

# 機能分散型計算機複合体のパイロット・モデル(PPS-R)

## A POLY-PROCESSOR SYSTEM for RESEARCH

村上 国男      西川 清史      佐藤 昌貞  
Kunio MURAKAMI    Seishi NISHIKAWA    Masasada SATO  
日本電信電話公社 武蔵野電気通信研究所

Musashino Electrical Communication Laboratory, NTT

### 1 まえがき

最近の超LSI技術の発達とマイクロプログラムミング技術の発展は、機能集中化に必ずしも計算機システム設計の最良の指導原理ではない事を明らかにした。将来のTSSにおける各種の要求を機能集中化アーキテクチャによって実現するのは、次の如く困難である。

- (機能相互間のインタフェースが簡明でなく) サービスの変更や新規ハードウェアの導入に対するシステムの拡張性が乏しい。
- (システム増設時の増設単位が大きすぎて高価である為) 価格性能比の最適領域が極めて狭い。
- (障害の局所化が困難で、部分的な障害がシステム・ダウンにつながる) 高信頼性の確保が困難である。

本稿で対象とする多プロセッサシステム(PPS)は、極めて密に結合された、機能的に専用化された、小型・多数のプロセッサにより構成される計算機複合体である。

従来から、小型・多数のプロセッサを密結合した計算機複合体として、多くのシステムが開発されてきたが、これらの多くは均質型マルチプロセッサシステムであった。[1,2] 一方、PPSは、多数の機能的に専用化されたプロセッサで構成される異質型マルチプロセッサシステムであり、従来の少数同種大型プロセッサを主記憶装置を介して結合したマルチプロセッサシステムに比し、次のような効果が期待できる。

(1) 増設単位の小型化および少品種多量生産に

よるLSIの経路化効果を活かし、広範なシステム規模に対して均一かつ良好なコスト当りの価格性能比を実現する事に可能である。

(2) プロセッサ対応の機能専用化によって、改造による変更範囲の局所化が可能となり、ハードウェア、ソフトウェアに関する新技術導入が容易となる。この為、システムの高度の拡張性と寿命化が達成可能である。

(3) 並列処理による応答時間の短縮化が可能である。

(4) 多数プロセッサ構成と機能の分散による、システムの高信頼化が達成可能である。

(5) 各プロセッサクラス対応に比較的小規模で単能なサブOSを構成し、標準化されたインタフェースでOS複合体に統合するので、OSの生産性の向上が可能である。

一方、以下に示すような欠点も予想され、これらに克服する技術の開発がPPSの有効性を左右すると考えられる。

(6) 機能分散化による、プロセッサ間通信処理の為にシステム・オーバーヘッドが増大する。

(7) 機能間の負荷不平衡がプロセッサ間の負荷不平衡とつながりあうため、ボトルネック・プロセッサの為にシステム性能が劣化する。

(8) ハードウェアの分割機、結合装置の追加等により、システム価格が上昇する。

本稿では、上記PPS方式の有効性の検証とアーキテクチャ実現の為に基本技術の確立を目的として開発したパイロット・モデル(PPS-R)システムを報告する。

2では、機能分散規則によって設定された構成要素およびシステムの論理構成を示し、3では、このような構成を持つPPSがどのような動作特性を示すかをシミュレーションによって解析する。4および5では、この解析結果に基づいて開発されたアーキテクチャ実現のための主要な技術およびPPS-Rの全体構成、諸元などを示す。

## 2 システムの基本構想

機能分散型システムを構成する際の第一の問題は、システムに課せられた外部条件に対するどのレベルの機能をいかに分散するかとなくである。本稿で述べるPPSは、図1に示すように、プロセッサ系、メモリ系、結合装置系で構成され、各系は、PPSの特性を最大限に発揮するよう、機能分散規則に従って、いくつかのクラスに分割されている。[3, 4]

### 2.1 プロセッサ系の構成

プロセッサ系における機能分散は、並列処理による性能向上、専用化による拡張性向上および信頼性の確保を担いとしている。クラス分割は、システム外乱(サービスの変更等)と実現すべき処理機能と制御の流れとに着目し、ハードウェア・ソフトウェア間インタフェース、外部系とのインタフェースおよびサービスとのインタフェースに関する分散規則によって、以下の6種のクラスを設定した。

**JPU**: 利用者によって起動された処理プログラムを実行する。

**LPU**: 通信回線、端末とのインタフェースを介し、通信制御、制御機能を実行する。

**SPU**: ジョブ、プロセスのスケジューリング、負荷の制御を行う。

**FPU**: ファイル記憶装置、入出力装置とのインタフェースを介し、ファイル管理、アクセス制御、入出力操作を行う。

**RPU**: スワッピング・メモリとのインタフェースを介し、ロールイン/アウト、実メモリ、プログラムの管理などを行う。

**MPU**: システムの各ハードウェア要素の診断、再構成処理を行う。

### 2.2 メモリ系の構成

メモリ系における機能分散の主目的は、プロセス可変性の実現と障害の局所化による可用性の確保、および、メモリ競合の削減による価格性能比の向上であり、格納情報の属性に着目して設定した分散規則により、メモリ系を次の6クラスに分割した。

