

AD-A063 407

BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 17/2
EXPLORATORY SYSTEMS CONTROL MODEL (ESM). BOOK 1. FORTRAN. SOFTW--ETC(U)
APR 77

DCA100-75-C-0054

NL

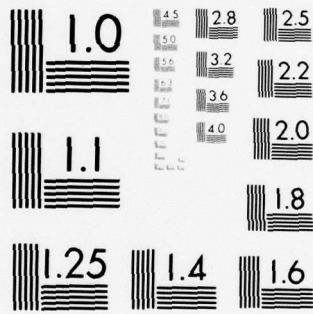
UNCLASSIFIED

66143-3

1 OF 3

AD
A063407





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-E100 140

① LEVEL III

66143-3

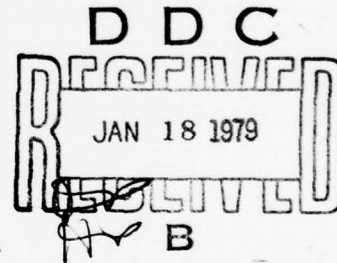
BOOK 1
FORTRAN

APRIL 1977



SOFTWARE MAINTENANCE MANUAL

EXPLORATORY SYSTEMS
CONTROL MODEL (ESM)



for
THE DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, D.C. 20305

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Burroughs Corporation

Federal and Special Systems Group

Paoli, Pa. 19301

78 02 11 201

ADA063407

DDC FILE COPY

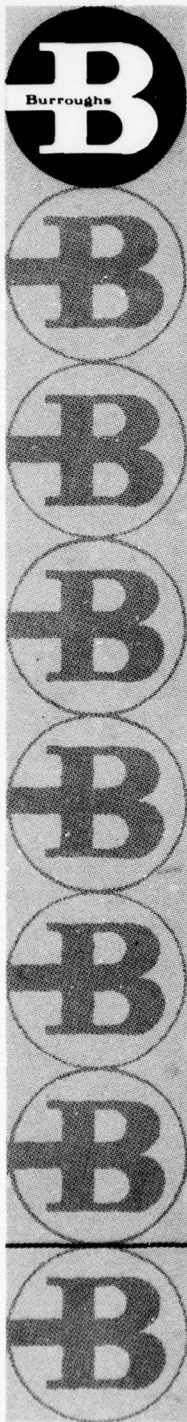
14 66143-3

BOOK 1
FORTRAN

A063357

11 APRIL 1977

12 252p.



SOFTWARE MAINTENANCE MANUAL

6 EXPLORATORY SYSTEMS
CONTROL MODEL (ESM).
Book 1, Fortran.

CONTRACT DCA 100-75-C-0054

15

for
THE DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, D.C. 20305

9 Final rept. Jul 75 - Apr 77.

BK 2 - A063 394

- 1 - A063 374
- 2 - A063 357

Burroughs Corporation

Federal and Special Systems Group

Paoli, Pa. 19301

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

070 040

↓
The software describe in this

TU10 Mag tapes

FOREWORD

→ IS Cont

This publication is the Software Maintenance Manual for the Exploratory Systems Control Model (ESM). The software described is contained on four system tapes. Book 1 contains description, flowcharts, and listings for programs written in FORTRAN. Book 2 contains description, flowcharts, and listings for programs written in MDMP Assembly Language. This manual was prepared by the Burroughs Corporation and is submitted in accordance with the requirements of contract DCA100-75-C-0054.

→ (cont on p 1x)

ACCESSION for	
MTS	White Section <input checked="" type="checkbox"/>
DOC	Staff Section <input type="checkbox"/>
UNARRANGED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISCUSSION OF DATA AND METHODS	
Dist.	SPECIAL
A	

CONTENTS

INTRODUCTION	<u>Page</u>
Book 1	
1. FORTRAN PROGRAMS	1-1
1.1. MDMP L Assembler	1-1
1.2. M1710 Common Area	1-82
1.3. User Language	1-85
1.4. ESM Loader Utility	1-203
1.5. Record Move Utility	1-207
1.6. Interprocess Communication and Resource Sharing Demonstration Program	1-220
1.7. Diagnostics	1-239
1.7.1. PDP-11 Interface - PDP	1-239
1.7.2. Control Memory - CONMEM	1-239
Book 2	
2. MDMP L MICROCODE PROGRAMS	2-1
2.1. NCU Microcode	2-1
2.2. CIE Microcode	2-11
2.3. Diagnostics	2-193
Appendix A. B7* Programming Manual	A-1
Appendix B. MDMP L Instruction List	B-1

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1-1	SUSAN - Main Program	1-17
1-2	RESCAN	1-27
1-3	SCAN	1-32
1-4	SQUASH	1-39
1-5	COLUMN	1-41
1-6	CONDIT	1-45
1-7	LITRL	1-52
1-8	LOGIC	1-61
1-9	P0000	1-94
1-10	P1000	1-108
1-11	P2000	1-117
1-12	P3000	1-121
1-13	P3001	1-133
1-14	P4000	1-140
1-15	P4001	1-160
1-16	P5000	1-177
1-17	RDLOOP	1-181
1-18	WRLOOP	1-183
1-19	HST	1-185
1-20	ESMLDR	1-204
1-21	RCMV5	1-208
1-22	PROC5	1-222
1-23	PDP	1-240
2-1	NCU State Diagram and Software Modules	2-2
2-2	Mailbox Page Description	2-3
2-3	ESM/NCU	2-5
2-4	CIE Software Modules	2-12
2-5	HST5.DAT	2-17
2-6	CRT4.DAT	2-71
2-7	GAT7.DAT	2-120
2-8	MEMCKO	2-198
2-9	BLOUT	2-206
2-10	GTBO	2-212
2-11	CTCCO	2-215
2-12	CTCGO	2-229
2-13	CRTOBJ	2-237
2-14	PDPO	2-248
2-15	LPCKO	2-251
2-16	GTBOA	2-256

(cont h piii)

Introduction

The ESM system software is contained on four TU10 Magtapes. Tape #1 contains the source, object, task, overlay description language, message, system, ATEC simulation, and command files used for the User Language, Record Move Utility, and Interprocess Communication and Resource Sharing Demonstration Programs for host processors A and B. Tape #2 contains microcode source and object files for loading the eleven B7* CIE microprocessors, and the microcode loader utility (ESMLDR). Tape #3 contains the task, source, object, and overlay description language files for the Mini-D Micro Programming Language (MDMPL) Assembler. Tape #4 contains the ESM Diagnostic Library.

References for the FORTRAN language used include the Digital Equipment Corporation documents PDP-11 FORTRAN Language Reference Manual (DEC-11-CFLRA-C-D) and IAS/RX-11 FORTRAN IV User's Guide (DEC-11-LMFUA-C-D). It is also assumed that the reader is familiar with the PDP-11 RSX11M operating system (Version 2) including MCR commands (Reference - RSX11M Operator's Procedures Manual - DEC-11-OM06A-B-D) and the utilities EDI, FLX, and PIP (Reference - RSX11M Utilities Procedures Manual - DEC-11-OMOGA-B-D).

References for the MDMPL Assembler include Appendix A of this manual which provides B7* programming information, Appendix B of the manual which provides an MDMPL Instruction List, and Section 4.6 of the ESM User's Manual which describes Assembler use, CIE Instruction Functions, and programming examples.

In general, FORTRAN programs are stored in UIC [20,20], CIE Microcode programs are stored in UIC[1,20], and Diagnostics are stored in UIC[1,4]. System Tape Directory Listings are presented below.

ESM TAPE #1 - USER LANGUAGE

MCR>FLX TTO:=MTO:[*,*]*.*./LI

DIRECTORY	MTO:[0,0]
18-MAR-77	
RCHV1.FOR	9. 18-MAR-77 <233> [20,20]
RCHV1.OBJ	22. 18-MAR-77 <233> [20,20]
RCHV1.ODL	1. 18-MAR-77 <233> [20,20]
RCHV1.TSK	53. 18-MAR-77 <233> [20,20]
RCHV5.ODL	1. 18-MAR-77 <233> [20,20]
RCHV5.FOR	9. 18-MAR-77 <233> [20,20]
RCHV5.TSK	53. 18-MAR-77 <233> [20,20]
RCHV5.OBJ	22. 18-MAR-77 <233> [20,20]
PRDC1.TSK	48. 18-MAR-77 <233> [20,20]
PROC1.OBJ	26. 18-MAR-77 <233> [20,20]
PROC1.ODL	1. 18-MAR-77 <233> [20,20]
PROC1.FOR	11. 18-MAR-77 <233> [20,20]
PROC5.OBJ	27. 18-MAR-77 <233> [20,20]
PROC5.ODL	1. 18-MAR-77 <233> [20,20]
PROC5.TSK	48. 18-MAR-77 <233> [20,20]
PROC5.FOR	11. 18-MAR-77 <233> [20,20]
M1710.FOR	1. 09-MAR-77 <233> [20,20]
M1710.OBJ	1. 09-MAR-77 <233> [20,20]
M1710.TSK	3. 09-MAR-77 <233> [1,1]
M1710.STB	1. 09-MAR-77 <233> [1,1]
INFOPH.OBJ	67. 09-MAR-77 <233> [20,20]
MSP.OBJ	36. 09-MAR-77 <233> [20,20]
EFTERD.OBJ	9. 09-MAR-77 <233> [20,20]
EFCKTD.OBJ	4. 09-MAR-77 <233> [20,20]
EFTRKD.OBJ	4. 09-MAR-77 <233> [20,20]
EFLOCF.OBJ	5. 09-MAR-77 <233> [20,20]
EFDIR.OBJ	2. 09-MAR-77 <233> [20,20]
STESH.CMD	1. 09-MAR-77 <233> [20,20]
ESMLDR.TSK	32. 09-MAR-77 <233> [20,20]
MDMPL.TSK	86. 09-MAR-77 <233> [20,20]
USROVL.ODL	1. 09-MAR-77 <233> [20,20]
P0000.OBJ	21. 09-MAR-77 <233> [20,20]
P0000.FOR	11. 09-MAR-77 <233> [20,20]
P00001.OBJ	21. 09-MAR-77 <233> [20,20]
P00001.FOR	11. 09-MAR-77 <233> [20,20]
P1000.FOR	6. 18-MAR-77 <233> [20,20]
P1000.OBJ	12. 18-MAR-77 <233> [20,20]
P10001.FOR	6. 18-MAR-77 <233> [20,20]
P10001.OBJ	12. 18-MAR-77 <233> [20,20]
P2000.FOR	5. 18-MAR-77 <233> [20,20]
P2000.OBJ	11. 18-MAR-77 <233> [20,20]
P3000.FOR	11. 18-MAR-77 <233> [20,20]
P3000.OBJ	26. 18-MAR-77 <233> [20,20]
P3001.OBJ	17. 18-MAR-77 <233> [20,20]
P3001.FOR	7. 18-MAR-77 <233> [20,20]
P4000.FOR	13. 18-MAR-77 <233> [20,20]
P4000.OBJ	29. 18-MAR-77 <233> [20,20]
P40001.FOR	13. 18-MAR-77 <233> [20,20]
P40001.OBJ	29. 18-MAR-77 <233> [20,20]
P4001.FOR	11. 18-MAR-77 <233> [20,20]

P5000.FOR	4.	18-MAR-77	<233>	[20,20]
P5000.OBJ	8.	18-MAR-77	<233>	[20,20]
RDLOOP.OBJ	2.	18-MAR-77	<233>	[20,20]
RDLOOP.FOR	1.	18-MAR-77	<233>	[20,20]
WRLOOP.FOR	1.	18-MAR-77	<233>	[20,20]
WRLOOP.OBJ	2.	18-MAR-77	<233>	[20,20]
HST.FOR	4.	18-MAR-77	<233>	[20,20]
HST.OBJ	11.	18-MAR-77	<233>	[20,20]
HST1.FOR	4.	18-MAR-77	<233>	[20,20]
HST1.OBJ	11.	18-MAR-77	<233>	[20,20]
USRLN5.TSK	98.	18-MAR-77	<233>	[20,20]
USRLN1.TSK	98.	18-MAR-77	<233>	[20,20]

TOTAL OF 1181. BLOCKS IN 65. FILES

>

ESM TAPE #2 - CIE MICROCODE

MCR>FLX TT1:=MT0:[*,*]*.*/LI

DIRECTORY	MT0:[0,0]	
12-MAR-77		
HST1. OBJ	9.	12-MAR-77 <233> [1,20]
GAT2. OBJ	8.	12-MAR-77 <233> [1,20]
GAT3. OBJ	8.	12-MAR-77 <233> [1,20]
CRT4. OBJ	9.	12-MAR-77 <233> [1,20]
HST5. OBJ	9.	12-MAR-77 <233> [1,20]
GAT6. OBJ	8.	12-MAR-77 <233> [1,20]
GAT7. OBJ	8.	12-MAR-77 <233> [1,20]
CRT8. OBJ	9.	12-MAR-77 <233> [1,20]
HST9. OBJ	8.	12-MAR-77 <233> [1,20]
GAT10. OBJ	8.	12-MAR-77 <233> [1,20]
GAT11. OBJ	8.	12-MAR-77 <233> [1,20]
HST1L. OBJ	9.	12-MAR-77 <233> [1,20]
CRT4L. OBJ	9.	12-MAR-77 <233> [1,20]
HST5L. OBJ	9.	12-MAR-77 <233> [1,20]
CRT8L. OBJ	9.	12-MAR-77 <233> [1,20]
CRT4S. OBJ	9.	12-MAR-77 <233> [1,20]
CRT8S. OBJ	9.	12-MAR-77 <233> [1,20]
HST9S. OBJ	8.	12-MAR-77 <233> [1,20]
CRT4. DAT	119.	12-MAR-77 <233> [1,20]
HST5. DAT	100.	12-MAR-77 <233> [1,20]
GAT7. DAT	97.	12-MAR-77 <233> [1,20]
HST9. DAT	102.	12-MAR-77 <233> [1,20]
ESMLDR. FOR	2.	12-MAR-77 <233> [20,20]
ESMLDR. OBJ	5.	12-MAR-77 <233> [20,20]
ESMLDR. TSK	32.	12-MAR-77 <233> [20,20]
MDMPL. TSK	86.	12-MAR-77 <233> [20,20]

TOTAL OF 697. BLOCKS IN 26. FILES

>

ESM TAPE #3 - MDML ASSEMBLER

FLX CLO:=MT0:[20,20]*./LI

DIRECTORY	MT0:[20,20]
26-FEB-77	

FASS.OBL	1.	26-FEB-77 <233>
MDMLST.COMD	1.	26-FEB-77 <233>
SUSAN.FOR	13.	26-FEB-77 <233>
BLOCK.FOR	3.	26-FEB-77 <233>
RESCAN.FOR	7.	26-FEB-77 <233>
WRT.FOR	9.	26-FEB-77 <233>
SQUASH.FOR	3.	26-FEB-77 <233>
SCAN.FOR	5.	26-FEB-77 <233>
COLUMN.FOR	7.	26-FEB-77 <233>
CONDIT.FOR	11.	26-FEB-77 <233>
LITRL.FOR	13.	26-FEB-77 <233>
LOGIC.FOR	16.	26-FEB-77 <233>
LOGICA.FOR	13.	26-FEB-77 <233>
MDML.TSK	86.	26-FEB-77 <233>
SUSAN.OBJ	27.	26-FEB-77 <233>
BLOCK.OBJ	1.	26-FEB-77 <233>
RESCAN.OBJ	10.	26-FEB-77 <233>
WRT.OBJ	11.	26-FEB-77 <233>
SQUASH.OBJ	3.	26-FEB-77 <233>
SCAN.OBJ	7.	26-FEB-77 <233>
COLUMN.OBJ	11.	26-FEB-77 <233>
CONDIT.OBJ	23.	26-FEB-77 <233>
LITRL.OBJ	23.	26-FEB-77 <233>
LOGIC.OBJ	38.	26-FEB-77 <233>
LOGICA.OBJ	38.	26-FEB-77 <233>

TOTAL OF 380. BLOCKS IN 25. FILES

>

ESM TAPE #4 - DIAGNOSTICS

MCR>FLX TT1:=MT0:[*,*]*.*/LI

DIRECTORY MT0:[0,0]
10-MAR-77

MEMCKO. OBJ	2.	26-FEB-77	<233>	[1,4]
BLOUT. OBJ	2.	26-FEB-77	<233>	[1,4]
LPCKO. OBJ	2.	26-FEB-77	<233>	[1,4]
GTBO. OBJ	1.	26-FEB-77	<233>	[1,4]
CRTOBJ. OBJ	2.	26-FEB-77	<233>	[1,4]
PDP0. OBJ	1.	26-FEB-77	<233>	[1,4]
GTBOA. OBJ	1.	26-FEB-77	<233>	[1,4]
CTCGO. OBJ	2.	26-FEB-77	<233>	[1,4]
FDP. OBJ	6.	26-FEB-77	<233>	[1,4]
CTCCO. OBJ	3.	26-FEB-77	<233>	[1,4]
CONMEM. OBJ	5.	26-FEB-77	<233>	[1,4]
BLKS. DAT	9.	26-FEB-77	<233>	[1,4]
MEMCK. DAT	10.	26-FEB-77	<233>	[1,4]
LPCK. DAT	8.	26-FEB-77	<233>	[1,4]
FDP. DAT	4.	26-FEB-77	<233>	[1,4]
GTB. DAT	4.	26-FEB-77	<233>	[1,4]
GTBA. DAT	5.	26-FEB-77	<233>	[1,4]
CTCG. DAT	11.	26-FEB-77	<233>	[1,4]
CTCC. DAT	23.	26-FEB-77	<233>	[1,4]
CRICK. DAT	21.	26-FEB-77	<233>	[1,4]
FDP. FOR	3.	26-FEB-77	<233>	[1,4]
FDP. TSK	31.	26-FEB-77	<233>	[1,4]
CONMEM. FOR	2.	26-FEB-77	<233>	[1,4]
CONMEM. TSK	32.	26-FEB-77	<233>	[1,4]
TI. DAT	5.	10-MAR-77	<233>	[1,4]
TI. OBJ	1.	10-MAR-77	<233>	[1,4]

TOTAL OF 196. BLOCKS IN 26. FILES

>

1. FORTRAN PROGRAMS

1.1 MDMPL MICROCODE ASSEMBLER

The eleven B7*CIE microprocessors are loaded with microcode object files that are stored on the PDP11 processor connected to loop #2. The object files consist of records made up of 128 12 bit micro-instructions. An MDMPL assembler written in FORTRAN is provided with the ESM for microcode creation or modification. The Mini-D Microprogramming Language (MDMPL) is described in the B7* documentation package. Microcode source files can be created or edited using the RX-11M Editor Utility (EDI). The assembler takes a microcode source file that consists of ASCII, 80 character fixed fields and translates it into a 256 byte binary object file.

An MDMPL source file has the following format:

```

$12BIT (first line - start in column 7)

PROGRAM-ID name. (second line - start in column 8)

value statements (start in column 15)

program statements (start in column 15)

END?. (start in column 15)

```

Statements are always terminated by a period. Labels start in column 8 and terminate with a period. Labels can consist of up to 7 alpha-numeric characters and may not contain embedded assembler reserved words, e.g., EXT, LCL, LST, MST, AOV, IF, STEP, SKIP, ELSE. Statements may not start at or before column 8, and by convention start at column 15. Comments following statements, by convention, start at column 40. A * in column 7 indicates a comment card. After the file is edited using the EDI Utility, it must be put into a fixed record, 80 character, formatted ASCII card images for input to the MDMPL assembler. This can be done by writing the file to tape, and then back again to disk using the file transfer (FLX) utility, e.g.

```
FLX MT0:/DO=DK0:[1,4]MICRO.DAT/RS
```

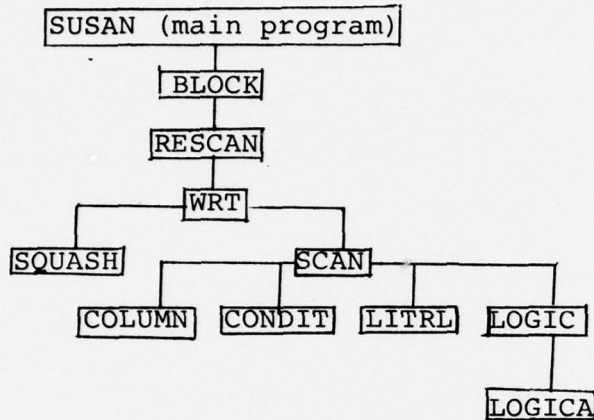
```
FLX DK0:/FA:80.=MT0:[1,4]MICRO.DAT
```

FLX and EDT commands are given in the RSX11M Utilities Procedures Manual.

When the source file is properly formatted on disk (latest version) run the MDMPL assembler by entering RUN [20,20]MDMPL on the DECSCOPE. The program will prompt for the source filename and object filename. By convention, source microcode files are of type DAT, and object microcode filenames end with the character "O" and of type OBJ. Default conditions allow for program listing with possible error messages on the DECSCOPE. Output may be stopped by entering control C, and it may be resumed by hitting the return key. For a hard copy printout enter RED TT0: = TT1: before running MDMPL. The number of errors is printed at the end of the program.

THE PROGRAM

The assembler is divided into eleven modules. The assembler's modules are structurally overlaid so that the assembler is able to be executed when only small partitions of main memory are available. The overlay structure is in the file FASS.ODL and is set up as follows:



In this type of structure, subroutines on the same line are not allowed to call each other, but are able to call any other subroutine. Following is a description of each module including the important variables of the program.

Main Program - SUSAN

Important Variables:

All variables are of type INTEGER except for those variables beginning with the letter "Z" or those explicitly stated otherwise.

ZA - ZA is the name of the source file that contains the micro-code to be translated. This variable is formatted as REAL*8 and is dimensioned as 3. This allows the name of the file to be up to 24 alpha-numeric characters in length but the first character of the name must be a letter.

ZX - The assembler writes the binary code represented by the micro-code of the source file to the object file, ZX. The variable is REAL*8 and is dimensioned as 3. The name of the file can be as long as 24 alpha-numeric characters but the first character of the name must be a letter.

MPAD - MPAD is initialized at 0. MPAD stands for the Memory Program Address Descriptor. Each address contains 8, 12, or 16 bits of information, depending on the value of DEV. This is an integer type incremented by one every time information is put into the address. MPAD is not incremented for comments, "VALUE" statements or labels but is incremented by two for "CALL" and "GOTO" statements if DEV equals 12 or 16.

Y - Y is similar to MPAD. Y is one greater than MPAD so that

a DO loop can be used in the program. Y is the subscript of CODE and is an INTEGER type variable.

CODE - CODE contains the information to be sent to the object file. CODE is 256 bytes long and is written over to the object file whenever the 256 bytes are filled or when the program is finished. CODE is dimensioned at 128 and is of type INTEGER. LITRL, LOGIC, LOGICA and CONDIT write the correct information into CODE.

COL - COL is the "workspace" for the assembler. Each record of the source file is read into COL, one at a time. The assembler looks at this array checking for the various conditions necessary. COL is of type INTEGER and is dimensioned at 80.

NCOL - NCOL is created by SQUASH by taking all spaces or blanks out of COL. This allows the assembler to look at a microcode instruction and expect a semi-ordered field. The array NCOL is dimensioned at 30 and of type INTEGER.

N - N keeps track of the number of errors. If N is equal to 10, the assembler will stop. N is an integer variable.

Alpha-numeric Characters - A - Z and 0 - 9 are represented as variables with Q's between the characters themselves. For example, "K" = QKQ and "5" = Q5Q. Other special variables are as listed:

QAR = "+"	QER = "."
QBR = "-"	QFR = "\$"
QCR = "*"	QGR = "G"
QDR = "="	QHR = " "

All alpha-numeric variables are of type LOGICAL*1.

Eg. Searching for an equal sign in column 4 is done by the code " IF (NCOL(4).EQ.QDR)..."

FVAR, VVAR, VCON, VAR, CON - FVAR contains the variable name from a "B =" literal assignment statement. This corresponds to the "VALUE" statement of label with the same variable. Each subscript of the array VVAR contains the variable before "VALUE" and VCON contains the constant corresponding to it. VAR contains the labels that start in column 8 and CON's array contains the address of where each label is located. FVAR, VVAR and VAR are LOGICAL*1 variables. VCON and CON are INTEGERS.

(Eg. B = ZERO. ZERO is put into FVAR. The assembler then looks at VVAR and VCON which respectively contain ZERO and 0 which was accomplished from the microcode instruction: ZERO VALUE 0.)

Description:

The main program (SUSAN) calls RESCAN, SQUASH, WRT, CONDIT, COLUMN, LITRL, LOGIC, SCAN. The only input is the source file ZA. The output files are the object file ZX, and the DECWRITER or the DECSCOPE. The main program looks at the file ZA eighty character records at a

time. It handles the "number of bits" record, the "Program - ID" record and the "END?." record. It also takes care of the preliminary testing necessary to send the microcode instruction line to the proper module for further testing and manipulation. The preliminary testing includes checking for a period, an equal sign, a character in column 8, an asterik, or any reserved words. Reserved words include "IF", "B=", "STEP", "EXEC". If an error occurs, the subroutine WRT is called to report the error to the user.

BLOCK DATA

Description:

The BLOCK DATA contains all the necessary COMMON areas and COMMON data necessary for the assembler. All variables are explicitly defined here.

COLUMN

Important Variables:

X - X contains a 1 if "VALUE" is in NCOL. If "VALUE" is not in NCOL, X is equal to 0 and informs the assembler that the instruction is a label. X is an integer.

DCOL - If a subscript of NCOL contains a numeric character, the same subscript of DCOL becomes its numeric decimal value. This array is dimensioned at 30 and is of INTEGER type.

NCOL - NCOL is created by SQUASH by taking all spaces or blanks out of COL. This allows the assembler to look at a microcode instruction and expect a semi-ordered field. The array NCOL is dimensioned at 30 and is of type INTEGER.

VVAR - VVAR is a list of the variables that come before the reserved word "VALUE" in a "VALUE" statement. Variables and labels are allowed to be up to 7 characters long. VVAR is dimensioned as (80, 8) and is of type LOGICAL*1.

C - The array C aids in the process of getting the variable before "VALUE" of a "VALUE" statement into VVAR. It is an INTEGER type and is dimensioned at 15.

VCON - VCON contains a list of all the constants from the "VALUE" statements. VVAR contains the variables corresponding to the constants in VCON. VCON receives its values from the variable AC. VCON is dimensioned at 80.

V - VVAR and VCON are lists of labels and constants corresponding to each "VALUE" statement found. V acts as a pointer for these queues. If the assembler is looking for the fifth "VALUE" statement, V will have a value of 5. V is an INTEGER.

AC - AC contains the constant following "VALUE" in decimal form. It uses the number represented by the decimal values in DCOL to form a constant. AC's maximum value is 255.

VAR - VAR is a list for all labels. VAR is dimensioned at (100, 8). The labels contained in VAR can be only 7 characters long.

L - L is similar to V. It is a pointer for VAR and CON. L's maximum value is 100.

CON - CON contains the MPAD value of the corresponding label found in VAR. CON's dimension is 100 and it is an INTEGER variable.

Description:

COLUMN is called by SUSAN. COLUMN calls SCAN and WRT. The variables V and L are sent as parameters between COLUMN and SUSAN. COLUMN is called by SUSAN if the reserved word "VALUE" is in the instruction or if there is a character in column 8. If the word "VALUE" is found, VVAR and VCON receive the correct values. If column 8 is occupied, VAR and CON receive their appropriate values. These arrays are used to calculate the code of an instruction when it reaches LITRL.

WRT

Important Variables:

VARF - VARF is the variable corresponding to the error messages. For every value of VARF there is an error message printed related to the syntax condition that was broken. VARF is an INTEGER. It is sent by the program calling WRT.

PCK - PCK is set at 1, 2, 3, 4 or 5, and is set by the program calling WRT. A "1" writes the line only (COL); a "2" writes the line (COL) with CODE(Y) = 77777, MPAD, the error message and increments N; a "3" writes the number of errors; a "4" writes the MPAD, CODE(Y) and the line (COL); and a "5" writes only the error message.

N - N keeps track of the number of errors. If N is equal to 10, the the assembler will stop. N is an INTEGER variable.

MPAD, CODE (Y), COL - The contents of all these are printed whenever a line is sent to the DECWRITER or DECSCOPE. This is executed if PCK equals "4".

MPAD - MPAD is initialized at 0. MPAD stands for the Memory Program Address Descriptor. Each address contains 8, 12 or 16 bits of information, depending on the value of DEV. This is an integer type variable incremented by one everytime information is put into the address. MPAD is not incremented for comments, "VALUE" statements or labels but is incremented by 2 for "CALL" or "GOTO" statements if DEV equals 12 or 16.

CODE - CODE contains the information to be sent to the object file. CODE is 256 bytes long and is written over to the object file whenever the 256 bytes are filled or when the program is finished. CODE is dimensioned at 128 and is of type INTEGER. LITRL, LOGIC, LOGICA, and CONDIT write the correct information into CODE.

COL - COL contains the "workspace" for the assembler. Each record of the source file is read into COL, one at a time. The assembler looks at this array checking for the various conditions necessary. COL is of type INTEGER and is dimensioned at 80.

Description:

WRT is called by SUSAN, COLUMN, CONDIR, LOGICA, LOGIC, LITRL and SQUASH. WRT writes its output to either the DECWRITER or the DECSCOPE. WRT is divided into 5 sections and depending on the value of PCK that section is executed. Depending on the value of MPAD, WRT will either write only the line; or write the line, the MPAD value, CODE(Y) value of 77777, the error message and increment N; or write the number of errors; or write the line, the MPAD value and the CODE(Y) value; or write only the error message. Most of the output writing of this assembler is accomplished by this subprogram.

CONDIT

Important Variables:

NCOL - NCOL is created by SQUASH by taking all spaces or blanks out of COL. This allows the assembler to look at a microcode instruction and expect a semi-ordered field. The array NCOL is dimensioned at 30 and of type INTEGER.

CODE - CODE contains the information to be sent to the object file. CODE is 256 bytes long and is written over to the object file whenever the 256 bytes are filled or when the program is finished. CODE is dimensioned at 128 and is of type INTEGER. LITRL, LOGIC, LOGICA and CONDIR write the correct information into CODE.

Y - Y is similar to MPAD. Y is one greater than MPAD so the a DO loop can be used in the program. Y is the subscript of CODE and is an INTEGER type variable.

MPAD - MPAD is initialized at 0. MPAD stands for the Memory Program Address Descriptor. Each address contains 8, 12 or 16 bits of information, depending on the value of DEV. This is an integer type variable incremented by one everytime information is put into the address. MPAD is not incremented for comments, "VALUE" statements or labels, but is incremented by 2 for "CALL" or "GOTO" statements if DEV equals 12 or 16.

Alpha-numeric characters - A - Z and 0 - 9 are represented as variables with Q's between the characters themselves. For example, "K" = QKQ and "5" = Q5Q. Other special variables are as listed:

QAR = "+"	QER = "."
QBR = "-"	QFR = "\$"
QCR = "*"	QGR = "G"
QDR = "="	QHR = " "

All alpha-numeric variables are of type LOGICAL*1.
(Eg. Searching for an equal sign in column 4 is done by the code
"IF (NCOL(4).EQ.QDR)...")

Description:

CONDIT calls SCAN and WRT. CONDIT is called by the main program SUSAN if any of the reserved words, "IF", "STEP", "SKIP", "JUMP", "EXEC" are found. All condition statements are sent here. There can not be any "-" or "+" signs and there must be a period. CONDIT checks for "STEP", "JUMP", "EXEC", "SKIP", "MST", "AOV", "LST", "ABT", "EXT", "LC1", "LC2" and "LC3". Depending on what is found, CODE(Y) representing that instruction is set to the correct value.

SQUASH

Important Variables:

NCOL - NCOL is created by SQUASH by taking all spaces or blanks out of COL. This allows the assembler to look at a microcode instruction and expect a semi-ordered field. The array NCOL is dimensioned at 30 and of type INTEGER.

COL - COL is the "workspace" for the assembler. Each record of the source file is read into COL, one at a time. The assembler looks at this array checking for the various conditions necessary. COL is of type INTEGER and is dimensioned at 80.

QER - Represents a "."

QHR - Represents a " ".

Description:

This subroutine calls WRT for its relay of error messages to the user. SQUASH is called by SUSAN and RESCAN. This subroutine creates NCOL. By ignoring all blanks, most microcode instructions are no more than 25 characters in length. This also enables fixed fields to be set up to allow for the scanning of reserved words in certain columns.
(Eg. such as "IF" in columns 1 and 2.)

SCAN

Important Variables:

A - A is either a 1, 2, 3, 4 or 5 depending on the length of the Reserved Word being scanned for. If A is greater than 5, then SCAN simply returns to the subprogram that called it.

B, C, D, E, F - Contain the characters of the reserved word that is being scanned for, one character per variable. If C, D, E or F are not being used they are sent to SCAN anyway set at 0. They are type INTEGER.

J - If the reserved word was found, J is sent back to the program that called SCAN with the value of the column where the last character being scanned for was found in NCOL. If the reserved word was not found, J is sent back equal to 0.

Alpha-numeric Characters - A - Z and 0 - 9 are represented as variables with Q's between the characters themselves. For example, "K" = QKQ and "5" = Q5Q. Other special variables are as listed:

QAR = "+"	QER = "."
QBR = "-"	QFR = "\$"
QCR = "*"	QGR = "G"
QDR = "="	QHR = " "

All alpha-numeric variables are of type LOGICAL*1.
(Eg. Searching for an equal sign in column 4 is done by the code "IF(NCOL(4).EQ.QDR)...")

G - G is SCAN's COMMON name for NCOL, the "squashed" COL. G does not have any spaces until after the period.

SCAN - SCAN is a variable since the subprogram is an integer function. If the word being scanned for is found, SCAN comes back with a 1; if not, it comes back with a 0.

Description:

SCAN does not call any subroutines. It is called by SUSAN, LITRL, LOGIC, LOGICA, RESCAN, COLUMN and CONDIR. SCAN is an integer function that determines whether or not a reserved word is in NCOL (or G). If it is, SCAN receives a 1 and if not, it receives a 0. The subprogram also states where the last character of the reserved word was found in NCOL. This subprogram is used extensively throughout the assembler program.

LITRL

Important Variables:

V - VVAR and VCON are lists of labels and constants corresponding to each "VALUE" statement found. V acts as a pointer for these queues. If the assembler is looking for the fifth value statement, V will have a value of 5. V is an INTEGER.

NCOL - NCOL is created by SQUASH by taking all spaces of blanks out of COL. This allows the assembler to look at a microcode instruction and expect a semi-ordered field. The array NCOL is dimensioned at 30 and of type INTEGER.

DCOL - If a subscript of NCOL contains a numeric character, the same subscript of DCOL becomes its numeric decimal value. This array is dimensioned at 30 and is an INTEGER.

AC - AC contains the constant following "VALUE" in decimal form. It takes the number represented by the decimal values in DCOL. AC's maximum value is 225.

VCON - VCON contains a list of all the constants from the "VALUE"

statements. VVAR contains the variables corresponding to the constants in VCON. VCON receives its constants from the variable AC. VCON is dimensioned at 80.

C - C is equal to 1 if the line contains the reserved word "CALL" and 0 if it does not.

FVAR - FVAR contains the variable name from a "B =" statement. This corresponds to the "VALUE" statements or labels with the same variable. FVAR is an array dimensioned at 8 and is a LOGICAL*1 type array.

MPAD - MPAD is initialized at 0. MPAD stands for the Memory Program Address Descriptor. Each address contains 8, 12 or 16 bits of information, depending on the value of DEV. This is an INTEGER type incremented by one everytime information is put into the address. MPAD is not incremented for comments, "VALUE" statements or labels but is incremented by 2 for "CALL" and "GOTO" statements if DEV equals 12 or 16.

Y - Y is similar to MPAD. Y is one greater than MPAD so that a DO loop can be used in the program. Y is the subscript of CODE and is an INTEGER type variable.

