NAL TM-583

ISSN 0452-2982
UDC 681.323

# 航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-583

# 並列計算機のアーキテクチャシミュレータ

原 田 公 一

1988 年 3 月

航 空 宇 宙 技 術 研 究 所
NATIONAL AEROSPACE LABORATORY

# 並列計算機のアーキテクチャシミュレータ*

原田公一**

## 1. はじめに

航空宇宙技術の巨大化は，研究開発の長期化と開発費の高騰をもたらし，それに伴うリスクは耐え難いまでに増大し，その打解策が切実に求められるに至った。このような状況を打破し，従来の方法では解明が困難な複雑な現象を高精度で効率よく解明するキーテクノロジーとして数値シミュレーション技術の重要性が認識されて久しい。この技術は物理工学現象を数学モデルで表わし，それを計算機で解析し，数値を現実の世界に対応させる手法であるため，その技術の進展は計算機の処理性能と密接に関連している。近年のスーパーコンピュータの出現は数値シミュレーション技術を実用の段階へと導き，現在も多くの努力が重ねられている。しかし，より複雑な現象を精度よく，迅速に解析するためには，さらに処理性能（特に演算速度と記憶容量）を大幅に増強した新しいスーパーコンピュータが必要である。

他方，近年の計算機技術の進展は素子の集積度の向上，信頼性の向上に依るところが大きく，記憶容量の増強を可能としたばかりでなく，論理回路の高速化，演算器の改良およびデータ供給能力向上のための様々な改良（メモリ・インタリーブ，キャッシュ・メモリ，ベクトル・レジスタ等）を容易にした。

もとより，演算速度の向上のためには，

(i) 格段に高速な新素子を開発すること

(ii) 演算器を改良すること

(iii) 演算器を増やし，並列化すること

が不可欠である。

(i)についてはガリウム砒素，ジョセフソン素子等が開発されつつあり近い将来に実用化されるであろう。(ii)については加算器の桁上げ（Carry）先取りに始まり，浮動小数点演算回路・掛算器等の開発を経て配列演算等の一連の処理を流れ作業として行うパイプライン方式が開発されている。(iii)については単純な演算器または計算機を多数配列する方式とパイプラインを並列化する方式があり，現在のスーパーコンピュータは後者を採用している。その理由は，パイプラインを構成する回路の稼動率が高く，回路の規模拡大以上に性能向上が可能であり，かつ従来の技術の延長上にあり，蓄積された応用ソフトウェアや計算法に配慮したためと思われる。

しかし，パイプライン方式は乗除加減算等の回路を1クロックで処理できるステップに分割し，それを縦に並べるためパイプライン1個を構成する回路が大きくなり，それを多数並列化するには，なお集積度の向上を待たねばならないのに対し，簡単な演算器や計算機は1チップに収納されるようになってきたため，これらを多数配列して負荷を分散し，全体で処理性能の向上を図る並列計算機がベンチャービジネスと言われる企業から市販されるようになってきた。

今後は，回路の集積度が進み，パイプラインの1チップ化が容易になればなおのこと性能向上策としての並列化は避けられないものと思われる。

しかし，処理性能は応用プログラムの性質に大きく依存し，演算器等の増設と性能向上は同一義ではないことから，単純な計算例を参考にし，机上で並列計算機の性能を論じていたのでは不充分である。

他方，各種多様な方式の並列計算機を開発し，それに応用プログラムを移植し，並列計算機で実行して計測すれば，確実な性能評価はできるが，そのためには多大な経費・期間が必要であり，開発リスクを伴い検討手続としては非現実的で不適切である。

そこで筆者は，将来型スーパーコンピュータの一

*昭和63年1月29日　受付

**数理解析部計算研究室

形態として並列計算機を想定し，航空宇宙の物理工学現象の解明に対するその有用性を以下の方法で検証することを計画した。

(1) 22 MHzのクロックで動作し，2 cm×2 cm×100 cm程度の空間に1 M FLOPSの演算器および128 KBの記憶装置が実装できるものと仮定し，それを128×256台並べ，全体で約32 G FLOPSの処理性能を有する並列計算機を想定する。

(2) 航技研の数値シミュレータで稼動し，並列計算機の動作をクロック単位で模擬するアーキテクチャシミュレータを開発する。

(3) 応用プログラムを並列計算機に適応させ，並列計算機の命令系で書き替える。

(4) 書き替えた応用プログラムをアーキテクチャシミュレータで一命令づつ追跡し，記憶装置に対する読出し／書込要求の競合・演算器の稼動状況を検討し，性能評価を行う。

本稿は並列計算機の動作を模擬するアーキテクチャシミュレータについての報告である。

## 2. 並列計算機の構成

計算機を(1)命令を解読し，制御する制御部(2)演算を行う演算部(3)データを格納する記憶部に分割した場合，並列処理の可能な計算機としては，次の形態が考えられる。

(1) 制御部・演算部から成る処理装置を複数台配し，記憶部を共有する形態（図1）

(2) 制御部・演算部・記憶部から成る計算機を結合し，必要に応じてデータの転送を行う形態（図2）

(3) 複数の演算部と複数の記憶部とを対応づける結合回路を有し，1個の制御部により同一処理を行う形態（図3）

科学技術計算の高速化のためには一連の演算負荷を均一に分散し，並列処理することが必要である。(1)および(2)は互いに同期を取るための負荷が避けられず，処理装置，計算機を増設してもそれに見合う性能向上があまり期待できないのに対し，(3)はハードウェアで完全な同期を取るため，増設に見合う性能向上が期待できる。このため並列計算機のイメージとして(3)の方式を基本として構成を決めることにした。

また航空宇宙技術計算では機体・翼等の周りの流れ場解析が大きな比重を占めており，これらは流れ方向に約200点，境界層方向／翼幅方向に各々約100点に分割することが実用に耐えられる目安となる。

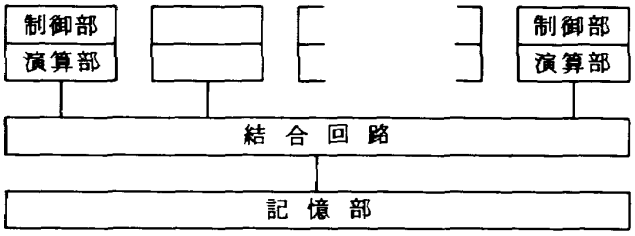



図1 マルチプロセッサ

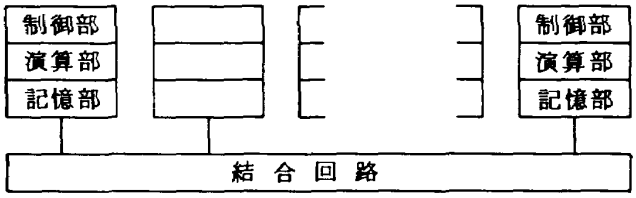



図2 複合計算機

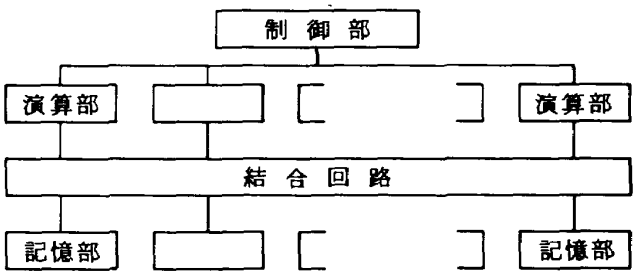



図3 並列計算機

そこでアレイデータの演算では演算器を128×256台設置し，データを供給するメモリを同数接続し，並列演算により処理速度の向上を図るべく，その構成を検討した。

複数の演算器と複数のメモリとを接続するネットワークについては多くの提案が発表されており[1]~[12]，これらを整理検討すると演算器／メモリの台数により接続ケーブル数等の実装上の限界から表1のようになり，128×256台の接続方式として隣接結合が最も合理的である。

また入出力処理およびシステム全体を制御・監視する処理装置をスカラ／アレイデータを演算する処理装置から各々独立して設けることを検討した。

図4に並列計算機の構成を示す。これはインストラクションメモリ部，制御処理部，データ処理部，

表1　要素数と結合方式の目安

| 要　素　数 | 結合方式 |
|---|---|
| ～ $10^2$程度 | Cross point switch等の完全結合 |
| ～ $10^3$程度 | Omega/N-cube network等の多段結合 |
| ～ $10^4$程度 | Shuffle network等の単段結合 |
| それ以上 | 隣接結合 |

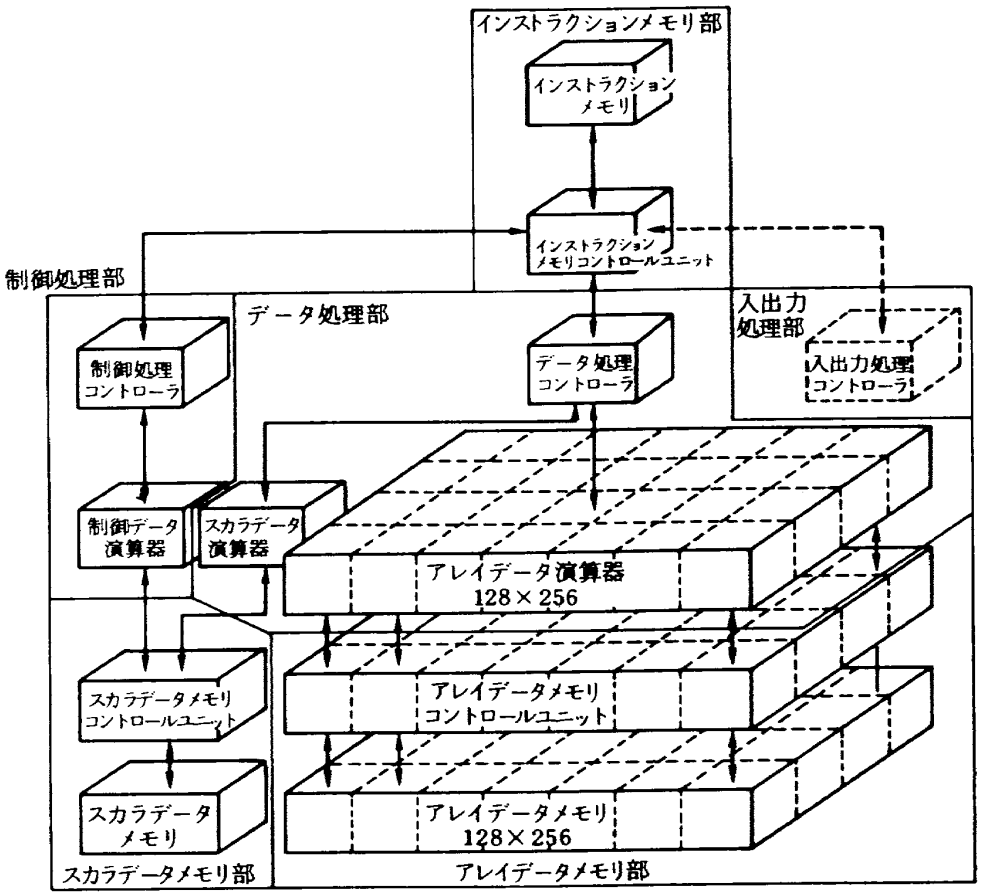



図4　並列計算機の構成

入出力処理部，スカラデータメモリ部，アレイデータメモリ部より成り，以下にその詳細を記す。

(1)　インストラクションメモリ部（図5）

これは制御処理部，データ処理部，入出力処理部の命令を格納するインストラクションメモリとメモリへの読出し／書込みの制御を行うインストラクションメモリコントロールユニットから成る。

