北海道大学工学部研究報告
第94号（昭和54年）

Bulletin of the Faculty of Engineering,
Hokkaido University, No. 94 (1979)

マルチ・マイクロプロセッサ HARPS のための ファームウェア作成支援システム FDS³

恩田邦夫*　津田孝夫**

*F*irmware *D*evelopment *S*upport *S*oftware *S*ystem for the multi-microprocessor HARPS

Kunio ONDA and Takao TSUDA

(Received December 28, 1978)

Abstract

A *F*irmware *D*evelopment *S*upport *S*oftware *S*ystem, FDS³, for HARPS was implemented on the host minicomputer system.

HARPS is a new type of multi-microprocessor system with hierarchical and flexible control structures. This system consists of 9 microprocessors. Each processor has user-writable control memory, called WCS (Writable Control Storage), and executes the control sequences of microprograms in WCS.

FDS³ supports users to develop microprograms. This software system consists of Command Interpreter and five main subsystems: Editor, Micro/Nano-Assembler, Relocatable Loader, Mapping Array Loader, and Micro-Octal-Editor. Micro/Nano-Assembler allows users to interactively correct their programs. Micro-Octal-Editor can directly check and correct microprograms in WCS.

In the implementations of a general purpose firmware system for parallel processing MOSES, FDS³ was used as a firmware development tool and the effectiveness and the usability of this system was proved in this process.

1. はじめに

本学情報工学専攻に設置されているマルチ・マイクロプロセッサ HARPS[1] は，9台のマイクロプロセッサからなる並列演算処理システムであり，並列処理計算機のアーキテクチュアと並列処理アルゴリズムの研究を目的として，本学情報工学専攻と日本ミニコンピュータ㈱により共同開発されたものである[2]。このシステムの特徴は，柔軟な階層制御構造[3]と全面的な WCS (Writable Control Storage：書き替え可能な制御記憶) の採用により広範なエミュレーション機能をもつことであり，HARPS の動作はこの WCS に収められたマイクロプログラムに完全に依存している。このため，HARPS の機能は各々マイクロプログラム・ルーチンによって定義してやらねばならず，マイクロプログラム作成のための支援システムの果たす役割は大きい。マイクロプログラムはファームウェアともよばれる。

本稿では，一般利用者が HARPS のファームウェアを手軽にかつ効率よく作成可能な支援シ

* 電気工学科　演算工学講座
** 京都大学工学部情報工学教室

ステム FDS³ (*F*irmware *D*evelopment *S*upport *S*oftware *S*ystem) を紹介する。FDS³ は，コマンド・インタプリタおよび13の機能モジュールからなり，利用者によるソース・プログラムの作成，WCS へのローディング，実行などを支援するほか，ファイル管理やデバッグのための機能をもつ。

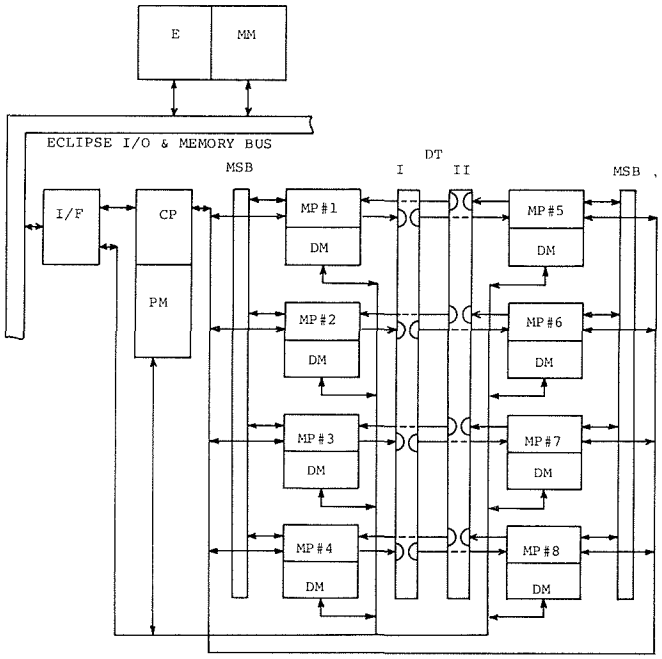
以下では，まず HARPS の概要についてふれた後，FDS³ の構成および機能について述べる。

2. HARPS の概要

HARPS はミニコンピュータ ECLIPSE S/200 をホスト計算機とし，8台のマイクロプロセッサ (MP) とこれを制御する1台のコントロール・プロセッサ (CP) の合計9台のマイクロプロセッサより構成される集中制御型のマルチ・マイクロプロセッサ・システムである。HARPS の構成の概要を図1に示す。各マイクロプロセッサは，それぞれ書き替え可能な制御記憶 (WCS) をもち，WCS に収められたマイクロプログラムの制御のもとで独立に動作可能である。とくに MP のマイクロプログラムをナノプログラムとよぶ（以下，MP のマイクロプログラムを nP，CP のそれを μP と略す)。

8台の MP は2つのグループに分けられ，同一のグループ内のプロセッサ間データ転送と，異グループに属するプロセッサ間データ転送のために，それぞれマトリクス・スイッチ・バスとデータ・トランスポータが2組ずつ用意されている。表1に HARPS の仕様の概略を示す。

表中の MA は CP の機械語命令を μP ルーチンのエントリ・ポインタに展開するためのデコーダとしてのメモリであり，NMA は CP より送られてくるマクロ・コマンドを8個の nP ルーチンのエントリ・ポインタに並列展開するための 8-way の並列デコーダとしてのメモリであ



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, MSB: Matrix Switch Bus

図1 HARPS の構成

表1　HARPS の仕様の概略

マイクロ命令サイクル	400 ns
CP	1台
CPU	16ビット，16レジスタ，Am 2901×4
マッピング・アレイ (MA)	8ビット×128語
WCS	55ビット×256語
プログラム・メモリ	16ビット×256語
NMA	8ビット×8フィールド×256語
MP	8台 (MP 1～MP 8)
CPU	16ビット，16レジスタ，Am 2901×4
WCS	54ビット×256語
データ・メモリ	16ビット×256語
外部メモリ	16ビット×4K 語 (MP 1～MP 4)

