ISSN 0452-2982
UDC 681.323

航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-586

並列計算機のアセンブラ

原田公一

1988年6月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

並列計算機のアセンブラ*

原田公一**

1. はじめに

数値シミュレーション技術は解析プログラムの開発・検証から航空機等の設計へと実用化に向けて年々進歩を重ね，その重要性はますます増大している。

当所は数値シミュレーション技術の重要性を認識し，永年にわたってその技術開発を推進してきたが，より一層の拡大・充実を図をため，昭和62年2月に大型スーパーコンピュータを中核とする数値シミュレータを構築した。

他方，より精密な数学モデルによる航空機等の実形状を扱う計算が可能になるにつれて，数値シミュレーションの需要は急増することが予測されており，これに対処するには格段に高性能なスーパーコンピュータが必要である。

もとより高性能なスーパーコンピュータの開発にはより高速な素子および演算器の開発および演算器等の増強・並列化が必要である。近年の演算素子の進展は目覚ましく，現在では32ビットのマイクロプロセッサが出現し，それらを多数組み合わせた並列計算機が市販されるに至っている。

また現在のスーパーコンピュータはパイプラインを並列化することにより性能向上を達成しようとしており，素子の集積度が向上し，パイプラインの1チップ化が可能になればなおのこと，将来型スーパーコンピュータとしての並列化の傾向は避けられないものと思われる。

しかし，並列計算機の性能は応用プログラムとの整合性が特に重要であり，その評価を適確に行うには応用プログラムを並列計算機に移植し，その命令を追跡して演算器・レジスタ・メモリ等の動作を解明する必要がある。

* 昭和63年3月30日受付

** 数理解析部計算研究室

そこで筆者は，航空宇宙の物理工学現象の解明に対する有用性を検証することを目的として並列計算機の動作を忠実に模擬するアーキテクチャシミュレータを開発した[1)]。

次に，応用プログラムを並列計算機の命令系で書き替えることが必要となったが，直接機械語に変換することは多大の労力が必要であり，誤りも起き易い。また高級言語で記述された応用プログラムを並列計算機の命令系に直接翻訳できれば理想的であるが，このためには並列計算機用のコンパイラを開発しなければならず，並列計算機の有用性を検証する手続きとしてはあまりにも大げさである。そこで前記二通りの手段の中間であるアセンブラを開発することを計画した。

本稿は並列計算機のアセンブラについての報告である。

2. 並列計算機の構成と命令系

2.1 構　成

並列計算機の構成を図1に示す。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，スカラデータメモリ部，アレイデータメモリ部より成る。なお，この他に入出力処理部があるが，今回は省略する。

(1) インストラクションメモリ部

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し，各処理部からのメモリアクセス要求に応じて命令を転送する。1語64ビット，最大256K語を想定している。

(2) 制御処理部

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スカラデータメモリを共有する。

制御処理コントローラは命令を解読し実行するための回路の他，命令の番地を保持するプログラ

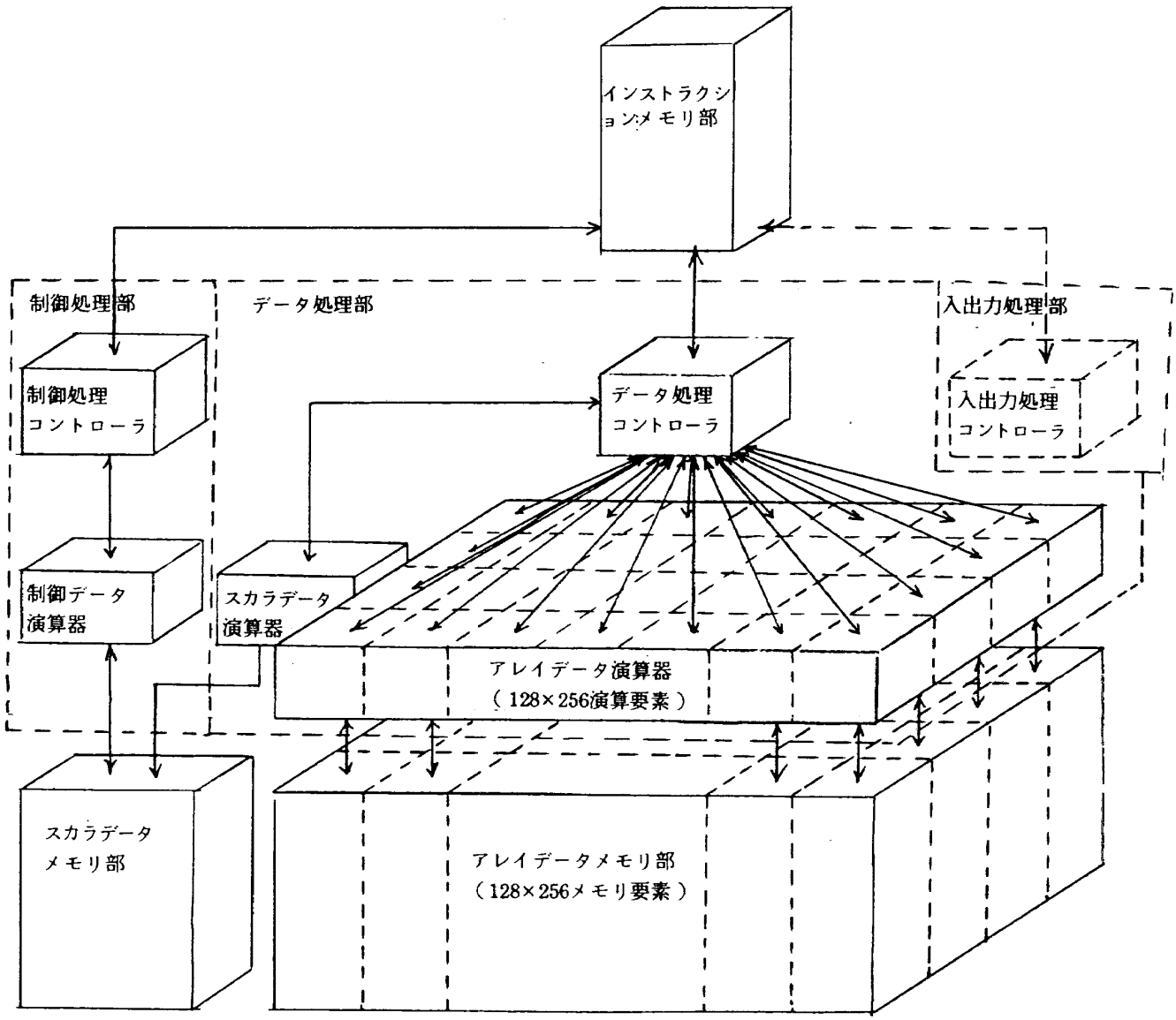


図1 並列計算機の構成

ムカウンタ (SPC) を有する（図2）。

制御データ演算器は乗除加減算回路の他，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会話をするためのコミュニケーションレジスタ (CMNCR 1/4) がある。

(3) データ処理部

数値シミュレーションを行い，データ処理コントローラ，スカラデータ演算器およびアレイデータ演算器より成る。

データ処理コントローラは命令を解読し，実行のための制御信号を発信する回路の他，命令の番地を保持するプログラムカウンタ(APC)を有する。

スカラデータ演算器は乗除加減算回路の他，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 ($F2_0$～$F2_7$) および制御データ演算器とアレイデータ演算器との間で会話をするためのコミュニケーションレジスタ (CMNCR 2/5) がある。