(a)　インストラクションメモリ

1語64ビット，最大256K語を想定し，アクセスタイム5クロックのメモリ（INSMEM）と，アドレスおよびデータを一時保持するレジスタ（INSMARおよびINSMBR）より成る。

(b)　インストラクションメモリコントロールユニット

制御処理部，データ処理部および入出力処理部に対応してデータレジスタ（CNTDR 1，CNTDR 2およびCNTDR 3），アドレスレジスタ(CNTAR 1，CNTAR 2およびCNTAR 3）が用意されておりメモリアクセス競合を

1. 入出力処理部
2. データ処理部
3. 制御処理部

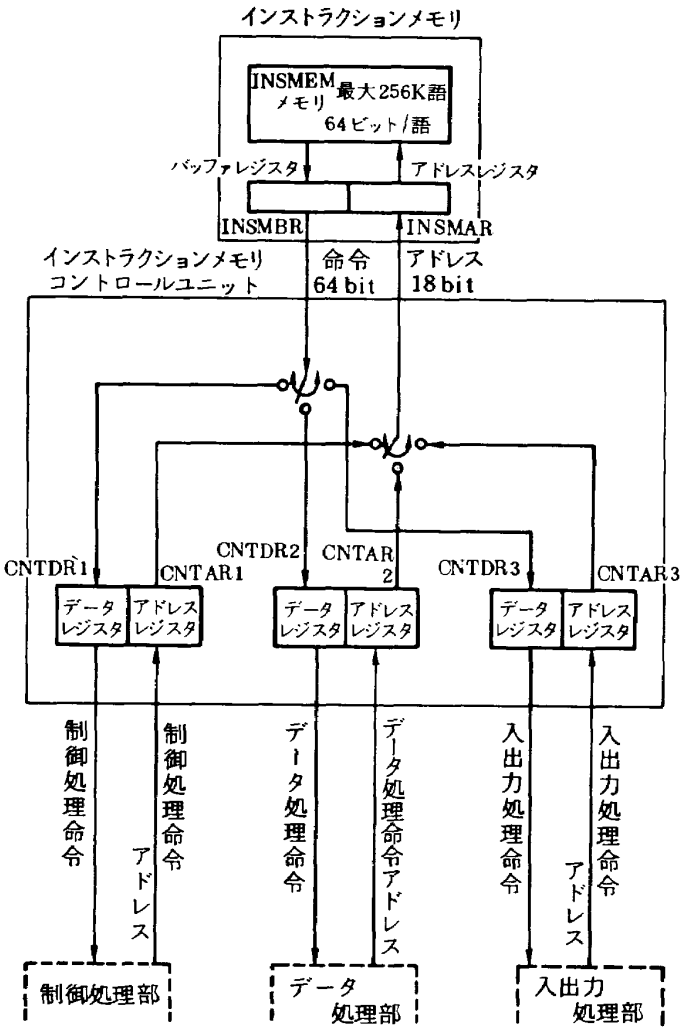



図5 インストラクションメモリ部の構成

の優先順位に従って制御する。

(2) 制御処理部（図6）

これは制御処理コントローラ，制御データ演算器から成り，システム全体を制御・監視するための処理を行う。また後述するデータ処理部と共にスカラデータメモリ部を共有する。

a) 制御処理コントローラ

インストラクションメモリ部に命令読出し要求を出し，命令を解読する。

その構成は命令の記憶番地を保持するプログラムカウンタ（SPC），命令を保持するインストラクションレジスタ（SIR）と命令を読み出し，解読しその実行を進める信号を発信する制御回路より成る。

b) 制御データ演算器

演算器は固定小数点乗除加減算回路（ADDER 1/4），固定小数点演算レジスタ8個（$R1_0$～$R1_7$，このうち$R1_1$～$R1_7$は指標レジスタを兼ねる），データ処理部との間でデータを転送するためのコミュニケーションレジスタ（CMNCR 1/4），アドレス用加算器（ADADR1）および定数発生器より成る。

(3) データ処理部（図7）

これはデータ処理コントローラ，スカラデータ演算器，アレイデータ演算器より成る。

a) データ処理コントローラ

これはインストラクションメモリより送られて来る命令を解読し，スカラデータ演算器/アレイデータ演算器で処理すべきかを判定し，処理を進める。その構成は命令の記憶番地を保持するプログラムカウンタ（ＡＰＣ），命令を保持するインストラクションレジスタ（ＡＩＲ）と命令を読み出し，解読しその実行を進める信号を発信する制御回路より成る。

b) スカラデータ演算器

この演算器は固定／浮動小数点乗除加減算回路（ADDER2/5），固定小数点演算レジスタ8個（$R2_0$～$R2_7$，このうち$R2_1$～$R2_7$は指標レジスタを兼ねる），浮動小数点演算レジスタ8個（$F2_0$～$F2_7$），アレイデータ演算器および制御データ演算器との間でデータを転送するためのコミュニケーションレジスタ（CMNCR 2/5），アドレス用加算器（ADADR2）および定数発生器より成る。

c) アレイデータ演算器

これは128×256台の演算器要素から構成されており，各要素は固定／浮動小数点乗除加減算回路（ADDER 3/6（$k$, $\ell$）），固定小数点演算レジスタ8個（$R3_0(k,\ell)$～$R3_7(k,\ell)$，このうち$R3_1(k,\ell)$～$R3_7(k,\ell)$は指標レジスタを兼ねる），浮動小数点演算レジスタ8個（$F3_0(k,\ell)$～$F3_7(k,\ell)$），マスクレジスタ（MK($k$, $\ell$)），スカラデータ演算器との間でデータを転送するためのコミュニケーションレジスタ（CMNCR 3/6($k$, $\ell$)），アドレス用加算器（ADADR 3($k$, $\ell$)）および定数発生器より成る。なお，各演算器のコミュニケーションレジスタは，CMNCR 1/4とCMNCR 2/5とは1対1で，CMNCR 2/5とCMNCR 3/6（1，1）～CMNCR 3/6（128，256）とはマスクレジスタ制御下のスキャナーを通して1対128×256で接続され，互にデータの送受ができる。

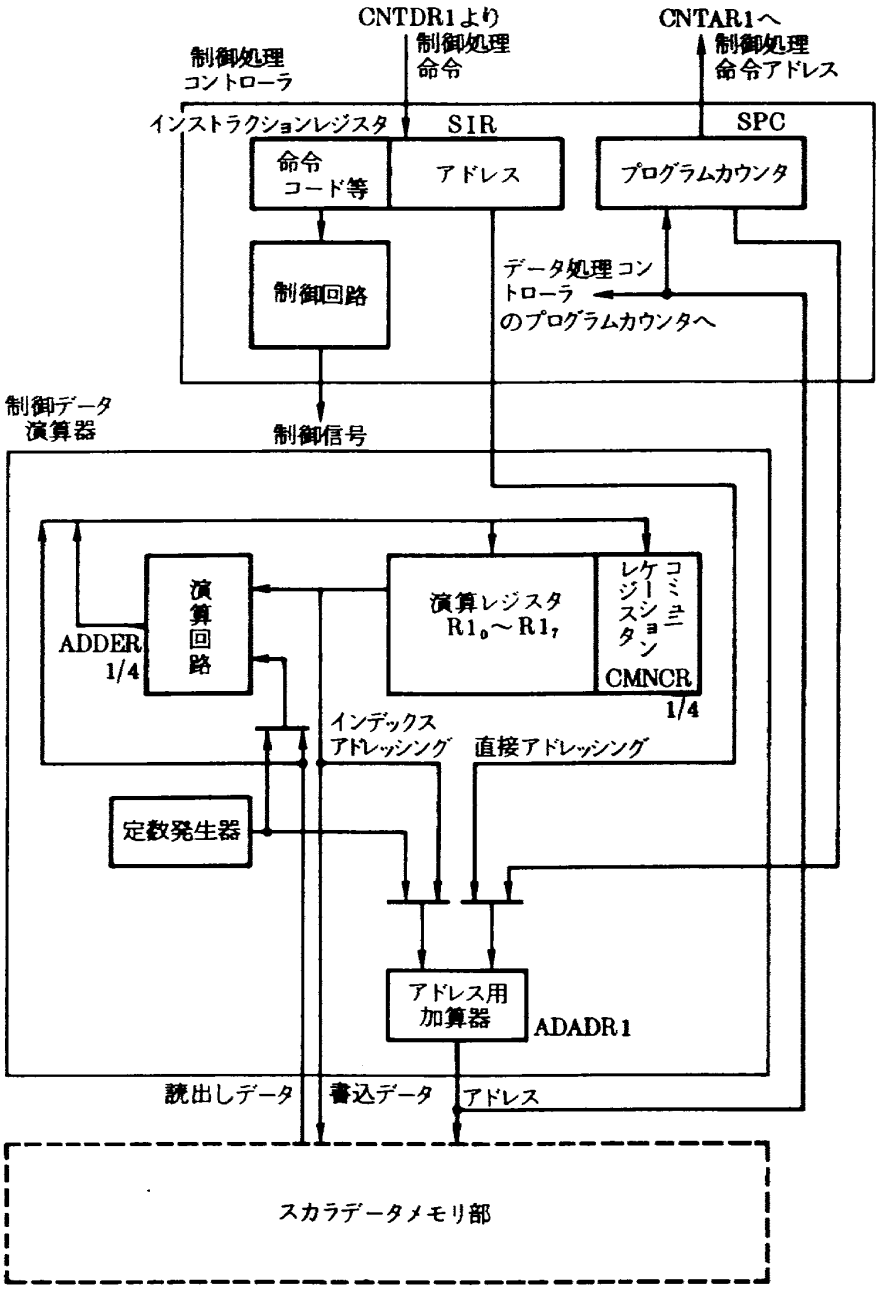



図6　制御処理部の構成

(4)　入出力処理部

並列計算機の性能評価を主題としたため，この部については省略する。

(5)　スカラデータメモリ部（図8）

これは制御処理部およびデータ処理部に共有され，スカラデータを格納するスカラデータメモリと，両処理部とスカラデータメモリとの間にあり，メモリへの読出し／書込みの制御を行うスカラデータメモリコントロールユニットから成る。