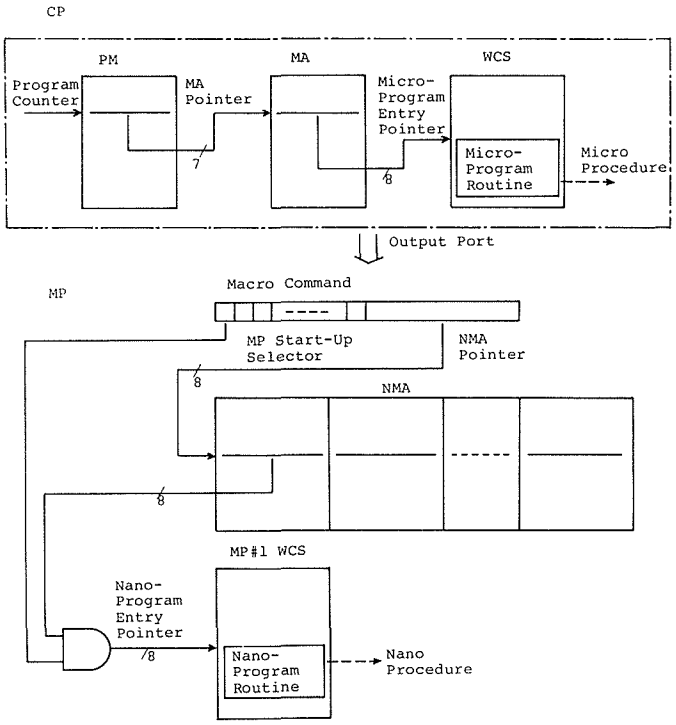


図2　MA，NMA による命令の階層制御構造

る。MA，NMA を HARPS の制御構造のなかでどのように位置づけるかはファームウェアにより決定される。MA，NMA による機械語命令の階層制御構造の様子を図2に示した。

また CP にはプログラム・メモリ (PM)，各 MP にはデータ・メモリ (DM) とよばれるローカル・メモリがある。

HARPS のこれらの各領域は，ホスト計算機である ECLIPSE のメイン・メモリ上にマッピングされており，ECLIPSE のメモリ参照命令によってこれらの領域のアクセスが可能である。HARPS のファームウェアを作成するうえで対象となる領域は，CP および各 MP の WCS のほかに MA と NMA が含まれる。FDS[3] では，ECLIPSE のメモリ参照命令を用いてこれらの領域へのプログラム・ローディングをおこなう。

3. FDS[3] の構成と機能

FDS[3] は HARPS のホスト計算機 ECLIPSE 上に作成され，この計算機のコンソール・タイプライタからのインタプリタ形式のコマンドにより動作する。コマンドはインタプリタで解釈され，各機能モジュールで実行される。コマンドおよびその機能の概略を表2に示す。ファームウェアの作成は図3に示すような手順によりおこなうことができる。

以下，各モジュールの機能について説明する。

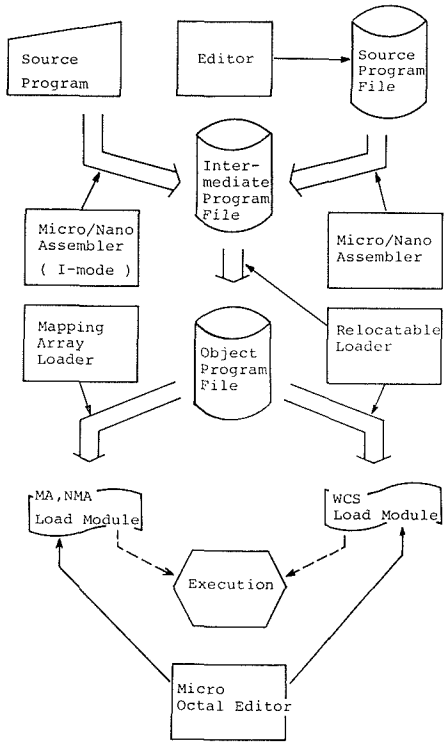


図3 ファームウェア作成のための機能モジュール

表2 システム・コマンドと機能

MASM	μP のアセンブル 中間プログラムの生成
NASM	nP のアセンブル 中間プログラムの生成
RLDR	中間プログラムの結合 オブジェクト・プログラムの生成，WCS へのロード
MLDR	MA，NMA データの生成とロード
EDIT	ソース・プログラムの作成と修正
MDEB	マイクロ・オクタル・エディタによる HARPS の各領域の参照と修正
LIST	プログラムのリスト
FREE	各プロセッサの WCS の使用状態の出力
INIT	HARPS の各領域の初期化
RUN	HARPS の実行
XFER	ソース・ファイルの転送
DELET	ソース・ファイルの削除
BYE	システムの終了

1) マイクロ/ナノ・アセンブラ

HARPS の μP，nP はそれぞれその記述形式にしたがって，マイクロ命令 (μI)，ナノ命令 (nI) およびアセンブラ命令を用いて記述される。μI，nI にはニモニック・コードが用意されている。アセンブラ命令はマイクロ/ナノ・アセンブラ (以下アセンブラとよぶ) に対する制御情報を与える。μI，nI の形式およびニモニック・コードを付録 A，B に示した。また付録 C に μP の記述形式の概略を BNF 記法で示した。この記述形式にしたがって記述されたプログラムをソース・プログラムとよぶ。アセンブラはこのソース・プログラムを翻訳し，中間プログラムを生成する。

中間プログラムは μI，nI のブランチ・アドレス (BA) 部に記述されたラベル・シンボルをビット・パターンに翻訳せずシンボルのまま保存している。図4に中間プログラム・ファイルの形式を示す。この BA 部は次に述べるリロケータブル・ローダにより翻訳される。

一般に，プログラムの作成ではプログラム全体をいくつかのリロケータブルなモジュールに分

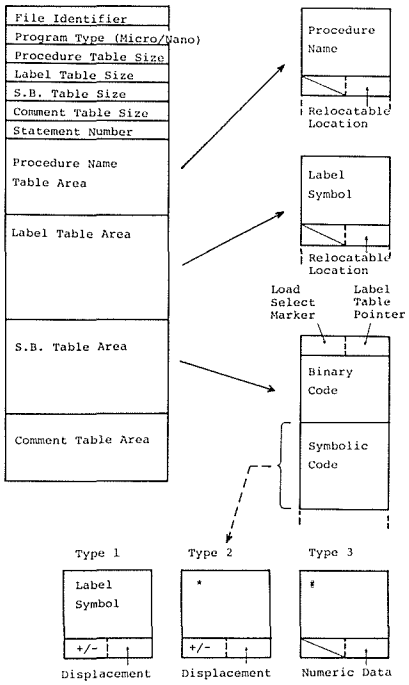


図4　中間プログラム・ファイルの構造

```
Line No.      Source Program
10  D, IOR, LBF; A0, B3, IMDT; #3
20  B, IOR, LQF; SWOT
15  AB, ADD, LSF; B6                ; insert program
                                      between 10 and 20
20  B, AND, LQF; SWOT               ; correct 20
10                                  ; delete 10
```

