JAERI-M
4820

オンラインFORTRANのFORMAT処理ルーチン

1972年5月

石黒美佐子・山田孝行・次田友宣

日本原子力研究所
Japan Atomic Energy Research Institute

オンラインFORTRANのFORMAT処理ルーチン

日本原子力研究所東海研究所原子炉工学部
石黒美佐子・山田孝行・次田友宣

（1972年4月24日受理）

オンラインデータ処理のプログラムをFORTRANで記述するにあたり，メーカから提供されたFORTRAN（F230－35 ROS）はオンラインを目的としたものでないために不便な点が多かった。特に入出力の書式変換（FORMAT処理）がおそいのは致命的であった。これに対処するために，独自で主記憶内で実行できる書式変換プログラムを作成した。処理方法は従来の方法と異なり，入力，出力一括方式で，しかもそのプログラム自身をFORTRANで記述する。これによって処理能力は約100倍となった。現在ROS FORTRANとインタフェースをとっているが，大枠は，計算機に依存しないプログラムとして作成する。サブルーチンの主記憶使用容量は約9Kバイトである。

FORMAT Processing Routine for Online FORTRAN

Misako ISHIGURO · Takayuki YAMADA · Tomonobu TSUGITA
Div. of Reactor Engineering, Tokai, JAERI
(Received 24 April 1972)

The F230-35 ROS FORTRAN by a computer manifacture is with many drawbacks. Though ROS FORTRAN is part of the ROS processors (ROS is for Real time Operating System), it is not suitable for the online use. Especially, the FORMAT processing is extremely time-consuming, which is fatal to the online system in JAERI. A faster FORMAT processor has therefore been developped in the JAERI computing center ; it is written in FORTRAN language to reduce its maintainance effort. The processor consists of input and output conversion routines, and the method of conversion differs from the existing ones, The speed of this processor is about 100 times as fast as the manufacture's one. The processor is resident in core and occupies 9 K bytes. The processor is being used at present, linked to ROS FORTRAN, For application in other computer systems, its modification in the interface is necessary.

目　　次

1. FORTRAN IOサブルーチン作成の主旨

1.1 ROS FORTRANについて

F230−35 ROS (Real Time Operating System) に対して，JIS 3000 レベルのFORTRANがメーカーから提供されている。

F230−35 ROSで今まで行なわれていたオンライン業務は，銀行業務など比較的単純なものが多く，ユーザ側で作成するプログラムはアセンブラで書かれるのが常であつた。メーカ側でROS FORTRANは，オンライン業務のバックグランドジョブを記述するための道具という意図で作成され，それはバッチ的色採の濃いものである。

原研においては，オンラインで科学技術計算を行なうという特種性から，ユーザプログラムのうちデータ解析の部分はどうしてもFORTRANで書かざるを得ない。ROS FORTRANを使用してみて特に書式変換で，処理速度とエラー処理に非常に不便を感じ，やむをえず原研側でFORTRAN IOに手を加えることになつた。

1.2 書式変換サブルーチンの主記憶常駐化

F230−35の主記憶容量は，cup処理時間に比して小さいということと相まつて，ROS FORTRAN IOサブルーチンは実行時の処理能力を上げるよりはむしろ主記憶の節約に重点をおいて作成されている。そのような背景から，FORTRAN IOサブルーチンは，実行時にはすべてドラム上に置かれ，必要に応じて，小きざみなオーバレイにより，ドラムから主記憶にプログラムを転送する方式をとつている。たとえば1行分の書式変換をするのに，10〜20回ドラムから主記憶へとプログラムの入れ換えがなされる。ドラムとの1回のアクセス時間を20m秒とすれば1,000行に対して約5分必要となる。

現在，原研システムにおいてデータ処理プログラムとしては，F230−60で完成しているBOB70を書換えたものが整備完了している。入力電文の書式変換，ラインプリンタ出力のための書式変換，送信電文のための書式変換をあわせると，書式変換だけで5分を要し，計算を含め6分強となつている。原研のオンラインシステムは，送信，受信，ラインプリンタ出力をディスクパック上のファイルを経由して行なうので，このようにデータ処理のプログラムに長時間要したのでは，ファイル領域がいつまでも解放されず，後続する電文の受信ができなくなつてしまうという事態が起こる。また応答時間も，送信受信時間も入れると10分以上になつてしまう。1データ受信するのに3分程度とし，5回線中の平均の回線使用率を1回線とすれば，全処理が2分程度で終わらないと，タスク間の処理時間のバランスがとれなくなり，オンラインシステムのパフォマンスが悪くなる。時間のネックをなくすために書式変換サブルーチンを主記憶常駐にすることがどうしても必要である。

1.3 入出力エラー処理対策

ROS FORTRANの書式変換におけるもう1つの問題点は，入力データエラー（入力電文エラー）やプログラムエラーのために書式変換が行なえない時の処理，つまり入出力エラー処理である。ROS−FORTRANではこの点もバッチ処理的な考えで作成されていて，入出力エ

ラーにかぎらず，プログラム割込によるエラー，基本関数のエラーはすべてタスクを異常終了させるようになつている。このうちプログラム割込はSPIEマクロの機能によりユーザで任意にエラー処理ができる。入出力エラーについてはすべて異常終了させている。基本外部関数のエラーも10回までは許されるがその後は異常終了となる。以上3つのエラーのうち，入出力エラーは非常に起こり易いものである。原研のオンラインシステムでは，ユーザプログラムはいくつかの処理タスクから構成されており，各タスクは，前者が後者を起動するというように一定の順序によつて実行される　。途中であるタスクが何らかの理由により異常終了すると一連のタスクの流れを乱し，ユーザクアイル領域の管理も乱れ，結局システムダウンせざるを得ない。入出力エラー処理をオンラインシステムと適合させて行なうには，書式変換サブルーチンそのものをユーザで作成するしかない。

1.4　書式変換サブルーチン作成方針

書式変換サブルーチン作成にあたり考慮したことを列記すると次のとおりである。

(1)　将来のことも考えて，なるべく計算機に依存しないものにする。そのためにはFORTRANでプログラムを作成する。

(2)　ROS－FORTRANには変更を加えないで，IOサブルーチンの作成にとどめる。従がつて仕様は全くROS－FORTRANと同じとする。

(3)　メーカはFORTRANコンパイラに関するソフトウエアの詳細をユーザに公開しないのが常なので，原研側で得られる情報にもとづいて作成できるものとする。

(4)　主記憶常駐のサブルーチンパッケージとする。ユーディングする際は主記憶の節約には気を配ばる。

(5)　従来は，X，H，A，I，F，E，D記述子に対し，それぞれしかも入出力別々にX変換，H変換，A変換，I変換，F変換，E変換，D変換に区別して処理されていた（F230－60 FORTRANのROS　FORTRAN等）。今回は，全体として次4つに分け

　　X，H変換

　　A変換

　　入力時I，F，E，D変換（数値変換）

　　出力時I，F，E，D変換（数値変換）

一括方式で処理する。これによつて今までだぶつてなされていた処理を統一した手法で取扱えるようにする。

(6)　リアルジョブにおいてはFORTRANでSYSIN，SYSOUTは禁じられているので，ENCODE／DECODEしか利用しないことを考えて，まずはENCODE／DECODEのみに原研作成IOサブルーチンを適用する。しかしながらREAD／WRITE時の書式変換への適用は将来必要となれば含める予定である。

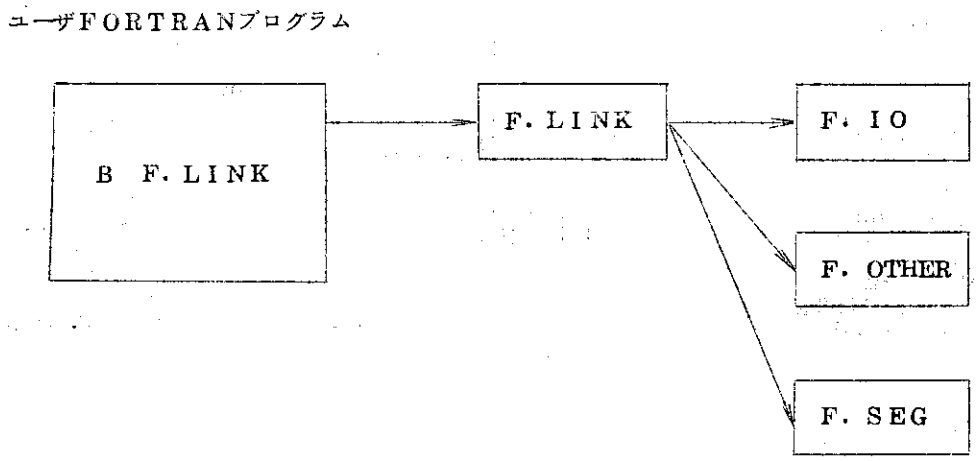
2. ROS FORTRANの入出力インタフェース

2.1 入出力インターフエースについての情報の入手について

FORTRANの書式変換プログラム作成にあたつては，ROS FORTRANのコンパイラに手を加えないことを前提に，IOサブルーチンのうち書式変換のみを新規に作成することにする。IOサブルーチンの大部分は書式変換サブルーチンであり，この他にSYSINとSYSOUT FILEへのアクセスがある。多重で働らくリアルタイムのFORTRANタスクにおいては，SYSIN, SYSOUTを実行時に行なうことを禁止されているため，原研システムでは書式変換（ENCODE/DECODE）のみしか使用していない。その上 FILEのOPEN, OLOSE, READ, WRITEに関する機能も組込むとなると，インターフエース関係のかなり詳細な資料が必要となり入手が困難なので今回は書式変換サブルーチンのみを原研で作成し，従来のものと結合することにする。我々が作成するサブルーチンは，従がつてROS FORTRANで規定された方式に基づいた構造でなければならない。そのためには，ROS FORTRANのIO処理に関する資料が必要となる。当然メーカから提出されるべきものであるが，メーカの準備不足のために，必要な資料を入手できる見込みがなかなか立たなかつた。そこで，いろんなテストプログラムを流し，実行時のメモリダンプによつて手さぐりで情報の集収につとめた。ほぼ必要な情報が確保できたので，それらに基づいて資料を作成し，原研側でやろうとしている方法も添えて，資料確認のためにメーカに提出した。それを見てメーカもやつと我々の意図を理解してくれて，その後我々の資料を補充した全体的なROS FORTRAN IOに関するインタフエース資料を提供してくれた。このように，メーカのソフトウエアを何らかの形で変更するためには，仕事そのものよりも，メーカから必要情報をいかにして入手するかに非常に労力が要ることを経験した。

2.2 インタフエース概略

ROS FORTRANで出力した相対形式プログラムは，LIEDにより実行形式プログラムとして編集される。このプログラムが実行時に使用するIOサブルーチンは，すべて下記の如くF. LINK経由でオーバレイ制御されている。



F. IO 入出力を行なう共通ルーチンで，バッチジョブには必ならず組込まれ，リアルジョブでは，主プログラムに

READ/WRITE,

ENCODE/DECODE,

FORMAT 文

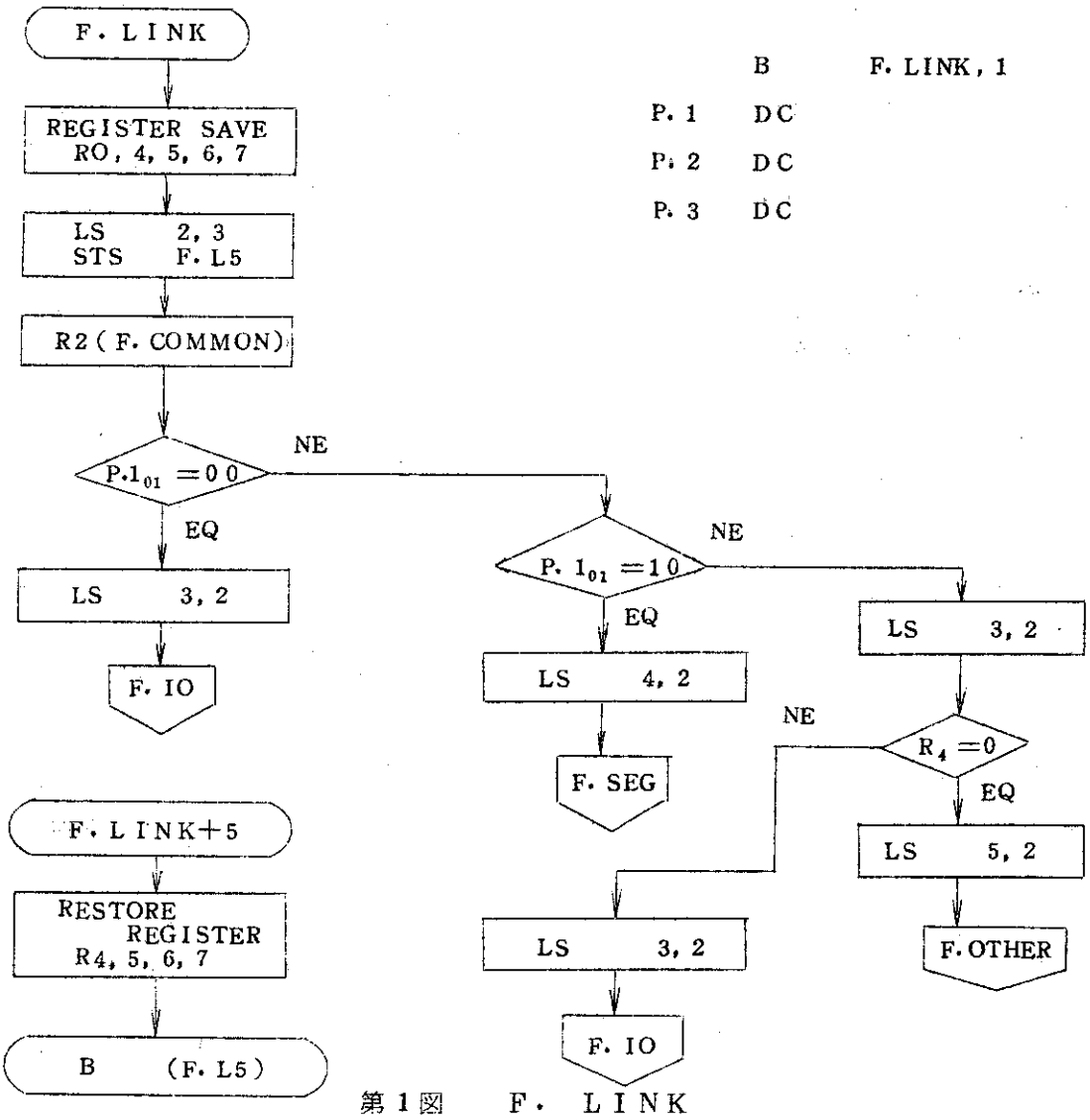
FILES 文

があつた時に組込まれる。

F. OTHERはF. IO が組込まれない時に組込まれる。

F. SEG はセグメント制御を行なうルーチンで，SEGMENT 文があると組込まれる。

F. LINKは，第1図に示すフローで，F. IO, F. OTHER, F. SEG への橋渡しをする。F. LINKはF. COMMONを共通セクションとして含んでいる。F. COMMON は第2図のようなテーブルでF. IO, F. OTHER, F. SEGのENTRY ADDRESSは LIED時に設定され，その他は，第2図の下半分に見えるようなコーディングにより実行時の最初に設定される。



第1図　F. LINK

F. COMMON	0	F. SYSCB ADDRESS
	1	(R_5) ⟷ R_1

F. SYSCB	0	(R_4)
	1	
	2	
	3	F. IO ENTRY ADDRESS
	4	F. SEG ENTRY ADDRESS
	5	F. OTHER ENTRY ADDRESS

第2図　F. COMMON

主プログラムの出だし（バッチの場合）

```
TR        5, 1
LXI       0, 1
ST        F. SYSCB,1, 4
LI        F. SYSCB,1, 4
STD       F. COMMON,1, 4
B         F. LINK,1
F. OPENB
:
:
:
```

2.3　ユーザプログラムとの関係

ユーザプログラムとは，F. LINKを経由してすべて行なわれる。ENCODE文を例にとれば

ENCODE (BNO, F, AREA)　A, B, C, D

F. LINKへは，ENCODEの最初に1回，入出力並びに対しては各1回づつ，ENCODEの最後に1回，上例では合計6回入いる。入出力並びに対しては，変数名，配列要素名に対しては各1回で1語単位で変換される。配列名のときは1回しかF. LINKを経由しないが，変換はDIMENSION により宣言された個数分の配列要素に対して行なわれる。DO型並びは，DOで指定された回数だけF. LINK を経由し，1回のF. LINKに対し，1配列要素づつ変換がなされる。

F. LINKのパラメータは，個数も使い方もそれぞれの場合によって異なるが，一応関係するところを列記すると次のとおりである。

(1)　READ (U, F)／WRITE (U, F) 文の初め

B　F. LINK,1

```
DC    DCFLG
DC    A@U@
DC    A@F@
```

(2) ENCODE (BNO, F, AREA), DECODE文の初め

```
B     F. LINK,1
DC    DCFLG
DC    A@AREA@
DC    A@BNO@
DC    A@F@
```

(3) READ/WRITEおよびENCODE/DECODE文の終り

```
B     F. LINK,1
DC    DCFLG
```

(4) READ/WRITEおよびENCODE/DECODE文の入出力並びに対して，入出力並びが配列名でないとき

```
B     F. LINK,1
DC    DCFLG
DC    A@IOLIST@
```