(a)　スカラデータメモリ

1語64ビット，最大256K語を想定し，アクセスタイム5クロックのメモリ（SDMEM）と，アドレスおよびデータを一時保持するレジスタ（SDMMARおよびSDMMBR）より成る。

(b)　スカラデータメモリコントロールユニット

制御処理部とデータ処理部に対応してデータレジスタ（DMBDR 1/4およびDMBDR 2/5），アドレスレジスタ（DMBAR 1およびDMBAR 2）が用意されており，両者からのメモリアクセス競合を優先順位

1.　データ処理部

2.　制御処理部

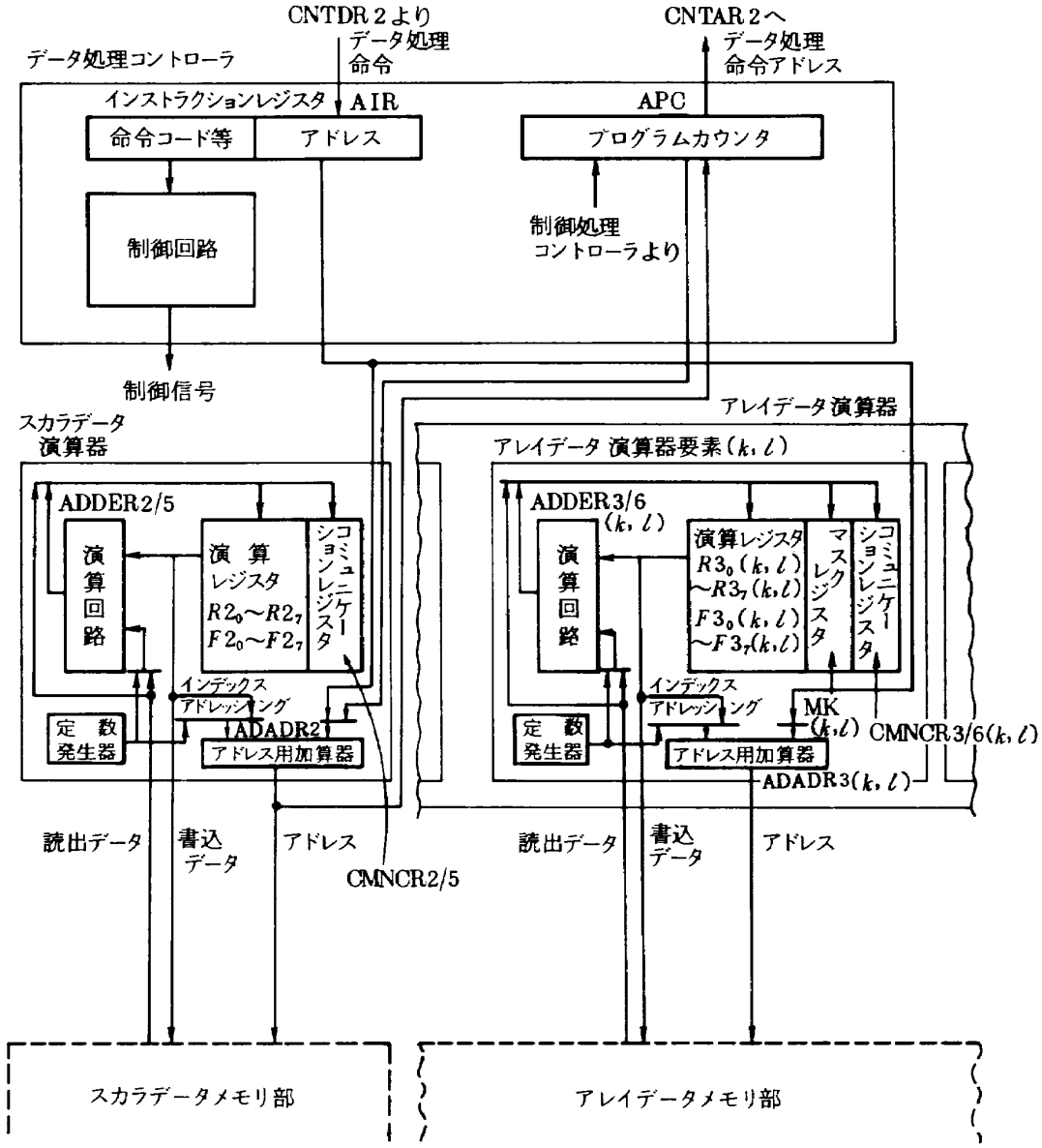



図7 データ処理部の構成

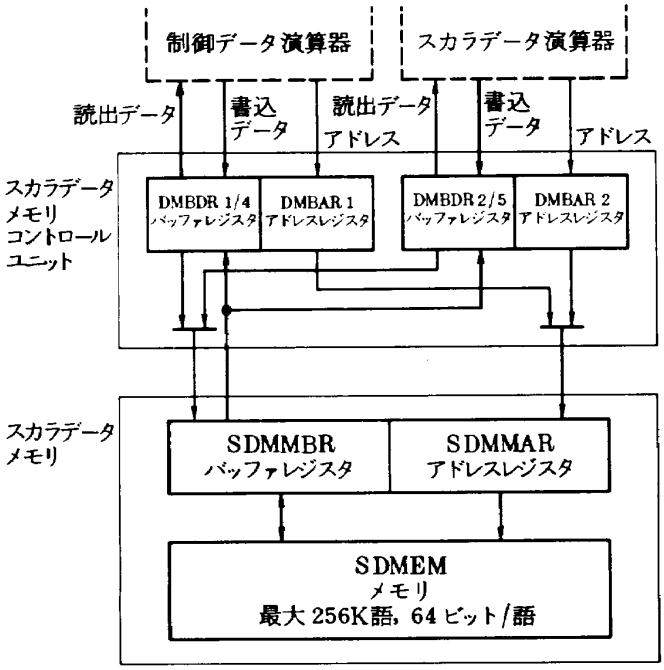



図8 スカラデータメモリ部の構成

に従って制御する。

(6) アレイデータメモリ部(図9)

これはアレイデータを格納するアレイデータメモリと演算器およびメモリとを接続するアレイデータメモリコントロールユニットより成る。

(a) アレイデータメモリ

これは128 × 256個のメモリ要素から構成されており，各要素は1語64ビット，最大16K語を想定した記憶装置(ADMEM(k, l))，およびデータ/アドレスを一時保持するレジスタ(ADMMBR(k, l), ADMMAR(k, l))より成る。メモリアクセスタイムは5クロックである。

(b) アレイデータメモリコントロールユニット

これはアレイデータ演算器要素とアレイデータメモリ要素とを対応づける結合網で，128 × 256対の

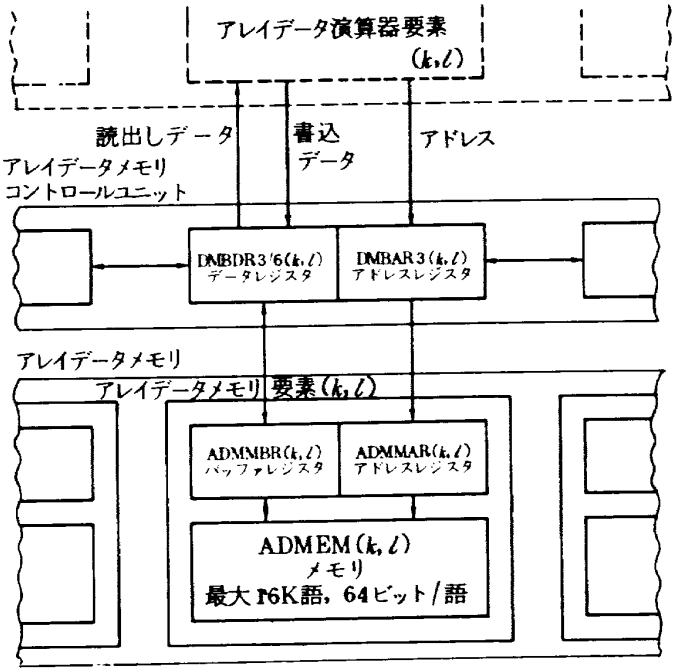



図9　アレイデータメモリ部の構成

データおよびアドレスを保持するレジスタ(DMBDR 3/6 ($k,\ell$) およびDMBAR 3 ($k,\ell$)) を行列各々両方向のシフトレジスタによる隣接結合で構成する (図10)。なお行および列の先端と終端は互いに接続してリングを成す。

メモリ要素からデータを読み出す時には，各演算器要素 ($k,\ell$) よりアドレスをDMBAR 3 ($k,\ell$) に移し，以後データ処理コントローラの制御により1クロック毎に行または列方向に1個だけ移動し，所定の位置 ($k',\ell'$) に到達した後，メモリ要素のADMMAR ($k',\ell'$) に移され読出しが行われる。5クロック後に読み出されたデータはADMMBR ($k',\ell'$) に入り，それをDMBDR 3/6 ($k',\ell'$) に移し，今度は逆方向に要求元の演算器要素の位置 ($k,\ell$) まで行および列を移動させ，データを演算器要素

