

# マルチマイクロプロセッサシステム HARPS Multimicroprocessor System HARPS

概要と汎用ファームウェアシステム  
An Outline of HARPS and Its General Purpose  
Firmware System

津田 孝夫 , 田中 譲  
Takao TSUDA Yuzuru TANAKA

北海道大学 ・ 工学部  
Hokkaido University

This paper introduces the new type of multimicroprocessor system constructed under the cooperation between the Graduate School of Information Engineering of Hokkaido University and Nippon Mini-Computer Corporation. This system named HARPS consists of 9 microprocessors ( 8 processing elements and 1 control processor ) and operates under the support from a host minicomputer ECLIPSE S/200. All the memory units in HARPS including WCS's and Mapping Arrays are shared with ECLIPSE as a part of its main memory. In this system, the specially designed hierarchical firmware architecture allows for the hierarchical parallel processing flexibility. The operation of this system completely depends on its firmware. MOSES is a firmware operating system as a parallel machine with various parallel instructions.

## [1] 序論

近年、マイクロプロセッサ技術の発展とマイクロプログラムの再評価は、並列処理システムの実現性を高めている。安価なマイクロプロセッサを多数使用したシステム<sup>1)</sup>や、ビット・スライス型のようにモジュラティーの高いプロセッサ<sup>3)</sup>を自由に組み合わせた可変構造型並列計算機等<sup>2)</sup>が発表されている。

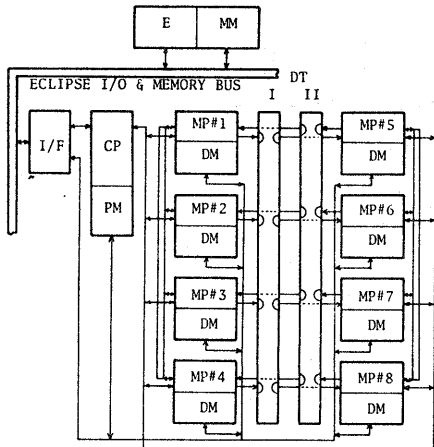
HARPS ( Hokkaido University Array Processor System ) は新しい形式の制御構造を持ったマルチマイクロプロセッサ・システムで、自由度の高い階層制御構造にその特徴がある。このシステムは9個のマイクロプロセッサからなり、1個が制御プロセッサ (CP) として、残りの8個がプロセッシング・エレメント (MP) として働く。HARPSのホスト計算機システムは、ECLIPSE S/200 2台が90MBと1MBのディスクを共有

するデュアル・プロセッサ・システムであり、HARPSは一方のECLIPSEにつながっている。HARPSの各プロセッサのALUは4個の4ビットスライス型マイクロプロセッサAm2901より成り、16ビットの語長を持つ。このシステムの動作はファームウェアに完全に依存し、種々のファームウェア・システムが計画されているが、MOSES ( Microprogram Oriented Supervisory Executive System ) がその1つである。MOSESは、HARPSに汎用並列計算機としての標準的命命群を準備し、ホスト・システムのOSの下で動作させることを最終目標とする。

## [2] HARPSのアーキテクチャ

図1にHARPSの構成を示す。CPは16ビット×256語のプログラム・メモリーと55ビット×256語のWCSを持ち、各々のMPは16ビット

×256語のデータメモリーと、54ビット×256語のWCSを持つ。それぞれのプロセッサは命令デコーダに似たマッピング・アレイを持ち、これは、上位レベルのコマンドを低位レベルのコマンドの集合に写す。特にMPのマッピングアレイは、8個が概念上まとめられ8ビット×256語×8フィールドのNMAを形成している。



E: ECLIPSE S/200, I/F: Interface, CP: Control Processor, MP: Micro-Processor, MM: Main Memory, PM: Program Memory, DM: Data Memory, DT: Data Transporter

Fig. 1. Configuration of HARPS.

