

# 資料

## 高速演算モジュール 取扱い説明書

昭和 55 年 3 月

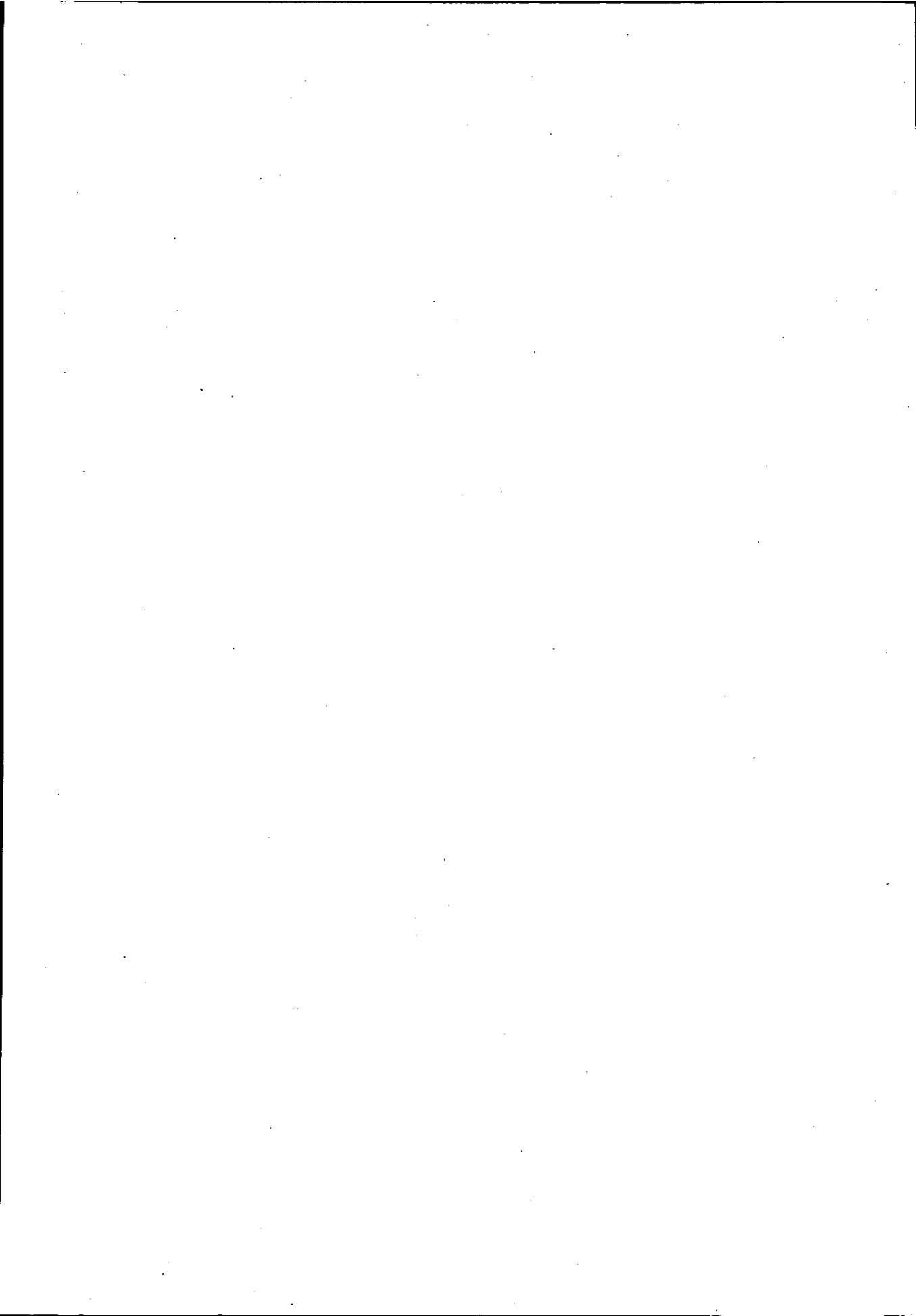
**JIPDEC**

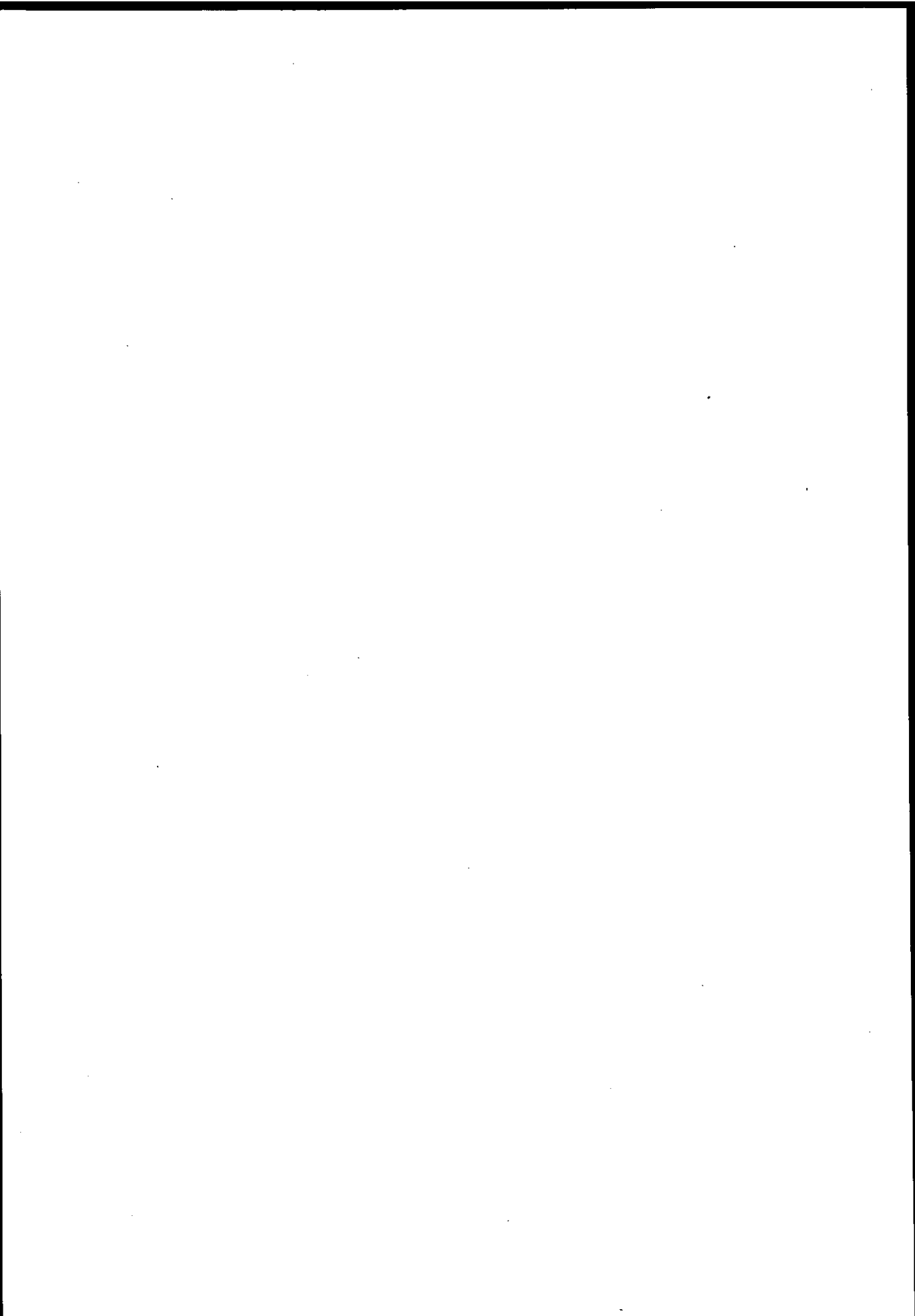
財団法人 日本情報処理開発協会

JIPDEC



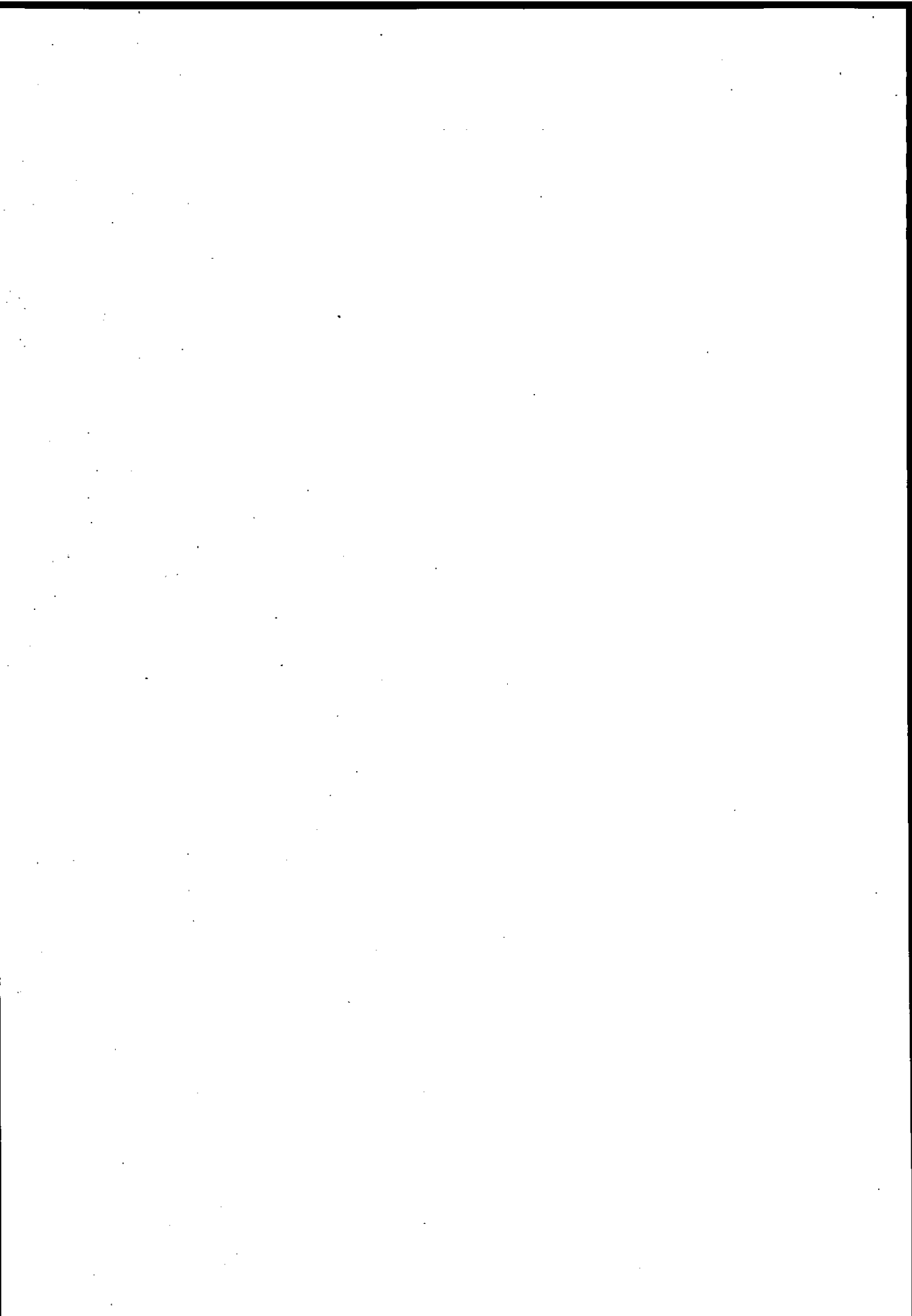
この報告書は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて、昭和54年度に実施した「マイクロコンピュータの応用に関する調査研究」の一環としてとりまとめたものであります。





# 目 次

I	装置製作の目的	1
II	装置の概要	1
III	機能詳細	2
IV	ハードウェア構成	7
4.1	高速演算モジュールの概要	7
4.2	プロセッサ部	8
4.3	命令メモリ部	10
4.4	ホストインタフェース部及びプロセッサインタフェース部	11
V	レジスタ構成	13
5.1	演算用レジスタ	13
5.2	シーケンスカウンタ	13
5.3	ステータスワード	13
VI	データフォーマット	15
6.1	固定小数点データ	15
6.2	浮動小数点データ	16
VII	命令セット	18
7.1	データ転送命令	20
7.2	データ交換命令	28
7.3	固定小数点演算命令	36
7.4	浮動小数点演算命令	47
7.5	浮動小数点関数命令	57
7.6	停止命令	77
VIII	命令一覧	78