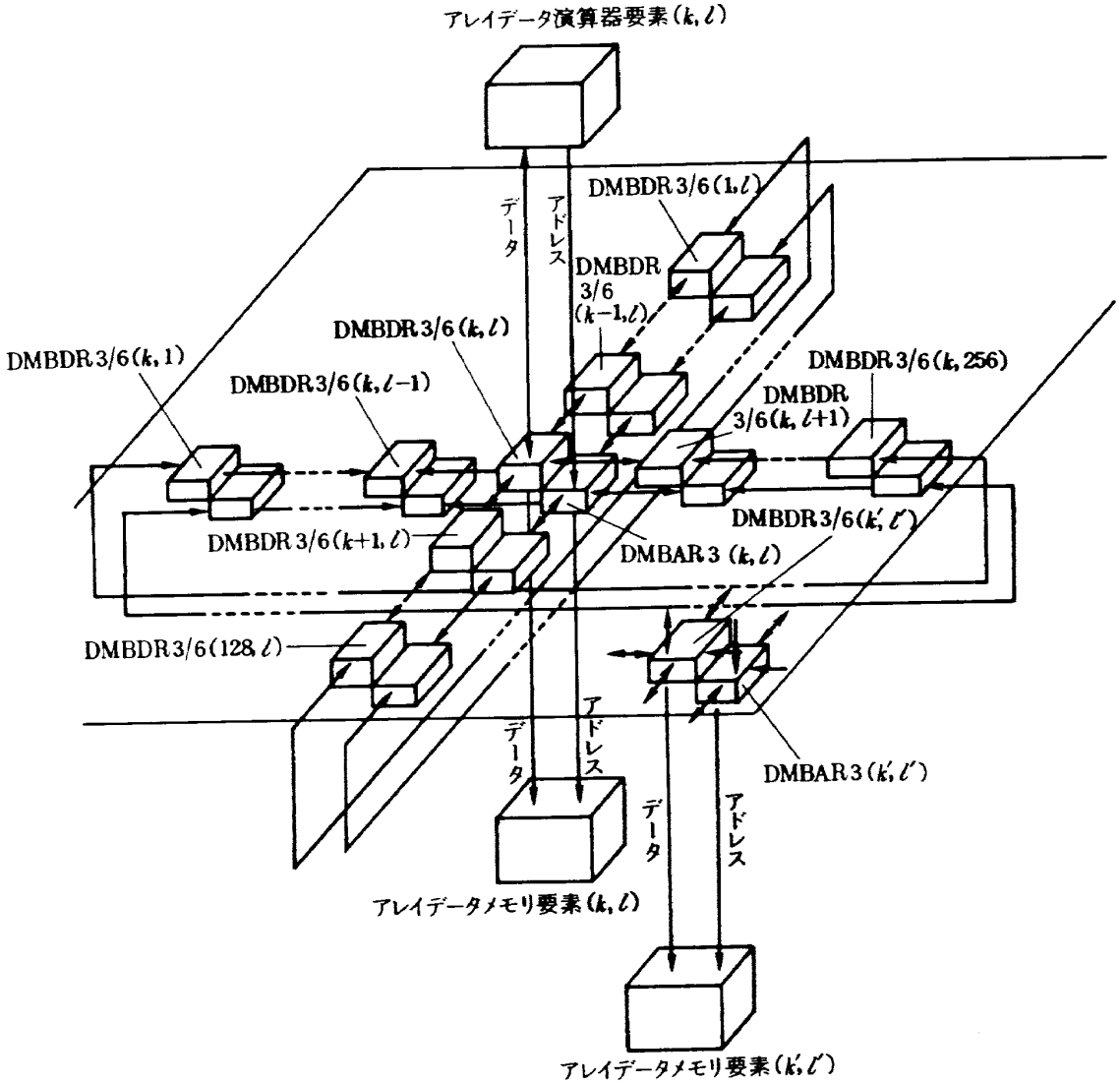



図10　アレイデータメモリコントロールユニットの構成

$(k, \ell)$に移す。データを書込む時にはデータおよびアドレスを演算器要素$(k, \ell)$よりDMBDR 3/6 $(k, \ell)$，DMBAR3 $(k, \ell)$に移し，以後所定の位置$(k', \ell')$まで行列を1個毎移動した後にメモリ要素のADMMBR$(k', \ell')$，ADMMAR$(k', \ell')$に移し，書き込む。

## 3. 命令系とその処理

並列計算機の命令はインストラクションメモリに格納され，各処理部より読み出され，解読・実行される。全命令は64ビットで構成され，表2に制御処理部，表3にデータ処理部の命令を，また略号を以下に示す。

$R/R_i/R_j$；制御データ演算器およびスカラデータ演算器の固定小数点演算レジスタ

$F/F_i/F_j$；制御データ演算器およびスカラデータ演算器の浮動小数点演算レジスタ

$R(k, \ell)$；アレイデータ演算要素$(k, \ell)$の固定小数点演算レジスタ

$F(k, \ell)$；アレイデータ演算要素$(k, \ell)$の浮動小数点演算レジスタ

$X$；アドレス

$\widetilde{X}$；コントロール命令ではインストラクションメモリの実効アドレス，スカラ演算命令ではスカラデータメモリの実効アドレス

$\widetilde{X}(k, \ell)$；アレイデータメモリ要素$(k, \ell)$の実効アドレス

$[\ \ ]$；レジスタおよび記憶装置の内容

$T$；指標修飾

$$\begin{pmatrix} 0 & ; \widetilde{X} = X/\widetilde{X}(k,\ell) = X \\ 1\sim7 & ; \widetilde{X} = X + [R]/\widetilde{X}(k,\ell) \\ & \quad = X + [R(k,\ell)] \end{pmatrix}$$

$C$；演算結果の条件判定（4；零，2；負 1；オーバーフロー（実装せず））

EC；命令実行のマスク条件

$$\begin{pmatrix} 0 & ; マスク状態に関係なく実行 \\ 1 & ; マスクが'ON'の要素のみ実行 \\ 2 & ; 〃 \quad 'OFF'の要素が実行 \\ 3 & ; マスク状態に関係なく実行 \end{pmatrix}$$

MO；条件(C)を満たす時のマスク操作

$$\begin{pmatrix} 0 & ; マスク操作なし \\ 1 & ; マスクを'ON'にする \\ 2 & ; 〃 \quad 'OFF' 〃 \\ 3 & ; マスク操作なし \end{pmatrix}$$

CB；アレイデータメモリコントロールユニットの結合状態（0；隣接結合）

LS；データの移動行数

CS；データの移動列数

SPC；制御処理部のプログラムカウンタ

APC；データ処理部の 〃 〃

CMNCR 1/4；制御データ演算器のコミュニケーションレジスタ

CMNCR 2/5；スカラデータ演算器のコミュニケーションレジスタ

CMNCR 3/6 $(k, \ell)$；アレイデータ演算器のコミュニケーションレジスタ（$k=1\sim128$，$\ell=1\sim256$）

MK$(k, \ell)$；マスクレジスタ（$k=1\sim128$，$\ell=1\sim256$）

全ての命令は厳密にその実行手順がきめられており，一例として図11にアレイデータの加算命令(AA)の実行手順を示す。ここで各タイミングの概略は

タイミング0～7；命令を読出す動作

タイミング8；命令を解読する動作

〃 9；命令を解読し各演算器に制御信号を伝達する動作

〃 10；オペランド番地の計算

〃 11；〃 〃 の設定と結合網の動作によるメモリ要素の選択

タイミング12～17；オペランドの読出し

〃 18；オペランドの逆転送

〃 19～20；配列演算の実行

を示す。またタイミング11～18で次の命令を先取りする様子も示されている。また，メモリの競合が起き，タイミングが進められない時にはメモリが解除されるまで待ちになる。

## 4. アーキテクチャシミュレータの動作

並列計算機の各部動作，効率等を検討するため，このシミュレータは命令を処理する手順をタイミング

## 表2　制御処理部の命令

| | 命令 | 命令記号 | 命令型式 | 内容 |
|---|---|---|---|---|
| コントロール命令 | Jump | J | 1100 0000 \| T \| X | $\widetilde{X}$ → SPC |
| | Jump Minus | JM | 1100 0001 \| R \| T \| X | [R]＜0の時 $\widetilde{X}$ → SPC |
| | Jump Zero | JZ | 1100 0010 \| R \| T \| X | [R]＝0 〃 〃 〃 |
| | Synchronize Jump | SJ | 1100 0101 \| T \| X | データ処理部が動作中の時 $\widetilde{X}$ → SPC |
| | Start Arithmetic Processor | SAP | 1100 0110 \| T \| X | $\widetilde{X}$ → APC とし，データ処理部を起動する。 |
| | Halt and Preceed | HP | 1100 0111 | 制御処理部が停止する。 |
| スカラ演算命令 | Add | A | 0100 0000 \| R \| T \| X | [R] + [$\widetilde{X}$] → R |
| | Subtract | S | 0100 0001 \| R \| T \| X | [R] − [$\widetilde{X}$] → R |
| | Multiply | M | 0100 0010 \| R \| T \| X | [R] × [$\widetilde{X}$] → R |
| | Divide | D | 0100 0011 \| R \| T \| X | [R] / [$\widetilde{X}$] → R |
| | Load | L | 0100 0100 \| R \| T \| X | [$\widetilde{X}$] → R |
| | Transfer | T | 0100 0101 \| R \| T \| X | [R] → $\widetilde{X}$ |
| | Add Register | AR | 0100 0110 \| $R_i$ \| $R_j$ \| C | [$R_i$] + [$R_j$] → $R_i$，Cを満す時次の命令をスキップ |
| | Subtract Register | SR | 0100 0111 \| $R_i$ \| $R_j$ \| C | [$R_i$] − [$R_j$] → $R_i$，〃 〃 |
| | Multiply Register | MR | 0100 1000 \| $R_i$ \| $R_j$ \| C | [$R_i$] × [$R_j$] → $R_i$，〃 〃 |
| | Divide Register | DR | 0100 1001 \| $R_i$ \| $R_j$ \| C | [$R_i$] / [$R_j$] → $R_i$，〃 〃 |
| | Move Register | MV | 0100 1010 \| $R_i$ \| $R_j$ \| C | [$R_j$] → $R_i$，〃 〃 |
| | Load Negative | LN | 0100 1110 \| $R_i$ \| $R_j$ \| C | −[$R_j$] → $R_i$，〃 〃 |
| | Compare | CMP | 0100 1111 \| $R_i$ \| $R_j$ \| C | [$R_i$] − [$R_j$] がCを満たす時次の命令をスキップ |
| | Increase | IC | 0100 0000 \| $R_i$ \| C | [$R_i$] + 1 → $R_i$，Cを満たす時次の命令をスキップ |
| 会話命令 | Read Arithmetic Communication Register | RAC | 1000 0101 | [CMNCR2/5] → CMNCR1/4 |
| | Load Supervisor Communication Register | LSC | 1000 0110 \| R | [CMNCR1/4] → R |
| | Store to 〃 〃 〃 | SSC | 1000 1010 \| R | [R] → CMNCR1/4 |