## 2.1. ホスト・ミニコンピュータ

HARPSのメモリー・ユニットは、マッピングアレイやWCSも含めECLIPSEによって共有されている。ECLIPSEはこれらのメモリーを自身の主記憶の一部としてアクセスすることが出来る。ECLIPSEはHARPSが実行可能なプログラムを生成し、これをCPのプログラムメモリーに転送し、CPに起動をかけたこのプログラムの実行を開始させる。これにより、CPを通して8台のプロセッサにも起動がかけられる。ローカルメモリーへのまとまったデータの転送、計算結果の格納、WCSやマッピング・アレイの動的書き換えもECLIPSEの仕事である。

## 2.2. インタフェース

ECLIPSEとHARPSの間のデータ、コマンド、ステータス情報等の転送はここを通して行われる。このユニット中には、転送モードレジスタと呼ばれる8ビットの両方向シフトレジスタがあり、ECLIPSEから8台のMPのメモリー・ユ

ニットへのデータの分配形式を指定する。転送モードレジスタの8ビットはそれぞれ8台のMPの各々に対応し、対応するビットが1のMPのメモリーがデータの転送先メモリーとして選ばれ、番地はECLIPSEの格納命令のオペランド部により決定される。これにより、ECLIPSEからHARPSへのデータの同時転送が可能であり、ECLIPSEのI/O命令により転送モードレジスタを左右に回転させ転送モードを変えることが出来る。HARPSからECLIPSEへデータを読み取る際には転送モードレジスタは関係しない。

## 2.3. 制御プロセッサ

CPは通常のミニコンピュータと似たアーキテクチャの16ビットプロセッサで、400nsecのマイクロインストラクション(MIR)サイクルを持つ。ECLIPSEからCPに送られたプログラムは、CPに対する命令(プログラム制御命令)とデータ部とから成り、データの一部はマイクロプロセッサ・アレイのためのコマンド列で、これに属する語はNMAの番地としてNMAに送られる。この番地をNMAポイントと呼ぶ。制御プロセッサはMPアレイの状態を調べることが出来る。

## 2.4. ナ) マッピング・アレイ(NMA)

NMAは8フィールドの並列デコーダである。CPより送られたNMAポイントは8フィールドから成る1つの語を指す。この語の各フィールドは、通常各々番目のMPに対するWCSのあるマイクロルーチンを選び起動をかける。それゆえ、NMAの各語の各フィールドはMPエントリー・ポイントと呼ばれる。

どのようなシステムであるかに依るが、各々のMPは通常40~50個のマイクロルーチンを持つ。この内の1つがMPエントリー・ポイントで指定される。HARPSでは8個のMPエントリー・ポイントの標準的な組み合わせを最大256種類NMAに記憶しておくことが出来る。256が足りない場合は、ECLIPSEによりNMAを部分的に動的に書き換えることも可能である。MP8台の起動は8ビットのセレクトタにより選択的に行われる。セレクトタはプログラムによりプログラム中に与えられ、NMAポイントと一緒にしてマイクロコマンドとしてNMAに送られる。

制御プロセッサは常に実行中のMPの集合を把握している、次のマクロコマンドのセレクト部で選ばれるMPがすべて実行を終了したとき、前のマクロコマンドの実行の完了に関係なく、次のマクロコマンドをNMAに送る。

## 2.5. グローバル・コマンド (GC)

図2にNMAの詳細を示す。

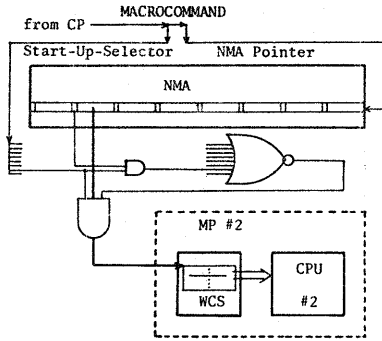


Fig. 2. Parallel expansion of a macro-command by NMA and the function of Global Command indicators.