アレイデータ演算器は128×256個の演算器要素から成り，各演算器要素 (k, l) の内には乗除加減算回路の他，固定小数点演算レジスタ8個（$R3_0$(k, l)～$R3_7$(k, l)，このうち$R3_1$(k, l)～$R3_7$(k, l)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, l)～$F3_7$(k, l))，マスクレジスタ1個(MK(k, l))，スカラデータ演算器との間で会話するためのコミュニケーションレジスタ(CMNCR

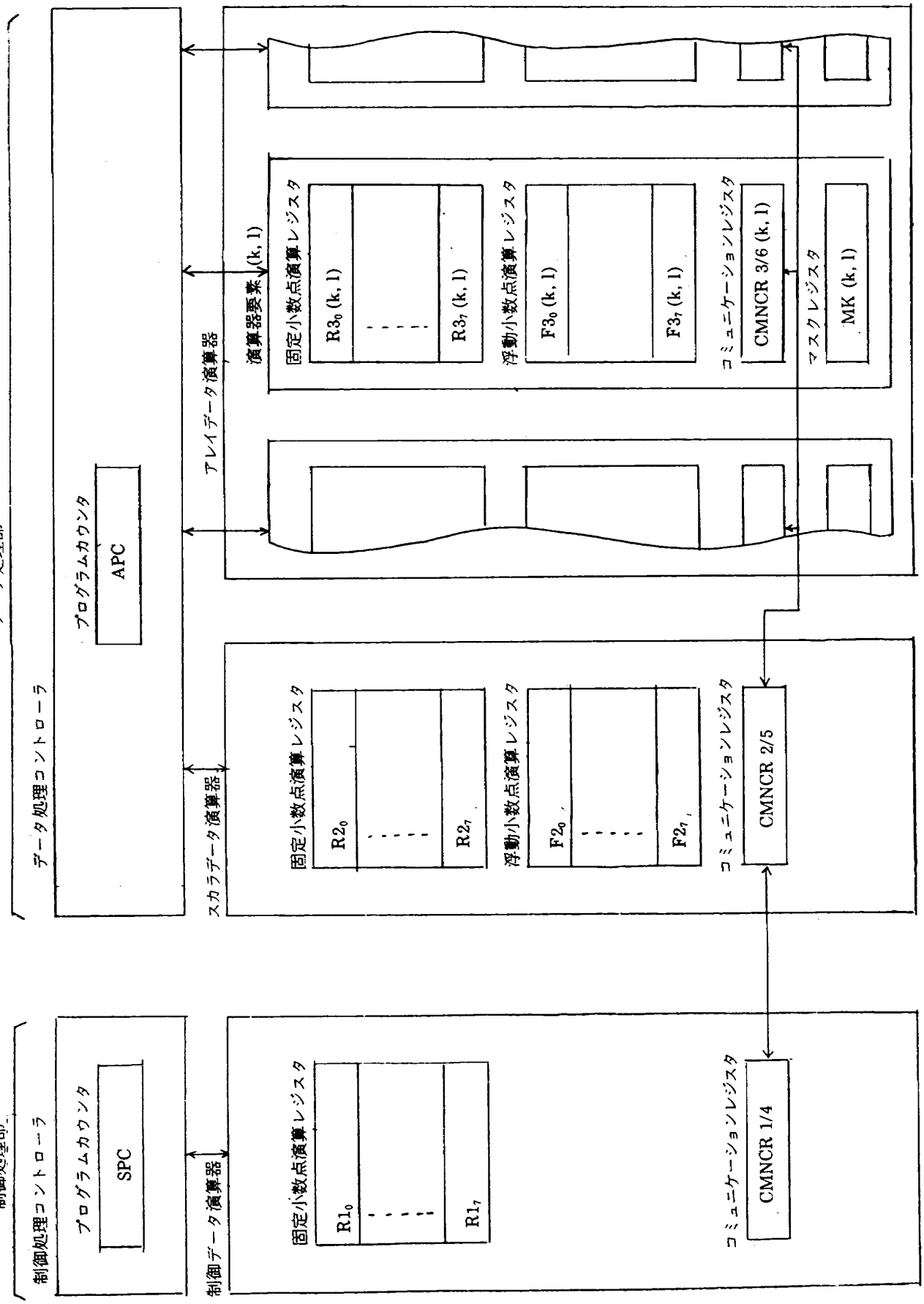


図 2　制御処理部 / データ処理部のレジスタ

表1 制御処理部の命令

	命令	記号命令記述型式	0 8 16 24
コントロール命令	Jump	J T, X	1100 0000 T X
	Jump Minus	JM R1i, T, X	1100 0001 R1i T X
	Jump Zero	JZ R1i, T, X	1100 0010 R1i T X
	Synchronize Jump	SJ T, X	1100 0101 T X
	Start Arithmetic Processor	SAP T, X	1100 0110 T X
	Halt and Preceed	HP	1100 0111
スカラ演算命令	Add	A R1i, T, X	0100 0000 R1i T X
	Subtract	S R1i, T, X	0100 0001 R1i T X
	Multiply	M R1i, T, X	0100 0010 R1i T X
	Divide	D R1i, T, X	0100 0011 R1i T X
	Load	L R1i, T, X	0100 0100 R1i T X
	Transfer	T R1i, T, X	0100 0101 R1i T X
	Add Register	AR R1i, R1j, C	0100 0110 R1i R1j
	Subtract Register	SR R1i, R1j, C	0100 0111 R1i R1j
	Multiply Register	MR R1i, R1j, C	0100 1000 R1i R1j
	Divide Register	DR R1i, R1j, C	0100 1001 R1i R1j
	Move Register	MV R1i, R1j, C	0100 1010 R1i R1j
	Load Negative	LN R1i, R1j, C	0100 1110 R1i R1j
	Compare	CMP R1i, R1j, C	0100 1111 R1i R1j
	Increase	IC R1i, C	0101 0000 R1i
会話命令	Read Arithmetic Communication Register	RAC	1000 0101
	Load Supervisor Communication Register	LSC R1i	1000 0110 R1i
	Store to Supervisor Communication Register	SSC R1i	1000 1010 R1i

3/6 (k, l) がある。

(4) スカラデータメモリ部

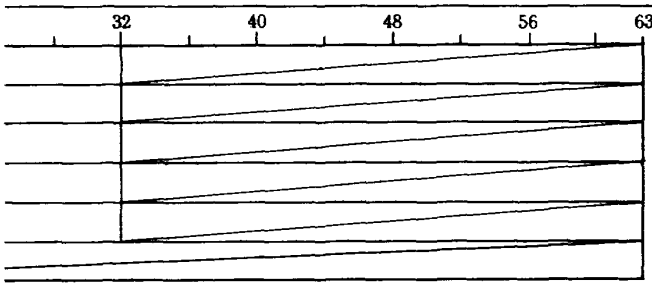
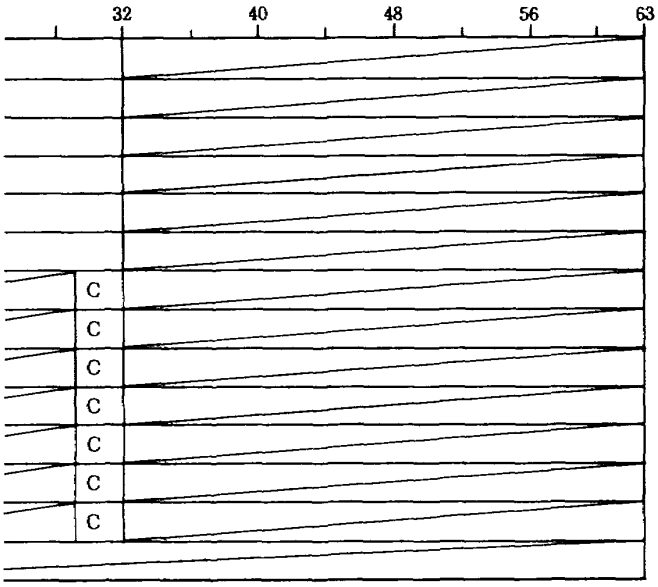
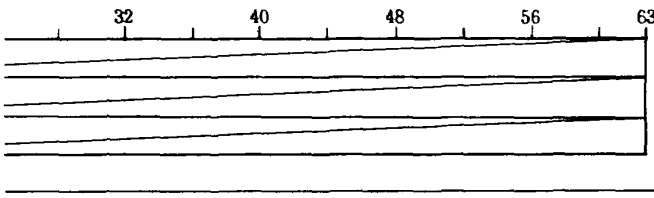
1語64ビット，最大256K 語を想定し，制御処理部およびデータ処理部のスカラデータを格納する。