## 表3　データ処理部の命令（その1）

| | 命令 | 命令記号 | 命令型式 | | | | | | | | | 内容 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0 | 8 | 16 | 24 | 32 | 40 | 48 | 56 | 63 | |
| コントロール命令 | Jump | J | 1100 0000 | | T | X | | | | | | $\widetilde{X} \rightarrow$ APC |
| | Jump Minus | JM | 1100 0001 | R | T | X | | | | | | [R]<0の時 $\widetilde{X} \rightarrow$ APC |
| | Jump Zero | JZ | 1100 0010 | R | T | X | | | | | | [R]=0 〃 〃 〃 |
| | Floating Jump Minus | FJM | 1100 0011 | F | T | X | | | | | | [F]<0 〃 〃 〃 |
| | Floating Jump Zero | FJZ | 1100 0100 | F | T | X | | | | | | [F]−0 〃 〃 〃 |
| | Halt and Preceed | HP | 1100 0111 | | | | | | | | | データ処理部が停止する。 |
| | Mask Initiate | MI | 1100 1000 | | | | | | | | | $0 \rightarrow \{MK(k,\ell);\ k=1\sim128,\ \ell=1\sim256\}$ |
| スカラ演算命令 | Add | A | 0100 0000 | R | T | X | | | | | | $[R]+[\widetilde{X}] \rightarrow R$ |
| | Subtruct | S | 0100 0001 | R | T | X | | | | | | $[R]-[\widetilde{X}] \rightarrow R$ |
| | Multiply | M | 0100 0010 | R | T | X | | | | | | $[R]\times[\widetilde{X}] \rightarrow R$ |
| | Divide | D | 0100 0011 | R | T | X | | | | | | $[R]/[\widetilde{X}] \rightarrow R$ |
| | Load | L | 0100 0100 | R | T | X | | | | | | $[\widetilde{X}] \rightarrow R$ |
| | Transfer | T | 0100 0101 | R | T | X | | | | | | $[R] \rightarrow \widetilde{X}$ |
| | Add Register | AR | 0100 0110 | $R_i$ | $R_j$ | | C | | | | | $[R_i]+[R_j] \rightarrow R_i$，Cを満す時次の命令をスキップ |
| | Subtract Register | SR | 0100 0111 | $R_i$ | $R_j$ | | C | | | | | $[R_i]-[R_j] \rightarrow R_i$，〃 〃 |
| | Multiply Register | MR | 0100 1000 | $R_i$ | $R_j$ | | C | | | | | $[R_i]\times[R_j] \rightarrow R_i$，〃 〃 |
| | Divide Register | DR | 0100 1001 | $R_i$ | $R_j$ | | C | | | | | $[R_i]/[R_j] \rightarrow R_i$，〃 〃 |
| | Move Register | MV | 0100 1010 | $R_i$ | $R_j$ | | C | | | | | $[R_j] \rightarrow R_i$，〃 〃 |
| | Load Negative | LN | 0100 1110 | $R_i$ | $R_j$ | | C | | | | | $-[R_i] \rightarrow R_i$，〃 〃 |
| | Compare | CMP | 0100 1111 | $R_i$ | $R_j$ | | C | | | | | $[R_i]-[R_j]$がCを満たす時次の命令をスキップ |
| | Increase | IC | 0101 0000 | $R_i$ | | | C | | | | | $[R_i]+1 \rightarrow R_i$，Cを満たす時次の命令をスキップ |
| | Floating Add | FA | 0110 0000 | F | T | X | | | | | | $[F]+[\widetilde{X}] \rightarrow F$ |
| | 〃 Subtract | FS | 0110 0001 | F | T | X | | | | | | $[F]-[\widetilde{X}] \rightarrow F$ |
| | 〃 Multiply | FM | 0110 0010 | F | T | X | | | | | | $[F]\times[\widetilde{X}] \rightarrow F$ |
| | 〃 Divide | FD | 0110 0011 | F | T | X | | | | | | $[F]/[\widetilde{X}] \rightarrow F$ |
| | 〃 Load | FL | 0110 0100 | F | T | X | | | | | | $[\widetilde{X}] \rightarrow F$ |
| | 〃 Transfer | FT | 0110 0101 | F | T | X | | | | | | $[F] \rightarrow \widetilde{X}$ |
| | 〃 Add Register | FAR | 0110 0110 | $F_i$ | $F_j$ | | C | | | | | $[F_i]+[F_j] \rightarrow F_i$，Cを満たす時次の命令をスキップ |
| | 〃 Subtract Register | FSR | 0110 0111 | $F_i$ | $F_j$ | | C | | | | | $[F_i]-[F_j] \rightarrow F_i$，〃 〃 |
| | 〃 Multiply Register | FMR | 0110 1000 | $F_i$ | $F_j$ | | C | | | | | $[F_i]\times[F_j] \rightarrow F_i$，〃 〃 |
| | 〃 Divide Register | FDR | 0110 1001 | $F_i$ | $F_j$ | | C | | | | | $[F_i]/[F_j] \rightarrow F_i$，〃 〃 |
| | 〃 Move Register | FMV | 0110 1010 | $F_i$ | $F_j$ | | C | | | | | $[F_j] \rightarrow F_i$，〃 〃 |
| | 〃 Load Negative | FLN | 0110 1110 | $F_i$ | $F_j$ | | C | | | | | $-[F_j] \rightarrow F_i$，〃 〃 |
| | 〃 Compare | FCMP | 0110 1111 | $F_i$ | $F_j$ | | C | | | | | $[F_i]-[F_j]$がCを満たす時次の命令をスキップ |

## 表3　データ処理部の命令（その2）

| | 命令 | 命令記号 | 命令形式 | 内容 |
|---|---|---|---|---|
| アレイ演算命令 | Add Array | AA | 0000 0000 R T EC MO C CB LS CS X | $[R(k,\ell)]+[\widetilde{X}(k,\ell)]\rightarrow R(k,\ell)$，Cを満たす時マスク操作 |
| | Subtruct Array | SA | 0000 0001 R T EC MO C CB LS CS X | $[R(k,\ell)]-[\widetilde{X}(k,\ell)]\rightarrow R(k,\ell)$，〃 |
| | Multiply Array | MA | 0000 0010 R T EC MO C CB LS CS X | $[R(k,\ell)]\times[\widetilde{X}(k,\ell)]\rightarrow R(k,\ell)$，〃 |
| | Divide Array | DA | 0000 0011 R T EC MO C CB LS CS X | $[R(k,\ell)]/[\widetilde{X}(k,\ell)]\rightarrow R(k,\ell)$，〃 |
| | Load Array | LA | 0000 0100 R T EC MO C CB LS CS X | $[\widetilde{X}(k,\ell)]\rightarrow R(k,\ell)$，〃 |
| | Transfer Array | TA | 0000 0101 R T EC MO C CB LS CS X | $[R(k,\ell)]\rightarrow\widetilde{X}(k,\ell)$，〃 |
| | Add Register Array | ARA | 0000 0110 $R_i$ $R_j$ EC MO C | $[R_i(k,\ell)]+[R_j(k,\ell)]\rightarrow R_i(k,\ell)$，Cを満たす時マスク操作 |
| | Subtruct Register Array | SRA | 0000 0111 $R_i$ $R_j$ EC MO C | $[R_i(k,\ell)]-[R_j(k,\ell)]\rightarrow R_i(k,\ell)$，〃 |
| | Multiply Register Array | MRA | 0000 1000 $R_i$ $R_j$ EC MO C | $[R_i(k,\ell)]\times[R_j(k,\ell)]\rightarrow R_i(k,\ell)$，〃 |
| | Divide Register Array | DRA | 0000 1001 $R_i$ $R_j$ EC MO C | $[R_i(k,\ell)]/[R_j(k,\ell)]\rightarrow R_i(k,\ell)$，〃 |
| | Move Register Array | MVA | 0000 1010 $R_i$ $R_j$ EC MO C | $[R_j(F,\ell)]\rightarrow R_i(k,\ell)$，〃 |
| | Load Negative Array | LNA | 0000 1110 $R_i$ $R_j$ EC MO C | $[R_j(k,\ell)]\rightarrow R_i(k,\ell)$，〃 |
| | Compare Array | CMPA | 0000 1111 $R_i$ $R_j$ EC MO C | $[R_i(k,\ell)]-[R_j(k,\ell)]$ がCを満たす時マスク操作 |
| | Increase Array | ICA | 0001 0000 $R_i$ EC MO C | $[R_i(k,\ell)]+1\rightarrow R_i(k,\ell)$，C 〃 〃 |
| | Floating Add Array | FAA | 0010 0000 F T EC MO C CB LS CS X | $[F(k,\ell)]+[\widetilde{X}(k,\ell)]\rightarrow F(k,\ell)$，Cを満たす時マスク操作 |
| | 〃 Subtract Array | FSA | 0010 0001 F T EC MO C CB LS CS X | $[F(k,\ell)]-[\widetilde{X}(k,\ell)]\rightarrow F(k,\ell)$，〃 |
| | 〃 Multiply Array | FMA | 0010 0010 F T EC MO C CB LS CS X | $[F(k,\ell)]\times[\widetilde{X}(k,\ell)]\rightarrow F(k,\ell)$，〃 |
| | 〃 Divide Array | FDA | 0010 0011 F T EC MO C CB LS CS X | $[F(k,\ell)]/[\widetilde{X}(k,\ell)]\rightarrow F(k,\ell)$，〃 |
| | 〃 Load Array | FLA | 0010 0100 F T EC MO C CB LS CS X | $[\widetilde{X}(k,\ell)]\rightarrow F(k,\ell)$，〃 |
| | 〃 Transfer Array | FTA | 0010 0101 F T EC MO C CB LS CS X | $[F(k,\ell)]\rightarrow\widetilde{X}(k,\ell)$，〃 |
| | 〃 Add Register Array | FARA | 0010 0110 $F_i$ $F_j$ EC MO C | $[F_i(k,\ell)]+[F_j(k,\ell)]\rightarrow F_i(k,\ell)$，Cを満たす時マスク操作 |
| | 〃 Subtract Register Array | FSRA | 0010 0111 $F_i$ $F_j$ EC MO C | $[F_i(k,\ell)]-[F_j(k,\ell)]\rightarrow F_i(k,\ell)$，〃 |
| | 〃 Multiply Register Array | FMRA | 0010 1000 $F_i$ $F_j$ EC MO C | $[F_i(k,\ell)]\times[F_j(k,\ell)]\rightarrow F_i(k,\ell)$，〃 |
| | 〃 Divide Register Array | FDRA | 0010 1001 $F_i$ $F_j$ EC MO C | $[F_i(k,\ell)]/[F_j(k,\ell)]\rightarrow F_i(k,\ell)$，〃 |
| | 〃 Move Register Array | FMVA | 0010 1010 $F_i$ $F_j$ EC MO C | $[F_j(k,\ell)]\rightarrow F_i(k,\ell)$，〃 |
| | 〃 Load Negative Array | FLNA | 0010 1110 $F_i$ $F_j$ EC MO C | $-[F_j(k,\ell)]\rightarrow F_i(k,\ell)$，〃 |
| | 〃 Compare Array | FCMPA | 0010 0000 $F_i$ $F_j$ EC MO C | $[F_i(k,\ell)]-[F_j(k,\ell)]$ がCを満たす時マスク操作 |
| 会話命令 | Move Arithmetic Communication Register | MAC | 1000 0000 | [CMNCR2/5]→CMNCR3/6(k,ℓ)（ただしMK(k,ℓ)が'OFF'） |
| | Move Communication Register | MCR | 1000 0001 | MK(k,ℓ)が'OFF'でk,ℓが最小の[CMNCR3/6(k,ℓ)]→CMNCR2/5 |
| | Store to Communication Register | SCR | 1000 0010 R | [R]→CMNCR2/5かつ[R(k,ℓ)]→CMNCR3/6(k,ℓ) |
| | Floating Store to Communication Register | FSCR | 1000 0011 F | [F]→CMNCR2/5かつ[F(k,ℓ)]→CMNCR3/6(k,ℓ) |
| | Read Supervisor Communication Register | RSC | 1000 0100 | [CMNCR1/4]→CMNCR2/5 |
| | Load Communication Register | LCR | 1000 1000 R | [CMNCR2/5]→Rかつ[CMNCR3/6(k,ℓ)]→R(k,ℓ) |
| | Floating Load Communication Register | FLCR | 1000 1001 F | [CMNCR2/5]→Fかつ[CMNCR3/6(k,ℓ)]→F(k,ℓ) |