NMAの各語は、フィールド毎にグローバル・コマンド (GC) 指示ビットと呼ばれる1ビットを持っており、このビットがオンのとき、そのフィールドの内容はMPエントリー・ポインタにはならず、グローバルコマンドであるとされる。選ばれたNMAの1語のセレクト部で選ばれたフィールドの内にGC指示ビットがオンであるものがあるならば、この語はどのフィールドの値もMPには送られない。CPはNMAポインタで選ばれた語のどのフィールドがGCであるかを8ビットの入力ポートを読むことにより識別でき、セレクトに関係なく、この語の各フィールドの値を読むことができる。これらの機能により、HARPSは帰納的かつ階層的な制御構造を実現することができる。これについては3章、4章で述べる。

## 2.6. マイクロプロセッサ・アレイ

各々のMPは16ビットの語長と400nsecのMIRサイクルを持つ。8台のMPは4台ずつ2つのグループに分けられている。8台のMPは同期しているが、各グループ中の4台には位相が少しづつずらされており、MP間データ転送時

の衝突を避けられている。WCSとデータメモリーはECLIPSEにより実行に先立って書き込まれ、実行中に動的に書き換えられる。データメモリーはマイクロ命令中に番地フィールドを持ち、読み出し-演算-書き込みか一つのマイクロ命令で完結する。この番地フィールドの値は、読み出し用、書き込み用のRAMパラメータレジスタの値と加えられて実際の番地を生成する。このレジスタの値はマイクロ命令で随時変更可能である。

8台のMPは、その任意の部分集合において、CPを介することなく自律的に処理の同期を取る機能を有している。

## 2.7. プロセッサ間のデータのやりとり

MP間のデータのやりとりに対しては2つの手段が用意されており、同一グループに属する任意の2台のMP間では、このそれぞれの組み合わせに対し独立に存在するバスとポートによってデータ転送が行われる。これを完備なマトリクスバスと呼ぶ。1つのグループ内の1台のMPが他グループ内の1台にデータを転送する際は、データトランスポートと呼ばれるぐるぐる回る転送装置が使用される。これら2手段により、任意のMPから他のMPへの選択的な同時データ転送が2マイクロ命令サイクルで実行できる。

これら転送装置の利用可能性は各MPで調べることができ、データ転送の衝突はMPのWCSアドレス"0"に割り込みをかける。

## [3] 設計理念とアーキテクチュア

並列という言葉の意味の広範さ、曖昧さのために並列処理の一般的定義を行うことは非常に難しい。計算機における並列処理には種々のレベルが考えられる。HARPSではこのレベルはファームウェアに依存する。HARPSの種々のハードウェア機能の的確な使用をファームウェアにより指定することにより、命令レベルの並列処理からタスクリレベルの並列処理まで広範な種類の並列処理がHARPS上で実行される。

### 3.1. 制御の流れ

HARPSにおける制御はECLIPSEからCPを通って8台のMPへと流れる。流れは下流に行く

に従って流れの幅が広がるが、これに対応して制御を表現するコマンドのレベルは高級言語の1ステートメントからマイクロ命令へと低下していかなければ流れの均衡が取れない。

HARPSは制御の流れを図3に示すような各レベルないしはステージのエントリー・ポインタをレベルの高いものからレベルの低いものへとつなげたエントリー・ポインタの連鎖として実現する。n番目のレベルのエントリー・ポインタとは、(n+1)番目のレベルのエントリー・ポインタの順序集合ないしは(n+1)番目のレベルに対するハードウェアに関するルーチンの先頭、つまりエントリー・ポインタを指すデータである。エントリー・ポインタの順序集合とルーチンをプログラムと総称する。これらの混合もプログラムであるが、特に混合プログラムと呼ぶ。

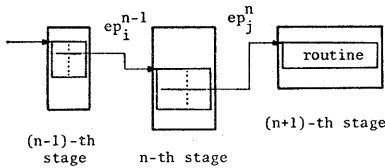
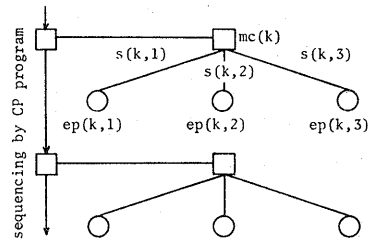


Fig. 3. A chain of entry pointers.