## I 装置製作の目的

高速演算処理を必要とする分野にまでマイクロコンピュータ応用を拡大するための部品としての高速演算モジュール（ハードウェア及びこれに必要なソフトウェア）の開発。

## II 装置の概要

本モジュールは、演算部及びホストインタフェース部の2部から構成される。各々は第1図に示すプリント板上に実装される。

第2図にブロック図を示す。

本モジュールは次の機能を有する。

- (1) 16ビット及び32ビット固定小数点演算：加、減、乗、除算
- (2) 32ビット及び64ビット浮動小数点演算：加、減、乗、除算
- (3) 固定小数点  $\longleftrightarrow$  浮動小数点相互の変換
- (4) 初等関数：SQRT, SIN, COS, TAN, ASIN, ACOS, ATAN,  
LOG, LN, EXP
- (5) 単独演算及び複合演算機能
- (6) 対象とするホストプロセッサ：8080AおよびZ80（ホストインタフェース部を交換することにより、他のプロセッサとも接続可能とする。）

### Ⅲ 機能詳細

#### (1) 演算速度

8080Aのソフトウェアのみで計算するよりは、1桁以上高速である。

#### (2) 演算コマンド

ホストプロセッサから、本モジュールに与えるコマンドとして、下記のものを用意する。

i) 前記Ⅱ章の(1), (2), (3)および(4)に記載した各演算に対応した演算指定コマンド

ii) 演算スタートコマンド

iii) コマンド拡張性をもつ

#### (3) ステータス情報

本モジュールの演算終了又は本モジュール内で演算中に下記の状態が発生した時は割込等の手段でホストプロセッサに対し知らせると共に、ステータス情報を送出する。

i) オーバフロー, ii) サイン, iii) ゼロ, iv) キャリー,

v)  $\div$ による除算, vi) 関数アークギュメント・エラー,

vii) オペコードエラー

#### (4) 演算シーケンス

下記の2方式の演算シーケンスを可能とする。

##### i) 単 独 演 算

単独の演算(例えば加算を1回、又はSINを1回等)を行う。ホストプロセッサは、各演算毎に、データ及び演算コマンドを本モジュールに転送し、結果を受け取る。

##### ii) 複 合 演 算

一連の複数次ステップの演算(例えば、一つの多項式の値を求める全ステップの演算)を連続して実行する。

ホストプロセッサは、1回の複合演算に必要なすべてのデータ、及び



演算コマンド列を、演算実行に先行してモジュール内のメモリに転送する。

本モジュールは、演算スタートコマンドにより、ホストプロセッサと無関係に、指定された一連の演算を実行し、演算終了後、演算結果データ（単数又は複数）をホストプロセッサに転送する。

#### (5) システム構成

本モジュールは、演算部及びホストインタフェース部の2部から構成される。

##### 1) 演算部

ブロック図を第2図に示す。演算制御部はビットスライスバイポーラCPUを使用し、コマンド解釈、演算シーケンスの制御を行う。高速マルチプライアはバイポーラLSIを使用する。各演算コマンドに対応した詳細動作シーケンスは、マイクロプログラミングされROMにファームウェア化されている。

本モジュール内には、下記のバッファメモリを有する。

- ① コマンドバッファメモリ
- ② データバッファメモリ

##### ii) ホストインタフェース部

演算部とホストプロセッサ共通バスを接続する。ホストプロセッサは8080A及びZ80とする。他のCPUと本演算部の接続は、ホストインタフェースを別途開発することにより、可能である。

