



Title	1次元循環アレイ型並列演算装置”PPA”
Author(s)	牧野, 圭二; Makino, Keiji
Citation	北海道大學工學部研究報告, 131, 49-59
Issue Date	1986-05-30
Doc URL	https://hdl.handle.net/2115/41984
Type	departmental bulletin paper
File Information	131_49-60.pdf



1 次元循環アレイ型並列演算装置 “PPA”

牧野 圭二

(昭和60年12月27日受理)

“PPA”: A One-Dimensional Processor-Array System Using Shared-Memory Technology

Keiji MAKINO

(Received December 27, 1985)

Abstract

This paper describes a parallel processor system PPA, which is the main component of the multi-purpose simulator HOSS at the Hokkaido University. PPA is a master-slave system consisting of two master processors and thirty-two slave processors. Its architecture is characterized by a one-dimensional array of slave processors sharing a three-port memory system. The shared memory, also, connects masters and slaves, and the masters control the slaves through it. The architecture allows each processor to exchange data with its neighbors through the shared memory without actual data transfer. This reduces the data-transfer portion of the overall execution time, the minimization of which is a difficult problem in a parallel processor system. It was demonstrated that PPA can be used as a real-time continuous-system simulator with a CSMP-type simulation language ICOSS.

1. はじめに

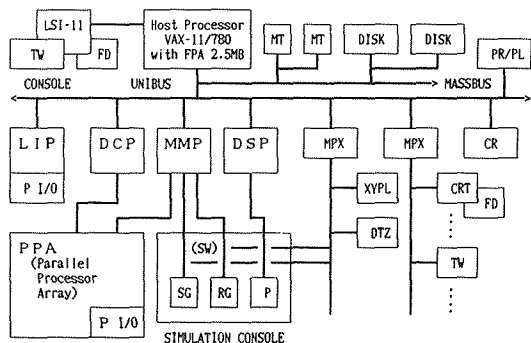
PPA (Parallel Processor Array) は、連続系実時間シミュレーションを行うことを主目的とする隣接密結合型の並列処理システムである。北海道大学では早くから並列処理方式の連続系シミュレータの検討並びに提案を行ってきた¹⁾²⁾が、PPAはこの提案をもとに実用機としての作成や保守の容易さを考慮して北海道大学全学共同利用施設汎用シミュレータ施設であらたに開発されたものである。

隣接密結合型の並列処理システムには多くの方式がある³⁾が、PPA⁶⁾⁸⁾は演算用プロセッサ32台とこれらを制御するプロセッサ2台の総計34台のプロセッサからなるマスタースレーブ方式のシステムである。演算用のスレーブ・プロセッサはシェアド・メモリ・システムによって1次元循環アレイ状に結合され、このシェアド・メモリ・システムを通してマスタ・プロセッサにより制御される。この構造により、プロセッサ間のデータ交換のそのほとんどをデータの移動なしにシェアド・メモリ上で行うことができ、並列処理システムでの性能低下の最大の原因であるプロセッサ間のデータ交換の問題を解決している。

本システムを構成している各エレメント・プロセッサは標準的な16ビット・ミニコンピュータである。PPAの開発と共に、演算自体の高速化のためアレイ・プロセッサとしても利用可能なハ
全学共同利用施設 汎用シミュレータ施設
Simulation Center

ードウェアの浮動小数点数演算機構を開発し全スレーブ・プロセッサに組み込んでいる。また、エレメント・プロセッサの持つマイクロプログラミング能力を生かし、32ビット・データの処理を中心に各種拡張命令を追加し、16ビット・マシンを32ビット・マシンのように取り扱えるようにしている。

連続系シミュレーションで用いるモデルとしては連立常微分方程式系を想定しており、そこでの並列化のタスクは演算子単位ではなくよりマクロな方程式単位で考えている。すなわち、スレーブ・プロセッサを一種の数値積分器とみなすことによりデータ交換の処理を減しつつ、アナログ計算機との相似性を持たせて利用者の理解あるいはソフトウェアの作成の容易さを目指している。



Control Processors
 LIP: Laboratory Interface Processor
 DCP: Data-Communication Processor
 MMP: Man-Machine communication Processor
 DSP: Discrete-system Simulation Processor

Simulation Console
 SG: Storage-type display
 RG: Refresh-type display
 P: display panel
 SW: Bus Switch

図1 HOSSの全体構成

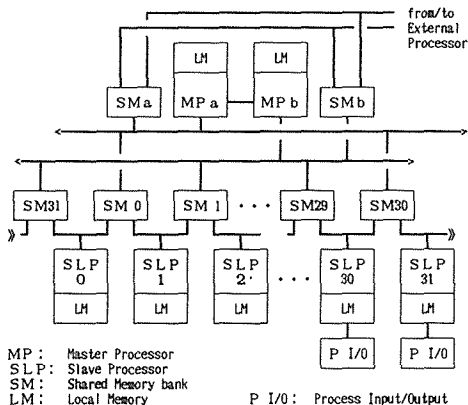
このことは、PPAと同時に開発されたCSMP型のシミュレーション言語ICOSS (Interactive Continuous-System Simulation System)⁴⁾で実証されている。