このような制御の流れの捕らええにより、制御の流れの展開は、エントリー・ポインタの並列分解として把握され、NMAという考えが導入された。NMAはエントリー・ポインタの並列展開器であり、これにより、図4に示すような、1レベルの本を1次元の順序に並べたような制御構造が実現される。木の順序はプログラムメモリー中のプログラム制御命令で指定され、通常モードの他、条件分岐、無条件分岐、ループ、サブルーチン・コール等のモードに対するプログラム制御命令が用意されている。図4のような制御構造のみならず、図5に示すような多レベルの本構造の制御を実現するため、グローバル・コマンドが導入された。多レベル本はグローバル・コマンドを新しいNMAポインタと見なし、NMA中でのリンク構造をつくることで実現される。このような構造はプロセッサ数をユーザーに対して透明にするのに役立つ。多レベル本の実行順序はCPのファームウェアに依るが、MOSESでは左から右、下から上である。



mc(k): the k-th macrocommand.  
s(k,i): the i-th bit of the start-up-selector defined in mc(k); this bit is the selector for the i-th MP.  
ep(k,i): the entry pointer for the i-th MP generated by expanding mc(k).

Fig. 4. Parallel control structure with one-level trees for 3 Micro-Processors.

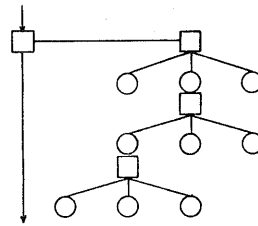


Fig. 5. Multi-level tree structure of the control for 3 Micro-Processors.

MOSESでは、多レベル本を展開するために使用されるセクタはプログラム可能であり、プログラムはシステムのつくった多レベル本の部分木を自由に選んで実行させることができる。これについて4.2で詳述する。

### 3.2. データの流れ

この報告では、データの流れというとき、それは、1つのメモリーから他の1つ以上のメモリーへのデータ集合の転送を指す。メモリーがWCSでデータがマイクロ命令であるときデータの流れとして扱う。このような転送は、ECLIPSEのメイン・メモリーとHARPS中の種々のメモリー（プログラムメモリー、データメモリー、WCS、マッピングアレキ）との間で起こる。MPアレキはMPのメモリーのアレキを形成し、メモリーアレキは記憶構造が2次元以上の配列となり、ECLIPSEのメイン・メモリーが次元が1であるのと異なる。この次元数の差異はストレージマッピングの概念を導入させた。ストレージマッピングというのは、データの流れにおける転送先のメモリーの記憶構造をソースメモリーの記憶構造に写す関数で、す

の形は転送データの論理構造に従って決定される。種々のデータ転送形態、例えば、逐次的同時転送、マトリクス型転送、転置マトリクス型転送、スキュードマトリクス型転送等がストレージマッピングによって統一的に扱える。HARPS設計段階においては、ストレージマッピング専用プロセッサとしてストレージマッピングコントローラ(SMC)を設ける提案もあったが、現在のHARPSではSMCはECLIPSEと転送モードレジスタにより後期的にエミュレートされる。

#### [4] ファームウェア・システムの概要 (MOSES)

MOSESは、HARPS使用の種々の並列処理における標準的処理パターンを機械命令として持つ汎用並列計算機システム実現を目指して作成されたHARPSのファームウェアシステムの名称である。このシステムではHARPSの制御用メモリーの内、9つのWCSと1つのマッピングアレイ、それにNMAは基本的には標準的の制御情報を蓄え動的に書き換えられることはなく、残った制御用メモリーであるCPのプログラムメモリーのみをプログラム領域としてユーザーなソフトウェアによるOSに解放する。CPのプログラムメモリーは256語しかないが、特別な命令を設けて、この命令の指示に従って、プログラムメモリーとECLIPSEのメインメモリーとの間で動的なメモリーの書き換えが可能にされている。

MOSESではマクロコマンドはCPにとってただのデータである。CPはプログラムの流れを変更する命令群、プロセッサアレイの状態チェックを行う命令群、以下に続くデータをマクロコマンドとして順にNMAに送ることを示す命令群を持つ。

NMAには、MPの種々の動作の標準的組み合わせが256組、並列動作として用意されている。MPは整数演算、浮動小数点演算、検索、ソート、論理演算、プロセッサ間データ転送等のルーチンを持つ。

CP、MPのWCSのために特に考えられたルーチン構造と、これを可能にするハードウェアにより、これらの強力なルーチン群を256語という少ない容量の中に効率的に作る事ができる。