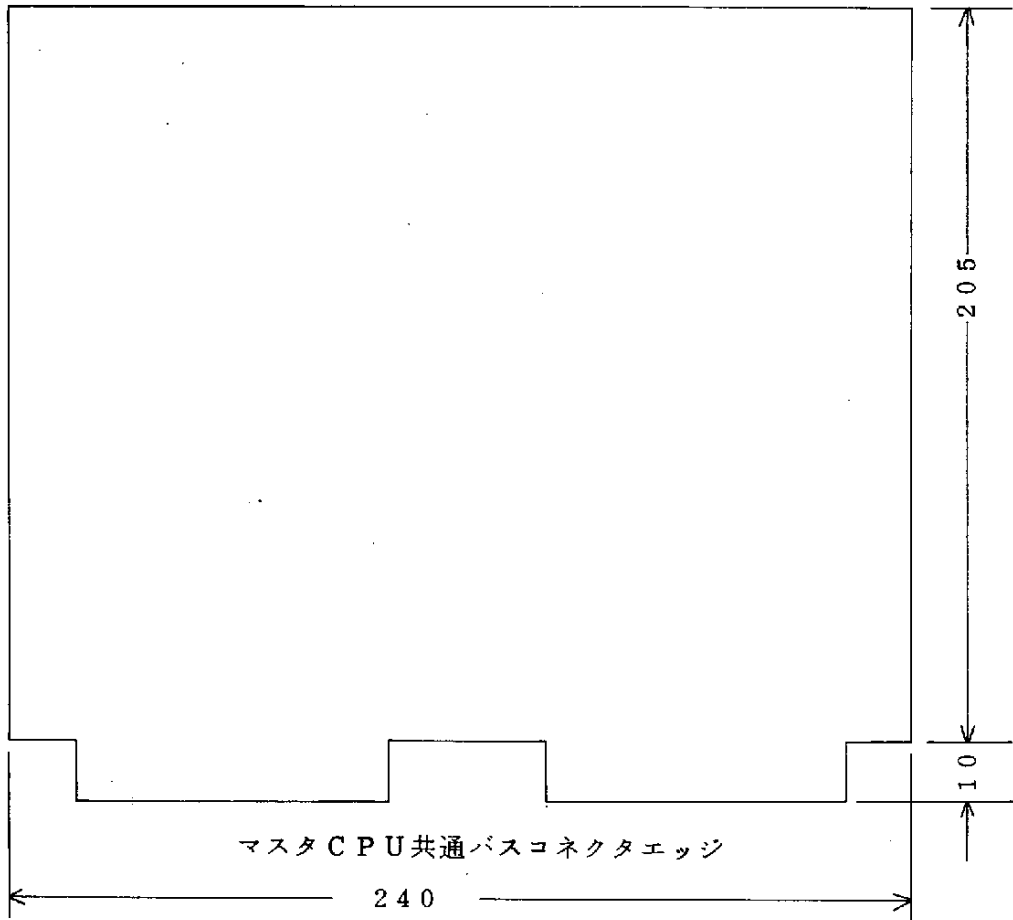
#### (6) 構造仕様

本モジュールは第1-1図、第1-2図に示めすプリント板2枚に実装される。

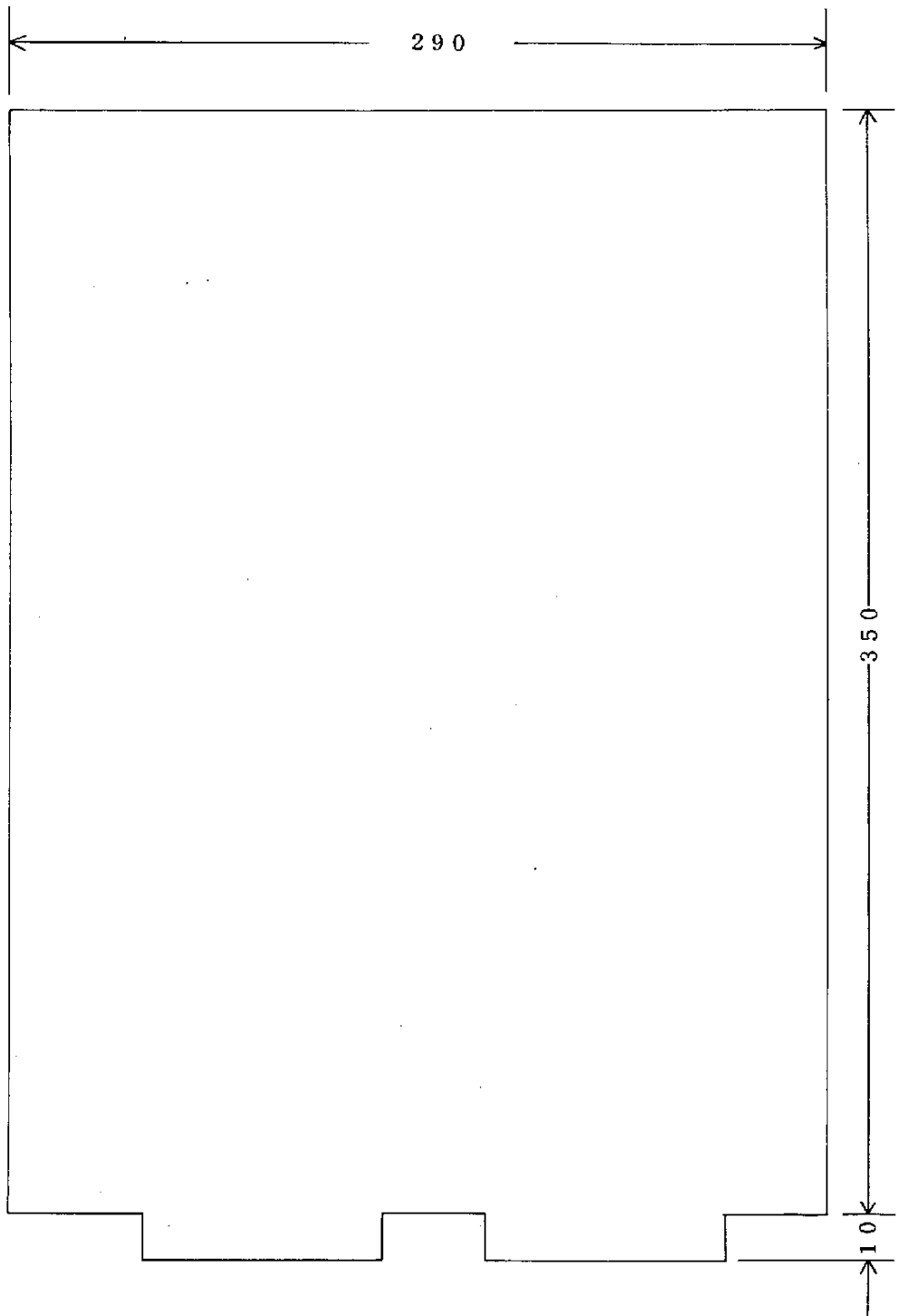
#### (7) 使用条件

温度 0～40℃、湿度 10～90%、結露しないこと。

入力電圧 DC + 5V ± 1%

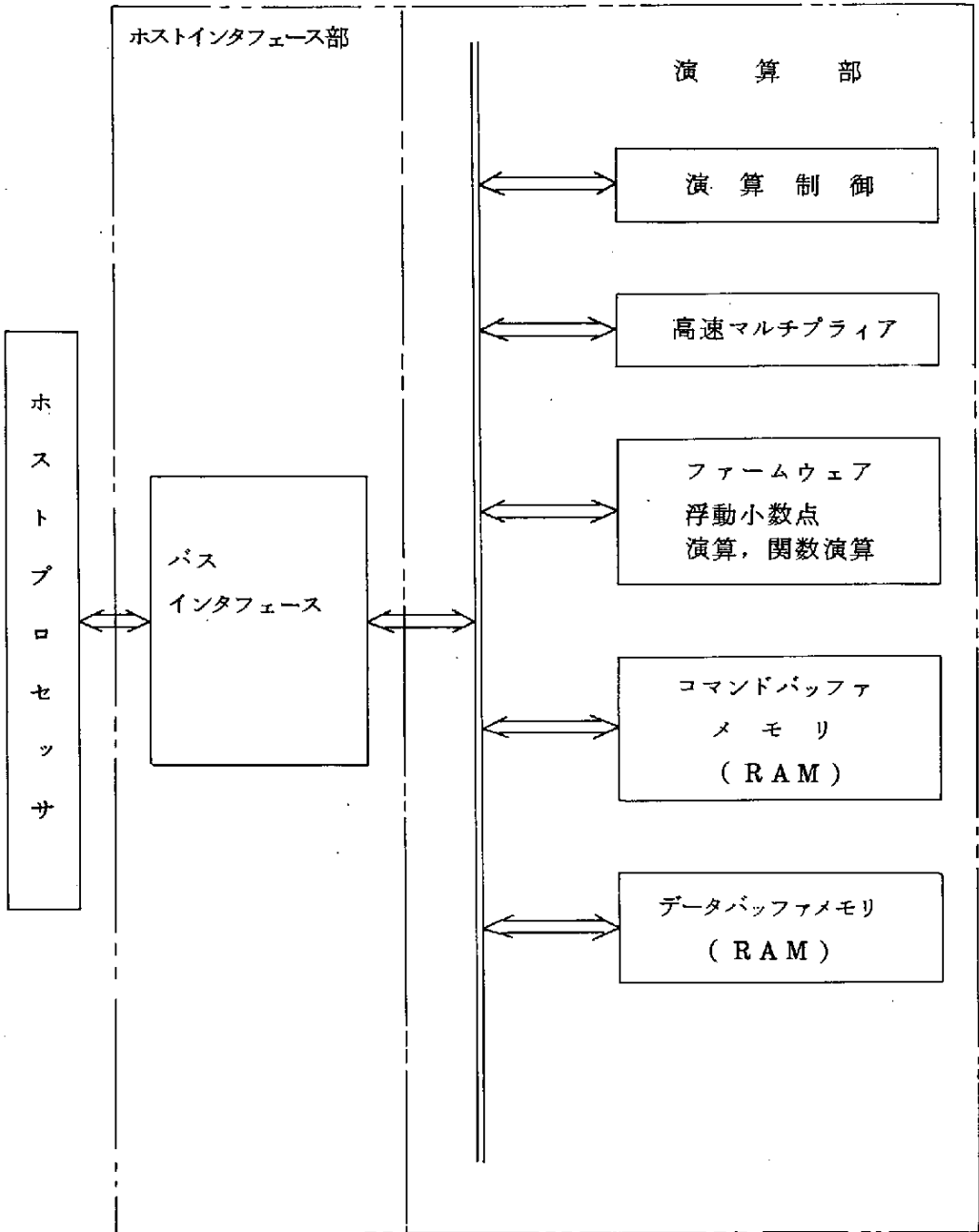


第1-1図 プリント板外形図(ホストインタフェース部)



第1-2図 プリント板外形図(演算部)

高速演算モジュール



第2図 ブロック図

#### IV ハードウェア構成

##### 4.1 高速演算モジュールの概要

本モジュールは、下記の3ブロックにより構成されます。

###### プロセッサ部

演算及びDATAの転送，制御を行います。

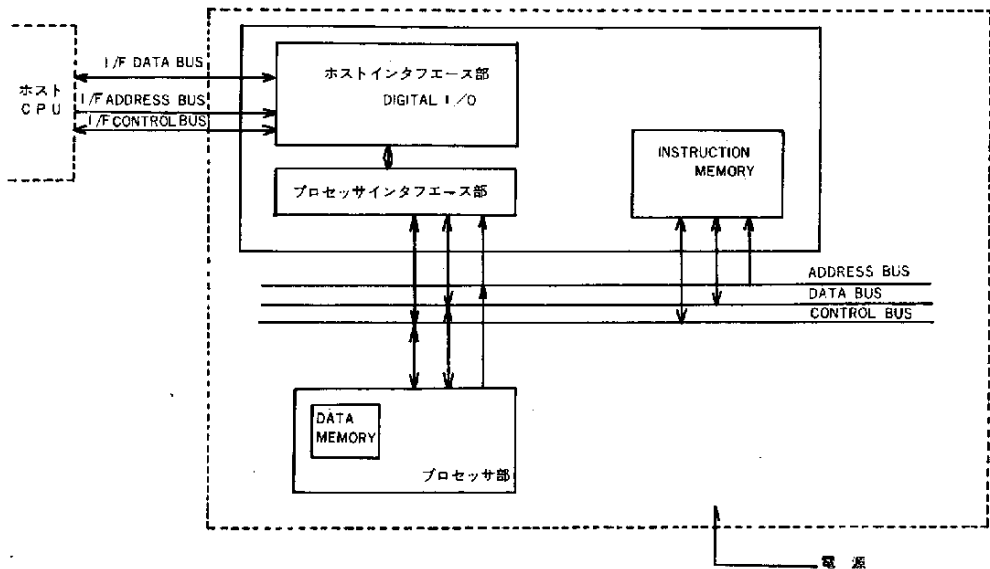
###### 命令メモリ部

複合演算を実行するためのインストラクションコードを記憶する。

###### ホストプロセッサインタフェース部及びプロセッサインタフェース部

ホストプロセッサとの間でDATA，コマンド，インストラクション及びステータスなどのDATA転送を行います。

命令メモリ及びホストプロセッサI/F及びプロセッサI/Fは，プロセッサ部よりの外部BUS信号（ADDRESS BUS，DATA BUS，CONTROL BUS）で制御されます。第3図に本モジュールの構成を示します。



第3図 モジュール構成

## 4.2 プロセッサ部

### 4.2.1 ALU

- 1) AM2903 バイポーラマイクロプロセッサを4個使用  
16ビットCPUを構成
- 2) ALU内部レジスタ 16ビット×16本  
内部レジスタは各種演算のアクキュムレタ, 及びテンポラリ, シーケンスカウンター等に使用
- 3) AM2902 キャリ・ルックアヘッドジェネレイタを使用し, 高速なキャリ伝搬が可能
- 4) 高速なDATAシフト機構
- 5) CLK 4MHz / 1マイクロインストラクション

### 4.2.2 マルチプライアー

- 1) MPY-16HJ (TRW) 高速TTL型16ビット×16ビット  
並列乗算器
- 2) マルチプライ・タイム 140nsec

### 4.2.3 マイクロプログラムシーケンサー

- 1) AM2910 使用
- 2) 制御記憶アドレスは1024ワード
- 3) 条件入力選択回路 16本  
FLG入力, 外部制御入力, 及びシーケンサーが無条件に命令を実行させるための入力等

### 4.2.4 制御記憶

- 1) マイクロ命令のアドレス指定
- 2) AM2910のインストラクション及び条件指定
- 3) コンスタントPROMのアドレス指定
- 4) マルチプライアー, AM2903インストラクション指定
- 5) DATAメモリ アドレス修飾
- 6) 外部バス制御

- 7) 80ビット/ワード, 1Kワードで構成
- 8) その他

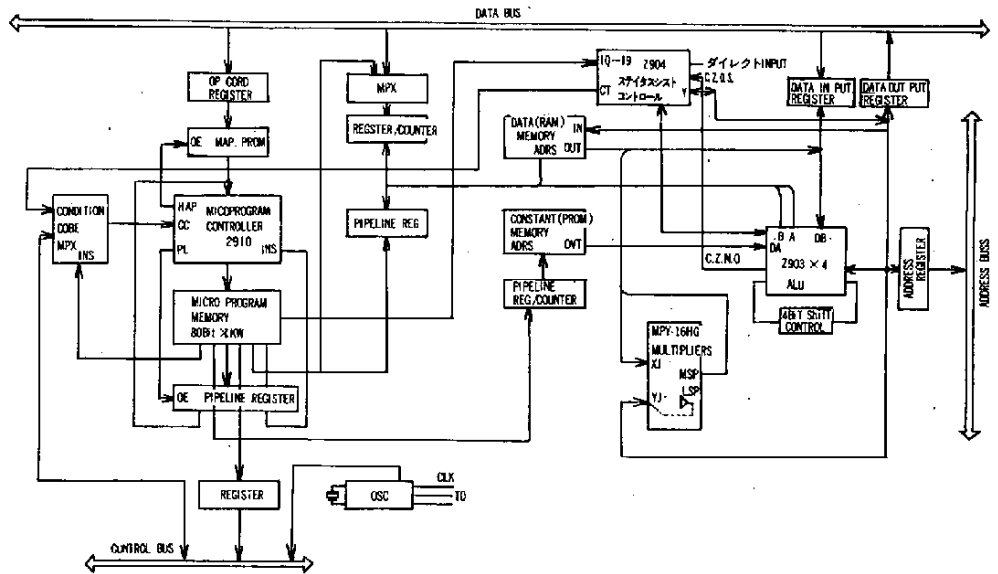
4.2.5 高速DATAメモリー

- 1) 演算用DATAを格納する
- 2) 1ワード16ビット, 256ワードを有する
- 3) ALUのDB-BUS, Y-BUSに接続され高速DATA転送可能

4.2.6 高速コンスタントPROM

- 1) 関数式の係数, 及び定数, 制御用DATAを格納
- 2) ALUのDA-BUSに接続され高速DATA転送可能
- 3) 1ワード16ビット 512ワードを有する

第4図に本モジュールのプロセッサ部のブロック図を示す。

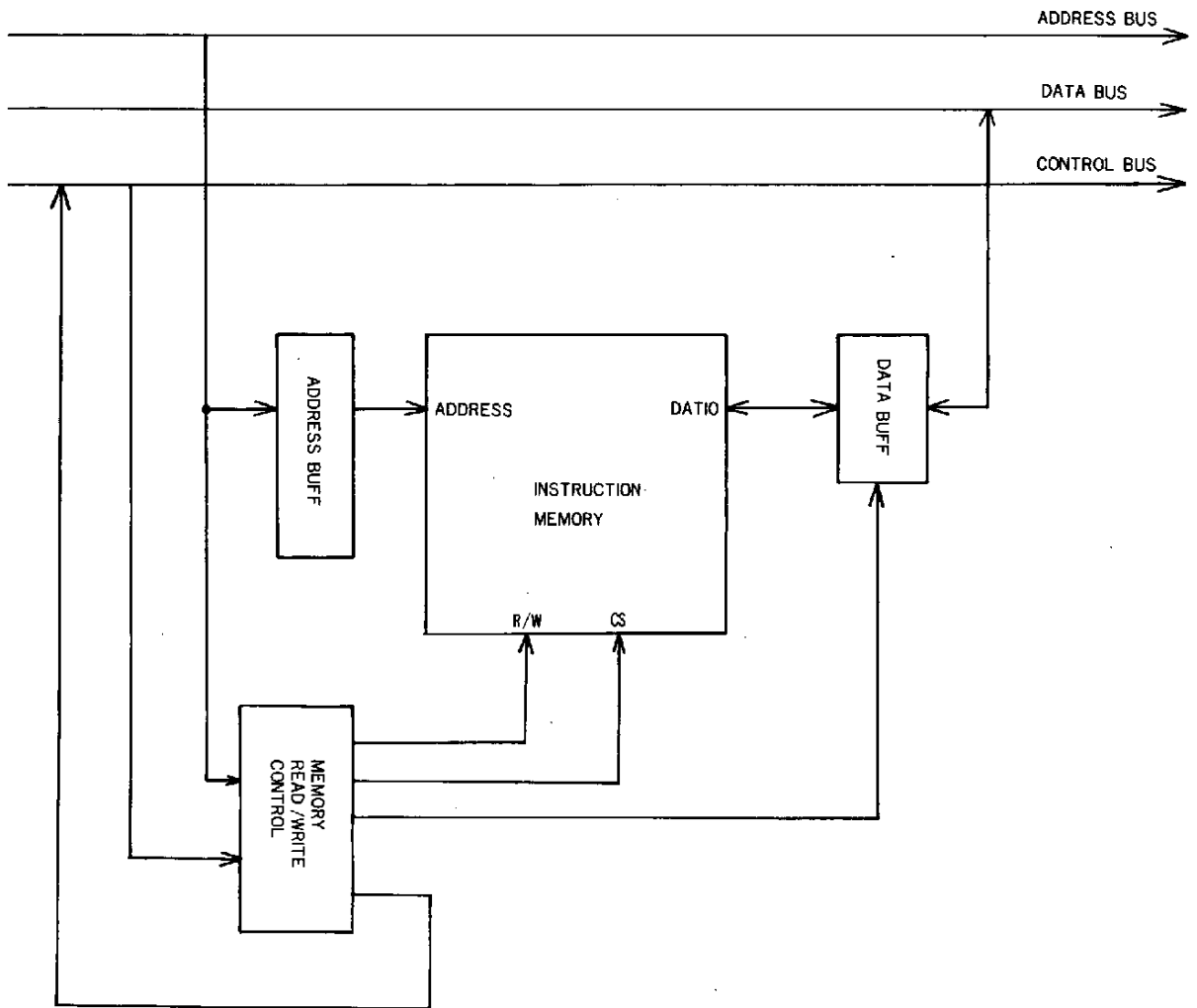


第4図 プロセッサ部ブロック図

### 4.3 命令メモリ部

- 1) 1ワード16ビット1024ワードで構成されます。
- 2) プロセッサ部の外部BUS信号でREAD WRITEされる
- 3) 複合演算を実行するためのインストラクションコードを格納する
- 4) プロセッサ部のシーケンスカウンタによりアクセスされ順次実行
- 5) 外部バス信号を使用する事により拡張が出来る。