図5　I-モード・アセンブラの使用例

```
.C
.C ### SWAPPING TRANSFER (MP1) ###
.C
.P SW1,SW2,SW3,SW4,SW5,SW6,SW7
.M OPORT= D,IOR,LBF;B2,XI,IMDT,XO,SWOT
SW1:      OPORT' ;;#42
          B,AND,LRC,CSET;B1;UCB,!SWAP
SW2:      OPORT' ;;#44
          A,ADC,LSF;A0,B1;UCB,!SWAP
SW3:      OPORT' ;;#50
          B,ADC,LBF;B1;UCB,!SWAP
.M J= B,ADC,LSF;B1;UCB,!SWAP
SW4:      OPORT' ;;#61
          J'
SW5:      OPORT' ;;#62
          J'
SW6:      OPORT' ;;#64
          J'
SW7:      OPORT' ;;#70
          J'
.M DORN= D,IOR,NLD;;PUSH,UCB
SWAP:     DORN' DA;A2,XI,SWIN,XO,OPRG;!OPERAN
          DORN' ;PR,#20,PO,#0;!RVDT
          DORN' ;PI,#0,PW,#20;!AVAI
          DORN' ;PR,#24,PO,#0;!RVDT
          DORN' ;PI,#0,PW,#24;POP
RVDT:     DORN' ;IX,RVDT;POP,BFZ,*+0
AVAI:     DA,AND,NLD;A1,XI,DASV;POP,BFZ,*+0
OPERAN:   D,IOR,LBF;B8,XI,RDCP,XO,SWOT
          AB,AND,LBF;A11,B8
          DA,AND,RRC,CRST;A11,B4,XI,SWIN
          DA,AND,LBF;A4,B15,XI,IMDT,XO,SWOT;#3
          DA,IOR,NLD;A15,XI,SWIN,XO,RPOT;POP
.F
```

図6　nP の記述例（MP 1 と MP 2～8 の間でのデータ転送のプログラム）

割し，これをローダにより結合する手法が用いられる。この場合，モジュール間でのラベルの参照には外部参照宣言を必要とする。本アセンブラでは，中間プログラムを採用することにより外部参照宣言をすることなく他のモジュールで使用しているラベルが BA 部に記述できる。また，アセンブラは１パスでおこなえる。

アセンブラへのソース・プログラムの入力方法は二通りある。一つは，あらかじめエディタを用いて作成したソース・プログラム・ファイルから入力する方法であり，もう一つはコンソールタイプライタ上から入力する方法（インタラクティブ・モード：I-モード）である。

I-モード・アセンブラでは，プログラムの前に文番号を付加したソースプログラムに対して，1行ずつアセンブルをおこない，中間プログラムに翻訳していく。この文番号は行の修正・挿入・削除に対して用いられる。その使用例を図５に示す。

I-モード・アセンブラを用いることにより，プログラムの作成とアセンブルは同時におこなえる。I-モード・アセンブルでは，アセンブラ命令がコマンドとして使える他，出来上ったプログラムのリスティングやファイルへの出力またファイルからの入力のためのコマンドが用意されている。行の修正・挿入・削除では BA 部の内容は動的に変化するため，プログラムの作成を完了するまでビット・パターンの生成ができない。この場合でもここで採用した中間プログラム形式によりロード時に BA 部を翻訳することで解決される。また中間プログラムからソース・プログラムを再現することも可能である。

アセンブラではソース・プログラム作成の効率化をはかるため，I-モード・アセンブラの他マクロ定義機能をもっている。これは，あらかじめ μI，nI をマクロ・シンボルによって定義して

おくことにより，以後マクロ・シンボルを用いて μI, nI の記述ができる。

図6に nP による記述の例を示す。図中の .P はプロセデュア宣言である。図2でも示したように，μP, nP の実行はそれぞれ MA, NMA で生成されたエントリ・ポインタの値をスタート・アドレスとして開始される。μP, nP 中でプロセデュア宣言されたラベル・シンボルがこのエントリ・ポインタとして登録される。

2) リロケータブル・ローダ

リロケータブル・ローダ（以下ローダとよぶ）は，2つのフェーズよりなる。フェーズ1ではアセンブラで生成された1ないし2個以上の中間プログラムを結合し，オブジェクト・プログラムを生成する。このときアセンブラでは未翻訳だったBA部が翻訳され，ビット，パターンが完成される。WCS は CP で 55 ビット/1 語，MP で 54 ビット/1 語であるので，ECLIPSE 上では1つの μI または nI に対して4語が割当てられる。オブジェクト・プログラム・ファイルの形式を図7に示す。図中のプロセデュア・ネーム・テーブルは，次に述べるマッピング・アレイ・ローダで MA または NMA データに変換された後，MA または NMA へロードされる。

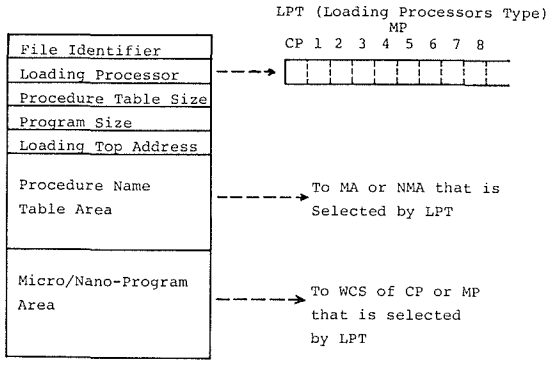


図7 オブジェクト・プログラム・ファイルの構造

フェーズ2では，図中のプログラム領域の μP または nP が CP または MP の WCS にロードされる。このときのロードすべきプロセッサおよび WCS の先頭番地は，コマンドにより与えられる。

ローダには選択的ローディングの機能がある。これは μP または nP の記述時に，ロード・セレクト・マーカ (LSM) が付加された μI または nI についておこなわれる。

中間プログラムでは，LSM が付加された μI または nI については LSM ビットがセットされており（図4），コマンドによって選択的ローディングの指定がされると，この LSM ビットがセットされている μI または nI については WCS へのローディングがおこなわれない。この機能は次のような場合有効である。ファームウェア作成時には，デバッグ用として特別なルーチンを必要とする場合がある。一方，デバッグ終了時にはこのルーチンを必要としない場合など，このルーチンに LSM を付加してソース・プログラムを作成しておき，デバッグ時には選択的機能をつけずに WCS にロードしておき，デバッグ終了時にはこの機能をつけてロードすれば，ソース・プログラムから修正する必要がなくなる。

3) マッピング・アレイ・ローダ

マッピング・アレイ・ローダは，オブジェクト・プログラム・ファイルのプロセデュア・ネーム・テーブルが使用される。最初に，プロセデュア・ネームとして登録されているラベル・シンボルが全てコンソール上に出力されたのち，次にそれが MA または NMA の何番地に入るかが聞かれる。利用者は，MA または NMA の番地を記述した後，ラベル・シンボルを記述すれば，そのラベルに割当てられた μP または nP の番地が，MA または NMA のデータとしてロードされる。