PPAはICOSSと共にその開発終了後、上記汎用シミュレータ施設的全デジタル式汎用シミュレータである北海道大学「高速システム・シミュレーション装置」HOSS(Hokkaido University High-Speed System Simulator)³⁾⁵⁾にその中心をなすサブシステムとして組み込まれ、実用機として広く学内の一般研究者に利用されている。図1にHOSSの全体構成とそこにおけるPPAの占める位置を示す。

2. PPAのアーキテクチャ

2.1 PPAの構成

並列演算装置PPAは隣接密結合型のマスタースレーブ方式のシステムである。各エレメント・プロセッサはそれぞれ独立なプログラムを実行でき、基本的にMIMD型(Multiple Instruction and Multiple Data Stream)のシステムである。



MP: Master Processor
 SLP: Slave Processor
 SM: Shared Memory bank
 LH: Local Memory

P I/O: Process Input/Output

図2 PPAの構成概念

PPAは、演算用の32台のスレーブ・プロセッサ(SLP_i, $i = 0, 1, 2, \dots, 31$; 以下同様)が3ポートのシェアド・メモリ・バンク(SM_i; SMバンク)を間に挟みながら1次元循環アレイ状に並び、このSMバンクと制御用の2台のマスター・プロセッサ(MP_a, _b)が接続されるという基本構造を持っている。このとき、2台のMPはこのプロセッサ・アレイ上のSMバンクと交互に接続が行われ、また、PPA外部のプロセッサとの通信のため、他に2個のSMバンク(SM_a, _b)が用意され接続されている。この接続関係の基本概念を図2に示す。一方、SMシステムを中心に、PPAの内部構造に着目して表現したのが

図3である。

MPは接続されている16個のSMバンクをアクセスでき、それを通して16台ずつのSLPを制御・

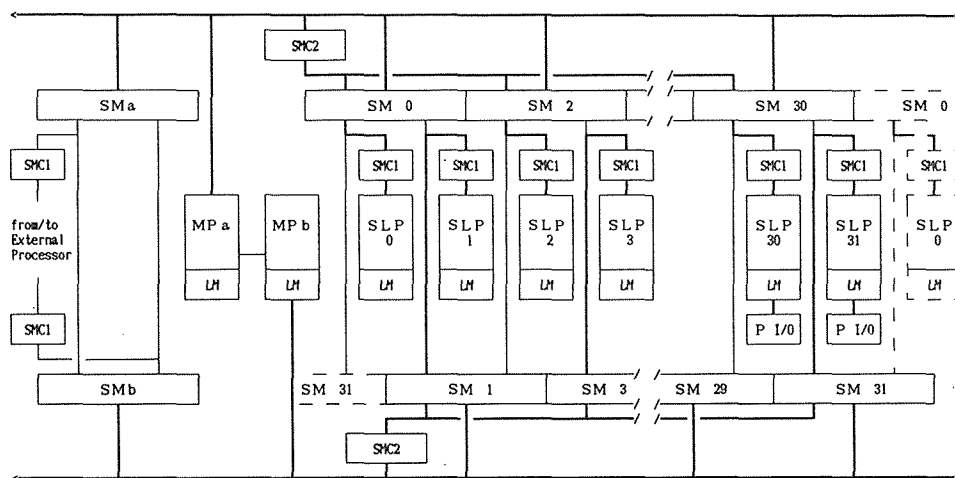


図3 PPAのアーキテクチャ

統括する。制御の上からは SM バンクと SLP は対をなし、16組ずつ 2 群に分かれている。従って、各 SLP は自分に接続されている 2 個の SM バンクのうち制御を受けるバンクがどちらかに決っており、一方の MP からのみ制御される。しかし、各 SLP は 2 個の SM バンクを同等にアクセスすることができ、データの上からは両 MP からのデータを受け取り処理することができる。また、SM システムには、後で述べるように、制御やデータ交換の効率化のためにアドレス切換機構、ブロードキャスト転送機構、割込機構などが付随している。これら SM システムの制御の主要機能は図 3 における SMC2 (シェアド・メモリ・コントローラ II) が受け持っており、その制御の経路がそれに伴う太線で示されている。

図 3 のアーキテクチャから明らかなように SLP 間でのデータ交換は SM バンクを通して行われる。その経路は複数用意されているが、一般に隣接 SLP 間の場合はデータの移動なしで直接行い、遠隔 SLP 間の場合だけ MP 経由で行う。この遠隔 SLP 間の場合、SM バンクの選び方によってどちらの MP 経由でも交換を行うことができる。

なお、32 台の SLP のうち 4 台 (SLP28~SLP31) には、P I/O (AD, DA, 及び、デジタル入出力ポート) が装備されており、非実時間シミュレーションのときには演算プロセッサとして働くが、実時間シミュレーションのときには PPA の外部機器に対する入出力用プロセッサとして働く。

