マルチプロセッサシステム MICS-Ⅱの システムコントロール

A SYSTEM CONTROL FOR THE MULTI PROCESSOR SYSTEM-MICS-Ⅱ

山崎 竹視　大森 健児　小池 誠彦　大宮 哲夫　出来 哲史郎
Takemi YAMAZAKI Kenji OHMORI Nobuhiko KOIKE Tetsuo OHMIYA Tetsushiro DEKI
日本電気株式会社 中央研究所
Central Research Laboratories, Nippon Electric CO., Ltd.

1. はじめに

MICS-Ⅱ はユーザの多様な処理要求を満足し得るプロセッサシステムを提供すべく設計されたLSI指向のマルチプロセッサシステムである。

我々は既にマルチマイクロプロセッサシステムMICSを開発しているが，MICS-Ⅱではこの経験を生かし，システムの規模，特性を要求に応じて柔軟に変更できるものとすることを主眼としている。

ユーザサイドから見たMICS-Ⅱの特徴は，同時に多数の利用者にサービスを行い，それらの利用者の種々の処理（計算，トランザクション処理，装置制御等）が同時に行われることである。従ってMICS-Ⅱ は従来の計算処理中心のTSSに対して多用途のマルチユーザサービスと見ることができる。

MICS-Ⅱの特徴は次の5点に要約される。

(1) ハードウェア的に均質なプロセッサ複合体
(2) プログラム実行とシステム管理の完全分離
(3) ユーザに対するプログラム実行プロセッサの完全解放
(4) 共用メモリを用いたページング方式仮想メモリシステム，I/O 装置を各プロセッサが共用するグローバルI/Oシステムのサポート
(5) マイクロプロセッサを指向したハードウェアプロセッサ

以下にMICSⅡのアーキテクチャ及びシステムコントロールについて述べる。

2. ハードウェア

MICS-Ⅱは図1に示すように最大8組までのプロセッサモジュールを中核として構成される。これらのプロセッサモジュールはコントロールバス（C-Bus），メモリバス（M-Bus）及びグローバルI/Oバス（GIO-Bus）の3つの共有バスによって結合され，プロセッサ相互間の通信，共用メモリ，共用I/O装置のアクセスなどによって有機的なつながりを保っている。

各プロセッサモジュールはユーザプログラム実行用のユーザプロセッサ（UP），コントロールプログラム実行用のコントロールプロセッサ（CP）を含んでいる。図中CSMはCPからUPのコントロールを行い状態センスするためのコンソールモジュール，IPP はソフトウェアレベルでCPとUPとのやりとりを行うレジスタ群・インタプロセッサポート，DMA はCPがUPのアドレス空間にアクセスするためのダイレクトメモリアクセスモジュールである。プロセッサモジュールと外界とは，プロセッサモジュール内のロジカルアドレスを実アドレスに変換するアドレストランスレータ（AT），UPのI/Oバスを GIO-Bus に結合するI/Oポート（IOP）及びプロセッサモジュール間通信を行うコミュニケーションモジュール（COMM）によって結ばれている。また各プロセッサモジュールは，ローカルメモリを独自に持ち，必要に応じてローカルI/Oデバイスを接続することもできる。

3. システムの概要

MICS-Ⅱのアーキテクチャの最大の特徴はハード／ソフト両面でプログラム実行と管理とを分離した点にある。従来のオペレーティングシステムに囲まれた内部でユーザプログラムを実行させるという概念は、ユーザがI/O命令を使ったり，別のオペレーティングシステムの下で開発したプログラムを使ったりできないという点で，多数のユーザに多種多様な処理を許すというMICS-Ⅱの設計思想と相容れない。

MICS-Ⅱではユーザプログラムを実行する部分とシステム全体の管理をも含めた管理部分との独立性を高め，ユーザプログラムに対する管理側からの制約を極力ヒリのぞき，あらゆる種類のユーザプログラムの実行を可能とした。

3.1 処理システム／処理モジュール

MICS-Ⅱではそのハードウェアの持つ能力を十分に発揮させる為，ハードウェアに立脚した処理モジュール，ユーザの視点に立脚した処理システムという概念を導入し，これを管理対象としている。

処理モジュールはハードウェアのプロセッサモジュールとメモリやI/O装置を得て動く，MICS-Ⅱ管理下の仮想的なプロセッサシステムであって，独自のアドレス空間／I/Oデバイス空間を有し，一般にいうプロセスに対応する。処理システムは、いくつかの処理モジュールから成りユーザのジョブステップがこれに対応するものと考えてよい。処理モジュールをプロセスと区別したのは，MICS-Ⅱではユーザが独自のマルチプログラム管理や割込管理を行うことを許している点と，I/O装置割当て後は物理的にI/O装置の制御を行うことを許していることによる。