4) マイクロ・オクタル・エディタ

マイクロ・オクタル・エディタでは，直接 HARPS の各領域のデータ参照および修正が可能である。エディタの使用はコンソール上でおこなわれ，モニタ・コマンドでエディタ・モードへの移行をおこなったのち，エディタ・コマンドを入力する。エディタ・コマンドには，プロセッサ指定および領域指定の命令があり，各領域のオクタル・データによる参照・修正ができる。

4) その他の機能モジュール

ソース・プログラムの作成とファイル管理用としてシンボリック・エディタ，ファイルの転送や削除のための機能モジュールがある。またプログラムのリスト機能や，WCS の使用状態管理機能，HARPS の実行機能のためのモジュールが用意されている。

4. おわりに

本システムは ECLIPSE のアセンブラ命令を用いて作成され，実行プログラム領域 20 K 語，作業領域 10 K 語よりなる。ソース・プログラムは HARPS の各プロセッサの WCS 容量 256 語分のプログラムが作成可能である。

HARPS のファームウェア・システムとしては，現在，当研究室において“汎用ファームウェア・システム・MOSES (Version 2)”が完成しており[6,7]，本システムはその支援システムとしてその有用性が確かめられた。

HARPS はそのアーキテクチェアに柔軟性があり，また WCS も利用者に全面的に開放されているので，今後も種々のファームウェア・システムが提案されていくものと思われる。本システムがそれらのファームウェア・システムの開発にあたり，強力な支援となることを望む次第である。

最後に本稿を書くにあたり多くの助言をいただいた田中譲氏，ならびに利用者として本システムに対し種々の問題点や誤りを指摘してくれた当研究室の院生・学生に感謝します。

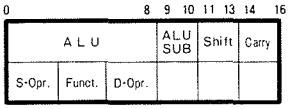
引用文献

1) Tanaka, Y. *etal.*: “HARPS (Hokkaido University Array Processor System): A New Hierarchical Array Processor System”, Proceedings of 2nd EUROMICRO Symposium, North Holland, 1976.
2) 宮下他：“HARPS の Hardware Implementation”，情報処理学会第 17 回全国大会，講演番号 22，1977.
3) 田中他：“HARPS の階層制御構造とファームウェア・システム”，同上，講演番号 21，1976,
4) 恩田他：“マルチ・マイクロプロセッサ HARPS のためのマイクロアセンブラ”，北大津田研技報，TR-TL 7707-0003，1977.
5) 恩田他：“HARPS のためのインタラクティブ・マイクロアセンブラ”，情報処理学会第 18 回全国大会，講演番号 277，1977.
6) 田中他：“HARPS の汎用並列処理ファームウェア・システム MOSES (Version 2): I. マルチプロセス”情報処理学会第 18 回全国大会，講演番号 278，1977.
7) 恩田他：“HARPS の汎用並列処理ファームウェア・システム MOSES (Version 2): II. 並列命令群”，情報処理学会第 18 回全国大会，講演番号 279，1977.
8) 日本ミニコン㈱：“多重マイクロプロセッサ装置暫定取扱い説明書” 1976.
9) 恩田：“HARPS MICRO-ASSEMBLER MANUAL，北大津田研技報，771201-003，1977.
他に HARPS の紹介記事として.
10) 田中他：“並列処理計算機アーキテクチュアと並列処理アルゴリズム研究のための汎用エミュレータ；マルチマイクロプロセッサ・システム HARPS” インターフェース No. 14 1978 年 2 月号.

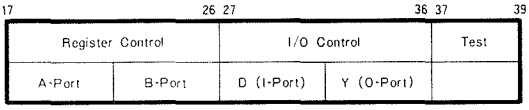
付　録　A　CPのマイクロ命令

〔図A-1〕 マイクロ命令のフォーマット

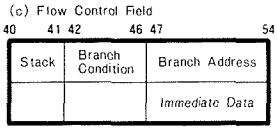
(a) Operation Control Field



(b) Operand Control Field



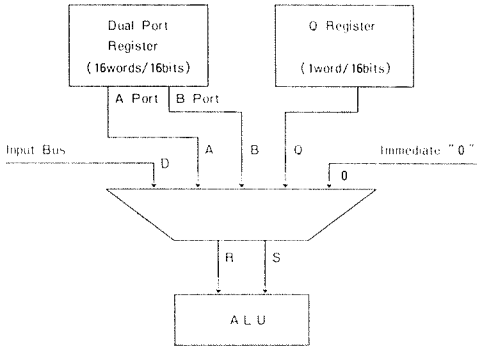
(c) Flow Control Field



〔表A-1〕 マイクロ命令のニモニック・コード

Operation Control Field				Operand Control Field					Flow Control Field	
ALU S-Opr.	ALU Funct.	ALU D-Opr.	Carry	A-Port	B-Port	I-Port	O-Port	Test	Stack	Branch Condition
AQ	ADX	LQF	CHLD	A0	B0	IREG	NOUT	NONE	NSTK	NOB
AB	ADD	NLD	CRST	\	\	SWIN	SWOT	TES1	PUSH	UCB
Q	ADC	LBA	CSET	A15	B15	ITDV	TSTC	TES2	POP	BCZ
B	CSUX	LBF	CCMT	DCA0	DCB0	INDT	OTDT	TES3		BCO
A	CSUB	LXX	CXXX	\	\	COSW	MADT	TESX		BFZ
DA	CSUC	LXF		DCA15	DCB15	MERD	MAIT			BFNZ
DQ	SUX	LSQ				MLCI	MEWD			BFN
D	SUB	LRQ				STAD	MLCO			BFP
	SUC	LSQC				NOIN	MPST			BTO
	IOR	LRQC				GCP1	XDMP			BTZ
	AND	LSF				\	XTMP			BFOD
	CAND	LRF				GCP8	DTST			BLCO
	XOR	LSC				SIP1	DSWI			BOFO
	COIN	LRC				\	IRPE			BOFZ
		RSQ				SIP8	IRPD			BGCS
		RRQ					HALT			BAFZ
		RSQC					OFP1			ALCB
		RRQC					\			NALC
		RSF					OFP8			
		RRF					CSDP			
		RSC					CMRD			
		RRC					ALMP			
		XXQ								
		XXQC								
		XXF								
		XXC								