第5図に本モジュール命令メモリ部のブロック図を示す。



第5図 インストラクション・メモリブロック図



#### 4.4 ホストインタフェース部及びプロセッサインタフェース部

ホストインタフェースは本モジュールとホストプロセッサとの間でDATA転送を行う機能を持ち、プロセッサ部のデジタルI/Oとして動作を行う。

##### ホストインタフェース部レジスタ

DATA転送はインタフェース内のレジスタを通して実行されます。

##### 1) DATA入力レジスタ

ホストプロセッサから送られる命令、及びDATAを格納するバッファである。

##### 2) DATA出力レジスタ

本モジュールよりの演算結果を格納するバッファである。

##### 3) モードレジスタ

1) DATA入力レジスタにセットされた情報の種別(命令かDATAか)を指定する

2) 複合演算のスタートを指定する。

3) DATA出力レジスタに演算結果の出力要求を指定する。

4) その他

##### 4) ステータスレジスタ

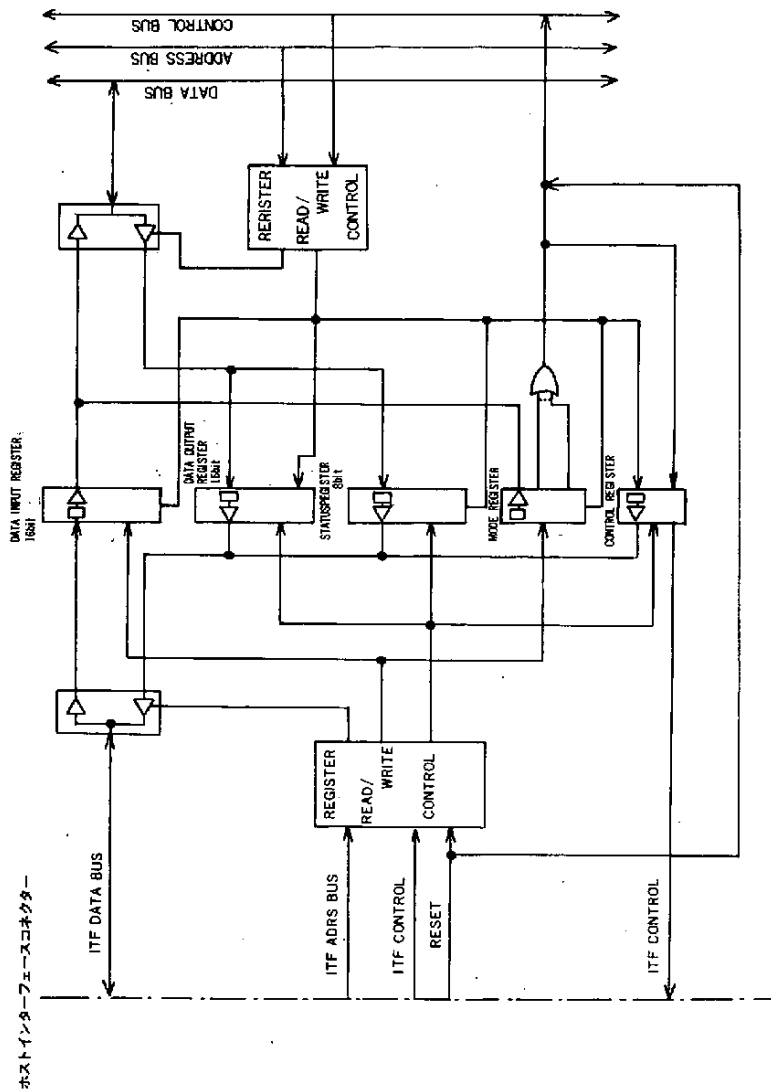
1) 演算の終了時、演算結果の状態(エラー等)を格納するバッファである。

##### ホストプロセッサとの接続

各レジスタはI/F DATA BUS(8ビット双方向)

I/F ADDRESS BUS, I/FコントロールBUSにより制御され、汎用ホストプロセッサに接続が可能です。

第6図に本モジュールのホストインタフェース部のブロック図を示す。



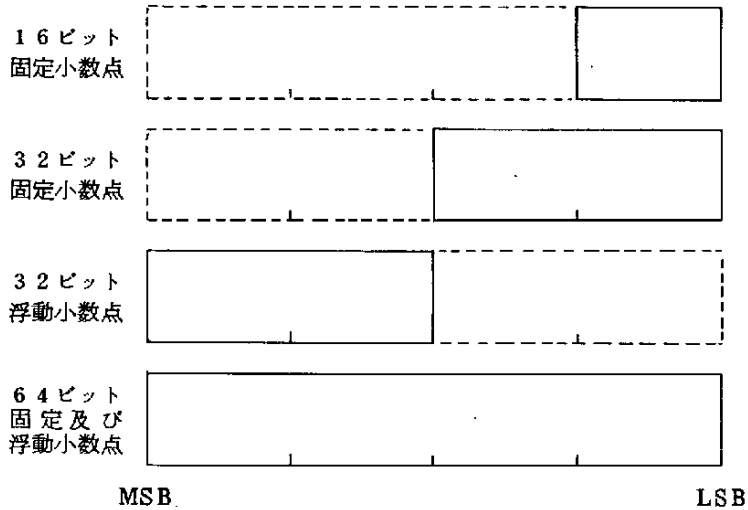
第 6 図 ホストインタフェースブロック図

## V レジスタ構成

### 5.1 演算用レジスタ

本モジュールには 64 ビットのレジスタが 1 つある。全ての演算はこのレジスタとデータメモリ間で行なわれ、結果もこのレジスタに格納される。

このレジスタは命令の種類によって 16 ビット、32 ビット、64 ビットで動作する。



### 5.2 シーケンスカウンタ

本モジュールが複合演算モードで動作する場合、シーケンスカウンタによって、命令メモリより 1 命令ずつ順に取り出し、実行される。

### 5.3 ステータスワード

本モジュールには命令の結果等によってセットされる各種フラグをもったステータス・ワードがある。



#### OV(Over flow flag)

演算等によってオーバーフローが発生した場合，“1”にセットされる。

#### ZD(Zero divide flag)

0による除算の実行によって“1”にセットされる。

#### FE(Function argument Error flag)

関数の引き数として与えられない数値をデータとして関数演算命令を実行した場合，“1”にセットされる。

#### II(Illegal Instruction error flag)

オペコードとして定義されていないコードを実行した場合，“1”にセットされる。

以上の4つのフラグの内1つが“1”にセットされると、本モジュールは直ちに実行を中止し、その旨ホストプロセッサに知らせる。

#### S (Sign flag)

演算結果の符号によってセットされる。

0……正，1……負

#### Z (Zero flag)

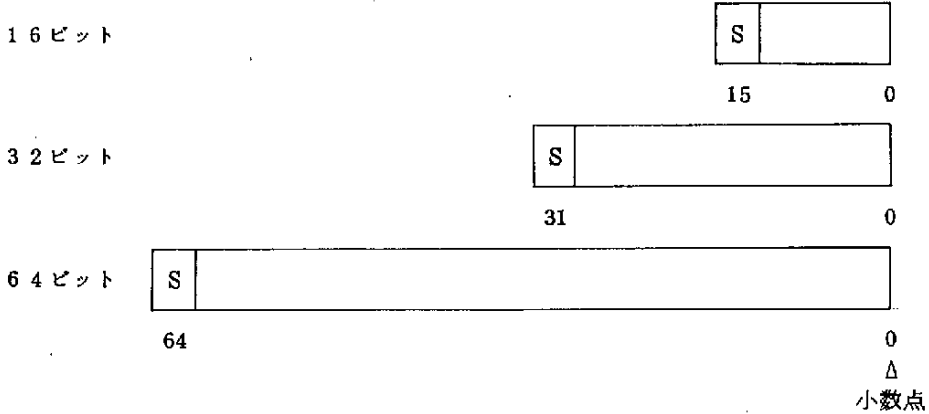
演算結果がゼロの場合，“1”にセットされる。

#### C (Carry flag)

固定小数点演算でキャリーが発生した場合，“1”にセットされる。

## VI データフォーマット

### 6.1 固定小数点データ



固定小数点データは通常の2の補数形式であり、小数点はLSB(ビット0)の右側にあるものとする。したがってすべての固定小数点データは整数である。

16ビット、32ビット及び64ビットのデータがあり、それぞれ1ワード、2ワード、4ワードと呼ぶ。

表わせる数値の範囲は次の通りである。

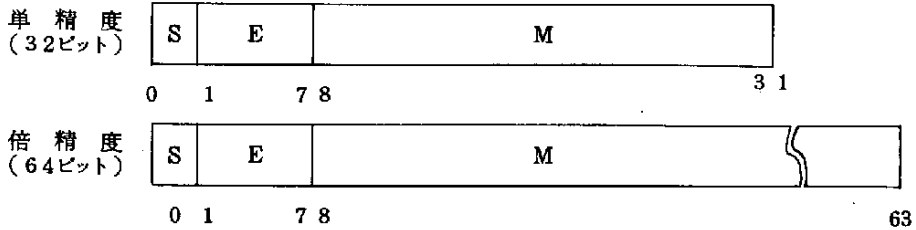
16ビット -32768 ~ 32767

32ビット -2147483648 ~ 2147483647

64ビット -9223372036854775808

~ 9223372036854775807

## 6.2 浮動小数点データ



### S……符号部

符号部1ビットによって仮数部の符号を表わす。S=1のとき負符号を表わし、S=0のとき正符号を表わす。

### E……指数部

指数部は16のべき乗となる、べき数を表わす。指数のべきは1~127に対応して-63~0~63である。これは指数部の示す値から64を引いたものに等しい。

又、指数部の示す値が0の場合、その浮動小数点データの値は0であるものとする。

指数部	値	べき数
7F <sub>16</sub>	127	63
47 <sub>16</sub>	71	7
01 <sub>16</sub>	1	-63

### M……仮数部

仮数部は最上位桁の左側(ビット位置7と8の間)に小数点を考えた16進数である。単精度(32ビット)のとき6桁を有し、倍精度(64ビット)のとき14桁を有する。したがって10進数において、それ

それ7桁及び16桁の精度を有する。

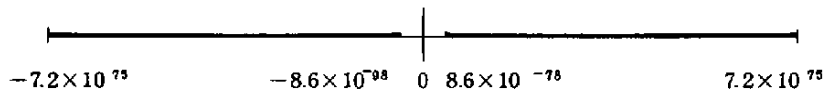
したがって浮動小数点データNの値は次のような式で表わすことができる。

$$N = (-1)^S \times M \times 16^{(E-64)}$$

・ 正 規 化

上位桁に16進数の0を含まない仮数をもつ浮動小数点データは「正規化された数」と言う。

全ての浮動小数点データは正規化されていなければならない。又、演算結果は全て正規化される。



浮動小数点データで表わせる数値の範囲

## Ⅶ 命令セット

本モジュールは58種の命令を有し、大別すると次のようになる。

- データ転送命令
- データ変換命令
- 固定小数点演算命令
- 浮動小数点演算命令
- 浮動小数点関数命令
- 停止命令

### 命令形式

全ての命令は16ビットで構成される。

次のページからは、個々の命令の詳細な記述を施したものである。次ページはそれぞれの個所の説明の例を示したものであり、一番上の左にニーモニックを示し、以下オペコード、オペレーション、解説、フラグの順に説明している。