(5) READ/WRITEおよびENCODE/DECODE文の入出力並びに対して，入出力並びが配列名のとき

```
B     F. LINK,1
DC    DCFLG
DC    A@IOLIST@
DC    L
```

U	Logical Unit No
F	Format No
BNO	文字数
AREA	変換領域
DCFLG	第1表参照
L	配列の大きさ
IOLIST	入出力並び

第1表　入出力文とDCFLGについて

入出力文	Start	End
READ	1000	1080
DECODE	1020	10A0
WRITE	1408	1488
ENCODE	1428	14A8

データの型	READ	DECODE		WRITE	ENCODE
I. TYPE (S)	1040	1060		1448	1468
〃 (D)	1041	1061		1449	1469
R. TYPE (S)	1042	1062		144A	146A
〃 (D)	1043	1063		144B	146B
I. TYPE (S, A)	1044	1064		144C	146C
〃 (D, A)	1045	1065		144D	146D
R. TYPE (S, A)	1046	1066		144E	146E
〃 (D, A)	1047	1067		144F	146F

(S)　単精度　　16進表示

(D)　倍精度

(A)　配列名

その他

F. OPENB　0400

F. STOP　4800

F. SEGIN　8000

2.4　FORMAT文

ROS FORTRANにおいては，FORMAT文はコンパイラでコード化され，第2表のように主記憶内に保存されている。

F230−60のような大型機では，FORMAT文は解読されないでそのまゝの形で保存され，実行読には解読される。しかしながら中小型機では，処理時間が遅い上に主記憶容量が少ないのでコンパイラで解読しコード化する方が有効とされている。

第2表　FORMAT文

nX	28	n			
rIw	38	r	w	00	
rAw	30	r	w	00	
rFw・d	40	r	w	d	16進表示
rEw・d	48	r	w	d	
rDw・d	50	r	w	d	
nHα_1 $\alpha_2 \cdots \alpha_n$ / @α_1 $\alpha_2 \cdots\cdots \alpha_n$@	20	n	α	α ・・・・・・	α_n
n (	10	n			
) （途中）	08	00			（カッコの深さは10以下）
/	18	00			
) （最後）	00	00			
, （区切）	変換されない				

（例）

FORMAT（1H1，3A4，4F5.2，5X，I1）

20	01	F1	40	30	03	04	00			
1H1				3A4						
40	04	05	02	28	05	38	01	01	00	
4F5.2				5X			I1			
00	00									
END)										

2.5　IOサブルーチンからの復帰

IOサブルーチンからユーザプログラムへの復帰は，F．LINKを経由しないで直接ユーザのプログラムへ復帰する。

2.6　FORTRANプログラムのアセンブラシート

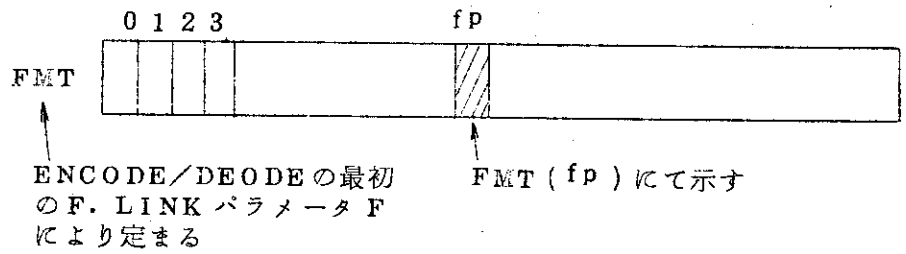
付録1により2.2および2.3で述べたことがわかる。

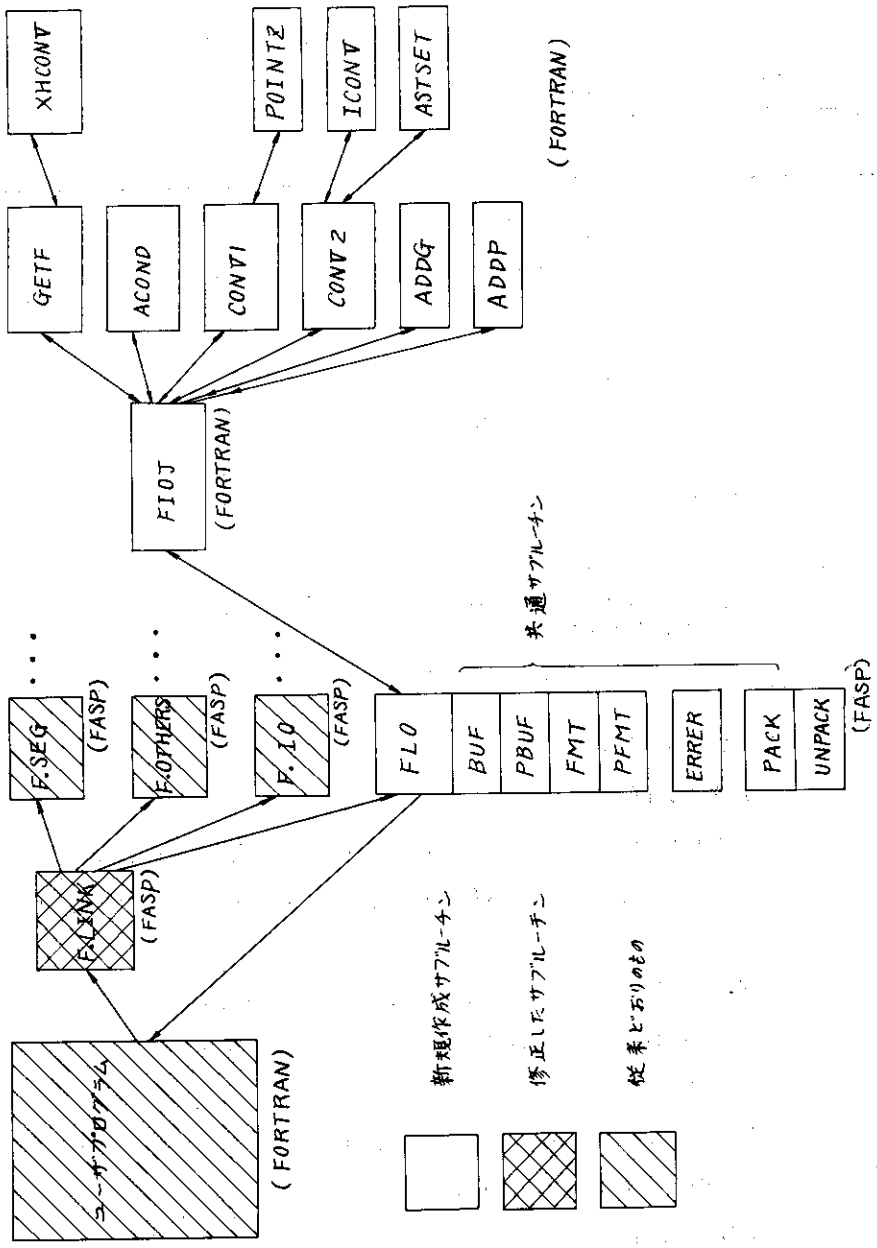
3. 原動作成FORTRAN IOサブルーチンの構造

3.1 サブルーチンの構造

第3図にて示す。

3.2 サブルーチンで使用する記号の説明（共通領域内）

IO	O　入力（DECODE） I　出力（ENCODE）
FMT	FORMATテーブル（FORMATが第2表のようにコード化されているところ）
fp	FORMATテーブルの位置 
BUF	IO Buffer
nc	IO Bufferの位置 BUFとncとの関係は，FMTとfpとの関係に同じ
BM	変換する文字数　ENCODE／DECODEのパラメータBNOより定まる。
A	F. LINKパラメータが入出力並びの場合，IOLISTを指す。 ENCODE／DECODEパラメータIOLISTから定まる。
L	入出力並びが配列名の場合，配列の大きさを示す。
np	FORMATのカッコの深さ。FORMAT（　　）の外側のカッコをnp＝0とし，np＝10まで許す。
FP（np）	FORMATのカッコの位置（左カッコの次を指す）。
FA（np）	FORMATのカッコのくりかえし数 例 FORMAT（1H, A2, 3（I2, 5E12.5,（3A4, 10X）, I8）, 10H‥） np　0　1　2　1　0 FP　FP（1）＝8　FP（2）＝19 FA　FA（1）＝3　FA（2）＝1
F	FORMAT種類 1　I変換 2　F変換


XHCONV
GETF
ACOND
CONV1
POINT2
CONV2
ICONV
ASTSET
ADDG
ADDP
(FORTRAN)
FIOJ
(FORTRAN)
F.SEG
(FASP)
F.OTHERS
(FASP)
F.IO
(FASP)
F.LINK
(FASP)
ユーザプログラム
(FORTRAN)
FLO
BUF
PBUF
FMT
PFMT
ERRER
PACK
UNPACK
(FASP)
共通サブルーチン
新規作成サブルーチン
修正したサブルーチン
従来どおりのもの

	3　E変換
	4　D変換
	5　A変換
	6　H変換
	7　X変換
	0　その他（エラー）
END	FORMATの状態
	0　正常（途中）
	1　FORMAT END
	2　XまたはH変換のみのFORMAT
	3　規定の文字数を越えた（nc＞BM）
D	IOLISTのデータの型
	1　整数（16ビット）
	2　倍精度整数（32ビット）
	3　実数（32ビット）
	4　倍精度実数（64ビット）
n, m, k	FORMATを　nIm　（k＝0）
	nFm. k
	nEm. k
	nDm. k
	nAm　（k＝0）
	nH‥‥　（k＝0, m＝0）
	nX　（k＝0, m＝0）

3.3　サブルーチンの機能

(1)　F. LINK

本来のF. LINKは2.2節で見るようにF. IO, F. SEG, F. OHER　への振分けを行なうためのものである。原研仕様においては，ENCODE／DECODE 関係のものをすべて原研作成の書式変換サブルーチンFIO に渡るように変更された。FASPによるコーディング修正は2語である。

(2)　FIO

FORTRANで書いたFIOJ以下のプログラムとの結合のためのもので，次の3点が主である。

a.　FIOJとの接続。

FORTRAN サブルーチンを呼ぶ形式でFIOJに分岐する。

b.　ユーザプログラムとの接続。

ユーザプログラムからFIOJ に入った時は，復帰アドレスを保存し，ユーザプログラムにもどる時は，そのアドレスによりもどり先を定める。特に復帰アドレスはインデックスレ

ジスタ（この場合 R3）を使用するので，アセンブラによりコーディングする必要がある。

c. F. LINKパラメータの保存

BNO（文字数）　　BNOの値をCOMMON 領域にセットする（BMにて示す）。

DCFLG
IOLIST　}　...　　FIOJ のパラメータとして，設定する。
L

F（FORMAT No）　　ユーザプログラムのFORMATが保存されているアドレスと対応づけ，サブルーチンFMTとPFMTがそのアドレスを使用してFORMATテーブルへのアクセスを行なう。

AREA（変数領域）　　ユーザプログラムのENCODE/DECODE における変数領域（IO Buffer 相当）が保存されているアドレスと対応づけ，サブルーチンBUFとPBUFがそのアドレスを使用し，IO Buffer へのアクセスを行なう。

(3) 共通サブルーチン

FORTANプログラムから呼ばれる形式で，アセンブラで作られたサブルーチンである。FORMATテーブルへのアクセス，IO Bufferへの入出力，FRROR 処理，文字のPACK/UNPACKから成り，FORTRANプログラムから共通に呼ばれる。FORTRANでは書けない部分を補助するためのものである。

a. FORMATテーブルへのアクセス

関数FMT（fp）

FORMATテーブルの第fp番目のバイトの内容を関数値の下方8バイトに入れる。上方は0。

サブルーチンPFMT（fp，A）

Aの下方8バイトがFormat テーブルの第fpバイトに格納される。

b. IO Bufferへの入出力

関数BUF（nc）

IO Buffer の第nc バイトの内容が下方8バイトに入いる。

サブルーチンPBUF（nc，A）

Aの下方8バイトがIO Buffer の第ncバイトに格納される。

c. ERROR処理

サブルーチンERROR（IE，IO）

IEはエラ番号，IOは入力（DECODE）のときは0，出力（ENCODE）のときは1。入力時のエラーは，インデックスレジスタR7＝5とし，FORTRAN インタフエースのERR ルーチンへ，出力時のエラーは呼ばれたところにもどす。出力時はエラーメッセージが出ないが，桁あふれ，バッファover 等は該当欄に＊＊＊印が入るのでエラーがあったことはわかる。IOエラーの一覧表は第6章に載せる。

d. 文字のPACK，UNPACK

サブルーチンPACK（M，N，A）

Aの下方8バイト（Aは1語　16ビット）を，Mの第Nバイトに格納する。Mのデータの型によりNの値の範囲は次表のとおりである。Mの他のバイトはもとのままである。

M	Nのとり得る値
整数（16ビット）	1，2
倍精度整数（32ビット）	1，2，3，4
実数（32ビット）	1，2，3，4
倍精度実数（64ビット）	1，2，3，4，5，6，7，8

サブルーチンUNPACK（M，N，A）

Aの下方8バイトにMの第Nバイトの内容を入れる。Aの上方は0である。A，M，Nのデータの関係はPACKと同じ。

(4)　FIOJ

CALL　FIOJ（DCFLG，A，L）でFIOから呼ばれる。

DCFLG　第1表に示すもの

A　　IOLIST

L　　入出力並びが配列名のとき，配列の大きさを表わす。配列名でないときは1

FIOJはDCFLGによつて処理を次のように振分ける。

a.　ENCODE／DECODEの初め

初期設定

nc　　（IO　Bufferカウンタ）＝0

fp　　（FORMATテーブルカウンタ）＝0

END　　（FORMATの状態）＝0

np　　（FORMATのカッコの深さ）＝0

FP（1）　（FORMATで右カッコに出合つた時にもどる位置）＝0

IO＝0 or 1　（DECODEのとき0，ENCODEのとき1）

GETF により最初の変換を伴つたFORMAT（A，I，F，E，D）を得る。

FORMATは先取り方式で，常に次のFORMATを準備している状態とする。入出力並びと対応しないFORMAT（X，H）は，GETFの中で処理される。

b.　ENCODE／DECODEの終り

残りのX，H変換を処理するためにもう一度GETFを呼ぶ。出力時（ENCODE）の場合はIO　Buffer の残りに空白をつめる。

c.　他のDCFLG（書式変換を要求するもの）

1回の呼込みに対し，データが配列名のときは配列の大きさLだけの回数の変換を実行する（Iでコントロール，$0 \leq I < L$）。配列名でないときは1回だけ変換を実行する。

FORMAT種類に応じて，次の処理ルーチンにより変換する。

A変換（入力，出力）　　ACONV（S）

I，F，E，D変換（入力）　　CONV1（S）

I，F，E，D変換（出力）　CONV2（S）

ここでSは倍精度実数型（8バイト）のパラメータで，FIOJ のパラメータAの型に応じて

整数	Sの1，2バイト
倍精度整数 実数	Sの1，2，3，4バイト
倍精度実数	Sの1，2，3，4，5，6，7，8バイト

が対応し，入力時には変換後のSの値がAにセットされる。出力時には，Aの値をSに上記のようにセットし，変換サブルーチンを呼ぶ。Aの内容をSに移すのにサブルーチンADDGが呼ばれ，Sの内容をAに移すのにサブルーチンADDPが呼ばれる。

(5) GETF

FORMATテーブルから1記述子（X，Hは除く）を取り出すことを目的とする。取り出す記述子は次のいずれかである。

nAm，nIm，nFm·k，nEn·k，nDm·k

ここで，n，m，kは整数で次に示す範囲の値である。

$1 \leqq n, m \leqq 255, \quad 0 \leqq k \leqq 255$

X，H変換はGETFサブルーチン内から，サブルーチンXHCONVを呼ぶことにより行なう。X，H変換が続く時は，X，H以外の記述子に出会うまでGETFの中でXHCONV を呼び続ける。もし，XとH交換から成るFORMATのときは，最後の石カッコが来た時END＝2とおき，呼ばれた場所にもどる。

処理は次の場合に分けて行なわれる。

a. nX

入力の場合は，IO Buffer のカウンタをnバイト進める（nc＝nc＋n）。

出力の場合は，IO Bufferにnバイト空白をつめる。

XHCONVを呼ぶことにより行なう。

b. nH

入力の場合は，IO Bufferの内容をnバイトFORMAT テーブルに移す。出力の場合は逆である。XHCONVによる。a，bいづれももう一度次のFORMATを探す工程に入いる。

c. n（

カッコの深さnp＝np＋1とする。

カッコの位置の保存（FP（np）の設定）

カッコのくりかえし数の保存FA（np）＝n

d. ）　最後

END＝1とおく。fp＝FP（1）とする。（FORMATのくりかえしは，外から2番目の左カッコより行なう。FORMATの中にカッコがないときは，FORMATの始めにもどる）

e. ）　途中

FA（np）＝1のとき，対応するカッコのくりかえしを終える。np＝np－1 としカッコの深さを減らす。

FA(np)＞1のとき，再びカッコ内をくりかえす。

FA(np)＝FA(np)－1により，くりかえし数から1をひく。

f. ／　エラー扱いとする。

今回はENCODE/DECODEに対する書式変換処理なので，／をFORMATで使用できない。READ/WRITE 処理にこのIOサブルーチンを適用するときは／によりIO動作が入いるようにしなければならない。

g. その他

I，F，E，D，A変換に対応するものである。3.2節の記号表のF，m，n，kをセットし，呼ばれたところにもどす。

(6) ACONV(S)