(5) アレイデータメモリ部（図3）

これは128×256個のアレイデータメモリ要素（各要素は1語64ビット，最大16K語を想定している）と，アレイデータ演算器の各演要素を対応づけ，かつデータを128×256の二次元隣接結合で移動できる結合網で構成されている[1]。図3にその様子を示す。図中 DMBDR 3/6 (k, l) および DMBAR 3 (k, l) はアレイデータ演算器要素 (k, l) とアレイデータメモリ要素 (k, l) との中間にあり，データおよびアドレスを保持するレジスタである。

2.2 並列計算機の命令系

並列計算機の命令は制御処理部とデータ処理部に分かれているが共に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1に制御処理部の，表2にデータ処理部の命令を，また略号を以下に示す。

命令型式	内容
	$\widetilde{X}$→SPC [R1i]<の時$\widetilde{X}$→SPC [R1i]=0の時$\widetilde{X}$→SPC データ処理部が動作中の時$\widetilde{X}$→SPC $\widetilde{X}$→APCとしデータ処理部を起動する。 制御処理部が停止する。
	[R1i]+[$\widetilde{X}$]→R1i [R1i]−[$\widetilde{X}$]→R1i [R1i]×[$\widetilde{X}$]→R1i [R1i]/[$\widetilde{X}$]→R1i [$\widetilde{X}$]→R1i [R1i]→$\widetilde{X}$ [R1i]+[R1j]→R1i, Cを満たす時次の命令をスキップ [R1i]−[R1j]→R1i, Cを満たす時次の命令をスキップ [R1i]×[R1j]→R1i, Cを満たす時次の命令をスキップ [R1i]/[R1j]→R1i, Cを満たす時次の命令をスキップ [R1j]→R1i, Cを満たす時次の命令をスキップ −[R1j]→R1i, Cを満たす時次の命令をスキップ [R1i]−[R1j]がCを満たす時次の命令をスキップ [R1i]+1→R1i, Cを満たす時次の命令をスキップ
	[CMNCR 2/5]→CMNCR 1/4 [CMNCR 1/4]→R1i [R1i]→CMNCR 1/4

R1i/R1j ；制御データ演算器の固定小数点演算レジスタ

R2i/R2j ；スカラデータ演算器の固定小数点演算レジスタ

R3i (k, l)/R3j (k, l)；アレイデータ演算器の固定小数点演算レジスタ

F2i/F2j ；スカラデータ演算器の浮動小数点演算レジスタ

F3i (k, l)/F3j (k, l)；アレイデータ演算器の浮動小数点演算レジスタ

T　；指標修飾

C　；演算結果の条件判定

（6；零負，4；零，2；負）

EC　；命令実行のマスク条件

- 0；マスクの状態に関係なく実行
- 1；マスクが'ON'の要素が実行
- 2；マスクが'OFF'の要素が実行
- 3；マスクの状態に関係なく実行

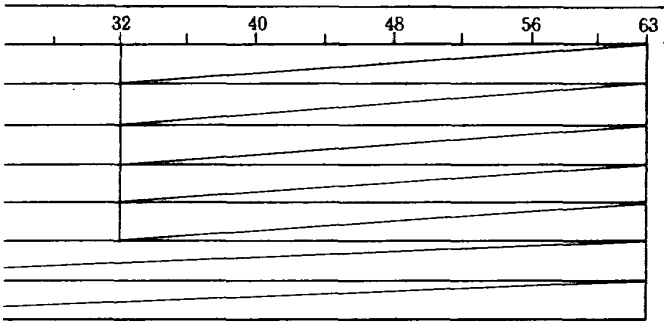
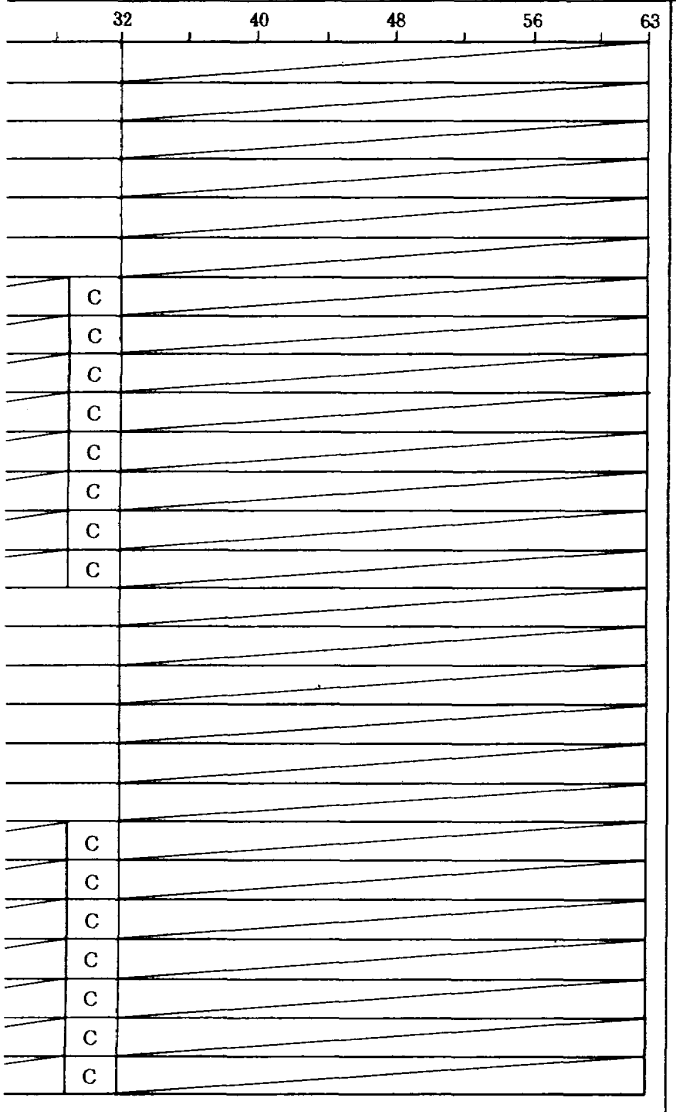
MO　；条件(C)を満たす時のマスク操作