ニーモニック

FLTS (FL oat Single word data)

オペレーション

FLTS 00100000 00000000

オペレーション

(レジスタ)<sub>2</sub> ← Float (レジスタ)<sub>1</sub>

オペコード

解説

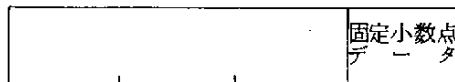
レジスタが動作しているワード数を示す。

レジスタの1ワードの固定小数点データを単精度浮動小数点データ(32ビット)に変換し、レジスタに格納する。

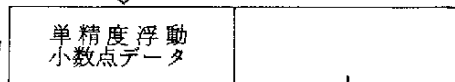
右辺の結果を左辺に格納する。

解説

レジスタ



レジスタ



付図によって付加的な説明を行う。

フラグ

Z, S データの内容によってセットされる。

ステータス・ワードのフラグのセット, リセットについて示す。

## 7.1 データ転送命令

LODS (LOaD Single word data)

LODS n 

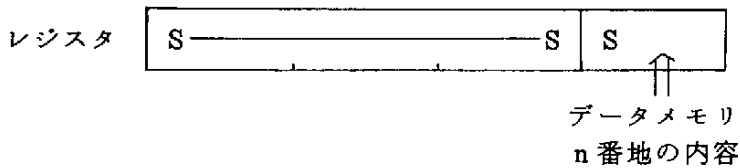
0 0 0 0 0 0 0 0	アドレス n
-----------------	--------

### オペレーション

(レジスタ)<sub>i</sub> ← (n)

### 解 説

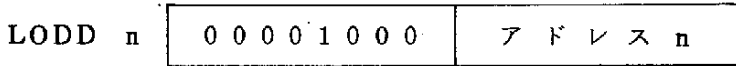
オペランドによって指定されたデータメモリの内容1ワードをレジスタにロードする。又、レジスタの上位のビットはn番地のデータの符号で満たされる。



### フ ラ グ

全てのフラグは影響されない。

LODD (LOaD Double word data)

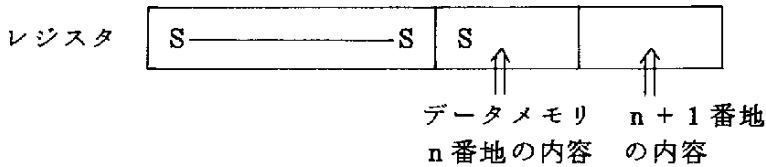


オペレーション

$$(\text{レジスタ})_2 \leftarrow (n, n+1)$$

解 説

オペランドによって指定されたデータメモリの内容及び次の番地の内容2ワードをレジスタにロードする。又、レジスタの上位のビットはn番地のデータの符号で満たされる。



フ ラ グ

全てのフラグは影響されない。

LODF (LOAD single precision Floating point data)

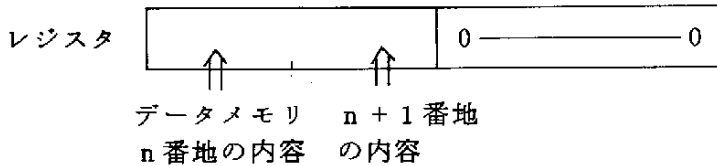
LODF n	0 0 0 1 0 0 0 0	アドレス n
--------	-----------------	--------

オペレーション

(レジスタ)<sub>2</sub> ← (n, n+1)

解 説

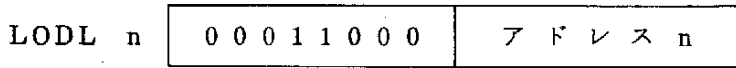
オペランドによって指定されたデータメモリ及び次の番地に格納されている単精度浮動小数点データをレジスタの上位にロードする。又、レジスタの下位のビットは全て“0”で満たされる。



フ ラ グ

全てのフラグは影響されない。

LODL (LOaD Long word data)

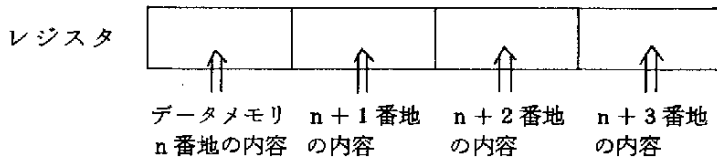


オペレーション

(レジスタ)<sub>r</sub> ← (n, n+1, n+2, n+3)

解 説

データメモリのオペランドで指定された番地から+3番地までの内容  
4ワードをレジスタにロードする。



フ ラ グ

全てのフラグは影響されない。

## STOS (STOre Single word data)

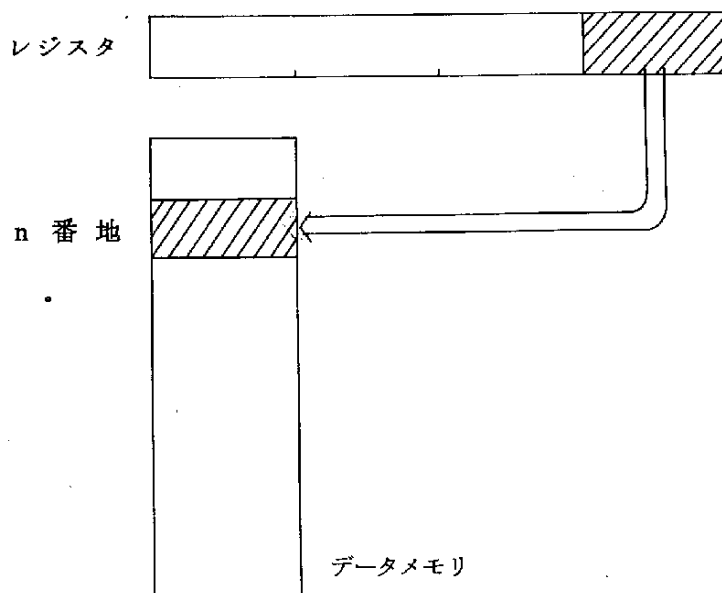
STOS n    0 0 0 0 0 1 0 0    アドレス n

### オペレーション

(n) ← (レジスタ),

### 解 説

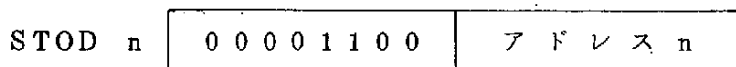
オペランドによって指定されたデータメモリにレジスタの内容1ワードを格納する。



### フ ラ グ

全てのフラグは影響されない。

STOD (STOre Double word data)

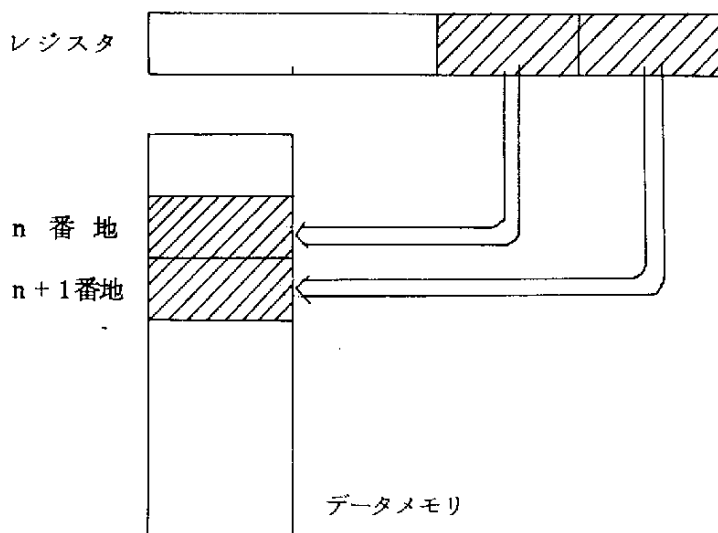


オペレーション

$$(n, n+1) \leftarrow (\text{レジスタ})_2$$

解 説

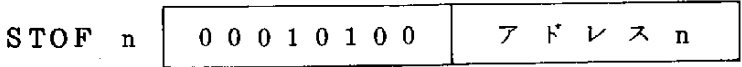
オペランドによって指定されたデータメモリにレジスタの内容2ワードを格納する。



フ ラ グ

全てのフラグは影響されない。

STOF (STOre single precision Floating point data)

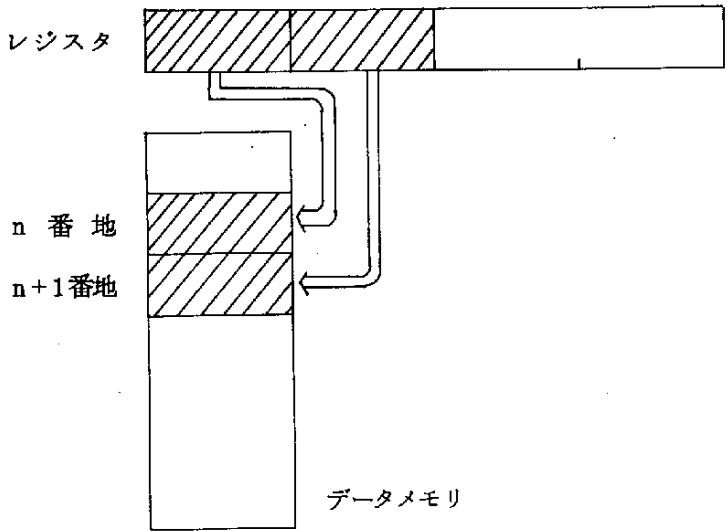


オペレーション

(n, n+1) ← (レジスタ)<sub>2</sub>

解 説

オペランドによって指定されたデータメモリ及び次の番地に、レジスタに格納されている単精度浮動小数点データを格納する。

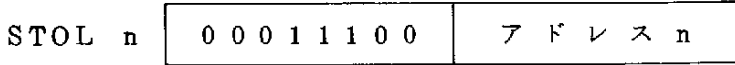


フ ラ グ

全てのフラグは影響されない。



STOL (STOre Long word data)

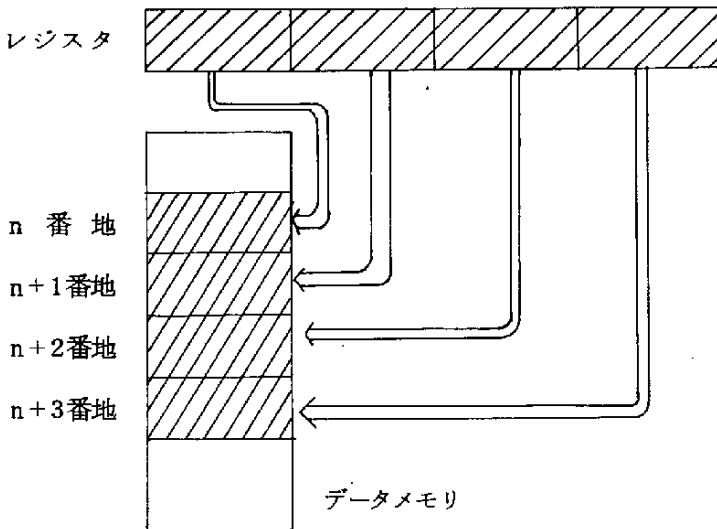


オペレーション

$(n, n+1, n+2, n+3) \leftarrow (\text{レジスタ})_4$

解 説

データメモリのオペランドで指定された番地から+3番地までにレジスタの内容4ワードを格納する。



フ ラ グ

全てのフラグは影響されない。

## 7.2 データ変換命令

FLTS (FLoat Single word data)

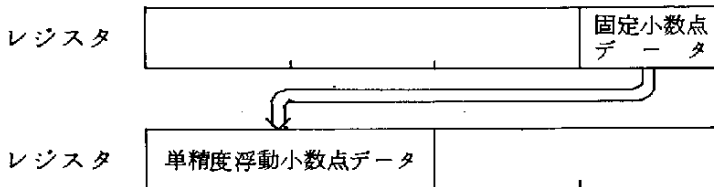
FLTS	0 0 1 0 0 0 0 0	0 0 0 0 0 0 0 0
------	-----------------	-----------------