〔図A-2〕 ALU ソース・オペランド制御の機能



Micro-Mnemonic	I_0	I_1	I_2	ALU S-Operand	
				R	S
AQ	0	0	0	A	Q
AB	0	0	1	A	B
ZQ	0	1	0	0	Q
ZB	0	1	1	0	B
ZA	1	0	0	0	A
DA	1	0	1	D	A
DQ	1	1	0	D	Q
DZ	1	1	1	D	0

オペレーション・コントロール

〔表A-3〕 ALU ファンクション・コントロール

Micro-Mnemonic	ALU Funct			ALU SUB		Function
	I_3	I_4	I_5	I_{15}	I_{13}	
ADX	0	0	0	0	0	R+S+C
ADD	0	0	0	0	1	R+S
ADC	0	0	0	1	0	R+S+1
CSUX	0	0	1	0	0	S−R+C
CSUB	0	0	1	0	1	S−R−1
CSUC	0	0	1	1	0	S−R
SUX	0	1	0	0	0	R−S+C
SUB	0	1	0	0	1	R−S−1
SUC	0	1	0	1	0	R−S
IOR	0	1	1	0	0	R∨S
AND	1	0	0	0	0	R∧S
CAND	1	0	1	0	0	$\overline{R}$∧S
XOR	1	1	0	0	0	R∀S
COIN	1	1	1	0	0	$\overline{R \vee S}$

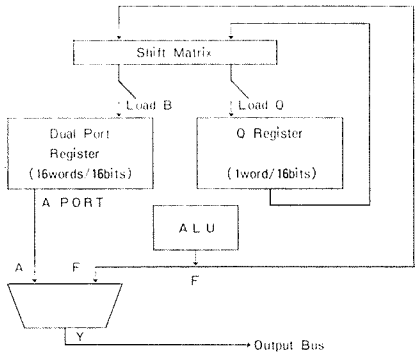
〔表A-9〕 テスト・コントロール

Micro-Mnemonic	I_{17}	$I_{18}/\neq 8$	$I_{19}/\neq 9$	Function
NONE	0	0	0	No Test
TES1	0	0	1	TEST 1
TES2	0	1	0	TEST 2
TES3	0	1	1	TEST 3
TESX	1	—	—	

〔表A-4〕 ALU Destination Control

Micro-Mnemonic	ALU D-Operand Shift						Shift Mode			Y
	I_6	$I_7/\neq 8$	I_8	I_{11}	I_{12}	$I_{13}/\neq 9$	Shift	Load B	Load Q	
LOF	0	0	0	0	—	—	None	None	F	F
NLD 注)	0	0	1	0	—	—	None	None	None	F
LBA	0	1	0	0	—	—	None	F	None	A
LBF	0	1	1	0	—	—	None	F	None	F
LXX	0	*	0	1	—	—	None	N/F	F/N	F/A
LXF	0	*	1	1	—	—	None	N/F	None	F
LSQ	1	0	0	0	0	0	Left Shift with Q	F_L	Q_L	F
LRQ	1	0	0	0	0	1	Left Rotate with Q	F_L	Q_L	F
LSQC	1	0	0	0	1	0	Left Shift with Q and Carry	F_L	Q_L	F
LRQC	1	0	0	0	1	1	Left Rotate with Q and Carry	F_L	Q_L	F
LSF	1	0	1	0	0	0	Left Shift	F_L	None	F
LRF	1	0	1	0	0	1	Left Rotate	F_L	None	F
LSC	1	0	1	0	1	0	Left Shift with Carry	F_L	None	F
LRC	1	0	1	0	1	1	Left Rotate with Carry	F_L	None	F
RSQ	1	1	0	0	0	0	Right Shift with Q	F_R	Q_R	F
RRQ	1	1	0	0	0	1	Right Rotate with Q	F_R	Q_R	F
RSQC	1	1	0	0	1	0	Right Shift with Q and Carry	F_R	Q_R	F
RRQC	1	1	0	0	1	1	Right Rotate with Q and Carry	F_R	Q_R	F
RSF	1	1	1	0	0	0	Right Shift	F_R	None	F
RRF	1	1	1	0	0	1	Right Rotate	F_R	None	F
RSC	1	1	1	0	1	0	Right Shift with Carry	F_R	None	F
RRC	1	1	1	0	1	1	Right Rotate with Carry	F_R	None	F
XXQ	1	*	0	1	0	*	L/R S/R with Q	F_X	Q_X	F
XXQC	1	*	0	1	1	*	L/R S/R with Q and Carry	F_X	Q_X	F
XXF	1	*	1	1	0	*	L/R S/R	F_X	None	F
XXC	1	*	1	1	1	*	L/R S/R with Carry	F_X	None	F

$I_7/\neq 8$: $I_{11}=0$ のとき I_7
$I_{11}=1$ 〃 $\neq 8$(命令のBIT 8)
$I_{13}/\neq 9$: $I_{11}=0$ のとき I_{13}
$I_{11}=1$ 〃 $\neq 9$(命令のBIT 9)



〔表A-5〕 キャリー・コントロール

Micro-Mnemonic	I_{14}	$I_{15}/\neq 10$	$I_{16}/\neq 11$	Carry Function
CHLD	0	0	0	$C_{n+1}=C_n$ Hold
CRST	0	0	1	$C_{n+1}=0$ Reset
CSET	0	1	0	$C_{n+1}=1$ Set
CCMT	0	1	1	$C_{n+1}=\overline{C_n}$ Complement
C-XXX	1	*	*	Instruction Definable

I_{15} $\neq 10$, $I_{16}/\neq 11$: $I_{14}=0$ のとき I_{15}, I_{16}
$I_{14}=1$ 〃 $\neq 10$, $\neq 11$

オペランド・コントロール

〔表A-6〕 Aレジスタ・コントロール（Bレジスタも類似）

Micro-Mnemonic	I_{17}	I_{18}/A0	I_{19}/A1	I_{20}/A2	I_{21}/A3	A-Port に Select される Register の Address
A 0	0	0	0	0	0	0_8
A 1	0	0	0	0	1	1
A 2	0	0	0	1	0	2
A 3	0	0	0	1	1	3
A 4	0	0	1	0	0	4
A 5	0	0	1	0	1	5
A 6	0	0	1	1	0	6
A 7	0	0	1	1	1	7
A 8	0	1	0	0	0	10
A 9	0	1	0	0	1	11
A 10	0	1	0	1	0	12_8
A 11	0	1	0	1	1	13
A 12	0	1	1	0	0	14
A 13	0	1	1	0	1	15
A 14	0	1	1	1	0	16
A 15	0	1	1	1	1	17
DCA 0	1	0	0	0	0	0_8 V DCA 0 ～ 3
DCA 1	1	0	0	0	1	1 〃 〃
DCA 2	1	0	0	1	0	2 〃 〃
DCA 3	1	0	0	1	1	3 〃 〃
DCA 4	1	0	1	0	0	4 〃 〃
DCA 5	1	0	1	0	1	5 〃 〃
DCA 6	1	0	1	1	0	6 〃 〃
DCA 7	1	0	1	1	1	7 〃 〃
DCA 8	1	1	0	0	0	10 〃 〃
DCA 9	1	1	0	0	1	11 〃 〃
DCA 10	1	1	0	1	0	12 〃 〃
DCA 11	1	1	0	1	1	13 〃 〃
DCA 12	1	1	1	0	0	14 〃 〃
DCA 13	1	1	1	0	1	15 〃 〃
DCA 14	1	1	1	1	0	16 〃 〃
DCA 15	1	1	1	1	1	17 〃 〃

DC REG. A2のオプション指定の時は DCA0, DCA1は0.