入出力時のA変換を行なう。

FORMATをAmにより示そう。

入力時

IO BufferからmバイトSへもつてくる。Sは上位バイトよりつまり，残るバイトはブランクである。

出力時

Sから上位のmバイトを取りIO Bufferに入れる。

(7) CONVI(S)

入力時の数値変換(I，F，E，D)を行なう。I，F，E，D 記述子の仕様は，F230－35 ROS FORTRANと全く同じである。またF230－60 FORTRANとも同じである。特記すべきは，F，E，Dの記述子に対し，入力データはFタイプ，Eタイプ，Dタイプいづれも可能であるということである。変換方法は[2]のR. W. Floydのフローチャートの1部であるFORMAT free の入力データ変換の部分からヒントを得て考え出したものである。IO Bufferの第ncバイトからmバイトを所定の記述子すなわちIm，Fm.k，Em.kまたはDm.kに従がつてバイナリデータに変換し，結果を(4)のCで述べた約束に従つて倍精度整数Sに格納することが目的である。

このサブルチンで使用した記号の意味は次のとおりである。

E　指数部分

INIS
- 0　頭の空白のscan中
- 1　ブランクを除く第1番目の文字出現後

POINT
- 0　小数点出現前
- 1　小数点出現後実数化前
- 2　実数化後(POINTZサブルーチンによる)

EXP
- 0　指数部なし
- 1　指数部符号＋
- 2　指数部符号－

SIGN
- 0　符　号＋
- 1　符　号－

ko　　小数点以下の桁数

例をあげて上記を説明し変換方法もあわせて述べる。

nc　　nc+m-1　m=14

IO Buffer	ゐ	ゐ	ゐ	+	1	2	3	.	4	5		E	−	2
INIS	0	0	0	1	1	1	1	1	1	1	1	1	1	1
POINT	0	0	0	0	0	0	0	1	1	1	1	2	2	2
EXP	0	0	0	0	0	0	0	0	0	0	0	1	2	2
SIGN	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ko	0	0	0	0	0	0	0	0	1	2	3			

INIS＝0のときの空白は無視する。

INIS＝1のときの空白は0とする。

ENCDICコードの10進数をバイナリに変換するには，

$$d_1\ d_2 \cdots d_I \cdot d_{I+1} \cdot d_{I+J}\ E + e_1\ e_2$$

のように実数データを例にとれば，

T＝0　とおき

$T = T \times 10 + d_i$　　$1 \leqq i \leqq I + J$　，d_i　バイナリ　を計算する。計算方法は，IOLISTのデータの型により，整数演算，倍精度整数演算，実数演算または倍精度実数による演算によるものとする。この例では前2者ではありえない。小数点が出現すればPOINT＝1とすると同時に小数以下の桁数 ko をかぞえる。今の場合 ko ＝J となる。変換予定文字に達するか，またはE，＋，－に出くわすまでこの操作を続け，その次は小数点調整のためにサブルーチンPOINTZを呼ぶ。サブルーチンPOINTZは，ko ＝0のときは，FORMATのFm・k，Em・k，Dm・kのkの値を小数以下の桁数ときめ，ko ≧1のときは，ko を小数以下の桁数ときめ，それぞれの場合に応じてTを10^kまたは10koで割る。これは，入力FORMATについては，入力データに小数点があれば，FORMATの小数点の指定に優先するというきまりに従っている。POINTZサブルーチンによる小数点調整がすめば，次にEまたはDを含むデータに対しては指数部によるデータの調整を行なう。最後に，SIGN によりデータの符号をきめる。

(7)　CONV2（S）

出力時の数値変換（I，F，E，D）を行なう。I，F，E，D記述子の仕様はROS FORTRAN と全く同じである。F230−60 FORTRAN の場合と大きく異なるのは，桁移動子であるPまたはQ（nP，またはnQとして使用する）が使えないことである。バイナリの倍精度実数データSを10進数のEBCDICに変換し，IO Buffer の第ncバイトから順にmバイト格納するのが目的である。バイナリデータを10進数に変換するのは，サブルーチンICONVによる。I変換ではICONVを1回呼ぶことにより10進の整数値を得，E/D 変換では小数以下の桁を求めるためにICONVをやはり1回だけ使用する。F変換にては，まず整数部分を10進化するために1回使用し，小数部分の数値変換するためにもう

1回ICONVを行なう。ICONV については誤差を考慮したり，桁あふれを避けたり，FORTRAN で行なうにはプログラム上工夫した点が多いので次に記す。

(8) ICONV (S, M_1, Mc, M_3, I_{ERR})

バイナリの実数データSを10進数に変換し，10進文字表現V＝($V_i V_{i-1} \cdots\cdots V_2 V_1$)，残りの小数部分Ro，桁数$M_3$を求める。

（算出方法）

$R_i = |S|$, exp＝log R_i　　exp...整数型

より，桁数M_3が求まる

M_3 ＝exp＋1

このとき，$M_3 > M_1$ ならば桁数オバーフローとなり

I_{ERR}＝1 とおく。

$S_i = R_i / 10^{exp}$　より i 桁目の整数S_iを求め

$V_i = 240 + S_i$　より i 桁目の文字V_iがEBCDICコードで求まる。

$R_{i-1} = R_i - S_i \cdot 10^{exp}$, exp＝exp－1

このようにして exp＜1になるまで繰り返し，V，Ro，M_3，I_{ERR}を求める。

注）

① 実数型の数を整数型に置き換えるときに誤差をともなうから，調整を必要とする。
exp＝$\log_{10}10$のとき exp（整数型）＝1とならない。

② M_1は0でない整数で，外部の欄の幅を示す。
McはSを格納する I/O Bufferの上限のバイト番号

4. プログラムの主記憶使用語数

第3表に示すとおりである。一応IOKバイト以内に納めようという初期の目標に達した。

第3表　原研作成FORTRAN　IOサブルーチンの主記憶使用語数

サブルーチン名	言語その他	使用語数 1語＝2バイト	担当者
FIO	FASPサブルーチン	103	山田
PFMT	〃	26	山田
PBUT	〃	24	山田
BUF	〃	17	山田
FMT	〃	17	山田
FIOJ	FORTRANサブルーチン	620	石黒
GETF	〃	344	石黒
XHCONV	〃	156	石黒
ACONV	〃	180	石黒
ADDG	〃	133	石黒
ADDP	〃	133	石黒
CONV2	〃	1,029	山田
ICONV	〃	271	山田
ASTSET	〃	51	山田
CONV1	〃	1,055	次田
POINTZ	〃	126	次田
PACKST	〃	47	次田
PACK/UNPACK	FASPサブルーチン	104	山田
COMMONデータ	COMMONセクション	107	
合計		4,543語 (＝9,086バイト)	

以上新規作成部分

その他にF・LINK　58語（修正前56語）

5. 原研作成IOサブルーチンとROS FORTRANのIOサブルーチンの処理時間の比較

第5表 処理時間の比較

DECODE（回数）	ENCODE（回数）	ROS FIO（秒）	原研 FIO（秒）	テスト・データ
8	8	74	0（1秒未満）	No. 1
16	16	147	0（〃）	〃
24	24	228	1	〃
40	40	383	2	〃
80	80	770	5	〃
200	200	1,790	12	〃
800	800		47	〃
1,120	1,120		68	〃
1,320	1,320		81	〃
2	2	16	0（1秒未満）	No. 2
4	4	36	〃	〃
6	6	55	〃	〃
10	10	94	1	〃
20	20	191	1	〃
600	600		49	〃
2,000	2,000		176	〃
2	2	16	0（1秒未満）	No. 3
4	4	36	〃	〃
6	6	60	〃	〃
10	10	93	1	〃
200	200		9	〃
2,000	2,000		80	〃
100	1	35	5	No. 4
1	100	41	10	〃
1	1	7	1	〃
	300	23	5	No. 5
	300	40	10	〃
	300	57	16	〃

(1) テストデータNo. 1

FORMAT（E9.2, 3X, F6.4, 5X, F6.3, D17.11, I5）

−960.E−02　5.6898　−0.010　7.12144491D 10　10

```
FORMAT(5I10)
  9999   5678   9012   3456   789
FORMAT(10I6)
  1   2   3   4   5   6   7   8   9   10
FORMAT(2X, E9.2, 2X, F4.1, 5X, F4.0, 2X, F6.0, 5X, I3)
  0.5E-79   4.8   -39.   38970.   100
FORMAT(3X, F5.1, 4X, E9.2, 2X, F7.2, E8.2, 8X, I5)
  -25.3   -3.25E70   300.12   0.05   32000
FORMAT(2X, F5.0, 2X, F4.1, 2X, F7.4, F5.0, 1X, I4)
  4000.   30.0   0.0005   -390.   1970
FORMAT(E12.5, 4X, F5.3, 5X, E10.4, F11.9, 5X, I5)
  -3.14780E-10   9.120   3.0000ED1   0.987654321   -2345
```

(2) テストデータ№2

```
FORMAT(F8.1, 5X, F3.1, E10.2, 3X, E15.3, F8.3)
  10.1   -9.39   -3.801101E10   1.1230   0.01E01
FORMAT(F7.1, 5X, F6.2, E18.6, 5X, F6.3, E8.2)
  -671.   3.2   1.23E2   -10.653E-10   1.6
```

(3) テストデータ№3

```
FORMAT(I5, 3X, I3, 5X, I9, I3, I10)
  399   -96   31321   999   -1234566
FORMAT(3X, I8, I1, 3X, I6, 5X, I2, I5)
  671   0   -32100   22   -591
```

(4) テストデータ№4

```
FORMAT(4A4, 5I5, 2F10.5, E20.10, D20.10)
ABCDEFGHIJKLMNOP   111   222   333   44   5
  -12.3456   99.00025   54123.9   123456.789
```

(5) テストデータ№5

```
FORMAT( QAZWSXEDCRFVTGYHNUJM2583690300123
  =, $.$-2AEDCRFT )
FORMAT(80X)
FORMAT(20A4)
```

第5表から言えるのは，I，F，E，D変換のように数値変換を伴うものについては100～150倍程度速くなる。XやH変換では4倍程度である。その差はROSのFORTRAN IOサブルーチンにおいてドラムからプログラムを主記憶にロードする回数が多いか少ないかにより決まる。

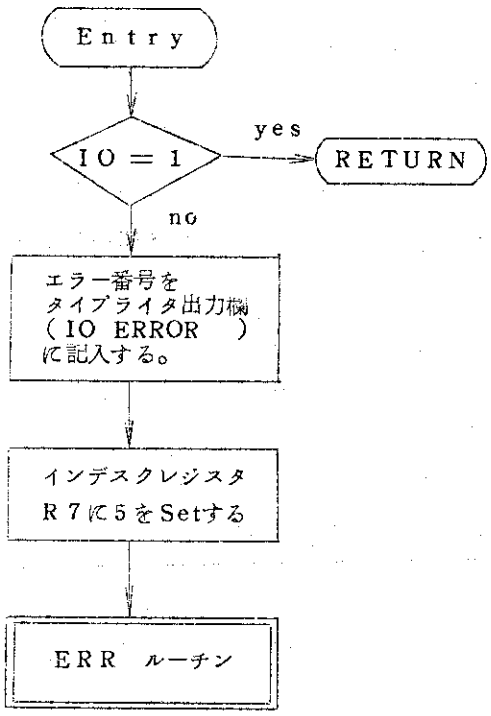
6. エラー処理およびエラーメッセージ

IOサブルーチンにおけるエラーは，第4表のとおりである。本来ならエラーメッセージもラインプリンタに出力すべきであるが，主記憶の節約のためにエラー番号のみを示すことにした。原則として入力時のエラーはエラー番号を出力し，タスクENDとする。出力時のエラーは，＊印をエラー対応欄に書込むことにより，何かのエラーがあることを示しそのままタスクを継続する。以下はオンラインユーザプログラムのエラー処理との関連によるエラー処理方法である。バッチ処理の場合のエラー処理はもっと簡単なものとなる。

エラーが起こると各サブルーチンから，

CALL ERROR(IE, IO)

にて呼ばれる。ここでIEはエラー番号でIOは，IO＝0のときは入力（またはDECODE），IO＝1のときは出力（またはENCODE）である。オンラインのERRORサブルーチン（ERR）はFASPで書かれ，FORTRANインタフェースパッケージの一部に置かれている。処理の内容は次のとおりである。なお出力時は，ASTSETサブルーチンで前もってエラー対応欄に＊印が置かれた後，ERRORサブルーチンが呼ばれる。



ERR ルーチンはオンラインデータ処理プログラム（FORSRANで書かれたもの）のエラー処理ルーチンで，インデックスR7にエラー種別を次のように入れ，

R7＝0	プログラム割込
1	インプットデータが足りない。
2	ラインプリンタ出力行が予定より多すぎる。
3	送信電文が予定より多すぎる。
4	余分なインプットデータがある。

5　　IO　エラー

それぞれ処理される。プログラム割込と，IO エラーについては，エラー種別以外にエラー番号も示される。エラーメッセージは，タイプライタ，送信電文，ラインプリンタに出力される。タイプライタには赤字で出力される。エラーメッセージを出した後はタスクとしては，正常終了という形で終わる。

第4表　エラーメッセージ表

エラー番号		エラー内容
01		変換文字数が予定より多すぎる。
02		X変換とH変換より成るFORMATに対して入出力並び（IOLIST）がある。
03＊		ソフトエラー（F. LINKパラメータの処理エラー）
11＊		ソフトエラー（FORMATテーブル処理のエラー）
12		FORMATの中に／がある。（当初は禁止）
42		A変換で欄の中（Amと書くとするとmを指す）が，対応する入出力並びが含み得る文字数を越えた。
入力時のエラー	51	FORMATと入力並びのデータの型が合わない。
	52	I変換に対して，入力データに指数部がある。
	53	I変換に対して，入力データに小数点がある。
	54	入力データに小数点が2つ以上ある。
	55	数値変換（I，F，E，D）に対し，入力データに数字，小数点，E，D，＋，－，ブランク以下の文字がある。
	56	実数型入力データの指数部分が大きすぎる。（＞76）
	57	実数型入力データに指数部が2ケ所ある。
出力時のエラー	(61)	指定外のFORMATがある。
	(62)	FORMATと出力並びのデータの型が合わない。
	(65)	出力データの値が，指定の欄の巾よりも大きい。

注1. ＊はFORTRAN　IOサブルーチンのエラーである。

2. （　）内の番号は出力されない。出力時のエラーで＊印が出たら，61，62，6,5または01，02のいづれかである。

7. 原研作成IOサブルーチンの成果、その他

現在，原研作成のFORTRAN IO を使用して，開発中のオンラインデータ処理システムを動かしている。FORTRANで書かれているタスクでROSでは6分10秒要したものが，55秒で行なえるようになった。このことにより，原研のオンラインシステムの癌の1つが取り除かれ，タスク間の処理時間のバランスが非常に良くなったことは大きな成果である。一方，主記憶使用量が10kバイト以内に納まったことも，ROSシステムにおいては主記憶に不自由していることを考えると大きな意味がある。F230−60のFORTRAN サブルーチンの主記憶使用量は，R変換やG変換等特殊のものを含まない部分だけで約7,400語（1語=4バイト）を要していることを考えると，我々のものはFORTRANで書いたのにかかわらずかなり小さいものである。またROSシステムのものと比べると，外部仕様は全く同じであるにかかわらず，ROSの方はエラー処理が甘く，エラーを出すべきところもそのまゝ進めたり，また欄の巾に関してはマニュアルにある通りではないらしく，よほど正確にFORMAT，入力データ，入出力並びの対応を取らないと実行できなくなるので，我々のものの方が機能的にも優れている。

仕事の分担は次のとおりで，47年1月に取りかかり3月に完成した。かなり早く完成したのは，FORTRANで書いたことと，F230−60が自由に使えたので，F230−60 によりFORTRANで書いた部分を完全にデッバックしたことによるものと考える。

石　黒
- プログラムの全般的な流れ
- FORMAT処理，各サブルーチンとの連絡部のコーデングおよびディバック
- プログラムテスト

山　田
- ROS FORTRANのインタフェースの調査
- F. LINKの修正
- F. LINKと原研IOサブルーチンとの橋渡し部分のコーディングとディバック
- 出力時の数値変換のコーディング・ディバック
- プログラムテスト
- 処理能力検査

次　田
- 入力時の数値変換のコーディング・ディバック
- プログラムテスト
- 処理能力検査

最後に，ROS FORTRANインタフェースについての情報提供に努力して下さった富士通SE山崎敏憲氏，真田勝吉氏ならびに富士通ソフトウエア技術部の諸氏に感謝します。

参　考　文　献

1. 石黒他，オンラインデータ処理システムデザイン，JAERI—memo 3828
2. R. W. Floyd, An Algorithm for Coding Efficient Arithmetic, Comm. acm, (Jan, 1961).