### オペレーション

(レジスタ)<sub>2</sub> ← Float (レジスタ)<sub>1</sub>

### 解 説

レジスタの1ワードの固定小数点データを単精度浮動小数点データ(32ビット)に変換し、レジスタに格納する。



### フ ラ グ

Z, S データの内容によってセットされる。

FLT (FLoaT Double word data)

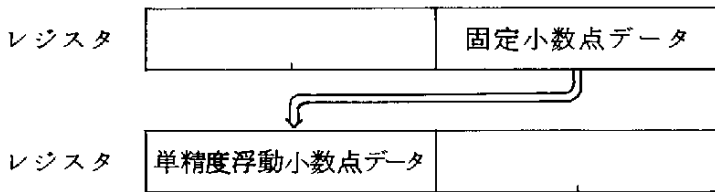
FLT	0 0 1 0 1 0 0 0	0 0 0 0 0 0 0 0
-----	-----------------	-----------------

オペレーション

(レジスタ)<sub>2</sub> ← Float (レジスタ)<sub>2</sub>

解 説

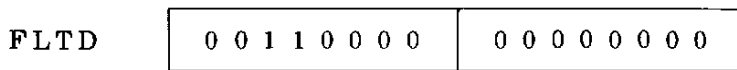
レジスタの2ワードの固定小数点データを単精度浮動小数点データに変換しレジスタに格納する。



フ ラ グ

Z, S データの内容によってセットされる。

FLTD (Float to Double precision)

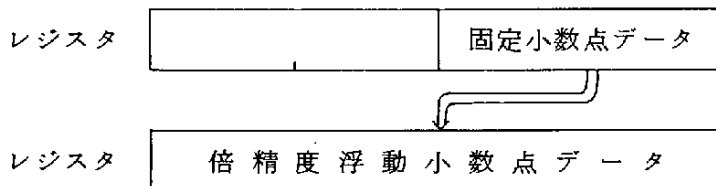


オペレーション

(レジスタ)<sub>4</sub> ← Float (レジスタ)<sub>2</sub>

解 説

レジスタの2ワードの固定小数点データを倍精度浮動小数点データ (64ビット) に変換し、レジスタに格納する。



フ ラ グ

Z, S データの内容によってセットされる。

FLTL (Float Long word data)

FLTL      

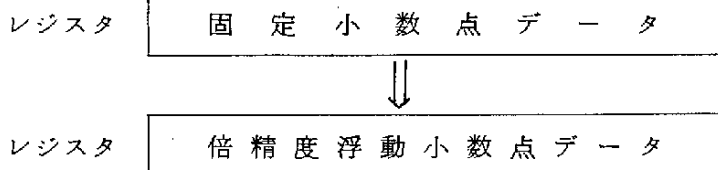
0 0 1 1 1 0 0 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>4</sub> ← Float (レジスタ)<sub>4</sub>

解 説

レジスタの4ワードの固定小数点データを倍精度浮動小数点データに変換し、レジスタに格納する。



フ ラ グ

Z, S データの内容によってセットされる。

## FIXS (FIX to Single word data)

FIXS      

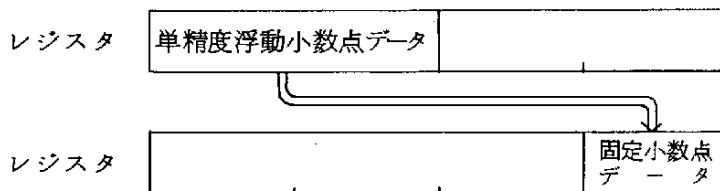
0 0 1 0 0 1 0 0	0 0 0 0 0 0 0 0
-----------------	-----------------

### オペレーション

(レジスタ)<sub>1</sub> ← Fix (レジスタ)<sub>2</sub>

### 解 説

レジスタの単精度浮動小数点データを1ワード固定小数点データに変換しレジスタに格納する。



### フ ラ グ

Z, S データの内容によってセットされる。

OV    1ワードの固定小数点で表わせない場合OV=1にセットされる。

**FIX (FIX to Double word data)**

**FIX**

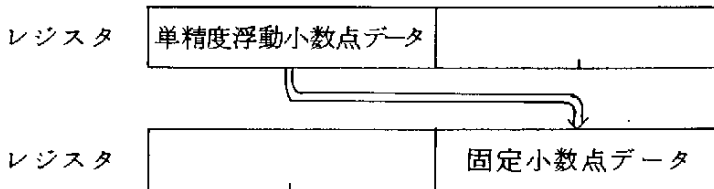
0 0 1 0 1 1 0 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>2</sub> ← Fix (レジスタ)<sub>2</sub>

解 説

レジスタの単精度浮動小数点データを2ワード固定小数点データに変換しレジスタに格納する。



フ ラ グ

**Z, S** データの内容によってセットされる。

**OV** 2ワードの固定小数点で表わせない場合OV=1にセットされる。

FIXD (FIX Double precision data)

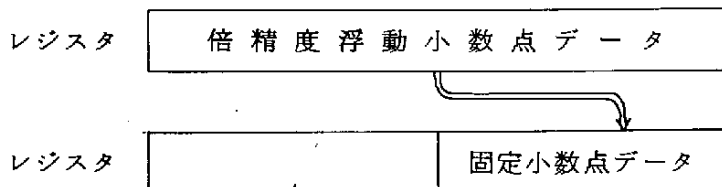
FIXD	0 0 1 1 0 1 0 0	0 0 0 0 0 0 0 0
------	-----------------	-----------------

オペレーション

(レジスタ)<sub>2</sub> ← Fix (レジスタ)<sub>4</sub>

解 説

レジスタの倍精度浮動小数点データを2ワード固定小数点データに変換しレジスタに格納する。



フ ラ グ

Z, S データの内容によってセットされる。

OV 2ワードの固定小数点で表わせない場合OV=1にセットされる。



**FIXL (FIX to Long word data)**

**FIXL**

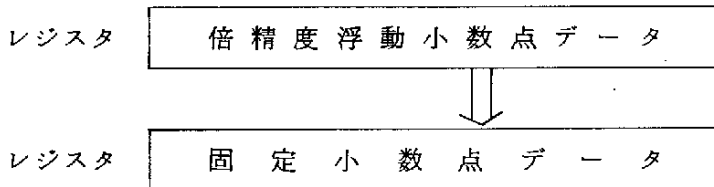
0 0 1 1 1 1 0 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>4</sub> ← Fix (レジスタ)<sub>4</sub>

解 説

レジスタの倍精度浮動小数点データを4ワード固定小数点データに変換し、レジスタに格納する。



フ ラ グ

**Z, S** データの内容によってセットされる。

**OV** 4ワードの固定小数点で表わせない場合OV=1にセットされる。

### 7.3 固定小数点演算命令

ADDS (ADD Single word data)

ADDS n	0 1 1 0 0 0 0 0	アドレス n
--------	-----------------	--------

#### オペレーション

$(\text{レジスタ})_4 \leftarrow (\text{レジスタ})_4 + (n)$

#### 解 説

オペランドによって指定されたデータメモリの1ワード固定小数点データをレジスタの固定小数点データに加算し、レジスタに格納する。

#### フ ラ グ

- Z, S 加算の結果によってセットされる。
- C 加算によってキャリーが生じた場合、C = 1 にセットされる。

ADDD (ADD Double word data)

0 1 1 1 0 0 0 0	ア ド レ ス n
-----------------	-----------

オペレーション

$$(\text{レジスタ})_4 \leftarrow (\text{レジスタ})_4 + (n, n+1)$$

解 説

オペランドによって指定されたデータメモリの内容及び次の番地の内容，2ワード固定小数点データをレジスタの固定小数点データに加算し，レジスタに格納する。

フ ラ グ

Z, S 加算の結果によってセットされる。

C 加算によってキャリーが生じた場合，C=1にセットされる。

SUBS (SUBtract Single word data)

SUBS n	0 1 1 0 0 1 0 0	アドレス n
--------	-----------------	--------

オペレーション

$(\text{レジスタ})_4 \leftarrow (\text{レジスタ})_4 - (n)$

解 説

オペランドによって指定されたデータメモリの1ワード固定小数点データをレジスタの固定小数点データより減算し、差をレジスタに格納する。

フ ラ グ

Z, S 減算の結果によってセットされる。

C 減算によってボローが生じた場合、C=0にセットされる。

SUBD (SUBtract Double word data)

0 1 1 1 0 1 0 0	アドレス n
-----------------	--------

オペレーション

$$(\text{レジスタ})_4 \leftarrow (\text{レジスタ})_4 - (n, n+1)$$

解 説

オペランドによって指定されたデータメモリの内容及び次の番地の内容2ワード固定小数点データをレジスタの固定小数点データより減算し、差をレジスタに格納する。

フ ラ グ

- Z, S 減算の結果によってセットされる。
- C 減算によってボローが生じた場合, C = 0 にセットされる。

MULS (MULTiply Single word data)

MULS n	0 1 1 0 1 0 0 0	アドレス n
--------	-----------------	--------

オペレーション

$(\text{レジスタ})_2 \leftarrow (\text{レジスタ})_1 \times (n)$

解 説

オペランドによって指定されたデータメモリの1ワード固定小数点データとレジスタの1ワードデータとを乗算し、その積(2ワード固定小数点データ)をレジスタに格納する。

フ ラ グ

Z, S 乗算の結果によってセットされる。

MULD (MULtiPLY Double word data)

MULD n 

0 1 1 1 1 0 0 0	アドレス n
-----------------	--------

オペレーション

$$(\text{レジスタ})_4 \leftarrow (\text{レジスタ})_2 \times (n, n+1)$$

解 説

オペランドによって指定されたデータメモリの内容及び次の番地の内容 2ワード固定小数点データとレジスタの 2ワードデータとを乗算し、その積(4ワード固定小数点データ)をレジスタに格納する。

フ ラ グ

Z, S 乗算の結果によってセットされる。

DIVS (DIVide by Single word data)

DIVS n 

0 1 1 0 1 1 0 0	ア ド レ ス n
-----------------	-----------

オペレーション

$$(レジスタ)_1 \leftarrow (レジスタ)_2 / (n)$$

解 説

レジスタの2ワードデータをアドレスによって指定されたデータメモリの内容1ワード固定小数点データで除し、商を1ワードデータとしてレジスタに格納する。又、余りも1ワードデータとして商の上位に格納する。

レジスタ 

	余 り	商
--	-----	---

フ ラ グ

- Z, S 商によってセットされる。
- OV 除算によってオーバーフローが生じた場合、OV=1にセットされる。
- ZD 除数がゼロの場合、ZD=1にセットされる。



DIVD (DIVide by Double word data)

DIVD n	0 1 1 1 1 0 0	アドレス n
--------	---------------	--------

オペレーション

$$(\text{レジスタ})_2 \leftarrow (\text{レジスタ})_4 / (n, n+1)$$

解 説

レジスタの4ワードデータをアドレスによって指定されたデータメモリの内容及び次の番地の内容2ワード固定小数点データで除し、商を2ワードデータとしてレジスタに格納する。又、余りも2ワードデータとして商の上位に格納する。

レジスタ	余 り	商
------	-----	---

フ ラ グ

- Z, S 商によってセットされる
- O.V 除算によってオーバーフローが生じた場合、O.V = 1にセットされる。
- Z.D 除数がゼロの場合、Z.D = 1にセットされる。

CHSS (CHange Sign Single word data)

CHSS	0 1 0 0 1 0 0 0	0 0 0 0 0 0 0 0
------	-----------------	-----------------

オペレーション

(レジスタ)<sub>1</sub> ← -(レジスタ)<sub>1</sub>

解 説

レジスタの1ワード固定小数点データの符号を反転し、レジスタに格納する。

フ ラ グ

Z, S 符号反転の結果によってセットされる。

CHSD (CHange Sign Double word data)

CHSD     

0 1 0 1 0 0 0 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>2</sub> ← - (レジスタ)<sub>2</sub>

解 説

レジスタの2ワード固定小数点データの符号を反転し、レジスタに格納する。

フ ラ グ

Z, S 符号反転の結果によってセットされる。

CHSL (CHange Sign Long word data)

CHSL

0 1 0 1 1 0 0 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>4</sub> ← -(レジスタ)<sub>4</sub>