**BCM**: 全プロセッサクラス共通の基本機能を実現するマイクロプログラム(μP)を格納する。この内容はプロセッサの発身が起っても不変である。

**FCM**: 各プロセッサをハードウェア的に特徴づけるμPを格納する。プロセッサ発身の度内容が変更される。

**PMEM**: 各プロセッサにおけるシステム処理を特徴づけるEXECおよび制御情報を格納する。

**JMEM**: ユーザプログラムおよびデータを格納する。

**ESWAP**: EXECの非常駐部を格納する。

**JSWAP**: ロールアウトされたプロセス情報、ライブラリなどを格納する。

### 2.3 結合装置系

本系における機能分散の主目的は、ハードウェア間インタフェースの標準化による増設性の確保である。この観点から、結合される

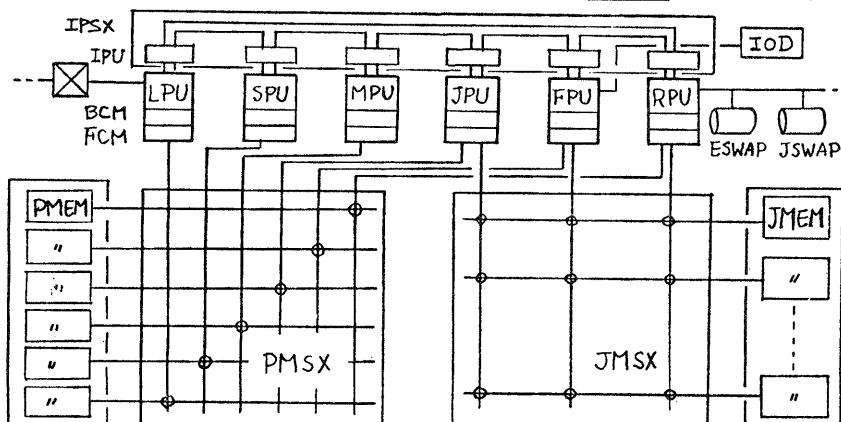


図1 PPSの論理構成

装置の属性と結合装置を通過する情報の質と量によってプロトコルを分類し、以下のようなクラス分割を行なう。

**PMSX**：PMEMと各プロセッサを結合する完全群構成のSXである。各PMEMは、対応するクラスのプロセッサと半固定的に接続され、障害時などの他クラスへ着身が必要な時のみ他クラスのプロセッサと結合される。

**JMSX**：JMEMとプロセッサを結合する完全群構成のSXであるが、全プロセッサクラスが結合されるとは限らない。

**IPSX**：高速データ伝送用転送路が多重に張られており、各転送路は全プロセッサクラスに接続され、独立な転送動作が可能である。

### 3 システム動作特性の解析

PPSのアーキテクチャを決定する目的には、上述の機能分散の特徴が、システム動作特性にどのような影響を及ぼすかを明確にしておく必要がある。この観点から、専用のソフトウェア・シミュレータ(SAP)を用いてプロセッサの処理動作特性およびプロセッサ間通信特性の解析を行なう。

#### 3.1 シミュレーションシステム

SAPシステムは、現行のOSおよびジョブをPPS上で実行させる場合を想定してシミュレーションを行ない、現行システムとの比較の上でPPSの特性データを収集するものである。図2にSAPシステムの構成を示す。[7]

#### 3.2 システムのモデル化

SAPシステムでは、PPSを以下のようモデル化した。[5,6]

##### ハードウェアモデル

ハードウェアのモデル化にあたっては、正常状態におけるシステムの動作特性を明確にすることを主要目的とし、一般性を失わない範囲で単純化を行なう。

(1) PMSXは、主にシステム再構成などの場合に使用される為着脱し、各PMEMは特定プロセッサに接続して動的切替えは行なわないとしか。

(2) 共用メモリについては、基本的アクセス情報のみを収集する事にし、主メモリバンク構成としか。競合等については、別途ローカルなシミュレーションを行なう事としか。

(3) プロセッサ間通信装置は単一バス構成とし、複数バス等については、ローカルなシミュレーション結果をパラメータとして導入するにしか。

##### OSモデル

ソフトウェアの互換性とこの観点から、PPS用OSが実現する機能は、現システムの持つ機能を包含する必要があり、PPS用OSの機能を現行OSの機能とほぼ等しいとしか。次に、OS機能を推うプログラムの最小単位をセクションと考え、現行OSのセクションを上述のクラス対応に分類し、セクションの統合体としてサフトOSを定義し、これをサフトOSの統合体として、PPS用OSの近似モデルを作成し、

OS機能は専用機能と共通機能を分類され、専用機能は特定のサフトOSのみ割当てられ、共通機能は全てのサフトOSに割当てられる。

##### 入力ジョブモデル

OSのモデルが現行OSを前提としている点を活用し、プログラムの処理動作モデルを、現行OSのセクションの処理系列として理物し、このモデルは現行システム上でテスト・トレースによって自動的に作成され、モデル化が容易となる。