- 0；マスク操作なし
- 1；マスクを'ON'にする
- 2；マスクを'OFF'にする
- 3；マスク操作なし

CB　；アレイデータメモリ部の結合状態

表2 データ処理部の命令

	命令	記号命令記述型式	0 8 16 24
コントロール命令	Jump	J T, X	1100 0000 T X
	Jump Minus	JM R2i, T, X	1100 0001 R2i T X
	Jump Zero	JZ R2i, T, X	1100 0010 R2i T X
	Floating Jump Minus	FJM F2i, T, X	1100 0011 F2i T X
	Floating Jump Zero	FJZ F2i, T, X	1100 0100 F2i T X
	Halt and Preceed	HP	1100 0111
	Mask Initiate	MI	1100 1000
スカラ演算命令	Add	A R2i, T, X	0100 0000 R2i T X
	Subtruct	S R2i, T, X	0100 0001 R2i T X
	Multiply	M R2i, T, X	0100 0010 R2i T X
	Divide	D R2i, T, X	0100 0011 R2i T X
	Load	L R2i, T, X	0100 0100 R2i T X
	Transfer	T R2i, T, X	0100 0101 R2i T X
	Add Register	AR R2i, R2j, C	0100 0110 R2i R2j
	Subtract Register	SR R2i, R2j, C	0100 0111 R2i R2j
	Multiply Register	MR R2i, R2j, C	0100 1000 R2i R2j
	Divide Register	DR R2i, R2j, C	0100 1001 R2i R2j
	Move Register	MV R2i, R2j, C	0100 1010 R2i R2j
	Load Negative	LN R2i, R2j, C	0100 1110 R2i R2j
	Compare	CMP R2i, R2j, C	0100 1111 R2i R2j
	Increase	IC R2i, C	0101 0000 R2i
	Floating Add	FA F2i, T, X	0110 0000 F2i T X
	Floating Subtract	FS F2i, T, X	0110 0001 F2i T X
	Floating Multiply	FM F2i, T, X	0110 0010 F2i T X
	Floating Divide	FD F2i, T, X	0110 0011 F2i T X
	Floating Load	FL F2i, T, X	0110 0100 F2i T X
	Floating Transfer	FT F2i, T, X	0110 0101 F2i T X
	Floating Add Register	FAR F2i, F2j, C	0110 0110 F2i F2j
	Floating Subtract Register	FSR F2i, F2j, C	0110 0111 F2i F2j
	Floating Multiply Register	FMR F2i, F2j, C	0110 1000 F2i F2j
	Floating Divide Register	FDR F2i, F2i, C	0110 1001 F2i F2j
	Floating Move Register	FMV F2i, F2j, C	0110 1010 F2i F2j
	Floating Lood Negative	FLN F2i, F2j, C	0110 1110 F2i F2j
	Floating Compare	FCMP F2i, F2j, C	0110 1111 F2i F2j

命令型式	内容
	$\widetilde{X}$→APC [R2i]　の時$\widetilde{X}$→APC [R2i]＝0の時$\widetilde{X}$→APC [F2i]＜0の時$\widetilde{X}$→APC [F2i]＝0の時$\widetilde{X}$→APC データ処理部が停止する。 0→{MK(k, l)；k=1～128, l=1～256}
	[R2i]＋[$\widetilde{X}$]→R2i [R2i]－[$\widetilde{X}$]→R2i [R2i]×[$\widetilde{X}$]→R2i [R2i]／[$\widetilde{X}$]→R2i [$\widetilde{X}$]→R2i [R2i]→$\widetilde{X}$ [R2i]＋[R2j]→R2i, Cを満たす時次の命令をスキップ [R2i]－[R2j]→R2i, Cを満たす時次の命令をスキップ [R2i]×[R2j]→R2i, Cを満たす時次の命令をスキップ [R2i]／[R2j]→R2i, Cを満たす時次の命令をスキップ [R2j]→R2i, Cを満たす時次の命令をスキップ －[R2j]→R2i, Cを満たす時次の命令をスキップ [R2i]－[R2j]がCを満たす時次の命令をスキップ [R2i]＋1→R2i, Cを満たす時次の命令をスキップ [F2i]＋[$\widetilde{X}$]→F2i [F2i]－[$\widetilde{X}$]→F2i [F2i]×[$\widetilde{X}$]→F2i [F2i]／[$\widetilde{X}$]→F2i [$\widetilde{X}$]→F2i [F2i]→$\widetilde{X}$ [F2i]＋[F2j]→F2i, Cを満たす時次の命令をスキップ [F2i]－[F2j]→F2i, Cを満たす時次の命令をスキップ [F2i]×[F2j]→F2i, Cを満たす時次の命令をスキップ [F2i]／[F2j]→F2i, Cを満たす時次の命令をスキップ [F2j]→F2i, Cを満たす時次の命令をスキップ －[F2j]→F2i, Cを満たす時次の命令をスキップ [F2i]－[F2j]がCを満たす時次の命令をスキップ

	命令	記号命令記述型式	0 8 16 24
アレイ演算命令	Add Array	AA R3i, T, EC, MO, C, CB, LS, CS, X	0000 0000 R3i T
	Subtruct Array	SA R3i, T, EC, MO, C, CB, LS, CS, X	0000 0001 R3i T
	Multiply Array	MA R3i, T, EC, MO, C, CB, LS, CS, X	0000 0010 R3i T
	Divide Array	DA R3i, T, EC, MO, C, CB, LS, CS, X	0000 0011 R3i T
	Load Array	LA R3i, T, EC, MO, C, CB, LS, CS, X	0000 0100 R3i T
	Transfer Array	TA R3i, T, EC, MO, C, CB, LS, CS, X	0000 0101 R3i T
	Add Register Array	ARA R3i, R3j, EC, MO, C	0000 0110 R3i R3j
	Subtruct Register Array	SRA R3i, R3j, EC, MO, C	0000 0111 R3i R3j
	Multiply Register Array	MRA R3i, R3j, EC, MO, C	0000 1000 R3i R3j
	Divide Register Array	DRA R3i, R3j, EC, MO, C	0000 1001 R3i R3j
	Move Register Array	MVA R3i, R3j, EC, MO, C	0000 1010 R3i R3j
	Load Negative Array	LNA R3i, R3j, EC, MO, C	0000 1110 R3i R3j
	Compare Array	CMP R3i, R3j, EC, MO, C	0000 1111 R3i R3j
	Increuse Array	ICA R3i, EC, MO, C	0001 0000 R3i
	Floating Add Array	FAA F3i, T, EC, MO, C, CB, LS, CS, X	0010 0000 F3i T
	Floating Subtract Array	FSA F3i, T, EC, MO, C, CB, LS, CS, X	0010 0001 F3i T
	Floating Multiply Array	FMA F3i, T, EC, MO, C, CB, LS, CS, X	0010 0010 F3i T
	Floating Divide Array	FDA F3i, T, EC, MO, C, CB, LS, CS, X	0010 0011 F3i T
	Floating Load Array	FLA F3i, T, EC, MO, C, CB, LS, CS, X	0010 0100 F3i T
	Floating Transfer Array	FTA F3i, T, EC, MO, C, CB, LS, CS, X	0010 0101 F3i T
	Floating Add Register Array	FARA F3i, F3j, EC, MO, C	0010 0110 F3i F3j
	Floating Subtract Register Array	FSRA F3i, F3j, EC, MO, C	0010 0111 F3i F3j
	Floating Multiply Register Array	FMRA F3i, F3j, EC, MO, C	0010 1000 F3i F3j
	Floating Divide Register Array	FDRA F3i, F3j, EC, MO, C	0010 1001 F3i F3j
	Floating Move Register Array	FMVA F3i, F3j, EC, MO, C	0010 1010 F3i F3j
	Floating Load Negative Array	FLNA F3i, F3j, EC, MO, C	0010 1110 F3i F3j
	Floating Compare Array	FCMPA F3i, F3j, EC, MO, C	0010 1111 F3i F3j
会話命令	Move Arithmetic Communication Register	MAC	1000 0000
	Move Communication Register	MCR	1000 0001
	Sture to Communication Register	SCR R2i	1000 0010 R2i
	Floating Store to Communication Register	FSCR F2i	1000 0011 F2i
	Read Supervisor Communication Register	RSC	1000 0100
	Load Communication Register	LCR R2i	1000 1000 R2i
	Floating Load Communication Register	FLCR F2i	1000 1001 F2i