付録1　FORTRANプログラムのアセンブラシート
付録2　原研作成FORTRAN IO のフローチャート
付録3　原研作成FORTRAN IO のプログラムシート
（但しFORTRANで書いた部分のみ）

付 録 1. FORTRANプログラムのアセンブラシート

```
FACOM 230-25/35  ROS  FORTRAN   (V02-L04)            SOURCE PROGRAM LIST

                 I.S.N.  LABEL    STATEMENT

                    1          DIMENSION BUFA(30)
                    2          READ (5,10) J,K,A,L,B,JJ,KK,AA,LL,BB
                    3       10 FORMAT (2(2I5,F5.1,I5,F10.1))
                    4          ENCODE (120,20,BUFA)J,K,A,L,B,JJ,KK,AA,LL,BB
                    5       20 FORMAT (2(11X,I4,3X,I6,3X,F7.1,5X,I6,5X,F8.1,2X))
                    6          WRITE (6,30) BUFA
                    7       30 FORMAT (10X,30A4)
                    8          STOP
                    9          END
```

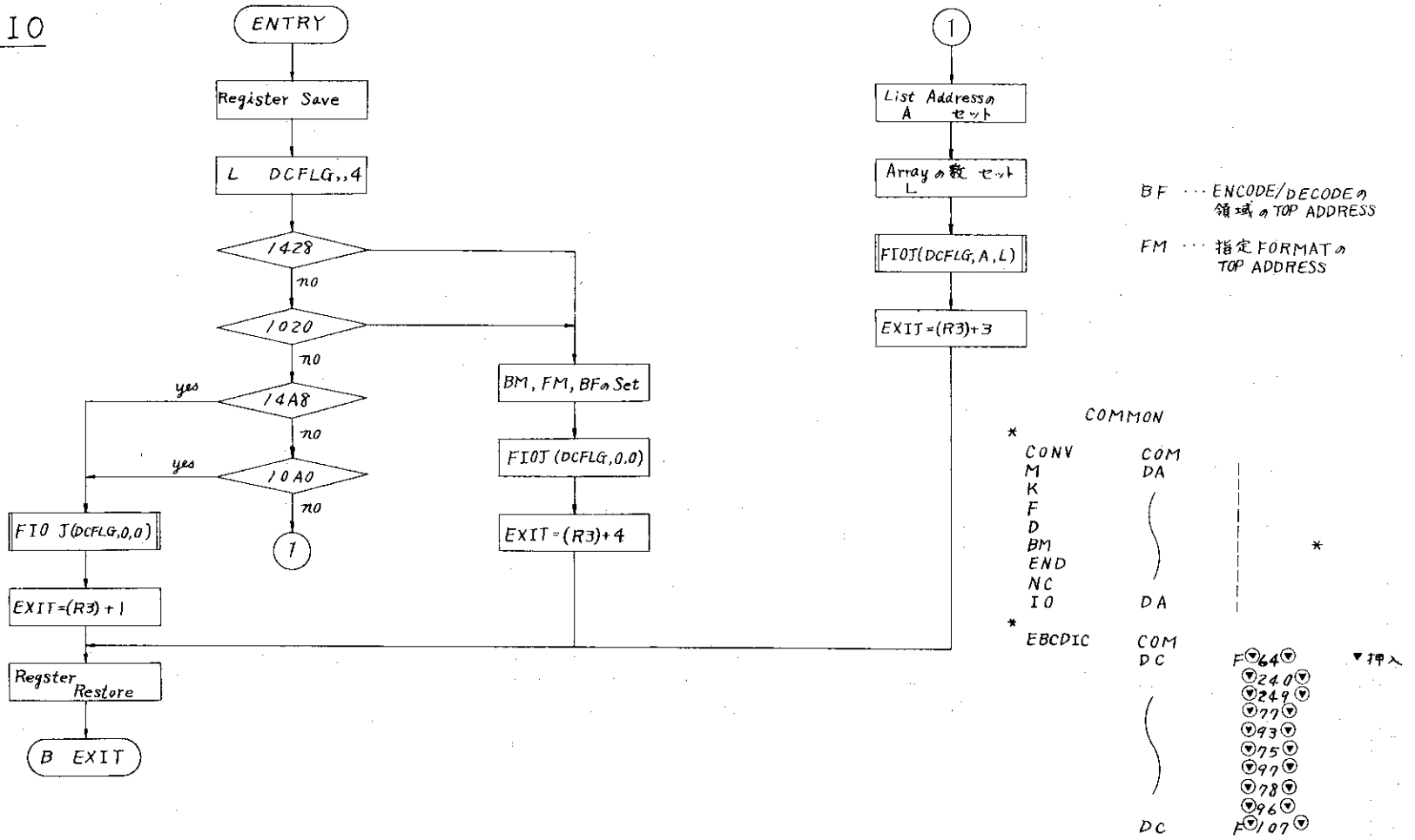
```
FACOM 230-25/35  ROS  FORTRAN   (V02-L04)      OBJECT PROGRAM LIST

      I.S.N.        LABEL        OP.      OPERAND                          LOC.  MCH CODE

                                 TR       5,1                              0000  0C29
                                 LX1      0,1                              0001  5700
                                 ST       F.SYSCB,1,4      F.COMMONの      0002  6B20 00A3
                                 LI       F.SYSCB,1,4        初期設定      0004  7720 00A3
                                 STD      F.COMMON,1,4                     0006  6B60 0000
                                 B        F.LINK,1                         0008  6F78 0000
                                 F.OPENB                                   000A  0400
                                 B        F.LINK,1                         000B  6F78 0000
                                 DC       F'4096'          READ Start      000D  1000
                                 DC       A'C1.3'                          000E  00A2
                                 DC       A'F.10'                          000F  007D
                                 B        F.LINK,1                         0010  6F78 0000
                                 DC       F'4160'          入力並び J      0012  1040
                                 DC       A'J'                             0013  00F4
                                 B        F.LINK,1                         0014  6F78 0000
                                 DC       F'4160'                K         0016  1040
                                 DC       A'K'                             0017  00F5
                                 B        F.LINK,1                         0018  6F78 0000
                                 DC       F'4162'                A         001A  1042
                                 DC       A'A'                             001B  00EC
                                 B        F.LINK,1                         001C  6F78 0000
                                 DC       F'4160'                L         001E  1040
                                 DC       A'L'                             001F  00F6
                                 B        F.LINK,1                         0020  6F78 0000
                                 DC       F'4162'                B         0022  1042
                                 DC       A'B'                             0023  00EE
                                 B        F.LINK,1                         0024  6F78 0000
                                 DC       F'4160'                JJ        0026  1040
                                 DC       A'JJ'                            0027  00F7
                                 B        F.LINK,1                         0028  6F78 0000
                                 DC       F'4160'                KK        002A  1040
                                 DC       A'KK'                            002B  00F8
                                 B        F.LINK,1                         002C  6F78 0000
                                 DC       F'4162'                AA        002E  1042
                                 DC       A'AA'                            002F  00F0
                                 B        F.LINK,1                         0030  6F78 0000
                                 DC       F'4160'                LL        0032  1040
                                 DC       A'LL'                            0033  00F9
                                 B        F.LINK,1                         0034  6F78 0000
                                 DC       F'4162'                BB        0036  1042
                                 DC       A'BB'                            0037  00F2
                                 B        F.LINK,1                         0038  6F78 0000
                                 DC       F'4224'          READ End        003A  1080
                                 B        F.LINK,1                         003B  6F78 0000
                                 DC       F'5160'                          003D  1428
                                 DC       A'BUFA'          ENCODE Start    003E  00B0
                                 DC       A'C1.2'                          003F  00A1
                                 DC       A'F.20'                          0040  0088
                                 B        F.LINK,1                         0041  6F78 0000
                                 DC       F'5224'          入力並び J      0043  1468
                                 DC       A'J'                             0044  00F4
                                 B        F.LINK,1                         0045  6F78 0000
                                 DC       F'5224'                K         0047  1468
                                 DC       A'K'                             0048  00F5
                                 B        F.LINK,1                         0049  6F78 0000
                                 DC       F'5226'                A         004B  146A
                                 DC       A'A'                             004C  00EC
                                 B        F.LINK,1                         004D  6F78 0000
```

```
FACOM 230-25/35  ROS  FORTRAN   (V02-L04)      OBJECT PROGRAM LIST                                    72.03.16   PAGE    2

        I.S.N.        LABEL        OP.        OPERAND                                    LOC.  MCH CODE

                                   DC         F'5224'                          L         004F  1468
                                   DC         A'L'                                       0050  00F6
                                   B          F.LINK.1                                   0051  6F78 0000
                                   DC         F'5226'                          B         0053  146A
                                   DC         A'B'                                       0054  00EE
                                   B          F.LINK.1                                   0055  6F78 0000
                                   DC         F'5224'                          JJ        0057  1468
                                   DC         A'JJ'                                      0058  00F7
                                   B          F.LINK.1                                   0059  6F78 0000
                                   DC         F'5224'                          KK        005B  1468
                                   DC         A'KK'                                      005C  00F8
                                   B          F.LINK.1                                   005D  6F78 0000
                                   DC         F'5226'                          AA        005F  146A
                                   DC         A'AA'                                      0060  00F0
                                   B          F.LINK.1                                   0061  6F78 0000
                                   DC         F'5224'                          LL        0063  1468
                                   DC         A'LL'                                      0064  00F9
                                   B          F.LINK.1                                   0065  6F78 0000
                                   DC         F'5226'                          BB        0067  146A
                                   DC         A'BB'                                      0068  00F2
                                   B          F.LINK.1                                   0069  6F78 0000
                                   DC         F'5288'                          ENCODE End 006B  14A8
          6                        B          F.LINK.1                                   006C  6F78 0000
                                   DC         F'5128'                          WRITE start 006E  1408
                                   DC         A'C1.1'                                    006F  00A0
                                   DC         A'F.30'                                    0070  009B
                                   B          F.LINK.1                                   0071  6F78 0000
                                   DC         F'5198'                          出力領域   0073  14AE
                                   DC         A'BUFA'                          BUFA      0074  00B0
                                   DC         F'30'                                      0075  001E
                                   B          F.LINK.1                                   0076  6F78 0000
                                   DC         F'5256'                          WRITE End 0078  14B8
          8                        B          F.LINK.1                                   0079  6F78 0000
                                   DC         F'18432'                         stop      007B  4800
                                   DC         F'0'                                       007C  0000
                        C1.1       DC         F'6'                                       00A0  0006
                        C1.2       DC         F'120'                                     00A1  0078
                        C1.3       DC         F'5'                                       00A2  0005
```

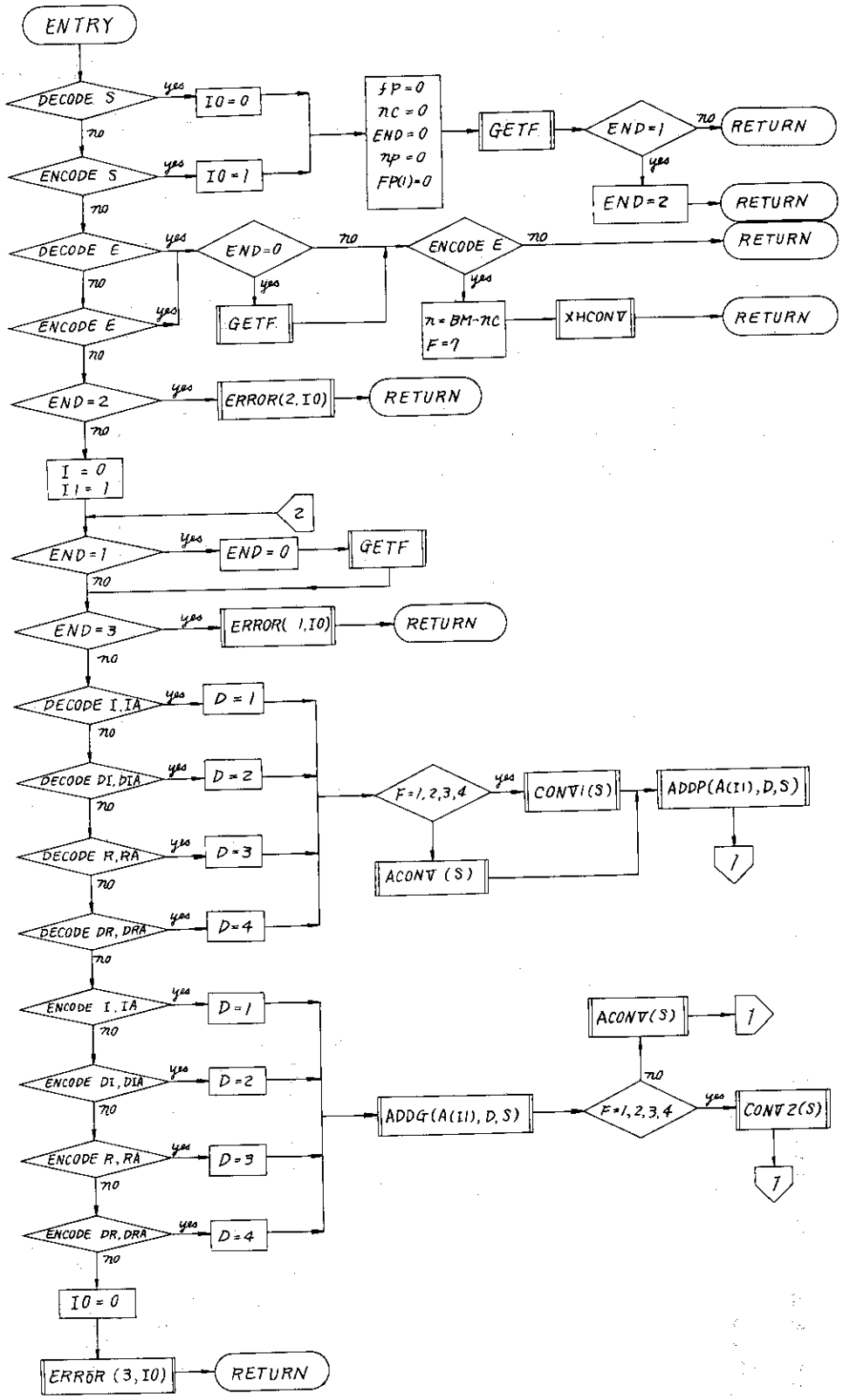
```
FACOM 230-25/35  ROS  FORTRAN   (V02-L04)      M E M O R Y   M A P                                    72.03.16   PAGE    1

  * PROGRAM *   MAIN PROGRAM

  * MODULE NAME *   ENDE

  * FIXED OBJECT FORM *

      ENDE      SECTION     0000-00F9 (  250)

          INSTRUCTIONS              0000-007C (  125)
          FORMATS                   007D-009E (   34)
          CONSTANTS                 009F-00A2 (    4)
          F.SYSCB                   00A3-00A8 (    6)
          DATA                      00A9-00F9 (   81)
      COMMON SECTION

          F.COMMON                  0000-0001 (    2)

  * THE NUMBER OF DICTIONARY ITEMS *                5

  * EXTERNAL NAMES *

      F.IO
      F.LINK

FT   NO ERROR.
```