処理モジュールはMICS-Ⅱの提供する処理モジュール間通信を介して相互の連絡を行い，また同一処理システム内にあってはアドレス空間を通じての連絡も可能である。

処理モジュールが実行時に割当てられるハードウェアリソースのうち，ユーザプロセッサ、ローカルメモリ（と必要に応じてローカルI/Oデバイス）を基本モジュール，メインメモリとグローバルI/Oデバイスを拡張部分と呼ぶ。

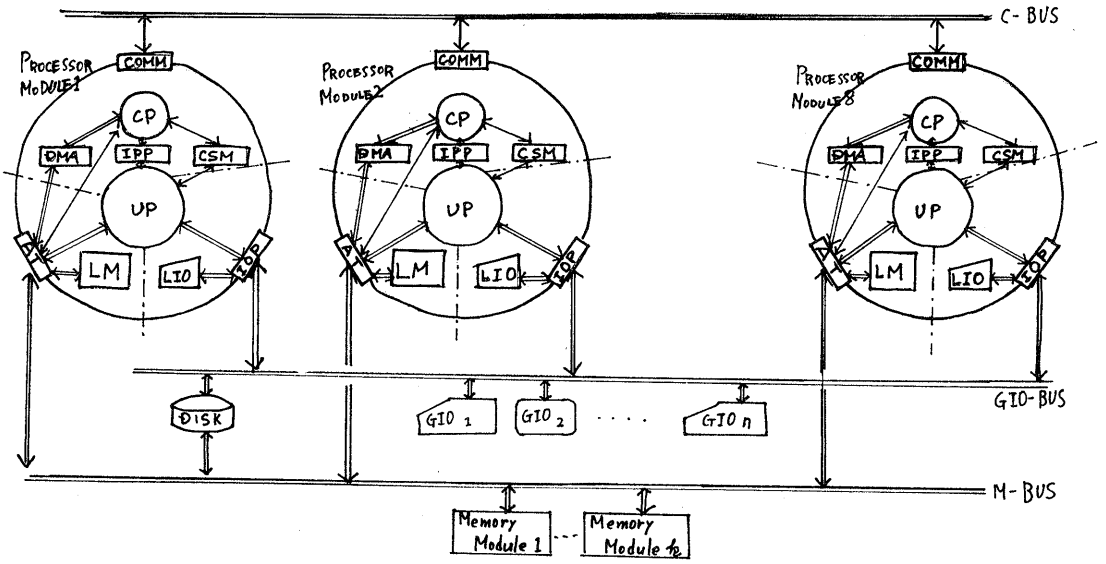


fig - 1

Hard ware Configulation of MICS-Ⅱ

3.2 制御形態

MICS-Ⅱは，ハードウェアの規模の変更を容易にし，システムの信頼性を向上させるためプロセッサモジュールを基本とするモジュール構成をとっている。

システムコントロール方式においても，上記の特性を十分考慮した上で，できるだけ各モジュール毎に完結した形をとるようになっている。たとえば，ユーザプロセッサの起す各種のイベントは必ずしもシステム全体にかかわるものとは限らないので，これらは各処理モジュール内で管理している。

他方，リソースに関しては，処理モジュール毎にリソース使用状況を保持し，割当てに際し相互に通知しあうのでは，処理モジュール間のやりとりが激増してオーバーヘッドの増大はまぬがれない。そこでリソース管理を行う処理モジュールを設け，集中管理を行っている。

4. 処理システム

MICS-Ⅱ上の処理システムはユーザが自由に設定するユーザ処理システムとMICS-Ⅱ管理のためのマスタ処理システムに分けられる。

4.1 マスタ処理システム

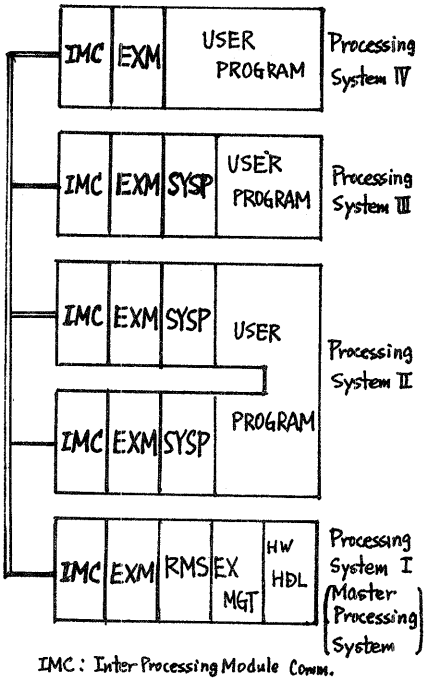
マスタ処理システムは，ユーザ処理システムの生成，消滅とこれに伴うリソースの管理，システムコンソールを介してのオペレータインタフェース等を担当する単一処理モジュールで，稼動時には手動で任意のプロセッサモジュールに割当てられる。マスタ処理システムはその内部構成が他の処理システムと多少異り，コントロールプロセッサに於ても後に述べる実行管理モジュール以外にリソース管理の一部が実行される。マスタ処理システムに割当てられたプロセッサモジュールをシステムプロセッサモジュールと呼ぶ。