〔表A-7〕 Input Port Control

Micro-Mnemonic	I_{27}	I_{28}	I_{29}	I_{30}	I_{31}	Address (8)	Input Port
IREG	0	0	0	0	0	0	Instruction Register
SWIN	0	0	0	0	1	1	Swapping Register
ITDV	0	0	0	1	0	2	Interrupt Device
INDT	0	0	0	1	1	3	In Data
IMDT	0	0	1	0	0	4	Immediate Data
COSW	0	0	1	0	1	5	Console Switch
MERD	0	0	1	1	0	6	Memory Read Data
MLCI	0	0	1	1	1	7	Micro Loop Counter
STAD	0	1	0	0	0	10	Start Address/Info
RDOF	0	1	0	0	1	11	Read Offset Register
GCSM	0	1	1	1	0	16	Global Command Sense MP
HWST	0	1	1	1	1	17	Hardware Status
GCP 1	1	0	0	0	0	20	Global Command MP 1
GCP 2	1	0	0	0	1	21	〃 MP 2
GCP 3	1	0	0	1	0	22	〃 MP 3
GCP 4	1	0	0	1	1	23	〃 MP 4
GCP 5	1	0	1	0	0	24	〃 MP 5
GCP 6	1	0	1	0	1	25	〃 MP 6
GCP 7	1	0	1	1	0	26	〃 MP 7
GCP 8	1	0	1	1	1	27	〃 MP 8
SIP 1	1	1	0	0	0	30	Status/Info MP 1
SIP 2	1	1	0	0	1	31	〃 MP 2
SIP 3	1	1	0	1	0	32	〃 MP 3
SIP 4	1	1	0	1	1	33	〃 MP 4
SIP 5	1	1	1	0	0	34	〃 MP 5
SIP 6	1	1	1	0	1	35	〃 MP 6
SIP 7	1	1	1	1	0	36	〃 MP 7
SIP 8	1	1	1	1	1	37	〃 MP 8

〔表A-8〕 Output Port Control

Micro-Mnemonic	I_{32}	I_{33}	I_{34}	I_{35}	I_{36}		Output Port
NOUT	0	0	0	0	0	0	No Output
SWOT	0	0	0	0	1	1	Swapping Register
TSTC	0	0	0	1	0	2	Test Tool Code
OTDT	0	0	0	1	1	3	Out Data
MADT	0	0	1	0	0	4	Memory Address (Data)
MAIT	0	0	1	0	1	5	Memory Address (Inst.)
MEWD	0	0	1	1	0	6	Memory Write (Data)
MLCO	0	0	1	1	1	7	Micro Loop Counter
MPST	0	1	0	0	0	10	MP Start
XDMP	0	1	0	0	1	11	X'fer Destination MP
XTMP	0	1	0	1	0	12	X'fer Data To MP
DTST	0	1	0	1	1	13	Data/Status
DSWI	0	1	1	0	0	14	Data/Status with Int
IRPE	0	1	1	0	1	15	Interrupt Enable
IRPD	0	1	1	1	0	16	Interrupt Disable
HALT	0	1	1	1	1	17	Halt
OFP1	1	0	0	0	0	20	Offset Register MP 1
OFP2	1	0	0	0	1	21	〃 MP 2
OFP3	1	0	0	1	0	22	〃 MP 3
OFP4	1	0	0	1	1	23	〃 MP 4
OFP5	1	0	1	0	0	24	〃 MP 5
OFP6	1	0	1	0	1	25	〃 MP 6
OFP7	1	0	1	1	0	26	〃 MP 7
OFP8	1	0	1	1	1	27	〃 MP 8
CSDP	1	1	0	0	0	30	Console Display
CMRD	1	1	0	0	1	31	Com. Req. Reset
ALMP	1	1	0	1	0	32	All MP Reset
LDOF	1	1	0	1	1	33	Load Offset Register
STOF	1	1	1	0	0	34	Start Offset Register Reference Mode
CLOF	1	1	1	0	1	35	Clear Offset Register Reference Mode

フロー・コントロール

〔表A-10〕 スタック・コントロール

Micro-Mnemonic	I_{30}	I_{31}	Function
NSTK	0	0	No Stack
PUSH	0	1	Push
POP	1	0	Pop

〔表A-11〕 Branch Condition Control

Micro-Mnemonic	サイクル延長の有無	I_{42}	I_{43}	I_{44}/$\neq 13$	I_{45}/$=14$	I_{46}/$=15$	Function
NOB	×	0	0	0	0	0	No Branch
UCB	×	0	0	0	0	1	Unconditional Branch
BCZ	○	0	0	0	1	0	Branch If Carry = 0
BCO	○	0	0	0	1	1	Branch If Carry = 1
BFZ	○	0	0	1	0	0	Branch If F = 0
BFNZ	○	0	0	1	0	1	Branch If F ≠ 0
BFN	○	0	0	1	1	0	Branch If F < 0
BFP	○	0	0	1	1	1	Branch If F ≥ 0
BTO	×	0	1	0	0	0	Branch If TEST = 1
BTZ	×	0	1	0	0	1	Branch If TEST = 0
BFOD	○	0	1	0	1	0	Branch If F = ODD
BLCO	×	0	1	0	1	1	Branch If Micro Loop Counter Not Overflow
BOFO	○	0	1	1	0	0	Branch If Overflow = 1
BOFZ	○	0	1	1	0	1	Branch If Overflow = 0
BGCS	×	0	1	1	1	0	Branch If Grobal Command Sense
BAFZ	×	0	1	1	1	1	Branch If Available Flag = 0
ALCB	×	1	0	1	1	1	
NALC	×	1	1	1	1	1	