解 説

レジスタの4ワード固定小数点データの符号を反転し、レジスタに格納する。

フ ラ グ

Z, S 結果の内容によってセットされる。

#### 7.4 浮動小数点演算命令

FADS (Floating point ADd Single precision)

FADS n	1 0 0 0 0 0 0 0	アドレス n
--------	-----------------	--------

#### オペレーション

$(レジスタ)_2 \leftarrow (レジスタ)_2 + (n, n+1)$

#### 解 説

オペランドによって指定されたデータメモリ及び次の番地に格納されている単精度浮動小数データをレジスタの単精度浮動小数点データに加算し、和をレジスタに格納する。

#### フ ラ グ

Z, S 加算の結果によってセットする。

OV 加算によってオーバーフローが生じた場合、OV = 1 にセットされる。

FADD (Floating point ADD Double precision)

FADD n	1 0 0 1 0 0 0 0	アドレス n
--------	-----------------	--------

オペレーション

(レジスタ)<sub>4</sub> ← (レジスタ)<sub>4</sub> + (n, n+1, n+2, n+3)

解 説

データメモリのオペランドで指定された番地から+3番地までに格納されている倍精度浮動小数点データをレジスタに格納されている倍精度浮動小数点データに加算し、レジスタに格納する。

フ ラ グ

Z, S 加算の結果によってセットする。

OV 加算によってオーバーフローが生じた場合、OV=1にセットされる。

FSBS (Floating point SuBtract Single precision)

FSBS n	1 0 0 0 0 1 0 0	アドレス n
--------	-----------------	--------

オペレーション

$$(\text{レジスタ})_2 \leftarrow (\text{レジスタ})_2 - (n, n+1)$$

解 説

オペランドによって指定されたデータメモリ及び次の番地に格納されている単精度浮動小数点データを、レジスタに格納されている単精度浮動小数点データより減算し、差をレジスタに格納する。

フ ラ グ

Z, S 減算の結果によりセットされる。

OV 減算によりオーバーフローが生じた場合、OV=1にセットされる。

FSBD (Floating point SuBtract Double precision)

FSBD n	1 0 0 1 0 1 0 0	アドレス n
--------	-----------------	--------

オペレーション

(レジスタ)<sub>4</sub> ← (レジスタ)<sub>4</sub> - (n, n+1, n+2, n+3)

解 説

データメモリのオペランドによって指定された番地から+3番地までに格納されている倍精度浮動小数点データをレジスタに格納されている倍精度浮動小数点データより減算し、差をレジスタに格納する。

フ ラ グ

Z, S 減算の結果によってセットする。

OV 減算によってオーバーフローが生じた場合、OV=1にセットされる。



## FMLS (Floating point MuLtiPLY Single precision)

FMLS n	1 0 0 0 1 0 0 0	アドレス n
--------	-----------------	--------

### オペレーション

(レジスタ)<sub>2</sub> ← (レジスタ)<sub>2</sub> × (n, n+1)

### 解 説

オペランドによって指定されたデータメモリの番地及び次の番地に格納されている単精度浮動小数点データと、レジスタに格納されている単精度浮動小数点データとを乗算し、積をレジスタに格納する。

### フ ラ グ

Z, S 乗算の結果によりセットされる。

OV 乗算によりオーバーフローが生じた場合、OV = 1 にセットされる。

FMLD (Floating point MuLtiply Double precision)

FMLD n	1 0 0 1 1 0 0 0	アドレス n
--------	-----------------	--------

オペレーション

(レジスタ)<sub>4</sub> ← (レジスタ)<sub>4</sub> × (n, n+1, n+2, n+3)

解 説

データメモリのオペランドによって指定された番地から+3番地までに格納されている倍精度浮動小数点データと、レジスタに格納されている倍精度浮動小数点データとを乗算し、積をレジスタに格納する。

フ ラ グ

Z, S 乗算の結果によってセットされる。

OV 乗算によってオーバーフローが生じた場合、OV=1にセットされる。

FDVS (Floating point DiVide Single precision)

FDVS n 

1 0 0 0 1 1 0 0	アドレス n
-----------------	--------

オペレーション

$(レジスタ)_2 \leftarrow (レジスタ)_2 / (n, n+1)$

解 説

オペランドによって指定されたデータメモリ及び次の番地に格納されている単精度浮動小数点データで、レジスタに格納されている単精度浮動小数点データを除し、商をレジスタに格納する。又、余りは保持されない。

フ ラ グ

- Z, S 除算の結果によってセットされる。
- OV 除算によってオーバーフローが生じた場合、OV=1にセットされる。
- ZD 除数がゼロの場合、ZD=1にセットされる。

## FDVD (Floating point DiVide Double precision)

FDVD n	1 0 0 1 1 1 0 0	アドレス n
--------	-----------------	--------

### オペレーション

(レジスタ)<sub>4</sub> ← (レジスタ)<sub>4</sub> / (n, n+1, n+2, n+3)

### 解 説

データメモリのオペランドによって指定された番地から、+3番地までに格納されている倍精度浮動小数点データで、レジスタに格納されている倍精度浮動小数点データを除し、商をレジスタに格納する。又、余りは保持されない。

### フ ラ グ

- Z, S 除算の結果によってセットされる。
- OV 除算によってオーバーフローが生じた場合、OV=1にセットされる。
- ZD 除数がゼロの場合、ZD=1にセットされる。

FCHS (Floating point CHange Sign)

FCHS

1 0 1 0 0 0 0 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

$$(\text{レジスタ})_2 \leftarrow -(\text{レジスタ})_2$$

解 説

レジスタに格納されている単精度又は倍精度浮動小数点データの符号を反転して、レジスタに格納する。

フ ラ グ

Z, S 符号反転の結果によってセットされる。

## FABS (Floating point ABSolute)

FABS	1 0 1 0 0 0 0 1	0 0 0 0 0 0 0 0
------	-----------------	-----------------

### オペレーション

$$(\text{レジスタ})_{\frac{2}{4}} \leftarrow |(\text{レジスタ})_{\frac{2}{4}}|$$

### 解 説

レジスタに格納されている単精度又は、倍精度浮動小数点データの絶対値をとりレジスタに格納する。

### フラグ

- Z データがゼロの場合、Z = 1 にセットされる。
- S S = 0 にセットされる。

## 7.5 浮動小数点関数命令

SQRS (Square Root of Single precision floating point)

SQRS

1 0 1 0 0 0 1 0
-----------------

0 0 0 0 0 0 0 0
-----------------

オペレーション

$$(\text{レジスタ})_2 \leftarrow \sqrt{(\text{レジスタ})_2}$$

解 説

レジスタに格納されている単精度浮動小数点データの平方根を求め、レジスタに格納する。

SQRS 命令を実行するとデータメモリの243～255番地の内容が破壊される。

フ ラ グ

Z データがゼロの場合、Z=1にセットされる。

FE データが負の場合、FE=1にセットされる。

SQRD (Square Roof of Double precision floating point)

SQRD

1 0 1 1 0 0 1 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

$$(\text{レジスタ})_4 \leftarrow \sqrt{(\text{レジスタ})_4}$$

解 説

レジスタに格納されている倍精度浮動小数点データの平方根を求め、レジスタに格納する。

SQRD命令を実行するとデータメモリの243～255番地の内容が破壊される。

フ ラ グ

Z データがゼロの場合、Z = 1にセットされる。

FE データが負の場合、FE = 1にセットされる。



SINS (SINe of Single precision floating point)

SINS

1 0 1 0 0 0 1 1
-----------------

0 0 0 0 0 0 0 0
-----------------

オペレーション

(レジスタ)<sub>2</sub> ← sin(レジスタ)<sub>2</sub>

解 説

レジスタに格納されている単精度浮動小数点データの正弦を求め、レジスタに格納する。

SINS命令を実行するとデータメモリの246～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

FE データの絶対値が $\frac{\pi}{4} \times 16^\circ$  以上の場合、FE = 1 にセットされる。

SIND (SINE of Double precision floating point)

SIND

1 0 1 1 0 0 1 1	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>4</sub> ← sin(レジスタ)<sub>4</sub>

解 説

レジスタに格納されている倍精度浮動小数点データの正弦を求め、レジスタに格納する。

SIND命令を実行すると、データメモリの246～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

FE データの絶対値が $\frac{\pi}{4} \times 16^{14}$ 以上の場合、FE = 1にセットされる。

COSS (COSine of Single precision floating point)

COSS	1 0 1 0 0 1 0 0	0 0 0 0 0 0 0 0
------	-----------------	-----------------

オペレーション

(レジスタ)<sub>2</sub> ← COS(レジスタ)<sub>2</sub>

解 説

レジスタに格納されている単精度浮動小数点データの余弦を求め、レジスタに格納する。

COSS命令を実行するとデータメモリの246～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

FE データの絶対値が $\frac{\pi}{4} \times 16^\circ$ 以上の場合、FE=1にセットされる。

COSD (COSine of Double precision floating point)

COSD

1 0 1 1 0 1 0 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>4</sub> ← COS(レジスタ)<sub>4</sub>

解 説

レジスタに格納されている倍精度浮動小数点データの余弦を求め、レジスタに格納する。

COSD命令を実行すると、データメモリの246～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

FE データの絶対値が $\frac{\pi}{4} \times 16^{14}$  以上の場合, FE = 1 にセットされる。

TANS (TANgent of Single precision floating point)

TANS	1 0 1 0 0 1 0 1	0 0 0 0 0 0 0 0
------	-----------------	-----------------

オペレーション

(レジスタ)<sub>2</sub> ← tan(レジスタ)<sub>2</sub>

解 説

レジスタに格納されている単精度浮動小数点データの正接を求め、レジスタに格納する。

TANS 命令を実行すると、データメモリの 242 ~ 255 番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

OV 演算によってオーバーフローが生じた場合、OV = 1 にセットされる。

FE データの絶対値が  $\frac{\pi}{4} \times 16^6$  以上の場合、FE = 1 にセットされる。

TAND (TANgent of Double precision floating point)

TAND	1 0 1 1 0 1 0 1	0 0 0 0 0 0 0 0
------	-----------------	-----------------

オペレーション

(レジスタ)<sub>4</sub> ← tan(レジスタ)<sub>4</sub>

解 説

レジスタに格納されている倍精度浮動小数点データの正接を求め、レジスタに格納する。

TAND命令を実行するとデータメモリの242～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

OV 演算によってオーバーフローが生じた場合、OV = 1にセットされる。

FE データの絶対値が $\frac{\pi}{4} \times 16^{14}$ 以上の場合、FE = 1にセットされる。

EXPS (EXPOnential of Single precision floating  
point)

EXPS      

1 0 1 0 0 1 1 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>2</sub> ← e (レジスタ)<sub>2</sub>

解 説

レジスタに格納されている単精度浮動小数点データの、指数関数  
( $e^x$ )を求め、レジスタに格納する。

EXPS命令を実行するとデータメモリの249～255番地の内容が破  
壊される。

フ ラ グ

Z    結果の内容によってセットされる。

S    S = 0 にセットされる。

OV  演算時にオーバーフローが生じた場合、OV = 1 にセットされ  
る。

EXPD (EXPOnential of Double precision floating point)

EXPD

1 0 1 1 0 1 1 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>4</sub> ← e (レジスタ)<sub>4</sub>

解 説

レジスタに格納されている倍精度浮動小数点データの、指数関数 ( $e^x$ ) を求め、レジスタに格納する。

EXPD命令を実行するとデータメモリの249～255番地の内容が破壊される。

フ ラ グ

Z 結果の内容によってセットされる。

S S = 0 にセットされる。

OV 演算時にオーバーフローが生じた場合、OV = 1 にセットされる。



LNS (Natural Logarithm of Single precision  
floating point)

LNS

1 0 1 0 0 1 1 1	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>2</sub> ← LN(レジスタ)<sub>2</sub>

解 説

レジスタに格納されている単精度浮動小数点データの、自然対数を求め、レジスタに格納する。

LNS命令を実行すると、データメモリの247～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

FE データがゼロ又は負の場合、FE=1にセットされる。

LND (Natural Logarithm of Double precision  
floating point)