4.2 ユーザ処理システム

ユーザ処理システムは，システムコンソールからのユーザ／オペレータの指示によるジョブエントリによって生成され，ジョブデリートによって消滅するが，この間マスタ処理システムによって必要なプロセッサ，I/Oデバイス，メモリの割当てを受ける。

すなわちロジカルには，ユーザ処理システムには処理モジュール毎にExecution Monitor(EXM), Inter Processing Module Communication(IMC), 及びユーザプログラムとMICS-Ⅱの仲立ちとなるSystem Primitives (SYSP)が付加される。これによって処理モジュールはその内部にイベント管理能力を持つことになる。図2にこの様子が示されている。図中処理システムⅠはマスタ処理システムである。処理システムⅡは2つの処理モジュールを含んでいる。また処理システムⅣはSYSPを持たないが，これは処理システムⅣがユーザ独自のコントロールの下で動きMICS-Ⅱに対してソフトウェア的な処理依頼を行わないことを示している。

ハードウェア的には処理モジュールはプロセッサモジュール上に投影される。ユーザプログラムとSYSPはユーザプロセッサで実行され，EXMとIMCはコントロールメモリに実装されてコントロールプロセッサで実行される。



IMC: Inter Processing Module Comm.
EXM: Execution Monitor
SYSP: System Primitives
RMS: Resource Management System
EX MGT: Execution Management System
HW HDL: Hardware Handlers

frg.-2
Processing System

4.3 リソース割当て

基本モジュール割当てはスタティックに行われ，処理システム消滅まで変更されることはない。拡張部分に関してはメモリは仮想メモリの形で完全に動的な割当てが行われており，またグローバルI/O装置についてはオペレータが割当てに介入する場合があるがユーザプログラム中の定義によってほとんど自動割当てが成されている。

こうしたリソース割当て方式を実現するため処理モジュール内でユーザの扱うメモリアドレス及びI/O装置番号，処理モジュール名はすべてロジカルアドレスとして扱われ，メモリアドレスについてはアドレストランスレータ(AT)がI/O装置番号についてはI/Oアドレストランスレータが，また処理モジュール名についてはコントロールプロセッサとコミュニケーションモジュールが変換を担当している。

各リソースの割当てはマスタ処理システムが集中的に行うが，一度割当てられたリソースについてのアドレス変換は各処理モジュール毎に実行時に行われ，マスタ処理システムの負担の軽減を図っている。

5. 仮想メモリシステム

MICS-IIの仮想メモリシステムは，実装メモリスペースに制約されない広いアドレス空間をユーザに与えるという目的と共に，もう一つの重要な目的を持つ。それは処理モジュールに対するメモリリソースの動的管理である。すなわち仮想メモリ空間の実スペースの形でメインメモリのページを各処理モジュールに与えており，バックアップメモリに書出したり，ここから読み込んだりするいわゆるページフォールトは，メモリ管理の補助的な手段と考えている。従ってハードウェアプロセッサのアドレス能力の拡張は行っていない。

5.1 仮想空間

MICS-IIでは処理モジュール毎に異るアドレス空間を有するページング方式仮想メモリシステムをサポートしている。各処理モジュールは基本的には機械語命令のアドレス能力一杯の32KWのアドレス空間を有し，基本モジュールで与えるローカルメモリの容量を超える部分についてはメインメモリ及び磁気ディスクから構成される仮想空間にマッピングされる。ページサイズ