CODE - CODE contains the information to be sent to the object file. CODE is 256 bytes long and is written over to the object file whenever the 256 bytes are filled or when the program is finished. CODE is dimensioned at 128 and is of type INTEGER. LITRL, LOGIC, and CONDIT write the correct information into CODE.

CON - CON contains the MPAD value of the label in VAR.

VAR - The variable FVAR is compared to the list VAR. If the value in FVAR is found in VAR, the corresponding MPAD value is found in CON and is added to the present value of CODE(Y). It is a LOGICAL*1 type array.

Alpha-numeric Characters - A - Z and 0 - 9 are represented as variables with Q's between the characters themselves. For example, "K" = QKQ and "5" = Q5Q. Other special variables are as listed:

QAR = "+"	QER = "\$"
QBR = "-"	QFR = "\$"
QCR = "*"	QGR = "G"
QDR = "="	QHR = " "

All alpha-numeric variables are of type LOGICAL*1.

Eg. Searching for an equal sign in column 4 is done by the code "IF (NCOL(4).EQ.QDR)..."

Description:

LITRL is called by SUSAN if the reserved words "DEV",

"CALL" or "GOTO" are found in NCOL. LITRL calls SCAN and WRT. The first part of the program handles all "DEV" statements. It calculates the constant to the right of the equal sign and sends the correct code to CODE(Y). The program also checks a microcode instruction to see if it is either a "CALL" or a "GOTO" statement. In RESCAN, VAR is complete so LITRL knows what address the "CALL" or "GOTO" statement is referring to even if it is a forward reference. "CALL" and "GOTO" statements receive 2 address locations and they receive the correct CODE from CON.

RESCAN

Important Variables:

CCT - CCT is the number of records read of the source file ZA. It is an INTEGER variable.

DEV - DEV takes on the value 8, 12 or 16. The way the assembler is now written, it really is not that important. However, the assembler can be programmed to act differently depending on the value of DEV. It can be programmed to work on an 8, 12 or 16 bit machine.

Alpha-Numeric Characters - A - Z and 0 - 9 are represented as variables with Q's between the characters themselves. For example, "K" = QKQ and "5" = Q5Q. Other special variables are as listed:

QAR = "+"	QER = "."
QBR = "-"	QFR = "\$"
QCR = "*"	QGR = "G"
QDR = "="	QHR = " "

All alpha-numeric variables are of type LOGICAL*1.
(Eg. Searching for an equal sign in column four is done by the code "IF (NCOL(4).EQ.QDR)...")

DUP - DUP is initialized at 0. If there is a duplicate label, DUP is set to 1 and an error message is written. DUP is an INTEGER.

VAR - VAR is a list for all labels. VAR is dimensioned at (100, 8). The labels contained in VAR can be only 7 characters long. VAR is a LOGICAL*1 array.

CON - CON contains the MPAD value of the corresponding label found in VAR. CON's dimension is 100 and is an INTEGER variable.

FVAR - FVAR contains the variable name from a "B =" statement. This corresponds to the "VALUE" statement or label with the same variable. FVAR is an array dimensioned at 8 and is a LOGICAL*1 type array.

COL - COL is the "workspace" for the assembler. Each record of the source file is read into COL, one at a time. The assembler looks at this array checking for the various conditions necessary. COL is of type INTEGER and is dimensioned at 80.

MP - MP is RESCAN's MPAD.

Description:

RESCAN is called by SUSAN. RESCAN calls SQUASH. This assembler is a two-pass assembler with the first pass being accomplished by RESCAN. RESCAN is call by SUSAN in order to pre-scan for all labels and label addresses so that VAR and CON are complete with the necessary information before the second pass is made. If there is a new label, that label is put into VAR. All "VALUE" statements must be before any other executable statement. RESCAN lets the user know when the first pass is being performed by relaying the message "WAIT FOR FIRST PASS-SCAN FOR LABELS" to him.

LOGIC

Important Variables:

NCOL - NCOL is created by SQUASH by taking all spaces of blanks out of COL. This allows the assembler to look at a microcode instruction and expect a semi-ordered field. The array NCOL is dimensioned at 30 and of type INTEGER.

CODE - CODE contains the information to be sent to the object file. CODE is 256 bytes long and is written over to the object file whenever the 256 bytes are filled or when the program is finished. CODE is dimensioned at 128 and is of type INTEGER. LITRL, LOGICA and CONDIRIT write the correct information into CODE.

Y - Y is similiar to MPAD. Y is one greater than MPAD so that a DO loop can be used in the program. Y is the subscript of CODE and is an INTEGER type variable.

MPAD - MPAD is initialized at 0. MPAD stands for the Memory Program Address Descriptor. Each address contains 8, 12 or 16 bits of information, depending on the value of DEV. This is an INTEGER type incremented by one everytime information is put into the address. MPAD is not incremented for comments, "VALUE" statements or labels but is incremented by 2 for "CALL" and "GOTO" statements if DEV equals 12 or 16.

Alpha-Numeric Characters - A - Z and 0 - 9 are represented as variables with Q's between the characters themselves. For example, "K" = QKQ and "5" = Q5Q. Other special variables are as listed:

QAR = "+"	QER = "."
QBR = "-"	QFR = "\$"
QCR = "*"	QGR = "G"
QDR = "="	QHR = " "

All alpha-numeric variables are of type LOGICAL*1.

(Eg. Searching for an equal sign in column 4 is done by the code "IF NCOL(4).EQ.QDR)...")

U - U is LOGIC's DCOL. It translates numeric characters into numeric decimal values. It is dimensioned at 30.

COL - COL is the "workspace" for the assembler. Each record of the source file is read into COL, one at a time. The assembler looks at this array checking for the various conditions necessary. COL is of the type INTEGER and is dimensioned at 80.

T - T represents the position of the column that follows the column that contains an "=" sign. This is used extensively to figure out what follows the "=" sign in a statement.

Description:

LOGIC calls SCAN, WRT and LOGICA. It is called by SUSAN (main program) when the reserved words "A1", "A2", "A3", "B=", "BEX", "=" or "ONES" are found. LOGIC, with the help of LOGICA, handles all the syntax required for the reserved words. The program checks for required syntax before the equal sign, finds the equal sign and then checks for the required syntax that follows it. If there are any syntax errors, the error message "FORMAT ERROR - UNDEFINED SEMANTICS" or a more explanatory message will be relayed to the user.

LOGICA

Important Variables:

NCOL - NCOL is created by SQUASH by taking all spaces of blanks out of COL. This allows the assembler to look at a microcode instruction and expect a semi-ordered field. The array NCOL is dimensioned at 30 and of type INTEGER.

CODE - CODE contains the information to be sent to the object file. CODE is 256 bytes long and is written over to the object file whenever the 256 bytes are filled of when the program is finished. CODE is dimensioned at 128 and is of type INTEGER. LITRL, LOGICA and CONDIT write the correct information into CODE.

T - T represents the position of the column that follows the column that contains an "=" sign. This is used extensively to figure out what follows the "=" sign in a statement.

Y - Y is similar to MPAD. Y is one greater than MPAD so that a DO loop can be used in the program. Y is the subscript of CODE and is an INTEGER type variable.

MPAD - MPAD is initialized at 0. MPAD stands for the Memory Program Address Descriptor. Each address contains 8, 12 or 16 bits of information, depending on the value of DEV. This is an INTEGER type incremented by one everytime information is put into the address. MPAD is not incremented for comments, "VALUE" statements or labels but is incremented by two for "CALL" and "GOTO" statements if DEV equals 12 or 16.

Alpha-Numeric Characters - A - Z and 0 - 9 are represented as variables with Q's between the characters themselves. For example,

"K" = QKQ and "5" = Q5Q. Other special variables are as listed:

QAR = "+"	QER = "."
QBR = "-"	QFR = "\$"
QCR = "*"	QGR = "G"
QDR = "="	QHR = " "

All alpha-numeric variables are of type LOGICAL*1.
(Eg. Searching for an equal sign in column 4 is done by the code
"IF (NCOL(4).EQ.QDR)...")

COL - COL is the "workspace" for the assembler. Each record of the source file is read into COL, one at a time. The assembler looks at this array checking for the various conditions necessary. COL is of type INTEGER and is dimensioned at 80.

Description:

LOGICA calls SCAN and WRT. LOGICA is called by LOGIC when the character after the equal sign is an A. It checks for reserved words and characters such as "+", "-", "NOR", and "EQV" that follow the "A1", "A2", "A3" of "AMPCR" that directly follows the equal sign in the microcode program. LOGICA works very similarly to LOGIC, with the correct information going into CODE(Y) when a certain syntax is met.

Task Building:

The RSX11M task builder utility (TKB) is used to build the MDMPL task from the object files and overlay description language file. The following TAB commands are used:

```
TKB [20,20]MDMPL,TSK=FASS.ODL/MP,[1,1]SYSLIB/LB:$SHORT
```

Options:

```
UNITS = 3  
ACTFIL = 3  
ASG = TT1:1, SY0:2:3  
MAXBUF = 256  
EXTSCT = $$FSR1:2264
```

Notes Concerning the MDMP L Assembler Flowchart:

Table A gives the flowchart letter connectors and their corresponding positions in the program. In the program, whenever a search for a reserved word occurred, the subfunction SCAN was called. SCAN sends back the position of the last letter of the reserved word found in NCOL and whether or not the reserved word was actually located. In the flowchart, LOGICA is included with LOGIC. The beginning of LOGICA is located at Z in the flowchart.

TABLE A

FLOWCHART LETTER CONNECTOR	PROGRAM LOCATION (MODULE - LABEL)
A	SUSAN - 30
B	SUSAN - 240
C	RESCAN - 47
D	RESCAN - 44
E	RESCAN - 48
F	SCAN - 1
G	SCAN - 2
H	SCAN - 3
I	SCAN - 4
J	SCAN - 5
K	LITRL - 1
L	LITRL - 11
M	LITRL - 9
N	LITRL - 15
O	COLUMN - 15
P	CONDIT - 15
Q	LOGIC - 100
R	LOGIC - 500
S	LOGIC - 696
T	LOGIC - 305
U	LOGIC - 509
V	LOGIC - 650
W	LOGIC - 600
X	LOGIC - 550
Y	LOGIC - 670
Z	LOGIC - 1000
AA	LOGIC - 695
BB	LOGIC - 690
CC	LOGIC - 2000
DD	LOGIC - the third executable statement after 2010.

```
|_EDI FASS.ODL
|PAGE 1J
*LI
.I: .ROOT SUSAN-*BLOCK-*RESCAN-*I
.J: .FCTR WRT-*(SQUASH,*J)
.K: .FCTR SCAN-*(COLUMN,CONDIT,LITRL,*K)
.FCTR LOGIC-*LOGICA
.END
*ED
[EXIT]
>
```

FLOWCHART OF THE MDMP
ASSEMBLER

SUSAN-MAIN PROGRAM

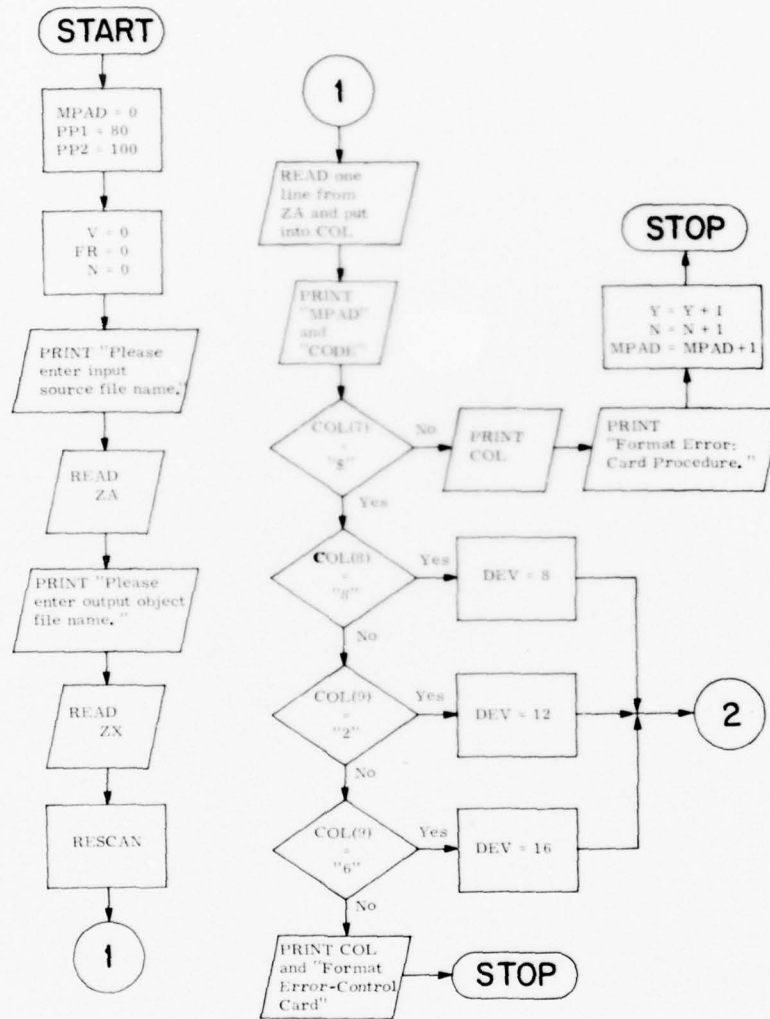


Figure 1-1. SUSAN - Main Program

SUSAN-MAIN PROGRAM (cont.)

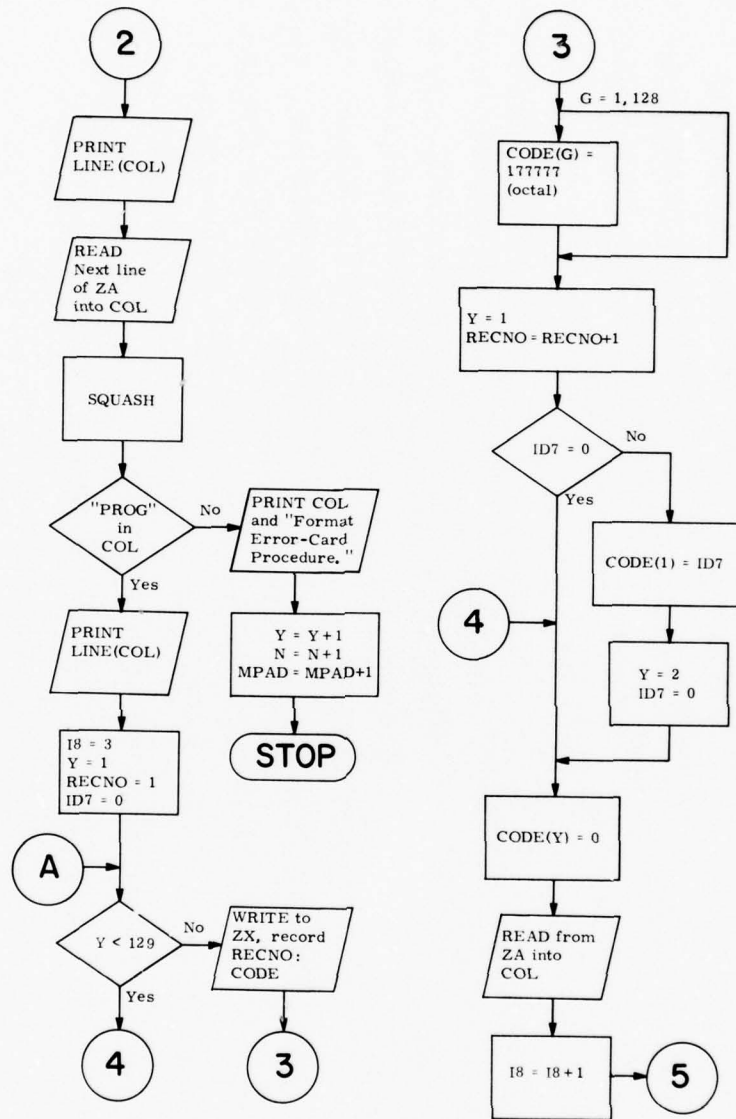


Figure 1-1. (Cont.)

SUSAN-MAIN PROGRAM (cont.)

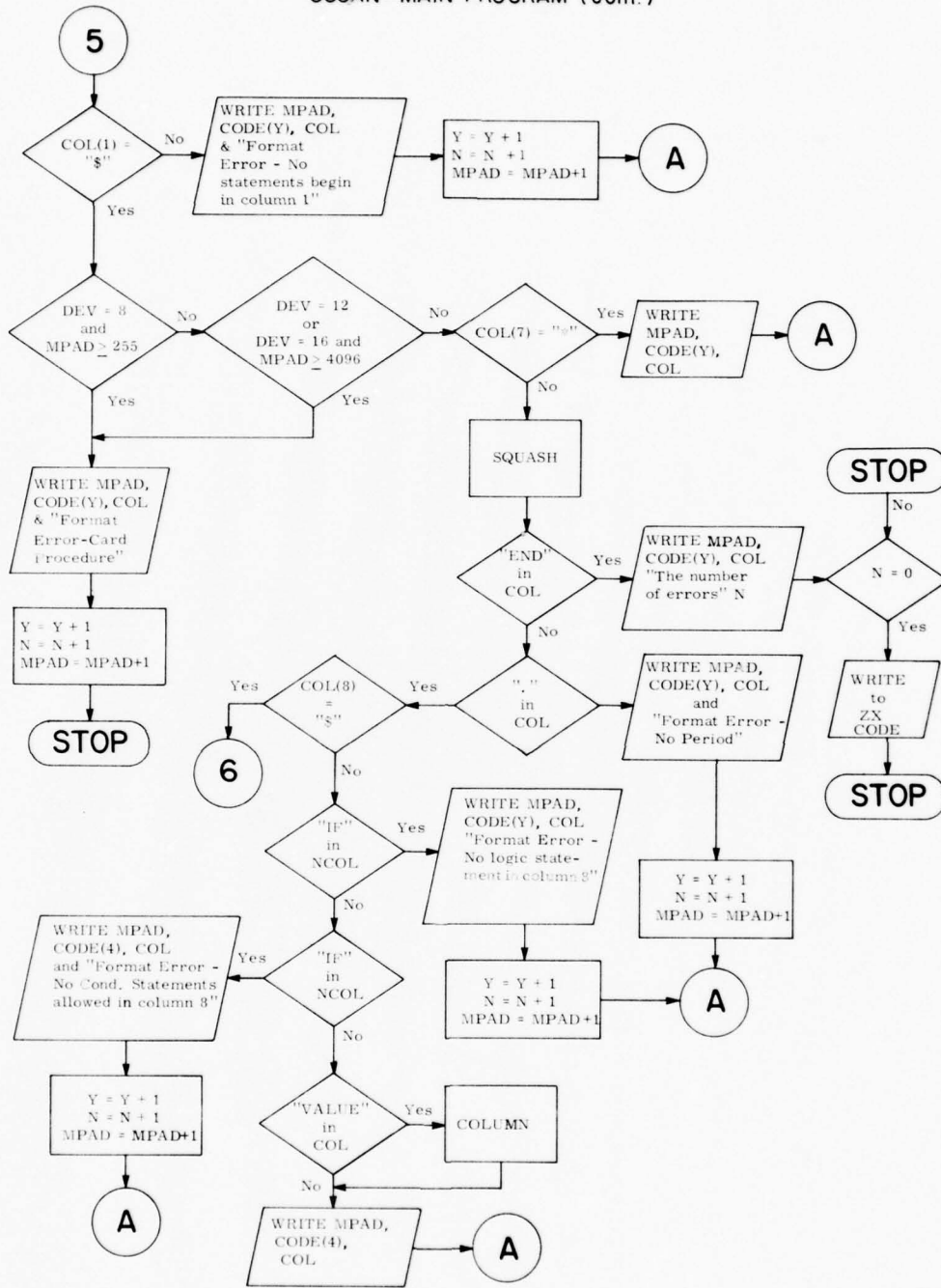


Figure 1-1. (Cont.)

SUSAN-MAIN PROGRAM (cont.)

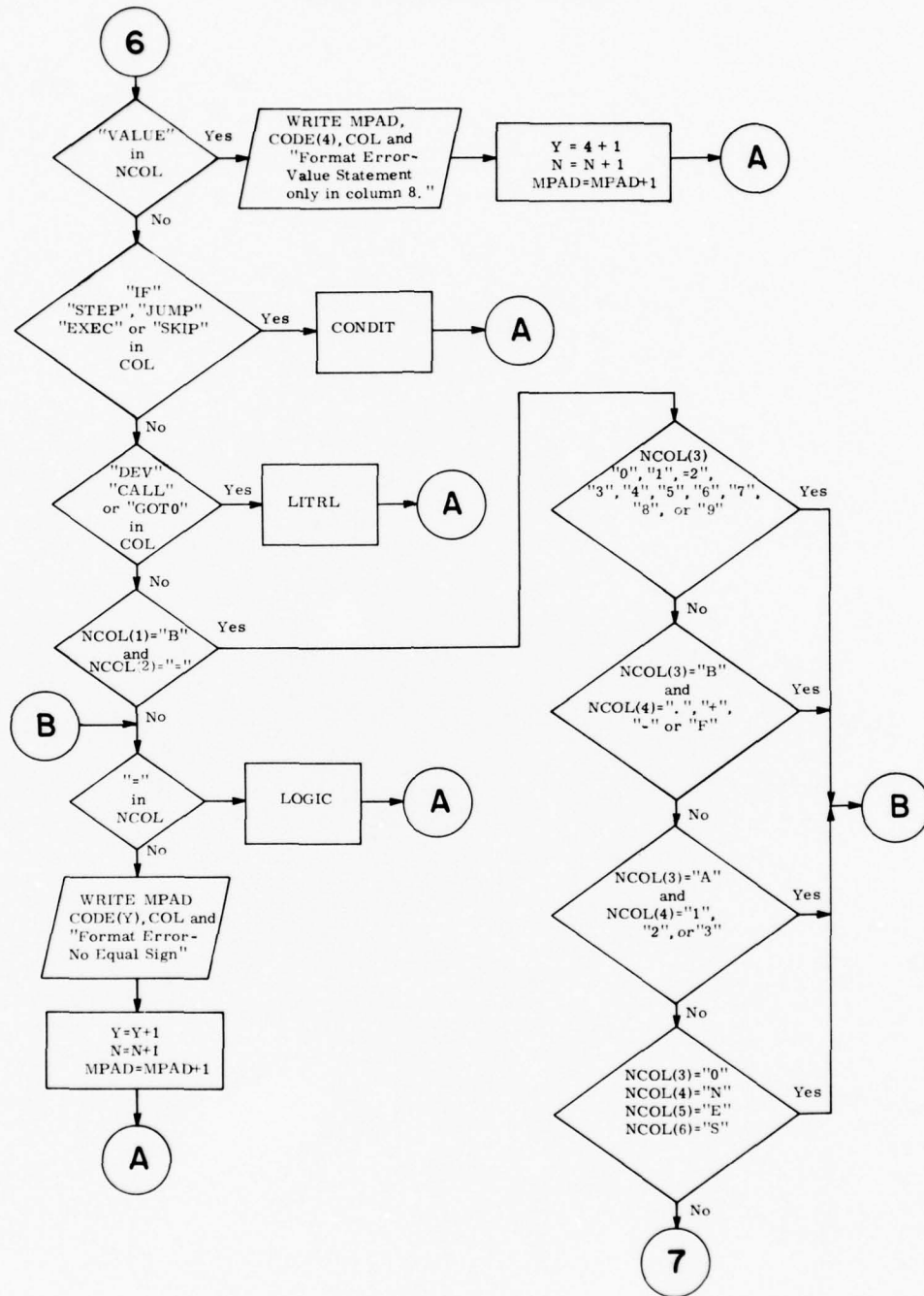


Figure 1-1. (Cont.)

SUSAN-MAIN PROGRAM (cont.)

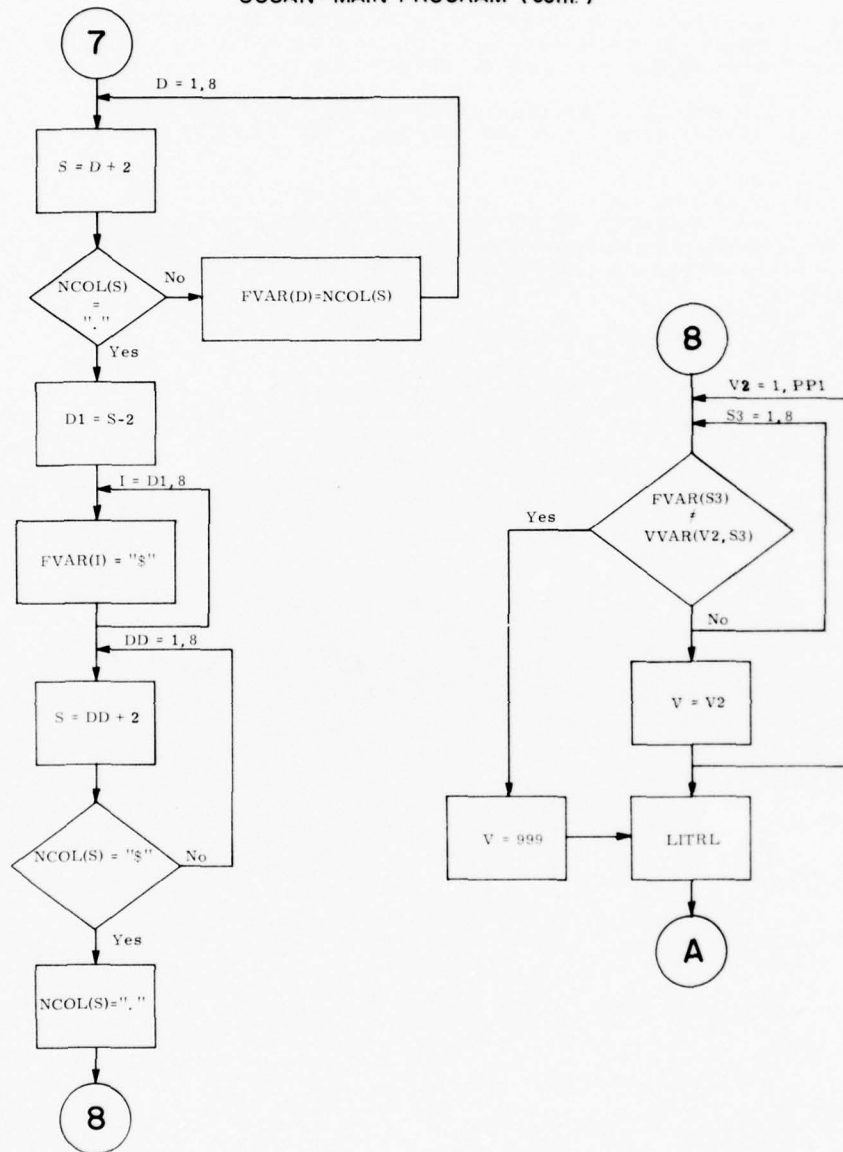


Figure 1-1. (Cont.)

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

SAT 26-FEB-77 17:29:16

PAGE 001

SUSAN.OBJ=SUSAN.FOR/NOSN/LI:1

```

C     SUSAN IS THE NAME OF AN ASSEMBLER MAIN PROGRAM DESIGNED TO
C     TRANSLATE MDMPLE INTO B7* MACHINE CODE. THE PROGRAM WILL BE USED
C     ON THE PDP-11 WHICH ONLY HAS A FORTRAN COMPILER.
C
C     BECAUSE THE PROGRAM ONLY USES INTEGERS, ALL VARIABLES EXCEPT
C     THOSE BEGINNING WITH A 'Z' ARE INTEGERS. THE PROGRAM HAS
C     EIGHT SUBPROGRAMS.
0001  IMPLICIT INTEGER (A-Y)
0002  LOGICAL*1 VVAR,FVAR,VAR
0003  LOGICAL*1 NCOL,QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
1QM, QNQ,QQQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,ROQ,
2Q1Q,Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,
3QGR,QHR,COL
0004  COMMON NCOL(30)
0005  COMMON /VAL/VVAR(80,8),VCON(80)
0006  COMMON /PAR/ PP1,PP2
0007  COMMON /SJI/FVAR(8),VAR(100,8),CON(100)
0008  COMMON /IMP/COL(80),CODE(128)
0009  COMMON /VAX/N,MPAD,FR,Y
0010  REAL*8 ZA,ZX
0011  COMMON /FILE/ZA(3),ZX(3),RECNO
0012  COMMON /CODES/QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
1QM, QNQ,QQQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,ROQ,Q1Q,
2Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,QGR,QHR
0013  COMMON /DSK/ V8,V9
0014  COMMON /SULIT/ID7
C     THE NUMBER OF BITS CONTROL CARD IS READ IN FIRST TO DESIGNATE
C     THE NECESSITY FOR A LIT-TO-IR CODE PRECEDING ALL CALL OR GOTO
C     STATEMENTS. THIS OCCURS FOR A 12 OR 16 BIT MACHINE.
C     THIS CARD ALSO STIPULATES THE MAXIMUM NUMBER OF CODES.
C     FOR THE 8 BIT MACHINE,IT IS 256, AND FOR THE 12 OR 16 BIT
C     MACHINE IT IS 4096.
0015  MPAD=0
0016  PP1=80
0017  PP2=100
0018  V=0
0019  FR=0
0020  N=0
0021  CALL ASSIGN(1,'TT1:')
0022  WRITE(1,11)
0023  11 FORMAT(1X,'PLEASE ENTER INPUT SOURCE FILE NAME')
0024  READ(1,12) ZA
0025  12 FORMAT(3A8)
0026  WRITE(1,13)
0027  13 FORMAT(1X,'PLEASE ENTER OUTPUT OBJECT FILE NAME')
0028  READ (1,12) ZX
0029  CALL ASSIGN(2,ZA)
0030  DEFINE FILE 2(3000,40,U,V8)
0031  CALL ASSIGN(3,ZX)
0032  DEFINE FILE 3(32,128,U,V9)
C     FIRST PRESCAN FOR LABEL ADDRESSES.
0033  CALL RESCAN(DEV)
C     NOW RETURN TO PRIMARY SCAN
0034  READ (2'1,ERR=999) COL
C     CREATE HEADINGS
0035  WRITE(1,1)

```

FORTRAN IV V01B-02
 CORE=08K, UIC=[20,20]

SAT 26-FEB-77 17:29:16

PAGE 002

SUSAN.OBJ=SUSAN.FOR/NOSN/LI:1

```

0036 1   FORMAT(' MPAD',SX,'CODE',/)
0037   IF (COL(7) .EQ. QFR) GO TO 5
0039   CALL WRT(0,1)
0040   CALL WRT(10,5)
0041   GO TO 9999
0042 5   IF (COL(8) .NE. Q8Q) GO TO 6
0044   DEV=8
0045   GO TO 10
0046 6   IF (COL(9) .NE. Q2Q) GO TO 7
0048   DEV=12
0049   GO TO 10
0050 7   IF (COL(9) .NE. Q6Q) GO TO 8
0052   DEV=16
0053   GO TO 10
0054 8   CALL WRT(0,1)
0055   CALL WRT(15,5)
0056   GO TO 9999
0057 10  CALL WRT (0,1)
      C   SCAN THE PROGRAM-ID CARD
0058   READ (2'2,ERR=999)COL
      C   SQUASH CREATES NCOL
0059   CALL SQUASH
0060   X=SCAN(5,QPQ,QRQ,QQQ,QGQ,QRQ,P)
0061   IF (X .EQ.1)GO TO 20
0063   CALL WRT(0,1)
0064   CALL WRT(10,5)
0065   GO TO 9999
0066 20  CALL WRT(0,1)
0067   IB=3
0068   Y=1
0069   RECNO=1
0070   ID7=0
      C   Y CORRESPONDS TO THE MPAD VALUE +1
0071   30 CONTINUE
0072   IF (Y .LT. 129) GOTO 300
      C   ELSE WRITE CODE TO DISK
0074   WRITE(3'RECNO)CODE
0075   DO 400 G=1,128
0076   400 CODE(G)='177777
0077   Y=1
0078   RECNO=RECNO+1
0079   IF (ID7 .EQ. 0) GOTO 300
      C   ELSE GOTO OR CALL AT BOUNDARY
0081   CODE(1)=ID7
0082   Y=2
0083   ID7=0
0084   300 CONTINUE
0085   CODE(Y)=0
0086   READ (2'IB,ERR=999)COL
0087   IB=IB+1
0088   IF (COL(1) .EQ. QHR) GO TO 32
0090   CALL WRT(43,2)
0091   GO TO 30
0092 32  IF (DEV.EQ.8.AND. MPAD.GE. 255) GO TO 35
0094   IF ((DEV.EQ.12.OR.DEV.EQ.16).AND.MPAD.GE.4096) GO TO 35
0096   GO TO 37

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

SAT 26-FEB-77 17:29:16

PAGE 003

SUSAN.OBJ=SUSAN.FOR/NOSN/LI:1

```
0097 35 CALL WRT(10,2)
0098 GO TO 9999
C IF*, THE CARD IS ONLY WRITTEN
0099 37 IF (COL(7) .NE.QCR) GO TO 50
0101 CALL WRT(0,1)
0102 GO TO 30
0103 50 CALL SQUASH
C CHECK THE $END CARD
0104 IF(NCOL(1).EQ.QEQ.AND.NCOL(2).EQ.QNQ.AND.
1NCOL(3).EQ.QDQ.AND.NCOL(4).EQ.QGR) GO TO 55
GO TO 60
0106 55 CALL WRT(0,1)
0108 CALL WRT(0,3)
0109 IF(N .EQ.0) GO TO 57
0111 GO TO 9999
0112 57 WRITE(3'RECNO) CODE
0113 GO TO 9999
C CHECK FOR A PERIOD
0114 60 X=SCAN(1,QER,0,0,0,0,P)
0115 IF (X .EQ.1) GO TO 70
0117 CALL WRT(0,1)
0118 CALL WRT(17,5)
0119 GO TO 30
C CHECK FOR A VALUE OR LABEL STATEMENT
0120 70 IF (COL(8) .EQ.QHR) GO TO 100
0122 XX=SCAN(1,QDR,0,0,0,0,P)
0123 IF (XX.NE.1) GO TO 80
0125 CALL WRT(37,2)
0126 GO TO 30
0127 80 IF (NCOL(1).NE.QIQ.OR.NCOL(2).NE.QFQ) GO TO 90
0129 CALL WRT(42,2)
0130 GO TO 30
0131 90 G=SCAN(5,QVQ,QAQ,QLQ,QUQ,REQ,P)
0132 IF (G .EQ. 1) CALL COLUMN(V,L)
0134 CALL WRT(0,1)
0135 GO TO 30
C CHECK FOR A CONDITIONAL STATEMENT
0136 100 XL=SCAN(5,QVQ,QAQ,QLQ,QUQ,REQ,P)
0137 IF (XL.NE.1) GO TO 105
0139 CALL WRT(41,2)
0140 GO TO 30
0141 105 F=SCAN(2,QIQ,QFQ,0,0,0,P)
0142 F1=SCAN(4,QSQ,QTQ,REQ,QFQ,0,P)
0143 F2=SCAN(4,QJQ,QUQ,QHQ,QFQ,0,P)
0144 F3=SCAN(4,REQ,QXQ,REQ,QCQ,0,P)
0145 F4=SCAN(4,QSQ,QKQ,RIQ,QFQ,0,P)
0146 IF (F.EQ.1.OR.F1.EQ.1.OR.F2.EQ.1.OR.F3.EQ.1.OR.
1F4.EQ.1) GO TO 110
GO TO 120
0148 110 CALL CONDIR
0149 GO TO 30
0151 120 G=SCAN (3,QDQ,REQ,QVQ,0,0,P)
0152 G1=SCAN(4,QCQ,QAQ,QLQ,QLQ,0,P)
0153 G2=SCAN(4,QGQ,QQQ,QTQ,QQQ,0,P)
0154 IF (G.EQ.1.OR.G1.EQ.1.OR.G2.EQ.1) GO TO 230
0156 IF (NCOL(1) .EQ. QBQ .AND.NCOL(2) .EQ.QDR) GO TO 150
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