る。

MOSESではデータ集合を演算処理の対象とするが、各MPは、何種類かのロード/ストア命令のそれぞれに対し、この集合を決定するオペランドテーブルを2つずつ持つてあり、データ集合の範囲とデータ集合内への掃引モードを指定する。オペランドテーブルの更新は、CPに対して機械語命令で指定でき、CPはこれを受けECLIPSEに割り込みをかけ、ECLIPSE中にあらかじめプログラムによって準備されているオペランドテーブルを対応するMP内に書き込むよう指示する。

#### 4.1. 制御プロセッサの命令群

表1にMOSESにより作られるCP命令の/群を示す。

Table 1. A part of the instruction set of CP

1. SEQ B/W, <u>n</u>	;( Sequence ) Immediately following n bytes or n words are NMA pointers ( NMAP ) or macro-commands, depending on the B/W indicator.
e.g., SEQB, 2 NMAP1, NMAP2 or SEQW, 2 s1, NMAP1 s2, NMAP2	s1, s2: start-up-selector
2. Control Flow Group	
JMP (@) <u>address</u>	;( Jump )
JSB (@) <u>address</u>	;( Jump to Subroutine )
LOOP <u>a</u> , <u>n</u>	;( Repeat the following program segment that ends with the corresponding NEXT for n times.
a: ASCII character	
NEXT <u>a</u>	
3. FORK-JOIN Group	
JOIN <u>selector</u>	;( Freeze CP until all the selected MP's halt.
4. Skip-If-MP's Group	
SAZ <u>selector</u>	;( Skip the next instruction if all the selected MP accumulators are zero.
SNZ <u>selector</u>	;( Skip the next instruction if neither of the selected MP accumulators are zero.
SAP <u>selector</u>	;( positive )
SNP <u>selector</u>	;( positive )
SAN <u>selector</u>	;( negative )
SNN <u>selector</u>	;( negative )
5. Status Check Group	
SST n, <u>selector</u>	;( Set Status ) The n-th bits of selected MP status registers are collected and stored in the high byte of CP status register.
6. Set Register Group	
SSL <u>selector</u>	;( Set Selector Register )
CRM <u>mask</u>	;( Set CR Mask Register )
CRR <u>address</u>	;( Set CR Routine Address Register )

etc.

SEQはプログラムメモリー中のマクロコマン

トを区別するための命令で、SEQが読まれるとCPはそのモードを通常モードからシーケンサモードに変更する。シーケンサモードではB/W指示部の値に従い、以下に続くnバイトないしn語をそれぞれNMAポインタ、マクロコマンドと解釈し、セレクタと共にNMAに送り、MPアレイに起動をかける。実際のマクロコマンドは、"W"モードではマクロコマンドのセレクタフィールドとCP中に用意されているセレクタレジスタの内容の論理ANDがとられ、これを新しいセレクタとして使用する。"B"モードでは、セレクタレジスタの値をセレクタとし、これとNMAポインタからマクロコマンドを生成する。セレクタレジスタはSSL命令によりセットされる。"W"モードにおけるセレクタ指定の重複化はプログラマが自分の書いた同じルーチンを使ったセレクタや何度も使用する際に役に立つ。

シーケンサモードでは、次のマクロコマンドのセレクタが選ぶすべてのプロセッサの実行完了を待って次のマクロコマンドがNMAに送られる。次のマクロコマンドのNMAへの転送の前に、MOSESはCPにMPアレイのハードウェアステータスを調べさせる。それぞれのMPは、実行中、処理完了、CPに対して要求中、同期待ちのうちのどれかの状態を持ち、1MPに対し2ビット、8台に対し16ビットからなる入力ポートを眺むことにより、CPはMPアレイの状態を知る。CPに対する要求は、ユーザーマスクやマスクかけが可能であるようにし、この要求に対する処理ルーチンもユーザーが定義可能となるよう設計されている。

MOSESでは、ユーザーの定義によるCP要求ルーチンなど、自由度の高い制御を可能にするため、CPの処理に関して計9個のスタックが用意される。これらの操作に依るCPのオーバーヘッドを減少させるため、スタック操作の多くはMPが肩代りし、並列処理を行う。この肩代りはMPの処理モードの内の特殊モード(短時間割り込みモード: SIM)内で実行される。SIMはMPの通常モードでの計算の途中結果を壊すことがないよう特に念入りに設計されている。これにより、SIMモード命令は、MPの状態の如何にかかわらず、即座にMPに割り込むことが可能である。