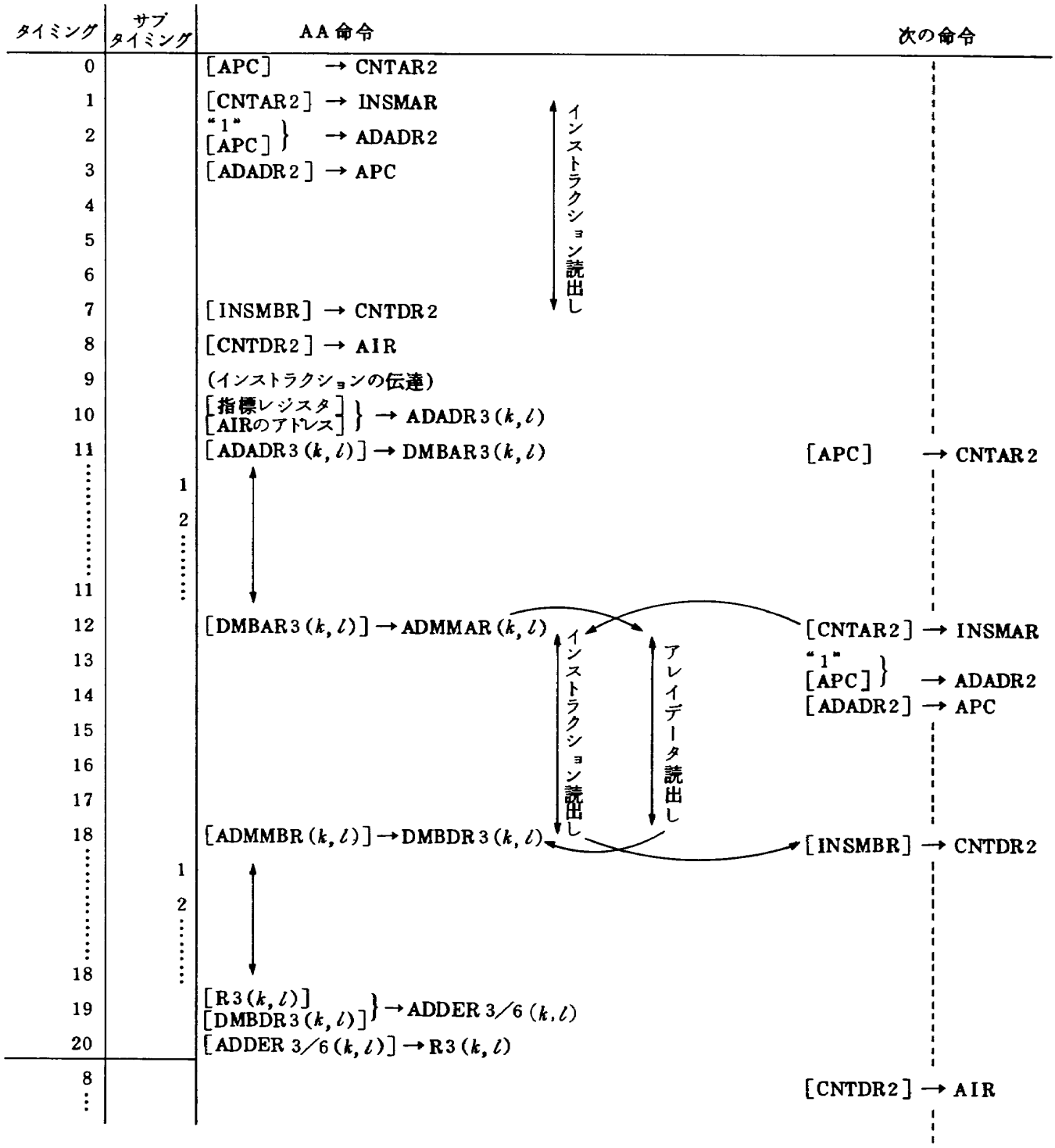
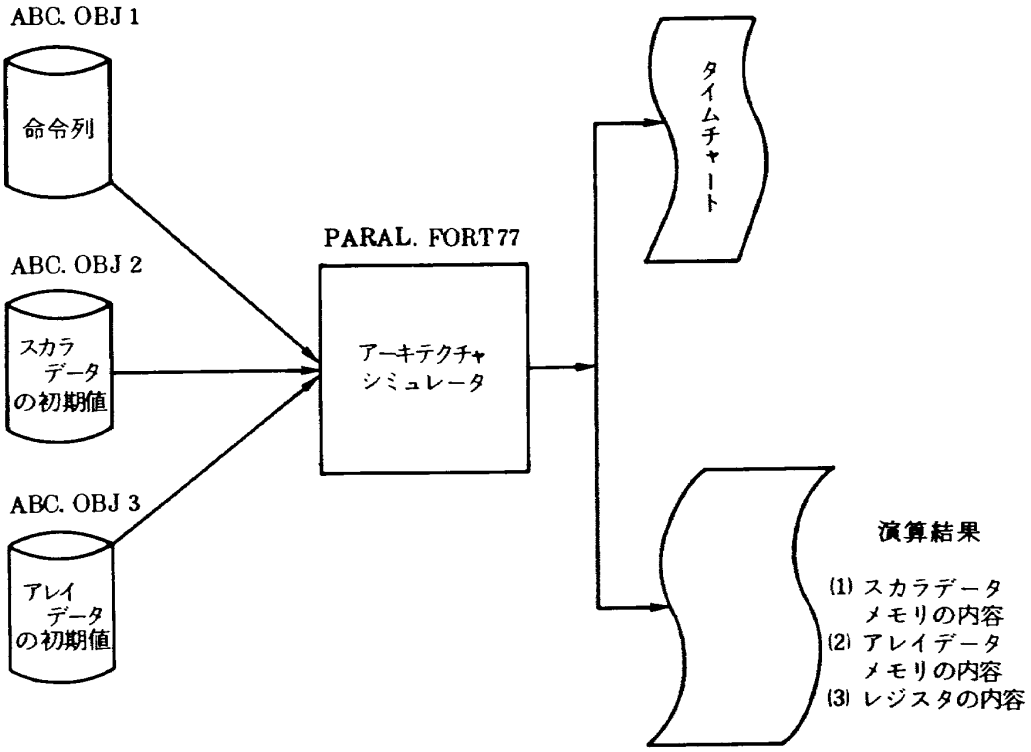



図11 Add Array (AA) 命令のタイムチャート

ごとに忠実に追跡するように設計されている。また命令を実行した計算結果をも得ることができる。

図12にシミュレータの入出力ファイルの関係とジョブ制御文を，図13にその動作の概略を示す。

シミュレータが起動されると命令列，スカラ／アレイデータ初期値が読み込まれた後，制御処理部が起動される。以後内部で刻々と時計を進行させ，それに応じて制御処理部の命令が解読・実行され，演算器・メモリ等の間でデータが移動する状況が模擬される。データ処理部起動命令（SAP）を解読するとデータ処理部へその命令列の先頭アドレスが送られ起動される。以後両処理部は独立して命令列を解読・実行するため，両処理部相互にタイミングに応じた処理を1クロックづつ進める。各自の命令列に停止命令（HP）を解読するとその処理部を停止し，両処理部が停止した時点で模擬は終了する。




アーキテクチャシミュレータのジョブ制御文例

```
// EXEC FORTC,PARAM='NOPRINT',SF='×××.PARAL. FORT77'
// EXEC LIED
// EXEC GO
// EXPAND USDK,RNO=10,FILE='×××.ABC.OBJ1'
// EXPAND USDK,RNO=11,FILE='×××.ABC.OBJ2'
// EXPAND USDK,RNO=12,FILE='×××.ABC.OBJ3(AP010100)'
//
```