また、入力ジョブの集合は、上記プログラムの到着分布を考慮してミックスで表現し、

#### 3.3 シミュレーション結果とその解析

プログラムモデルとしては、現行システムベンチマーク・ジョブを用いる。本稿では、典型的な2つのモデルから得られる結果を中心叙述

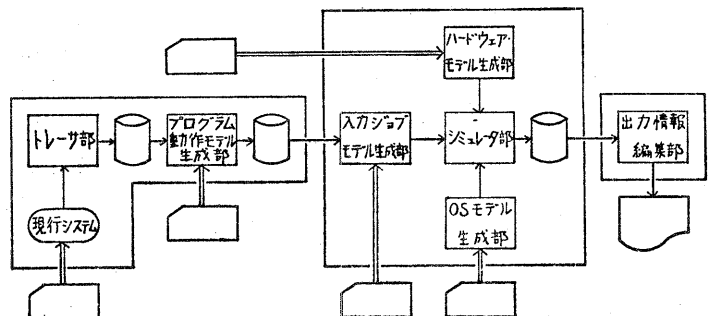


図2 SAPシステムの構成

べる。

モデルSは、端末から入力データを読み込み、マトリックス計算の結果を端末に出力するプログラムのモデルである。総走行ステップ数は365 kstepであり、11回の端末入出力、21回のファイル入出力を含む。

モデルCは、FORTRAN コンパイラのモデルである。総走行ステップ数は530 kstepであり、5回の端末入出力、49回のファイル入出力を含む。

70セッサ間通信特性 [9]

一度70セッサに入ってから、次の70セッサ通信を行おうとする平均走行ステップ数を図3に示す。若70セッサからの一通信当りの平均メッセージ長を図4に示す。

上記の結果から次の事が指摘できる。

- (1) 70セッサは約1.3 kstep 走行毎に70セッサ間通信を要求し、通信の頻度は比較的高い。
- (2) 平均メッセージ長は約30 Byteとほり、メッセージ転送時間は70セッサの自由走行時間に比べて短い。

70セッサの負荷特性 [8]

若70セッサでの全走行ステップ数の分布を図5に示す。これは若70セッサの負荷を表わしていると考えられ、次の事が指摘できる。

- (1) 若70セッサ・クラスを独立とするに足る負荷がある。しかし、本データは単一ジョブの解析結果であり、RPU, SPU, MPUの負荷は極めて小さくはっている。また、EXEC 70セッサの全走行ステップ数はJPUの全走行ステップ数にほぼ等しいため、JPUクラスは若EXECクラスの3~5倍の能力を持つ必要がある。

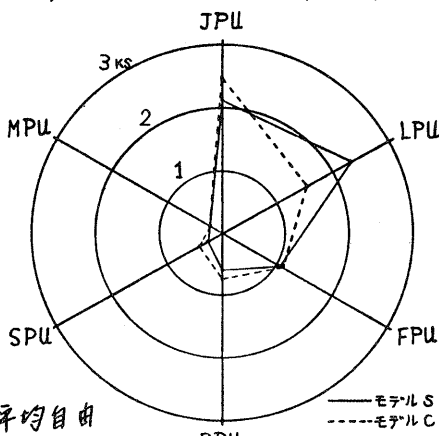


図3 平均自由走行ステップ数

がある。

(2) 若70セッサ・クラスの負荷分布は、ユーザプログラムの種類によって大巾に変化し、ジョブの特性によって負荷不平衡が生ずる可能性がある。この為、70セッサ間の負荷不