2.2 エLEMENT・プロセッサの性能

PPA を構成している ELEMENT・プロセッサは、三井造船(株)製の MAP16 と呼ばれるマイクロプログラム演算制御方式を用いた 16 ビット・ミニコンピュータである。MAP16 は DEC 社の PDP-11/34 と類似の機械語命令 (マクロ命令) 体系を有し、ユニバスと互換性があり性能も同等 (表 1) の標準的ミニコンピュータである。

PPA 内部での基本演算を浮動小数点演算にするため新たにハードウェアの高速浮動小数点演算機構 (FFP; Fast Floating-point-data Processor) が開発され全 SLP に装備されてい

演算制御方式	マイクロ・プログラム
内部クロック	230 ns
汎用レジスタ	8 個
最大アドレス空間	32 kw
最大 RAM 領域	28 kw
語長	16 bit/word
最高データ転送速度	2 Mw/sec
命令	基本 90 拡張
演算速度	マイクロプログラムにより可能 (単位: μs)
固定小数点	
加・減算	1.1 (R-R), 1.6 (R-M)
乗算	11.1
除算	11.2

表 1 ELEMENT・プロセッサの基本性能

接続方法 演算制御方式	SLPの内部バス マイクロプログラム バイプライン方式
レジスタ・ファイル 語長	16個 指数部 8 bit (2進) 仮数部 32 bit ROMによるまるめ (単位: μs)
仮数部の短縮処理 演算速度	
浮動小数点数 加・減算	2.11
乗・除算	5.12 (4.47: R-R \rightarrow R)
ロード	2.21
ストア	1.81

表2 高速浮動小数点数演算機構の性能

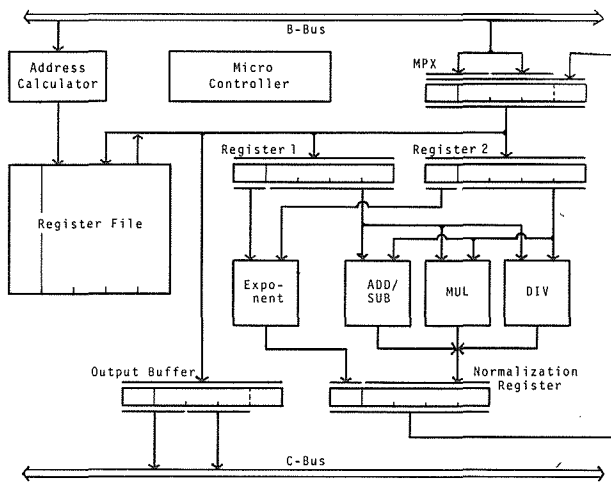


図4 高速浮動小数点数演算機構 FFP の構成

る。FFP 内でのデータ表現は指数部2進8ビット、仮数部24ビット(含む付号ビット)であり、1語が40ビットのデータとして演算が行われる。SLP 本体にデータが戻されるときには、拡張された仮数部の下8ビットがROMを使ったまるめ処理により除去される。FFPはSLPの内部バスに接続され、それ自体16語のレジスタ・ファイルを内蔵しベクタ演算機構を有しておりアレイ・プロセッサとしても機能する。その構成と基本性能を図4及び表2に示す。

SLPと対で用いられる各SMバンク(表3)は4KBの容量を持ち3ポートである。ポート1がMPとの接続用、ポート2,3がSLP間の接続用である。ポート2,3の切り換え制御のためのSMC1(シェアド・メモリ・コントローラI)が各SMバンク毎に、また、全SMバンクの制御のためのSMC2がポート1側のバスに、SMシステムの一部として用意されている。

エレメント・プロセッサMAP16は基本的には16ビット・マシンである。

記憶容量 接続方法	1KB単位の増設 3ポート (ポート1: 共通バス・バックプレーンに直接接続 ポート2,3: シェアド・メモリ・コントローラを介して他の共通バスと接続) 各ポート共、1KB単位で個々に設定可能
アドレス割り当て	
サイクル・タイム (他ポートのアクセス状態)	350 nsec (アクセス無し) 460 nsec (1 アクセス) 570 nsec (2 アクセス)
各ポートのサービス	First Access First Service

表3 シェアド・メモリの性能

(32-bit floating-point-data instruction; extracted)

(move)	MOV, BMOV (block move)
(arithmetic)	MADD, MSUB, MMUL, MDIV
(vector)	VADD, VSUB, VMUL, AMUL, VSML
(complex number)	CADD, CSUB, CMUL, CDIV, CONJ
(conversion)	FLIN (floating to integer) INFL (integer to floating) CMPL (real to complex) REAL (complex to real) IMAG (complex to real)
(miscellaneous)	INTGRL, DMT (address on dimension) MAX, MIN, POLY (polynomial)

表4 スレーブ・プロセッサの付加機械語命令

しかし、マイクロプログラム制御の特徴を生かして各種の32ビット・データの処理命令やプロセッサの機能別の専用命令を用意し、オーバーヘッドを減らしながら32ビット・マシンであるかのような取り扱いを可能にしている。例えば、32ビット・データの移動命令、FFP用の演算命令、

