限する。柔軟性を重視し、エミュレータ、仮想プロセッサとしての使用法に徹して、特定の機械語を想定しない。

(2) ポリプロセッサ・システムに使用できるよう相互同期、相互通信の手段を用意し、結合が容易になるよう、インターフェースを整理する。

(3) 拡張性をはかるため、バス方式をとり、マイクロ命令形式に拡張の余地を残す。

(4) 将来の LSI 化を考え、外部とのインターフェースをできるだけ整理し、簡単なものにする。

(5) RAS 技術については、大形システムに比して遜色ないものにして、可用性、保全性を重視する。

(6) 仮想記憶方式がとれるように、最小限のハードウェアを準備する。

(7) 性能/価格比が高くなるよう、IBM 370 系のエミュレータとして使用できる機能を、最小限確保する。

(8) プロセッサ間通信を、直接可能にするとともに、他プロセッサによる診断も可能にする。

3.2 制御記憶

制御記憶 CM は、4~8 kW 実装することを考えているが、論理的には、64 kW まで指定できる。マイクロ命令語は 24 ビットであるが、このほかに、診断のために、1 と 0 との反転を示す反転ビットが 1 ビッ

トと、パリティビットが 3 ビットついているので、記憶ビット数は、1 語当たり 28 ビットである。マイクロ命令カウンタとしては、CMADR I が用いられ、割込みやサブルーチン呼出し時に用いるためのスタックが、8 W 用意されている。CM へのデータ転送用アドレス・レジスタとして、CMADR II が準備されている。

3.3 演算機能など

プロセッサのデータ語は、16 ビット構成で、ゼネラルレジスタ用に、64 W のローカル記憶 (LM) が用意されている。マイクロ命令による LM アドレスの指定には、直接指定のほか、BR 0~2 による間接指定も可能になっている。

演算機能は X, Y レジスタの内容について、語またはバイト単位の加・減算、論理演算が可能で、結果は A バスにつながるレジスタと LM に入れることができる。

このほか、4 ビットからなるディジット単位の演算が可能である。シフト機能は、Y と YE レジスタに組み込まれている。

マイクロプログラムの分岐のための機能には、分岐の条件と分岐先の相対番地とを指定できるものと、無条件ブランチとがある。後者では、絶対番地による直接ブランチ、間接ブランチ、スタックによるブランチアンドリンクのほかに、レジスタでアドレスの下位ビットを指定する、テーブル・ブランチがある。指定レジスタには、BR 0~2, OP などがあ

3.4 主記憶とアドレス方式

主記憶の読出し・書込みには、仮想アドレス方式が使えるようになっており、実アドレスは 16 ビット (64 kB) であるが、仮想アドレスには 24 ビットが用いられ、アドレス・レジスタ MAR は、2 つ用意されている。仮想記憶用のハードウェアとしては、仮想頁アドレスと実頁アドレスの対応を示すレジスタ群が 6 W 準備されており、仮想頁アドレス 14 ビットが、実頁アドレス 6 ビットに変換される。対応のつかないときは、後述するマイクロレベルでの割込みを生じ、その読出し・書込み命令 RW は実行されない。このアドレス変換バッファ TLB は、シフトレジスタ形式であるため、この書換えには、もっとも古く書き込まれたものから、順次書換えに使われることになる。

主記憶とのデータ交換をするのは Y レジスタであるが、マイクロ命令 RW はおいてきぼり制御方式をとっているから、Y レジスタを使用するマイクロ命令

み関係、プログラム割込みなど、9種が用意されている。

前者は、マイクロ命令の終了時点で発生する割込みであり、マシンチェック、TLB フォールト、入出力の3種の割込みがあり、優先順位も、この順になっている。割込みアドレスは、それぞれ 0100, 0102, 0101 (16進数) になっているから、3レベルに分かれているといえる。割り込まれたアドレスは、スタックにたくわえられる。

3.6 プロセッサ間通信機能

ダイレクトインターフェースの目的は、プロセッサ間の通信と故障診断とに大別でき、このインターフェースによって、プロセッサ相互間、およびプロセッサとコンソール間の通信手段が提供される。インターフェースの制御権はコンソールにあるから、プログラマは、まず使用許可を求め、許可があると、MTR にデータをセットして授受を行なう。前述の2目的に使うため、ラン状態とマニュアル状態では、動作が異なる。ラン状態では、MTR の上位バイトが転送の対象になり、送信命令の下位バイトで送・受信プロセッサが指定される。受信命令の下位バイトは、送信プロセッサとそのレジスタの指定に使う。相手方プロセッサがマニュアル状態にあるとき、受信命令はラン状態のときと変わらないが、送信命令では、下位バイトが、受信プロセッサとそのレジスタの指定になり、指定できるレジスタに、BUFF のレジスタの上, 中, 下位バイトと MTR のほかに、DCONT がある。DCONT の各ビットを使って、受信プロセッサの動作を指定できる。これと BUFF レジスタのセットを使って他プロセッサを動かす、診断を行なう。