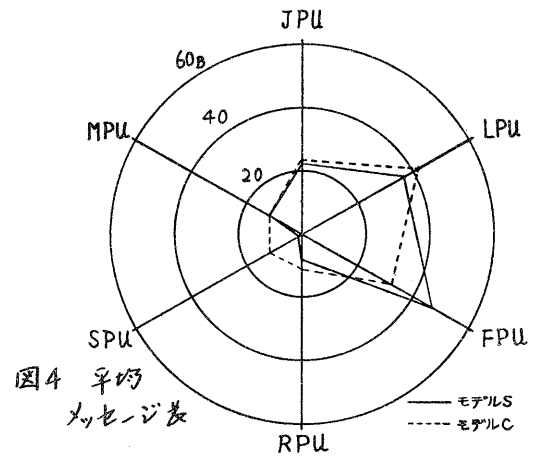


図4 平均メッセージ長

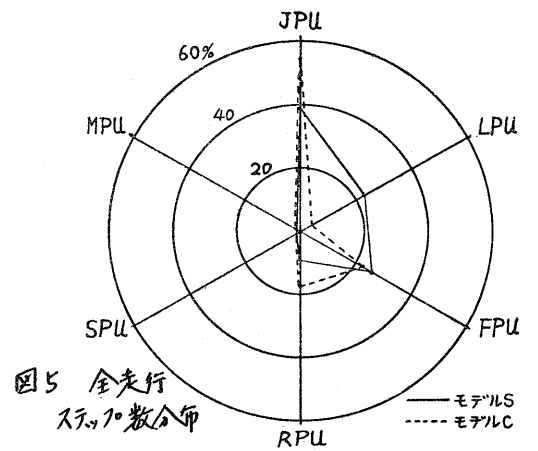
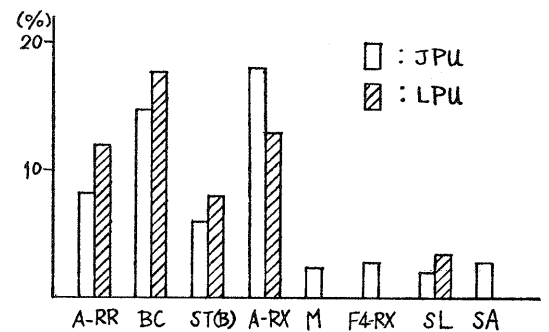


図5 全走行ステップ数分布



A-RR: RRタイプの演算 SL: 論理シフト命令  
A-RX: RXタイプの演算 SA: 算術シフト命令  
F4-RX: RXタイプの浮動小数点演算(4B)

図6 命令使用頻度

解を解消する機構が必要となる。  
 その他の主要な特性 [7, 8]

モジュールSについて、JPU, LPUでの命令使用頻度を図6に示す。図から分かるように、JPUでは使用されるが、EXECプロセッサでは全く使われない命令が存在し、JPUはEXECプロセッサとハードウェアの異なる構成を必要とする。しかし、EXECプロセッサ間ではこのような相異は見られず、均質のハードウェア構成が可能である。

その他、プロセッサ-メモリアン結合装置への要求条件を導く生データ、および、PPS用OSの構成条件を与える各プロセッサクラスでのセクション使用頻度等の情報が得られている。

#### 4 アーキテクチャ実現の基本技術

ここで得られる解析結果から明らかになるように、本システムを実現するためには、PPS特有のハードウェア基本技術を開発する必要がある。PPS-Rで開発された技術は以下の各項である。

- (1) プロセッサの動的変身機能: プロセッサの機能専用化と負荷不均衡の平滑化を同時に解決する手法として、プロセッサ機能の動的変身が可能なるプロセッサ構成を採用する。
- (2) プロセッサ間結合機能: 解析結果から得られる変信特性に従い、最適結合方式として単一バスによるデータ転送方式を採用する。
- (3) プロセッサ-メモリアン結合機能: プロセッサとメモリアン(MEM)間は、動的に開閉可能な交差から成るスイッチマトリックス構成とし、プロセッサクラスのアクセス要求特性に従って、アクセスの優先度を設定する。

#### 4.1 プロセッサ機能の動的変身方式 [13, 14]

PPSにおいて、μPing技術は次のような目的で用いられる。

- (1) プロセッサ基本機能の充実と高級化: BCMに格納される共通の基本機能として、基本命令セット以外に、動データ処理、プロセッサ間交信処理、同期基本機能などを実現する。
- (2) プロセッサの専用化: FCMに格納される各プロセッサの専用機能と、制御プログラムのファームウェア化も含めて実現する。
- (3) 専用機能の動的入替え: プロセッサ機能の

分散、専用化が行はれられ結果、3に示したように、システムに投入されるジョブの種類により各プロセッサに与えられる負荷の分布が大幅に異なる。負荷分布を時系列情報として観測すると、周期的な負荷不均衡状態が發生する。この解決策として、プロセッサの専用化機能を担うμPを動的に入替え、過負荷プロセッサを変身し、その負荷を処理する。

上述のように、PPSにおいては、プロセッサ機能の動的入替えが重要な役割を果たし、これを実現する動的μPing技術が必要となる。

#### 4.1.1 動的μPing方式

PPS-Rでは、BCMとROMで、FCMとWCSで構成される。

μPの動的入替え手段として、任意の時点で、動的にμPを交換する。又、ソフトウェアおよびμPの双方から変更可能とする為、メモリアンのアクセスと同じ手順で入替え可能とされる。しかし、ソフトウェアとμP間の干渉を減少させる為、MEMとCSを別空間とされる。またCS空間を動的に拡張し、μPの作動を容易にするため、MEM上のμPを直接実行可能とされる。

以上の点を考慮して、CSとMEM間に空間共用Windowを設ける。空間が共用されるとは、図7に示す2組のパラメータより、

- (1) MEMアドレスがCSアドレスに写像されること、および、
- (2) CSアドレスがMEMアドレスに写像されること。

である\* 即ち、メモリアン空間内のオペランドアドレス(M)が、

$$\alpha_S^M \leq M \leq \alpha_E^M$$

である場合には、CS空間上のアドレス(M')中B番目のブロックの内容に格納される。但し、M', Bは次式により決定される。

$$M - \alpha_S^M = (M' - \alpha_S^C) \cdot 5 + B$$

逆にCS空間内の次に実行すべきμIアドレス(m)が、

$$\beta_S^C \leq m \leq \beta_E^C$$

\* MEMの1語は16ビットでWCSの1語は80ビットであるため、WCSの1語を16ビット単位でブロックに分割し、WCSへのアクセスはこのブロック単位で行なう。