注) k′=k+LS, l′=l+CS

命令型式							内容
32		40		48		56 63	
EC	MO	C	CB	LS	CS	X	[R3i(k, l)]+[$\widetilde{X}$(k', l')]$\overset{注)}{\rightarrow}$R3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[R3i(k, l)]−[$\widetilde{X}$(k', l')]→R3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[R3i(k, l)]×[$\widetilde{X}$(k', l')]→R3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[R3i(k, l)]/[$\widetilde{X}$(k', l')]→R3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[$\widetilde{X}$(k', l')]→R3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[R3i(k, l)]→$\widetilde{X}$(k', l'), Cを満たす時マスク操作
EC	MO	C					[R3i(k, l)]+[R3j(k, l)]→R3i(k, l), Cを満たす時マスク操作
EC	MO	C					[R3i(k, l)]−[R3j(k, l)]→R3i(k, l), Cを満たす時マスク操作
EC	MO	C					[R3i(k, l)]×[R3j(k, l)]→R3i(k, l), Cを満たす時マスク操作
EC	MO	C					[R3i(k, l)]/[R3j(k, l)]→R3i(k, l), Cを満たす時マスク操作
EC	MO	C					[R3j(k, l)]→R3i(k, l), Cを満たす時マスク操作
EC	MO	C					−[R3j(k, l)]→R3i(k, l), Cを満たす時マスク操作
EC	MO	C					[R3i(k, l)]−[R3j(k, l)]がCを満たす時マスク操作
EC	MO	C					[R3i(k, l)]+1→R3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[F3i(k, l)]+[$\widetilde{X}$(k', l')]→F3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[F3i(k, l)]−[$\widetilde{X}$(k', l')]→F3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[F3i(k, l)]×[$\widetilde{X}$(k', l')]→F3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[F3i(k, l)]/[$\widetilde{X}$(k', l')]→F3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[$\widetilde{X}$(k', l')]→F3i(k, l), Cを満たす時マスク操作
EC	MO	C	CB	LS	CS	X	[F3i(k, l)]→$\widetilde{X}$(k', l'), Cを満たす時マスク操作
EC	MO	C					[F3i(k, l)]+[F3j(k, l)]→F3i(k, l), Cを満たす時マスク操作
EC	MO	C					[F3i(k, l)]−[F3j(k, l)]→F3i(k, l), Cを満たす時マスク操作
EC	MO	C					[F3i(k, l)]×[F3j(k, l)]→F3i(k, l), Cを満たす時マスク操作
EC	MO	C					[F3i(k, l)]/[F3j(k, l)]→F3i(k, l), Cを満たす時マスク操作
EC	MO	C					[F3j(k, l)]→F3i(k, l), Cを満たす時マスク操作
EC	MN	C					−[F3j(k, l)]→F3i(k, l), Cを満たす時マスク操作
							[F3i(k, l)]−[F3j(k, l)]がCを満たす時マスク操作

32 40 48 56 63	
	[CMNCR 2/5]→CMNCR 3/6(k, l)(ただし MK(k, l)が'OFF')
	MK(k, l)が'OFF'で(k, l)が最小の[CMNCR 3/6(k, l)]→CMNCR 2/5
	[R2i]→CMNCR 2/5かつ[R3i(k, l)]→CMNCR 3/6(k, l)
	[F2i]→CMNCR 2/5かつ[F2i(k, l)]→CMNCR 3/6(k, l)
	[CMNCR 1/4]→CMNCR 2/5
	[CMNCR 2/5]→R2iかつ[CMNCR 3/6(k, l)]→R3i(k, l)
	[CMNCR 2/5]→F2iかつ[CMNCR 3/6(k, l)]→F3i(k, l)

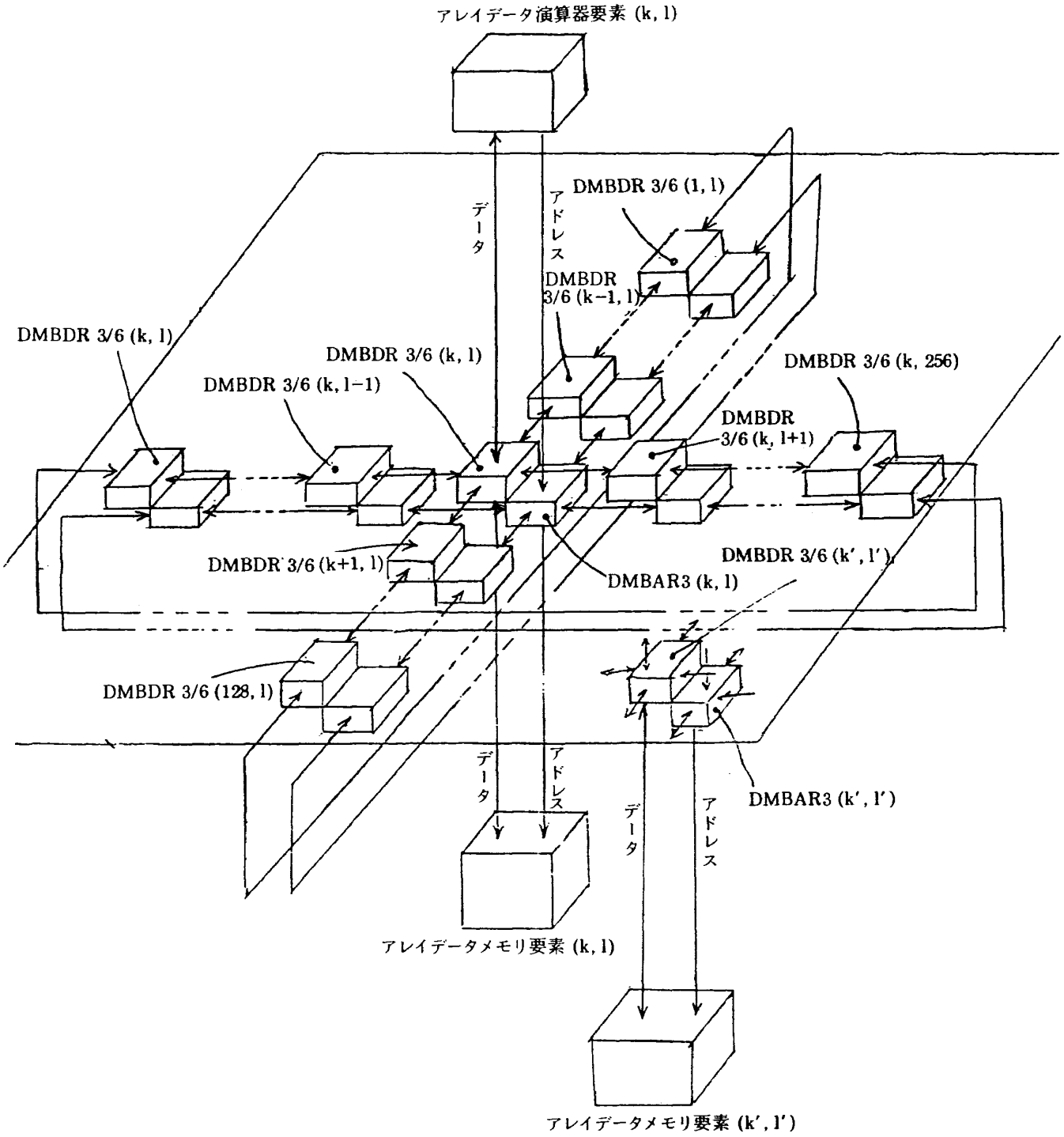


図3　アレイデータメモリ要素と結合網

（0；隣接結合のみ）

LS　；データの移動行数

CS　；データの移動列数

X　；アドレス

[ ]　；レジスタおよびメモリの内容

$\widetilde{X}$　；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

$$\begin{cases} T=0 & ; \widetilde{X}=X \\ T=1\sim7 & ; \widetilde{X}=X+[R1_T] \text{または} \\ & \quad X+[R2_T] \end{cases}$$