データ変換命令，連続系シミュレーション演算用の命令，さらに，制御用の判定命令など多岐にわたっている。表4に主なものを示している。

3. メモリ・アドレッシングと実行の制御

図2，図3に示されるように，PPAの構造はMPに関して対称であり，両MPは対等である。また，各SLPと各SMバンクは1次元リング状に構成されており，SLP側から見る場合それぞれPPA内で互いに対等であり交換可能である。以下説明はこの対称性に基づき，構成の核であるシェアド・メモリ・システムを中心に行う。なお，アドレスに関しては8進数で表すものとする。

3.1 メモリ・アドレッシング

本システムのメモリ構成は3ポートのSMバンクからなるSMシステムにその特徴がある。同一のSMバンクでもそのアドレスはアクセスする側により異っている。以下では，マスタ・プロセッサ及びスレーブ・プロセッサのメモリ・アドレッシングを示した後，その間のSMバンクのアドレスの対応関係を示す。

● マスタ・プロセッサのメモリ・アドレッシング

MPのメモリ領域は，図5に示すようにワード・アドレスでアドレスが付けられているSMシステムの領域とバイト・アドレスでアドレスが付けられている固有メモリの領域（内部メモリ；SLPのローカル・メモリに対応）との2系統に大きく分かれている。またこの他に，制御プログラム用のROM領域が内部メモリの後ろに用意されている。

SM領域に属する各SMバンクはそれぞれ1.5KW(3KB)ずつのアドレスが割り当てられている。また，ブロードキャスト転送(後述)用のSMBCR(SMブロードキャストリング・レジスタ)のアドレスがSMバンクの先頭から0.5KWに対応する形で割り当てられている。なお，MPにと

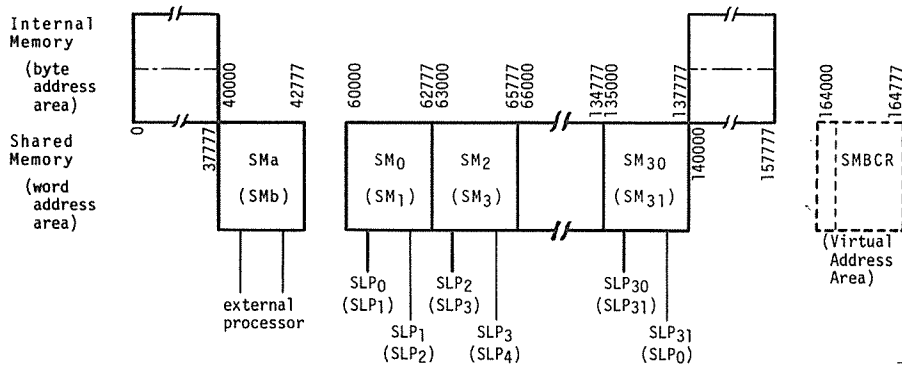


図5 マスタ・プロセッサのメモリ・アドレッシング

って，ワード・アドレスが付けられているこのSM領域はデータ・バッファ領域であり，MP側ではこの領域内にプログラムを置いて実行することはできない。

● スレーブ・プロセッサのメモリ・アドレッシング

SLPのメモリ領域はすべてバイト・アドレスでアドレスが付けられている。

図6に示すように，各SLPには異なる制御系

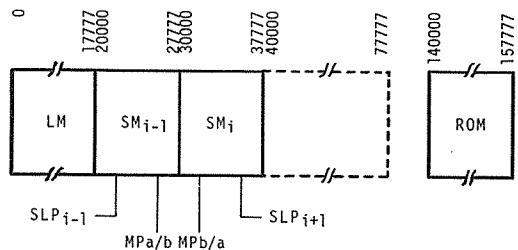


図6 スレーブ・プロセッサのメモリ・アドレッシング

統 (MP) に属する 2 個の SM バンクが組み込まれ、各 SM バンクに 4KB ずつのアドレスが割り当てられている。そのうち制御を受ける SM バンクはアドレスの高位の側に割り当てられている。しかし、それらの差異は意識されず、SLP 内ではデータのアクセスに関して対等な立場でかつローカル・メモリ (LM) と同様にアクセス可能である。なお、現在 LM の拡張が予定されている。