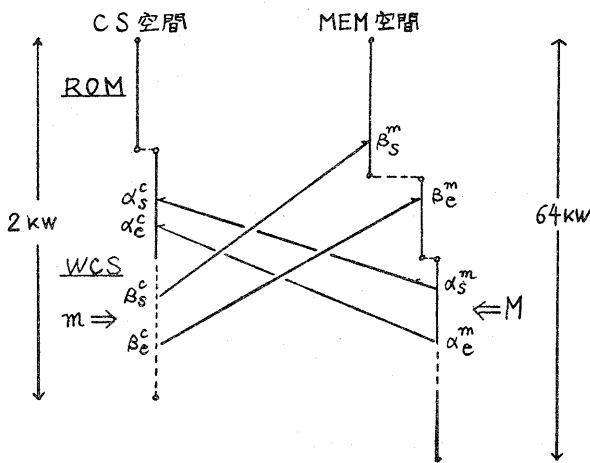


図7 空間共用パラメータの関係

の場合には、MEM空間上のアドレス(m')以降は  
 諾がフェッチされ、 $\mu I$ として実行される。但  
 し、mは次式により決定される。

$$m' = (m - \beta_s^c) \cdot 5 + \beta_s^m$$

上記空間共用の可否は、OSおよびイニエアル  
 スイッチで制御され、 $\mu P$ の効果的に保護される。

#### 4.1.2 空間共用の実現機構

CSとMEM間の空間共用を実現し、WCS中の動的  
 $\mu P$ ingを可能とする機構は、WCSに装着する  
 拡張制御ユニット(ECS)、CSとMEM間の写像を  
 制御するアドレス空間制御ユニット(ASC)、さ  
 らにECSとASC間を結ぶ $\mu P$ バス(MP-Bus)  
 から成る。ECSおよびASCの論理構成を図8に  
 示す。

WCSへアクセスする際、M-Bus上のMEMアド  
 レスを空間共用Window内にセットすると、A  
 SC中のアドレス変換装置(M $\rightarrow$  $\mu$ )が作動しM  
 EMへアクセスし、WCSへアクセスする。  
 このアクセス動作はROM中の基本 $\mu P$ により制  
 御される。

MEM上の $\mu P$ を実行する場合は、次に実行すべ  
 き $\mu I$ アドレスが空間共用Window内に入ると、  
 ASC中のアドレス変換装置( $\mu$  $\rightarrow$ M)が作動し、  
 MEM上の $\mu I$ がフェッチ、実行される。

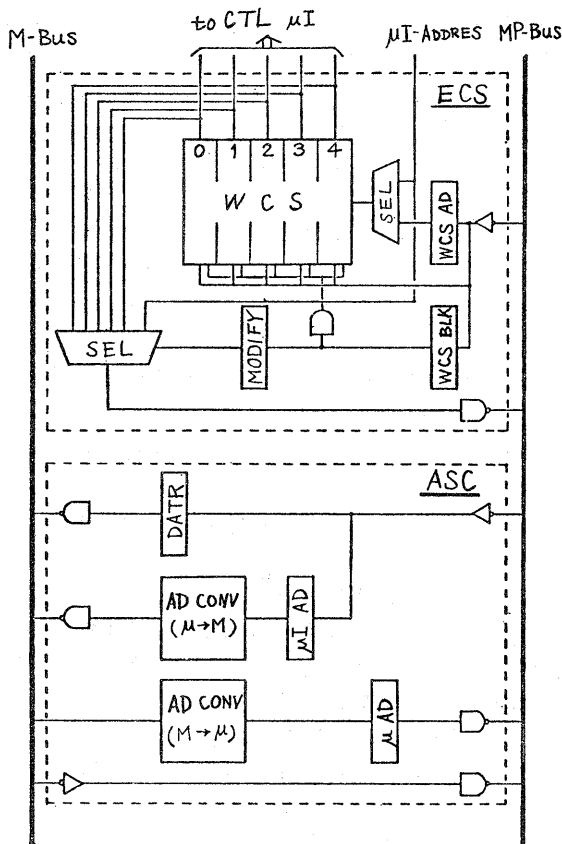


図8 空間共用機構の論理構成

#### 4.2 プロセッサ間結合方式

共通バスを介して複数のプロセッサ間でデー  
 タを転送する場合、大部分のデータ転送方式を  
 その送受信方式によって表1のように分類し、  
 その適用条件を定める事ができる。

そのシミュレーション結果から、PPS-Rのデー  
 タ転送方式としてSII-RII方式を採用し、以  
 下の交信機構を与える。

##### 4.2.1 交信装置(IPSX)の構成 [12]

IPSXは、図9に示す各部分から成る。

(1) 転送路とアービタ: 転送路として単一バス  
 方式、バスアービタとしてBAP方式を採用し  
 た。単一バス(Q-bus)は、16ビット中であ  
 り、非同期確認方式でデータを転送する。B  
 AP方式は、バス使用権を示す制御信号をCPU  
 間で環状に巡回させる方式であり、その動作  
 特性は求められている。[10, 11]