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

$$\begin{cases} T=0 & ; \widetilde{X}(k, l)=X \\ T=1\sim7 & ; \widetilde{X}(k, l) \\ & \quad =X+[R3_T(k, l)] \end{cases}$$

SPC　；制御処理部のプログラムカウンタ

APC　；データ処理部のプログラムカウンタ
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, l)；アレイデータ演算器要素のコミュニケーションレジスタ
MK(k, l) ；アレイデータ演算器要素のマスクレジスタ

3. 並列計算機のアセンブラ

3.1 アセンブラの発展

初期の計算機は記憶容量が小さく，かつ入出力機器も貧弱だったことから，そのアセンブラは紙テープ等から原始プログラムを入力し，変換して紙テープ等に出力する形態が広く行われた。しかしアセンブラは原始プログラムを読み込みながら記号（ラベル）の値を確定して行くものであり，記号の値が後で確定するようなプログラムの場合にはその記号を使用した文の値をただちに確定することができない。

図4では，記号'DEF'，'ABC'をラベル，'DC'をオペランドに記された式の値を設定する命令とすると，アセンブラが①の段階まで原始プログラムを読み込んだ時には'DEF'は確定しているのに対し'ABC'は未確定であり，それを使用した式の値

ABC+1 および ABC−10

を確定することができない。

このため原始プログラムを一度入力して全記号の値を確定した後，再度原始プログラムを入力し，機械語等に変換する2パス方式が取られたが，紙テープ等による原始プログラムの2度入力は煩わしいため，記号テーブルの操作・管理に様々な検討・改良を重ね原始プログラムの入力を一度で済ます1パス方式アセンブラが開発された。

図4の原始プログラムを①まで読み込んだ時の

図4　記号 'ABC' が後で確定する例

```
ラベル　命令　オペランド
DEF　　DC　　ABC+ 1
　　　 ⋮
　　　 DC　　ABC−10
　　　 ⋮ ····· ←　①
ABC　　DC　　100 ····· ←　②
```

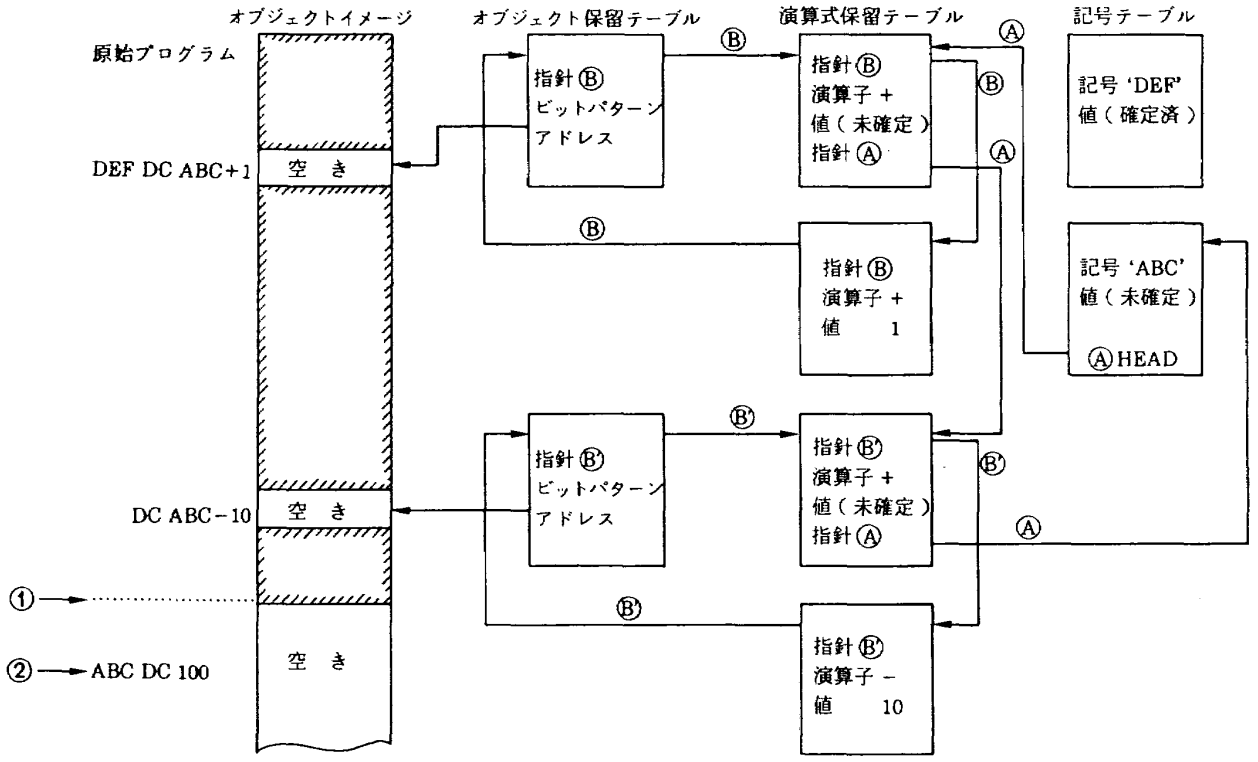


図5　1パスアセンブラの記号テーブル管理例

1パス方式アセンブラの記号テーブル管理例を図5に示す。この時記号'DEF'は確定済みであるのに対し記号'ABC'は未確定であり，②の段階まで解読した時に確定する。この時，保留されていた処理をさかのぼって行うためには，その保留の種類および数等を記録しているオブジェクト保留テーブル，演算式保留テーブルが必要で，これらは未確定記号'ABC'から始まる論理関係を示す指針によって環状に連なるように記録されており，その指針をたどって行くと全ての関連事項を参照し元にもどることができる。

記号'ABC'が確定すると，これらのテーブルを巡って対応する処理（オブジェクトイメージの空きを埋める等）を行い，不要なテーブルを消去する。

しかし，その操作が複雑であり，かつディスク等の大容量化が進むにしたがい原始プログラムをディスク上から何度も入力できるようになったため，再び2パス方式アセンブラが開発されるようになった。その後，複数の命令をまとめて簡単な表記で登録し，命令列中に埋め込み，展開するマクロ機能が付加され現在に至っている。

本稿で述べる並列計算機のアセンブラは2パス方式であり，かつ並列計算機の特殊性から，使い易さよりも，アセンブラとしての機能の確保に重点を置いて作成した。

3.2 機　能

並列計算機のプログラムは文から成り，文は

ラベル　命令　オペランド

の型式を有し，ラベルおよび命令は記号で表わされ，オペランドは命令によりレジスタを示すレジスタオペランド，アドレスを示すアドレスオペランド等に分かれるが，共に記号および数値による式で表わされる。