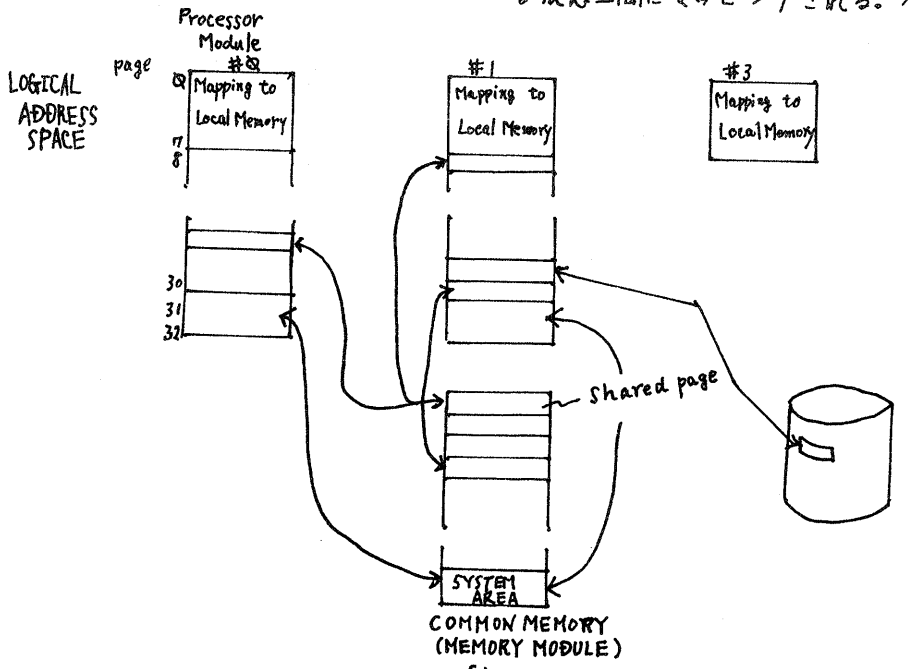


fig-3

Memory Space Allocation and Sharing of Physical Memory Space

1kW 固定であるので，0~7ページはローカルメモリに，8~31ページは共通メモリ（及びバックアップメモリ）に対応する。従って8kW以下のメモリで動くユーザモジュールはメモリに関しては基本モジュール内で完結する形となる。

5.2 メモリ割当てとアドレス変換

ユーザプロセッサの用いるロジカルアドレスはプロセッサモジュール毎にアドレストランスレータ(AT)によって実アドレスに変換される。この時実ページがメインメモリ中に存在しているか否か，書込禁止ではないか等のチェックを受けページフォールトであれば，コントロールプロセッサによって検出され，マスタ処理システムにページフォールト通知が行われてメモリの割当てとページロードが要求されることになる。

メモリスペースの割当に伴うリプレースメントのアルゴリズムは，メインメモリの各ページ毎に設けられたハードウェアのアクセスタイマによる擬似ワーキングセット方式であり，一定時間内にアクセスのないページを置換対象として選び出し，マスタ処理システムがスワップアウト/インを行っている。この方式ではタイマの時定数の調整によって十分効率のよいリプレースメントを行うことができる。

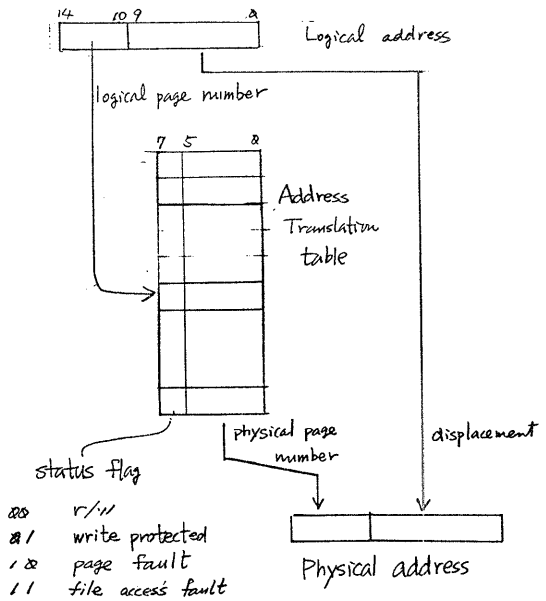


Fig-4
Address translation

5.3 ページの共有

処理システム内では，複数の処理モジュール間でページ単位でメモリ空間を共用することができる。

マスタ処理システムはMICS-II上のすべての処理システム/処理モジュールの使用リソースリストを有するが，このリストにページ共有が指示されるとページアロケート，デアロケート時に共有の相手方に対しても正しい通知を行い，異るアドレススペースに対し同一フィジカルスペースを与えることができる。この機能を用いて共通データ，共通プロシジャによる使用メモリリソースの低減が行え，処理モジュール相互間の大量データのやりとり，ユーザ独自のプログラム同期操作を行うことも可能である。

MICS-IIではすべての処理モジュールの30，31ページをシステムエリアとして全処理モジュールの共有ページとし，ここをSYSP用のエリアとしている。

6. グローバル I/O システム

仮想メモリシステムでロジカルなアドレス空間に対してフィジカルなメモリを割当て，実行時にアドレス変換を行ってメモリリソース使用の効率化を図ったと同様に，グローバルI/Oシステムでは，ロジカルなI/O装置番号に対し実際のI/O装置の割当を行い実行時に装置番号変換を行うという方式でI/Oリソース使用の効率化を図っている。

中ないし小規模システムを複数のユーザが使用したり，並列処理を行ったりする場合にはI/O装置の種類と数，そして制御方法が問題となる。疎結合のコンピュータネットワークの場合は，各コンピュータ毎に必要なI/O装置を設けたりユーザの必要とするI/O装置を有するプロセッサにソフトウェアレベルでI/Oオペレーションを依頼する方法がとられ，密結合のマルチプロセッサシステムではI/O装置のコントロールを一手に行うI/Oプロセッサを置くのが一般的である。しかし，いずれの方法もユーザに全命令を解放し，多様なサービスを与えるというMICS-IIの基本方針と一致しない。そこでMICS-IIではグローバルI/Oバスという共通I/Oバスに I/O装置を接続しすべてのプロセッサモジュー