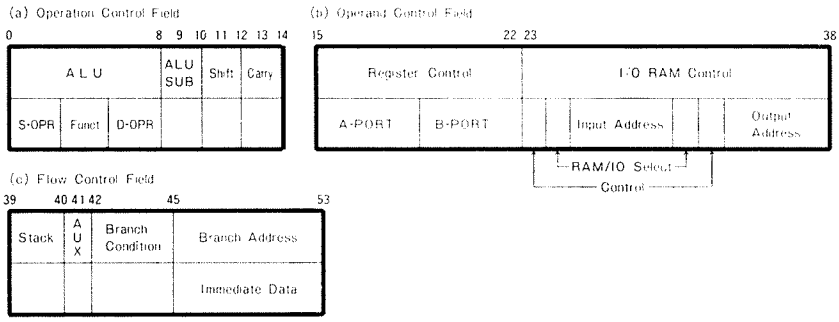
I_{44}/$\neq 13$, I_{45}/$=14$, I_{46}/$=15$

$I_{42}=0$ のとき I_{44}, I_{45}, I_{46}

$I_{42}=1$ のとき $(I_{44} \wedge \neq 13)$, $(I_{45} \wedge =14)$, $(I_{46} \wedge =15)$

付　録　B　　MPのナノ命令

〔図B-1〕 ナノ命令フォーマット



〔表B-1〕 ナノ命令ニモニック・コード一覧表

Operation Control Field				Operand Control Field						Flow Control Field		
ALU S-Opr	ALU Funct	ALU D-Opr	Carry	A-Port	B-Port	I-Select	I-Address	O-Select	O-Address	Stack	Branch Condition	Branch Address
AQ	ADX	LQF	CHLD	A0	B0	XI	SWIN	XO	SWOT	NSTK	NOB	DA
AB	ADD	NLD	CRST	\	\	PI	PRIN	PO	RPOT	PUSH	BCO	PM
Q	ADC	LBA	CSET	A15	B15	XR	IMDT	XW	ITEN	POP	BCZ	
B	CSUX	LBF	CCMT			PR	COSW	PW	CODP		BFZ	
A	CSUB	LSQ					OFRI		OFRO		BFNZ	
DA	CSUC	LRQ					IPRG		OPRG		BFN	
DQ	SUX	LSQC					NLCI		NLCO		BFP	
D	SUB	LRQC					SRQI		ITDS	XDA0	UCB	
	SUC	LSF					SAKI		SRQO	XDA1	BTO	
	IOR	LRF					SYIN		SAKO	XDA01	BTZ	
	AND	LSC					EICM		SYOT	XDA2	BFOD	
	CAND	LRC					DASV		EITM	XDA02	BLCNO	
	XOR	RSQ					RVDT		EMAR	XDA12	BOVF	
	COIN	RRQ					EMRD		EMAW	XDA012	BNOVF	
		RSQC					EMST		EMWD	XDA3	BAFO	
		RRQC					IDWD		EMCL	XDA03	BAFZ	
		RSF					RDS1		HALT	XDA13		
		RRF					RDS2		DTTR	XDA013		
		RSC					RDS3		XDS1	XDA23		
		RRC					RDCP		XDS2	XDA023		
							RDA0		XDS12	XDA123		
							RDA1		XDS3	XDAAL		
							RDA2		XDS13			
							RDA3		XDS23			
									XDS123			
									XDCP			

1.オペレーションコントロール

1.a　ALU Source Operand
　$I_0 \sim I_2$：CPの$I_0 \sim I_2$と同じ．

1.b　ALU Function Control
　$I_3 \sim I_5$, $I_9 \sim I_{10}$：CPの$I_3 \sim I_5$, $I_9 \sim I_{10}$に同じ．

1.c　ALU Destination Control
　$I_5 \sim I_8$, $I_{11} \sim I_{12}$：CPの$I_6 \sim I_9$, $I_{12} \sim I_{13}$に同じ．

1.d　Carry Flag Control
　$I_{13} \sim I_{14}$：CPの$I_{15} \sim I_{16}$に同じ．

2.オペランドコントロール

2.a　Register Control
　$I_{15} \sim I_{22}$：CPの$I_{18} \sim I_{21}$, $I_{23} \sim I_{26}$に同じ．

I/O & RAMパラメータ・コントロール

〔表B-2〕 I-Select Control

Nano-Mnemonic	I_{23}	I_{24}	Effective Address
XI	0	0	I-Port Address
PI	1	0	Parameter+I-Port Address
XR	0	1	Read Memory Address
PR	1	1	Parameter+Read Memory Address

Input Port Parameter Reg. とRAM Read Parameter Reg. による修飾を制御。

〔表B-3〕 O-Select Control

Nano-Mnemonic	I_{25}	I_{26}	Effective Address
XO	0	0	O-Port Address
PO	1	0	Parameter+O-Port Adderss
XW	0	1	Write Memory Address
PW	1	1	Parameter+Write Memory Address

Output Port Parameter Reg. とRAM Write Parameter Reg. による修飾を制御

〔表B-5〕 Output Port

Nano-Mnemonic	I_{33}	I_{34}	I_{35}	I_{36}	I_{37}	I_{38}	Address (8)	Output Port
SWOT	0	0	0	0	0	1	1	Swapping Register
RPOT	0	0	0	0	1	0	2	RAM Parameter Register
ITEN	0	0	0	0	1	1	3	Interrupt Enable Flag Set
CODP	0	0	0	1	0	0	4	Console Disp
OFRO	0	0	1	0	0	0	10	Offset Register
OPRG	0	0	1	0	0	1	11	O-Parameter Register
NLCO	0	0	1	0	1	0	12	Nano Loop Counter
ITDS	0	0	1	0	1	1	13	Interrupt Disable Flag Set
SRQO	0	1	0	0	0	0	20	Sync Request Out
SAKO	0	1	0	0	0	1	21	Sync Ack Out
SYOT	0	1	0	0	1	0	22	Sync Out
EITM	0	1	0	0	1	1	23	External Interrupt Mask
EMAR	0	1	1	0	0	0	30	External Memory Address (Read)
EMAW	0	1	1	0	0	1	31	External Memory Address (Write)
EMWD	0	1	1	0	1	0	32	External Memory Write Data
EMCL	0	1	1	0	1	1	33	External Memory Control
HALT	0	1	1	1	1	1	37	Halt
DTTR	1	0	0	0	0	0	40	Data Transporter Reset
XDS1	1	0	0	0	1	0	42	X'fer Data to MP of Same Group MPB1
XDS2	1	0	0	1	0	0	44	〃 MPB2
XDS12	1	0	0	1	1	0	46	〃 MPB1・2
XDS3	1	0	1	0	0	0	50	〃 MPB3
XDS13	1	0	1	0	1	0	52	〃 MPB1・3
XDS23	1	0	1	1	0	0	54	〃 MPB2・3
XDS123	1	0	1	1	1	0	56	〃 MPB1・2・3
XDCP	1	1	0	0	0	0	60	X'fer Data to Control Processor
XDA0	1	1	0	0	0	1	61	X'fer Data to MP of Another Group MPX0
XDA1	1	1	0	0	1	0	62	〃 MPX1
XDA01	1	1	0	0	1	1	63	〃 MPX0・1
XDA2	1	1	0	1	0	0	64	〃 MPX2
XDA02	1	1	0	1	0	1	65	X'fer Data to MP of Another Group MPX0・2
XDA12	1	1	0	1	1	0	66	〃 MPX1・2
XDA012	1	1	0	1	1	1	67	〃 MPX0・1・2
XDA3	1	1	1	0	0	0	70	〃 MPX3
XDA03	1	1	1	0	0	1	71	〃 MPX0・3
XDA13	1	1	1	0	1	0	72	〃 MPX1・3
XDA013	1	1	1	0	1	1	73	〃 MPX0・1・3
XDA23	1	1	1	1	0	0	74	〃 MPX2・3
XDA023	1	1	1	1	0	1	75	〃 MPX0・2・3
XDA123	1	1	1	1	1	0	76	〃 MPX1・2・3
XDAAL	1	1	1	1	1	1	77	〃 MPX0・1・2・3