以下にアセンブラの機能を示す。

(1) プログラム構造の定義

並列計算機のプログラムは図6に示す構造であることを前提とし，表3に示す命令で構造を指定する。

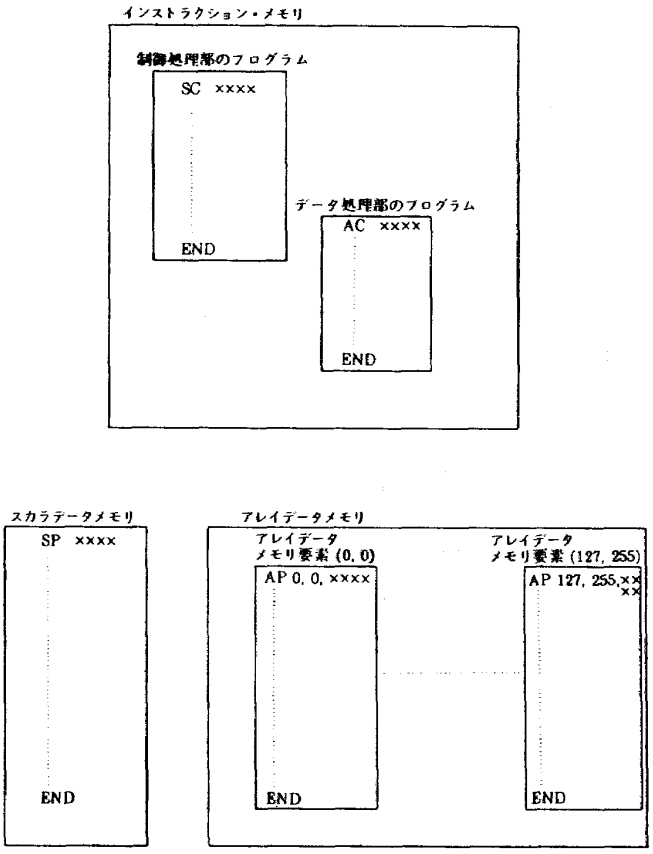


図6　並列計算機のプログラム構造

(2) 記号の直接定義

記号に数値を直接対応づける。数値としては整数および実数であり，その表記法は定義済み記号を用いた式である（表4）。

(3) アドレスの自動割当ておよび機械命令語の生成

表1，表2に示す命令に記号を対応させ1命令ごとに機械命令に変換し，この命令と同じ文に付けられた記号（ラベル）にアドレスの値を割り当てる。アセンブラの中核を成す機能である。

(4) 作業域の確保と定数の生成

スカラデータメモリおよびアレイデータメモリ要素内に作業域および定数を生成する（表5）。

アセンブラの定数は全て64ビットで構成されていて，整数の時には下32ビットに，実数の時には上32ビットに定数を生成する。

3.3 特　徴

このアセンブラは並列計算機の構成に適合するため以下の特徴を有する。

表3 プログラム構造を定義する命令

表記	意味
SC アドレス	制御処理部の命令列を定義する。アドレスはインストラクションメモリ内における命令列の先頭アドレスを示す。
AC アドレス	データ処理部の命令列を定義する。アドレスはインストラクションメモリ内における命令列の先頭アドレスを示す。
SP アドレス	スカラデータメモリの作業域および定数を定義する。アドレスはスカラデータメモリ内における作業域および定数の先頭アドレスを示す。
AP 行，列，アドレス	アレイデータメモリ要素の作業域および定数を定義する。行および列は128×256個のアレイデータメモリ要素を指定し，アドレスは各要素内における作業域および定数の先頭アドレスを示す。
END	プログラムの終了を示す。

表4 記号を定義する命令

表記	意味
記号 EQ 数値	記号の値を設定する。

表5 作業域を確保し定数を生成する命令

表記	意味
記号 BS 語数	作業域を指定した語数だけ確保する。
記号 DC 数値	定数を生成する。

(1) 制御処理部およびデータ処理部の命令列を一時にアセンブルする。

通常の計算機は命令解読部が1ケであるため複数の命令列を扱うことは無い。これに対して制御処理部とデータ処理部は独立した命令解読部を各々有し，単独で処理を行い得るため，両処理部の命令列を生成する必要がある。ただし，制御処理部がデータ処理部を起動する際に制御処理からデータ処理部に対して先頭命令アドレスが渡される。このため両者の命令列を一時にアセンブルする必要がある。

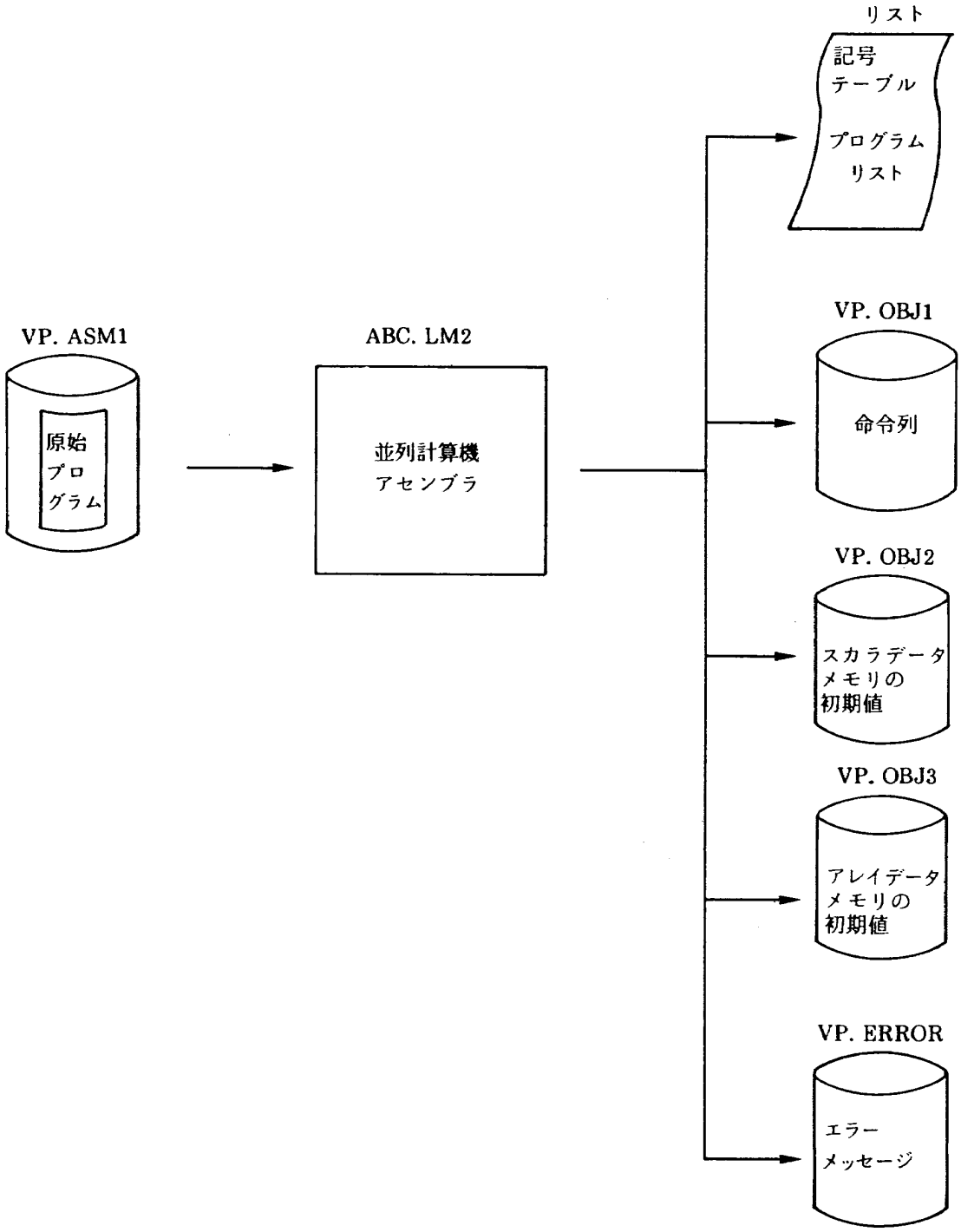
(2) 並列計算機はインストラクション・メモリ，スカラデータ・メモリ，アレイデータ・メモリを有するため，コントロール命令のアドレス部はインストラクション・メモリのアドレス，スカラ演算命令のアドレス部はスカラデータメモリのアドレス，アレイ演算命令のアドレス部はアレイデータメモリのアドレスを指す。