4. マイクロ命令の形式

マイクロ命令の形式には、以下に示す 14 種の形式がある。命令形式のビット割当てを、表 1 に示す。

(1) ROP (Register Operation)

X レジスタと Y レジスタの内容に演算をほどこして、LM (Local Memory) か、内部レジスタ群の1つ (図3のブロック図 (a) にあるレジスタ) に結果を格納する。演算の種類には、加算、減算、論理積、論理和、排他的論理和、通抜けの6種があり、桁上げの処理も細かく規定できる。データの語長は1語 (2B) か1Bで、演算結果 (正、負、零など) の STAT レジスタへの格納や、桁上げの処理を制御できる。

LM のアドレスは、直接アドレス指定のほか、レジ

スタの内容による間接アドレス指定も可能であり、演算結果によるスキップ機能もある。

(2) DOP (Digit Operation)

Y, YE などディジット指定可能なレジスタの内容と、マイクロ命令の一部で指定するデータ (直接データ) との間に演算を行ない、結果を、ディジット指定可能なレジスタに格納する。演算結果の零、非零によるスキップ機能もある。演算結果を格納しないときは結果格納用の D レジスタ (Destination Register) が、演算数用の S レジスタ (Source Register) のマスクレジスタとして用いられる。

(3) RW (Read Write)

主記憶に対する読出し・書込み命令で、主記憶のアドレス (論理アドレス) は MAR で指定され、データの入出力レジスタとしては、Y レジスタが用いられる。データの語あるいはバイト単位での読出し・書込みのほか、Test and Set, Key の読出し・書込み、ステータスの読出しなどが指定できる。また、MAR とカウンタ (BR 0 か BR 1) の更新も同時にでき、カウンタの内容によるスキップ機能もある。読出し時、Y レジスタへのセットは、この命令の終了段階では保証されないから、他命令では、wait bit を用いて、セットの終了を確認する必要がある。

(4) LIM (Load Immediate)

マイクロ命令の一部で指定したデータ (直接データ) を、語あるいはバイト単位で、X, Y, BR, STAT レジスタに格納する。STAT レジスタを指定したときには、IBM 360/370 シリーズの condition code を作成するとき用いるのに便利で、特殊な制御を行なう。

(5) BRAN (Branch)

直接、間接の無条件ブランチのほか、BR レジスタの内容によるテーブル・ブランチ機能、CMADRI の値をスタック (STK) に保存するブランチアンドリンク機能がある。テーブル・ブランチ機能は、機械語命令別に分岐する場合に有効であり、ブランチアンドリンク機能は、機械語レベルの多重割込み機能処理に有効である。

(6) EXTR (Extract)

Y または YS レジスタ (YS は Y の2バイトを交換したもの) の内容を、直接データでマスクして、Y または BR レジスタに格納する。

(7) MLTR (Move LM to R)

LM の内容を、内部レジスタに格納する。Y レジスタに対しては、バイト単位の格納も可能である。

表 1 マイクロ命令形式

命 令	形 式																記 事										
	17	18	19	20	21	22	23	24	12	13	14	15	16	17	18	19											
Register Operation (ROP)	0	0	0	1	D	I	ALU	CF	Ci	SA	WB	HX	L	EX	WT	Skip. Modif	INV	xfy→LM, Register									
Digit Operation (DOP)	0	0	1	0	M/D	S	ALU	CF		Imm			MD	WT	WT	Skip. Modif	INV	Sf Imm under Mask →Skip, Sf Imm→D									
Read Write (RW)	0	0	1	1	Interface Bit		ALU	CF		Imm			Counter Select			Skip. Modif	INV	Memory Read, Write MAR 更新 Counter 更新									
Lead Immediate (LIM)	0	1	0	0			Immediate data																INV	Immediate→D Reg			
Branch (BRAN)	0	1	0	1			Immediate data																INV	無条件直接ブランチ " " 間接 無条件ブランチアンドリンク " " テーブルブランチ			
Extract (EXTR)	0	1	1	0			Immediate data																INV	Y or YS→BR, Y under Mask			
Move L to R (MLTR)	0	1	1	1	S	I	D										Y	B	Y	Variant	INV	LM→D					
Move R to L (MRTL)	1	0	0	0	D	I	D	S												Skip. Modif	INV	S→LM					
Move Ex Register (MEXR)	1	0	0	1	S/D		RQO	ACLO	CTLN	CTLF										WT	SD	INV	EX→Y				
Control (CONT)	1	0	1	0	Stat No		IORST	DiT	DiR	STKUP	TLBP	STP	SWAP	PSTR						WT	Set	Reset	INV	システム特殊制御			
Move R to R (MRTR)	1	0	1	1	D			S												WT		INV	S→D				
Control Memory Read Write (CMRW)	1	1	0	0	D			S												INC	WT	DEC	RW	INV	CM→D S→CM		
Branch On Bit (BOB)	1	1	0	1	TB		Displacement																INV	条件付きブランチ			
Shift (SHIFT)	1	1	1	0	Def		Shift No																WT	RL	YR	INV	ソフト制御