● シェアド・メモリ・バンクのアドレッシング

図 7 は、1 台の SLP に組み込まれている SM バンクに着目し、特に、制御を受けている SM バンクに重点を置いて、SLP 側からと MP

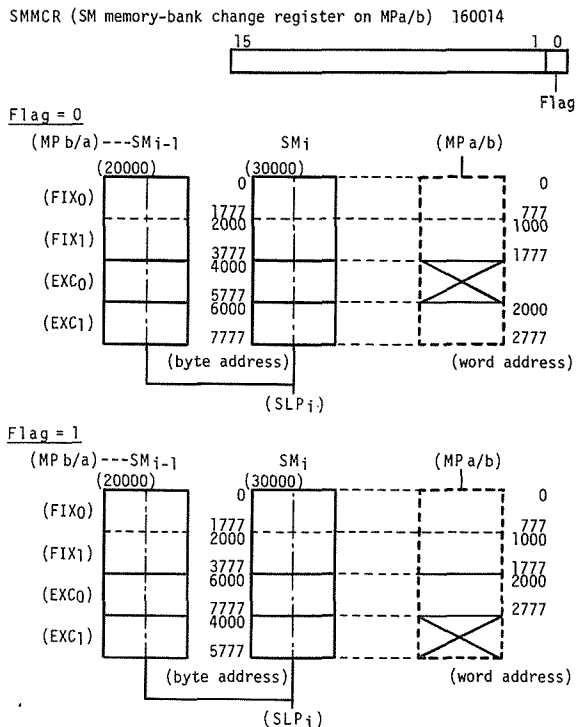


図 7 シェアド・メモリ・バンクのアドレッシング

ロードキャスト転送データの受け取り領域でもあるためブロードキャスト可能領域とも呼ばれる。

各プロセッサから割り当てられたアドレスに対し、アドレス固定領域は全ポート共にアドレスの設定値が固定され、アドレス切換可能領域は MP の SMMCR (SM メモリバンク切換レジスタ) の最下位ビット (フラグ) の状態に応じてアドレスの設定値が決まるようになっている。しかし、ポート 1 (MP 側) では二つのアドレス切換可能領域に割り当てることができるアドレスが 1 領域分 (0.5 KW) しかないため、片方には、空アドレス (アクセスできない) を設定せざるを得ない。従って、MP からはフラグの状態によりアドレス切換領域のどちらか一方だけが見え、SLP からはフラグの状態にかかわらず両方が見えることになる。

MP の SMMCR のフラグを切り換えるとアドレス切換機構により、対応する各ポートのアドレス切換可能領域のそれぞれの物理アドレスが一斉に入れ換る。その結果、MP からはアドレス切換可能領域の見える領域が変わり、SLP からは両領域のそれぞれの全データがブロック転送により交換されたように見える。

3.2 ブロードキャストと割り込み

PPA はマスタースレーブ方式であり制御の主導権は MP が持っている。MP が SLP を制御す

に重点を置いて、SLP 側からと MP 側からとのアドレッシングの様子を両者対応付けて示したものである。なお便宜上、両者共にその SM バンク内での相対アドレス (下 4 桁) で示してある。

それぞれが 4KB の大きさで 3 ポート構成の各 SM バンクは、前述のように、MP からはポート 1 に対してワード・アドレスで 1.5 KW、また、SLP からはポート 2 及び 3 に対してバイト・アドレスで 4KB ずつそれぞれアドレスの割り当てを受けている。一方この SM バンクは、各バンク内でそれぞれ 1KB ずつの 4 個の領域、すなわち、二つのアドレス固定領域 (FIX0, FIX1) と二つのアドレス切換可能領域 (EXC0, EXC1) とに分かれている。その各領域は、ポート 1 (MP 側) に 0.5 KW、ポート 2 及び 3 (SLP 側) に 1KB のアドレスの設定が可能となっている。なお、このアドレス固定領域のうち FIX0 領域は MP の SMMCR とも対応しており、MP からのブ

る機構として、MP からその制御下にある全 SLP へ同一のデータを送るためのブロードキャスト機構、また、このブロードキャスト機構を利用して、MP による SLP の制御や処理の同期を行うための割込機構が用意されている。これらの機構は SM システムを中心に図 8 に示す各レジスタの補助のもと SMC2 により制御されており、その処理の関係は図 9 のようになっている。

MP は特定の SLP だけを選択的にブロードキャスト転送や割り込みの処理の対象とすることができる。すなわち、MP が持っている SMACR (SM アクティブ・レジスタ) の各ビットがその MP の制御下にある各 SLP に対応しており、SMACR のそのとき ON 状態に設定されている各ビットに対応する SLP が MP からの処理の対象となる。

ブロードキャスト転送は MP が SMBCR にデータを書き込むことにより行われる。すなわち、MP が SMBCR にライト・アクセスすると SMACR で ON 状態にある SLP と対になっている全 SM バンクへそのデー