〔表B-4〕 Input Port

Nano-Mnemonic	I_{25}	I_{26}	I_{27}	I_{28}	I_{29}	I_{30}	Address (8)	Input Port
SWIN	0	0	0	0	0	1	1	Swap Register
PRIN	0	0	0	0	1	0	2	RAM Parameter Register
IMDT	0	0	0	0	1	1	3	Immediate Data
COSW	0	0	0	1	0	0	4	Console Switch
OFRI	0	0	1	0	0	0	10	Offset Register
IPRG	0	0	1	0	0	1	11	I-Parameter Register
NLCI	0	0	1	0	1	0	12	Nano Loop Counter
SRQI	0	1	0	0	0	0	20	Sync Request In
SAKI	0	1	0	0	0	1	21	Sync Ack In
SYIN	0	1	0	0	1	0	22	Sync In
EICM	0	1	0	0	1	1	23	External Interrupt Mask
DASV	0	1	1	0	0	0	30	Destination Available
RVDT	0	1	1	0	0	1	31	Receive Data
EMRD	0	1	1	0	1	0	32	External Memory Read Data
EMST	0	1	1	0	1	1	33	External Memory Status
IDWD	1	0	0	0	0	0	40	Identification Word
RDS1	1	0	0	0	1	0	42	Receive Data from Micro Processors of Same Group
RDS2	1	0	0	1	0	0	44	
RDS3	1	0	1	0	0	0	50	
RDCP	1	1	0	0	0	0	60	Receive Data from Control Processor
RDA0	1	1	0	0	0	1	61	Receive Data from Micro Processors of Another Group
RDA1	1	1	0	0	1	0	62	
RDA2	1	1	0	1	0	0	64	
RDA3	1	1	1	0	0	0	70	

フローコントロール （スタック・コントロール 139・140 はCPの140・141に同じである）

ブランチ・コンディション・コントロール

〔表B-6〕 Branch Condition

Nano-Mnemonic	サイクル延長の有無	I_{42}	I_{43}	I_{44}	I_{45}	Function
NOB	×	0	0	0	0	Non Branch
BCO	○	0	0	0	1	Branch If Carry =1
BCZ	○	0	0	1	0	Branch If Carry =0
BFZ	○	0	0	1	1	Branch If F=0
BFNZ	○	0	1	0	0	Branch If F≠0
BFN	○	0	1	0	1	Branch If F<0
BFP	○	0	1	1	0	Branch If F>0
UCB	×	0	1	1	1	Unconditional Branch
BTO	×	1	0	0	0	Branch If TEST=1
BTZ	×	1	0	0	1	Branch If TEST=0
BFOD	○	1	0	1	0	Brach If F=ODD
BLCNO	×	1	0	1	1	Branch If Nano Loop Counter Not Overflow
BOVF	○	1	1	0	0	Branch If Overflow Flag=1
BNOVF	○	1	1	0	1	Branch If Overflow Flag=0
BAFO	×	1	1	1	0	Bracch If Available Flag=1
BAFZ	×	1	1	1	1	Branch If Available Flag=0

〔表B-7〕 Offset Control

Nano-Mnemonic	I_{46}	I_{47}	I_{48}	I_{49}	I_{50}	I_{51}	I_{52}	I_{53}	Branch Adrs. / Immediate Data
DA	0	*	*	*	*	*	*	*	Direct Address
PM	1	*	*	*	*	*	*	*	Parameter Modified Address

PMはOFFSET REG. によるWCS Addressの修飾を指定

付　録　C　　μP の記述形式

〈Microprogram〉::={〈Micro Statement〉}

〈Micro Statement〉::=〈Assembler Instruction〉|〈Micro Instruction〉

〈Assembler Instruction〉::=〈Comment Statemnt〉|〈Procedure Definition Statement〉|〈Macro Definition Statement〉|〈End Statement〉

〈Comment Statement〉::=.CMT 〈Comment String〉

〈Procedure Definition Statement〉::=.PRC 〈Procedure Symbol〉

〈Macro Definition Statement〉::=.MCR 〈Macro Symbol〉=〈Control Field〉

〈End Statement〉::=.FIN

〈Procedure Symbol〉::=〈Label〉{,〈Label〉}

〈Micro Instruction〉::=〈Load Select Marker〉〈Program Field〉|〈Program Field〉

〈Load Select Marker〉::=*

〈Program Field〉::=〈Label〉: 〈Normal Field〉|〈Normal Field〉

〈Normal Field〉::=〈Control Field〉|〈Macro Symbol〉'|〈Macro Symbol〉' 〈Control Field〉

〈Control Field〉::=〈Null〉|〈OP Control Field〉; 〈OPR Control Field〉; 〈Flow Control Field〉

〈OP Control Eield〉::=〈Null〉|〈ALU S-Opr.〉, 〈ALU Funct.〉, 〈ALU D-Opr.〉, 〈Carry〉

〈OPR Control Field〉::=〈Null〉|〈A-Port〉, 〈B-Port〉, 〈I-port〉, 〈O-Port〉, 〈Test〉

〈Flow Control Field〉::=〈Null〉|〈Stack〉, 〈Branch Condition〉, 〈Flow Operand〉

〈Flow Operand〉::=〈Branch Address〉|〈Immediate Data〉

〈Branch Address〉::=!〈Branch Symbol〉

〈Immediate Data〉::=#〈Numeric String〉

〈Branch Symbol〉::=〈Label〉±〈Displacement〉|*±〈Displacement〉

以下　略