(2) 交信制御装置(IPU): IPUは各プロセッサ

毎々分散配置され、受信動作を制御する。Q-busの使用効率を高めるための、各IPUは送受信それぞれ32語のバッファを持ち送受信動作が不可能である。

MEM上の送信データは、メッセージ毎々送信バッファ内に読み出され、Q-busを介して相手IPUの受信バッファへ転送される。送信命令の終結は、転送動作終了後である。

受信バッファ内のデータは、受信側MEMに設定されるサイクリックバッファ上に書き込まれ、受信CPUに受渡される。

#### 4.2.2 サイクリックバッファ受信方式 [17]

前述のRⅡ方式を実現するため、PPS-Rでは、サイクリックバッファ受信方式を採用した。本方式は、データ受信に伴う割込みを極力減少させ、非同期的な可変長のデータをMEM内に効率良く蓄えることを意図している。

サイクリックバッファ(CB)は、MEM上の連続したアドレスに置かれ、その上下限値はそれぞれIPU内のULR, LLRで管理される。未処理データの始めと終りのアドレスは、それぞれIPU内のTMR, BMRで示される。IPUは受信データをBMRで示すアドレスから格納し、BMRを更新する。CPUはTMRから始まるデータを処理し、TMRを更新する。これらの4個のレジスタは、いずれもCPUからR/W可能であり、TMRとBMRは、その値がULRの値を越えるとハードウェア的にLLRの値がセットされる。

CPUへ割込みが発生するのは、

- (1) CBがすべて使用中となった時、
  - (2) 転送中、データエラーが発生した時、
  - (3) 緊急通信を受信した時、
  - (4) その他のIPU障害時、
- のみであり、割込み発生頻度を大巾に減少した。

#### 4.2.3 可変長バースト転送方式

プロセッサ間で転送されるメッセージ数は、3で示されるように、平均15語(30バイト)であり、32語以下のデータを持つ送信頻度は全体の92%(モデルC)とされる。これらの結果から、Q-bus上の転送方式として、以下の可変長バースト転送方式を採用した。

- (1) 相手IPUアドレスと共に、メッセージの転

表1. データ転送方式の分類

受信方式 送信方式	RⅠ方式 (受信割込み)	RⅡ方式 (ルックイン)
SⅠ方式 (チャネル起動)	送信頻度 小 メッセージ長 大	送信頻度 大 メッセージ長 大
SⅡ方式 (MOVE命令)	送信頻度 小 メッセージ長 小	送信頻度 大 メッセージ長 小

送語数も受信側へ送る。1メッセージの最大転送語数は、32語である。

- (2) 1メッセージの転送は、2.5MWord/secの転送速度でバースト転送され、その間、他IPUはQ-busを使用できない。

#### 4.2.4 緊急通信機能

主として、プロセッサクラスの障害通信手段として、PPS-Rでは任意の相手プロセッサと緊急通信不能な機能を具備している。緊急通信は特定アドレスを送る事により開始され、他プロセッサは全て緊急通信受信状態に設定される。緊急通信で転送されるデータは、メッセージ単位で受信割込みが発生する。

#### 4.3 プロセッサ-メモリ結合方式

PPS-Rのプロセッサ-メモリ間結合装置(JMSX)は、スイッチトリックス方式を採用し、以下の特徴を持つ。[16]

- (1) 6台のプロセッサと8台のメモリユニットをスイッチトリックスで結合し、メモリアクセス毎々動的に交差を閉鎖可能である。
- (2) 各プロセッサのメモリアクセス特性を考慮して、メモリ競合解消の優先度を設定した。
- (3) JMSXの各交差を、LPUおよびMPUから動的に閉鎖可能である。
- (4) 論理アドレスから物理アドレスへの変換をメモリバンク対応に動的に変換可能である。
- (5) 各メモリバンク対応に、交差の連続回数に計数可能である。

標準構成下のPPS-Rでは、周辺機器はFPU, RPU, LPUにそれぞれ接続される。これら各クラスのメモリアクセス特性を考慮して、回IDを示す競合解消手順を設定した。即ち、あるプロセッサが現在の番目のメモリユニット(JMi)に

アクセス要求が出ると、メモリ制御装置(MCU)は、FPU, RPU, LPUの順に要求プロセスを調べ、これら3台とも要求を出していない時に限り、残り3台のCPUを調べる。残り3台から1台を送る方式は、先着優先方式に従う。

5 PPS-Rシステム [4, 15~20]

開発目的

PPS-Rシステムは、2で述べた機能分散型の妥当性を、ハードウェア、ファームウェア、ソフトウェアの各観点から総合的に検証する

事と目的として開発された。特にハードウェアシステムの開発は、3のシミュレーションによる解析結果に従ってアーキテクチャ実現の基本的技術を確認し、その構成を定める事と目的とした。