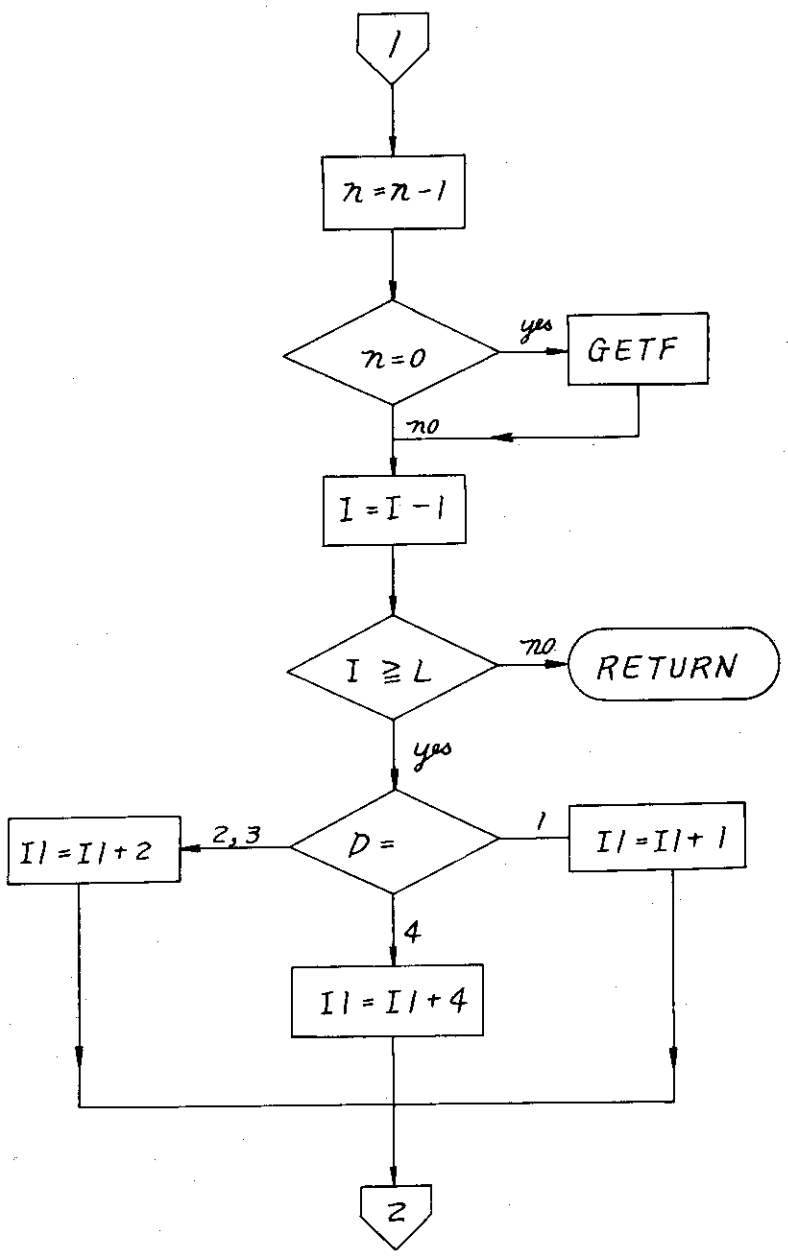
付録 2. 原研作成 FORTRAN IO のフローチャート

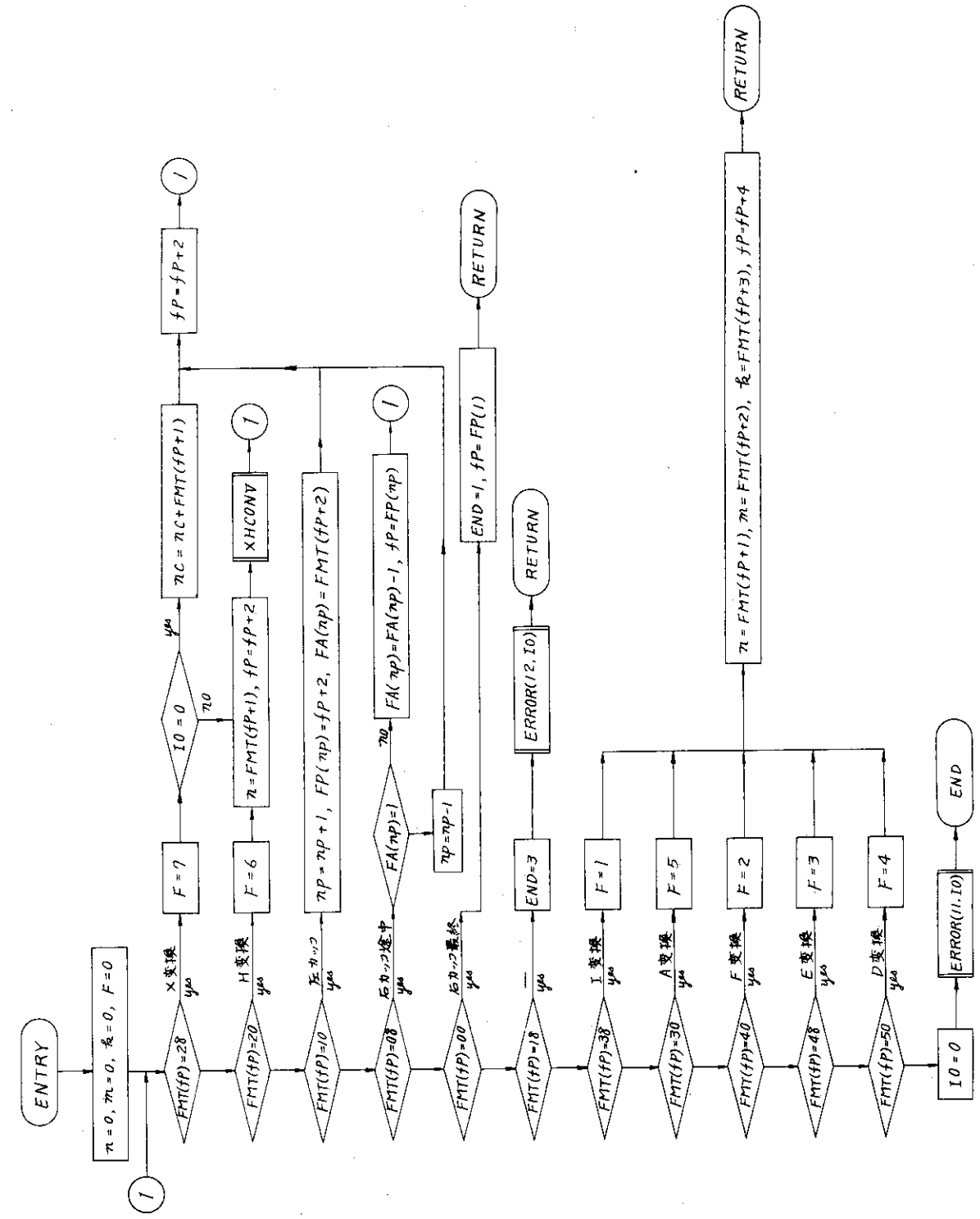
1. FIO



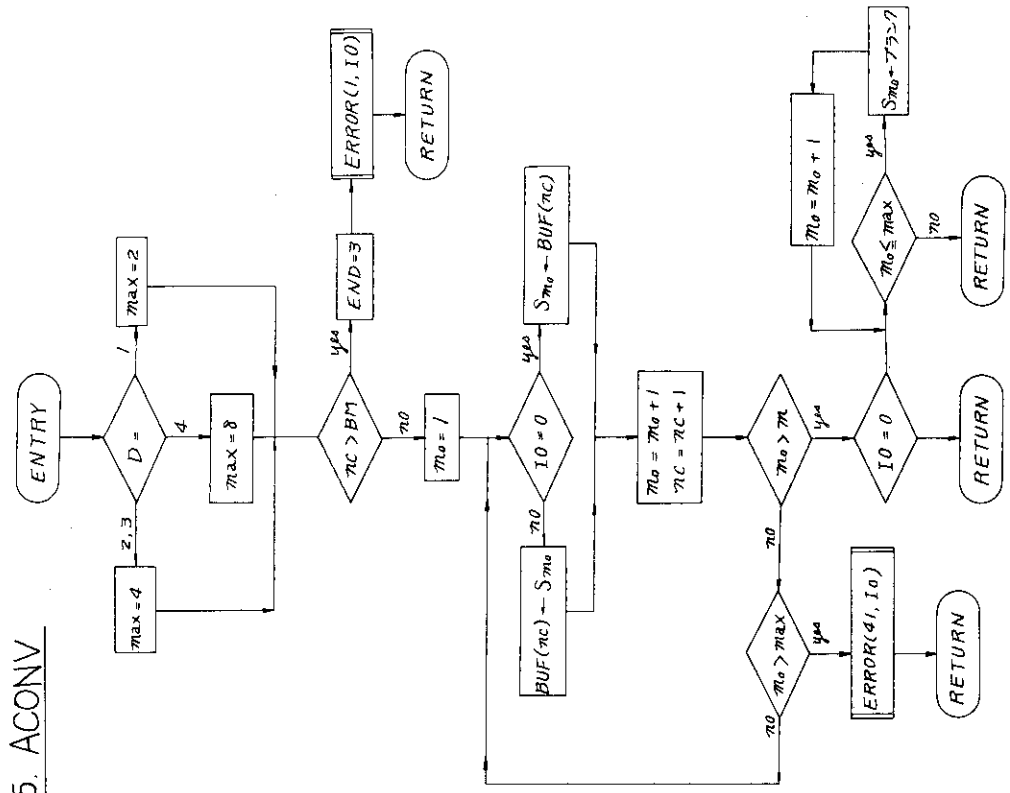
2 FIOJ(DCFLG,A,S)

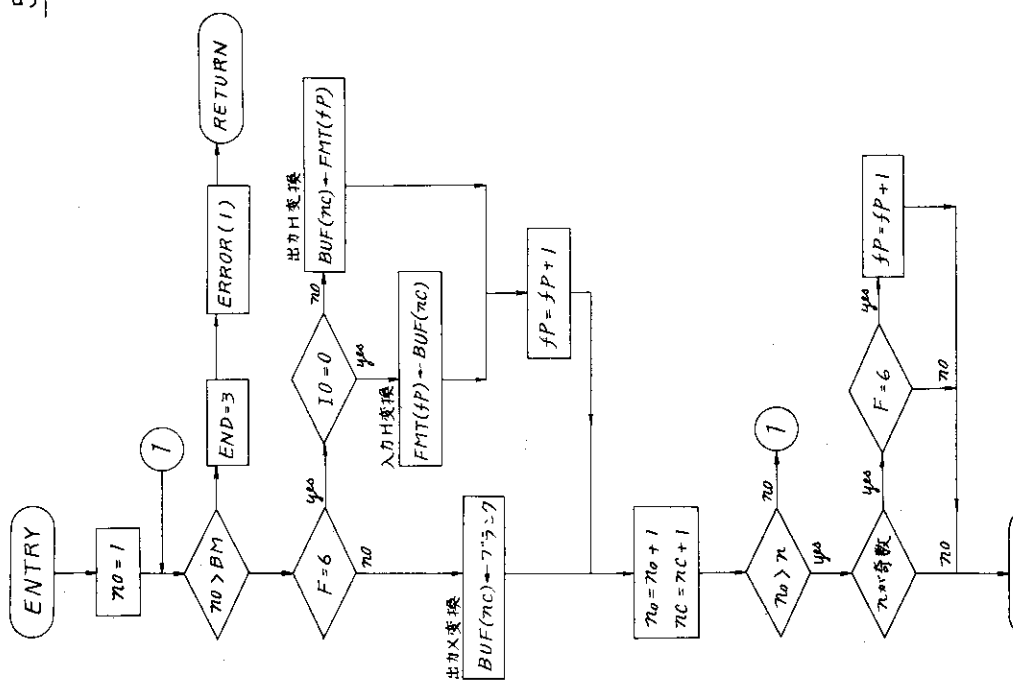



1
n = n - 1
n = 0
yes
GETF
no
I = I - 1
I ≧ L
no
RETURN
yes
D =
2, 3
I1 = I1 + 2
1
I1 = I1 + 1
4
I1 = I1 + 4
2

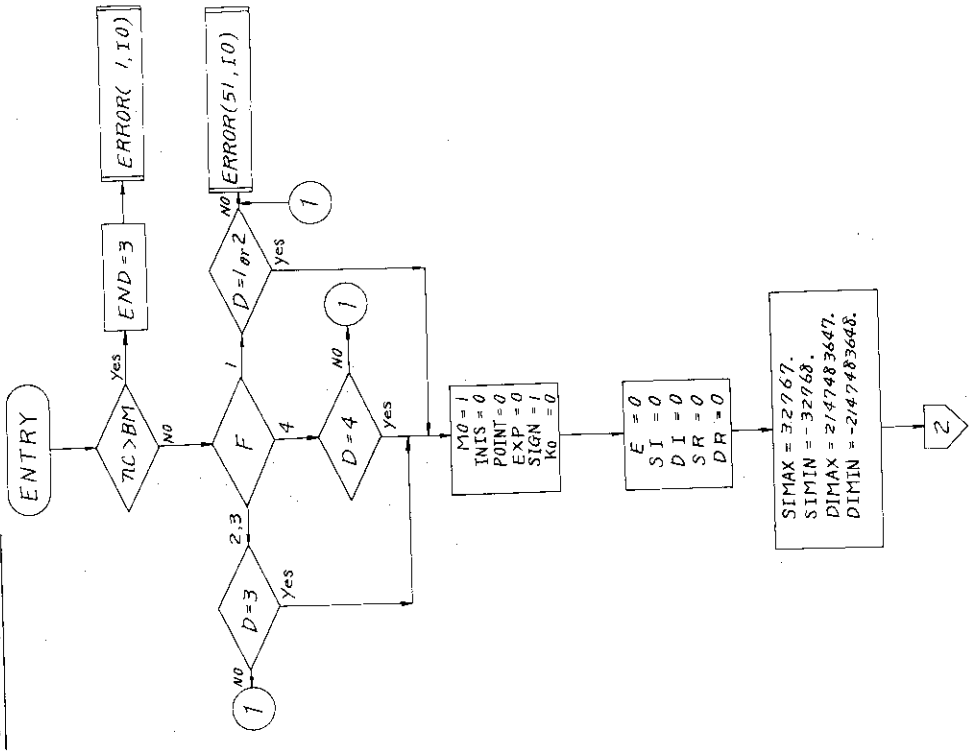

ENTRY
n = 0, m = 0, k = 0, F = 0
FMT(fP) = 28
X変換
F = 7
I0 = 0
nC = nC + FMT(fP+1)
fP = fP + 2
FMT(fP) = 20
H変換
F = 6
n = FMT(fP+1), fP = fP + 2
XHCONV
FMT(fP) = 10
np = np + 1, FP(np) = fP + 2, FA(np) = FMT(fP+2)
FMT(fP) = 08
FA(np) = 1
FA(np) = FA(np) - 1, fP = FP(np)
np = np - 1
FMT(fP) = 00
END = 1, fP = FP(1)
RETURN
FMT(fP) = 18
END = 3
ERROR(12, I0)
RETURN
FMT(fP) = 38
I変換
F = 1
FMT(fP) = 30
A変換
F = 5
FMT(fP) = 40
F変換
F = 2
FMT(fP) = 48
E変換
F = 3
FMT(fP) = 50
D変換
F = 4
n = FMT(fP+1), m = FMT(fP+2), k = FMT(fP+3), fP = fP + 4
RETURN
I0 = 0
ERROR(11, I0)
END

3. GETF

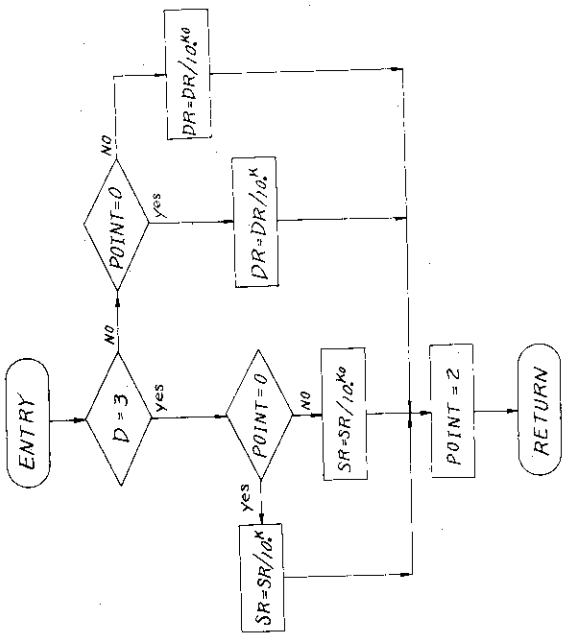

5. ACONV
ENTRY
D =
1
2, 3
4
max = 2
max = 4
max = 8
nc > BM
yes
no
END = 3
ERROR(1, I0)
RETURN
no = 1
I0 = 0
yes
no
Smo ← BUF(nc)
BUF(nc) ← Smo
no = no + 1
nc = nc + 1
no > m
yes
no
no = no + 1
no > max
yes
no
ERROR(41, Io)
RETURN
I0 = 0
yes
no
RETURN
no ≦ max
yes
no
Smo ← ブランク
RETURN


4. XHCONV
ENTRY
no = 1
no > BM
yes
no
END = 3
ERROR(1)
RETURN
1
F = 6
yes
no
I0 = 0
yes
no
出力H変換
BUF(nc) ← FMT(fP)
入力H変換
FMT(fP) ← BUF(nc)
出力X変換
BUF(nc) ← ブランク
fP = fP + 1
no = no + 1
nc = nc + 1
no > n
yes
no
1
F = 6
yes
no
fP = fP + 1
n.*奇数
yes
no
RETURN