SAT 26-FEB-77 17:29:16

PAGE 004

SUSAN.OBJ=SUSAN.FOR/NOSN/LI:1

```

0158      GO TO 240
0159  150  IF (NCOL(3).EQ.Q0Q.OR.NCOL(3).EQ.Q1Q.OR.NCOL(3).EQ.Q2Q
          1.OR.NCOL(3).EQ.Q3Q.OR.NCOL(3).EQ.Q4Q.OR.NCOL(3).EQ.Q5Q
          2.OR.NCOL(3).EQ.Q6Q.OR.NCOL(3).EQ.Q7Q.OR.NCOL(3).EQ.Q8Q
          3.OR.NCOL(3).EQ.Q9Q) GO TO 240
0161      IF (NCOL(3).EQ.QBQ.AND.(NCOL(4).EQ.QER.OR.NCOL(4).EQ.QAR
          1.OR.NCOL(4).EQ.QBR.OR.NCOL(4).EQ.QFQ)) GO TO 240
0163      IF (NCOL(3).EQ.QAQ.AND.(NCOL(4).EQ.Q1Q.OR.NCOL(4).EQ.Q2Q
          1.OR.NCOL(4).EQ.Q3Q)) GO TO 240
0165      IF (NCOL(3).EQ.Q0Q.AND.NCOL(4).EQ.QNQ.AND.NCOL(5).EQ.QEQ
          1.AND.NCOL(6).EQ.QSQ) GO TO 240
0167      DO 210 D=1,8
0168          S=D+2
0169          IF (NCOL(S).EQ.QER) GO TO 212
0171  210  FVAR(D)=NCOL(S)
0172  212  D1=S-2
0173          DO 213 I=D1,8
0174  213  FVAR(I)=QHR
0175          DO 215 DD=1,8
0176          S=DD+2
0177          IF (NCOL(S).EQ.QHR) GO TO 217
0179  215  CONTINUE
0180  217  NCOL(S)=QER
0181          DO 225 V2=1,PF1
0182          DO 220 S3=1,8
0183          IF (FVAR(S3).NE.VVAR(V2,S3)) GO TO 225
0185  220  CONTINUE
0186          V=V2
0187          GO TO 230
0188  225  CONTINUE
0189          V=999
0190  230  CALL LITRL(V,DEV)
0191          GO TO 30
          C
          CHECK FOR LOGIC STATEMENTS
0192  240  XV=SCAN(1,QDR,0,0,0,0,P)
0193          IF (XV.EQ.1) GO TO 250
0195          CALL WRT(36,2)
0196          GO TO 30
0197  250  CALL LOGIC
0198          GO TO 30
0199  999  N=N+1
0200          WRITE (1,998)
0201  998  FORMAT(' ERROR IN A READ STATEMENT')
0202          GO TO 30
0203  9999 STOP
0204          END

```

> 9

FORTRAN IV V01B-02 MON 28-FEB-77 08:59:37 PAGE 001
 CORE=08K, UIC=[20,20] BLOCK.OBJ=BLOCK.FOR/NOSN/LI:1

```

0001      BLOCK DATA
0002      LOGICAL*1 VVAR,FVAR,VAR
0003      COMMON /VAL/VVAR(80,8),VCON(80)
0004      COMMON /SJI/FVAR(8),VAR(100,8),CON(100)
0005      COMMON /VAX/N,MPAD,FR,Y
0006      LOGICAL*1 NCOL,QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
1QM, QNQ,QQQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,QQQ,
2Q1Q,Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,
3QGR,QHR,COL
0007      COMMON NCOL(30)
0008      COMMON /PAR/PP1,PP2
0009      COMMON /IMP/COL(80),CODE(128)
0010      COMMON /CODES/QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
1QM, QNQ,QQQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,QQQ,Q1Q,
2Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,QGR,QHR
0011      DATA QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,QMQ,QNQ,QQQ,
1QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,Q0Q,Q1Q,Q2Q,Q3Q,Q4Q,
2Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,QGR,QHR/'A','B','C',
3'D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S',
4'T','U','V','W','X','Y','Z','0','1','2','3','4','5','6','7','8',
5'9','+','-','*','=','.', '$','?', ' /
0012      DIMENSION C(30),DCOL(30),WZ(30),XZ(30)
0013      END

```

FORTRAN IV DIAGNOSTICS

[WARNING] MSG #094 NON-STANDARD STATEMENT ORDERING

FOR -- [.MAIN.] ERRORS: 0, WARNINGS: 1
 >

RESCAN

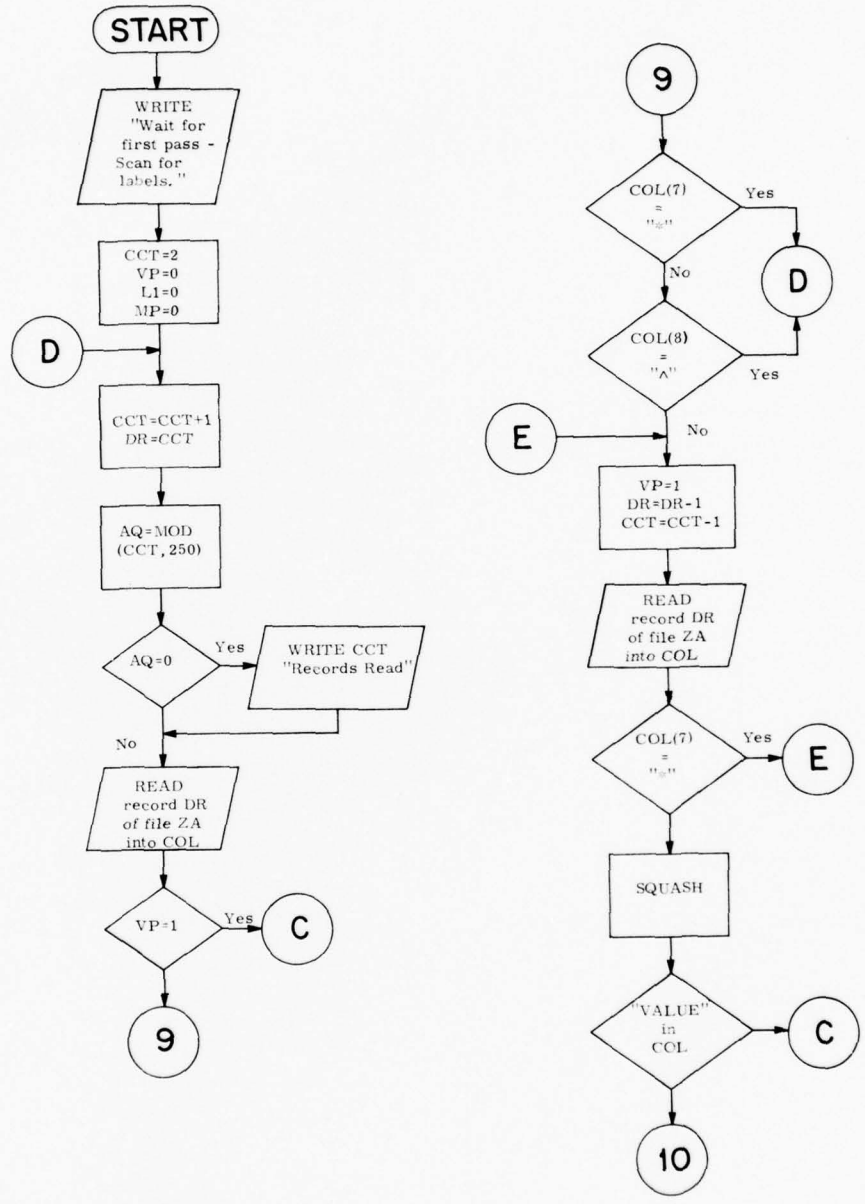


Figure 1-2. RESCAN

RESCAN (cont.)

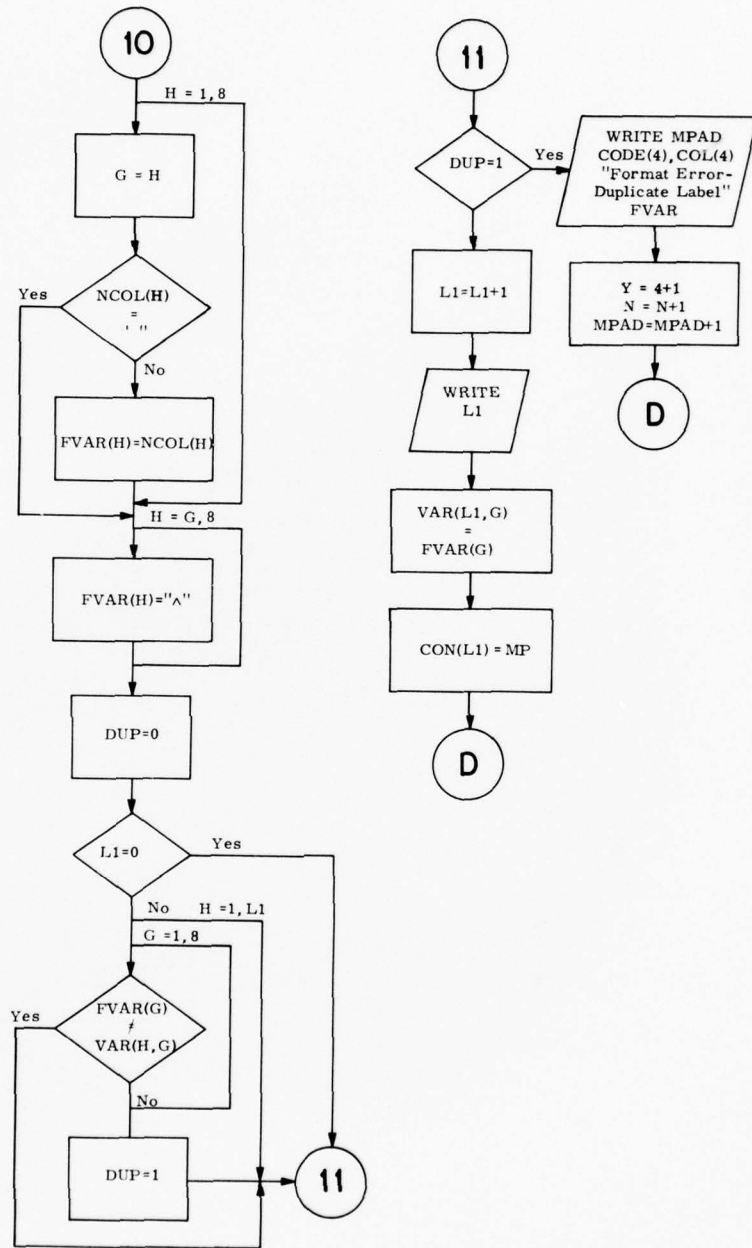


Figure 1-2. (Cont.)

RESCAN (cont.)

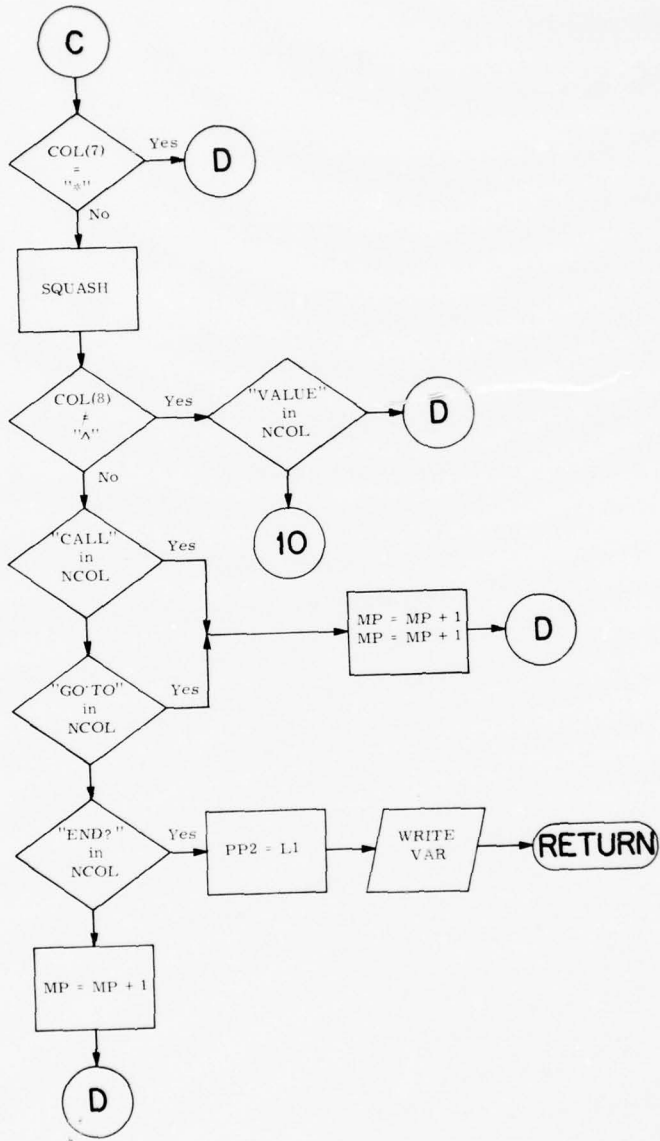


Figure 1-2. (Cont.)

FORTRAN IV V01B-02
CORE=08K, UIC=C20,20J

MON 28-FEB-77 09:02:05

PAGE 001

RESCAN.OBJ=RESCAN.FOR/NOSN/LI:1

```

0001      SUBROUTINE RESCAN(DEV)
0002      IMPLICIT INTEGER (A-Y)
0003      LOGICAL*1 UVAR,FVAR,VAR
0004      LOGICAL*1 NCOL, COL, RAQ, RBQ, RCQ, RDQ, REQ, RFQ, RGQ, RHQ, RIQ, RQJ, RQK,
1QLQ, QMQ, QNQ, QOQ, QPQ, QQQ, QRQ, QSQ, QTQ, QUQ, QVQ, QWQ, QXQ, QYQ, QZQ,
2QQQ, Q1Q, Q2Q, Q3Q, Q4Q, Q5Q, Q6Q, Q7Q, Q8Q, Q9Q, QAR, QBR, QCR, QDR, QER,
3QFR, QGR, QHR
0005      COMMON NCOL(30)
0006      COMMON /SJI/FVAR(8),VAR(100,8),CON(100)
0007      COMMON /VAL/ UVAR(80,8),VCON(80)
0008      COMMON /IMP/COL(80),CODE(128)
0009      COMMON /VAX/N,MPAD,FR,Y
0010      REAL*8 ZA,ZX
0011      COMMON /FILE/ZA(3),ZX(3),RECND
0012      COMMON /FAR/PP1,FP2
0013      COMMON /CODES/RAQ, RBQ, RCQ, RDQ, REQ, RFQ, RGQ, RHQ, RIQ, RQJ, RQK, RLQ,
1QMQ, QNQ, QOQ, QPQ, QQQ, QRQ, QSQ, QTQ, QUQ, QVQ, QWQ, QXQ, QYQ, QZQ, QOQ, Q1Q,
2Q2Q, Q3Q, Q4Q, Q5Q, Q6Q, Q7Q, Q8Q, Q9Q, QAR, QBR, QCR, QDR, QER, QFR, QGR,
3QHR
0014      COMMON /DSK/ V8,V9
C      FRESCAN FOR LABELS AND LABEL ADDRESSES.
0015      WRITE(1,43)
0016      43 FORMAT(' WAIT FOR FIRST PASS - SCAN FOR LABELS')
0017      CCT=2
0018      VP=0
0019      L1=0
0020      MP=0
C      READ SOURCE RECORD INTO COL ARRAY
0021      44 CCT=CCT+1
0022      DR=CCT
0023      AQ=MOD(CCT,250)
0024      IF (AQ .EQ. 0) WRITE(1,50) CCT
0026      50 FORMAT(1X,I5,' RECORDS READ')
0027      READ(2'DR,ERR=999) COL
C      TEST FOR END OF VALUE DECLARATIONS
0028      IF (VP .EQ. 1) GO TO 47
C      CHECK FOR COMMENT
0030      IF (COL(7) .EQ. QCR) GO TO 44
0032      IF (COL(8) .NE. QHR) GO TO 44
C      FIRST STATEMENT FOUND. SET UP FLAG
0034      VP=1
C      NOW BACK UP TO FIND POSSIBLE LABEL
0035      48 DR=DR-1
0036      CCT=CCT-1
0037      READ(2'DR,ERR=999) COL
0038      IF (COL(7) .EQ. QCR) GO TO 48
0040      CALL SQUASH
0041      H=SCAN(5,QVQ,QAQ,QLQ,QUQ,REQ,G)
0042      IF (H .EQ. 1) GO TO 47
C      LABEL AT MPAD=0
C      FOLLOWING MAKES LABEL REFERENCE IN VAR AND CON.
0044      400 DO 403 H=1,8
0045          G=H
0046          IF (NCOL(H) .EQ. QER) GO TO 404
0048      403 FVAR(H)=NCOL(H)
0049      404 DO 410 H=G,8

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:02:05

PAGE 002

RESCAN.OBJ=RESCAN.FOR/NOSN/LI:1

```

0050 410 FVAR(H)=QHR
0051     DUP=0
      C   TEST FOR DUPLICATE LABEL.
0052     IF (L1 .EQ. 0) GO TO 417
0054     DO 415 H=1,L1
0055     DO 416 G=1,8
0056 416 IF (FVAR(G) .NE. VAR(H,G)) GO TO 415
0058     DUP=1
0059     GO TO 417
0060 415 CONTINUE
0061 417 IF (DUP .EQ.1) GO TO 430
      C   LABEL IS NEW
0063     L1=L1+1
      D   WRITE(1,32) L1
      D 32 FORMAT(1X,'L1= ',I3)
0064     DO 420 G=1,8
0065 420 VAR(L1,G)=FVAR(G)
0066     CON(L1)=MP
0067     GO TO 44
      C   IF LABEL IS DUPLICATED, WRITE ERROR.
0068 430 N=N+1
0069     WRITE(1,431) FVAR
0070 431 FORMAT(' FORMAT ERROR - DUPLICATE LABEL -- ',8A1)
0071     GO TO 44
      C   NOW ADD MPAD VALUES 1 OR 2 TO MP.
      C   ELIMINATE COMMENTS
0072 47 IF (COL(7) .EQ. QCR) GOTO 44
0074     CALL SQUASH
      D   WRITE(1,31)(NCOL(I),I=1,8)
      D 31 FORMAT(1X,'NCOL= ',A8)
      C   FIND LABELS OR VALUES
0075     IF (COL(8) .NE. QHR) GOTO 480
0077     G=SCAN(4,QCQ,QAQ,QLQ,QLQ,0,P)
      C   FIND CALL
0078     IF (G .EQ. 1) GOTO 440
0080     G=SCAN(4,QGQ,QQQ,QTQ,QQQ,0,P)
      C   FIND GOTO
0081     IF (G .EQ. 1) GOTO 440
0083     G=SCAN(4,REQ,QNQ,QDQ,QGR,0,P)
      C   FIND END
0084     IF (G .EQ. 1) GO TO 490
0086     MP=MP+ 1
0087     GOTO 44
0088 440 MP=MP+1
0089     MP=MP+1
0090     GOTO 44
0091 999 WRITE(1,998)
0092 998 FORMAT(1X,'SOURCE DISK READ ERROR')
0093 480 G=SCAN(5,QVQ,QAQ,QLQ,QUQ,REQ,P)
0094     IF (G .EQ. 1) GOTO 44
0096     GOTO 400
0097 490 PP2=L1
      D   WRITE(1,30)((VAR(I,J),J=1,8),I=1,PP2)
      D 30 FORMAT(1X,80A1)
0098     RETURN
0099     END

```

SCAN

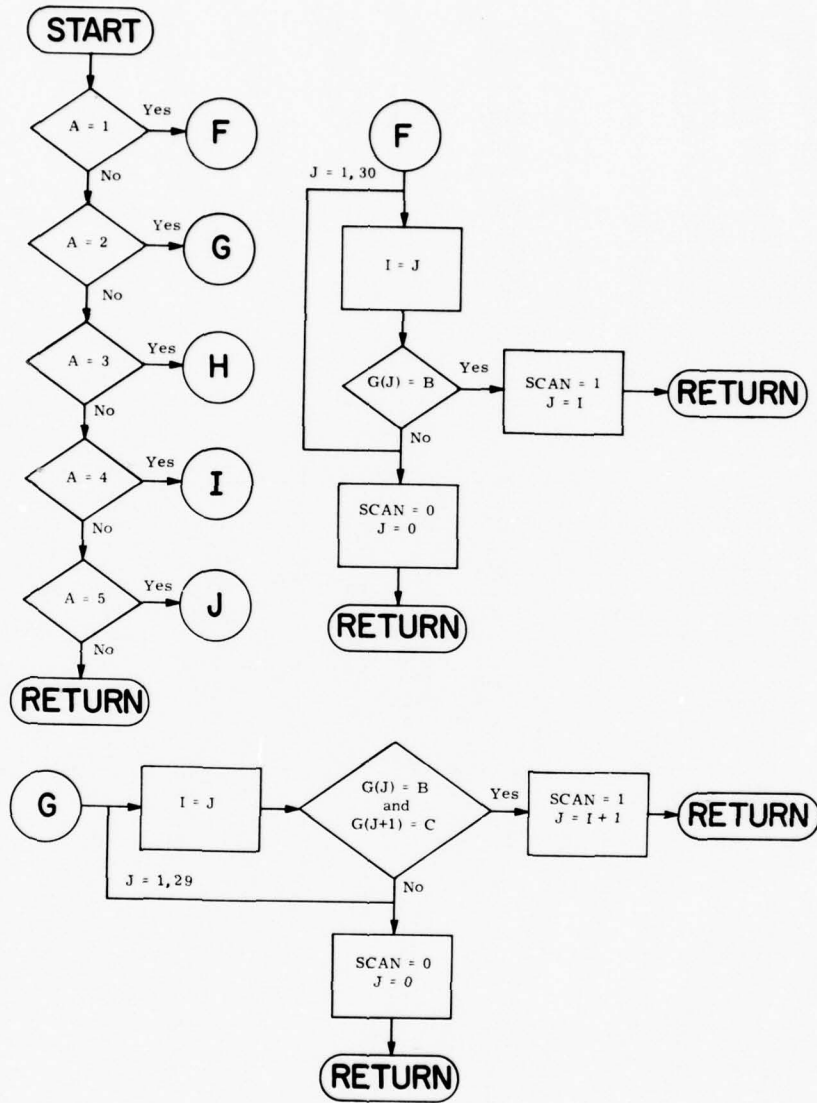


Figure 1-3. SCAN

SCAN (cont.)

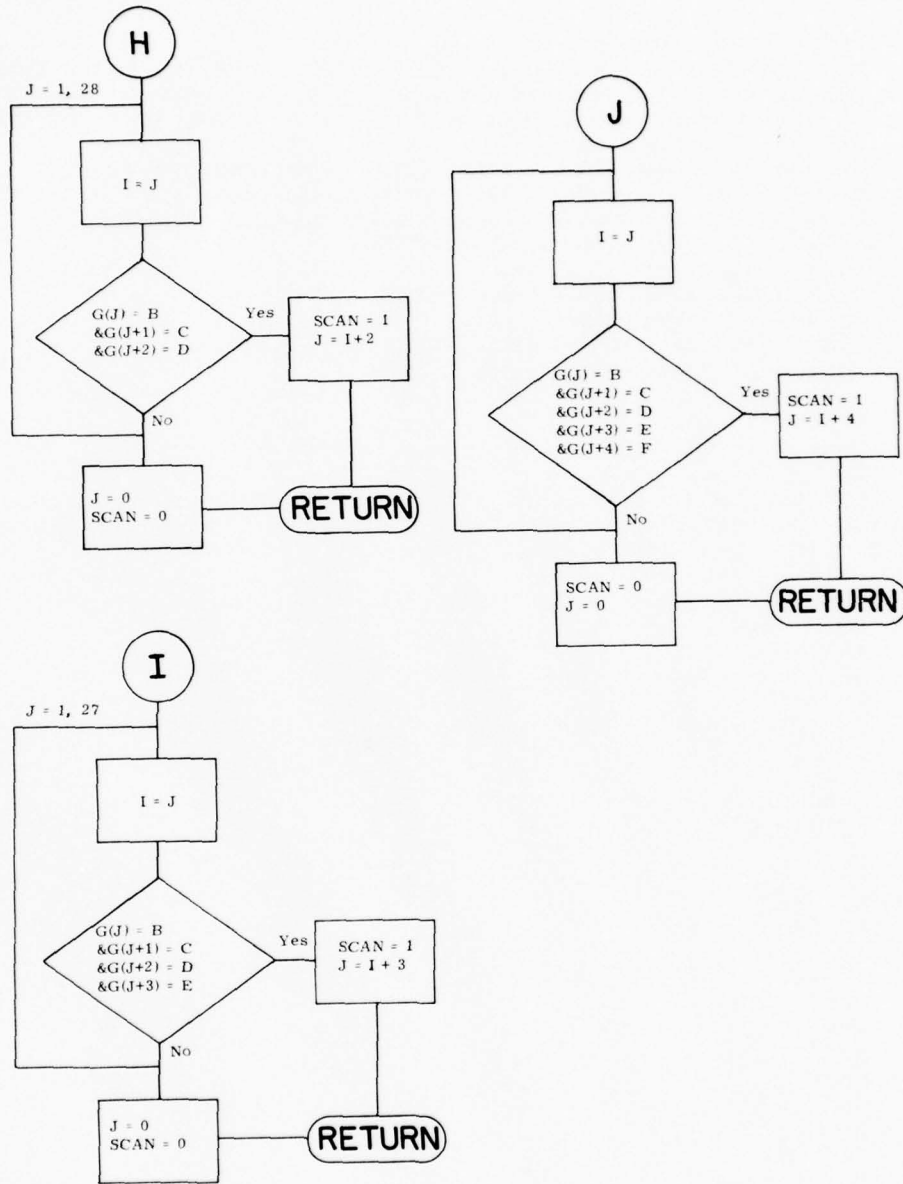


Figure 1-3. (Cont.)

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

WED 22-DEC-76 08:31:37

PAGE 001
SCAN7=SCAN7.FOR/LI:1

```

0001      INTEGER FUNCTION SCAN(A,B,C,D,E,F,J)
          C      SCAN IS USED TO CHECK FOR RESERVED WORDS OF THE MDMPL LANGUAGE
          C      DEPENDING ON THE VALUE OF A,SCAN WILL CHECK FOR A CHARACTERS
          C      IF SCAN COMES BACK WITH A VALUE OF 1,THEN THE TEST WAS SUCCESSFUL
0002      IMPLICIT INTEGER (A-Y)
0003      LOGICAL*1 G,QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
          1QM,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,ROQ,Q1Q,
          2Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,QGR,
          3QHR,B,C,D,E,F,COL,VVAR,FVAR,VAR
0004      COMMON G(30)
0005      COMMON /VAL/VVAR(80,8),VCON(80)
0006      COMMON /FAR/FF1,FF2
0007      COMMON /SJI/FVAR(8),VAR(100,8),CON(100)
0008      COMMON /IMP/COL(80),CODE(128)
0009      COMMON /VAX/ N,MPAD,FR,Y
0010      REAL*8 ZA,ZX
0011      COMMON /FILE/ZA(3),ZX(3),RECNO
0012      COMMON /DISK/V8,V9
0013      COMMON /CODES/QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
          1QM,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,ROQ,Q1Q,
          2Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,QGR,QHR
          D      WRITE (1,60)A,B,C,D,E,F
          D60     FORMAT (1X,'A=',I1,' B=',A1,' C=',A1,' D=',A1,' E=',A1,' F=',A1)
0014      GOTO (1,2,3,4,5) A
0015      RETURN
0016      1      DO 10 J=1,30
0017             I=J
0018             IF (G(J) .EQ. B) GOTO 15
0020      10     CONTINUE
0021             J=0
0022             SCAN=0
0023             RETURN
0024      15     SCAN=1
0025             J=I
0026             RETURN
0027      2      DO 20 J=1,29
0028             I=J
0029             IF (G(J).EQ.B.AND.G(J+1) .EQ. C) GOTO 25
0031      20     CONTINUE
0032             J=0
0033             SCAN=0
0034             RETURN
0035      25     SCAN=1
0036             J=I+1
0037             RETURN
0038      3      DO 30 J=1,28
0039             I=J
0040             IF (G(J).EQ.B.AND.G(J+1).EQ.C.AND.G(J+2).EQ.D) GOTO 35
0042      30     CONTINUE
0043             J=0
0044             SCAN=0
0045             RETURN
0046      35     SCAN=1
0047             J=I+2
0048             RETURN
0049      4      DO 40 J=1,27

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

WED 22-DEC-76 08:31:37

PAGE 002
SCAN7=SCAN7.FOR/LI:1

```
0050      I=J
0051      IF (G(J).EQ.B.AND.G(J+1).EQ.C.AND.G(J+2).EQ.D.AND.G(J+3).EQ.E)
          1GOTO 45
0053  40   CONTINUE
0054      J=0
0055      SCAN=0
0056      RETURN
0057  45   SCAN=1
0058      J=I+3
0059      RETURN
0060  5    DO 50 J=1,26
0061      I=J
0062      IF (G(J).EQ.B.AND.G(J+1).EQ.C.AND.G(J+2).EQ.D.AND.G(J+3).EQ.E
          1.AND.G(J+4).EQ.F) GOTO 55
0064  50   CONTINUE
0065      J=0
0066      SCAN=0
0067      RETURN
0068  55   SCAN=1
0069      J=I+4
0070      RETURN
0071      END
```

FOR>

FORTRAN IV V01B-02
CORE=08K, UIC=L20,20J

MON 28-FEB-77 09:05:44

PAGE 001

WRT.OBJ=WRT.FOR/NDSN/LI:1

```

0001      SUBROUTINE WRT(VARF,PCK)
0002      IMPLICIT INTEGER (A-D)
0003      IMPLICIT INTEGER (G-Y)
0004      LOGICAL*1 COL,NCOL,QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,
1QLQ,QMQ,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,
2QOQ,Q1Q,Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,
3QER,QFR,QGR,QHR
0005      LOGICAL*1 VVAR,FVAR,VAR
0006      COMMON /IMP/COL(80),CODE(128)
0007      COMMON NCOL(30)
0008      COMMON /VAL/ VVAR(80,8),VCON(80)
0009      COMMON /PAR/PP1,PP2
0010      COMMON /SJI/ FVAR(8),VAR(100,8),CON(100)
0011      REAL*8 ZA,ZX
0012      COMMON /FILE/ZA(3),ZX(3),RECNO
0013      COMMON /CODES/QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
1QM, QNQ, QOQ, QPQ, QQQ, QRQ, QSQ, QTQ, QUQ, QVQ, QWQ, QXQ, QYQ, QZQ, QOQ, Q1Q,
2Q2Q, Q3Q, Q4Q, Q5Q, Q6Q, Q7Q, Q8Q, Q9Q, QAR, QBR, QCR, QDR, QER, QFR, QGR, QHR
0014      COMMON /VAX/N,MPAD,FR,Y
0015      COMMON /DSK/ V8,V9
0016      GO TO (1,2,3,4,5) ,PCK
          C      WRITE CARD ONLY
0017 1      WRITE (1,9)(COL(I),I=1,80)
0018 9      FORMAT(20X,80A1)
0019      RETURN
          C      WRITE ADDRESS, CODE, CARD AND ERROR MESSAGE
0020 2      CODE(Y)=32767
0021      WRITE (1,6)MPAD, CODE(Y), (COL(I), I=1,80)
0022 6      FORMAT(1X,05,4X,05,5X,80A1)
          C      PRINT ERRORS CORRESPONDING TO VARF
0023      N=N+1
0024      MPAD=MPAD+1
0025      Y=Y+1
0026 5      IF (VARF .NE. 10) GO TO 12
0028      WRITE (1,11)
0029 11      FORMAT(' FORMAT ERROR-CARD PROCEDURE' )
0030      RETURN
0031 12      IF (VARF .NE.15) GO TO 50
0033      WRITE (1,13)
0034 13      FORMAT(' FORMAT ERROR-CONTROL CARD' )
0035      RETURN
0036 50      IF (VARF .NE. 16) GO TO 60
0038      WRITE (1,55)
0039 55      FORMAT(' FORMAT ERROR-OVERFLOW' )
0040      RETURN
0041 60      IF (VARF .NE. 17) GO TO 70
0043      WRITE (1,65)
0044 65      FORMAT (' FORMAT ERROR- NO PERIOD' )
0045      RETURN
0046 70      IF (VARF .NE. 18) GO TO 80
0048      WRITE (1,75)
0049 75      FORMAT(' FORMAT ERROR- MISCELLANEOUS' )
0050      RETURN
0051 80      IF (VARF .NE. 19) GO TO 87
0053      WRITE (1,85)
0054 85      FORMAT (' FORMAT ERROR- NO CONDITION SELECT' )

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:05:44

PAGE 002

WRT.OBJ=WRT.FOR/NOSN/LI:1

```
0055            RETURN
0056 87        IF (VARF .NE.20) GO TO 90
0058            WRITE (1,88)
0059 88        FORMAT(' FORMAT ERROR-INVALID CHARACTER' )
0060            RETURN
0061 90        IF (VARF .NE. 21) GO TO 100
0063            WRITE (1,95)
0064 95        FORMAT(' FORMAT ERROR- NO TRUE SUCCESSOR' )
0065            RETURN
0066 100       IF (VARF .NE. 23) GO TO 110
0068            WRITE (1,105)
0069 105       FORMAT(' FORMAT ERROR- UNDEFINED VALUE CONSTANT' )
0070            RETURN
0071 110       IF (VARF .NE. 24) GO TO 120
0073            WRITE (1,115)
0074 115       FORMAT (' FORMAT ERROR- MISSING DIGIT AFTER DEV' )
0075            RETURN
0076 120       IF (VARF .NE. 25) GO TO 130
0078            WRITE (1,125)
0079 125       FORMAT (' FORMAT ERROR- MISSING DIGIT AFTER LC' )
0080            RETURN
0081 130       IF (VARF .NE. 26) GO TO 140
0083            WRITE (1,135)
0084 135       FORMAT (' FORMAT ERROR- MISSING DIGIT AFTER BEX' )
0085            RETURN
0086 140       IF (VARF .NE. 31) GO TO 150
0088            WRITE (1,145)
0089 145       FORMAT(' FORMAT ERROR- MISSING DIGIT AFTER OUT' )
0090            RETURN
0091 150       IF (VARF .NE. 32) GO TO 160
0093            WRITE (1,155)
0094 155       FORMAT(' FORMAT ERROR- NO DESTINATION SELECT' )
0095            RETURN
0096 160       IF (VARF .NE. 33) GO TO 170
0098            WRITE (1,165)
0099 165       FORMAT (' FORMAT ERROR- MISSING DIGIT AFTER A' )
0100            RETURN
0101 170       IF (VARF .NE. 34) GO TO 180
0103            WRITE (1,175)
0104 175       FORMAT (' FORMAT ERROR- UNDEFINED SEMANTICS' )
0105            RETURN
0106 180       IF (VARF .NE. 35) GO TO 186
0108            WRITE (1,185)
0109 185       FORMAT (' FORMAT ERROR- UNDEFINED OPERATION' )
0110            RETURN
0111 186       IF (VARF .NE. 36) GO TO 188
0113            WRITE(1,187)
0114 187       FORMAT (' FORMAT ERROR- NO EQUAL SIGN' )
0115            RETURN
0116 188       IF (VARF .NE. 37) GO TO 190
0118            WRITE (1,189)
0119 189       FORMAT(' FORMAT ERROR- NO LOGIC STATEMENT IN COLUMN 8' )
0120            RETURN
0121 190       IF (VARF .NE. 40) GO TO 196
0123            WRITE (1,195)
0124 195       FORMAT (' FORMAT ERROR- VALUE CONSTANT OVERFLOW' )
```

FORTRAN IV V01B-02
CORE=08K, UIC=C20,20J

MON 28-FEB-77 09:05:44

PAGE 003

WRT.OBJ=WRT.FOR/NOSN/LI:1