図12　アーキテクチャシミュレータの入出力

## 5. 命令追跡例

アーキテクチャシミュレータで各アレイデータメモリ要素の先頭に格納されているデータの内で最大値を求め，スカラデータメモリの先頭に格納するプログラム命令を追跡した。

このアレイデータを

$$\{ a_{i,j} ;\ i = 1 \sim 128,\ j = 1 \sim 256 \}$$

とするとその手順は以下の様である。

① 制御処理部よりデータ処理部を起動する。

② アレイデータ演算器のマスクを全て払う。

③ $k=0$ とした時，$a_{i,j}$ を $a_{i+2^k,j}$ と比較し，小さければマスクを設定する。

④ マスクが設定された要素だけを大きい $a_{i+2^k,j}$ で置き換える。

⑤ $k = 1 \sim 6$ について②～④の操作を繰り返すと $a_{1,j} \sim a_{128,j}$ の最大値が列 $j$ の全要素に格納される。

⑥ 同様の操作を列間で行うと，行列の全要素に最大値が格納される。

⑦ 最大値をアレイデータ演算器のコミュニケーションレジスタを経てスカラデータ演算器のコミュニケーションレジスタに移し，さらに固定小数点演算レジスタを経てスカラデータメモリに格納する。

このプログラムの一部を図14に示す。図中①～⑦は前記の①～⑦に対応する。また

SC　0 および AC　10

は並列計算機のアセンブラの擬似命令で前者は制御処理部の，後者はデータ処理部の各プログラムがインストラクションメモリの0番地および10番地より始まることを示す。また，このプログラムを処理した時のタイムチャートの一部を図15に示す。図中SPC/APCは制御処理部/データ処理部のプログラ

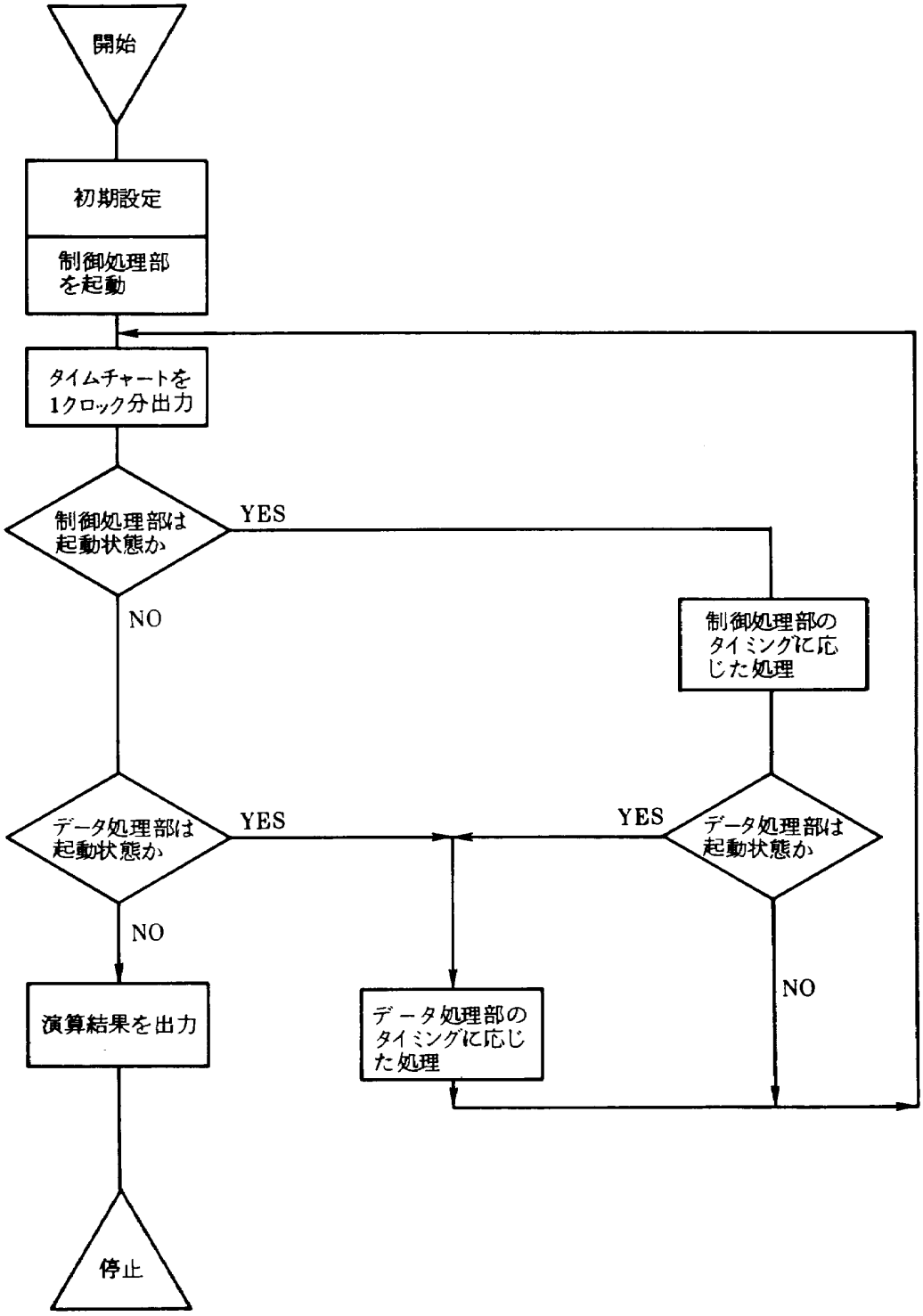
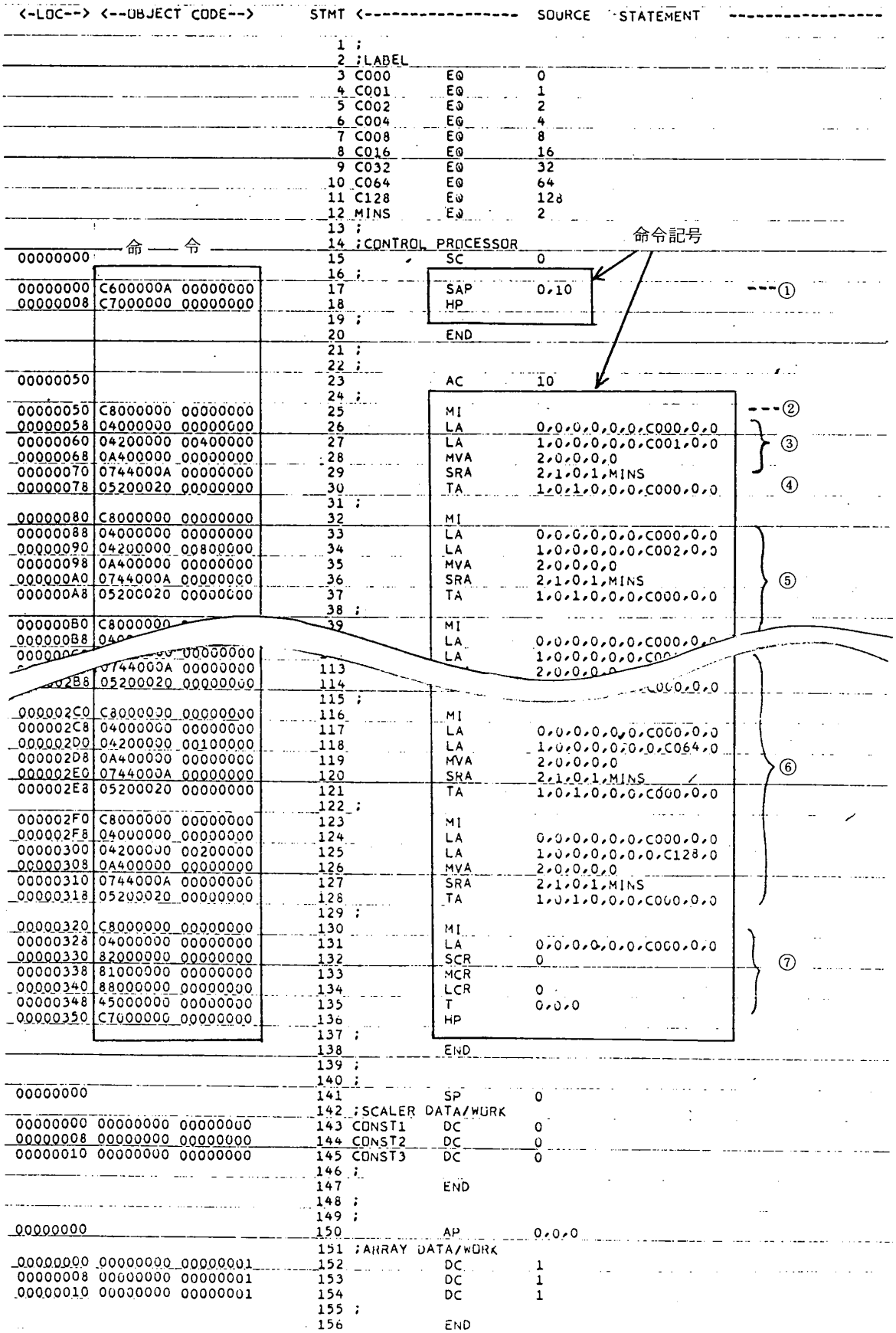