ルからこれにアクセス可能とするという方法を採っている。また、この方式によって、プログラムに対し何等変更をすることなく、使用入出力装置を変更することも可能となる。

6.1 入出力装置の統合

MICS-Ⅱの管理下にあるすべての入出力インタフェースはテレタイプに代表される標準入出力装置とファイル装置の2種類に統合されている。標準入出力装置の命令セットのうち、通常の入出力に要する命令から成るサブセット(標準入出力命令)は全ての標準入出力装置に対して共通化されており、ユーザは標準入出力命令を使用する限り入出力装置の個性を意識する必要がない(図5)

もちろん装置に特有の動作(たとえばCRTディスプレイに対する表示位置決め命令)を使用したい場合は非標準I/O命令を用いて装置の特性を十分に引き出すことができる。

6.2 入出力装置の割当て

ロジカルデバイス番号に対する実際の入出力装置の割当ては、処理システム生成時に処理モジュール単位で行われる。ユーザは処理モジュール毎にロジカルデバイス番号と入出力装置の種類を定義する。この定義には2つのレベルがあり、1つは標準入出力装置とファイル装置の区別によるマクロレベル、もう1つは詳細にTTY, PTR, CRT等を表示するミクロレベルでありこのいずれの表現を用いてもかまわない。

I/Oデバイス定義はマスタ処理システムによって処理されデバイスが割当てられて処理モジュール毎にLOATがセットされる。この時点で割当てができなくても、I/Oフォールト(後述)が生ずるかもしれないという条件付きで割当て処理を続行する。

処理システム内では処理モジュール間で同一のI/Oデバイスを共有することができ、処理システム独自のI/Oデバイス制御が可能である。

6.3 機番変換と表示

ロジカルデバイス番号から実I/Oデバイス番号への機番変換は実行時に行われる。実行時にI/O装置の割当てられていないロジカルデバイス番号に対するアクセスがあると、アクセスしようとしたプロセッサモジュールはI/Oフォールトとなる。コントロールプロセッサはI/Oフォールトを検出し、イベント登録した後、マスタ処理システムに対しI/Oフォールト通知を出し以後の処理を仰ぐ。

マスタ処理システムでは、必要な入出力装置が空いていればこれを割当て、空いていない時はシステムコンソールにその旨表示してオペレータの指示を待つことになる。

またマスタ処理システムは入出力装置割当て変更の都度、システムコンソールに入出力装置とプロセッサモジュールの使用状況を表示している。従ってオペレータはこの表示を参考に、適切な入出力装置の割当て変更によってI/Oフォールトの解消を図ることができる。

Standerd I/O Instructions

INPUT
OUTPUT
CONDITIONAL INPUT
CONDITIONAL OUTPUT
INPUT SENSE
OUTPUT SENSE

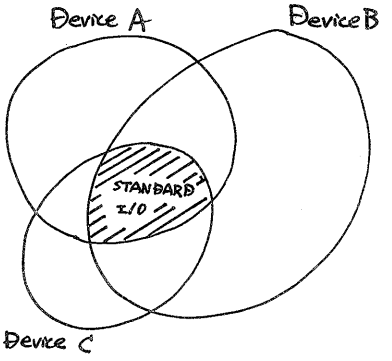


fig-5
Standard I/O Instructron Set

7. 処理モジュール間通信システム

処理モジュール間通信は実行時にはプロセッサモジュール間通信として実行される。通信はコントロールバス(C-Bus)と呼ばれる共通バス経由でエレメントスイッチングで伝送されるがロジカルにはプリミティブなプロトコルを有するメッセージスイッチング方式の通信を構成している。

7.1 通信の手順

伝送の手順は次のようになっている。

i) EXM レベル

ユーザプログラム内の通信要求はIPPを経由してCP内のEXMに伝えられる。EXMでは通信要求の種類毎に，データにCP-CP プロトコルが付加される。EXMがハードウェア的に検出した各種イベントに対する通信に関しても同様である。CP-CP プロトコルは受側のEXMに対して処理内容を指示するものであり，その種類は表1に示す。

ii) IMC レベル

専ら伝送手順として使われるCS-CS プロトコルが付加される。

すなわち　LINK (優先度, 送り先/元 指定)
START (メッセージ番号, 伝送語数)
データ
END

の形に編集される。(カッコ内は付加データの内容)
この段階で通信の送り先/元 は処理モジュール名からプロセッサモジュール番号に変換される。

以上の表示を設定した後，通信データはIMC中のCommunication Driver に渡り図6に示した手順で通信の相手方に対してエレメントスイッチングで通信を行う。リンクが成立したか否か，通信が正しく完了したかどうかの確認のためCommunication Driver は通信の相手方からLINK ACK及びEND ACK の引取りを行うが，これが得られない時は通信異常として異常通信を行い回復に努める。通信全般の形式を図6に示す。