システム諸元

PPS-Rハードウェアシステムの構成と主たる諸元を図1および表2に示す。

システムの特徴と現状

PPS-Rのハードウェア上の特徴は、以下の各点である。

- (1) CPUは、RAMおよびコアと各々主メモリ域制御メモリ域としても共用可能であり、この

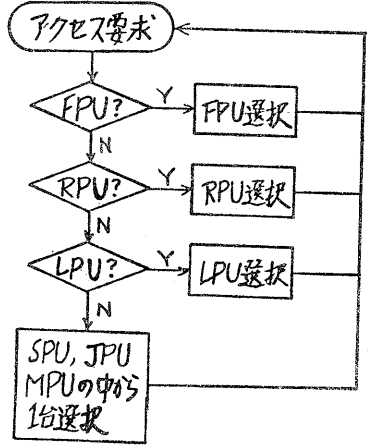
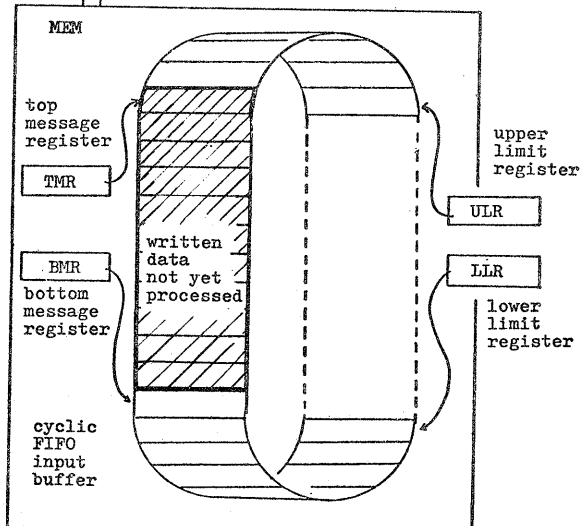
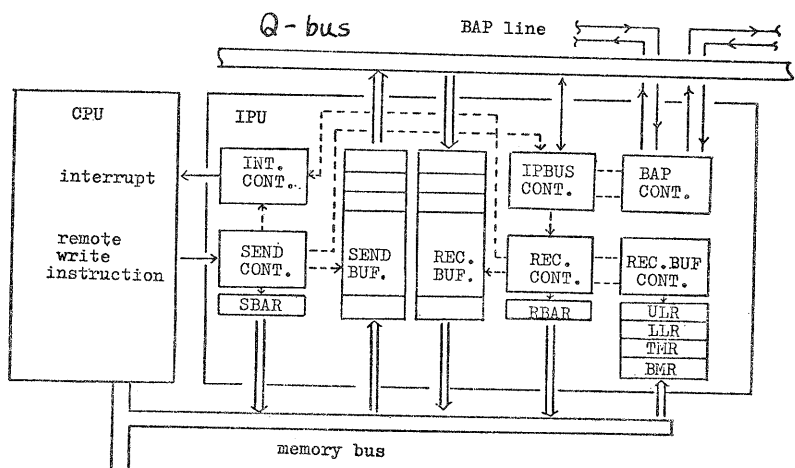


図9 10プロセス間通信装置の構成

図10 JMSXメモリ競合解消手順



共用域は動的に可変である。

- (2) プロセッサ間通信処理において、IPU-CPU間の情報授受は、主メモリ上のサイクルリックバッファを介して非同期的に行われ、CPUへの割込み頻度を減少させている。
- (3) プロセッサ間プロトコルに正常、緊急の2系列を設け、障害時の処理、再構成を容易にする手段で行はせる。
- (4) CPU-JMSXに強力な競合解消機構を設ける。
- (5) システム全体の集中制御・監視機能を強化し、マイルCPUのデバク機構を充実させる。

現在、上記ハードウェア上で稼動するファームウェア、OSを開発中である。[20]

6 あとがき

機能分散型システムを構成するための、機能分散の考え方を示し、それによって設定された構成要素およびシステムの論理構成を示した。更し、このような構成を持つPPSの動作特性をシミュレーションにより解析し、この結果に基づいて開発したパイロットモデルとその構成技法を示した。