```
0125            RETURN
0126 196        IF (VARF .NE. 41) GO TO 198
0128            WRITE (1,197)
0129 197        FORMAT(' FORMAT ERROR- VALUE STATEMENT ONLY IN COLUMN 8' )
0130            RETURN
0131 198        IF (VARF .NE. 42) GO TO 220
0133            WRITE (1,199)
0134 199        FORMAT(' FORMAT ERROR- NO CONDITION STATEMENTS IN COLUMN 8' )
0135            RETURN
0136 220        IF (VARF .NE. 43) GO TO 200
0138            WRITE (1,225)
0139 225        FORMAT(' FORMAT ERROR- NO STATEMENTS BEGIN IN COLUMN 1' )
0140            RETURN
0141 200        IF (VARF .NE. 45) GO TO 210
0143            WRITE (1,205)
0144 205        FORMAT(' FORMAT ERROR- NO FALSE SUCCESSOR' )
0145            RETURN
0146 210        WRITE(1,250)VARF
0147 250        FORMAT(1X,016,' IS NOT A VALID NUMBER FOR VARF' )
0148            RETURN
          C        LAST WRITE-WRITE THE NUMBER OF ERRORS
0149 3           WRITE (1,7)N
0150 7           FORMAT(' THE NUMBER OF ERRORS=' ,I2)
0151            RETURN
          C        WRITE THE ADDRESS AND CODE WITH THE CARD
0152 4           WRITE (1,8)MPAD,CODE(Y),(COL(I),I=1,80)
0153 8           FORMAT (1X,05,4X,05,5X,80A1)
0154            MPAD=MPAD+1
0155            Y=Y+1
0156            RETURN
0157            END
```

>

SQUASH

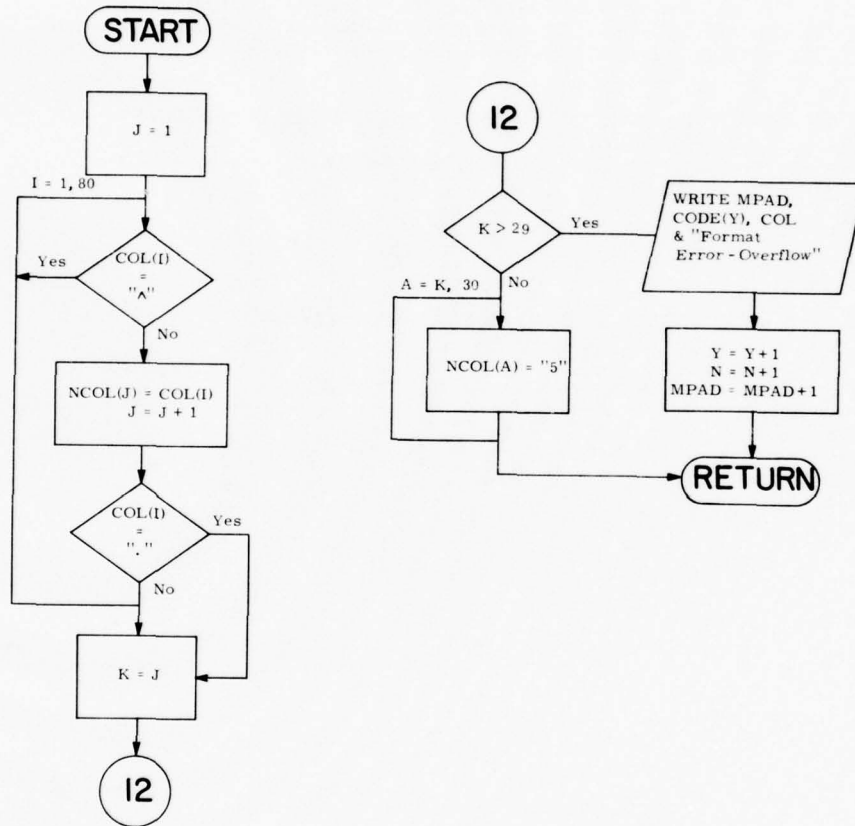


Figure 1-4. SQUASH

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:10:49

PAGE 001

SQUASH.OBJ=SQUASH.FOR/NOSN/LI:1

```
0001 SUBROUTINE SQUASH
0002 IMPLICIT INTEGER (A-Y)
0003 LOGICAL*1 COL,NCOL,QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,
1QLQ,QMQ,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,
200Q,Q1Q,Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,
3QFR,QGR,QHR
0004 COMMON NCOL(30)
0005 COMMON /IMP/COL(80),CODE(128)
0006 COMMON /CODES/QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
1QM,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,Q0Q,Q1Q,
2Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,QGR,QHR
0007 J=1
0008 DO 20 I=1,80
0009 C THIS CHECKS FOR A SPACE,IF THERE IS ONE, IT WILL IGNORE IT
IF (COL(I).EQ.QHR) GOTO 20
0010 C THIS ASSIGNS THE CHARACTER TO THE NEW ARRAY
0011 NCOL(J)=COL(I)
0012 J=J+1
0013 C EVERYTHING AFTER A PERIOD IS IGNORED
IF (COL(I).EQ.QER) GOTO 30
0015 20 CONTINUE
0016 30 K=J
0017 C THIS IS A CHECK TO SEE THAT THE STATEMENT BEFORE
C THE PERIOD IS NOT TOO LONG
IF(K.GT.29)GOTO 60
0018 C THIS IGNORES EVERYTHING AFTER THE FIRST PERIOD
0019 DO 50 A=K,30
0020 50 NCOL(A)=QHR
0021 RETURN
0022 60 CALL WRT(16,2)
0023 RETURN
0024 END
```

>

COLUMN

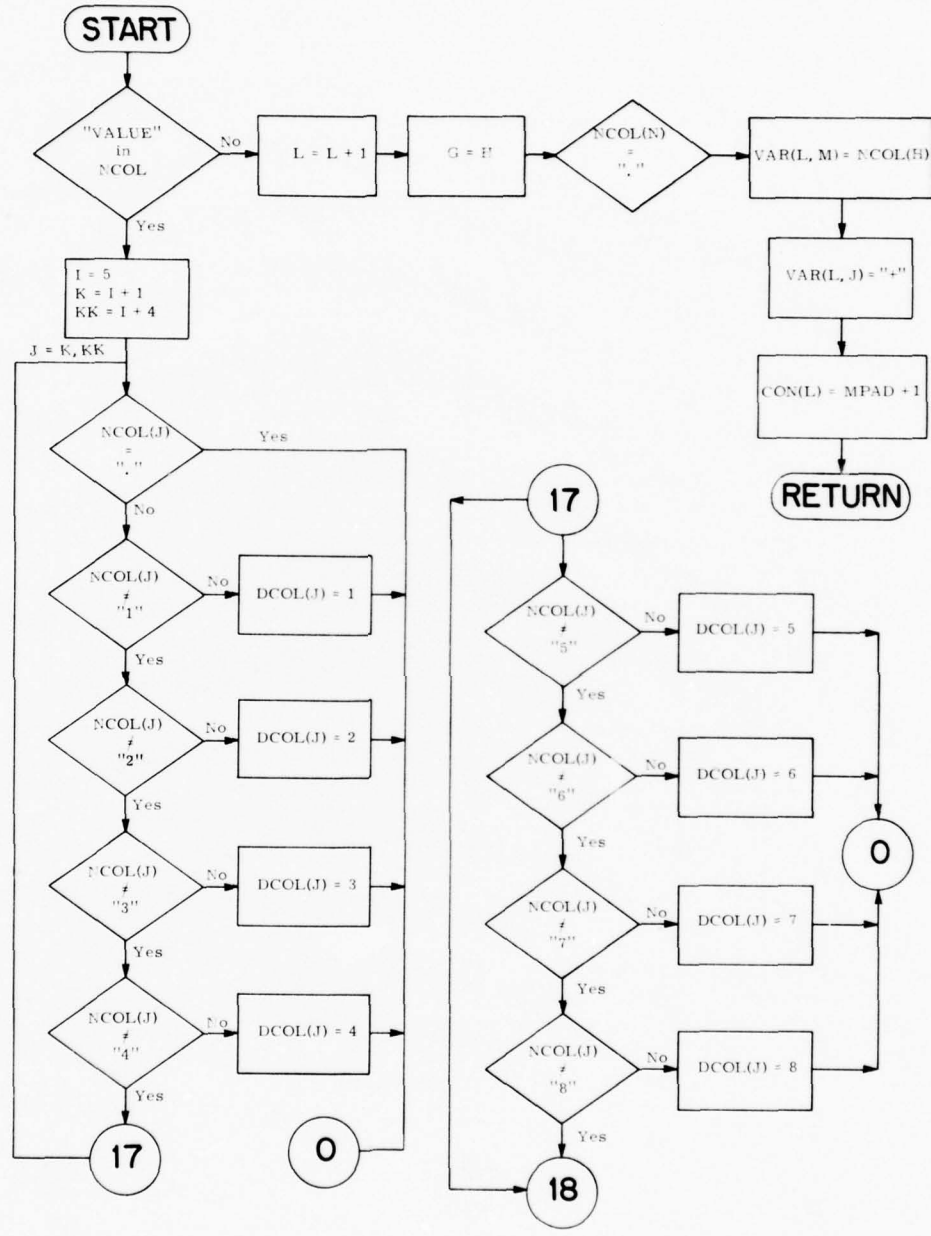


Figure 1-5. COLUMN

COLUMN (cont.)

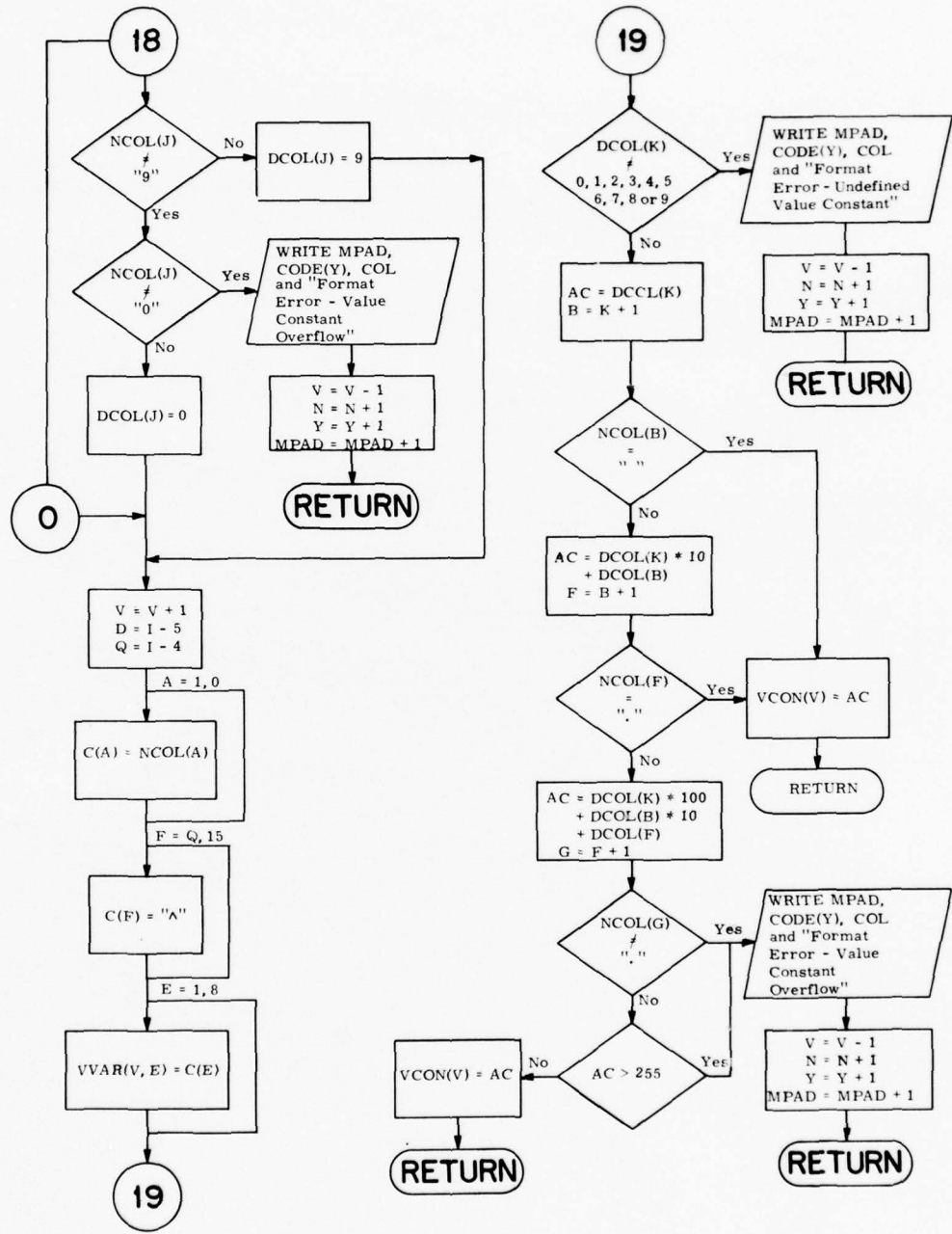


Figure 1-5. (Cont.)

FORTRAN IV V01B-02 MON 28-FEB-77 09:13:00 PAGE 001
 CORE=08K, UIC=[20,20] COLUMN.OBJ=COLUMN.FOR/NOSN/LI:1

```

0001      SUBROUTINE COLUMN(V,L)
0002      IMPLICIT INTEGER (A-Y)
0003      LOGICAL*1 VVAR,FVAR,VAR,C
0004      LOGICAL*1 COL,NCOL,QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,
1QLQ,QMQ,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,
2QOQ,Q1Q,Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,
3QFR,QGR,QHR

0005      COMMON NCOL(30)
0006      COMMON /VAL/ VVAR(80,8),VCON(80)
0007      COMMON /SJI/ FVAR(8),VAR(100,8),CON(100)
0008      COMMON /IMP/ COL(80),CODE(128)
0009      COMMON /VAX/N,MFAD,FR,Y
0010      COMMON /CODES/QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
1QMQ,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,QOQ,Q1Q,
2Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,QGR,QHR

0011      DIMENSION C(15),DCOL(30)
          C      CHECK TO SEE IF IT IS A VALUE STATEMENT
0012      X=SCAN(5,QVQ,QAQ,QLQ,QUQ,QEQ,I)
0013      IF (X .EQ. 0) GOTO 70
          C      CHANGE FROM HOLLERITH TO DECIMAL
0015      K=I+1
0016      KK=I+4
0017      DO 15 J=K,KK
0018      IF (NCOL(J).EQ.QER) GOTO 16
0020      IF (NCOL(J) .NE. Q1Q) GOTO 6
0022      DCOL(J)=1
0023      GOTO 15
0024    6      IF (NCOL(J).NE.Q2Q) GOTO 7
0026      DCOL(J)=2
0027      GOTO 15
0028    7      IF (NCOL(J).NE. Q3Q) GOTO 8
0030      DCOL(J)=3
0031      GOTO 15
0032    8      IF (NCOL(J).NE. Q4Q) GOTO 9
0034      DCOL(J)=4
0035      GOTO 15
0036    9      IF (NCOL(J) .NE. Q5Q) GOTO 11
0038      DCOL(J)=5
0039      GOTO 15
0040    11      IF (NCOL(J) .NE. Q6Q) GOTO 12
0042      DCOL(J)=6
0043      GOTO 15
0044    12      IF (NCOL(J).NE.Q7Q) GOTO 13
0046      DCOL(J)=7
0047      GOTO 15
0048    13      IF (NCOL(J) .NE. Q8Q) GOTO 14
0050      DCOL(J)=8
0051      GOTO 15
0052    14      IF (NCOL(J).NE. Q9Q) GOTO 17
0054      DCOL(J)=9
0055      GOTO 15
0056    17      IF (NCOL(J) .NE.QOQ) GOTO 60
0058      DCOL(J)=0
0059    15      CONTINUE
          C      ADVANCE LOOP NUMBER BY 1
0060    16      V=V+1

```

FORTRAN IV V01B-02
CORE=08K, UIC=C20,20J

MON 28-FEB-77 09:13:00
COLUMN.OBJ=COLUMN.FOR/NOSN/LI:1

PAGE 002

```

      C      FIND VARIABLE BEFORE VALUE STATEMENT
0061      D      D=I-5
0062      Q      Q=I-4
0063      DO 10 A=1,D
0064 10      C(A)=NCOL(A)
0065      DO 20 F=Q,15
0066 20      C(F)=QHR
0067      DO 30 E=1,8
0068 30      VVAR(V,E)=C(E)
      D      WRITE (1,31)(VVAR(V,E),E=1,8)
      D31     FORMAT (' VVAR=',A8)
      C      FIND CONSTANT AFTER VALUE STATEMENT
0069      IF(DCOL(K).NE.0.AND.DCOL(K).NE.1.AND.DCOL(K).NE.2.AND.DCOL(K).NE.3
1.AND.DCOL(K).NE.4.AND.DCOL(K).NE.5.AND.DCOL(K).NE.6.AND.DCOL(K)
2.NE.7.AND.DCOL(K).NE.8.AND.DCOL(K).NE.9)GOTO 50
0071      AC=DCOL(K)
0072      B=B+1
0073      IF (NCOL(B).EQ. QER) GOTO 40
0075      AC=DCOL(K)*10+DCOL(B)
0076      F=B+1
0077      IF (NCOL(F).EQ.QER) GOTO 40
0079      AC=DCOL(K)*100+DCOL(B)+DCOL(F)
0080      G=F+1
      C      THESE ARE OVERFLOW CHECKS
0081      IF (NCOL(G).NE.QER)GOTO 60
0083      IF (AC.GT.255) GOTO 60
      C      THIS WILL ASSIGN THE CONSTANT TO THE PROPER ARRAY
0085 40      VCON(V)=AC
      C      THIS WILL WRITE THE CARD
0086      RETURN
      C      THIS WILL WRITE THE OVERFLOW ERROR CHECKED FOR ABOVE
0087 50      CALL WRT(0,1)
0088      CALL WRT(23,5)
      C      BECAUSE OF AN ERROR,THE VALUE LOOP NUMBER IS SUBTRACTED BY 1
0089      V=V-1
0090      RETURN
0091 60      CALL WRT(0,1)
0092      CALL WRT(40,5)
0093      V=V-1
0094      RETURN
      C      THIS IS THE LABEL LOOP NUMBER ADDER
0095 70      L=L+1
      C      THIS WILL ASSIGN THE VARIABLE INTO THE PROPER ARRAY
0096      DO 80 H=1,8
0097      G=H
0098      IF (NCOL(H) .EQ. QER) GOTO 83
0100 80      VAR(L,H)=NCOL(H)
0101 83      DO 84 J=G,8
0102 84      VAR(L,J)=QHR
      C      THIS WILL ASSIGN THE CONSTANT TO THE CORRESPONDING VARIABLE
0103      CON(L)=MPAD+1
0104      RETURN
0105      END

```

>

CONDIT

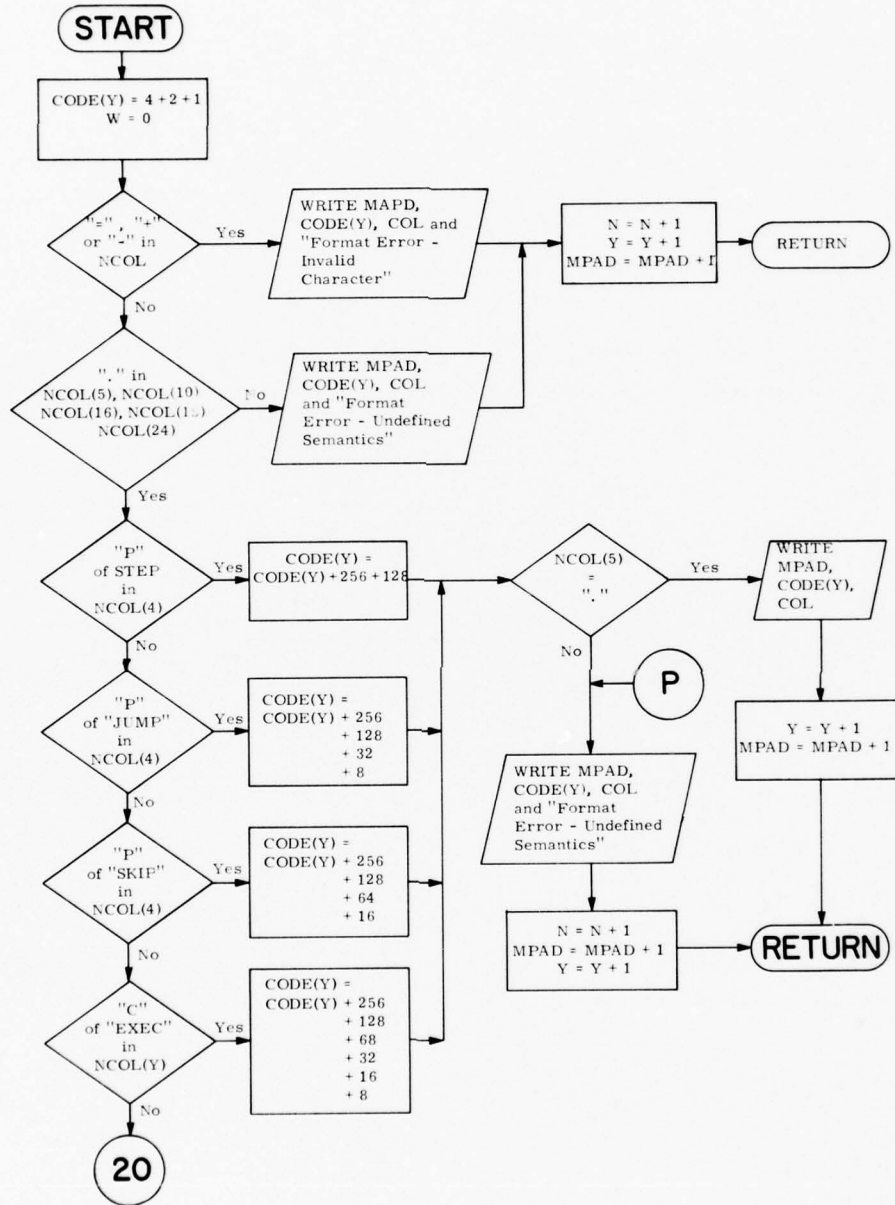


Figure 1-6. CONDIT

CONDIT (cont.)

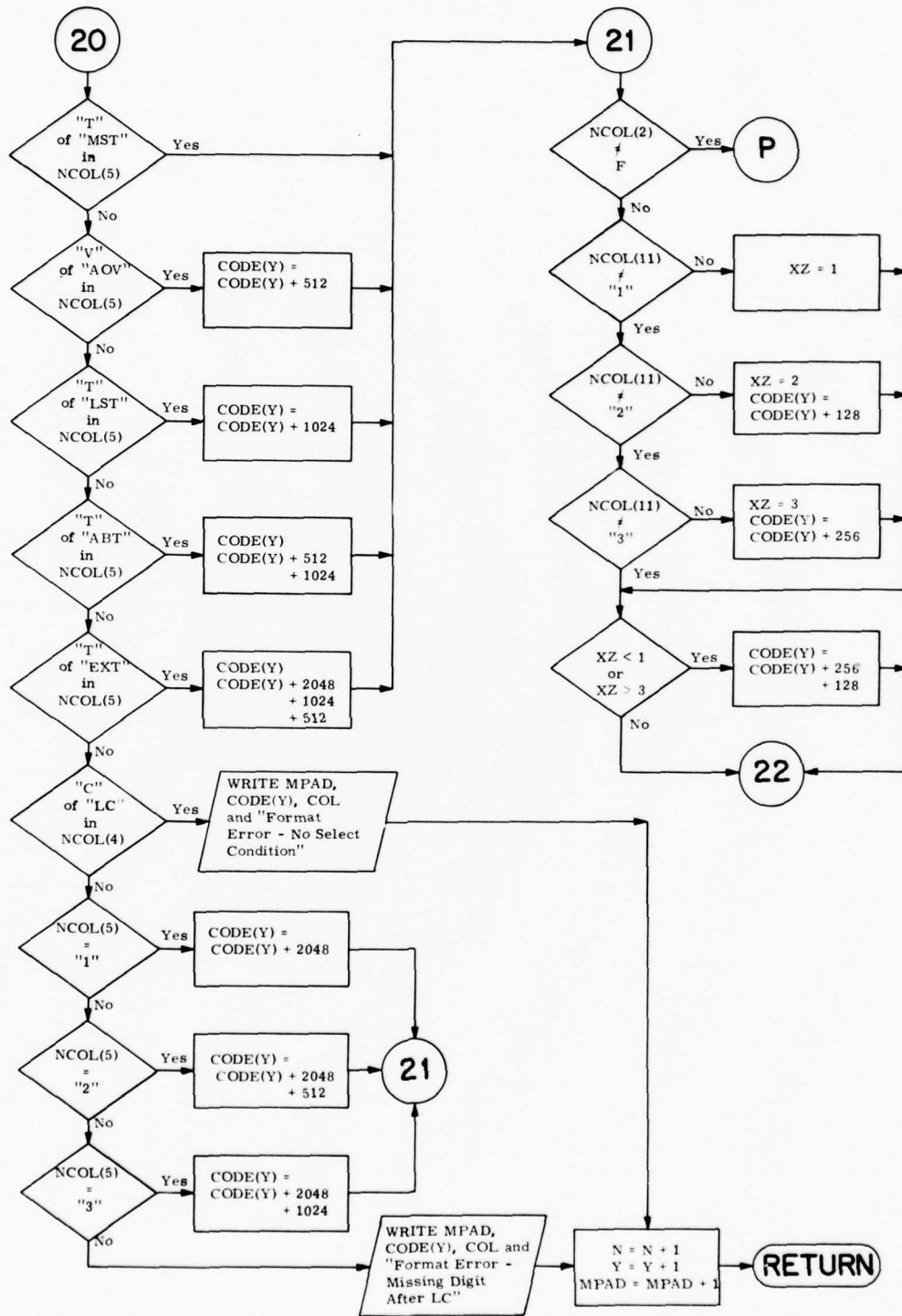


Figure 1-6. (Cont.)

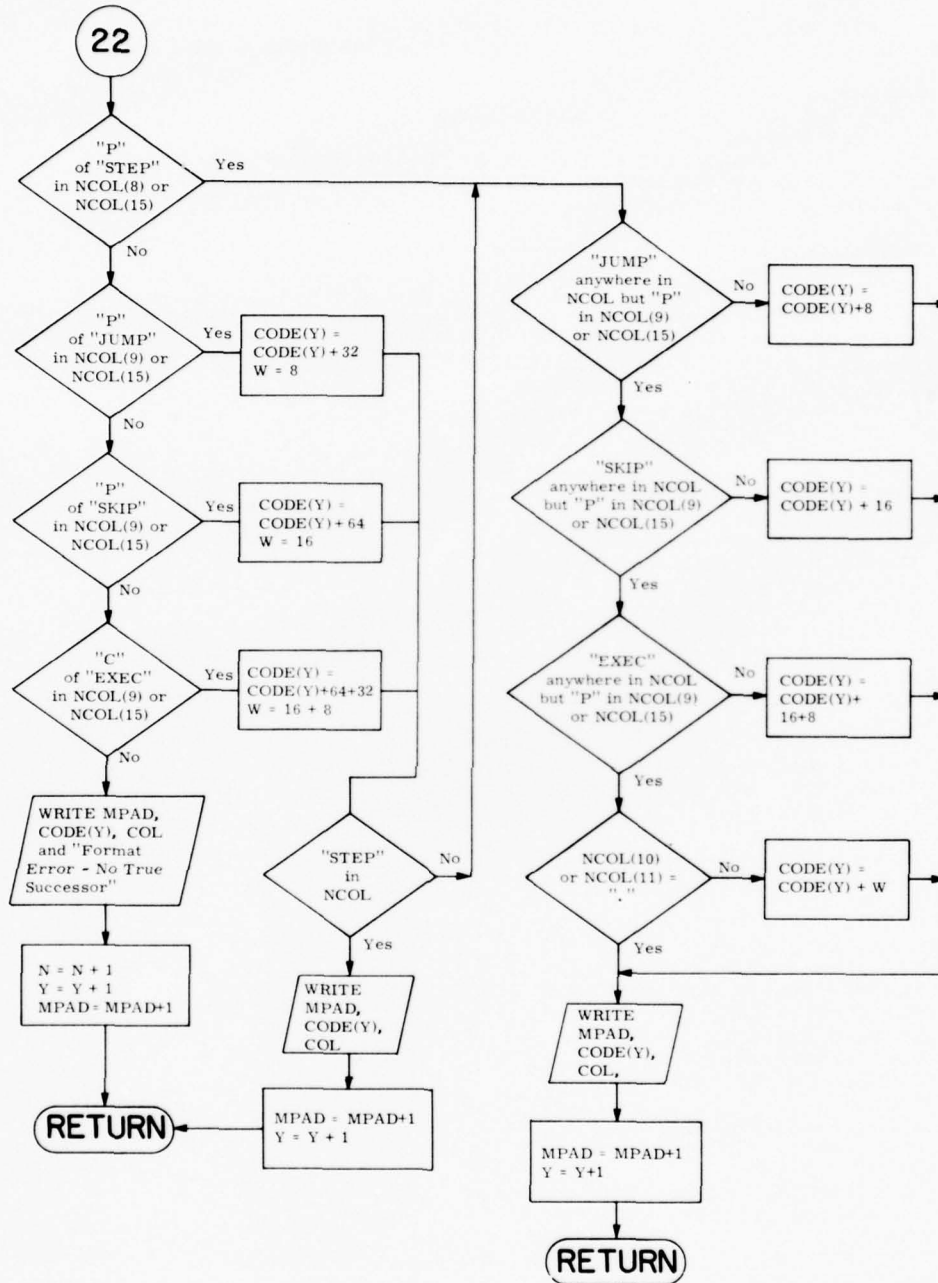


Figure 1-6. (Cont.)

FORTRAN IV VO1B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:16:45 PAGE 001
CONDIT.OBJ=CONDIT.FOR/NOSN/LI:1

```

0001      SUBROUTINE CONDIT
          C      CONDIT CHECKS FOR THE CONDITION TESTS
0002      IMPLICIT INTEGER (A-Y)
0003      LOGICAL*1 COL,NCOL,QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,
          1QLQ,QMQ,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,
          200Q,Q1Q,Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,
          3QER,QFR,QGR,QHR
0004      COMMON NCOL(30)
0005      COMMON /IMP/COL(80),CODE(128)
0006      COMMON /VAX/ N,MPAD,FR,Y
0007      COMMON /CODES/QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
          1QMQ,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,Q0Q,Q1Q,
          2Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,QGR,QHR
          D      WRITE (1,1) NCOL
          D1     FORMAT (1X,'NCOL=',30A1)
          C      SET BITS 14-16
0008      CODE(Y)=4+2+1
0009      W=0
0010      R3=SCAN(1,QER,0,0,0,0,Q)
0011      R=SCAN(1,QDR,0,0,0,0,P)
0012      R1=SCAN(1,QAR,0,0,0,0,P)
0013      R2=SCAN(1,QBR,0,0,0,0,P)
          D      WRITE (1,2) Q,R,R1,R2
          D2     FORMAT (1X,'Q=',I1,'R=',I1,' R1=',I1,' R2=',I1)
0014      IF (R.EQ.1.OR.R1.EQ.1.OR.R2.EQ.1) GOTO 520
0016      IF (Q.NE.5.AND.Q.NE.10.AND.Q.NE.16.AND.Q.NE.18.AND.Q.NE.24)
          1GOTO 530
0018      X=SCAN(4,QSQ,QTQ,QEQ,QPQ,0,K)
0019      IF (K.NE.4) GOTO 10
          C      SET BITS 8-9
0021      CODE(Y)=CODE(Y)+256+128
0022      IF (NCOL(5).EQ.QER) GOTO 510
0024      5      CALL WRT(34,2)
0025      RETURN
0026      10     X=SCAN(4,QJQ,QUQ,QMQ,QPQ,0,L)
0027      IF (L.NE.4) GOTO 20
          C      SET BITS 8,9,11,13
0029      CODE(Y)=CODE(Y)+256+128+32+8
0030      IF (NCOL(5).EQ.QER) GOTO 510
0032      GOTO 5
0033      20     X=SCAN(4,QSQ,QKQ,QIQ,QPQ,0,L1)
0034      IF (L1.NE.4) GOTO 30
          C      SET BITS 8-10,12
0036      CODE(Y)=CODE(Y)+256+128+64+16
0037      IF (NCOL(5).EQ.QER) GOTO 510
0039      GOTO 5
0040      30     X=SCAN(4,QEQ,QXQ,QEQ,QCR,0,L2)
0041      IF (L2.NE.4) GOTO 40
          C      SET BITS 8-13
0043      CODE(Y)=CODE(Y)+256+128+64+32+16+8
0044      IF (NCOL(5).EQ.QER) GOTO 510
0046      GOTO 5
          C      CHECK FOR RESERVED WORDS IN COLUMNS 3-5
0047      40     X=SCAN(3,QMQ,QSQ,QTQ,0,0,L3)
0048      IF (L3.EQ.5) GOTO 200
0050      X=SCAN(3,QAQ,QOQ,QVQ,0,0,L4)

```

FORTRAN IV V01B-02
 CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:16:45

PAGE 002

CONDIT.OBJ=CONDIT.FOR/NOSN/LI:1

```

0051      IF (L4 .NE. 5) GOTO 50
          C      SET BIT 7
0053      CODE(Y)=CODE(Y)+512
0054      GOTO 200
0055  50    X=SCAN(3,QLQ,QSQ,QTQ,0,0,L5)
0056      IF (L5.NE.5) GOTO 60
          C      SET BIT 6
0058      CODE(Y)=CODE(Y)+1024
0059      GOTO 200
0060  60    X=SCAN(3,QAQ,QBQ,QTQ,0,0,L6)
0061      IF (L6 .NE. 5) GOTO 70
          C      SET BITS 6-7
0063      CODE(Y)=CODE(Y)+1024+512
0064      GOTO 200
0065  70    X=SCAN(3,REQ,QXQ,QTQ,0,0,L7)
0066      IF (L7 .NE. 5) GOTO 80
          C      SET BITS 5-7
0068      CODE(Y)=CODE(Y)+2048+1024+512
0069      GOTO 200
0070  80    X=SCAN(2,QLQ,QCQ,0,0,0,L8)
0071      IF (L8 .EQ.4) GOTO 90
0073      CALL WRT(19,2)
0074      RETURN
          C      CHANGE HOLLERITH TO DECIMAL
0075  90    IF (NCOL(5) .NE. Q1Q) GOTO 601
0077      WZ=1
0078      GOTO 610
0079  601   IF (NCOL(5) .NE. Q2Q) GOTO 602
0081      WZ=2
0082      GOTO 610
0083  602   IF (NCOL(5).NE.Q3Q) GOTO 610
0085      WZ=3
0086  610   IF (WZ.GT.3.OR.WZ.LT.1) GOTO 95
0088      GOTO (100,110,120),WZ
0089  95    CALL WRT(25,2)
0090      RETURN
          C      SET BIT 5
0091  100   CODE(Y)=CODE(Y)+2048
0092      GOTO 200
          C      SET BITS 5,7
0093  110   CODE(Y)=CODE(Y)+2048+512
0094      GOTO 200
          C      SET BITS 5-6
0095  120   CODE(Y)=CODE(Y)+2048+1024
          C      CHANGE COLUMN 11 TO DECIMAL
0096  200   IF (NCOL(2).NE. QFQ) GOTO 5
0098      IF (NCOL(11) .NE. Q1Q) GOTO 701
0100      XZ=1
0101      GOTO 710
0102  701   IF (NCOL(11) .NE. Q2Q) GOTO 702
0104      XZ=2
0105      GOTO 710
0106  702   IF (NCOL(11) .NE. Q3Q) GOTO 710
0108      XZ=3-
0109  710   IF (XZ.LT.1.OR.XZ.GT.3) GOTO 205
0111      GOTO (230,220,210),XZ

```

FORTRAN IV U01B-02
CORE=08K, UIC=[20,20]MON 28-FEB-77 09:16:45
CONDIT.OBJ=CONDIT.FOR/NOSN/LI:1

PAGE 003