7.2 優先度制御

任意の優先度でLINKが成立し，通信が行われている間は，他の通信相手方が送り出した通信は，たとえ送り先が一致していてもハードウェア的に受取りが拒否される。しかし実行中の通信より優先度の高いリンク要求はCommunication Driver によって検出される。従って受信側は適切な時点で通信を中断して高優先度の通信に切換えることができる。

優先度の割当ては次のようである。

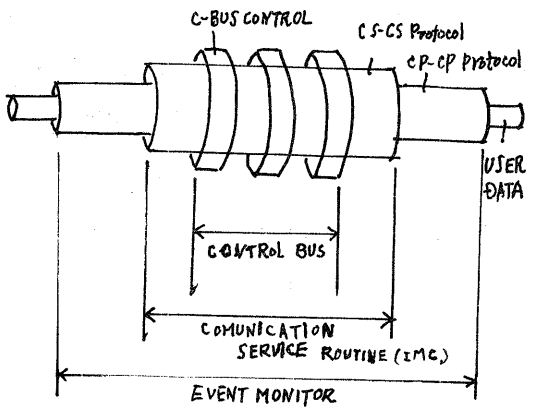
Priority α (最高)	要素通信
1	ページフォールト関係
2	I/O フォールト関係
3	ユーザ処理モジュール間通信及びコントロール

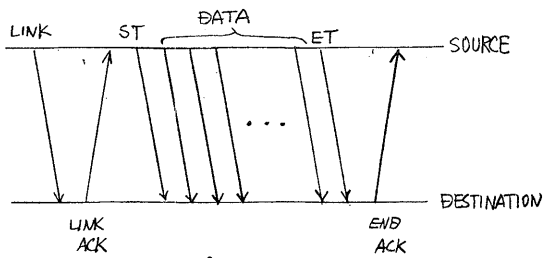
7.3 ユーザ処理モジュール間同期

ユーザ処理モジュール間の同期にはユーザ処理モジュール間通信を用いることができる。同通信のうち条件転送は受け側のユーザ処理モジュールの受信要求と送り側の要求との一致を得

PROCESSOR CONTROL
RESOURCE ASSIGNMENT　Page fault
I/O fault
INTER PROCESSING MODULE COMMUNICATION
Conditional Transfer
Non Conditional Transfer

Table－1
Communicatron Type





fig－6
Inter Processing Module Communication

た後，受け側のユーザプログラムのアドレス空間内にデータが引き渡される。これに対し無条件転送では受け側の意向にかかわらずデータは受け側のユーザプログラムのアドレス空間に書き込まれる。

処理モジュール間の同期は前者を利用して行われている。すなわちユーザプログラムが受信要求を発することがセマフォオペレーションにおけるPオペレーションに，条件転送を依頼することがVオペレーションに対応する。この方法によると同期操作そのものは常に受側の処理モジュール内で行われることになる。

同期操作に関する更に細いレベルの操作（たとえば複数条件の AND, OR 等）に関してはユーザプログラムの内部の処理に任されている。

8. コントロールプログラムの構成

8.1 機能構成

MICS-II OSは基本的にはExecutron Monitor によって検出される種々のイベントに対応した処理を行うモジュールで構成されている。（図7）

論理的には，すべてのスーパーバイザリな処理はExecutron Monitorの管理下にある。Executron Monitorによって検出された各種イベントはその性格に応じてResource Management System, Execution Management System, Inter Processing Module Communication System の3種のシステムのいずれか，あるいは複数を起動し適切な処理をさせる。この中で，Resource Management System は Execution Management を経て働くだけではなく，定時リソース管理という形で Executron Monitor によって直接アクセスされ，リソースの整備を行うことができるようになっている。通常の計算機システムではこのような方式はOS オーバーヘッドの点で必ずしも好ましくはないが，プログラム実行用のプロセッサと管理用のプロセッサが完全に分離しているMICS-IIに於ては極めて有効な方法と考えられる。以下に各モジュールの機能を簡単に述べる。