(8) MRTL (Move R to LM)

内部レジスタの内容を、LMに格納する。Sレジスタの特定ビット位置の、1, 0によるスキップ機能がある。

(9) MEXR (Move Ex Reg)

Yレジスタから外部レジスタ(図3のブロック図(b)にあるレジスタ)、外部レジスタからYレジスタへの転送命令で、入出力命令も兼ねる。

入出力命令のときには、IOインターフェースの制御線に出す信号を、直接データの形で指定する。入出力データ用には、Yレジスタが用いられる。

(10) CONT (Control)

STATレジスタのセット・リセット、他プロセッサとの交信など、各種の特殊制御に用いる。

(11) MRTR (Move R to R)

内部レジスタ間の転送命令で、Sレジスタの指定ビットの、1, 0によるスキップ機能がある。

(12) CMRW (Control Memory Read Write)

制御記憶CMの読出し・書込み命令で、アドレスはCMADR IIによって指定される。データの入出力レジスタは、命令の一部で指定する。アドレスの増・減指定や、スタックSTKからアドレスを取り出して用いCMADR Iにセットする機能などもある。

(13) BOB (Branch on Bit)

条件付きブランチ命令で、変位置と、条件として使用するビットが指定できる。割込み条件などが指定できる。

(14) SHFT (Shift)

Y, または Y と YE レジスタの桁送りを指定する。

マイクロ命令の取出し段階と実行段階は、図4に示すように、原則として並列処理される。したがって、マイクロ命令は、普通2クロック(440 ns)で処理されることになる。ただし、スキップが起ると、取出し段階をやりなおす必要が起るから、4クロックかかることになる。

主記憶のRW命令では、読出しに440 ns、書込みには660 nsを予定している。読出しデータがYレジスタで有効になるには、1クロックの遅れがある。このほか、間接ブランチは5クロックCMRW、条件付きブランチは4クロック、SHFTは2+シフト数だけのクロックが必要である。入出力、ダイレクト・インターフェースなどの命令では、不定である。

5. リソース間インターフェース

主記憶制御とプロセッサ間は、41本の信号線で接続され、2バイトのデータが、660 nsで転送される。すなわち、3 MB/sの転送レートである。データ転送用バスは、双方向性で、パリティを含め18本からなり、データ、アドレス、ステータス、制御コードなどは、すべてこのバスによって時分割転送される。記憶バンクに対する要求は、メモリ要求1ビット、バンク・アドレス2ビットの情報で、各バンクに割り振られる。制御線8本、エラー割込み線3本など計16本とバス線18本が、スイッチの対象になる。

同期方式をとっていること、インターフェース線の多いことなど、今後研究をすすめ、改良を要する点である。

主記憶容量は、バンク当たり16 kBで、1頁=1 kBごとに保護用のKeyがある。Test and Setのときは、読み出したバイトの最上位ビットを、最下位ビットとしてプロセッサに送り、記憶には、All 1を書き込む。プロセッサ間の競合や、同期の問題の解決用に準備した機能である。

入出力インターフェースとしては、プロセッサへ接続される比較的低速(30 kB/s以下)の入出力を考えており、高速インターフェースについては、今後、方式を検討する予定である。各入出力装置は、入出力アタッチメント(IOA)を介して、以下に述べる入出力共通インターフェースに接続される。各IOAには、複数台(論理的には16台、ただし6台実装)の入出

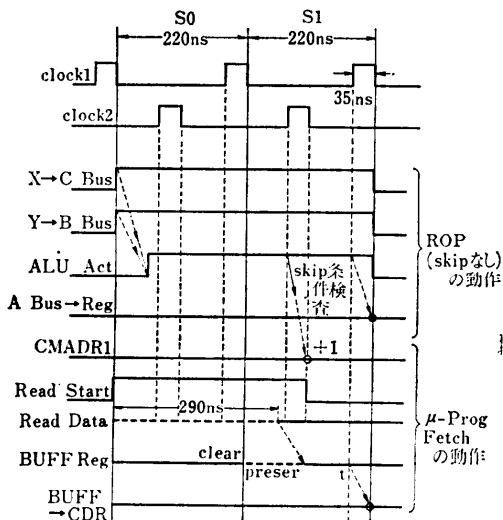


図4 PPS-1 タイムチャート例
(ROS命令でスキップのない場合)