(3) 機械語の命令とデータに変換されたプログラムは命令列，スカラデータの初期値およびアレイデータの初期値として分割されて出力される。特にアレイデータはアレイデータメモリ要素ごとに区分される。

4. アセンブラの実行

アセンブラを起動する時のジョブ制御文と入出力ファイルの関係を図7に示す[2]。

並列計算機の記号命令で書かれた原始プログラ



アセンブラのジョブ制御文例

```
//VPASM  EXEC PGM=ASO,PARM='###.VP.OBJ3',REGION=1024K
//STEPLIB DD DSN=###.ABC.LM2,DISP=SHR
//FT01F001 DD DSN=###.VP.ASM1,DISP=SHR
//FT07F001 DD SYSOUT=*,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=141)
//FT08F001 DD SYSOUT=*,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=141)
//FT09F001 DD DSN=###,VP.OBJ1,DISP=SHR
//FT10F001 DD DSN=###.VP.OBJ2,DISP=SHR
//FT11F001 DD DYNAM
//FT12F001 DD DSN=###.VP.ERROR,DISP=SHR
//
```

図7　並列計算機のアセンブラ入出力ファイル

ムはファイル'VP. ASM1'に格納されている。アセンブラは，これを並列計算機の機械語に変換し，命令列をファイル'VP. OBJ1'へ，スカラデータ初期値をファイル'VP. OBJ2'へ出力し，アレイデータ初期値をアレイデータメモリ要素ごとにファイル VP. OBJ3に区分データセットとして出力する。

これらのファイルは並列計算機の動作を忠実に模擬するアーキテクチャシミュレータ[1]の入力となる。

またリストは次の2種類より成る。

(1) 記号テーブル（図8）

原始プログラムの中で使用した記号の値および値が確定した時の文の位置を示す。例では左側の記号が中央の文番号で確定し(DEFN)，その値(VALUE)を16進数で右側に示す。

(2) プログラム・リスト（図9）

これは原始プログラムとそれに対応する機械語を示すリストで，左よりアドレス欄(LOC)，機械語欄(OBJECT CODE)，文番号欄(STMT)，原始プログラム欄(SOURCE STATEMENT)で構成されている。機械語は全て64ビットであり，アドレス欄はバイトアドレスで，機械語欄と共に16進数で示される。

さらに，この図のプログラムは

(1) EQ命令による記号定義

(2) 制御処理部のプログラム
（SC命令～ END命令まで）

(3) データ処理部のプログラム
（AC命令～ END命令まで）

(4) スカラデータメモリの初期値
（SP命令～ END命令まで）

(5) アレイデータメモリ要素の初期値
（AP命令～ END命令まで）

のサブプログラムに分かれており，アレイデータメモリ要素についても要素(0, 0)と要素(127, 255)についてだけ示してある。また並列計算機には3種類のメモリと3種の演算器があるため，各サブプログラムごとに上記の欄の意味が異なり，その詳細を表6に示す。

5. おわりに

並列計算機の性能評価は，応用プログラムを並列計算機で処理して始めて可能となる。このため応用プログラムを並列計算機に移植するためにこのアセンブラを開発した。しかし，並列計算機の構成および命令系を改良する事もおこり得るため，本アセンブラは命令の追加・変更が可能なように命令名，機械語命令コード，オペランド型式等から成るテーブルを有している。今後は応用プログラムを並列計算機に移植し性能評価を行う予定であり，その手順を蓄積して効率の良い中間言語と

SYMBOL	DEFN	VALUE
CONST1	158	00000000
CONST2	159	00000001
CONST3	160	00000002
C000	3	00000000
C001	4	00000001
C002	5	00000002
C004	6	00000004
C008	7	00000008
C016	8	00000010
C032	9	00000020
C064	10	00000040
C128	11	00000080
MINS	12	00000002

図8　記号テーブル

```
* * * * * * * * * * * * *     PROGRAM   LIST     * * * * * * * * * * * * * * * * * *

<-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 ;SUPERVISOR PROCESSOR
00000000                          15           SC        0
                                  16 ;
00000000 C600000A 00000000        17           SAP       0,10
00000008 C7000000 00000000        18           HP
                                  19 ;
                                  20           END
                                  21 ;
                                  22 ;DATA PROCESSOR
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
                                  ...
                                 122           MI
000002F0 C8000000 00000000       123           LA        0,0,0,0,0,0,C000
000002F8 04000000 00000000       124                     1,0,0,0,0,0
00000300 04200000 00200000       125           LA                        ,0
00000308 0A400000 00000000       126           MVA
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
                                 157 ;
00000000                         158           AP        127,255,0
                                 159 ;ARRAY DATA/WORK
00000000 00000000 00008000       160           DC        32768
00000008 00000000 00000080       161           DC        128
00000010 00000000 00000100       162           DC        256
                                 163 ;
                                 164           END
```

記号定義

制御処理部のプログラム

データ処理部のプログラム

スカラデータメモリの初期値

アレイデータメモリ要素の初期値

図9 プログラムリスト

表6　プログラム欄での意味

サブプログラム	アドレス欄	機械語欄	文番号欄	原始プログラム欄
記号定義			原始プログラムの文番号を示す。	ラベルの記号をオペランドの値とする。
制御処理部のプログラム	命令の入るインストラクションメモリのバイトアドレス（16進数）を示す。	1命令を8バイトで構成した機械語を16進で示す。	同　上	○スカラ演算命令のレジスタオペランドは制御データ演算器のレジスタを，アドレスオペランドはスカラデータメモリのアドレスを示す。 ○コントロール命令のレジスタオペランドは制御データ演算器のレジスタを，アドレスオペランドはインストラクションメモリのアドレスを示す。 ○会話命令のレジスタオペランドは制御データ演算器のレジスタを示す。
データ処理部のプログラム	命令の入るインストラクションメモリのバイトアドレス（16進数）を示す。	1命令を8バイトで構成した機械語を16進で示す。	同　上	○スカラ演算命令のレジススオペランドはスカラデータ演算器のレジスタを，アドレスオペランドはスカラデータメモリのアドレスを示す。 ○アレイ演算命令のレジスタオペランドは全アレイデータ演算器要素のレジスタを，アドレスオペランドは全アレイデータメモリ要素のアドレスを示す。 ○コントロール命令のレジスタオペランドはスカラ演算器のレジスタを，アドレスオペランドはインストラクションメモリのアドレスを示す。 ○会話命令のレジスタオペランドはスカラ演算器および全アレイデータ演算器要素のレジスタを示す。
スカラデータメモリの初期値	スカラデータメモリのバイトアドレス（16進数）を示す。	1データを8バイトで構成した機械語を16進数で示す。	同　上	スカラデータメモリの初期値（DC 命令）および作業域（BS 命令）を示す。
アレイデータメモリ要素の初期値	アレイデータメモリ要素のバイトアドレス（16進数）を示す。	同　上	同　上	アレイデータメモリ要素の初期値（DC 命令）および作業域（BS 命令）を示す。 AP 命令ごとに区分データセットとして生成される。

してのマクロ機能について検討することが可能となるものと考えている。

おわりに，本アセンブラの開発にあたり，富士通(株)の協力があったことを記し，感謝の意を表したい。

参 考 文 献

1) 原田；“並列計算機のアーキテクチャシミュレータ”，航空宇宙技術研究所資料，TM-583，1988年3月

2) 富士通；“低レベル命令系処理プログラム使用説明書”，1987年

航空宇宙技術研究所資料586号

昭和63年6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丁目44番地1
電話三鷹(0422)47-5911(大代表)〒182

印刷所　株式会社　東京プレス
東京都板橋区桜川2－27－12

Printed in Japan