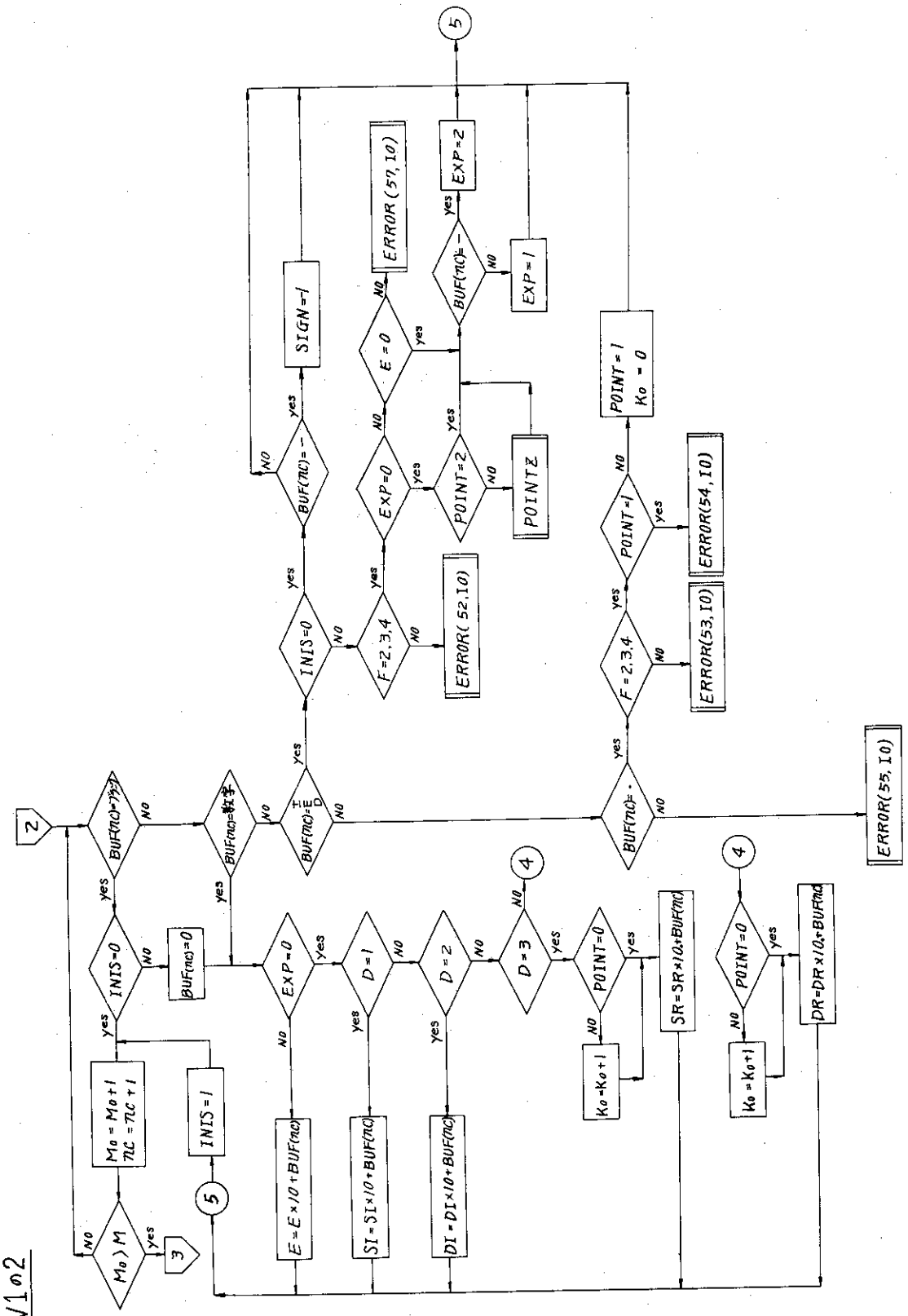
6. CONV1(S)

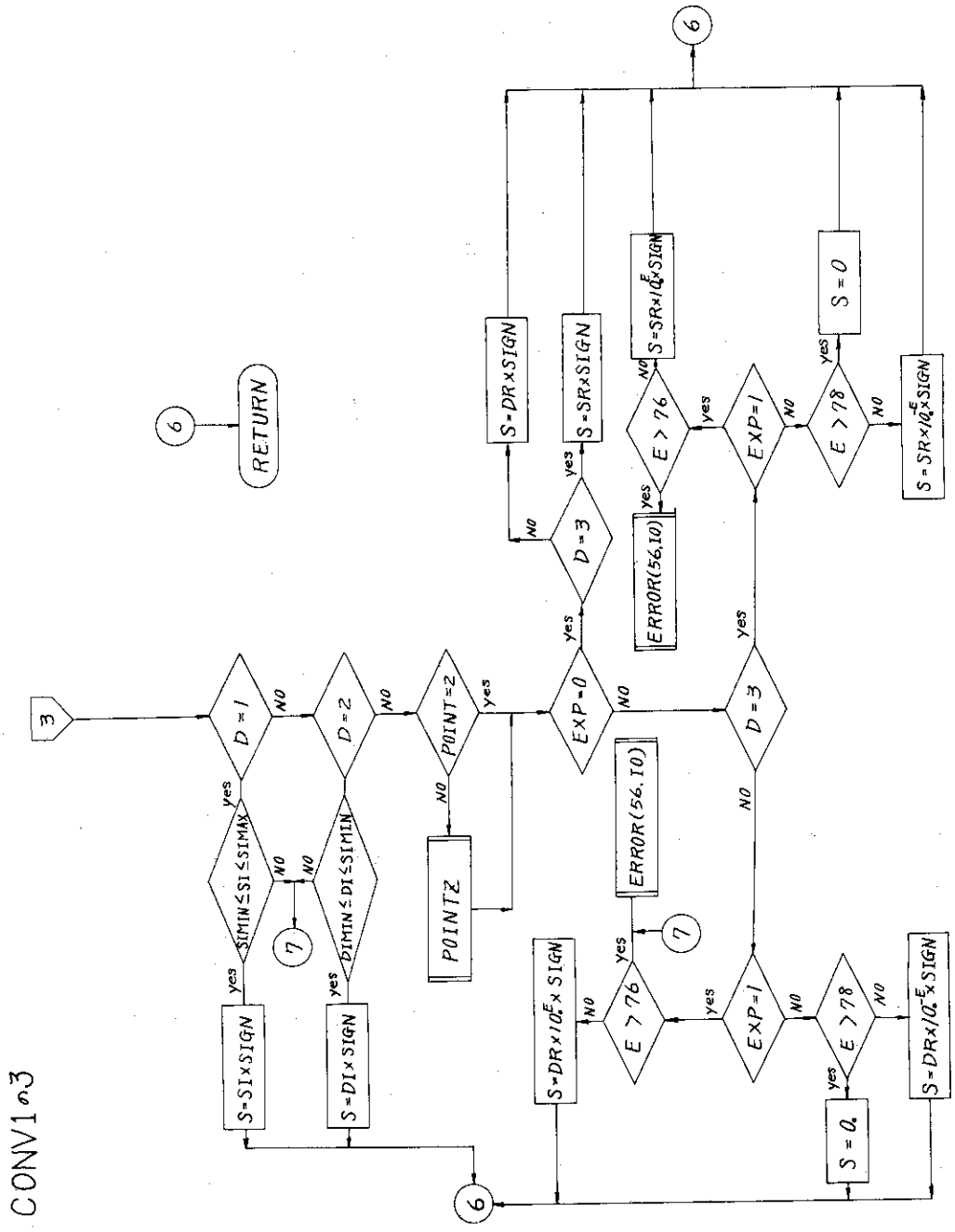


7. POINTZ (Point調整)

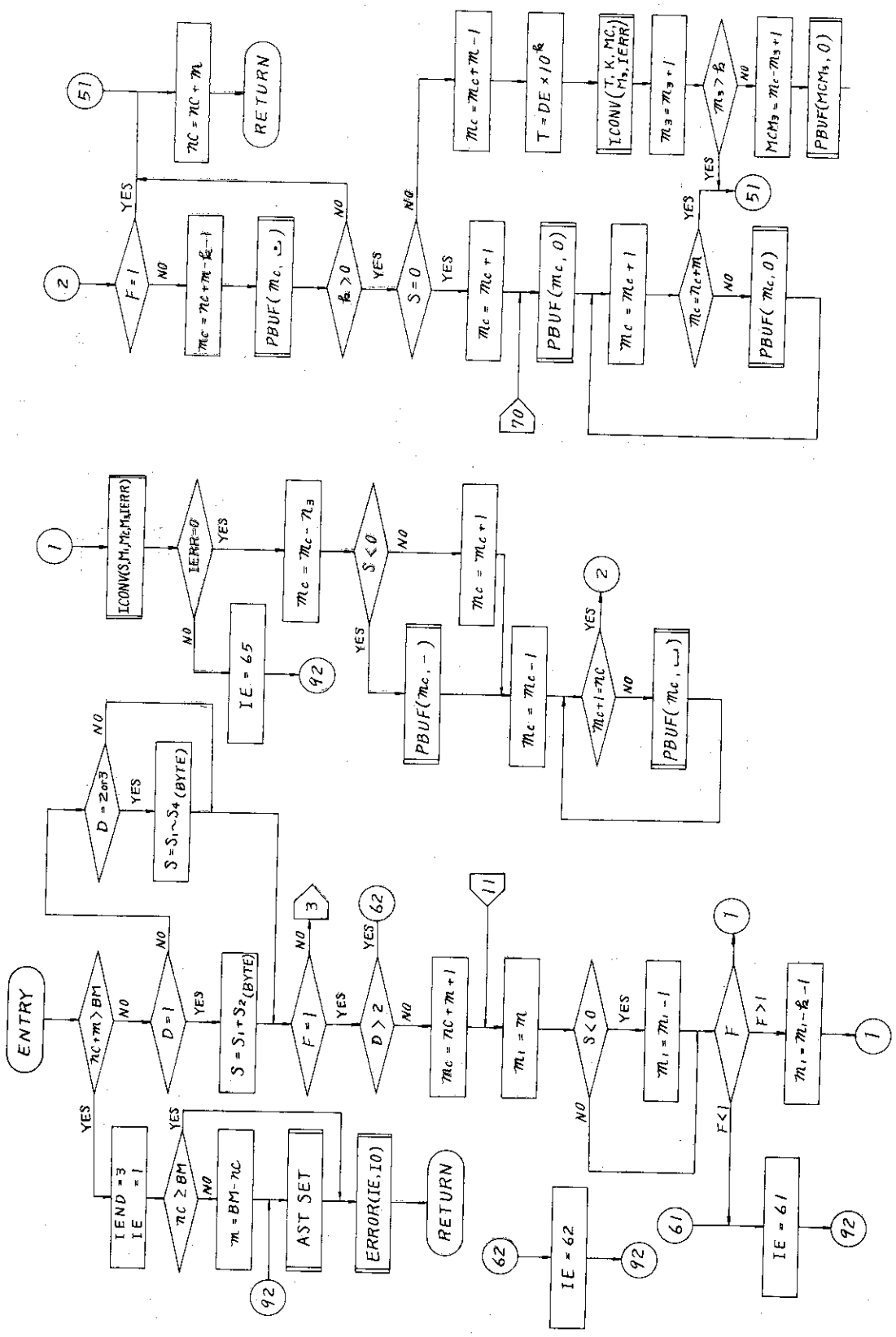


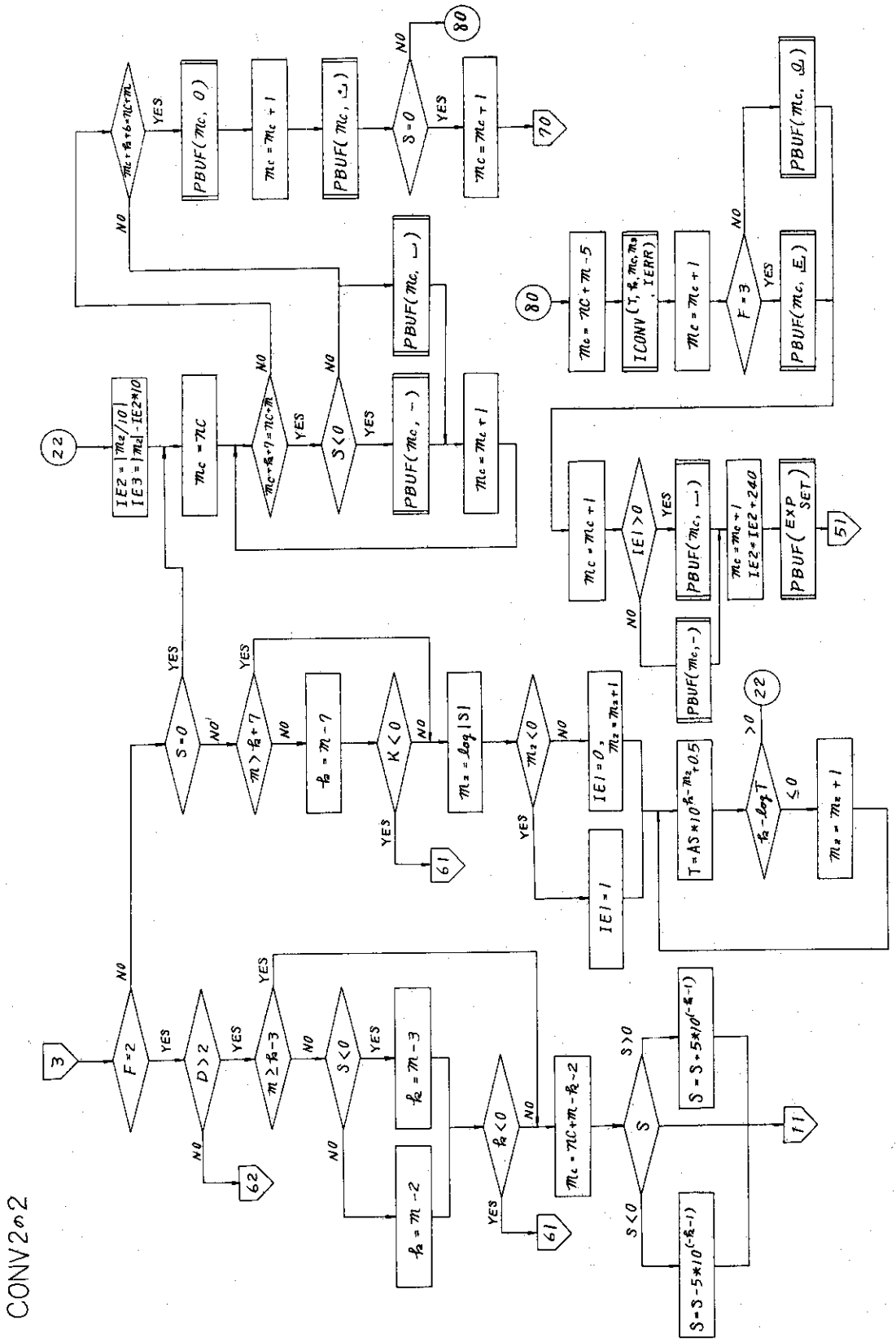
CONV1 n2


2
BUF(nc)=ﾌﾟﾗｽ
BUF(nc)=数字
BUF(nc)=E D
BUF(nc)=.
INIS=0
Mo > M
Mo = Mo+1
nc = nc+1
INIS=1
BUF(nc)=0
EXP=0
D=1
D=2
D=3
POINT=0
Ko=Ko+1
SR=SR×10+BUF(nc)
DR=DR×10+BUF(nc)
E=E×10+BUF(nc)
SI=SI×10+BUF(nc)
DI=DI×10+BUF(nc)
F=2,3,4
ERROR(52,10)
BUF(nc)= -
SIGN=-1
E=0
ERROR(57,10)
POINT=2
POINTZ
BUF(nc)= -
EXP=1
EXP=2
POINT=1
Ko = 0
ERROR(54,10)
ERROR(53,10)
ERROR(55,10)
3
4
5

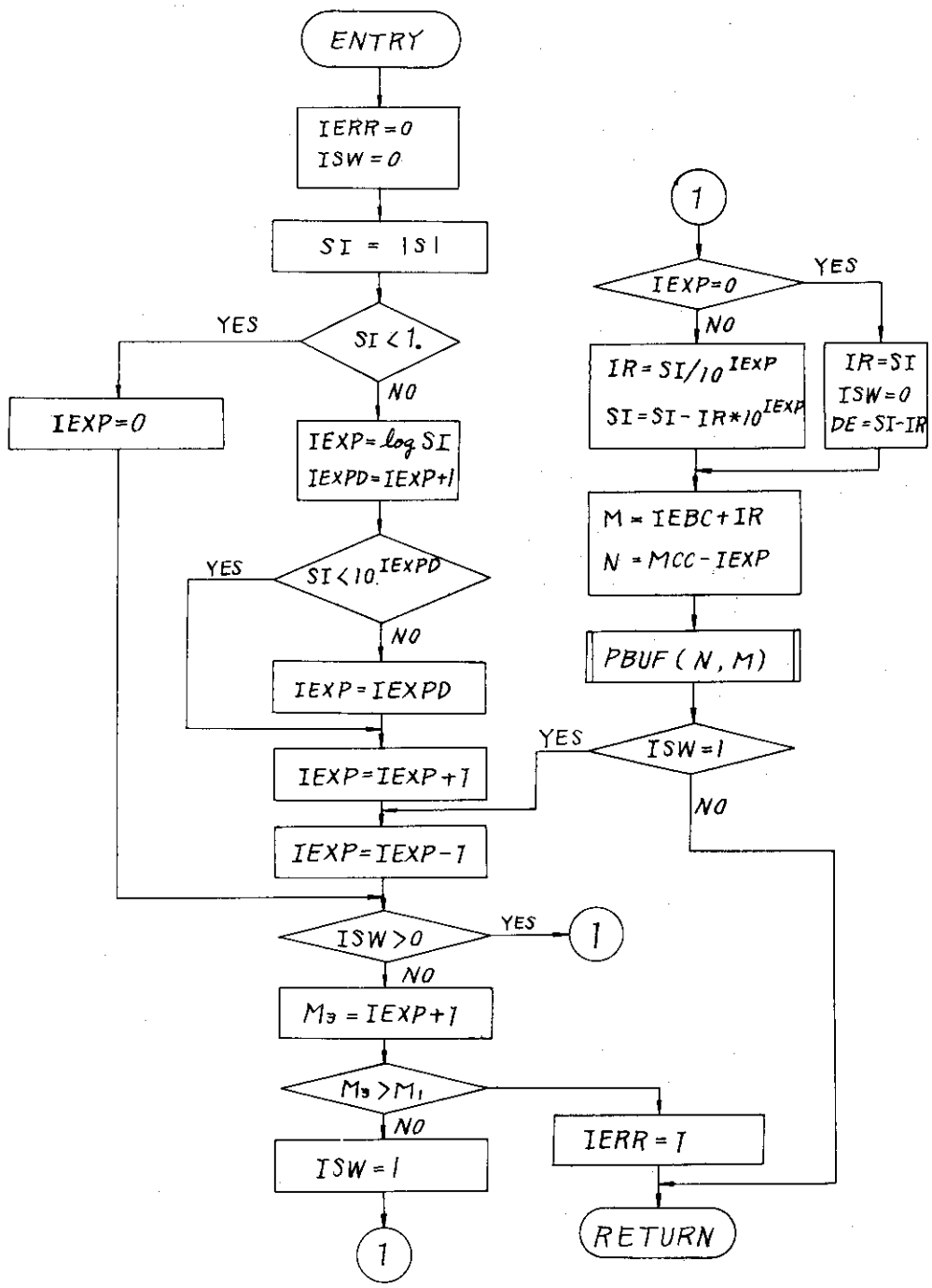

CONV1 n3
3
D = 1
yes
NO
SIMIN ≤ SI ≤ SIMAX
S = SI × SIGN
7
D = 2
DIMIN ≤ DI ≤ SIMIN
S = DI × SIGN
POINT = 2
POINT2
EXP = 0
ERROR(56, 10)
D = 3
S = DR × SIGN
S = SR × SIGN
E > 76
S = DR × 10^E × SIGN
S = SR × 10^E × SIGN
EXP = 1
E > 78
S = 0
S = DR × 10^-E × SIGN
S = SR × 10^-E × SIGN
6
RETURN