LND

1 0 1 1 0 1 1 1	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>4</sub> ← LN(レジスタ)<sub>4</sub>

解 説

レジスタに格納されている倍精度浮動小数点データの、自然対数を求め、レジスタに格納する。

LND命令を実行すると、データメモリの247～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

FE データがゼロ又は負の場合、FE=1にセットされる。

LOGS (common LOGarithm of Single precision  
floating point)

LOGS      

1 0 1 0 1 0 0 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

$(レジスタ)_2 \leftarrow \log_{10} (レジスタ)_2$

解 説

レジスタに格納されている単精度浮動小数点データの、常用対数を求め、レジスタに格納する。

LOGS 命令を実行するとデータメモリの247～255番地の内容が破壊される。

フ ラ グ

Z, S    結果の内容によってセットされる。

FE    データがゼロ又は負の場合、FE=1にセットされる。

LOGD (common LOGarithm of Double precision  
floating point)

LOGD

1 0 1 1 1 0 0 0
-----------------

0 0 0 0 0 0 0 0
-----------------

オペレーション

(レジスタ)<sub>4</sub> ← log<sub>10</sub> (レジスタ)<sub>4</sub>

解 説

レジスタに格納されている、倍精度浮動小数点データの常用対数を求め、レジスタに格納する。

LOGD命令を実行するとデータメモリの247～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

FE データがゼロ又は負の場合、FE=1にセットされる。

ASNS (inverse SiNe of Single precision floating  
point)

ASNS

1 0 1 0 1 0 0 1	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>2</sub> ←  $\sin^{-1}$  (レジスタ)<sub>2</sub>

解 説

レジスタに格納されている、単精度浮動小数点データの、逆正弦を求め、レジスタに格納する。

ASNS 命令を実行するとデータメモリの245～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

FE データの絶対値が1より大きい場合、FE=1にセットされる。

ASND (inverse SiNe of Double precision floating point)

ASND	1 0 1 1 1 0 0 1	0 0 0 0 0 0 0 0
------	-----------------	-----------------

オペレーション

(レジスタ)<sub>4</sub> ←  $\sin^{-1}$  (レジスタ)<sub>4</sub>

解 説

レジスタに格納されている、倍精度浮動小数点データの、逆正弦を求め、レジスタに格納する。

ASND命令を実行するとデータメモリの245～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

FE データの絶対値が1より大きい場合、FE=1にセットされる。

ACSS (inverse CoSine of Single precision floating point)

ACSS      

1 0 1 0 1 0 1 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>2</sub> ←  $\text{COS}^{-1}$  (レジスタ)<sub>2</sub>

解 説

レジスタに格納されている単精度浮動小数点データの、逆余弦を求め、レジスタに格納する。

ACSS命令を実行するとデータメモリの245～255番地の内容が破壊される。

フ ラ グ

Z, S    結果の内容によってセットされる。

FE    データの絶対値が1より大きい場合、FE = 1にセットされる。

ACSD (inverse CoSine of Double precision floating point)

ACSD

1 0 1 1 1 0 1 0	0 0 0 0 0 0 0 0
-----------------	-----------------

オペレーション

(レジスタ)<sub>4</sub> ←  $\cos^{-1}$  (レジスタ)<sub>4</sub>

解 説

レジスタに格納されている倍精度浮動小数点データの、逆余弦を求め、レジスタに格納する。

ACSD命令を実行すると、データメモリの245～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

FE データの絶対値が1より大きい場合、FE=1にセットされる。



ATNS (inverse TaNgent of Single precision  
floating point)

ATNS

1 0 1 0 1 0 1 1
-----------------

0 0 0 0 0 0 0 0
-----------------

オペレーション

(レジスタ)<sub>2</sub> ←  $\tan^{-1}$  (レジスタ)<sub>2</sub>

解 説

レジスタに格納されている単精度浮動小数点データの、逆正接を求め、レジスタに格納する。

ATNS命令を実行するとデータメモリの245～255番地の内容が破壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

ATND (inverse TaNgent of Double precision  
floating point)

ATND

1 0 1 1 1 0 1 1
-----------------

0 0 0 0 0 0 0 0
-----------------

オペレーション

(レジスタ)<sub>4</sub> ←  $\tan^{-1}$  (レジスタ)<sub>4</sub>

解 説

レジスタに格納されている倍精度浮動小数点データの、逆正接を求め、  
レジスタに格納する。

ATND命令を実行するとデータメモリの245～255番地の内容が破  
壊される。

フ ラ グ

Z, S 結果の内容によってセットされる。

## 7.6 停止命令

HALT (HALT)

HALT

1 1 1 0 0 0 0 0
-----------------

0 0 0 0 0 0 0 0
-----------------

解 説

演算処理は停止され、その旨ホスト・プロセッサに知らせる。

フ ラ グ

全てのフラグは影響されない。

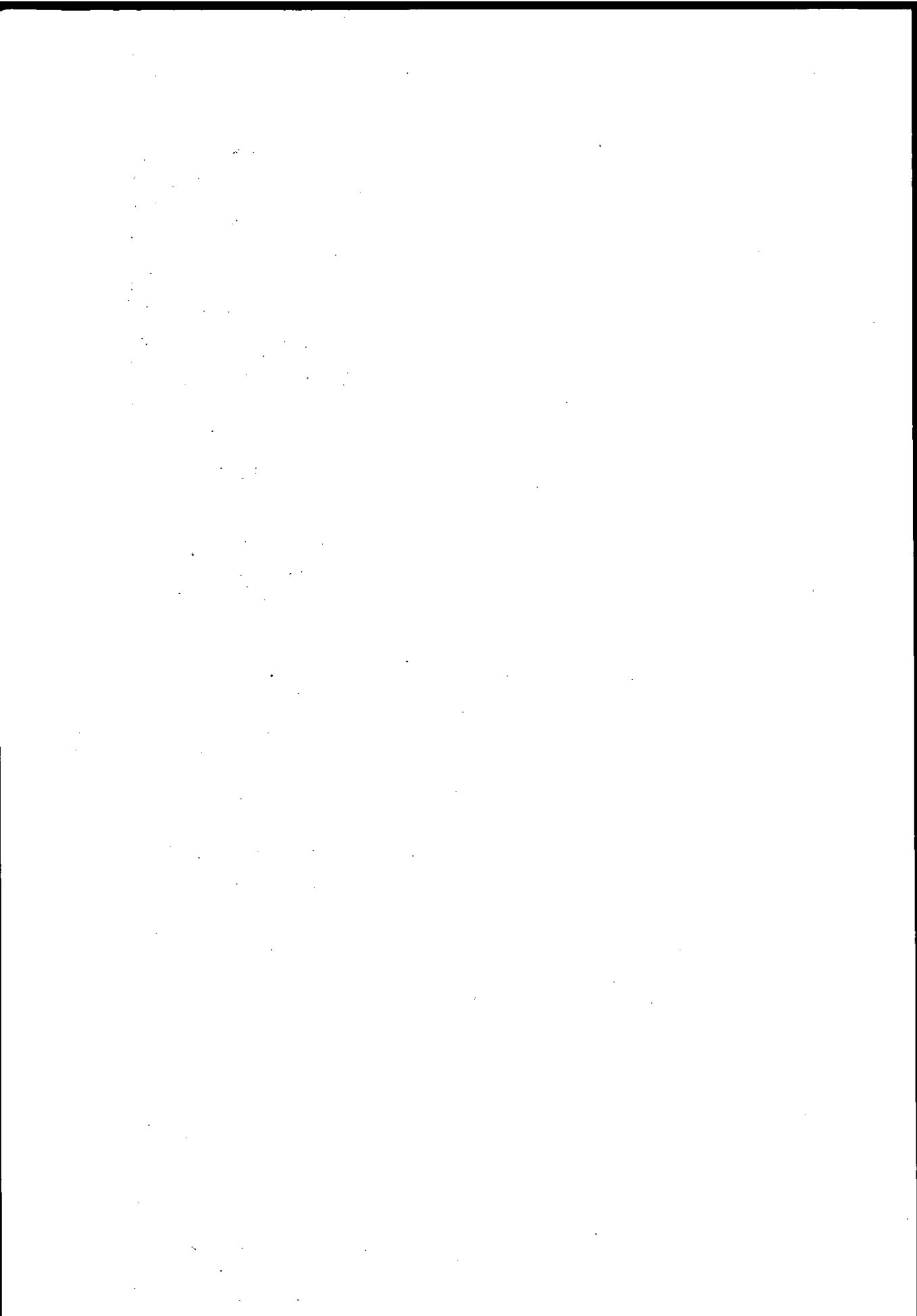
## Ⅷ 命 令 一 覧

MNEMONIC	OP-code	
LODS	00××	16ビット固定小数点データロード
LODD	08××	32ビット固定小数点データロード
LODF	10××	32ビット浮動小数点データロード
LODL	18××	64ビットデータロード
STOS	04××	16ビット固定小数点データストア
STOD	0C××	32ビット固定小数点データストア
STOF	14××	32ビット浮動小数点データストア
STOL	1C××	64ビットデータストア
FLTS	2000	16ビット固定→32ビット浮動
FLT	2800	32ビット固定→32ビット浮動
FLTD	3000	32ビット固定→64ビット浮動
FLTL	3800	64ビット固定→64ビット浮動
FIXS	2400	32ビット浮動→16ビット固定
FIX	2C00	32ビット浮動→32ビット固定
FIXD	3400	64ビット浮動→32ビット固定
FIXL	3C00	64ビット浮動→64ビット固定
ADDS	60××	16ビット固定小数点加算
ADDD	70××	32ビット固定小数点加算
SUBS	64××	16ビット固定小数点減算
SUBD	74××	32ビット固定小数点減算
MULS	68××	16ビット固定小数点乗算

MULD	78××	32ビット固定小数点乗算
DIVS	6C××	16ビット固定小数点除算
DIVD	7C××	32ビット固定小数点除算
CHSS	4800	16ビット固定小数点符号反転
CHSD	5000	32ビット固定小数点符号反転
CHSL	5800	64ビット固定小数点符号反転
FADS	80××	32ビット浮動小数点加算
FADD	90××	64ビット浮動小数点加算
FSBS	84××	32ビット浮動小数点減算
FSBD	94××	64ビット浮動小数点減算
FMLS	88××	32ビット浮動小数点乗算
FMLD	98××	64ビット浮動小数点乗算
FDVS	8C××	32ビット浮動小数点除算
FDVD	9C××	64ビット浮動小数点除算
FCHS	A000	浮動小数点符号反転
FABS	A100	浮動小数点絶対値
SQRS	A200	32ビット浮動小数点平方根
SQRD	B200	64ビット浮動小数点平方根
SINS	A300	32ビット浮動小数点正弦
SIND	B300	64ビット浮動小数点正弦
COSS	A400	32ビット浮動小数点余弦

COSD	B 4 0 0	6 4 ビット浮動小数点余弦
TANS	A 5 0 0	3 2 ビット浮動小数点正接
TAND	B 5 0 0	6 4 ビット浮動小数点正接
EXPS	A 6 0 0	3 2 ビット浮動小数点指数関数
EXPD	B 6 0 0	6 4 ビット浮動小数点指数関数
LNS	A 7 0 0	3 2 ビット浮動小数点自然対数
LND	B 7 0 0	6 4 ビット浮動小数点自然対数
LOGS	A 8 0 0	3 2 ビット浮動小数点常用対数
LOGD	B 8 0 0	6 4 ビット浮動小数点常用対数
ASNS	A 9 0 0	3 2 ビット浮動小数点逆正弦
ASND	B 9 0 0	6 4 ビット浮動小数点逆正弦
ACSS	A A 0 0	3 2 ビット浮動小数点逆余弦
ACSD	B A 0 0	6 4 ビット浮動小数点逆余弦
ATNS	A B 0 0	3 2 ビット浮動小数点逆正接
ATND	B B 0 0	6 4 ビット浮動小数点逆正接
HALT	E 0 0 0	停 止







禁無断転載

昭和55年3月発行

発行所 財団法人 日本情報処理開発協会  
東京都港区芝公園3-5-8  
機械振興会館内  
TEL 03(434)8211 (代表)

印刷所 日本製版株式会社  
東京都新宿区市ヶ谷河田町7  
(フジテレビ内)