```
      C      SET BITS 8-9
0112 205  CODE(Y)=CODE(Y)+256+128
0113      GOTO 230
      C      SET BIT 8
0114 210  CODE(Y)=CODE(Y)+256
0115      GOTO 230
      C      SET BIT 9
0116 220  CODE(Y)=CODE(Y)+128
0117 230  X=SCAN(4,QSQ,QTQ,REQ,QPQ,0,J1)
      C      CHECK FOR "STEP" IN COLUMNS 6-9 OR 12-15
0118      IF (J1 .NE. 9 .AND. J1 .NE. 15) GOTO 240
0120      GOTO 350
      C      SAME WITH "JUMP" OR "EXEC"
0121 240  X1=SCAN(4,QJQ,QUQ,QMQ,QPQ,0,J2)
0122      IF (J2 .NE. 9 .AND. J2 .NE. 15) GOTO 250
      C      SET BIT 11
0124      CODE(Y)=CODE(Y)+32
0125      W=8
0126      GOTO 300
0127 250  X2=SCAN(4,QSQ,QKQ,QIQ,QPQ,0,J3)
0128      IF (J3 .NE. 9 .AND. J3 .NE. 15) GOTO 260
      C      SET BIT 10
0130      CODE(Y)=CODE(Y)+64
0131      W=16
0132      GOTO 300
0133 260  X3=SCAN(4,REQ,QXQ,REQ,QCQ,0,J4)
0134      IF (J4 .EQ. 9 .OR. J4 .EQ. 15) GOTO 270
0136      CALL WRT(21,2)
0137      RETURN
      C      SET BITS 10-11
0138 270  CODE(Y)=CODE(Y)+64+32
0139      W=16+8
0140 300  IF (X.EQ. 0) GOTO 350
0142      GOTO 510
0143 350  X4=SCAN(4,QJQ,QUQ,QMQ,QPQ,0,J5)
0144      IF (X4.EQ.0) GOTO 370
0146      IF (J5.EQ.9.OR.J5.EQ.15) GOTO 370
      C      SET BIT 13
0148      CODE(Y)=CODE(Y)+8
0149      GOTO 510
0150 370  X5=SCAN(4,QSQ,QKQ,QIQ,QPQ,0,J6)
0151      IF (X5.EQ.0) GOTO 400
0153      IF (J6.EQ.9.OR.J6.EQ.15) GOTO 400
      C      SET BIT 12
0155      CODE(Y)=CODE(Y)+16
0156      GOTO 510
0157 400  X7=SCAN(4,REQ,QXQ,REQ,QCQ,0,J8)
0158      IF (X7.EQ.0) GOTO 500
0160      IF (J8.EQ.9.OR.J8.EQ.15) GOTO 500
      C      SET BITS 12-13
0162      CODE(Y)=CODE(Y)+16+8
0163      GOTO 510
      C      IF CODE ENDS IN COLUMN 10 OR 16, DO NOT ADD W TO CODE(Y)
0164 500  IF (NCOL(10).EQ.QER.OR.NCOL(16).EQ.QER) GOTO 510
0166      CODE(Y)=CODE(Y)+W
0167 510  WRITE (1,511) MPAD, CODE(Y), COL
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:16:45

PAGE 004

CONDIT.OBJ=CONDIT.FOR/NOSN/LI:1

```
0168 511  FORMAT (1X,05,4X,05,5X,80A1)
0169      MPAD=MPAD+1
0170      Y=Y+1
0171      RETURN
0172 520  CALL WRT(20,2)
0173      RETURN
0174 530  CALL WRT(34,2)
0175      RETURN
0176      END
```

>

LITRL

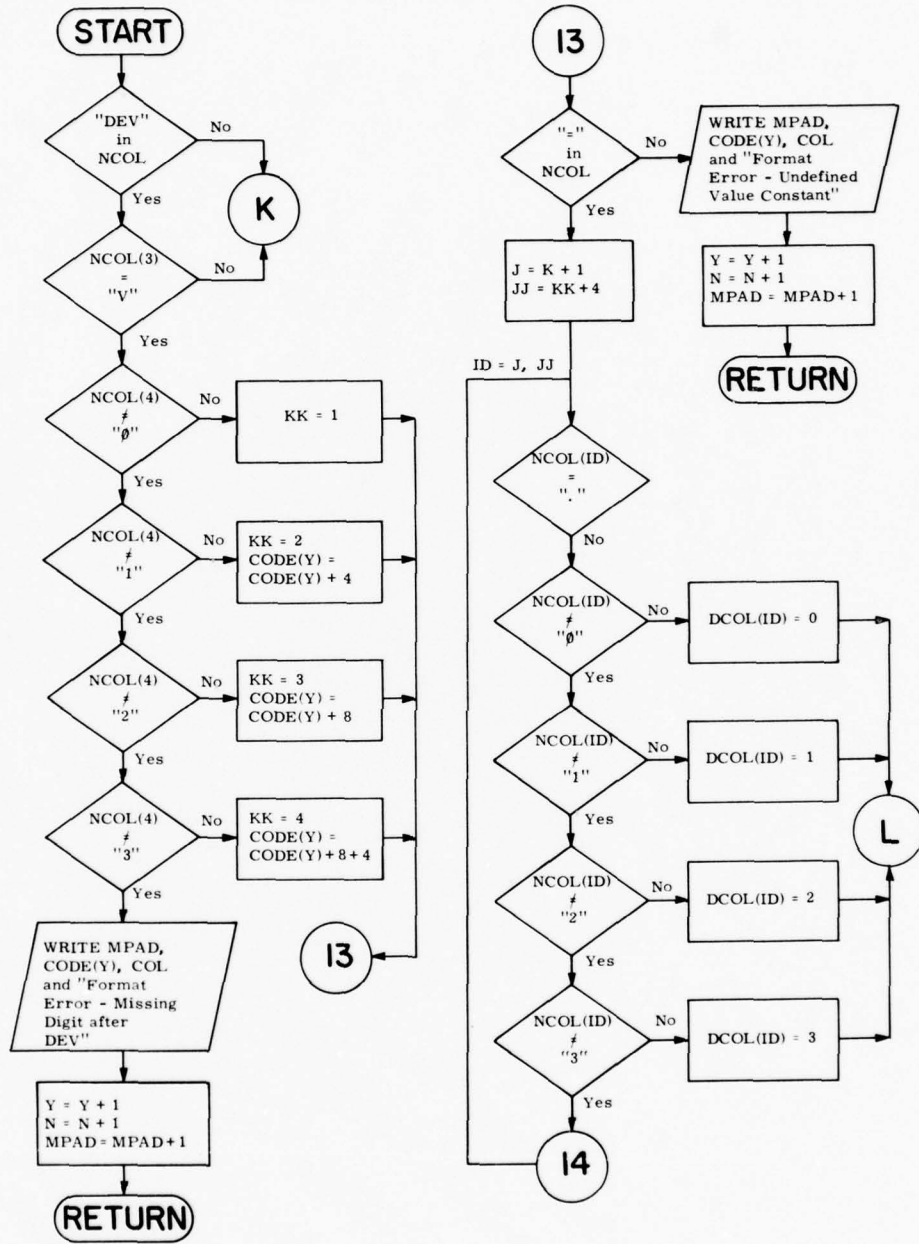


Figure 1-7. LITRL

LITRL (cont.)

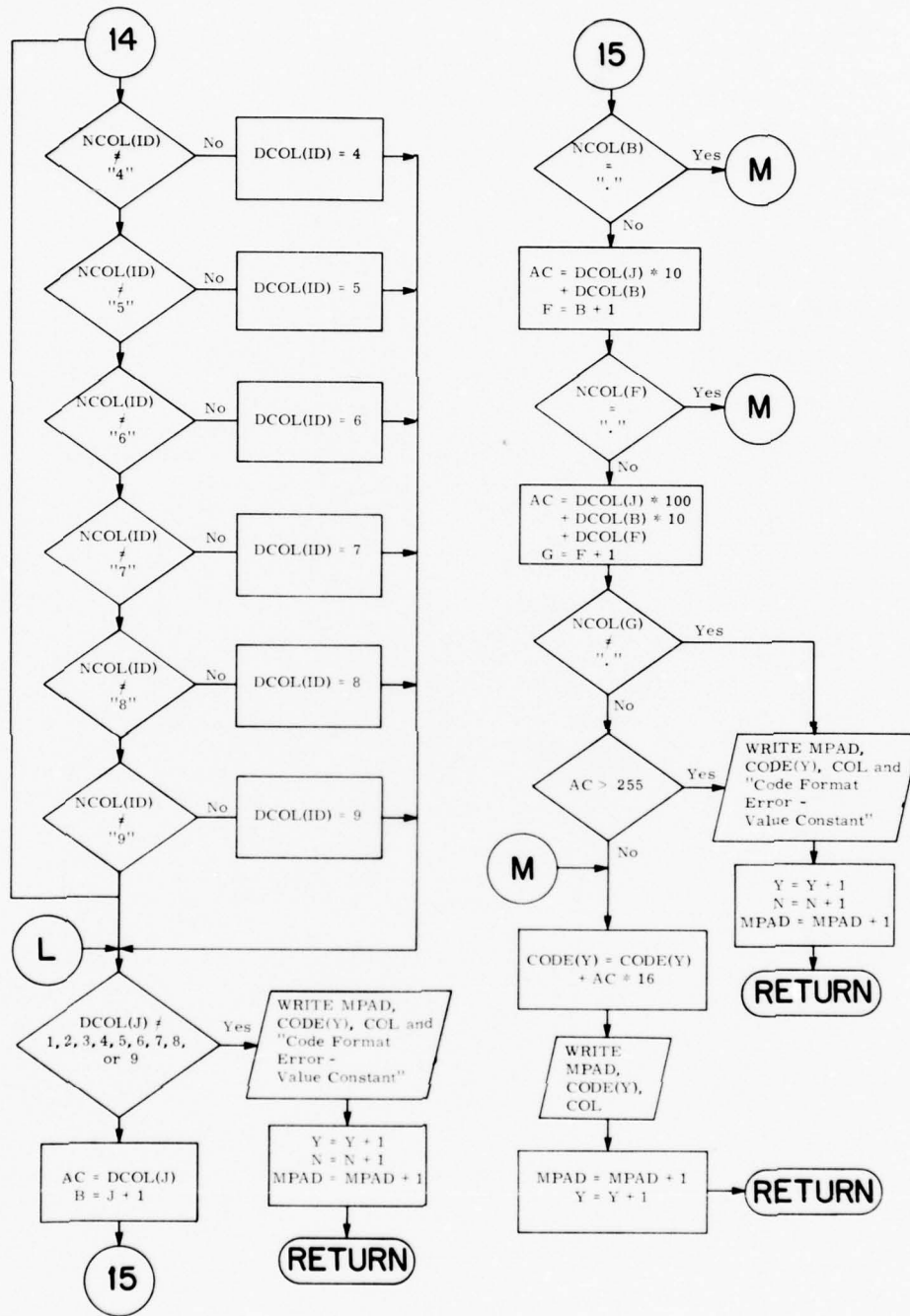


Figure 1-7. (Cont.)

LITRL (cont.)

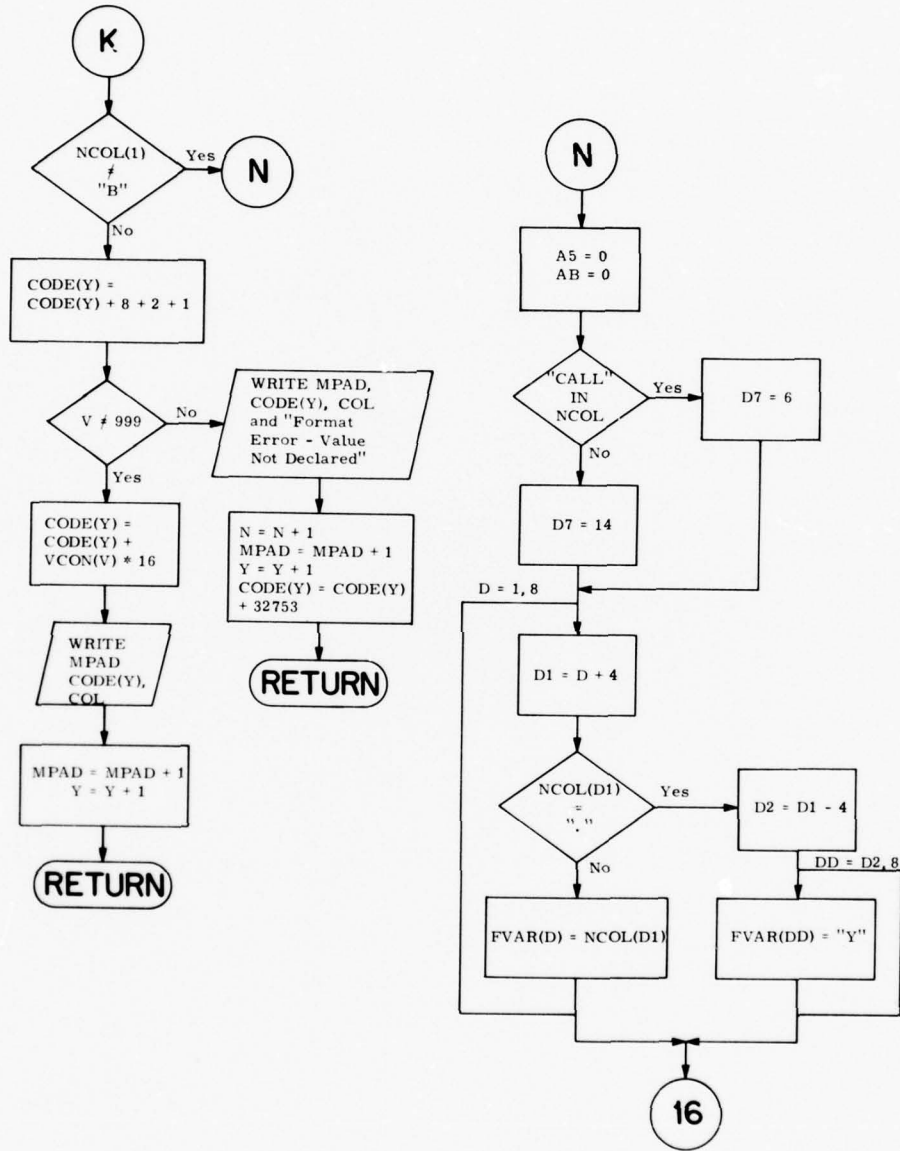


Figure 1-7. (Cont.)

LITRL (cont.)

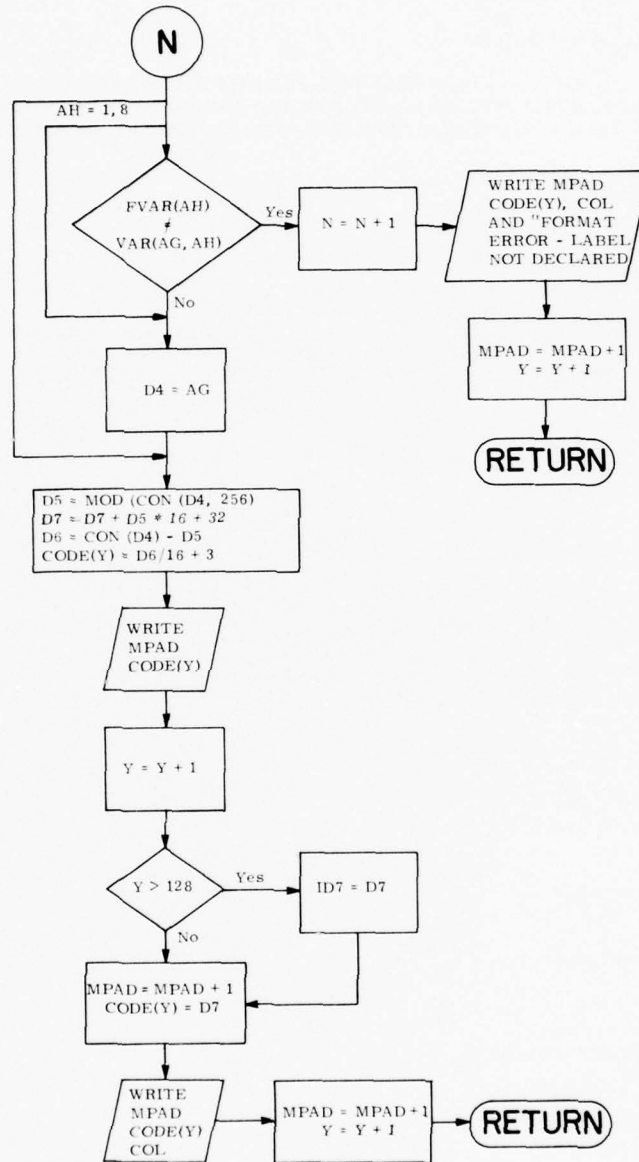


Figure I-7. (Cont.)

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

SAT 26-FEB-77 17:35:56

PAGE 001

LITRL.OBJ=LITRL.FOR/NOSN/LI:1

```

0001 SUBROUTINE LITRL(V,DEV)
0002 IMPLICIT INTEGER (A-Y)
0003 LOGICAL*1 VVAR,FVAR,VAR
0004 LOGICAL*1 DCOL
0005 LOGICAL*1 COL,NCOL,QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,
1QLQ,QMQ,QNQ,QQQ,QFQ,QQQ,QRQ,RSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,
2QQQ,Q1Q,Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,
3QFR,QGR,QHR
0006 DIMENSION DCOL(30)
0007 COMMON /FAR/FP1,FP2
0008 COMMON NCOL(30)
0009 COMMON /IMP/COL(80),CODE(128)
0010 COMMON /VAX/N,MPAD,FR,Y
0011 COMMON/VAL/VVAR(80,8),VCON(80)
0012 COMMON /SJI/FVAR(8),VAR(100,8),CON(100)
0013 COMMON /CODES/QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
1QM, QNQ,QQQ,QFQ,QQQ,QRQ,RSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,QOQ,Q1Q,
2Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,
3QGR,QHR
0014 COMMON /SULIT/ID7
0015 VW=SCAN(3,QDQ,QEQ,QVQ,0,0,JF)
D WRITE (1,890) VW
D890 FORMAT(1X,'VW=',I1)
0016 IF (VW.NE.1) GO TO 10
0018 IF (JF.NE.3) GO TO 10
0020 IF (NCOL(4) .NE. QOQ) GOTO 420
0022 KK=1
0023 GOTO 1
0024 420 IF (NCOL(4) .NE. Q1Q) GOTO 430
0026 KK=2
0027 GOTO 1
0028 430 IF (NCOL(4) .NE. Q2Q) GOTO 440
0030 KK=3
0031 GOTO 1
0032 440 IF (NCOL(4) .NE. Q3Q) GOTO 8
0034 KK=4
0035 1 GOTO(5,2,3,4),KK
0036 8 CALL WRT(24,2)
0037 RETURN
C SET BIT 14
0038 2 CODE(Y)=CODE(Y)+4
-0039 GOTO 5
C SET BIT 13
0040 3 CODE(Y)=CODE(Y)+8
0041 GOTO 5
C SET BITS 13+14
0042 4 CODE(Y)=CODE(Y)+8+4
0043 5 X=SCAN(1,QDR,0,0,0,0,K)
D WRITE (1,891) X
D891 FORMAT(' X=',I1)
0044 IF(X.EQ.1)GOTO 7
0046 6 CALL WRT(23,2)
0047 RETURN
C THIS CHANGES FROM HOLERITH TO DECIMAL
0048 7 J=K+1
0049 JJ=K+4

```

FORTRAN IV V01B-02
 CORE=08K, UIC=C20,20J

SAT 26-FEB-77 17:35:56

PAGE 002

LITRL.OBJ=LITRL.FOR/NOSN/LI:1

```

0050      DO 11 ID=J,JJ
0051      IF (NCOL(ID).EQ.QER) GOTO 13
0053      IF (NCOL(ID).NE.Q0Q) GOTO 501
0055      DCOL(ID)=0
0056      GOTO 11
0057 501  IF (NCOL(ID).NE.Q1Q) GOTO 502
0059      DCOL(ID)=1
0060      GOTO 11
0061 502  IF (NCOL(ID).NE.Q2Q) GOTO 503
0063      DCOL(ID)=2
0064      GOTO 11
0065 503  IF (NCOL(ID).NE.Q3Q) GOTO 504
0067      DCOL(ID)=3
0068      GOTO 11
0069 504  IF (NCOL(ID).NE.Q4Q) GOTO 505
0071      DCOL(ID)=4
0072      GOTO 11
0073 505  IF (NCOL(ID).NE.Q5Q) GOTO 506
0075      DCOL(ID)=5
0076      GOTO 11
0077 506  IF (NCOL(ID).NE.Q6Q) GOTO 507
0079      DCOL(ID)=6
0080      GOTO 11
0081 507  IF (NCOL(ID).NE.Q7Q) GOTO 508
0083      DCOL(ID)=7
0084      GOTO 11
0085 508  IF (NCOL(ID).NE.Q8Q) GOTO 509
0087      DCOL(ID)=8
0088      GOTO 11
0089 509  IF (NCOL(ID).NE.Q9Q) GOTO 11
0091      DCOL(ID)=9
0092 11   CONTINUE
C        THIS MAKES SURE THERE IS A CONSTANT IN THE DEV STATEMENT
0093 13   IF (DCOL(J).NE.1.AND.DCOL(J).NE.2.AND.DCOL(J).NE.3.AND.DCOL(J).NE.4
        1.AND.DCOL(J).NE.5.AND.DCOL(J).NE.6.AND.DCOL(J).NE.7.AND.DCOL(J)
        2.NE.8.AND.DCOL(J).NE.9.AND.DCOL(J).NE.0) GOTO 6
D        WRITE (1,895) DCOL(J)
DB95     FORMAT(1X,'DCOL(J)=' ,I1)
C        THIS FINDS THE CINSTANT IN THE DEV STATEMENT
0095     AC=DCOL(J)
0096     B=J+1
0097     IF (NCOL(B).EQ.QER) GOTO 9
0099     AC=DCOL(J)*10+DCOL(B)
0100     F=B+1
0101     IF (NCOL(F).EQ.QER) GOTO 9
0103     AC=DCOL(J)*100+DCOL(B)*10+DCOL(F)
C        THIS MAKES SURE THE CONSTANT IS NOT TOO LARGE
0104     G=F+1
0105     IF (NCOL(G).NE.QER) GOTO 6
0107     IF (AC.GT.255) GOTO 6
0109     9   CODE(Y)=CODE(Y)+AC*16
D        WRITE (1,61) AC, CODE(Y).
D61     FORMAT(' AC=' ,I3, ' CODE(Y)=' ,O5)
D        WRITE (1,893)
DB93     FORMAT(' ABOUT TO CALL WRT(0,4) AT 53 IN LITRL')
0110     GOTO 400

```

FORTRAN IV V01B-02
 CORE=08K, UIC=E20,20J

SAT 26-FEB-77 17:35:56

PAGE 003

LITRL.OBJ=LITRL.FOR/NOSN/LI:1

```

      C      THE CONSTANT CORRESPONDING TO THE VARIABLE AFTER THE B IS NOW
      C      ADDED TO THE CODE.  THE VARIABLE WAS ALREADY FOUND IN THE MAIN
      C      PROGRAM.
0111 10     IF(NCOL(1).NE.QBQ) GOTO 15
0113      CODE(Y)=CODE(Y)+8+2+1
      D      WRITE (1,21) V
      D21    FORMAT (' V=',I3)
0114      IF (V.NE.999) GO TO 14
0116      N=N+1
0117      WRITE(1,29)MPAD, CODE(Y), (COL(I), I=1,80)
0118 29     FORMAT(1X,05,4X,05,5X,80A1/' FORMAT ERROR-VALUE NOT DECLARED')
0119      MPAD=MPAD+1
0120      CODE(Y)=CODE(Y)+32753
0121      Y=Y+1
0122      RETURN
0123 14     CODE(Y)=CODE(Y)+VCON(V)*16
      D      WRITE (1,17) VCON(V)
      D17    FORMAT (' VCON(V)=' ,I3)
0124      GOTO 400
      C      THIS ASSIGNS THE CODE FOR THE CALL OR GOTO STATEMENTS
0125 15     A5=0
0126      AB=0
      D      WRITE (1,892)
      D892   FORMAT(1X,'CHECKING FOR CALL OR GOTO IN LITRL')
      C      THIS CHECKS TO SEE IF A LIT TO IR CODE IS NEEDED
0127      C=SCAN(4,QCQ,QAQ,QLQ,QLQ,0,C1)
0128      IF (C.NE.1) GO TO 20
0130      D7=6
0131      GOTO 25
0132 20     D7=14
0133 25     DO 27 D=1,8
0134      D1=D+4
0135      IF (NCOL(D1).EQ.QER) GO TO 30
0137 27     FVAR(D)=NCOL(D1)
0138      GO TO 33
0139 30     D2=D1-4
0140      DO 32 DD=D2,8
0141 32     FVAR(DD)=QHR
0142 33     CONTINUE
      D      WRITE(1,34) FVAR
0143 34     FORMAT(' LABEL IS ',8A1)
0144      DO 40 AG=1,PF2
0145      DO 35 AH=1,8
0146      IF (FVAR(AH).NE.VAR(AG,AH)) GO TO 40
0148 35     CONTINUE
0149      D4=AG
0150      GO TO 45
0151 40     CONTINUE
0152      N=N+1
0153      WRITE(1,23)MPAD, CODE(Y), (COL(I), I=1,80)
0154 23     FORMAT(1X,05,4X,05,5X,80A1/' FORMAT ERROR-LABEL NOT DECLARED')
0155      MPAD=MPAD+1
0156      Y=Y+1
0157      RETURN
0158 45     D5=MOD(CON(D4),256)
0159      D7=D7+D5*16

```

FORTRAN IV V01B-02
 CORE=08K, UIC=[20,20]

SAT 26-FEB-77 17:35:56

PAGE 004

LITRL.OBJ=LITRL.FOR/NOSN/LI:1

```

0160      D6=CON(D4)-D5
0161      CODE(Y)=D6/16+3
0162      WRITE(1,46) MPAD, CODE(Y)
0163  46   FORMAT(1X,05,4X,05)
0164      Y=Y+1
0165      IF (Y .GT. 128) ID7=D7
0166      MPAD=MPAD+1
0168      CODE(Y)=D7
0169      GO TO 400
0170  C    THIS ASSIGNS THE LIT TO IR CODE
50      CODE(Y)=CODE(Y)+2+1
C      THIS FINDS THE PROPER LABEL FOR THE STATEMENT
0171      DO 60 H=1,8
0172      H1=H+4
0173      IF (NCOL(H1).EQ.QER) GO TO 63
0175  60   FVAR(H)=NCOL(H1)
0176  63   IN=H1-4
0177      DO 64 I=IN,8
0178  64   FVAR(I)=QHR
0179      DO 70 AB=1,PF2
0180      DO 65 AC=1,8
0181      IF (FVAR(AC).NE.VAR(AB,AC)) GO TO 70
0183  65   CONTINUE
0184      AZ=AB
0185      GOTO 100
0186  70   CONTINUE
0187      AB=0
0188      WRITE(1,23)MPAD, CODE(Y), COL
0189      MPAD=MPAD+1
0190      Y=Y+1
0191      N=N+1
0192      GOTO 150
C      THE LABEL HAS BEEN FOUND AND THIS ADDS THE CORRESPONDING
C      VALUE.
0193  100  AB=AZ
0194      A5=(CON(AB)-16)/256
0195      CODE(Y)=CODE(Y)+A5*16
0196      WRITE (1,401) MPAD, CODE(Y), COL
0197      MPAD=MPAD+1
0198      Y=Y+1
0199      CODE(Y)=-16
C      IT IS PAST THE LIT TO IR AND THE PROGRAM IS CHECKING TO SEE IF
C      THE PROGRAM WILL NOW DO TWICEWHAT THE LIT TO IR PART DID,
C      ONCE FOR THE CALL AND ONCE FOR THE GOTO
0200  150  L1=SCAN(4,QCQ,QAQ,QLQ,QLQ,0,D3)
0201      IF (L1.NE.1) GO TO 270
0203      CODE(Y)=CODE(Y)+4+2
0204  170  IF (AB .NE. 0) GOTO 200
0206      MPAD=MPAD+1
0207      RETURN
0208  200  XY=MOD(CON(AB),256)
0209      CODE(Y)=CODE(Y)+XY*16
0210      GOTO 400
0211  270  CODE(Y)=CODE(Y)+14
0212      GOTO 170
0213  400  WRITE(1,401) MPAD, CODE(Y), COL

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

SAT 26-FEB-77 17:35:56

PAGE 005

LITRL.OBJ=LITRL.FOR/NOSN/LI:1