SIMモード命令の他の例はSkip-If-MP's Gel

ープ命令とSST命令である。

#### 4.2. 本構造プログラム

先述のように、GCは制御の多レベル本構造を可能にする。図6に帰納的な多レベル本構造とそのNMAによる表現を示す。

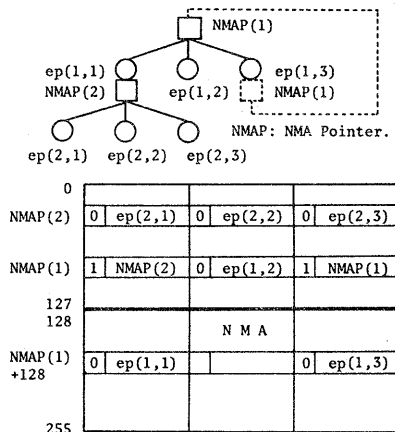


Fig. 6. A multi-level tree and its representation by NMA.

図7は図6のマクロコマンドの呼出し系列とその結果得られる部分木を示す。図6, 図7では説明を簡単にするために、MP3台の場合を扱っている。

```

Calling Sequence
*format: selector/level
SSL 111
SEQW, 1
111, NMAP(1)
101 / 1 ; selector for node B
111 / 1 ; selector for node C
001 / 2 ; selector for node D
xxx / 0 ; terminator of subtree
xxx: don't care
  
```

Subtree

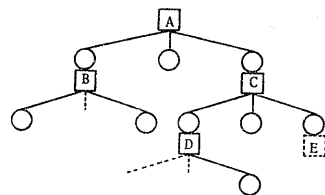


Fig. 7. Calling sequence for the multi-level tree in Fig. 6 and the resultant subtree.

呼出し系列の[111, NMAP(1)]により、ノードAが展開される。これにより、ノードBとノードCの2つのGCが出る。Aの実行は延期される。Bのレベルが多レベル本展開用セレクタのリストの最初のどのレベル指定部に等

しいので、このセレクタを使ってノードBの展開がNMAで行われる。このときはGCが出ないのでBが実行される。この実行はマクロコマンド[101, NMAP(2)]の実行に等しい。セレクタリストの次のセレクタのレベルと、次の実行順になるノードCのレベルが等しいのでノードCが展開される。この展開により2つのGCが現われ、ノードDとノードEが生成される。Cの実行は延期される。セレクタリスト中の次のセレクタ 001/2 はDに使用されDが展開され実行される。次に展開されるべきなのはEであるがセレクタリスト中にこれに対応するセレクタがなくEは展開されない。次に実行が延期され、スタックに蓄えられていたCが実行され最後にAが実行される。上述の処理の制御はマクロコマンドスタックと呼ばれるスタックを使用して行われる。

MOSESでは、NMAによる多レベル木の定義の他に、ユーザーが自由に多レベル木構造をプログラムメモリー中に定義可能にするために特別なプログラム制御命令を用意している。

#### 4.3. 並列動作命令とオペランドのパラメータ化

各々のMPは、CPの処理のためのスタック操作の肩代り部を除いて、ほぼ同じ内容のWCSルーチンを持つ。すべての2項演算命令はレジスタ間演算命令である。MOSESでは各々のMPは論理的に3個の浮動小数点Acc(32ビット)と4個の固定小数点Acc(16ビット)を持つ。物理的には浮動小数点と固定小数点や同じレジスタを共有している(表2)。

Table 2. Accumulators of each MP.

```

ac: fixed-point accumulator
fac: floating-point accumulator

register #0 : ac A      : fac A
          #1 : ac B      : fac B
          #2 : ac C      : fac C
          #3 : ac D      : fac D
          #4 :           :
          #5 :           : fac C

LOAD/STORE instructions

FLA : ( MEM ), ( MEM+1 ) → fac A
FLB : ( MEM ), ( MEM+1 ) → fac B

FST : fac C → MEM, MEM+1
STC : ac C → MEM
STD : ac D → MEM

( MEM: memory )