御討論後いかに基一壘諸氏に深謝する。

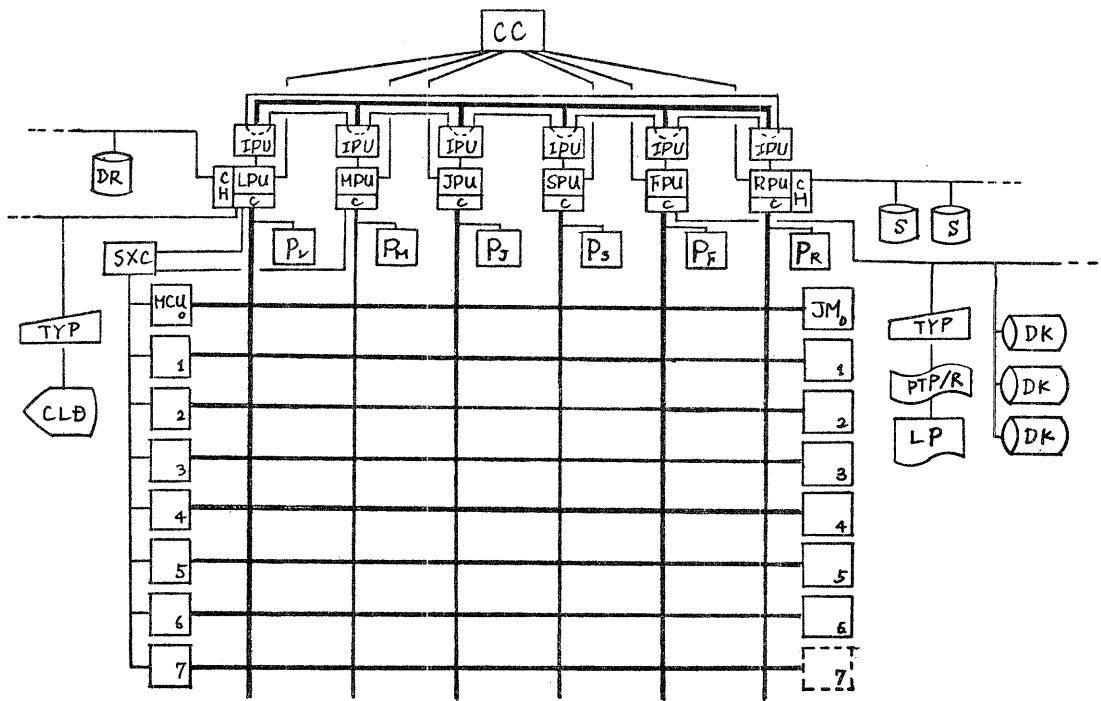


図11 PPS-Rのハードウェア構成

表2 PPS-Rの代表諸元

メモリ系			速度	語中	容量	結合系	IPU	2.5 MWord/sec
	制御メモリ(C)	ROM	240 msec	80 bit	256 w/PU		周辺系	BAP
主メモリ	RAM	480 msec	80 bit	1792 w/PU	ドラム(DR)	256KW x 3		
	専用(P <sub>z</sub> )	1.2 μsec	16 bit	8 kw/PU	ディスク(DK)	150kw x 3		
共用(JM)	1.2 μsec	16 bit	7x8 kw		タイフ(TYP)	3		
スワップメモリ(S)		8.9 μsec	16 bit	2x256 kw		その他	PTP/P, LP, CLM	

## 参考文献

- [1] 小特集：コンピュータコンプレックス、情報処理, 15, 7, 1974.
- [2] *Distributed-Function Computer Architectures, Computer, Special issue*, pp.15-37, 1974.
- [3] 村上, 西川, 佐藤, 岡田: ポリプロセッサ・システムの一構成法, 昭和49信学全大, S1-6.
- [4] Murakami, Nishikawa and Sato: *Poly-Processor System Analysis and Design, Proc. of 4th Annual Symp. on Computer Architecture, 1977.*
- [5] 小山, 佐藤: ポリプロセッサ・システムのソフトウェアシミュレータ, 信学技報, EC75-61.
- [6] 長谷川, 日比野: OSの動作解析システム, 信学技報, EC75-60.
- [7] 佐藤, 日比野, 小山, 長谷川: ポリプロセッサ・システムのソフトウェアシミュレーション, 昭和51信学会総全大, 1287.
- [8] 長谷川: シミュレーションによるPPS処理特性の解析, 昭和51情処大会, 107.
- [9] 小山: シミュレーションによるPPSのプロセッサ間交信特性の解析, 昭和51情処大会, 194.
- [10] 加藤, 西川: 分散処理システムにおけるハードウェア資源結合方式の解析的検討, 信学技報, EC75-59.
- [11] 加藤, 西川: 分散処理システムにおけるハードウェア結合方式の検討, 昭和51信学総全大, 1286.
- [12] 加藤, 後藤, 岡田, 西川: ポリプロセッサ実験システム(ESP)のハードウェア構成, 信学技報, EC76-11.
- [13] 村上, 佐藤, 小山, 長谷川: 動的マイクロプログラム方式評価システムの一構成法, 昭和51信学会総全大, 1256.
- [14] 村上, 佐藤, 長谷川: 動的マイクロプログラム実験システム, 信学技報, EC76-69.
- [15] 村上, 西川, 佐藤: 機能分散型システムの構成法, 昭和52信学会総全大, 1260.
- [16] 加藤, 小川, 西川: 機能分散型システムにおけるプロセッサ-メモリ結合装置の一構成法, 昭和52信学会総全大, 1261.
- [17] 小川, 加藤, 村上: 機能分散型システムにおけるプロセッサ間交信のCyclic Buffer受信方式, 昭和52信学会総全大, 1263.
- [18] 西川, 佐藤, 小川: 機能分散型システムにおけるIPL機構, 昭和52信学会総全大, 1262.
- [19] 佐藤, 加藤, 村上: 機能分散型システムにおけるデバグ機構, 昭和52信学会総全大, 1295.
- [20] 村上, 尾内: 機能分散型システムにおける言語処理系の一構成法, 昭和52信学会総全大, 1294.