表 2 各種インターフェース

名称	目的	総本数	データ転送幅	制御方式
DMA	プロセッサと主記憶間	41	18 ビット	各主記憶モジュール側で分散制御
Direct	プロセッサ (コンソールを含む) 相互間	76	9 ビット	コンソールで集中管理する集中制御
I/O 共通	プロセッサと IO アダプタ間	25	9 ビット	プロセッサごとに非同期処理

力装置を接続できる。IOA とプロセッサ間は、前述のように、マトリックス制御スイッチによって結合されており、このスイッチは、切り換えられるまで保持される。スイッチの切替えの際は、プロセッサがコンソールに指令を発生し、コンソール部がこれをうけて、スイッチ制御の集中管理をする。入出力インターフェースでは、データはバイト単位で転送され、信号線は、総数 25 本である。IOA からのサービス要求の割込みは、その時点で IOA の接続されているプロセッサに送られ、マイクロレベルの割込みになる。

プロセッサ相互間、およびプロセッサとコンソール間でデータ転送を行なうためのインターフェースを、ダイレクトインターフェースと呼ぶ。このインターフェースは、母線方式をとっており、3.6 節で詳述したように、母線の使用優先権管理はコンソールが行なっているから、各プロセッサは、コンソールに要求を出し、コンソールの許可を得て母線を使うことになる。故障プロセッサの診断や、コンソール上への表示などにも、このインターフェースを用いているので、76 本になっているが、データの転送は、バイト単位である。

以上、3 種のリソース間インターフェースをまとめると、表 2 のようになる。

6. ソフトウェア

このプロセッサは、制御記憶に通常の半導体記憶を用いており、半固定記憶は 0~127 番地までの 128 語にすぎず、ここには、初期マイクロプログラムローダ (IMPL) が入っているにすぎない。したがって、通常の機械語は存在せず、現在のところ、マイクロプログラムのクロスアセンブラが用意されているにすぎない。管理プログラムや各種コンパイラの作成は、今後の研究課題である。管理プログラムは、ジョブの基本単位をプロセスとし、このプロセス間通信の形で、相互の交信・同期の問題を解決する方式をまずとる予定

である。システム全般にわたる管理に関係した管理プログラムは、特定プロセッサの担当とし、これが故障した段階で、他プロセッサに切り換える方式を考えている。コンパイラなどに対しては、通常の機械語よりはやレベルの高いプリミティブを設定し、このプリミティブを、マイクロプログラムで実現する形式をとる予定である。

保守用ソフトウェアとしては、プロセッサが相互に診断する方式を考えている。基本的にはマイクロ診断の手法になるが、マイクロ診断プログラムを稼働させる核部の診断は、他プロセッサからの相互通信によって行なう予定である。

7. おわりに

PPS-1 は、今後の計算機システム構成法の 1 つと考えられるポリプロセッサ・システムの構成要素となるプロセッサに要求される機能と、リソース間のインターフェースを研究するために設計された試作機である。現在、試作を完了した段階であり、性能の測定・評価は、今後の課題である。プロセッサ単体の性能についても、本機のようにダイナミック・プログラム形式をとる場合、プリミティブの選択法の適否が、性能に大きな影響をおよぼすことが予想される。また、システムとしての評価になると、管理プログラムの方式の適否がこれに加えられ、簡単に結論は得られないかもしれない。エミュレータとして、原システムにどの程度対抗できるかが、1 つの評価基準となろう。

おわりに、本システムの基本設計に当たってご討論いただいた田中英彦助教授はじめ研究室の諸兄、ならびに市原第 3 開発課長はじめ富士通(株)第 2 開発技術部の諸兄に、深甚なる謝意を表する。

なお、本研究の一部は、昭和 48 年度科学研究費補助金 (特定研究 (I)) によった。

参考文献

- 1) 元岡: コンピュータ・コンプレクスの展望, 情報処理, 15, 7, pp. 525-533 (昭 49. 7).
- 2) 元岡: ポリプロセッサ・システム, 昭和 48 年電気四学会連合大会予稿集, pp. 1305-1308 (昭 48. 10).
- 3) 勝又, 元岡: 共通母線による計算機群結合方式, 電通学会計算機研究会資料, EC-71-22 (昭 46. 9).

(昭和 49 年 4 月 4 日受付)