```

MP エントリー・ポインタにはオペランドアドレスに対する情報がない。そこでオペランドテーブルという概念が導入された。MOSESは2つのロード命令と3つのストア命令を定義するが、この各々に対し、2個のオペランドテーブルが各々のMP毎にデータメモリー中に常駐している。これらのテーブルはECLIPSEにより更新可能である。これら2つのオペランドテーブルに対し、それぞれの命令は2つのエントリー・ポイントを持つ。例えば、浮動小数点Acc Aに対するロード命令FLAにはFLA1とFLA2があり、FLAオペランドテーブルの1, 2に対応している。オペランドテーブルは、ベース番地部、インデックス部、インデックス上限部、第1増分部、第2増分部の5つのフィールドから成り、NMAにより、MPのロード/ストア命令のいずれかに起動がかかった場合、オペランドアドレスは、対応するオペランドテーブルにより、(ベース番地)+(インデックス)と決定される。このルーチンの終りに、インデックスは第1増分だけ加えられ、その結果インデックス上限を越えた場合は"0"にセットされる。それぞれのテーブルを記憶するために、データメモリーの3語つづが使用される。これらのテーブルを処理するためにいくつかのCPに対する命令が用意されている。それらを表3に示す。

Table 3. CP instructions to manipulate operand tables in each MP

```

REQ selector, table name ; Request ECLIPSE to
                           renew the operand table
                           in the selected MP's.

RIX selector, table name ; Reset the index to
                           "0" in the specified
                           tables.

SINC selector, table name ; Swap the 1st
                           increment with the 2nd
                           increment in the specified
                           tables.

```

例として、8x8の行列AとBの積に対するプログラムを以下に示すことにする。まず、ここで使用される並列動作命令のいくつかを説明する。今NMA エントリー・ポインタをその名前を2つのスラッシュで囲むことイテ示し、MPエントリー・ポインタは名前を2つの2重スラッシュで囲むことイテ示すことにする。例えばFLA1に対し、それを//FLA1//と表現される。マトリクスの積では8個の//FM//から成る/FM//と8個の//FA//から成る/FA//を使用する。//FM//は浮動小数点乗算のエント

リーポイントであり、//FA//は浮動小数点加算のエントリーポイントである。MPルーチンの例を表4に示す。

Table 4. Examples of MP routines. (MP #i)

FA	fac A + fac B → fac C
FS	fac A - fac B → fac C ; floating-
FM	fac A * fac B → fac C ; point
FD	fac A / fac B → fac C
A	ac A + ac B → ac C
S	ac A - ac B → ac C
M	ac A * ac B → ac C
D	ac A : ac B → ac C (quotient)
	ac D (residue)

XFK ac A → XFO # (i+k mod 8)  
 RECVK XFI # (i+k mod 8) → ac C

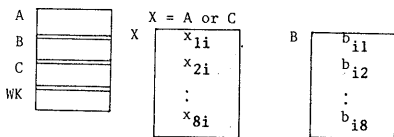
XFO : output port used to transfer data to another MP.

XFI : input port used to receive data from another MP.

/SWAP k/ (k=1, 2, or 4)の指すMPルーチンはMP #iに対し、(i/k)以上の最小整数が奇数であれば、浮動小数点Acc B (fac B)の内容をMP #(i+k)に転送し、次にMP #(i+k)から送られてくるデータをfac Bに格納するという一連の動作を指示する。上述の条件に反する場合、MP #iはfac Aの内容をMP #(i-k)に転送し、次に、MP #(i-k)から転送されたデータをfac Aに格納する。

図8に8x8の行列の積に対するプログラムを示す。このプログラムでは、内積の計算がトリ、ハイト、リダクションと呼ばれるアルゴリズムで計算されている。図8は説明用のプログラムであることに注意しておく。

core map of each MP (MP #i):



operand tables (stored in each MP):

FLA1; A	FLB1; B	FST1; WK
0 / 15	0 / 15	0 / 15
2 / -	2 / -	2 / -
FLA2; WK	FLB2; WK+2	FST2; C
0 / 15	0 / 15	0 / 15
4 / -	4 / -	2 / -

format of operand tables

base address  
 index / upper-bound-of-index  
 1st increment / 2nd increment

program:

SSL 377  
 LOOP A, 8  
 SEQB, 1  
 FLA1.