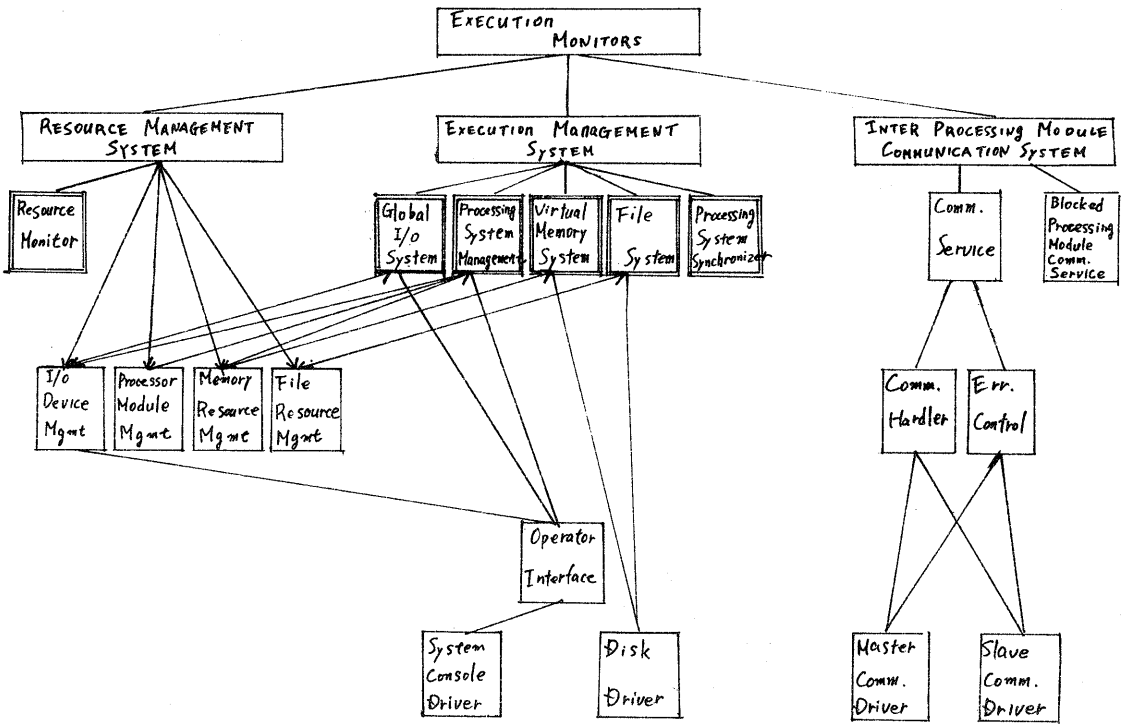


fig - 7
Control Software Configuration

● Execution Monitor (EXM)

実行中のイベント検出と管理を行う。必要に応じて Inter Processing Module Communication を用いて Execution Management System の稼動を要求する。このソフトウェアモジュールはすべてのプロセッサモジュールに分配される。

● Inter Processing Module Communication (IMC)

処理モジュール間通信の実行を管理するモジュールで CS-CS プロトコルの付加/解読, C-Bus Control の設定, 確認, 伝送手順の実行, 高優先度通信の検出, エラーリトライ等を行う。

Blocked Processing Module Communication (BPMC) は, 通信発生時点でプロセッサモジュールにアサインされていない処理モジュールに対する通信をすべて受信し, 対象処理モジュールが起動されるまでその内容を保持しておく。BPMC 以外はすべてのプロセッサモジュールに分配されている。

● Execution Management System (EXMGT)

実行管理の本体であり, 実行中発生する各種イベントに対するリソースアサインメントとその変更を主として行う。システムプロセッサモジュールに集中されている。

Processing System Management は処理システムの生成消滅に関する処理を行うが, 現在この処理はユーザのジョブエントリ操作と対応している。すなわち, システムコンソールから入出された指示に従って必要なプロセッサモジュール, I/O 装置, 共有ページの割当て等を各 Resource Management に依頼し処理システムテーブル上に処理システムの状況をコピーしておく。

● Resource Management System (RMS)

システムリソースとして集中管理しているリソースに関するリソース管理と Resource Monitor より成る。システムプロセッサモジュールに集中されている。

Resource Monitor は I/O, プロセッサ, メモリリソース等の使用状況のモニタリングを行うモジュールである。

8.2 コントロールプログラム制御構造とオペレーション

MICS-Ⅱ OS の実装上の特徴の1つに割込を用いないオペレーションがある。これには次の意味がある。

(1) プロセッサモジュール相互の自律的な相互チェックに備え強制的にコントロールの移る割込は好ましくない。

(2) 実行時に生ずる各種イベントを管理のきっかけとするだけでなく積極的に定時管理を行う上に有利である。

(3) コントロールプログラムを実行するプロセッサを浮動割当てとした時に有利である。

(4) ユーザプロセッサのレベルでは, ユーザの使用するアドレススペースに対する制約を極力避けることと及びユーザレベルでの割込の使用を制限しないという方針による。

MICS-Ⅱ のようなマルチプロセッサシステムで分散制御を行っている場合は, イベントの発生に伴うプロセッサモジュールの状態登録・イベントの解消・リソース割当て等が異るプロセッサに分散して処理される関係上, ロジカルには