図13 アーキテクチャシミュレータの動作概略




```
<-LOC--> <--OBJECT CODE-->      STMT <---------------- SOURCE STATEMENT ----------------
                                   1 ;
                                   2 ;LABEL
                                   3 C000      EQ      0
                                   4 C001      EQ      1
                                   5 C002      EQ      2
                                   6 C004      EQ      4
                                   7 C008      EQ      8
                                   8 C016      EQ      16
                                   9 C032      EQ      32
                                  10 C064      EQ      64
                                  11 C128      EQ      128
                                  12 MINS      EQ      2
                                  13 ;
                                  14 ;CONTROL PROCESSOR
00000000                          15           SC      0
                                  16 ;
00000000 C600000A 00000000        17           SAP     0,10
00000008 C7000000 00000000        18           HP
                                  19 ;
                                  20           END
                                  21 ;
                                  22 ;
00000050                          23           AC      10
                                  24 ;
00000050 C8000000 00000000        25           MI
00000058 04000000 00000000        26           LA      0,0,0,0,0,0,C000,0,0
00000060 04200000 00400000        27           LA      1,0,0,0,0,0,C001,0,0
00000068 0A400000 00000000        28           MVA     2,0,0,0,0
00000070 0744000A 00000000        29           SRA     2,1,0,1,MINS
00000078 05200020 00000000        30           TA      1,0,1,0,0,0,C000,0,0
                                  31 ;
00000080 C8000000 00000000        32           MI
00000088 04000000 00000000        33           LA      0,0,0,0,0,0,C000,0,0
00000090 04200000 00800000        34           LA      1,0,0,0,0,0,C002,0,0
00000098 0A400000 00000000        35           MVA     2,0,0,0,0
000000A0 0744000A 00000000        36           SRA     2,1,0,1,MINS
000000A8 05200020 00000000        37           TA      1,0,1,0,0,0,C000,0,0
                                  38 ;
000000B0 C8000000                 39           MI
000000B8 040
                                               LA      0,0,0,0,0,0,C000,0,0
                                               LA      1,0,0,0,0,0,C
                                               2,0,0,0,0
                                  113          SRA     2,1,0,1,MINS
                                  114          TA      1,0,1,0,0,0,C000,0,0
                                  115 ;
000002C0 C8000000 00000000        116          MI
000002C8 04000000 00000000        117          LA      0,0,0,0,0,0,C000,0,0
000002D0 04200000 00100000        118          LA      1,0,0,0,0,0,0,C064,0
000002D8 0A400000 00000000        119          MVA     2,0,0,0,0
000002E0 0744000A 00000000        120          SRA     2,1,0,1,MINS
000002E8 05200020 00000000        121          TA      1,0,1,0,0,0,C000,0,0
                                  122 ;
000002F0 C8000000 00000000        123          MI
000002F8 04000000 00000000        124          LA      0,0,0,0,0,0,C000,0,0
00000300 04200000 00200000        125          LA      1,0,0,0,0,0,0,C128,0
00000308 0A400000 00000000        126          MVA     2,0,0,0,0
00000310 0744000A 00000000        127          SRA     2,1,0,1,MINS
00000318 05200020 00000000        128          TA      1,0,1,0,0,0,C000,0,0
                                  129 ;
00000320 C8000000 00000000        130          MI
00000328 04000000 00000000        131          LA      0,0,0,0,0,0,C000,0,0
00000330 82000000 00000000        132          SCR     0
00000338 81000000 00000000        133          MCR
00000340 88000000 00000000        134          LCR     0
00000348 45000000 00000000        135          T       0,0,0
00000350 C7000000 00000000        136          HP
                                  137 ;
                                  138          END
                                  139 ;
                                  140 ;
00000000                          141          SP      0
                                  142 ;SCALER DATA/WORK
00000000 00000000 00000000        143 CONST1   DC      0
00000008 00000000 00000000        144 CONST2   DC      0
00000010 00000000 00000000        145 CONST3   DC      0
                                  146 ;
                                  147          END
                                  148 ;
                                  149 ;
00000000                          150          AP      0,0,0
                                  151 ;ARRAY DATA/WORK
00000000 00000000 00000001        152          DC      1
00000008 00000000 00000001        153          DC      1
00000010 00000000 00000001        154          DC      1
                                  155 ;
                                  156          END
```

図14　最大値を求めるプログラム

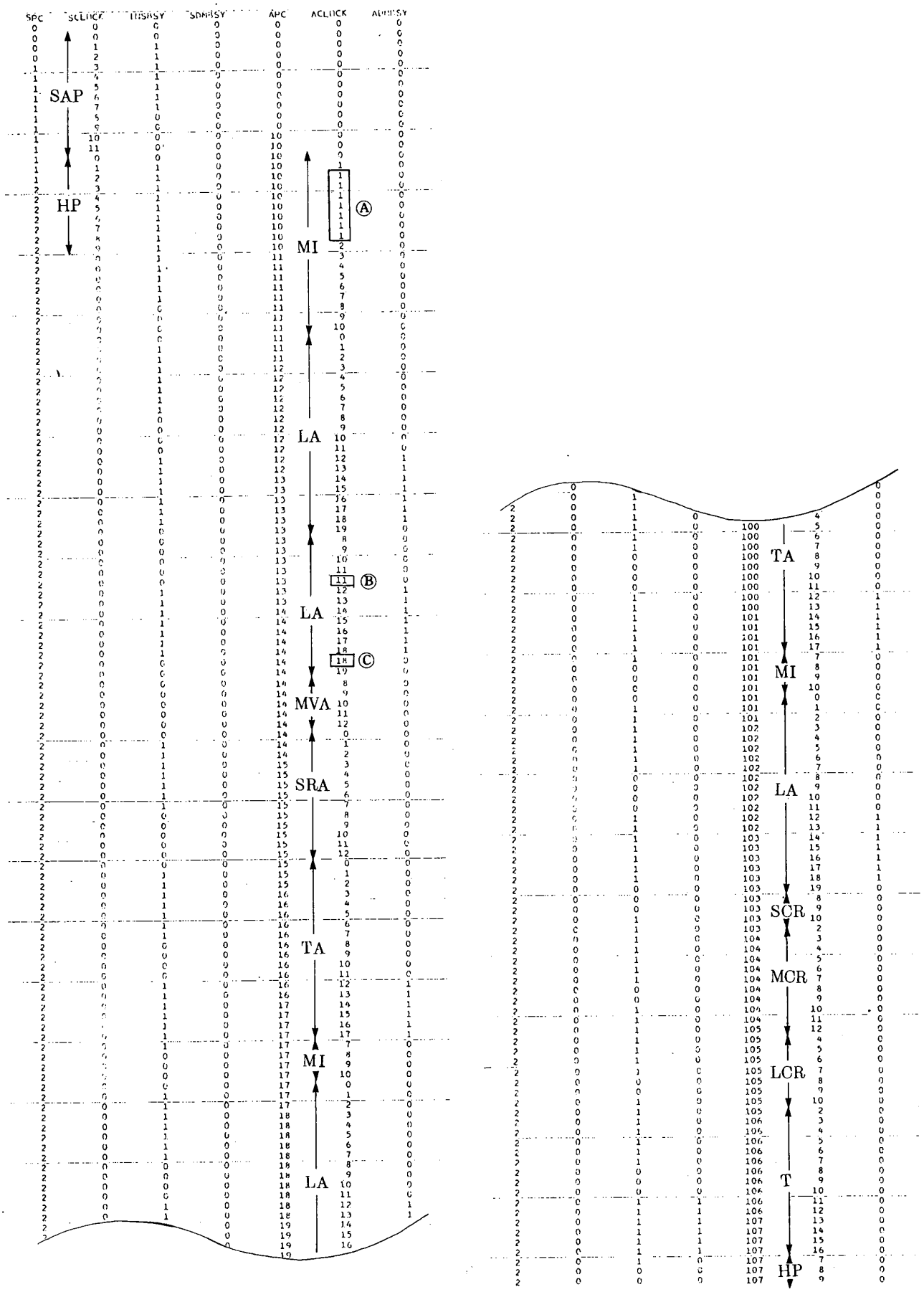



図15　最大値を求めるタイムチャート

ムカウンタの内容を，SCLOCK/ACLOCKは各処理部のタイミングを，INSBSY/SDMBSY/ADMBSYはインストラクションメモリ/スカラデータメモリ/アレイデータメモリの稼動状態を示す。またⒶはインストラクションメモリの競合による待ちを，Ⓑはアドレス情報を目的のアレイデータメモリ要素に転送するタイミングを，Ⓒは読み出したデータを要求元の演算器に返送するタイミングを示す。

## 6. おわりに

アーキテクチャシミュレータは約5000フォートランステートメントから成り，実行に必要な記憶容量は，インストラクションメモリおよびスカラデータメモリを各々$10^3$語，アレイデータメモリ要素各々を100語とすると約30MBである。今後は，実行に際しての様々な制限を考慮しながら各種応用プログラムを並列計算機に適応させ，その命令を追跡することを計画しており，それにより，このシミュレータに実装した各構成および命令系の検討・評価および改良を行う予定である。

最後に，並列計算機の検討に際し，富士通(株)を始めとする多くの方々から有益なる御意見をいただいた。こゝに感謝の意を表する。

### 参考文献

1) Howard Jay Siegel, "A Madel of SIMD Machines and a Camparision of Various Interconnection Network", IEEE Trans. on Camp.,Vol.C-28, №12, December 1979
2) Harold S. Stone, "Parallel Processing with the Perfect Shuffle," IEEE Trans.on Comp., Vol.C-20, №2, February 1971
3) Roger C. Swanson, "Interconnections for Parallel Memories to Unscramble p-Ordered Vectors",IEEE Trans. on Comp., Vol. C-23, №11, November 1974
4) Raphael A. Finkel and Marvin H. Soloman, "Processor Interconnection Strategies", IEEE Trans. on Comp., Vol.C-29, №5, May 1980, p360
5) Kenneth E. Batcher, "The Flip Network in STARAN", Proc. of the 1976. Internatinal Conference on Parallel Processing
6) Marshall C. Pease, "The Indirect Binany n-Cube Microprocessor Array", IEEE Trans.on Comp., Vol. C-26, №5, May1977
7) Sullivan H.T.R.Bashkow and K.Klappholz, "A large scale homogeneous fully distributed parallel machine",Fourth Annual Symposium on Computer Architecture, March 1977, p105
8) Howard Jay Siegel, Robert J. McMillen and Philip T. Mueller JR, "A Sarvay of interconnection methods for reconfigurable parallel processing systems", National Computer Conference, 1979, p529
9) Duncan H. Lawrie, "Access and Alignment of Data in an Array Processor", IEEE Trans. on Comp.,Vol.C-24, №12,December 1975, p1145
10) G. Jack Lipovski, Anand Tripathi, "A Reconfigurable Varistructure Array Processor", Proc. of the 1977 International Conference on Parallel Processing, p165
11) Tse-Yan Feng, "Data Manipulating Functions in Parallel Processors and their Implementations", IEEE Trams. on Comp., Vol.C-23, №3, March 1974, p309
12) 星野，"PAXコンピュータ"，オーム社，昭和60年3月15日。

**航空宇宙技術研究所資料583号**

昭和63年3月発行

発行所　航空宇宙技術研究所
東京都調布市深大寺東町7－44－1
電話三鷹(0422)47-5911(大代表)〒182

印刷所　株式会社　共進
東京都杉並区久我山5－6－17

Printed in Japan