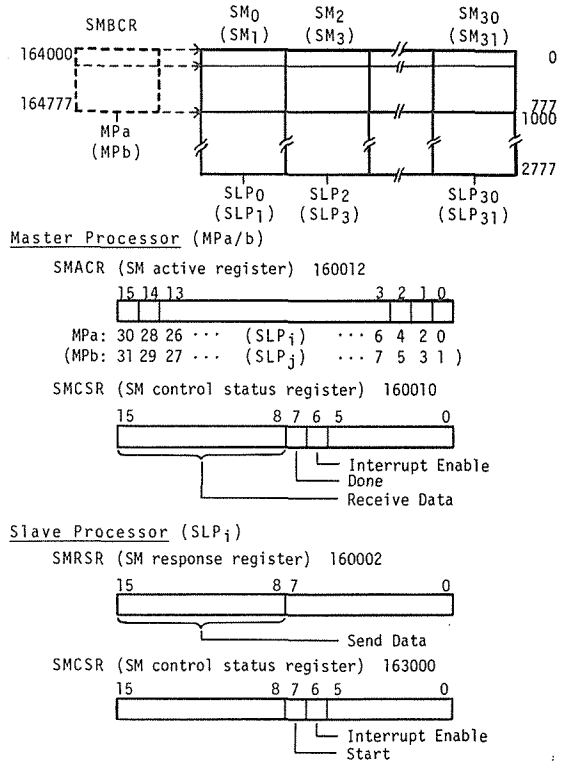


図 8 ブロードキャストと割り込み制御

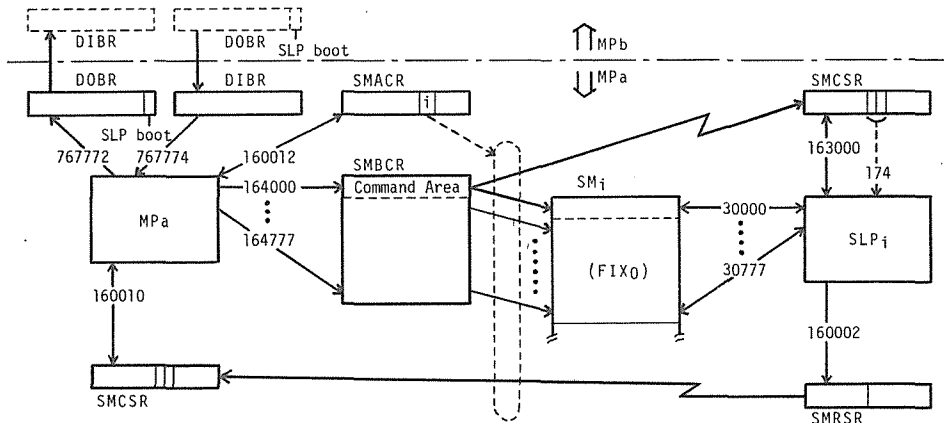


図 9 MP 間の同期、ブロードキャスト、割り込みの関係

タが送られ、SMBCR と下位 9 ビットが等しいブロードキャスト可能領域のそれぞれの位置にそのデータが同時に書き出される。このとき、SMBCR の先頭の 1 語はコマンド・エリアとも呼ばれ、この語への書き込みはブロードキャスト転送と同時に MP から各 SLP への割り込みも引き起す。

MP がコマンド・エリアへの書き込みを行うと、対応する各 SLP に対しバス・リクエストを起こし、その SLP の SMCSR (SM コントロール・ステータス・レジスタ) のスタート・ビットを ON にする。このとき、その SMCSR のインタラプト・イネーブル・ビットが ON にセットされ

ていれば実際に割り込みが発生し、その SLP は (174)番地へのトラップを起こす。この番地が ROM 内の割り込み処理ルーチンへの入り口となっている。通常コマンド・エリアには SLP での実行ルーチンの実行開始アドレスが書かれるが、この割り込み処理ルーチンによってそこで指示された実行ルーチンに制御が渡される。なお、コマンド・エリアへの書き込みと同時に MP の SMCSR がクリアされる。

SLP から MP への割り込みは各 SLP が MP から指示された処理の終了を伝えるためのものである。すなわち、この SLP からの割り込みは、MP の SMACR で指示されている全 SLP が各自の処理終了と共にそれぞれの SMRSR (SM レスポンス・レジスタ) にデータを書き込むのを待つて発生する。従って、SMACR で指示される以外個々の SLP から個別に MP へ割り込みをかけることはできず、SLP から MP への個々の情報の伝達はそれぞれの SM バンクを通して行うことになる。

SLP が各自の SMRSR にライト・アクセスを行うと、その SLP が制御を受けている SM バンクの第 2 ワード目 (アドレス固定領域 FIX0 の (30002)番地) にそのデータを書き出し、同時に MP の SMCSR へライト・アクセスを行う。MP の SMCSR はその MP の SMACR が ON 状態にある全 SLP からのライト・アクセスの完了を待つてレシープ・フィールドに 1 をセットする。この結果、MP の ROM に書き込まれている制御プログラムに制御が戻ることになる。なお、SLP の SMRSR へのライト・アクセスと同時にその SLP の SMCSR がクリアされる。

MP 間の同期操作は割込機構からは独立しており、MP 相互のデジタル入出力ポートを用いて行っている。

4. 連続系シミュレーションへの応用

PPA 開発の主目的は実時間の連続系シミュレーションを行うことである。このため PPA と同時に CSMP 型の連続系シミュレーション言語 ICOSS が開発されている。そのトランスレータでは、ソーティングをはじめ、並列処理のためのタスクの分割や各 SLP へのタスクの割り当てなどの処理を自動的に行っており、PPA のハードウェア構成を意識することなく会話形式でオンライン実時間シミュレーションを実行することができる。以下では IOCSS を例に PPA の利用方法について述べる。