```
0214    401 FORMAT(1X,05,4X,05,5X,80A1)
0215            MPAD=MPAD+1
0216            Y=Y+1
0217            RETURN
0218            END
```

>

LOGIC

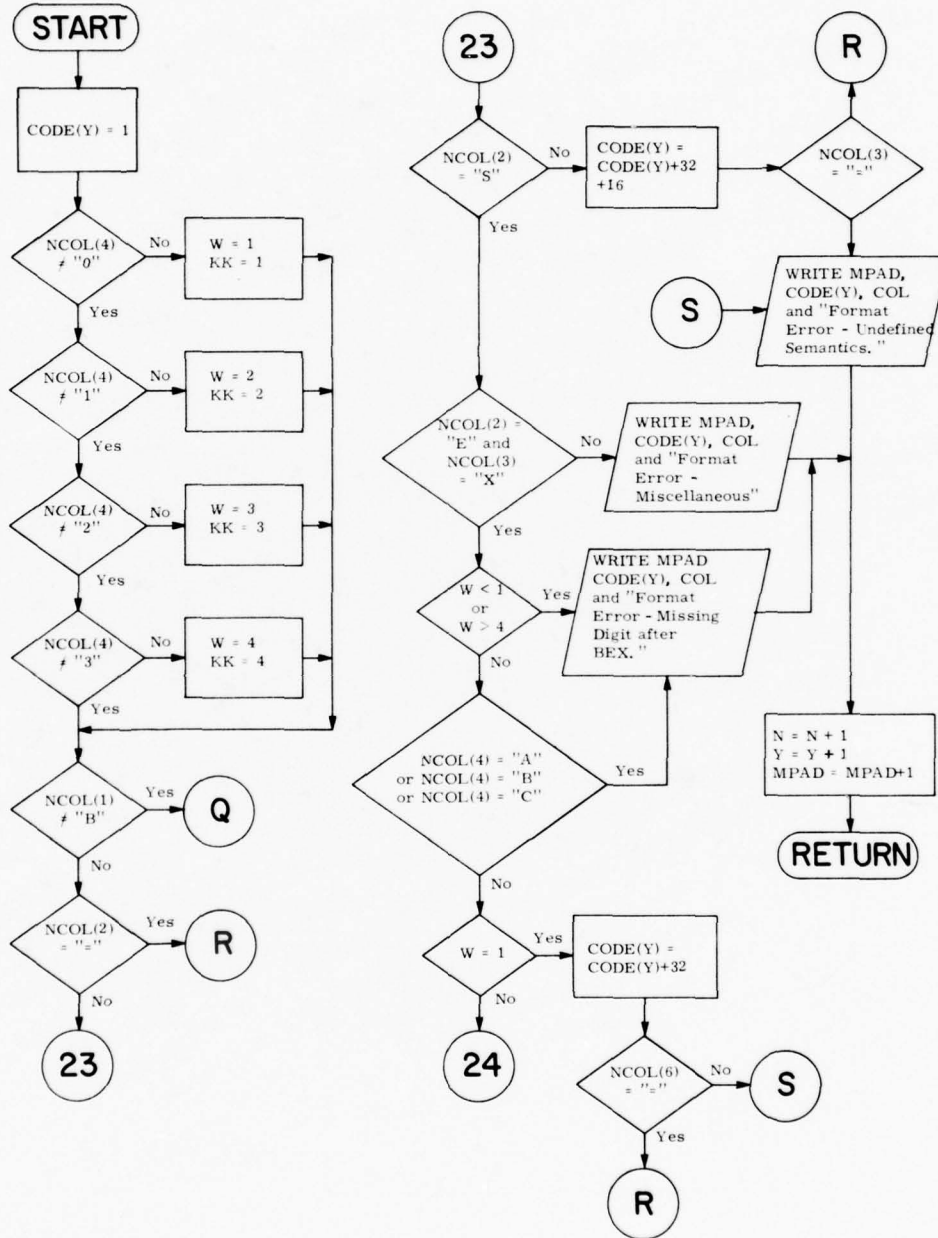


Figure 1-8. LOGIC

LOGIC (cont.)

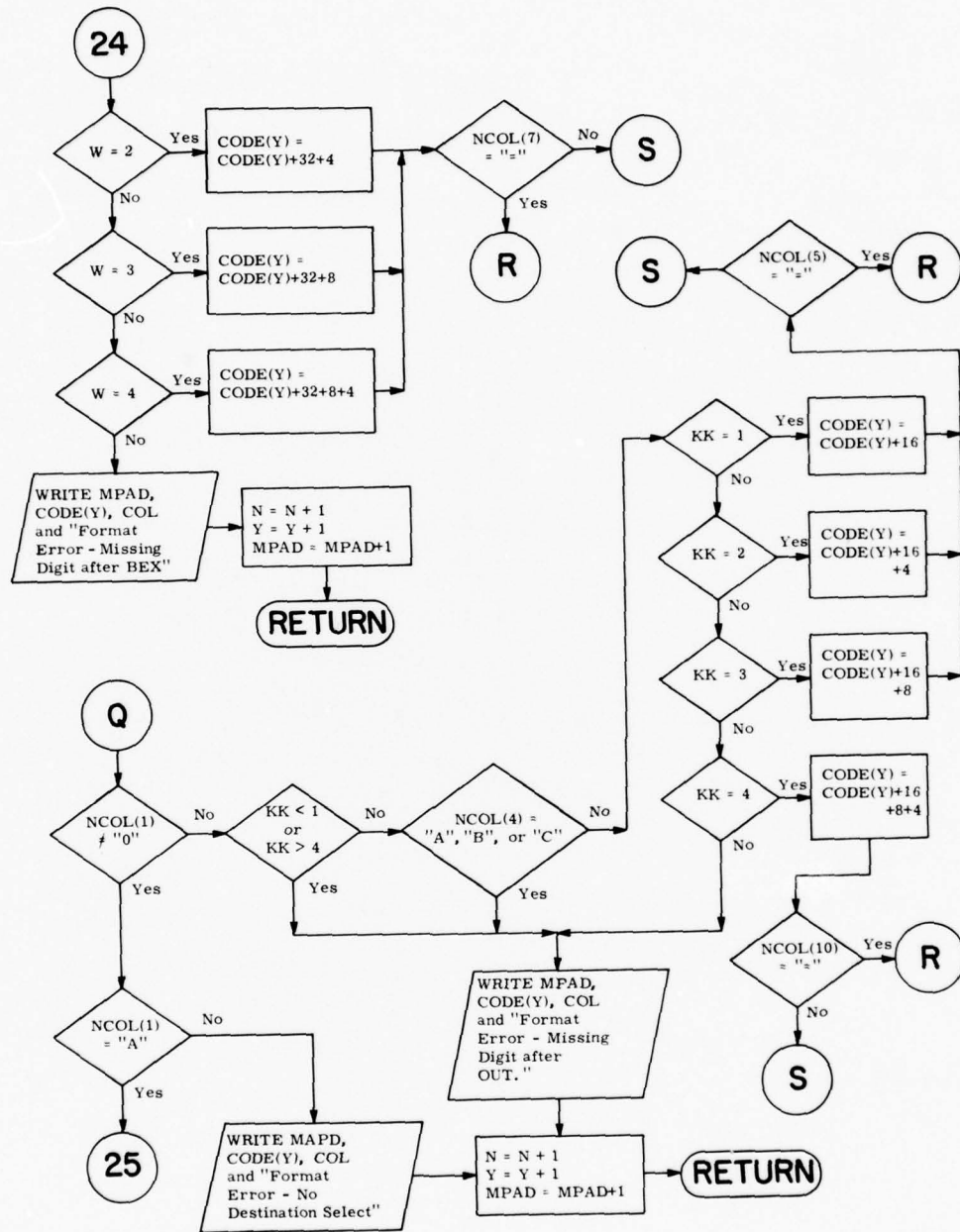


Figure 1-8. (Cont.)

LOGIC (cont.)

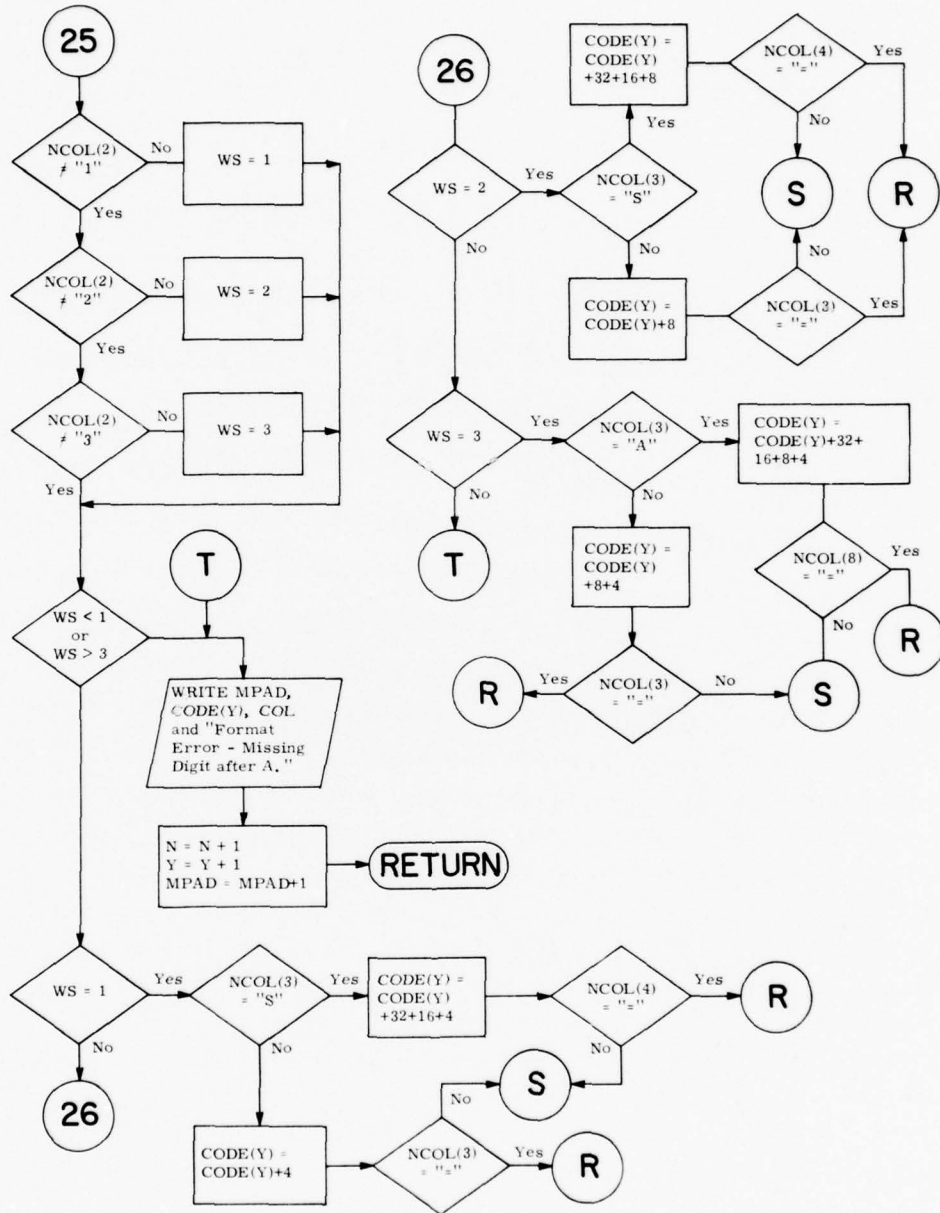


Figure 1-8. (Cont.)

LOGIC (cont.)

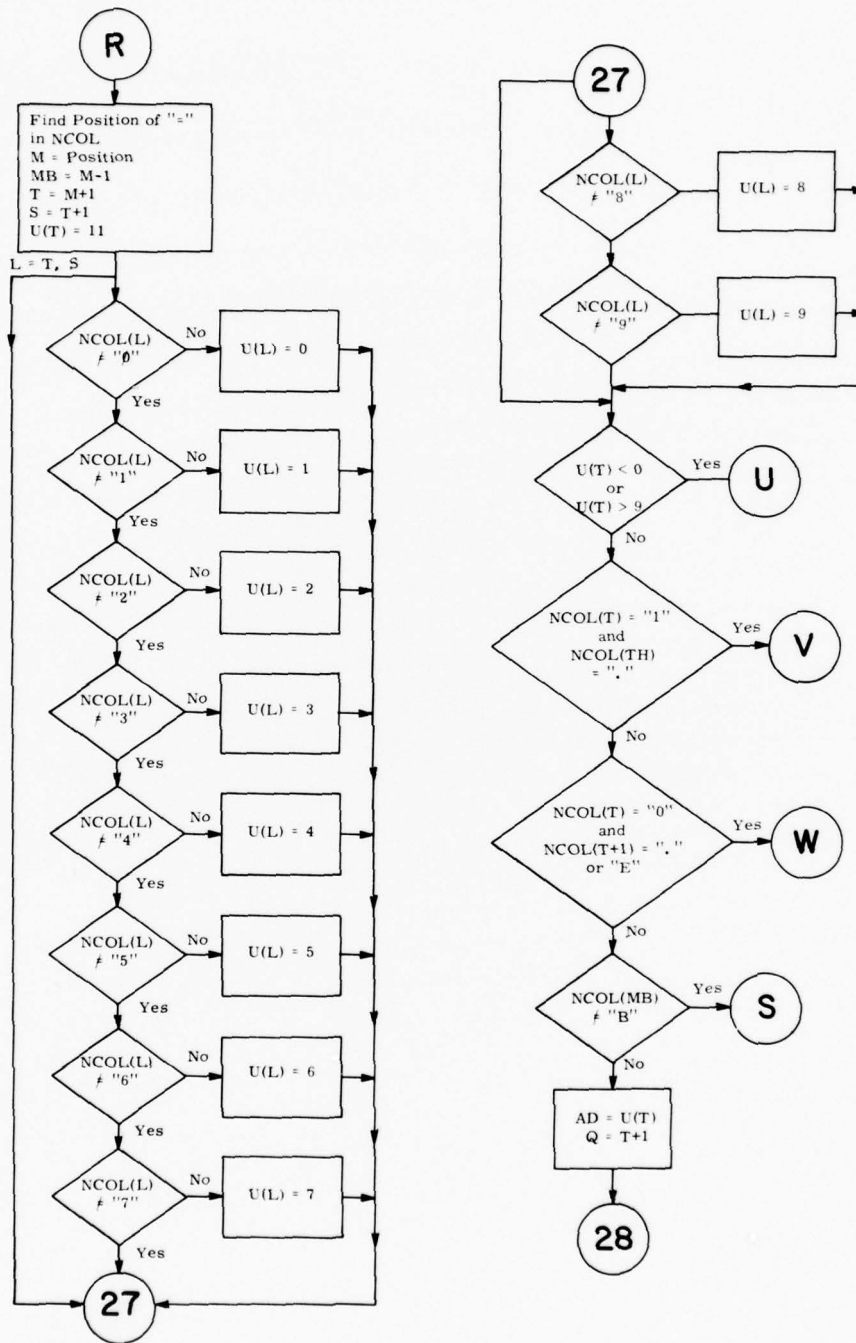


Figure 1-8. (Cont.)

LOGIC (cont.)

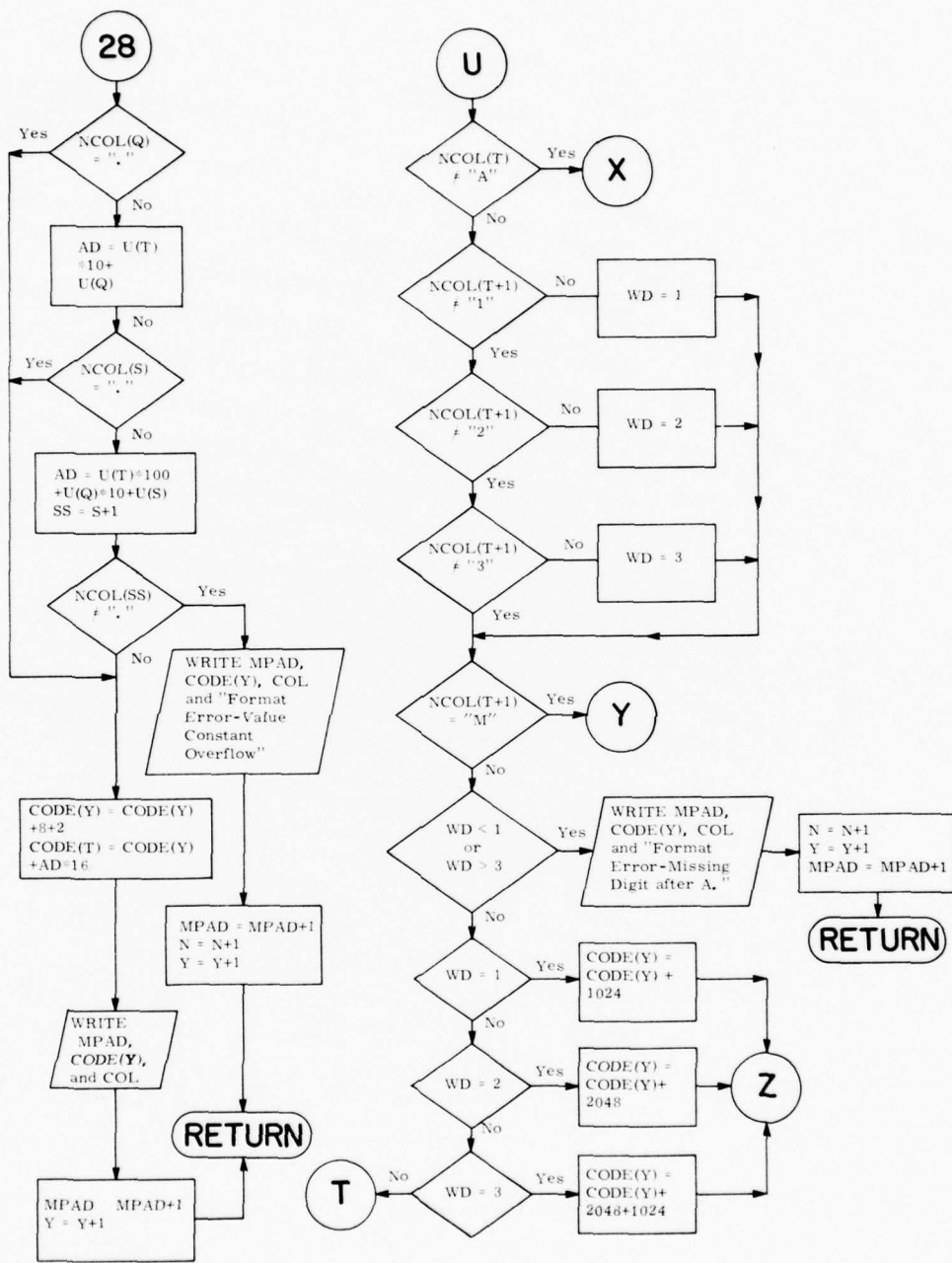


Figure 1-8. (Cont.)

LOGIC (cont.)

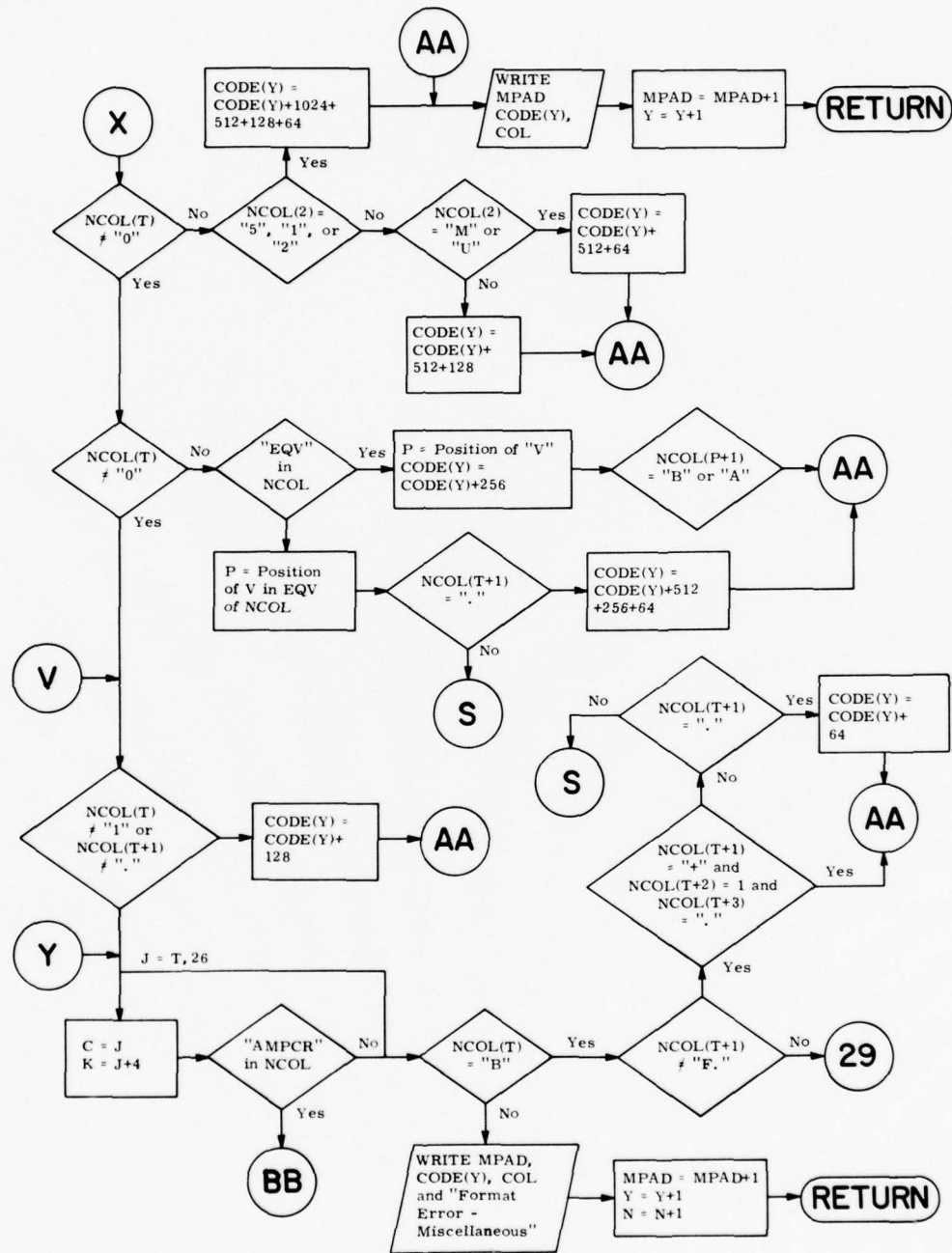


Figure 1-8. (Cont.)

LOGIC (cont.)

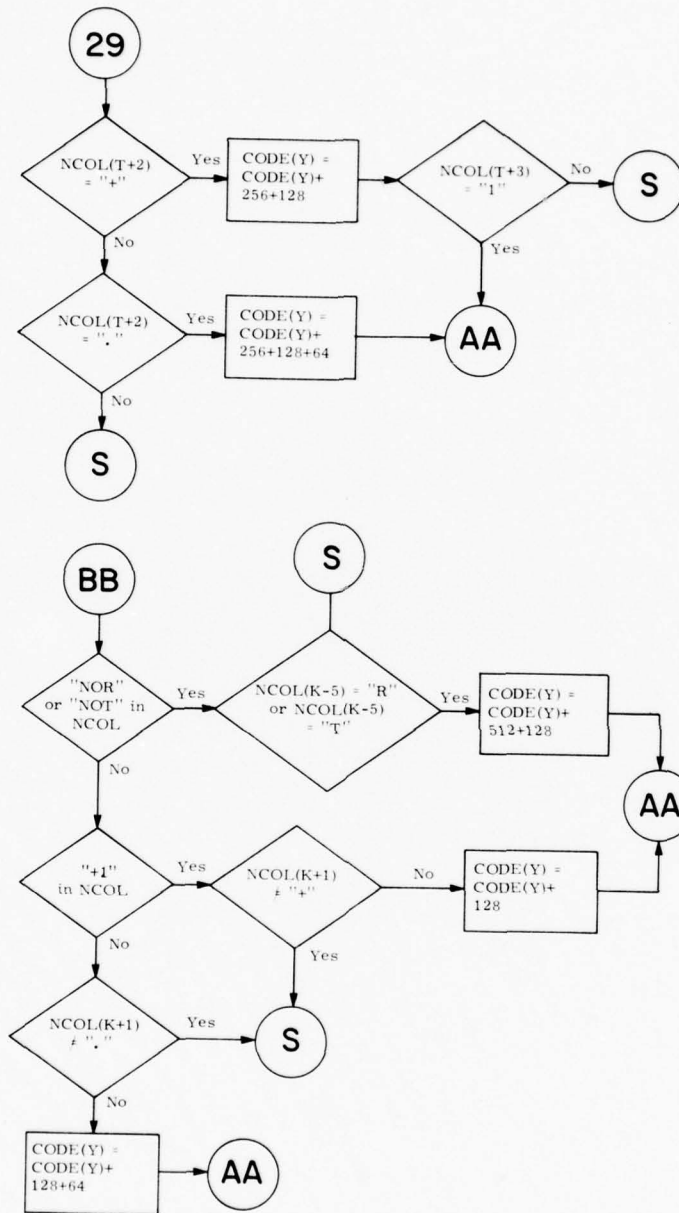


Figure 1-8. (Cont.)

LOGIC (cont.)

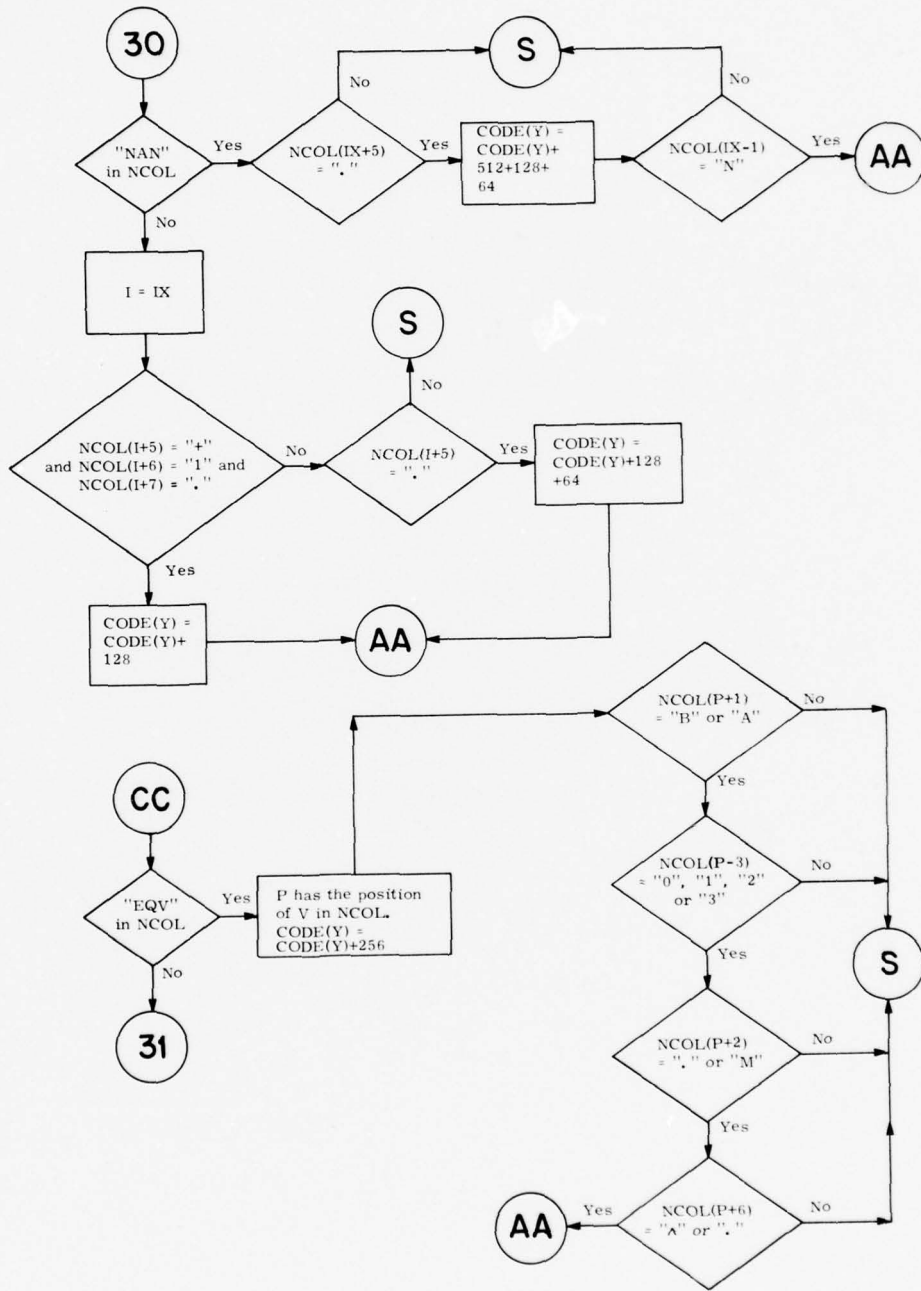


Figure 1-8. (Cont.)

LOGIC (cont.)

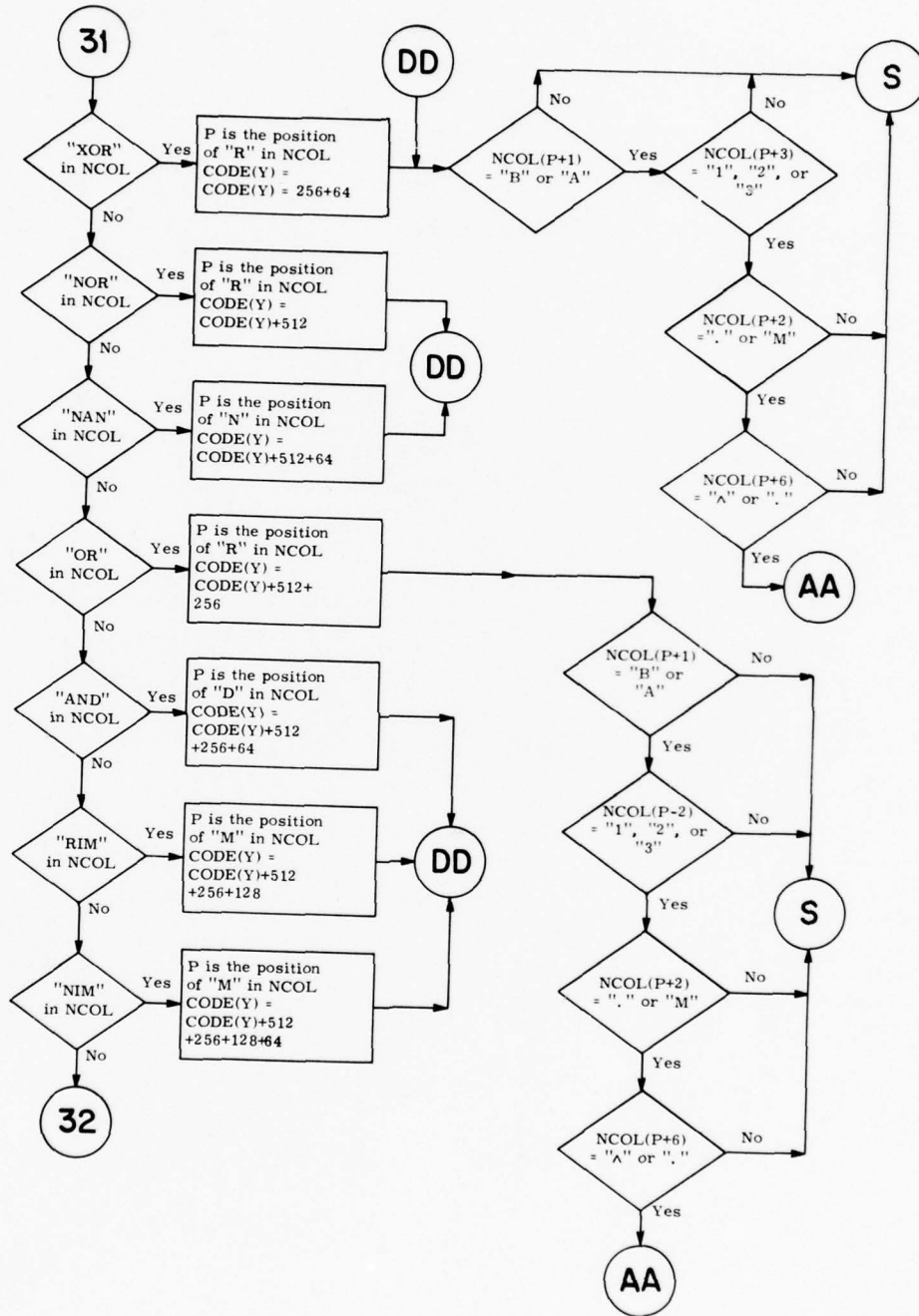


Figure 1-8. (Cont.)

LOGIC (cont.)

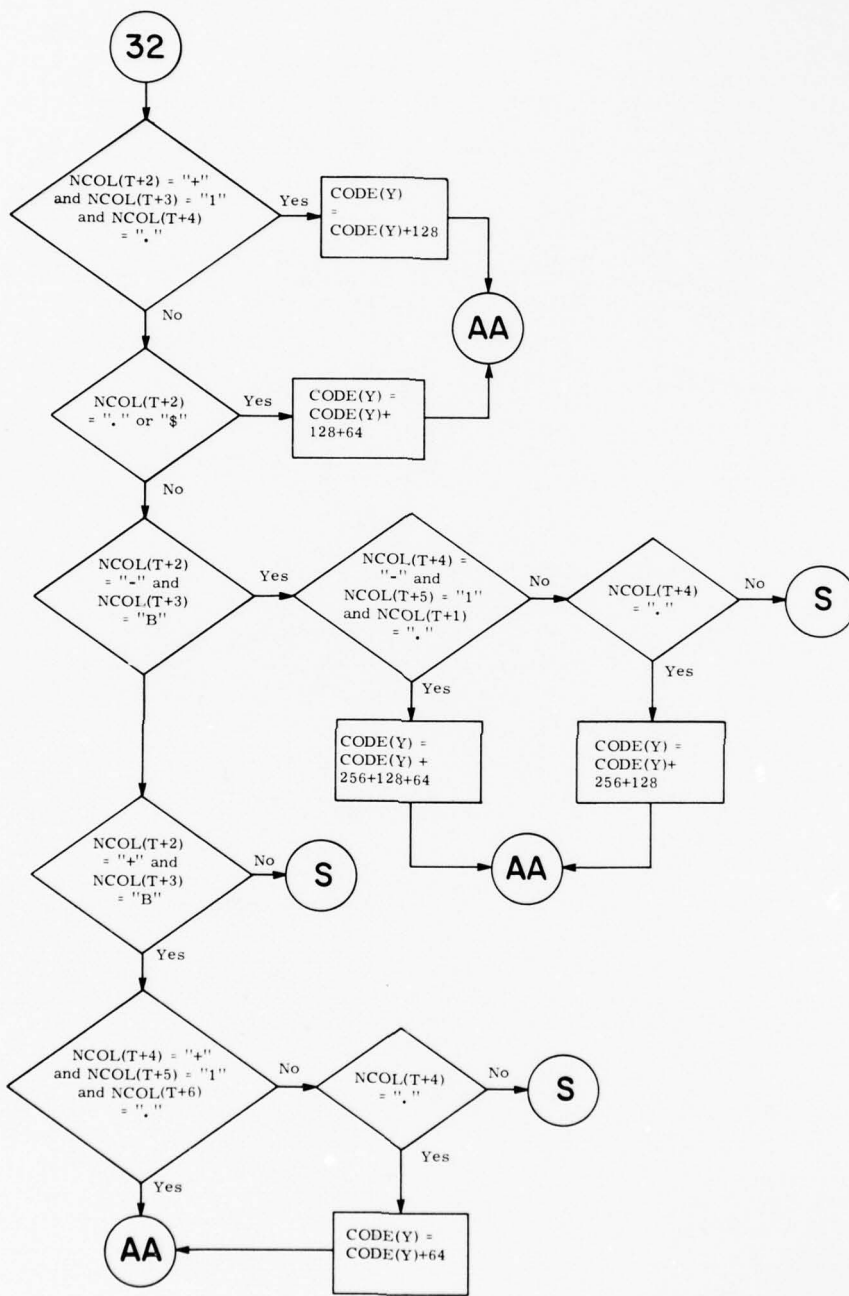


Figure 1-8. (Cont.)

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]MON 28-FEB-77 09:22:07
LOGIC.OBJ=LOGIC.FOR/NOSN/LI:1

PAGE 001

```
0001      SUBROUTINE LOGIC
0002      IMPLICIT INTEGER (A-Y)
0003      LOGICAL*1 COL,NCOL,QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,
1QLQ,QMQ,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,
2QOQ,Q1Q,Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,
3QFR,QGR,QHR
0004      COMMON NCOL(30)
0005      COMMON /IMP/COL(80),CODE(128)
0006      COMMON /VAX/N,MPAD,FR,Y
0007      COMMON /CODES/QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
1QMQ,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,QOQ,Q1Q,
2Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,QGR,QHR
0008      DIMENSION U(30)
C      SET BIT 16
0009      CODE(Y)=1
0010      IF (NCOL(4).NE. QOQ) GOTO 2
0012      W=1
0013      KK=1
0014      GOTO 10
0015 2      IF (NCOL(4) .NE. Q1Q) GOTO 3
0017      W=2
0018      KK=2
0019      GOTO 10
0020 3      IF (NCOL(4) .NE. Q2Q) GOTO 4
0022      W=3
0023      KK=3
0024      GOTO 10
0025 4      IF (NCOL(4) .NE. Q3Q) GOTO 10
0027      W=4
0028      KK=4
0029 10      IF (NCOL(1) .NE. QBQ) GOTO 100
0031      IF (NCOL(2) .EQ. QDR) GOTO 500
C      GO CHECK AFTER THE EQUAL SIGN
0033      IF (NCOL(2) .NE. QSQ) GOTO 60
C      SET BITS 11,12
0035      CODE(Y)=CODE(Y)+32+16
0036      IF (NCOL(3) .EQ. QDR) GOTO 500
0038      GOTO 696
0039 60      IF (NCOL(2) .EQ. QEQ .AND. NCOL(3) .EQ. QXQ) GOTO 65
0041      CALL WRT(18,2)
0042      RETURN
0043 65      IF (W.LT.1.OR.W.GT.4) GOTO 67
0045      IF (NCOL(4).EQ.QAQ.OR.NCOL(4).EQ.QBQ.OR.NCOL(4).EQ.QCQ) GOTO 67
0047      GOTO (70,80,90,95) W
0048 67      CALL WRT(26,2)
0049      RETURN
C      SET BIT 11
0050 70      CODE(Y)=CODE(Y)+32
0051      IF (NCOL(6).EQ.QDR) GOTO 500
0053      GOTO 696
C      SET BITS 11,14
0054 80      CODE(Y)=CODE(Y)+32+4
0055      IF (NCOL(7).EQ.QDR) GOTO 500
0057      GOTO 696
CSET BITS 11,13
0058 90      CODE(Y)=CODE(Y)+32+8
```

FORTRAN IV V01B-02
 CORE=08K, UIC=220,20J

MON 28-FEB-77 09:22:07

PAGE 002

LOGIC.OBJ=LOGIC.FOR/NOSN/LI:1

```

0059      IF (NCOL(7).EQ.QDR) GOTO 500
0061      GOTO 696
          C      SET BITS 11,13,14
0062  95    CODE(Y)=CODE(Y)+32+8+4
0063      IF (NCOL(7).EQ.QDR) GOTO 500
0065      GOTO 696
          C      CHECK "OUT"
0066  100   IF (NCOL(1) .NE. Q0Q) GOTO 200
0068      IF (KK.LT.1.OR.KK.GT.4) GOTO 105
0070      IF (NCOL(4).EQ.QAQ.OR.NCOL(4).EQ.QBQ.OR.NCOL(4).EQ.QCQ) GOTO 105
0072  103   GOTO (110,120,130,140) KK
0073  105   CALL WRT(31,2)
0074      RETURN
          C      SET BIT 12
0075  110   CODE(Y)=CODE(Y)+16
0076      IF (NCOL(5).EQ.QDR) GOTO 500
0078      GOTO 696
          C      SET BITS 12,14
0079  120   CODE(Y)=CODE(Y)+16+4
0080      IF (NCOL(5).EQ.QDR) GOTO 500
0082      GOTO 696
          C      SET BITS 12,13
0083  130   CODE(Y)=CODE(Y)+16+8
0084      IF (NCOL(5).EQ. QDR) GOTO 500
0086      GOTO 696
          C      SET BITS 12-14
0087  140   CODE(Y)=CODE(Y)+16+8+4
0088      IF (NCOL(10) .EQ. QDR) GOTO 500
0090      GOTO 696
0091  200   IF (NCOL(1).EQ.QAQ) GOTO 300
0093      CALL WRT(32,2)
0094      RETURN
0095  300   IF (NCOL(2).NE. Q1Q) GOTO 11
0097      WS=1
0098      GOTO 20
0099  11    IF (NCOL(2).NE.Q2Q) GOTO 12
0101      WS=2
0102      GOTO 20
0103  12    IF (NCOL(2) .NE. Q3Q) GOTO 20
0105      WS=3
0106  20    IF (WS.LT.1.OR.WS.GT.3) GOTO 305
0108      IF (NCOL(2).EQ.QAQ.OR.NCOL(2).EQ.QBQ.OR.NCOL(2).EQ.QCQ)GOTO 696
0110  301   GOTO (310,320,330) WS
0111  305   CALL WRT(33,2)
0112      RETURN
0113  310   IF (NCOL(3).EQ.QSQ)GOTO 315
          C      SET BIT 14
0115      CODE(Y)=CODE(Y)+4
0116      IF (NCOL(3).EQ.QDR) GOTO 500
0118      GOTO 696
          C      SET BITS 11,12,14
0119  315   CODE(Y)=CODE(Y)+32+16+4
0120      IF (NCOL(4).EQ.QDR) GOTO 500
0122      GOTO 696
0123  320   IF (NCOL(3).EQ.QSQ) GOTO 325
          C      SET BIT 13