LOOP B, 8  
 SEQB, 3  
 FLB1, FM  
 FST1,  
 NEXT B

LOOP C, 4  
 SEQB, 5  
 FLB2, FLA2  
 SWAP1, FA  
 FST1,  
 NEXT C  
 RIX 377, FST1

LOOP D, 2  
 SEQB, 5  
 FLB2, FLA2  
 SWAP2, FA  
 FST1,  
 NEXT D  
 RIX 377, FST1  
 RIX 377, FLB2  
 RIX 377, FLA2

SEQB, 5  
 FLB2, FLA2  
 SWAP4, FA  
 FST2,  
 RIX 377, FLB2  
 RIX 377, FLA2  
 NEXT A  
 RIX 377, FST2  
 HLT

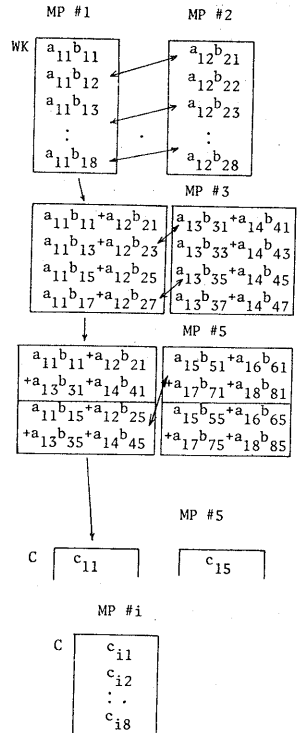


Fig. 8. An example program for 8-by-8 matrix multiplication.

## [5] 結論

現在設計されているMOSESは低いレベルでの並列処理を可能にする機能を用意すべく設計されている。この状況は将来変化する。MOSESの最終的な形はタスクの動的スケジューリング、割り付け等を含むシステムが同指される。現在の形でのMOSESは仕様はほぼ決定しており、マイクロ命令のコート化がかなり進んでいる。現在、ホストのOSの8000のHARPSのファームウェア開発のためのユーティリティが開発中である。

MOSESを基礎とした高級言語に関する研究が行いつつあるが仕様決定はまだである。ここが問題になるのは、処理を受けるデータの論理構造と物理構造の対称、並列処理可能性の解析、並列処理の表現法等である。

HARPSの定量的評価と問題点の追求も大きな



問題であるが、またデータ不足である。

HARPSの特殊な応用も種々考えられつつある。著者等以外によるものも含め列記すると、DDA (ディジタル・ディファレンシャル・アナライザ), FFT, DNNs (ディジタル神経回路網エミュレータ), データベースマシン開発のためのエミュレータ等である。

## [謝辞]

HARPSは多くの方々の参加により実現された実験システムである。特にHARPSのハードウェア開発に大きな力をなされた日本ミニコンピュータ(株)の宮下氏を始め、鈴木、赤城、玉地の各諸氏並びにMMP開発グループの各諸氏、HARPS開発の中心となられた北海道大学工学部の小山、宮本、板内助教授、桃内、恩田両氏、大学院生の阿部、田畑両君に感謝の意を表します。なお、この研究は特定研究経費「マイクロコンピュータ群による汎用高速並列演算システムの研究」より一部補助を受け行われた。

## Bibliography

- 1) Hammersley, J.M. Some speculation on a sense of nicely calculated chances. SIAM Review 16 (1974) pp.237-255.
- 2) Lesser, V.R. Direct emulation of control structures by a parallel micro-computer. SLAC-127, Stanford Linear Accelerator Center, Stanford (1970).
- 3) Rallapalli, K., and Verhofstadt, P. MACROLOGIC-Versatile functional blocks for high performance digital systems. Proc. AFIPS 1975 NCC Vol.44 pp.67-74.
- 4) Tanaka, Y., Miyashita, K., et al. HARPS (Hokkaido University Array Processor System): A new hierarchical array processor system. Proc. EURO-MICRO 1976.