4.1 PPA における実行方式

IOCSS は CSMP 型の言語であり連立常微分方程式系の数値積分法に適した言語である。PPA での並列化はこの積分演算に着目し、連立常微分方程式系を積分演算をもとに方程式レベルで分割し各 SLP に割り当てる方法によっている。このとき各積分演算に必要な代数式はその積分演算に付随するものとして各 SLP へ重複を許して割り当てている。すなわち、各 SLP は積分に付随する演算も含めて計算を行う一種の数値積分器である。さらに、各 SLP がこのような数値積分器を複数台分担できるものとしている。

この考え方によると、各 SLP は他の SLP で計算を行っている積分演算の出力値を必要としないならばその SLP だけで独立して計算を続けることができる。また、もし他の SLP で計算を行っている積分演算の出力値を必要とする場合でも、それが隣接 SLP であれば MP の介在なしに SM バンクを通して直接受け取り計算を続けることができる。しかし、一般的な場合として遠隔 SLP の積分演算の出力値を必要とする場合を考慮すると、各積分刻みの終了毎の MP によるこの値の交換処理過程を省くわけにはいかない。

このため ICOSS では、まず各 SLP でそれぞれに割り当てられた計算を 1 積分刻み分行い、そ

の後 MP に制御を帰し次の積分刻みの計算のために交換が必要な積分出力があればそこでその交換を行っている。従って、必然的に各積分刻みの計算毎に同期がとられることになり、これを利用して実時間実行時のシミュレーション・タイムの管理を行っている。

4.2 シェアド・メモリへの変数の割り当て

SLP 間でのデータの交換は、前に述べたように SM システムを用いて、隣接 SLP 間の場合はデータの移動なしで直接交換ができ、遠隔 SLP 間の場合だけ MP 経由で行うことが必要であった。しかし、この遠隔 SLP 間の場合にも SM バンクの選び方によってどちらの MP 経由でも交換を行うことができるので、多数のデータの場合には各 MP で半数ずつ受け持てば全データの交換ができデータ交換の時間が半分に短縮できた。さらに、多数の SLP への転送はブロードキャスト転送を用いることができた。

IOCSS ではこのような SM システムの特徴を生かすため、SM を変数領域、LM をプログラム領域として使用している。また、SM バンク内の個々の領域の持つ特徴を考慮して、アドレス固定領域には MP がシミュレーションの実行途中で操作する可能性のある変数を割り当て、アドレス切換可能領域には各 SLP の実行時の作業変数を割り当てている。その割り当ての様子を図10に示す。

アドレス固定領域の内ブロードキャスト可能領域には優先的にシステム変数と全積分変数を割り当て、残りのアドレス固定領域に全パラメータ変数（関数発生器で使用するテーブルを含む）を割り当てている。このシステム変数域は連続系シミュレーションの実行を制御する上で必要な MP からの情報、すなわち実行コマンド、ステータス、シミュレーション・タイム、積分法、積分刻み幅などを受け取るための領域である。また、積分出力変数域は遠隔 SLP からの積分出力値を受け取るための領域である。この領域では、特に、SLP 内の隣り合う SM バンクに全積分変数のそれぞれ半分ずつを割り当て各 MP の負荷を半分に減らしている。なおこの転送にはブロードキャスト転送を用いている。

二つのアドレス切換可能領域のそれぞれには、1積分刻みの計算を行うときの各 SLP の入力データ域と出力データ域として、まったく同じ形に積分変数と代数・論理演算変数（パラメータ変数を除く）とを割り当てている。その場合、特に積分アクセス領域にはその SM バンクをアクセス可能な2台の SLP の積分変数を割り当て、隣接 SLP 間でのデータ転送をなくしている。従ってシミュレーション実行時には、MP が積分刻み毎に SMMCR を切り換えることによって、各 SLP はアドレス修飾なしに同一のアドレスをアクセスしながら計算を続けることができる。このとき入力データ領域は、MP が SLP での演算と並行して解表示用のデータを取り出せるように、MP からアクセス可能な方にとられる。

現システムでは、1問題で処理可能な積分演算個数は各 SLP 当り6個（192個/システム）である。パラメータ変数への領域の割り当て個数を減らしてブロードキャスト可能領域すべてを積分出力変数域とするならば、処理方式を変更せずに各 SLP 当り15個（480個/システム）の積分演算まで取り扱えるようになる。しかし、この場合このような大規模なプログラムをどのように入力

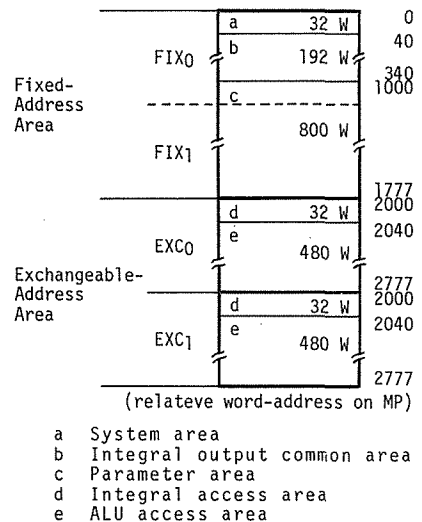


図10 シェアド・メモリへの変数割り当ての例 (IOCSS)