8. CONV2(S)


ENTRY
IEND = 3
IE = 1
nc + m > BM
nc ≥ BM
m = BM - nc
AST SET
ERROR(IE,I0)
RETURN
D = 1
D = 2 or 3
S = S1 ~ S4 (BYTE)
S = S1 + S2 (BYTE)
F = 1
D > 2
IE = 62
nc = nc + m + 1
m1 = m
S < 0
m1 = m1 - 1
F
F > 1
m1 = m1 - k - 1
IE = 61
ICONV(S,M1,M2,IERR)
IERR=0
IE = 65
mc = mc - m3
PBUF(mc, -)
mc = mc + 1
mc+1=nc
PBUF(mc, ⌴)
nc = nc + m
mc = mc + m - k - 1
PBUF(mc, ⌴)
k > 0
mc = mc + m - 1
T = DE × 10^k
ICONV(T, K, MC, m3, IERR)
m3 = m3 + 1
m3 > k
MCM3 = mc - m3 + 1
PBUF(MCM3, 0)
mc = mc + m
PBUF(mc, 0)


CONV2o2
3
F=2
D>2
m≥k-3
S<0
k=m-3
k=m-2
k<0
mc=nc+m-k-2
S
S>0
S=S+5*10^(-k-1)
S<0
S=S-5*10^(-k-1)
11
62
61
S=0
m>k+7
k=m-7
K<0
m2=log|S|
m2<0
|E|=0, m2=m2+1
|E|=1
T=AS*10^(k-m2+0.5)
k-logT
≤0
>0
22
mc=m2+1
S=0
mc=nc
mc+k+7=nc+m
S<0
PBUF(mc, -)
mc=mc+1
PBUF(mc, ⊔)
IE2=|m2/10|
IE3=|m2|-IE2*10
mc+k+6=nc+m
PBUF(mc, 0)
mc=mc+1
PBUF(mc, ⊔)
S=0
mc=mc+1
70
80
mc=nc+m-5
ICONV(T, k, mc, m2, IERR)
mc=mc+1
F=3
PBUF(mc, E)
PBUF(mc, Q)
mc=mc+1
|E|>0
PBUF(mc, ⊔)
PBUF(mc, -)
mc=mc+1
IE2=IE2+240
PBUF(EXP SET)
51

9. ICONV(S, M_1, M_c, M_3, IERR)



付 録 3.　原研作成FORTRAN IOのプログラムシート　...6....*....7....*....8

```
C
      SUBROUTINE FIOJ(DCFLG,A,L)
C   M. ISHIGURO
      COMMON/FMTC/FB,FPP,FA,N,NP
      COMMON/CONV/M,K,F,D,BM,IEND,NC,IO
      DIMENSION FPP(10),FA(10)
      DIMENSION A(2)
      DOUBLEPRECISION S
      INTEGER F,D,BM,FP,FPP,FA,DCFLG
      INTEGER H1020,H1428,H10A0,H14A8,H1060,H1064,H1061,H1065,H1062,H1066
     16,H1063,H1067,H146B,H146C,H1469,H146D,H146A,H146E,H146B,H146F
      INTEGER A
      INTEGER BLANK,DMIN,DMAX,LP,RP,PIRI,SLASH,PLUS,MINUS,COMMA
      DATA BLANK/64/,DMIN/240/,DMAX/249/,RP/93/,LP/77/,COMMA/107/,PIRI/
     175/,SLASH/97/,PLUS/78/,MINUS/96/
      DATA H1020/4128/,H1428/5160/,H10A0/4256/,H14A8/5288/,H1060/4192/,
     1H1061/4193/,H1062/4194/,H1063/4195/,H1064/4196/,H1065/4197/,
     2H1066/4198/,H1067/4199/,H1468/5224/,H1469/5225/,H146A/5226/,
     3H146B/5227/,H146C/5228/,H146D/5229/,H146E/5230/,H146F/5231/
C     ENCODE/DECODE START END FLAG TEST
      IF(DCFLG-H1020) 1,1000,1
    1 IF(DCFLG-H1428) 2,2000,2
    2 IF(DCFLG-H10A0) 3,30,3
    3 IF(DCFLG-H14A8) 24,30,24
C     END TEST
   24 IF(IEND-2)4,50,4
    4 I=0
      I1=1
    5 IF(IEND-1)6,60,6
    6 IF(IEND-3)7,70,7
    7 IF(DCFLG-H1060) 8,80,8
    8 IF(DCFLG-H1064) 9,80,9
    9 IF(DCFLG-H1061)10,90,10
   10 IF(DCFLG-H1065)11,90,11
   11 IF(DCFLG-H1062) 12,100,12
   12 IF(DCFLG-H1066) 13,100,13
   13 IF(DCFLG-H1063) 14,110,14
   14 IF(DCFLG-H1067) 15,110,15
   15 IF(DCFLG-H146B) 16,120,16
   16 IF(DCFLG-H146C) 17,120,17
   17 IF(DCFLG-H1469) 18,130,18
   18 IF(DCFLG-H146D) 19,130,19
   19 IF(DCFLG-H146A) 20,140,20
   20 IF(DCFLG-H146E) 21,140,21
   21 IF(DCFLG-H146B) 22,150,22
   22 IF(DCFLG-H146F) 23,150,23
   23 IO=0
      CALL ERROR(03,IO)
C    DECODE START
 1000 IO=0
 1001 FP=0
      NC=0
      IEND=0
      NP=0
      FPP(1)=0


      CALL GETF
      IF(IEND-1)1003,1002,1003
 1002 IEND=2
 1003 RETURN
C     ENCODE START
 2000 IO=1
      GO TO 1001
C     DECODE ENCODE END
   30 IF(IEND)31,32,31
   32 CALL GETF
   31 IF(IO)  33,1003,33
   33 N=BM-NC
      F=7
      CALL XHCONV
      GO TO 1003
C    DECODE
C
C     SINGLE INTEGER
   80 D=1
   81 IF(F-5) 82,83,83
C     I,F,E,D CONVERSION
   82 CALL CONV1(S)
      GO TO 84
C     A CONVERSION
   83 CALL ACONV(S)
C    PUT ADDRESS
   84 CALL ADDP(A(I1),D,S)
  125 N=N-1
      IF(N)  85,86,85
   86 CALL GETF
   85 I=I+1
      GO TO (87,88,88,89),D
   87 I1=I1+1
      GO TO 101
   88 I1=I1+2
      GO TO 101
   89 I1=I1+4
  101 IF(I-L) 5,1003,1003
C     DOUBLE INTEGER
   90 D=2
      GO TO 81
C     REAL
  100 D=3
      GO TO 81
C     DOUBLE PRECISION
  110 D=4
      GO TO 81
C    ENCODE
C     SINGLE INTEGER
  120 D=1
C     GET ADDRESS (A TO S)
  121 CALL ADDG(A(I1),D,S)
      IF(F-5) 122,123,123
C     I,F,E,D CONVERSION
  122 CALL CONV2(S)
```

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

      GO TO 125
C     A-CONVERSION
  123 CALL ACONV(S)
      GO TO 125
C     DOUBLE INTEGER
  130 D=2
      GO TO 121
C     REAL
  140 D=3
      GO TO 121
C     DOUBLE PRECISION
  150 D=4
      GO TO 121
C     END=2
   50 CALL ERROR(02,10)
      RETURN
C     END=1
   60 IEND=0
      CALL GETF
      GO TO 6
C     END=3
   70 CALL ERROR(01,10)
      RETURN
      END
C
      SUBROUTINE GETF
C     GET A FORMAT FROM FORMAT TABLE
C     FORMAT ARE   I-FORMAT    (F=1)       IM
C                  F-FORMAT    (F=2)       FM,K
C                  E-FORMAT    (F=3)       EM,K
C                  D-FORMAT    (F=4)       DM,K
      COMMON/FMTC/FP,FPP,FA,N,NP
      COMMON/CONV/M,K,F,D,BM,IEND,NC,IO
      INTEGER F,D,BM,FP,FPP,FA,DCFLG
      COMMON/EBCDIC/BLANK,DMIN,DMAX,LP,RP,PIRI,SLASH,PLUS,MINUS,COMMA
      DIMENSION FPP(10),FA(10)
C     SPECIAL CHARACTERS
      INTEGER BLANK,DMIN,DMAX,LP,RP,PIRI,SLASH,PLUS,MINUS,COMMA
      INTEGER FMT
      N=0
      M=0
      K=0
      F=0
    1 I1=FMT(FP)
      IF(I1-40) 2,20,2
    2 IF(I1-32) 3,30,3
    3 IF(I1-16) 4,40,4
    4 IF(I1-8) 5,50,5
    5 IF(I1-0) 6,60,6
    6 IF(I1-24) 7,70,7
    7 IF(I1-56) 8,80,8
    8 IF(I1-48) 9,90,9
    9 IF(I1-64) 10,100,10
   10 IF(I1-72) 11,110,11
   11 IF(I1-80) 12,120,12


   12 IO=0
C     SOFT ERROR
      CALL ERROR(11,10)
C     X-FORMAT  (NX)
   20 F=7
      IF(IO) 21,22,21
   22 NC=NC+FMT(FP+1)
   23 FP=FP+2
      GO TO 1
   21 N=FMT(FP+1)
      FP=FP+2
      CALL XCONV
      GO TO 1
C     H-FORMAT  (NH.....)
   30 F=6
      GO TO 21
C     LEFT PARENTHESIS    (N(  )
   40 NP=NP+1
      FPP(NP)=FP+2
      FA(NP)=FMT(FP+1)
      GO TO 23
C     RIGHT PARENTHESIS    (  ) )
   50 IF(FA(NP)-1) 52,52,51
   51 FA(NP)=FA(NP)-1
      FP=FPP(NP)
      GO TO 1
   52 NP=NP-1
      GO TO 23
C     LAST RIGHT PARENTHSIS  (  ) )
   60 IEND=1
      FP=FPP(1)
      RETURN
   70 IEND=3
C     / IS ILLEGAL
      CALL ERROR(12,10)
      RETURN
C     I-FORMAT(NIM)
   80 F=1
   81 N=FMT(FP+1)
      M=FMT(FP+2)
      K=FMT(FP+3)
      FP=FP+4
      RETURN
C     A-FORMAT(NAM)
   90 F=5
      GO TO 81
C     F-FORMAT
  100 F=2
      GO TO 81
C     E-FORMAT
  110 F=3
      GO TO 81
C     D-FORMAT
  120 F=4
      GO TO 81
```

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8
      END
C
C
C
      SUBROUTINE ADDG(A,D,SS)
C     D=1   MOVE A(I)  1-2   TO  S  1-2
C     D=2   MOVE A(I)  1-4   TO  S  1-4
C     D=3   MOVE A(I)  1-4   TO  S  1-4
C     D=4   MOVE A(I)  1-4   TO  S  1-4
C           MOVE A(I+1)  1-4   TO   S  5-8
      DIMENSION SS(2)
      DIMENSION A(2)
      INTEGER D
      J=1
      K=1
  100 I=1
  101 CALL UNPACK(A(J),I,N)
      CALL PACK(SS(K),I,N)
      GO TO (1,2,1,3),I
C     NEXT BYTE
    1 I=I+1
      GO TO 101
C     IF INTEGER
    2 IF(D-1)5,5,1
    5 RETURN
C     IF DOUBLE INTEGER OD REAL
    3 IF(D-2)5,5,6
    6 IF(D-3)5,5,7
    7 IF(K-2)8,5,5
    8 K=2
      J=2
      GO TO 100
      END
C
C
      SUBROUTINE ADDP(A,D,SS)
C     D=1   MOVE S  1-2   TO   A(I)  1-2
C     D=2   MOVE S  1-4   TO   A(I)  1-4
C     D=3   MOVE S  1-4   TO   A(I)  1-4
C     D=4   MOVE S  1-4   TO   A(I)  1-4
C           MOVE S  5-8   TO  A(I+1)  1-4
      DIMENSION SS(2)
      DIMENSION A(2)
      INTEGER D
      J=1
      K=1
  100 I=1
  101 CALL UNPACK(SS(K),I,N)
      CALL PACK(A(J),I,N)
C
      GO TO (1,2,1,3),I
C     NEXT BYTE
    1 I=I+1
      GO TO 101
C     IF INTEGER


    2 IF(D-1)5,5,1
    5 RETURN
C     IF DOUBLE INTEGER OD REAL
    3 IF(D-2)5,5,6
    6 IF(D-3)5,5,7
    7 IF(K-2)8,5,5
    8 K=2
      J=2
      GO TO 100
      END
C
      SUBROUTINE XHCONV
C     X-CONVERSION (OUTPUT)
C     H-CONVERSION (IN/OUT)
      COMMON/FMTC/FP,FPP,FA,N,NP
      COMMON/EBCDIC/BLANK,DMIN,DMAX,LP,RP,PIRI,SLASH,PLUS,MINUS,COMMA
      DIMENSION FPP(10),FA(10)
      COMMON/CONV/M,K,F,D,BM,IEND,NC,IO
      INTEGER BLANK,DMIN,DMAX,LP,RP,PIRI,SLASH,PLUS,MINUS,COMMA
      INTEGER F,D,BM,FP,FPP,FA,DCFLG
      INTEGER BUF,FMT
      NO=1
    1 IF(NC-BM) 3,3,2
    2 IEND=3
C     IO BUFFER OVER
      CALL ERROR(1,IO)
      RETURN
    3 IF(F-6)  5,4,5
C     H-CONVERSION (INPUT)
    4 IF(IO) 6,7,6
C     H-CONVERSION(OUTPUT)
    6 I1=FMT(FP)
      CALL PBUF(NC,I1)
    8 FP=FP+1
    9 NO=NO+1
      NC=NC+1
      IF(NO-N) 1,1,11
C   H-CONVERSION INPUT
   11 IF(F-6) 14,14,12
C     IF N IS ODD
   14 IF(N-(N/2)*2) 12,12,13
   13 FP=FP+1
   12 RETURN
C     X-CONVERSION (OUTPUT)
    5 CALL PBUF(NC,BLANK)
      GO TO 9
C     H-CONVERSION (INPUT)
    7 I1=BUF(NC)
      CALL PFMT(FP,I1)
      GO TO 8
      END
C
C
      SUBROUTINE ACONV(S)
C     A-CONVERSION
```

```
...,*....1....,*....2....*....3....*....4....*....5....*....6....*....7....*....8

      COMMON/FMTC/FP,FPP,FA,N,NP
      COMMON/CONV/M,K,F,D,BM,IEND,NC,IO
      COMMON/EBCDIC/BLANK,DMIN,DMAX,LP,RP,PIRI,SLASH,PLUS,MINUS,COMMA
      DIMENSION FPP(10),FA(10)
      INTEGER BLANK,DMIN,DMAX,LP,RP,PIRI,SLASH,PLUS,MINUS,COMMA
      INTEGER F,D,BM,FP,FPP,FA,DCFLG
      INTEGER GET,BUF
      IF(NC-BM) 5,5,6
    5 GO TO (1,2,3,3),D
    1 MAX=2
      GO TO 4
    2 MAX=4
      GO TO 4
    3 MAX=8
    4 MO=1
    7 IF(IO) 8,9,8
    8 CALL UNPACK(S,MO,I1)
      CALL PBUF(NC,I1)
   10 MO=MO+1
      NC=NC+1
      IF(MO-M) 12,12,11
   12 IF(MO-MAX) 7,7,13
C     FORMAT ERROR
   13 CALL ERROR(41,IO)
      RETURN
   11 IF(IO) 14,15,14
   14 RETURN
   15 IF(MO-MAX)  16,16,14
   16  CALL PACK(S,MO,BLANK)
      MO=MO+1
      GO TO 15
    9 I1=BUF(NC)
       CALL PACK(S,MO,I1)
      GO TO 10
    6 IEND=3
C     IO BUFFER OVER
      CALL ERROR(1,IO)
      RETURN
      END
C
C
      SUBROUTINE CONVI(S)
C   INPUT I,F,E,D CONVERSION
C   T.TSUGITA
      COMMON /FMTC/FP,FPP,FA,N,NP
      COMMON/CONV/M,K,F,D,BM,IEND,NC,IO
C
      COMMON /EBCDIC/BLANK,DMIN,DMAX,LP,RP,PIRI,SLASH,PLUS,MINUS,COMMA
      COMMON/POIN/DR,SI,DI,SR,POINT,SIGN,KO
      COMMON /VVVV/V
      DIMENSION FPP(10),FA(10)
C
      DATA EBCD/196/,EBCE/197/
C
      INTEGER F,D,BM,FP,FPP,FA,DCFLG


      INTEGER BLANK,DMIN,DMAX,RP,LP,COMMA,PIRI,SLASH,PLUS,MINUS
      INTEGER EBCD,EBCE,EXP,E,SIGN,POINT
      DOUBLEPRECISION S,T,AS,SS,DR
      INTEGER  BUF
      DOUBLEINTEGER IDI
C
      SIMAX=32767.
      SIMIN=-32768.
      DIMAX=2147483647.
      DIMIN=-2147483648.
C
      IF (NC-BM) 10,10,11
C  ERROR---01
   11 IEND=3
      CALL ERROR(1,IO)
      GO TO 700
   10 GO TO (12,13,13,14),F
   12 IF (D-2) 15,15,20
C  ERROR---51
   13 IF(D-3) 20,15,20
   14 IF (D-4) 20,15,20
   20 CALL ERROR(51,IO)
      GO TO 700
C
C
   15 MO=1
      INIS=0
      POINT=0
      EXP=0
      SIGN=1
      KO=0
C
      E=0
      SI=0.
      DI=0.
      SR=0.
      DR=0.
      IOVER0=76
      IOVER1=-78
C  BLANK CHECK
   55 MM=BUF(NC)
C
C
      IF (MM-BLANK)  30,35,30
   35 IF (INIS-0) 40,44,40
   45 INIS=1
   44 MO=MO+1
      NC=NC+1
      IF (MO-M) 55,55,200
C
C  NUMERIC CHECK
   30 IF (MM-DMIN) 60,65,65
   65 IF (MM-DMAX)  67,67,60
C   BLANK = ZERO
   40 MM=DMIN
```