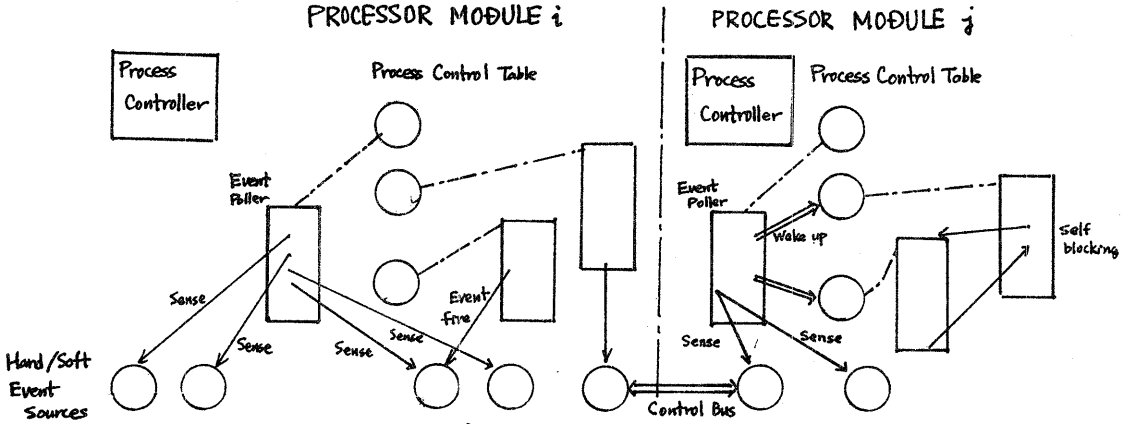


fig-8

System Event Poller and Process Handling

一連の処理と考えられるもの（ページフォールトの検出，ページリプレースメント，再実行）がプロセッサ間に分割されることがある。このような場合にはコントロールプログラムは必然的にシステム管理処理の途中で中断され，他の処理要求のサービスを行うことになりコントロールプログラムの制御構造が問題となる。

MICS-Ⅱ OS では，コントロールプログラムの処理能力に拡張性を持たせると共に各サービス毎に完結した形を持たせ見通しを良くすべく各種ハンドラを1つのプロセスと見てマルチプロセスで働く形をとっている。

また割込を使用しないので各プロセスを起動する機会はソフトウェア的に作る必要がある。MICS-Ⅱではシステム状況の定時チェック及び相互診断のための方策も含めコントロールプロセスの1つとしてSystem Event Poller (SEP)を設けこれが常にイベントの発生監視を行っている。

各コントロールプロセスは一定長以下のセグメントに分けられ，各セグメントの終りにSEPを起動し自身をブロックして他のコントロールプロセス起動の機会を与えている。コントロールプロセスにはそれぞれID番号とプライオリティが与えられコントロールプロセスの発生，消滅同期操作はすべてIDによって管理される。各セグメントはいずれもStatus Saveを必要としないように作られており，これによってプロセス管理ルーチンは小規模なものとなっている。（ほぼ100 ステップ）

コントロールプロセス間の同期は同一プロセッサ内の場合を基準とし，他プロセッサ間の場合はIMCを介して行われている。すなわち処理に他プロセッサモジュールがかかわるときは，EXMはIMCを介して通信を行うが，この通信がコントロールプロセス同期のひきがねとなる。同一プロセッサ内の場合は，直接アオペレーションを行って他プロセスを起動する他，コントロールイベントとしてSEPに管理されているイベントを起し，SEPに対してインプリシットに他プロセス起動を依頼する場合がある。コミュニケーションバッファフルに伴う通信プロセスの起動がその例である。この様子を図8に示す。

9. 現状

現在MICS-Ⅱはプロセッサモジュール6組，メモリモジュール32KW，グローバルI/Oデバイスとしてメモリバックアップ用ディスク1台，ASR-33（及び相当品）3台，インクジェットプリンタ1台，シリアルプリンタ／キーボード2台TVディスプレイ／キーボード1台，CRTディスプレイ／キーボード1台，紙テープリーダ2台が接続されており，既に開発されたコントロールプログラムの1部を利用してコントロールプログラム開発中である。

現状では処理モジュールの動的管理がサポートされていないので，これにかかわる各コントロールプロセスの開発が行われていないが，今後この点を含めて処理モジュール管理の追加及びコントロールプログラム全体のリファインメントを行ってゆく予定である。

謝辞

本システムの研究を行うにあたって，常に有意義な助言を与えて下さる日本電気(株)中央研究所周辺機器研究部 木地部長，福津課長に深謝します。

尚，本研究は通産省工業技術院大型プロジェクト「パターン情報処理システムの研究開発」の一環として行われているものである。

参考文献

(1) 大森他，MICS マルチマイクロプロセッサシステムについて，情処・研資 75-8

(2) 小池他，マルチマイクロプロセッサの制御方式，信学会研資 EC 74-11

(3) 大森他，複合プロセッサにおける入出力共有機構，情処，17回全大

(4) 山崎他，複合プロセッサにおけるコントロールプログラムについて，同上．