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:22:07

PAGE 003

LOGIC.OBJ=LOGIC.FOR/NOSN/LI:1

```
0125     CODE(Y)=CODE(Y)+8
0126     IF (NCOL(3).EQ.QDR) GOTO 500
0128     GOTO 696
      C   SET BITS 11-13
0129 325  CODE(Y)=CODE(Y)+32+16+8
0130     IF (NCOL(4).EQ.QDR) GOTO 500
0132     GOTO 696
0133 330  IF (NCOL(3).EQ.QAQ) GOTO 335
0135     CODE(Y)=CODE(Y)+8+4
0136     IF (NCOL(3).EQ.QDR) GOTO 500
0138     GOTO 696
      C   SET BITS 11-14
0139 335  CODE(Y)=CODE(Y)+32+16+8+4
0140     IF (NCOL(8).EQ.QDR) GOTO 500
0142     GOTO 696
      C   SEARCH FOR "="-SAVE POSITION
0143 500  X=SCAN(1,QDR,0,0,0,0,M)
0144     MB=M-1
0145     T=M+1
0146     S=T+2
      C   U(T) SET AT 11 TO GET OUT OF TEST LATER
0147     U(T)=11
0148     DO 40 L=T,S
0149     IF (NCOL(L).NE.Q0Q) GOTO 21
0151     U(L)=0
0152     GOTO 40
0153 21   IF (NCOL(L).NE.Q1Q) GOTO 22
0155     U(L)=1
0156     GOTO 40
0157 22   IF (NCOL(L).NE.Q2Q) GOTO 23
0159     U(L)=2
0160     GOTO 40
0161 23   IF (NCOL(L).NE.Q3Q) GOTO 24
0163     U(L)=3
0164     GOTO 40
0165 24   IF (NCOL(L).NE.Q4Q) GOTO 25
0167     U(L)=4
0168     GOTO 40
0169 25   IF (NCOL(L).NE.Q5Q) GOTO 26
0171     U(L)=5
0172     GOTO 40
0173 26   IF (NCOL(L).NE.Q6Q) GOTO 27
0175     U(L)=6
0176     GOTO 40
0177 27   IF (NCOL(L).NE.Q7Q) GOTO 28
0179     U(L)=7
0180     GOTO 40
0181 28   IF (NCOL(L).NE.Q8Q) GOTO 29
0183     U(L)=8
0184     GOTO 40
0185 29   IF (NCOL(L).NE.Q9Q) GOTO 40
0187     U(L)=9
0188 40   CONTINUE
      D   WRITE (1,41) U(T),NCOL
      D41  FORMAT (' U(T)=' ,I2,' NCOL=' ,30A1)
0189     IF (U(T).LT.0.OR.U(T).GT.9) GOTO 509
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:22:07 PAGE 004
LOGIC.OBJ=LOGIC.FOR/NOSN/LI:1

```

0191      IF (NCOL(T).EQ.Q1Q.AND.NCOL(T+1).EQ.QER) GOTO 650
0193      IF (NCOL(T).EQ.Q0Q.AND.(NCOL(T+1).EQ.QER.OR.NCOL(T+1).EQ.QEQ))
1GOTO 600
0195      IF (NCOL(MB).NE.QBQ) GOTO 696
0197      AD=U(T)
0198      Q=T+1
0199      IF (NCOL(Q).EQ.QER) GOTO 502
0201      AD=U(T)*10+U(Q)
0202      IF (NCOL(S).EQ.QER) GOTO 502
0204      AD=U(T)*100+U(Q)*10+U(S)
0205      SS=S+1
0206      IF (NCOL(SS).NE.QER) GOTO 504
0208      IF (AD.GT.255) GOTO 504
0210 502   CODE(Y)=CODE(Y)+8+2
0211      CODE(Y)=CODE(Y)+AD*16
0212      WRITE (1,43) MPAD, CODE(Y), COL
0213      MPAD=MPAD+1
0214      Y=Y+1
0215      RETURN
0216 504   CALL WRT(40,2)
0217      RETURN
0218 509   IF (NCOL(T).NE. QAQ) GOTO 550
0220      IF (NCOL(T+1) .NE. Q1Q) GOTO 551
0222      WD=1
0223      GOTO 555
0224 551   IF (NCOL(T+1) .NE. Q2Q) GOTO 552
0226      WD=2
0227      GOTO 555
0228 552   IF (NCOL(T+1) .NE. Q3Q) GOTO 555
0230      WD=3
0231 555   IF (NCOL(T+1).EQ.QMQ) GOTO 670
0233      IF (WD.LT.1.OR.WD.GT.3) GOTO 508
0235      GOTO (510,520,530) WD
0236 508   CALL WRT(33,2)
0237      RETURN
0238 C     SET BIT 6
0239 510   CODE(Y)=CODE(Y)+1024
0239      GOTO 1000
0240 C     SET BIT 5
0240 520   CODE(Y)=CODE(Y)+2048
0241      GOTO 1000
0242 C     SET BITS 5-6
0242 530   CODE(Y)=CODE(Y)+1024+2048
0243      GOTO 1000
0244 550   IF (NCOL(T).NE.Q0Q) GOTO 600
0246      IF (NCOL(2).EQ.QSQ.OR.NCOL(2).EQ.Q1Q.OR.NCOL(2).EQ.Q2Q)
1GOTO 570
0248      IF (NCOL(2).EQ.QMQ.OR.NCOL(2).EQ.QUQ) GOTO 560
0250 C     SET BITS 7,9
0251      CODE(Y)=CODE(Y)+512+128
0251      GOTO 695
0252 C     SET BITS 7,10
0252 560   CODE(Y)=CODE(Y)+512+64
0253      GOTO 695
0253 C     SET BITS 6,7,9,10
0254 570   CODE(Y)=CODE(Y)+1024+512+128+64

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:22:07

PAGE 005

LOGIC.OBJ=LOGIC.FOR/NOSN/LI:1

```
0255      GOTO 695
0256 600   IF (NCOL(T).NE. QOQ) GOTO 650
0258      X=SCAN(3,REQ,QQQ,QVQ,0,0,P)
0259      IF (X.NE.1) GOTO 610
          C   SET BIT 8
0261      CODE(Y)=CODE(Y)+256
0262      IF (NCOL(P+1).EQ.QBQ.OR.NCOL(P+1).EQ.QAQ) GOTO 695
0264      GOTO 696
0265 610   IF (NCOL(T+1).EQ.QER) GOTO 620
0267      GOTO 696
          C   SET BITS 9-10
0268 620   CODE(Y)=CODE(Y)+128+64+512+128
0269      GOTO 695
0270 650   IF (NCOL(T).NE.Q1Q.OR.NCOL(T+1).NE.QER) GOTO 670
          C   SET BIT 9
0272      CODE(Y)=CODE(Y)+128
0273      GOTO 695
0274 670   DO 680 J=T,26
0275      C=J
0276      K=J+4
0277      IF (NCOL(J).EQ.QAQ.AND.NCOL(J+1).EQ.QMQ.AND.NCOL(J+2).EQ.QPQ
0279 680   1.AND.NCOL(J+3).EQ.QCQ.AND.NCOL(J+4).EQ.QRQ)GOTO 690
          CONTINUE
0280      GOTO 750
0281 690   X=SCAN(3,QNQ,QQQ,QRQ,0,0,P)
0282      XN=SCAN(3,QNQ,QQQ,QTQ,0,0,P)
0283      IF (X.NE.1.AND.XN.NE.1) GOTO 700
0285      IF (NCOL(K-5).EQ.QRQ.OR.NCOL(K-5).EQ.QTQ) GOTO 697
0287      GOTO 696
          C   SET BITS 7,9
0288 697   CODE(Y)=CODE(Y)+512+128
0289      GOTO 695
0290 700   X=SCAN(3,QAR,Q1Q,QER,0,0,P)
0291      IF (X.NE.1) GOTO 710
0293      IF (NCOL(K+1).NE. QAR) GOTO 696
          C   SET BIT 9
0295      CODE(Y)=CODE(Y)+128
0296      GOTO 695
          C   CHECK AFTER "AMPCR"
0297 710   IF (NCOL(K+1) .NE. QER) GOTO 696
          C   SET BITS 9-10
0299      CODE(Y)=CODE(Y)+128+64
0300      GOTO 695
0301 750   IF (NCOL(T).EQ. QBQ) GOTO 760
0303      CALL WRT(18,2)
0304      RETURN
0305 760   IF (NCOL(T+1) .NE. QFQ) GOTO 800
0307      IF (NCOL(T+2) .EQ. QAR) GOTO 780
0309      IF (NCOL(T+2) .EQ. QER) GOTO 770
0311      GOTO 696
          C   SET BITS 8-10
0312 770   CODE(Y)=CODE(Y)+256+128+64
0313      GOTO 695
          C   SET BITS 8-9
0314 780   CODE(Y)=CODE(Y)+256+128
0315      IF (NCOL(T+3).EQ.Q1Q) GOTO 695
```

FORTTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:22:07

PAGE 006

LOGIC.OBJ=LOGIC.FOR/NOSN/LI:1

```
0317      GOTO 696
0318 800   IF (NCOL(T+1).EQ.QAR.AND.NCOL(T+2).EQ.Q1Q.AND.NCOL(T+3).EQ.
          1QER) GOTO 695
0320      IF (NCOL(T+1).NE. QER) GOTO 696
          C
          SET BIT 10
0322      CODE(Y)=CODE(Y)+64
0323 695   WRITE (1,43) MPAD, CODE(Y), COL
0324 43    FORMAT (1X,05,4X,05,5X,80A1)
0325      MPAD=MPAD+1
0326      Y=Y+1
0327      RETURN
0328 696   CALL WRT(34,2)
0329      RETURN
0330 1000  CALL LOGICA(T)
0331      RETURN
0332      END
```

>

FORTRAN IV V01B-02 MON 28-FEB-77 09:30:01 PAGE 001
 CORE=08K, UIC=L20,20J LOGICA.OBJ=LOGICA.FOR/NOSN/LI:1

```

0001      SUBROUTINE LOGICA(T)
0002      IMPLICIT INTEGER (A-Y)
0003      LOGICAL*1 NCOL,COL,QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,
1QLQ,QMQ,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,
200Q,Q1Q,Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,
3QFR,QGR,QHR

0004      COMMON NCOL(30)
0005      COMMON /IMP/COL(80),CODE(128)
0006      COMMON /VAX/N,MPAD,FR,Y
0007      COMMON /CODES/QAQ,QBQ,QCQ,QDQ,QEQ,QFQ,QGQ,QHQ,QIQ,QJQ,QKQ,QLQ,
1QMQ,QNQ,QOQ,QPQ,QQQ,QRQ,QSQ,QTQ,QUQ,QVQ,QWQ,QXQ,QYQ,QZQ,Q0Q,Q1Q,
2Q2Q,Q3Q,Q4Q,Q5Q,Q6Q,Q7Q,Q8Q,Q9Q,QAR,QBR,QCR,QDR,QER,QFR,QGR,QHR

      C      IF NCOL(N)=A
0008      X=SCAN(1,QFQ,0,0,0,0,Q)
      D      WRITE (1,1)T,X
      D1     FORMAT (' IN LOGICA:T=',I2,' X=',I1)
0009      IF(X.NE. 1) GO TO 1050
0011      IF (NCOL(Q-1).NE.QBQ.AND.NCOL(Q-1).NE.Q1Q.AND.NCOL(Q-1).NE.Q2Q
1.AND.NCOL(Q-1).NE.Q3Q) GO TO 696
0013      IF (NCOL(Q+1).NE.QAR.AND.NCOL(Q+1).NE.QER) GO TO 696
0015      X=SCAN(3,QAQ,QNQ,QDQ,0,0,P)
0016      IF(X.NE.1) GO TO 1010
      C      SET BITS 7-10
0018      CODE(Y)=CODE(Y)+512+256+128+64
0019      IF(NCOL(P+1).NE.QBQ) GO TO 696
0021      IF(NCOL(Q+1).NE.QER) GO TO 696
0023      GO TO 695
0024    1010 X=SCAN(2,QOQ,QRQ,0,0,0,P)
0025      IF(X.NE.1) GO TO 1020
      C      SET BITS 7-9
0027      CODE(Y)=CODE(Y)+512+256+128
0028      IF(NCOL(P+1).NE.QBQ) GO TO 696
0030      IF(NCOL(Q+1).NE.QER) GO TO 696
0032      GO TO 695
0033    1020 IF (NCOL(Q+1).NE.QER) GO TO 696
      C      SET BITS 7-9
0035      CODE(Y)=CODE(Y)+512+128
0036      GO TO 695
0037    1050 DO 5 I=T,26
0038      IX=I
0039      IF (NCOL(I).EQ.QAQ.AND.NCOL(I+1).EQ.QMQ.AND.NCOL(I+2).EQ.QPQ
1.AND.NCOL(I+3).EQ.QCQ.AND.NCOL(I+4).EQ.QRQ) GO TO 1060

0041    5      CONTINUE
0042      GO TO 2000
0043    1060 X=SCAN(3,QNQ,QOQ,QRQ,0,0,P)
0044      IF(X.NE.1) GO TO 1070
0046      IF (NCOL(IX+5).NE.QER) GO TO 696
      C      SET BITS 7,9
0048      CODE(Y)=CODE(Y)+512+128
0049      IF (NCOL(IX-1).EQ.QRQ) GO TO 695
0051      GO TO 696
0052    1070 X=SCAN(3,QNQ,QAQ,QNQ,0,0,P)
0053      IF (X.NE.1) GO TO 1080
0055      IF (NCOL(IX+5).NE.QER) GO TO 696
      C      SET BITS 7,9,10
0057      CODE(Y)=CODE(Y)+512+128+64

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:30:01
LOGICA.OBJ=LOGICA.FOR/NOSN/LI:1

PAGE 002

```

0058      IF (NCOL(IX-1).EQ.QNQ) GO TO 695
0060      GO TO 696
0061 1080  I=IX
0062      IF (NCOL(I+5) .NE.QAR.OR.NCOL(I+6).NE.Q1Q.OR.NCOL(I+7).NE.QER)
1GO TO 1095
      C    SET BIT 9
0064      CODE(Y)=CODE(Y)+128
0065      GO TO 695
0066 1095  IF (NCOL(I+5).NE.QER) GO TO 696
      C    SET BITS 9-10
0068      CODE(Y)=CODE(Y)+128+64
0069      GO TO 695
0070 2000  X=SCAN(3,QEQ,QQQ,QVQ,0,0,P)
0071      IF (X.NE.1) GOTO 2010
      C    SET BIT 8
0073      CODE(Y)=CODE(Y)+256
0074      IF (NCOL(P+1).NE.QBQ.AND.NCOL(P+1).NE.QAQ) GO TO 696
0076      IF (NCOL(P-3).NE.QOQ.AND.NCOL(P-3).NE.Q1Q.AND.NCOL(P-3).NE.Q2Q
1.AND.NCOL(P-3).NE.Q3Q) GO TO 696
0078      IF (NCOL(P+2).NE.QER.AND.NCOL(P+2).NE.QMQ) GO TO 696
0080      IF (NCOL(P+6).NE.QHR.AND.NCOL(P+6).NE.QER) GO TO 696
0082      GO TO 695
0083 2010  X=SCAN(3,QXQ,QQQ,QRQ,0,0,P)
0084      IF (X.NE.1) GO TO 2020
      C    SET BITS 8,10
0086      CODE(Y)=CODE(Y)+256+64
0087      IF (NCOL(P+1).NE.QBQ.AND.NCOL(P+1).NE.QAQ) GO TO 696
0089      IF (NCOL(P-3).NE.Q1Q.AND. NCOL(P-3).NE.Q2Q.AND.NCOL(P-3).NE.Q3Q)
1GO TO 696
0091      IF (NCOL(P+2).NE.QER.AND.NCOL(P+2).NE.QMQ) GO TO 696
0093      IF (NCOL(P+6).NE.QHR.AND.NCOL(P+6).NE.QER) GO TO 696
0095      GO TO 695
0096 2020  X=SCAN(3,QNQ,QQQ,QRQ,0,0,P)
0097      IF (X.NE.1) GO TO 2030
      C    SET BIT 7
0099      CODE(Y)=CODE(Y)+512
0100      IF (NCOL(P+1).NE.QBQ.AND.NCOL(P+1).NE.QAQ) GO TO 696
0102      IF (NCOL(P-3).NE.Q1Q.AND.NCOL(P-3).NE.Q2Q.AND.NCOL(P-3).NE.Q3Q)
1GO TO 696
0104      IF (NCOL(P+2).NE.QER.AND.NCOL(P+2).NE.QMQ) GO TO 696
0106      IF (NCOL(P+6).NE.QHR.AND.NCOL(P+6).NE.QER) GO TO 696
0108      GO TO 695
0109 2030  X=SCAN(3,QNQ,QAQ,QNQ,0,0,P)
0110      IF (X.NE.1) GOTO 2040
      C    SET BITS 7,10
0112      CODE(Y)=CODE(Y)+512+64
0113      IF (NCOL(P+1).NE.QBQ.AND.NCOL(P+1).NE.QAQ) GOTO 696
0115      IF (NCOL(P-3).NE.Q1Q.AND.NCOL(P-3).NE.Q2Q.AND.NCOL(P-3).NE.Q3Q)
1GOTO 696
0117      IF (NCOL(P+2).NE.QER.AND.NCOL(P+2).NE.QMQ) GOTO 696
0119      IF (NCOL(P+6).NE.QHR.AND.NCOL(P+6).NE.QER) GOTO 696
0121      GOTO 695
0122 2040  X=SCAN(2,QQQ,QRQ,0,0,0,P)
0123      IF (X.NE.1) GO TO 2050
      C    SET BITS 7-8
0125      CODE(Y)=CODE(Y)+512+256

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:30:01

PAGE 003

LOGICA.OBJ=LOGICA.FOR/NOSN/LI:1

```

0126      IF (NCOL(P+1).NE.QBQ.AND.NCOL(P+1).NE.QAQ) GO TO 696
0128      IF (NCOL(P-2).NE.Q1Q.AND.NCOL(P-2).NE.Q2Q.AND.NCOL(P-2).NE.Q3Q)
          1 GO TO 696
0130      IF (NCOL(P+2).NE.QER.AND.NCOL(P+2).NE.QMQ) GO TO 696
0132      IF (NCOL(P+6).NE.QHR.AND.NCOL(P+6).NE.QER) GO TO 696
0134      GO TO 695
0135 2050 X=SCAN(3,QAQ,QNQ,QDQ,0,0,P)
0136      IF (X.NE.1) GO TO 2060
          C      SET BITS 7,8,10
0138      CODE(Y)=CODE(Y)+512+256+64
0139      IF (NCOL(P+1).NE.QBQ.AND.NCOL(P+1).NE.QAQ) GO TO 696
0141      IF (NCOL(P-3).NE.Q1Q.AND.NCOL(P-3).NE.Q2Q.AND.NCOL(P-3).NE.Q3Q)
          1GO TO 696
0143      IF (NCOL(P+2).NE.QER.AND.NCOL(P+2).NE.QMQ) GO TO 696
0145      IF (NCOL(P+6).NE.QHR.AND.NCOL(P+6).NE.QER) GO TO 696
0147      GO TO 695
0148 2060 X=SCAN(3,QRQ,QIQ,QMQ,0,0,P)
0149      IF (X.NE.1) GO TO 2070
          C      SET BITS 7-9
0151      CODE(Y)=CODE(Y)+512+256+128
0152      IF (NCOL(P+1).NE.QBQ.AND.NCOL(P+1).NE.QAQ) GO TO 696
0154      IF (NCOL(P-3).NE.Q1Q.AND.NCOL(P-3).NE.Q2Q.AND.NCOL(P-3).NE.Q3Q)
          1GO TO 696
0156      IF (NCOL(P+2).NE.QER.AND.NCOL(P+2).NE.QMQ) GO TO 696
0158      IF (NCOL(P+6).NE.QHR.AND.NCOL(P+6).NE.QER) GO TO 696
0160      GO TO 695
0161 2070 X=SCAN(3-QNQ,QIQ,QMQ,0,0,P)
0162      IF (X.NE.1) GO TO 2080
          C      SET BITS 7-10
0164      CODE(Y)=CODE(Y)+512+256+128+64
0165      IF (NCOL(P+1).NE.QER.AND.NCOL(P+1).NE.QAQ) GO TO 696
0167      IF (NCOL(P-3).NE.Q1Q.AND.NCOL(P-3).NE.Q2Q.AND.NCOL(P-3).NE.Q3Q)
          1GO TO 696
0169      IF (NCOL(P+2).NE.QER.AND.NCOL(P+2).NE.QMQ) GO TO 696
0171      IF (NCOL(P+6).NE.QHR.AND.NCOL(P+6).NE.QER) GO TO 696
0173      GO TO 695
0174 2080 IF (NCOL(T+2).NE.QAR.OR.NCOL(T+3).NE.Q1Q.OR.NCOL(T+4).NE.QER)
          1GO TO 2100
          C      SET BIT 9
0176      CODE(Y)=CODE(Y)+128
0177      GO TO 695
0178 2100 IF (NCOL(T+2).NE.QER.AND.NCOL(T+2).NE.QHR) GO TO 2110
          C      SET BITS 9-10
0180      CODE(Y)=CODE(Y)+128+64
0181      GO TO 695
0182 2110 IF (NCOL(T+2).NE.QBR.OR.NCOL(T+3).NE.QBQ) GO TO 2160
0184      IF (NCOL(T+4).NE.QBR.OR.NCOL(T+5).NE.Q1Q.OR.NCOL(T+6).NE.QER)
          1GO TO 2140
          C      SET BITS 8-10
0186      CODE(Y)=CODE(Y)+256+128+64
0187      GO TO 695
0188 2140 IF (NCOL(T+4).NE.QER) GO TO 696
          C      SET BITS 8-9
0190      CODE(Y)=CODE(Y)+256+128
0191      GO TO 695
0192 2160 IF (NCOL(T+2).NE.QAR.OR.NCOL(T+3).NE.QBQ) GO TO 696

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

MON 28-FEB-77 09:30:01

PAGE 004

LOGICA.OBJ=LOGICA.FOR/NOSN/LI:1

```
0194            IF (NCOL(T+4).EQ.QAR.AND.NCOL(T+5).EQ.Q1Q.AND.NCOL(T+6).EQ.QER).  
              1GO TO 695  
0196            IF (NCOL(T+4).EQ.QER) GO TO 2180  
0198    696      CONTINUE  
0199            CALL WRT(34,2)  
0200            RETURN  
              C    SET BIT 10  
0201    2180     CODE(Y)=CODE(Y)+64  
0202    695      CONTINUE  
0203            WRITE (1,699) MPAD, CODE(Y), COL  
0204    699      FORMAT (1X,05,4X,05,5X,80A1)  
0205            MPAD=MPAD+1  
0206            Y=Y+1  
0207            RETURN  
0208            END
```

>

1.2 M1710 Common Area

Communication between a host PDP-11 processor and an ESM loop CIE microprocessor is accomplished through the use of an M1710 General Purpose Interface board. A packet (256 bytes) can be transferred between the two machines in the form of 16-bit words at a maximum rate of 560 Kilobaud. A more detailed description of loop-host interfacing can be found in the ESM Hardware Maintenance Manual.

PDP-11 Software interfacing is accomplished by six registers which are provided by the M1710 board. The parameters of these six registers are tabulated below:

<u>Register Address</u>	<u>Array Element</u>	<u>Functional Description</u>
761000	DEVST(1)	Read Data Word
761002	DEVST(2)	Write Data Word
761004	DEVST(3)	Read Input Buffer Status
761006	DEVST(4)	Read Output Buffer Status
761010	DEVST(5)	Clear Input Buffer Status
761012	DEVST(6)	Clear Output Buffer Status

When a packet is to be read by the PDP-11, the input buffer status register (DEVST(3)) is odd. The packet is then read a word at a time using DEVST(1). Since the interface operates at a 560 Kilobaud rate, a timing loop must be executed between word reads accomplished by a null DO loop from I=1 to 3 or more. A total of 129 word reads are performed with the first word read ignored. Consecutive bytes are stored in the 16-bit words in the order least significant (right), most significant (left) byte. After the packet is read DEVST(5) clears the Input Buffer Status Register to zero. Writing a packet to the loop is accomplished in a similar manner except DEVST(4) is odd when the output buffer is empty, DEVST(2) is used for 128 word writes, and DEVST(6) is used to clear the Output Buffer Status Register.

Application programs communicate with the six M1710 board registers via a COMMON BLOCK DATA program which is installed into a DEVICE type partition M1710. The procedure for generating this interface is listed below:

- a) Create the M1710 DEVICE partition using the MCR command,
SET /MAIN = M1710:7610:1:DEV

Note: It may be necessary to first remove any other partitions that overlap the memory space 761000-761100 using the SET /NOMAIN command.

b) Compile the M1710.FOR BLOCK DATA program listed below (M1710.FOR and M1710.OBJ both reside on ESM Tape #1).

c) Build the M1710 task and symbol table into UIC [1,1] using the task builder utility (TKB).

```
TKB [1,1]M1710.TSK/PI, TT0:/SH, DPO:[1,1]M1710.STB/-HD
    = [20,20]M1710.OBJ
```

Note: For host processor B use DKO rather than DPO.

Enter Options:

```
STACK=0
UNITS=0
PAR= M1710:0:100
```

d) Install the M1710 task into the M1710 partition using the MCR command,

```
INS [1,1]M1710/PAR=M1710
```

e) For application programs interfacing to the ESM loop, include the FORTRAN statements

```
INTEGER DEVST(6)
COMMON/M1710/DEVST
```

and use the TKB option when building the task,

```
COMMON=M1710:RW
```

Burroughs Corporation

V01B-02
SIC=(20,20)

WED 09-MAR-77 18:17:27

PAGE 001

M1710. OBJ=M1710. FOR/NOSN/LI:1

BLOCK DATA
INTEGER DEVST(6)
COMMON /M1710/DEVST
END

1.3 User Language

The ESM User Language provides the Host-CRT dialogue described in Section 4.4 of the ESM User Manual. The User Language consists of a main program module (P0000) and ten subroutine modules (P1000, P2000, P3000, P3001, P4000, P4001, P5000, RDLOOP, WRLOOP, HST) residing on ESM Tape #1. The User Language Task is contained in UIC [20,20], and it exists in two forms: USRLN5.TSK for processor B loop 2, and USRLN1.TSK for processor A loop 1. The above listed modules are used for USRLN5. Modules P00001, P10001, P40001, P40011, and HST1 are used for USRLN1. Modules P2000, P3000, P3001, RDLOOP, and WRLOOP are the same for the two processors. The differences result primarily from the different addresses used in the LID pair header word (ICODE(3)). Header control character format is given in Table 5-1 of the ESM User Manual. Other differences are found in the main program module (P0000). Processor A uses the DECSCOPE designated as TTO: for the message log while Processor B uses DESCOPE TTL:. The message log contains the header and first two information bytes of all packets into the host processor and all dialogue messages destined to terminals. The input messages are displayed as octal 16-bit words with the first arriving byte stored as least significant (e.g., D2 D1 D4 D3 D6 D5 D8 D7).

The program normally waits for an input packet from the loop while checking the input buffer status residing in the M1710 common area. When the packet is received the program passes control to the proper module depending on the status of the dialogue for the terminal that sent the packet. The input is processed and responses are formatted and sent to the loop to prompt the sending terminal and provide system control functions when necessary.

The User Language is divided into five modes of operation:

1. CRT-to-CRT (P1000)
2. System Inquiry (P2000) : Operates on system file INFO.DAT.
3. System Control (P3000, P3001): Operates on system file INFO.DAT.
4. File Access (P4000, P4001, HST): Uses directory file EFDIR, and ATEC simulation files EFLOCF, EFTRKD, EFTERD, EFCKTD. Implements a distributed file system where records of a file are distributed between host processors A and B.
5. Card Format (P5000): Not yet implemented.

Host processor A does not have access to modes 2 and 3 so that system control update changes to INFO.DAT are only stored on processor B. Mode 5 on processor B contains a CRT broadcast demonstration where all terminals receive a common packet. The file MSG.DAT is used for holding 80 character records that are used for terminal displays.

AD-A063 407

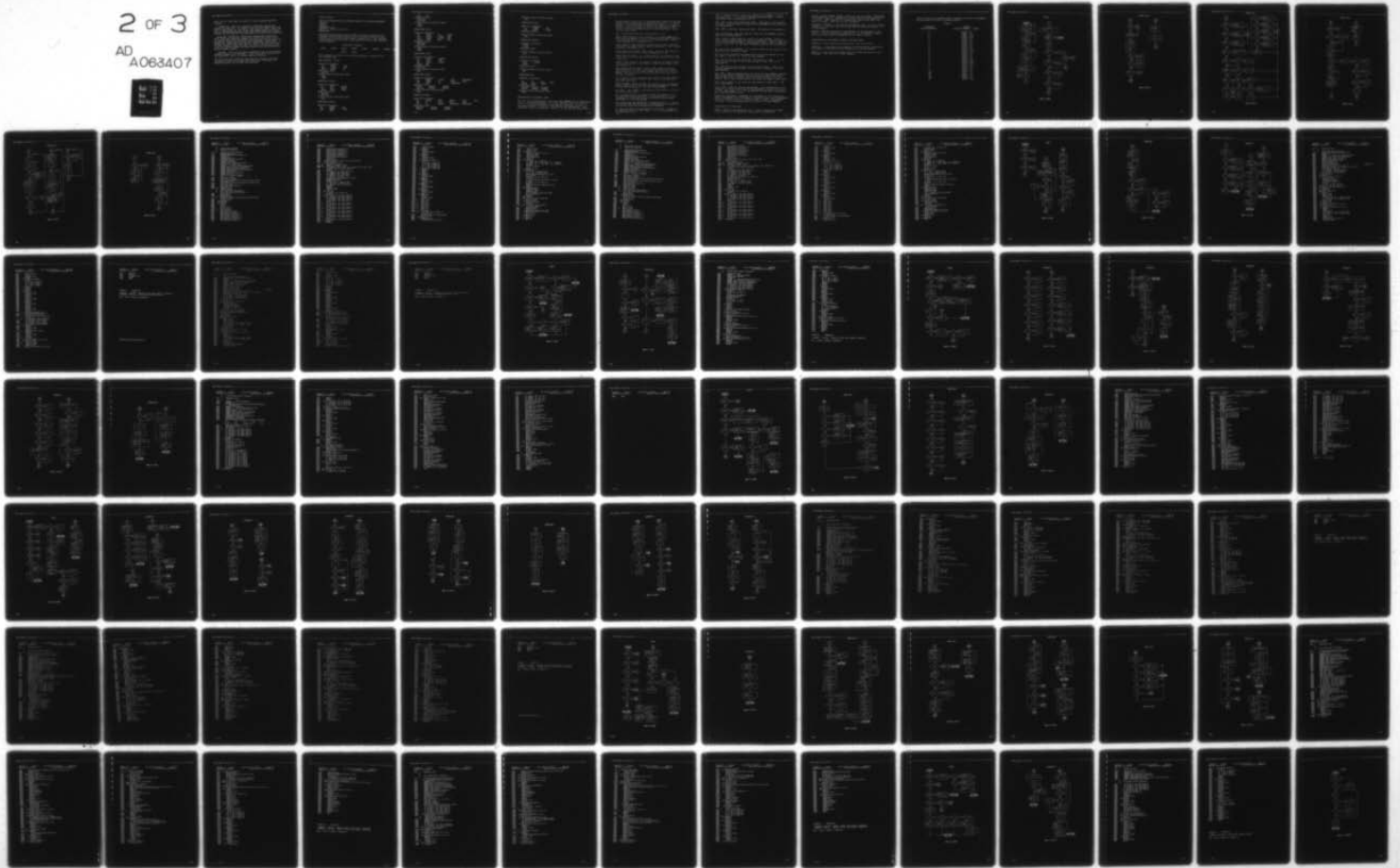
BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 17/2
EXPLORATORY SYSTEMS CONTROL MODEL (ESM). BOOK 1. FORTRAN. SOFTW--ETC(U)
APR 77 DCA100-75-C-0054
NL

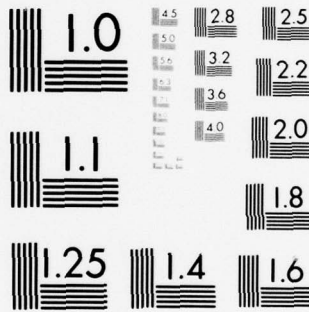
UNCLASSIFIED

66143-3

2 OF 3

AD
A063407





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Other files on ESM Tape #1 related to User Language operation include:

- INFOPM.OBJ - This file contains a permanent copy of the system file INFO.DAT. To rebuild this file move INFOPM.OBJ from tape to disk using the FLX FB:80. fixed binary option, rename the file to INFOPM.DAT and then invoke the STESM command file which deletes the old INFO.DAT file and makes a new copy from INFOPM.DAT.

-EFDIR.OBJ, EFTERD.OBJ, EFCKTD.OBJ, EFTRKD.OBJ, EFLOCF.OBJ - These files are permanent copies of the ATEC simulation files. To rebuild these files move EFDIR.OBJ from tape to disk using the FLX FB:200. fixed binary option, and EFTERD.OBJ, EFCKTD.OBJ, EFTRKD.OBJ, EFLOCF.OBJ using FB:240. When files have been moved to disk, rename to drop the .OBJ, and build a consistent distributed file system without multiple copies using the record move utility (RCMV1, RCMV5).

- MSG.OBJ - This file contains a permanent copy of the message terminal display file (MSG.DAT). This file may be edited using the RXS11M EDI utility to obtain modified displays.

The following pages describe task building; overlay structure; variables, files, and library functions used; flowcharts and program listings for the ESM User Language.

Task Building:

```
TKB[20,20]USRLNG.TSK=[20,20]USROVL.ODL/MP,[1,1]SYSLIB/LB:$SHORT
Options:
UNITS=8
ACTFIL=8
COMMON=M1710:RW
ASG=TT1:1, SY0:2:3:4:5:6:7:8
MAXBUF=240
```

The RSX11M task builder utility (TKB) is used to build the User Language task from the object files of the main program and sub-routines and the overlay description language file. The commands listed above build the task. Shown below is the overlay structure of the program.