するかという別の問題が生じる。

5. おわりに

本稿では、並列演算装置 PPA について、そのアーキテクチャを示し、連続系シミュレーション・システム ICOSSE における利用方法について論じた。

一般に、並列処理システムでは演算量の増加につれてデータ転送量が多くなりその処理時間が無視できなくなる傾向がある。特に、マスタスレーブ方式のシステムでは各 SLP の並列化の効果は大きいですがデータ交換の処理が MP に集中しやすく、全体としての処理効率低下の原因となりやすい。本システムでは、これを改善するため 3 ポートのシェアド・メモリ・システムを用意しそれを中心にデータの交換経路を種々のレベルで複数化すると共に、極力データの移動なしでデータの交換ができるようにしてデータ転送時間の転減をはかっている。

しかし、本システムで用いている SM バンクの容量並びに各ポートへの変則的なアドレスの割り当て方は PPA のアーキテクチャにとって本質的なことではない。エレメント・プロセッサが 16 ビット・マシンであるので、その基本アドレス空間内にシェアド・メモリを含めようとしたために生じた制約である。現在普及しつつある 32 ビット・マシンを用いるならばこれらの制約はほとんど解消し、その結果、設計の自由度が非常に高くなり PPA の性能の一層の向上を図ることができる。

PPA ではマスタスレーブ方式を採用し、そこでの処理として連続系シミュレーションを想定しているということもあって、割込機構はエレメント・プロセッサが本来持っている機能をかなり簡略化して用いている。しなわち、マスタスレーブ方式であるため制御の主権は MP が持っており、各 SLP は単独で積極的に MP へ割り込みをかけることはできず、あくまでも MP からの問いかけに受動的に答えるためのものとされている。また、各プロセッサは想定された処理目的の処理効率の向上のため、その目的に沿って制御機能の簡略化を行い制御プログラムのほとんどを ROM 化している。そのため、割込機構自体もこれに合わせた形で 1 レベルに簡略化されており、一般の並列処理システムとしてみるならば割込機構自体は弱いものとなっている。

PPA にはまだ改善すべき点もあるが、3 ポート・シェアド・メモリ・システムを用いた 1 次元循環アレイ構造によるシステムの構成は対象プログラムの並列割り当てに柔軟性を持ち、多くのアプリケーションへの応用を可能にしている。これは、ICOSSE 以外にも、有限要素シミュレータ⁹⁾あるいは原子炉シミュレータ¹⁰⁾のプログラム開発が行われていることでも示されている。また、本システムの構造は並列処理システムにとって致命的となりかねない一部の演算プロセッサの障害に対しても強い構造である。

謝 辞

本 PPA は北海道大学「高速システム・シミュレーション装置」HOSS の中心となるサブシステムであり、この PPA を含む HOSS は、1978 年度及び 1979 年度の 2 年度にわたる文部省特別設備費によって設置された。仕様の決定まで長期間詳細な検討を行われた北海道大学全学共同利用施設汎用シミュレータ施設利用開発小委員会、製作全般にわたり積極的協力を得た三井造船株式会社システム本部、実現まで多大な努力を払われた多数の関係各位に対し深く敬畏の意を表すものである。

上記利用開発小委員会の委員として長期にわたる検討期間を通じ多くのアイデアを出され実現に熱意を傾けられた故小山昭一先生（当時北海道大学工学部精密工学科助教授）に謹んで哀悼の

意を表します。

参考文献

- 1) Koyama, S. and Miura, R.: Proc. 8th AICA World Congress, (1976), p. 263-269.
- 2) Koyama, S. and Miura, R.: Proc. 7th IFAC World Congress, (1978), p. 781-788.
- 3) Makino, K., Koyama, S., Miki, N., Iseki, Y., Kobayashi, H. and Sakai, Y.: Proc. 9th IMACS World Congress, (1979), p. 171-177.
- 4) Iino, Y., Makino, K., Iseki, Y., Koyama, S. and Miki, N.: Proc. 9th IMACS World Congress, (1979), p. 261-270.
- 5) 牧野圭二, 小山昭一, 三木信弘: 信学技報, Vol. 80, (1980), No. 75, p. 55-66.
- 6) Koyama, S., Makino, K., Miki, N., Iino, Y. and Iseki, Y.: Proc. 8th IFAC World Congress, (1981), p. 1715-1720.
- 7) 中川徹: 計測と制御, Vol. 21, (1982), No. 4, p. 45-53.
- 8) 牧野圭二, 三木信弘: 信学技報, Vol. 82, (1982), No. 62, p. 57-68.
- 9) 土肥俊, 小山昭一: 電子通信学会論文誌, Vol. J65-D, (1982), No. 4, p. 464-470.
- 10) 勝海和彦, 辻雅司, 小川雄一: 汎用シミュレータ室だより, No. 15, (1985), p. 84-91.