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

C
   67 IF (EXP-0) 75,70,75
   75 E=E*10+MM-DMIN
      GO TO 45
C
   70 GO TO (71,72,73,74),D
   71 SI=SI*10.+FLOAT(MM-DMIN)
      GO TO 45
C
   72 DI=DI*10.+FLOAT(MM-DMIN)
      GO TO 45
C
   73 IF (POINT-0) 86,85,86
   86 K0=K0+1
   85 SR=SR*10.+FLOAT(MM-DMIN)
      GO TO 45
C
C
   74 IF (POINT-0) 91,90,91
   91 K0=K0+1
   90 DR=DR*10.+FLOAT(MM-DMIN)
      GO TO 45
C
C  CHARACTER  CHECK
   60 IF (MM     -PLUS) 105,100,101
  101 IF (MM     -MINUS) 105,100,102
  102 IF (MM-EBCD) 105,100,106
  106 IF (MM-EBCE) 111,100,111
C
  105 IF (MM     -PIRI) 111,150,111
C
C   ERROR --55
  111 CALL ERROR(55,IO)
      GO TO 700
C
  100 IF (INIS-0) 121,120,121
  120 IF (MM     -MINUS) 45,130,45
  130 SIGN=-1
      GO TO 45
  121 IF (F-2) 141,142,142
C
C  ERROR--52
  141 CALL ERROR(52,IO)
      GO TO 700
C
  142 IF (EXP-0) 153,151,153
  153 IF (E-0) 741,154,741
  741 CALL ERROR(57,IO)
      GO TO 700
  151 IF (POINT-2) 152,154,152
C
  152 CALL POINTZ
C
  154 IF (MM     -MINUS) 160,161,160
  161 EXP=2


      GO TO 45
  160 EXP=1
      GO TO 45
C
  150 IF (F-1) 165,165,166
C
C  ERROR ---53
  165 CALL ERROR(53,IO)
      GO TO 700
C
  166 IF (POINT-1) 170,171,170
C
C  ERROR---54
  171 CALL ERROR(54,IO)
      RETURN
C
  170 POINT=1
      K0=0
      GO TO 45
C
C   INTEGER  UNDER FLOW AND OVER FLOW CHECK
  200 IF (D-1) 201,205,201
  205 IF (SIMIN-SI) 206,206,339
  206 IF (SI-SIMAX) 207,207,339
  207 ISI=SI
      ISI=ISI*SIGN
C
      DO 371 II=1,2
       CALL UNPACK (ISI,II,NSI)
      CALL PACK (S,II,NSI)
C
  371 CONTINUE
      RETURN
C
  201 IF (D-2) 202,210,202
  210 IF (DIMIN-DI) 211,211,339
  211 IF (DI-DIMAX) 212,212,339
C
  212 IDI=DI
      IDI=IDI*SIGN
      DO 372 JJ=1,4
      CALL UNPACK (IDI,JJ,MDI)
      CALL PACK (S,JJ,MDI)
  372 CONTINUE
      RETURN
C
  202 IF (POINT-2) 220,225,220
  220 CALL POINTZ
      GO TO 225
C
  230 IF (D-3) 232,231,232
  231 V=SR*FLOAT(SIGN)
      DO 8712 II=1,4
      CALL UNPACK (V,II,MSR)
      CALL PACK (S,II,MSR)
```

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

 8712 CONTINUE
      GO TO 700
  232 S=DR*FLOAT(SIGN)
      GO TO 700
C
  225 IF (EXP-0) 240,230,240
  240 IF (D-3)   245,241,245
C
  241 IF (EXP-1) 247,248,247
  247 IF (E-76) 702,702,703
  703 S=0.
      GO TO 700
  702 V=SR*FLOAT(SIGN)/10.**E
      DO 8710 II=1,4
      CALL UNPACK (V,II,MSR)
      CALL PACK (S,II,MSR)
 8710 CONTINUE
      GO TO 700
  248 IF (E-76) 318,318,309
C
C  ERROR---56
  309 CALL ERROR(56,IO)
      GO TO 700
C
  318 V=SR*FLOAT(SIGN)*10.**E
      DO 8711 II=1,4
      CALL UNPACK (V,II,MSR)
      CALL PACK (S,II,MSR)
 8711 CONTINUE
      GO TO 700
C
  245 IF (EXP-1) 325,320,325
  325 IF (E-76) 326,326,327
  327 S=0.
      GO TO 700
  326 S=DR*FLOAT(SIGN)/10.**E
      GO TO 700
C
  320 IF (E-76) 340,340,339
C
C  ERROR---56
  339 CALL ERROR(56,IO)
      GO TO 700
C
  340 S=DR*FLOAT(SIGN)*10.**E
      GO TO 700
C
  700 RETURN
      END
C
      SUBROUTINE  POINT2
      COMMON/POIN/DR,SI,DI,SR,POINT,SIGN,K0
      COMMON/CONV/M,K,F,D,BM,IEND,NC,IO
      INTEGER SIGN,POINT,D,BM,F
      DOUBLEPRECISION DR


C
      IF (D-3)  900,910,900
  910 IF (POINT-0) 920,930,920
  930 SR=SR/10.**K
      GO TO 960
C
  900 IF (POINT-0) 940,950,940
  940 DR=DR/10.**K0
      GO TO 960
C
  950 DR=DR/10.**K
      GO TO 960
  920 SR=SR/10.**K0
  960 POINT=2
      RETURN
      END
C
C
      SUBROUTINE CONV2 (S)
C    T.YAMADA
C  OUTPUT  I,F,E,D CONVERSION
      DOUBLEPRECISION DE
      COMMON /FTYPE/DE
      COMMON/FMTC/FP,FPP,FA,N,NP
      COMMON/CONV/M,K,F,D,BM,IEND,NC,IO
      COMMON/EBCDIC/BLANK,DMIN,DMAX,LP,RP,PIRI,SLASH,PLUS,MINUS,COMMA
      DIMENSION FPP(10),FA(10)
      INTEGER F,D,BM,FP,FPP,FA,DCFLG
C     SPECIAL CHARACTERS
      INTEGER BLANK,DMIN,DMAX,LP,RP,PIRI,SLASH,PLUS,MINUS,COMMA
      INTEGER EBCD,EBCE
      DOUBLEINTEGER IO
      DOUBLEINTEGER LIABS,LDFIXS
      DOUBLEPRECISION S,T,AS
      DOUBLEPRECISION DC10
      DATA EBCD/196/,EBCE/197/
C
C     *** CHECK OF BYTE COUNTS ***
C
C
      DC10=10.
      IF (NC+M-BM)  100,100,9001
 9001 IEND=3
      IE=1
      IF (NC- BM) 9100,9300,9300
 9100 M=BM-NC
 9200 CALL ASTSET
 9300 CALL ERROR (IE,IO)
 9400 RETURN
  100 GO TO (110,120,130,150),D
  110 ID=0
      DO 111 I=1,2
          CALL UNPACK (S,I,LD)
  111 CALL PACK (IS,I,LD)
      S=IS
```

```
...,*...,1...,*...,2...,*...,3...,*...,4...,*...,5...,*...,6...,*...,7...,*...,8

      GO TO 150
  120 DO 121 I=1,4
          CALL UNPACK (S,I,LD)
  121 CALL PACK (ID,I,LD)
      S=ID
      GO TO 150
  130 DO 131 I=1,4
          CALL UNPACK (S,I,LD)
  131 CALL PACK (RS,I,LD)
      S=RS
C
  150 IF (  F=1  ) 9061,1000,200
  200 IF (  F=2  ) 3000,2000,3000
C
C
C     *** I CONVERTION ( F=1 ) ***
C
 1000 IF (  D=2  ) 1010,1010,9062
C
C     *** ERROR 62 ***
C
C
 9061 IE=61
      GO TO 9200
 9062 IE=62
      GO TO 9200
C
 1010 MC=NC+M-1
 1011 M1=M
      IF (   S   ) 1012,1013,1013
 1012 M1=M1-1
 1013 IF (  F=1  ) 9061,1015,1014
 1014 M1=M1-K-1
 1015 CALL ICONV (S,M1,MC,M3,IERR)
C
      IF ( IERR  ) 1016,1017,1016
C    ** ERROR 65  **
C   WIDTH OVER
 1016 IE=65
      GO TO 9200
 1017 MC=MC-M3
      IF (   S   )1018,1030,1030
C
C     *** SET OF MINUS  SIGN ***
C
 1018 CALL PBUF(MC,MINUS)
      GO TO 1040
 1030 MC=MC+1
C
 1040 MC=MC-1
      IF (MC=NC+1) 1041,1050,1041
C
C     *** SET OF BLANK ***
C
 1041 CALL PBUF(MC,BLANK)


      GO TO 1040
 1050 IF (  F=1  ) 1060,1051,1060
 1051 NC=NC+M
      RETURN
C     *** F CONVERTION ***
C
 1060 MC=NC+M-K-1
C
C
      CALL PBUF(MC,PIRI)
      IF (   K   ) 1051,1051,1061
C
C     *** F CONVERTION ( F=2 ) ***
C
 1061 IF (   S   ) 1090,1062,1090
C
C     *** S=0 ANS=0.0 ***
C
 1062 MC=MC+1
 1070 CALL PBUF(MC,DMIN)
C
 1080 MC=MC+1
      IF (MC=NC=M) 1081,1051,1081
 1081 CALL PBUF(MC,BLANK)
      GO TO 1080
C
 1090 MC=NC+M-1
      T=DE*10.**K
      CALL ICONV (T,K,MC,M3,IERR)
 2010 M3=M3+1
      IF (C-M3)   1051,2011,2011
 2011 MCM3=MC=M3+1
      CALL PBUF(MCM3,DMIN)
      GO TO 2010
C
 2000 IF (  D=2  ) 9062,9062,2100
 2100 IF (M=K=3  ) 2201,2200,2200
 2201 IF(S) 2202,2203,2203
 2202 K=M-3
      GO TO 2204
 2203 K=M-2
 2204 IF(K) 9061,2200,2200
 2200 MC=NC+M-K-2
      IF (S) 2301,1011,2302
 2301 S=S-5.*10.**(-K-1)
      GO TO 1011
 2302 S=S+5.*10.**(-K-1)
      GO TO 1011
C
C
C     *** E,D CONVERTION ***
C
 3000 IF (   S   ) 3001,3040,3007
 3007 IF (M=K=6) 3008,3003,3003
 3008 K=M-6
```

```
....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8

      GO TO 3009
 3001 IF (M-K-7  ) 3002,3003,3003
 3002 K=M-7
 3009 IF (K) 9061,3003,3003
 3003 IF (   K   ) 3040,3004,3004
 3004 AS= DABS (S)
      M2= DLOG10 (AS)
      IF (  M2   ) 3010,3005,3005
 3005 IE1=0
      M2=M2+1
      GO TO 3020
 3010 IE1=1
 3020 T=AS*DC10**(-M2+K)+0.5
      MADJ=DLOG10 (T)
      IF (K-MADJ) 3021,3022,3022
 3021 M2=M2+1
      GO TO 3020
 3022 IE2= IABS (M2/10)
      IE3= IABS (M2) -IE2*10
 3040 MC=NC
 3041 IF (MC-NC-M+K+7) 3070,3042,3070
 3042 IF (   S   ) 3043,3050,3050
 3043 CALL PBUF(MC,MINUS)
      GO TO 3060
 3050 CALL PBUF(MC,BLANK)
 3060 MC=MC+1
      GO TO 3041
C
 3070 IF (MC-NC-M+K+6) 3050,3071,3050
 3071 CALL PBUF(MC,DMIN)
      MC=MC+1
      CALL PBUF(MC,PIRI)
      IF (   S   ) 3080,3072,3080
 3072 MC=MC+1
      GO TO 1070
 3080 MC=NC+M-5
      CALL ICONV (T,K,MC,M3,IERR)
      MC=MC+1
      IF (  F-3  ) 3081,3090,3081
 3081 CALL PBUF(MC,FBCD)
      GO TO 3100
 3090 CALL PBUF(MC,FBCF)
 3100 MC=MC+1
      IF (  IE1  ) 3101,3101,3110
 3101 CALL PBUF(MC,BLANK)
      GO TO 3120
 3110 CALL PBUF(MC,MINUS)
 3120 MC=MC+1
      IE2=IE2+DMIN
      CALL PBUF(MC,IE2)
      MC=MC+1
      IE3=IE3+DMIN
      CALL PBUF(MC,IE3)
      GO TO 1071
      END


C
       SUBROUTINE ICONV (SS,IM,MCC,M3,IERR)
      REAL ISS
      DOUBLEPRECISION SS,SI,USI
      DOUBLEPRECISION DC10
      DOUBLEPRECISION DE
      COMMON /FTYPE/DE
      DATA IERC /240/
      DC10=10.
      IERR=0
      ISW=0
      SI = DABS (SS)
      IF(SI-1.) 6,7,7
    6 IEXP=0
      GO TO 8
    7 IEXP=DLOG10 (SI)
      IEXPD=IEXP+1
      ISS=SI-   DC10**IEXPD
      IF (ISS) 500,400,400
  400 IEXP=IEXPD
  500 IEXP=IEXP+1
    5 IEXP=IEXP-1
    8 IF (ISW) 10,10,20
   10 M3=IEXP+1
      IF ( M3 -IM) 11,11,100
   11 ISW=1
   20 IF (IEXP-1) 40,30,30
   40 IR=SI
      DE=SI-FLOAT(IR)
      ISW=0
      GO TO 50
   30 USI=DC10**IEXP
      IR=SI/USI
      SI=SI-USI*FLOAT (IR)
   50 MMMM=IERC+IR
      MMM=MCC-IEXP
      CALL PBUF(MMM,MMMM)
      IF (ISW) 200,200,5
  100 IERR=1
  200 RETURN
      END
C
      SUBROUTINE  ASTSET
      COMMON/CONV/M,K,F,D,BM,IEND,NC,ID
      INTEGER F,D,BM,FP,FFP,FA,DCLD
      DATA IAST/92/
      DO 100 I=1,M
      I1=NC+I-1
  100 CALL PBUF(I1,IAST)
      NC=NC+M
      RETURN
      END
C
```