```

                                P0000(main program)
P1000      P2000      P3000      P4000      P5000      RDLOOP      WRLOOP      HS
                                P3001      P4001
```

This structure is coded in the overlay descriptor language file [20,20]USROVL.ODL.

MAIN PROGRAM - P0000:

Important Variables:

```
ST          NOREC          I1ST
IND          ICODE          ICON
DEVST       NRCNO          NN
```

Important Files:

```
MSG.DAT
```

Fortran Library Functions Used:

```
IAND
ISHFT
```

SUBROUTINE P1000:

Important Variables:

```
ST          MOUT          NRCNO
IND          MSK          NOREC
ICODE       ICFLG        ICON
```

Important Files:

```
MSG.DAT
INFO.DAT
```

Fortran Library Functions Used:

```
IAND
```

SUBROUTINE P2000:

Important Variables:

```
ST          NRCNO          NDI
IND          NOREC          ICODE
ND          MOUT
```

Burroughs Corporation

LSK NR
Important Files:
INFO.DAT
MSG.DAT
Fortran Library Functions Used:
IAND
ISHFT

SUBROUTINE P3000:

Important Variables:
ST MOUT NR ICFLG
IND NRCNO ICHAR INFA
LSK NOREC LIDTBD INV
ICODE ICON MSK NEWV
Important Files:
INFO.DAT
MSG.DAT
Fortran Library Functions Used:
IAND
ISHFT

SUBROUTINE P3001:

Important Variables:
ST NRCNO NDWPM
IND LSK NIM
NR ICODE ICHAR
NOREC MSK
Important Files:
INFO.DAT
Fortran Library Functions Used:
IAND

SUBROUTINE P4000:

Important Variables:
ST NRNCO ICODE OFIL KEYTYPEFM
IND NOREC DI MOUT IFAC
IFS LSK LU NOCHARKEY
Important Files:
EFDIR EFLOCF EFTRKD
MSG.DAT EFCKTD EFTERD
Fortran Library Function Used:
IAND

SUBROUTINE P4001:

Important Variables:
ST ICODE MOUT NOREC LINE IFAC
IND DI OFIL LSK IFAR
IFS LU NRCNO NOCHARKEY KEYTYPEFM
Important Files:
MSG.DAT EFLOCF EFTRKD
EFDIR EFCKTD EFTERD

Fortran Library Functions Used:
IAND

SUBROUTINE P5000:

Important Variables:
ST NOREC IND
LSK NRCNO ICODE

Important Files:
- - - - -

Fortran Library Functions Used:
IAND

SUBROUTINE RDLOOP:

Important Variables:
DEVST
ICODE

Important Files:
- - - - -

Fortran Library Functions Used:
MOD

SUBROUTINE WRLOOP:

Important Variables:
ILST ICODE
DEVST

Important Files:
- - - - -

Fortran Library Functions Used:
MOD

SUBROUTINE HST:

Important Variables:

ST DI OFIL MOUT
NRCNO NOREC ICODE

Important Files:
MSG.DAT EFLOCF EFTRAD
EFDIR EFCATD EFTERD

Fortran Library Functions Used:
IAND

DESCRIPTION OF VARIABLES USED:

ST - ST is the status array. ST is of type INTEGER and is dimensioned (3,9). The first parameter of ST (IND) indicates the user (CRT #1, 2 or 3). The second contains a maximum of 9 statuses. Status 1 is the module status. Status 2 is the CRT-CRT node designator. Only values of 4 and 8 are allowed. Status 3 is the type of system infor-

mation desired. Values of 1 - 4 are permitted. Status 4 is the node designator for modules P3000 and P3001 with values of 1 - 11 allowed. Status 5 is the network device parameter with values of 1 - 4 permissible. Status 6 contains the "KEY" to be checked in the directory. Status 7 is the record number of the file with the correct "KEY". Statuses 8 and 9 are different types of cross-referencing. Values of 1 to 3 are allowed.

IND - IND identifies the user as being CRT #1, 2 or 3.

DEVST - DEVST contains the six registers of the added COMMON area M1710 which corresponds to the M1710 Interface Board Memory Partition. This allows the PDP-11 to talk to the B7* microprocessor. DEVST is of type INTEGER and is dimensioned at 6.

ICODE - ICODE is formed by RDLOOP by getting the values of DEVST(1) from the B7* interface buffer. ICODE is dimensioned at 128. ICODE(3) is used for the LOGICAL I.D.'s.

ICON - ICON is the control packet array. It is put into ICODE and is written out to the loop. ICON is dimensioned at 128.

ICFLG - ICFLG is either 0 or 1. When equal to 1 it indicates to the modules that ICON exists and there is a control packet to be written to the loop.

ICHAR - ICHAR is used in the process of changing the header (ICODE(1 - 4)) from type INTEGER to type REAL*8. ICHAR is type INTEGER and is dimensioned at 4.

NEWV, LIDTBD, LDNFAD, NDWPMD - NEWV, LIDTBD, LDNFAD are used in P3000 and NDWPMD in P3001. They are all equivalent with ICHAR. They receive the value of ICHAR (INTEGER type) and hold it as type REAL*8. They hold the first 4 words that were sent by the loop to the PDP-11 processor.

ILST - ILST is sent as a parameter from -0000 to the module WRLOOP. It is equal to 0 before the first PDP-11 write to the loop and 1 after the first write.

NRCNO and NOREC - NRCNO and NOREC are pointers to the file MSG.DAT. Certain combinations of these two variables correspond to messages that are sent to the CRT's. Both are of type INTEGER.

NN - NN is used in P0000 as the pointer to the records to be read in from MSG.DAT into ICODE.

LU - LU is the logical unit number variable corresponding to the file that MOUT is to be written to. LU is equal to 3, 4, 5, or 6 respectively corresponding to the files EFLOCF, EFCKTD, EFTRKD, and EFTERD. LU is of type INTEGER.

MOUT - MOUT is of type REAL*8 and is dimensioned (10, 11). Records are read from MSG.DAT and INFO.DAT into MOUT and MOUT is written to the different files and eventually to the DECSCOPE.

DI - DI is the queue for the directory of the files. In P4000, DI is checked to see if the "KEY" exists. DI is of type INTEGER and is dimensioned at 100.

OFIL - Records from the files with logical unit numbers 3 -6 are written in OFIL. It is used on modules P4000 and P4001. It is of type REAL*8 and is dimensioned (10, 3).

LSK - LSK is the least significant mask. LSK equals a bit pattern 01111111. LSK and ICODE are used with the FORTRAN function IAND for bit manipulation.

MSK - MSK is the most significant mask. MSK equals a bit pattern of

0111111100000000. MSK and ICODE are used with the FORTRAN function IAND for bit manipulation.

IFAC - IFAC is dimensioned at 3 and is of type INTEGER. IFS is a local variable which holds the value of IFAC(IND). IFAC(1) holds the file to be accessed by CRT#1. Permissible values of IFS (or IFAC (1, 2 or 3)) are 1 -4. If the value is greater than 4, IFS is defaulted to 4.

ND - ND is of type INTEGER. It is used in P2000 as the value of the anding of LSK or MSK and ICODE.

NR - NR is the variable corresponding to the record number of the file MOUT is written to. NR is of type INTEGER.

NDI - NDI is equal to the hollerith equivalent of "NDI ". In P2000, a test is made to see if ND is equal to NDI. NDI is of type REAL*8.

INV - INV aids in the creation of the control packet. INV is the decimal representation of INVH, INVH and INVO combined. INV is of type INTEGER.

NFA, INFA - INFA is dimensioned at 4 and is of type INTEGER. Because INFA is equivalent with NFA, whatever value that goes into INFA is also stored in NFA. However, the value stored in NFA is of type REAL*8. NFA is then tested against the input and output array, MOUT.

NIM - NIM is equal to the hollerith equivalent of "NOT IMPL". NIM is of type REAL*8.

LINE, IFAR - LINE and IFAR are EQUIVALENT. LINE, dimensioned at 40, receives its values from ICODE as type INTEGER, transfers its value to IFAR, dimensioned at 10, as type REAL*8 through their equivalence, OFIL receives IFAR's values.

NOCHARKEY, KEYTYPEPFM - NOCHARKEY and KEYTYPEPFM are for display purposes. NOCHARKEY tells the number of characters per key and KEYTYPEPFM tells the form the key is in (either letter - digit, alpha-numeric or digit). Both are dimensioned at 10 and are of type INTEGER.

DESCRIPTION OF FILES USED:

EFDIR - EFDIR is the directory file. It has a logical unit number of 2, 10 records, 100 words per record and is unformatted.

EFLOCF, EFCKTD, EFTRKD, EFTERD - These are the files ATEC - SIMULATION that EFDIR refers to. Respectively, their logical unit numbers are 3, 4, 5 and 6. Each contains 100 records with each record being 120 words in length. Each file is also formatted.

INFO.DAT - INFO.DAT is the system information file. It has a logical number of 7. The file has 396 records with 40 words per record. It is unformatted.

MSG.DAT - MSG.DAT contains all the messages to be displayed to the CRT's. It has a logical unit number of 8. It contains 211 records with each record being unformatted and 40 words in length.

DESCRIPTION OF THE FORTRAN LIBRARY FUNCTIONS USED:

IAND(m,n) - This function logically ands m and n bit by bit.

ISHFT(m,n) - m designates the argument to be shifted and n specifies the number of positions and the direction m is to be shifted.

MOD(I,J) - This function divides I by J and gives the value of the remainder. The value is of type INTEGER.

Table B gives the flowchart letter connectors and their corresponding positions in the User Language program.

TABLE B

FLOWCHART LETTER CONNECTOR	PROGRAM LOCATION (MODULE - LABEL)
A	P0000 - 25
B	P0000 - 12
C	P0000 - 250
D	P0000 - 330
G	P1000 - 14
H	P1000 - 53
I	P1000 - 54
J	P1000 - 30
K	P2000 - 53
L	P2000 - 72
M	P2000 - 51
N	P3000 - 102
O	P4000 - 22
P	P4000 - 23
Q	P4000 - 24
R	P4000 - 25
T	P4000 - 575
U	P4000 - 580
V	P4000 - 585
W	P4001 - 28
X	P4001 - 29
Y	P4001 - 30
Z	P4001 - 31
AA	P4001 - 32
BB	P4001 - 34
CC	P5000 - 21
DD	P5000 - 24
EE	P3000 - 250

P0000

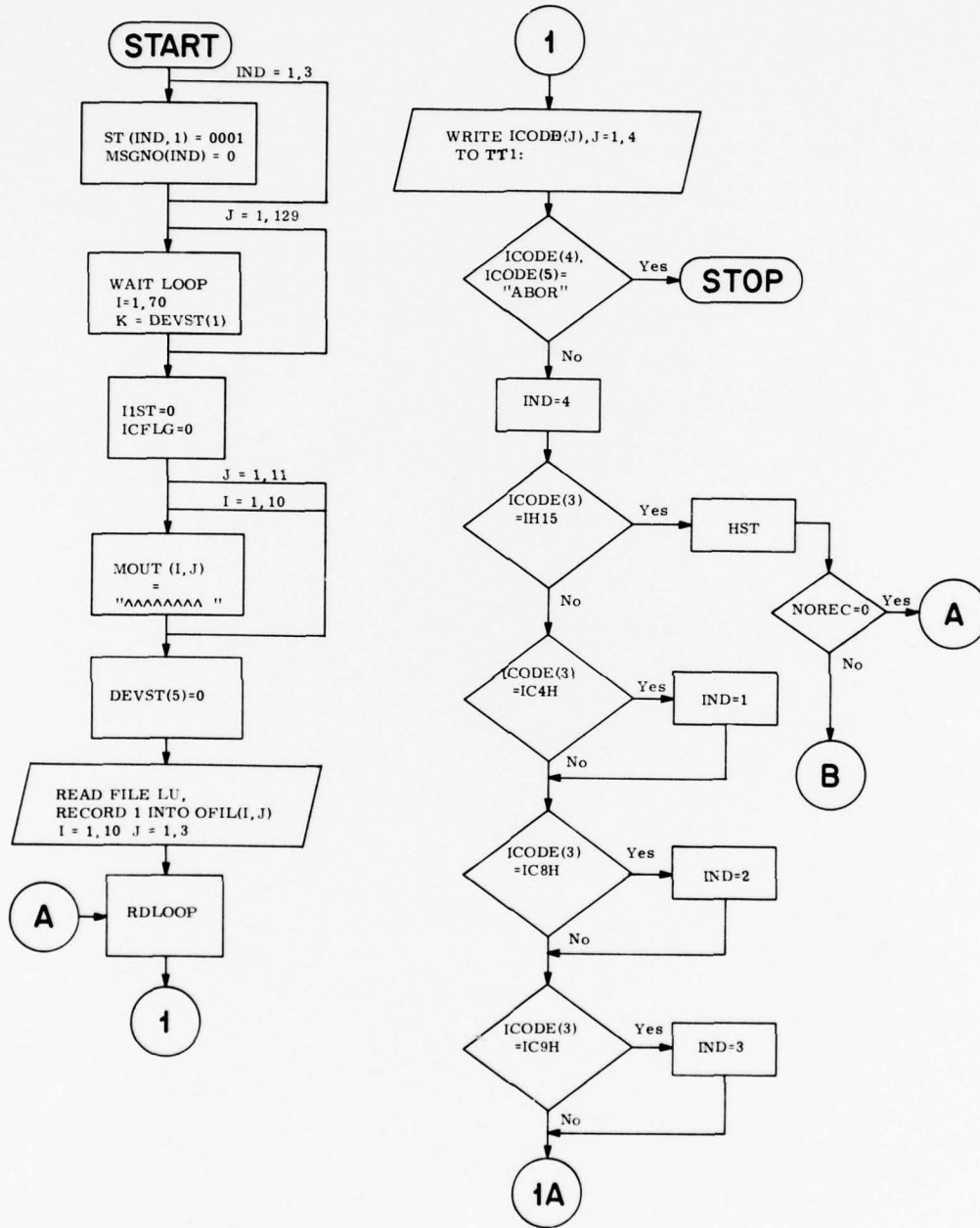


Figure 1-9. P0000

P0000 (cont.)

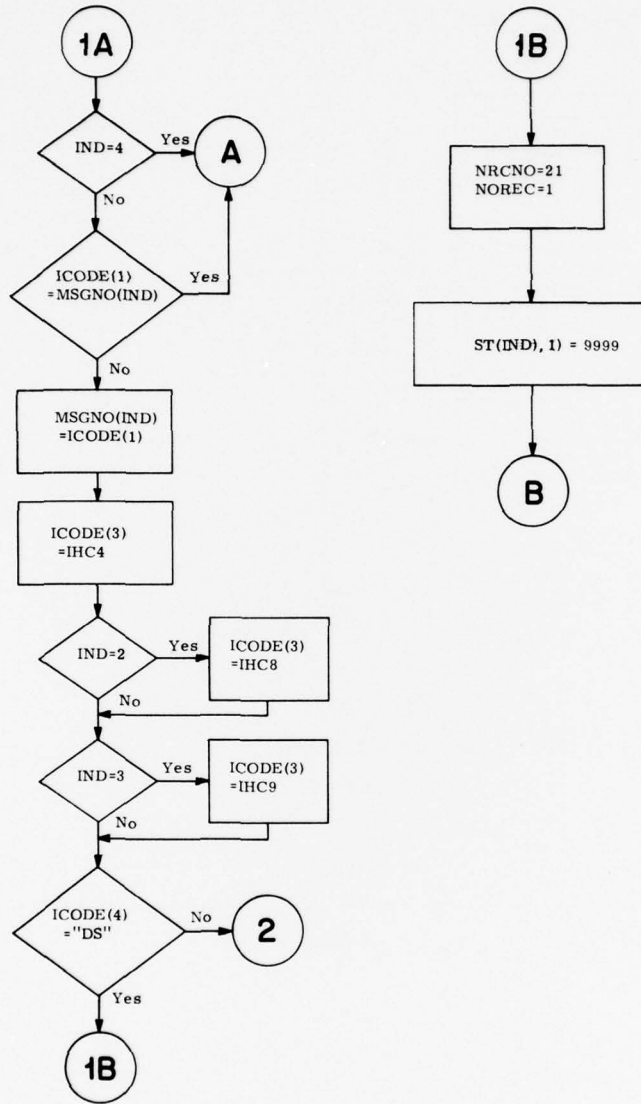


Figure 1-9. (Cont.)

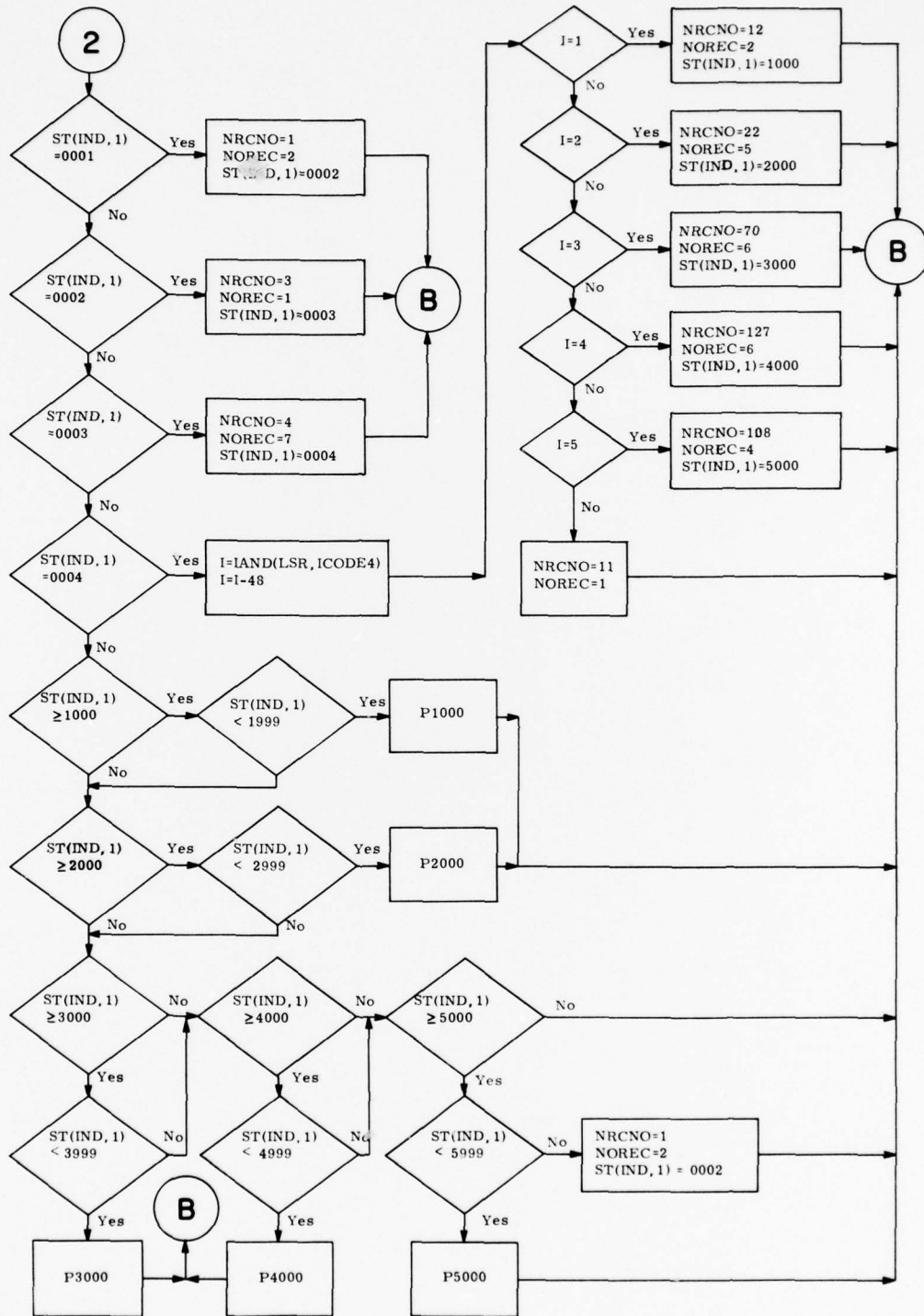


Figure 1-9. (Cont.)

POOOO (cont.)

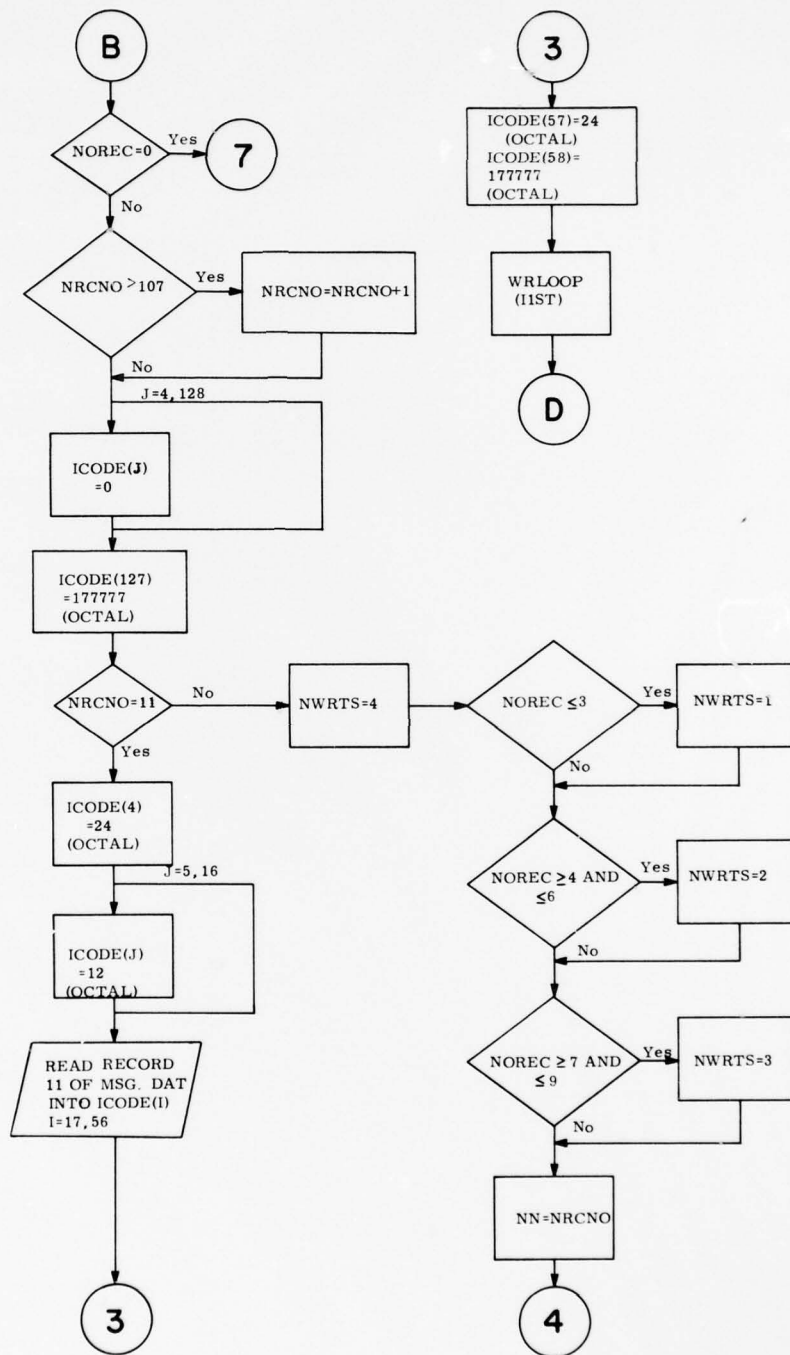


Figure 1-9. (Cont.)

P0000 (cont.)

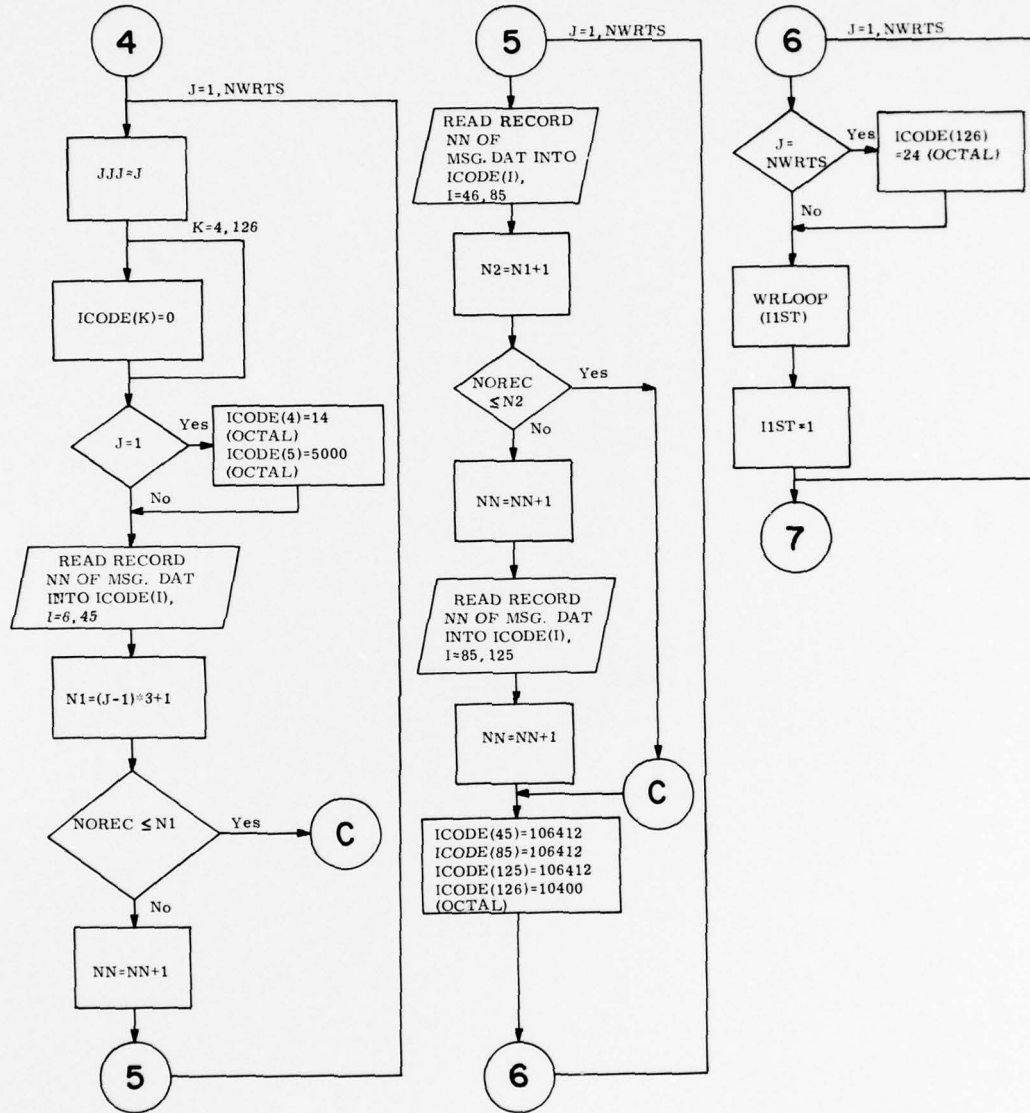


Figure 1-9. (Cont.)

P0000 (cont.)

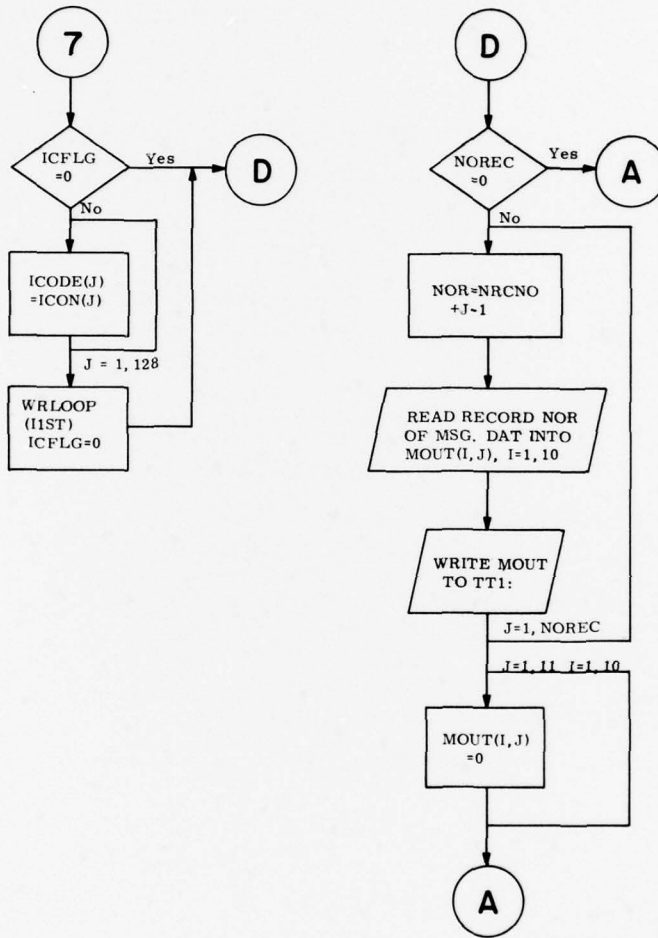


Figure 1-9. (Cont.)

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 18:40:56

PAGE 001

P0000,OBJ=P0000,FOR/NOSN/DE/LI:1

```

C
C   PRELIM USER LANG PROG
C   WRITTEN IN FORTRAN IV
C
0001  REAL*8 NDI,LID,TAB,NWD,MOUT
      1,SK8,CRTINMSG,ND
0002  REAL*8 Q2Q,Q4Q,KEYTYPEFM,OFIL,DI
0003  INTEGER ST
0004  INTEGER DEVST(6)
0005  DIMENSION OFIL(10,3),DI(100)
0006  DIMENSION MSGNO(3)
0007  COMMON /M1710/DEVST
0008  COMMON /LOOP/ ICODE(128),MSK,LSK
0009  DATA MSK,LSK/'77400','177/
0010  COMMON      NDI,LID,TAB,NWD,ST(3,9),IND
0011  COMMON /U000/ MOUT(10,11),NRCNO,NOREC
0012  COMMON /DSK/I1,I2,I3,I4,I5,I6,I16
0013  COMMON /MD04/NOCHARKEY(10),KEYTYPEFM(10)
0014  COMMON /F01/IFAC(3)
0015  COMMON /CPAC/ICON(128),ICFLG
0016  COMMON /S230/ ND
0017  DATA NDI,LID,TAB,NWD,SK8/'NDI      ',
      1'LID      ','TAB      ','NNWD      ','      '/
0018  DATA ISPC /'120240/
0019  DATA ICRLF /'106412/
0020  DATA LF,LHOME,LEDP,LFF4,LFF5,IDC1/'12','24','177777',
      1'14','5000','10400/
0021  DATA IC4H,IHC4,IC3H,IHC8/'2005','2404','4005','2410/
0022  DATA ILO/'51504/
0023  DATA IC9H,IHC9/'4405','2411/
0024  DATA IH15/'405/
0025  15  FORMAT(1X,10A8)
      C   START DIALOGUE
      C   IND IS THE PAIR INDEX NUMBER
      C   ST IS THE STATUS ROUTINE # ARRAY
0026  DO 20 IND=1,3
0027  MSGNO(IND)=0
0028  20  ST(IND,1)=0001
      C   THROW AWAY PACKET TO HAVE B7* INP BUF INIT EMPTY
0029  DO 130 J=1,129
0030  DO 140 I=1,70
0031  140 CONTINUE
0032  130 K=DEVST(1)
0033  I1ST=0
0034  ICFLG=0
0035  DO 22 J=1,11
0036  DO 22 I=1,10
0037  22  MOUT(I,J)=SK8
0038  CALL ASSIGN(1,'TT1:')
0039  CALL ASSIGN (2,'EFDIR')
0040  DEFINE FILE 2(10,100,U,I1)
0041  CALL ASSIGN(3,'EFLOCF')
0042  DEFINE FILE 3(100,120,U,I2)
0043  CALL ASSIGN(4,'EFCKTD')
0044  DEFINE FILE 4(100,120,U,I3)
0045  CALL ASSIGN(5,'EFTRKD')

```

FORTRAN IV V01B-02 FRI 18-MAR-77 18:40:56 PAGE 002
 CORE=08K, UIC=[20,20] P0000.OBJ=P0000.FOR/NOSN/DE/LI:1

```

0046     DEFINE FILE 5(100,120,U,I4)
0047     CALL ASSIGN(6,'EFTERD')
0048     DEFINE FILE 6(100,120,U,IS)
0049     CALL ASSIGN(7,'INFO.DAT')
0050     DEFINE FILE 7(396,40,U,I6)
0051     CALL ASSIGN(8,'MSG.DAT')
0052     DEFINE FILE 8(211,40,U,I16)
0053     DEVST(5)=0
0054     DO 23 LU=3,6
0055     23 READ(LU,1,ERR=17)((OFIL(I,J),I=1,10),J=1,3)
0056     25 CALL RDLOOP
      C   CALC IND, FORM LIDS
      D   WRITE(1,16)(ICODE(J),J=1,4)
0057     D16  FORMAT(1X,'HEADER= ',408)
0058     IF (ICODE(4) .EQ. '41101' .AND. (ICODE(5) .EQ. '51117' .OR.
0059     1 ICODE(5) .EQ. '151317')) GOTO 500
0061     IND=4
0062     IF (ICODE(3) .EQ. IH15) GOTO 700
0064     IF (ICODE(3) .EQ. IC4H) IND=1
0066     IF (ICODE(3) .EQ. IC8H) IND=2
0068     IF (ICODE(3) .EQ. IC9H) IND=3
0070     IF (IND .EQ. 4) GOTO 25
0072     IF (ICODE(1) .EQ. MSGNO(IND)) GOTO 25
0074     MSGNO(IND)=ICODE(1)
0075     ICODE(3)=IHC4
0076     IF (IND .EQ. 2) ICODE(3)=IHC8
0078     IF (IND .EQ. 3) ICODE(3)=IHC9
0080     IF (ICODE(4) .NE. ILO) GOTO *630
0082     NRCNO=21
0083     NOREC=1
0084     ST(IND,1)=9999
0085     GOTO 12
0086     700  CALL HST
0087     IF (NOREC .EQ. 0) GOTO 25
0089     GOTO 12
0090     630  IF (ST(IND,1) .EQ. 0001) GOTO 51
0092     IF (ST(IND,1) .EQ. 0002) GOTO 52
0094     IF (ST(IND,1) .EQ. 0003) GOTO 53
0096     IF (ST(IND,1) .EQ. 0004) GOTO 54
0098     IF (ST(IND,1) .GE. 1000) GOTO 1
0100     GOTO 2
0101     1  IF (ST(IND,1) .LE. 1999) GOTO 61
0103     2  IF (ST(IND,1) .GE. 2000) GOTO 3
0105     GOTO 4
0106     3  IF (ST(IND,1) .LE. 2999) GOTO 62
0108     4  IF (ST(IND,1) .GE. 3000) GOTO 5
0110     GOTO 6
0111     5  IF (ST(IND,1) .LE. 3999) GOTO 63
0113     6  IF (ST(IND,1) .GE. 4000) GOTO 7
0115     GOTO 8
0116     7  IF (ST(IND,1) .LE. 4999) GOTO 64
0118     8  IF (ST(IND,1) .GE. 5000) GOTO 9
0120     GOTO 12
0121     9  IF (ST(IND,1) .LE. 5999) GOTO 65
0123     51  NRCNO=1
0124     NOREC=2

```

FORTRAN IV V01R-02
CORE=08K, UIC=[20,20]FRI 18-MAR-77 18:40:56 PAGE 003
P0000.OBJ=P0000.FOR/NOSN/DE/LI:1

```
0125     ST(IND,1)=0002
0126     GOTO 12
0127     52 NRCNO=3
0128     NOREC=1
0129     ST(IND,1)=0003
0130     GOTO 12
0131     53 NRCNO=4
0132     NOREC=7
0133     ST(IND,1)=0004
0134     GOTO 12
0135     54 I=IAND(LSK,ICODE(4))
0136     I=I-48
0137     IF (I .EQ. 1) GOTO 101
0139     IF (I .EQ. 2) GOTO 102
0141     IF (I .EQ. 3) GOTO 103
0143     IF (I .EQ. 4) GOTO 104
0145     IF (I .EQ. 5) GOTO 105
0147     17 NRCNO=11
0148     NOREC=1
0149     GOTO 12
0150     101 NRCNO=12
0151     NOREC=2
0152     ST(IND,1)=1000
0153     GOTO 12
0154     102 NRCNO=22
0155     NOREC=5
0156     ST(IND,1)=2000
0157     GOTO 12
0158     103 NRCNO=70
0159     NOREC=6
0160     ST(IND,1)=3000
0161     GOTO 12
0162     104 NRCNO=127
0163     NOREC=6
0164     ST(IND,1)=4000
0165     GOTO 12
0166     105 NRCNO=108
0167     NOREC=4
0168     ST(IND,1)=5000
0169     GOTO 12
0170     61 CALL F1000
0171     GOTO 12
0172     62 CALL F2000
0173     GOTO 12
0174     63 CALL F3000
0175     GOTO 12
0176     64 CALL F4000
0177     GOTO 12
0178     65 CALL F5000
0179     12 IF (NOREC .EQ. 0) GOTO 710
C       WRITE TO LOOP
0181     IF (NRCNO .GE. 107) NRCNO=NRCNO+1
0183     DO 200 J=4,128
0184     200 ICODE(J)=0
0185     ICODE(127)=LEOP
0186     IF (NRCNO .NE. 11) GOTO 210
```

FORTRAN IV V01B-02 FRI 18-MAR-77 18:40:56 PAGE 004
 CORE=08K, UIC=[20,20] P0000.OBJ=P0000.FOR/NOSN/DE/LI:1

```

0188      ICODE(4)=LHOME
0189      DO 220 J=5,16
0190      220 ICODE(J)=LF
0191      READ(8'11)(ICODE(I),I=17,56)
0192      ICODE(57)=LHOME
0193      ICODE(58)=LEOP
0194      CALL WRLOOP(I1ST)
0195      GOTO 330
0196      210 NWRTS=4
0197      IF (NOREC .LE. 3) NWRTS=1
0199      IF (NOREC .GE. 4 .AND. NOREC .LE. 6) NWRTS=2
0201      IF (NOREC .GE. 7 .AND. NOREC .LE. 9) NWRTS=3
0203      NN=NRCNO
0204      DO 310 J=1,NWRTS
0205      JJJ=J
0206      DO 450 K=4,126
0207      450 ICODE(K)=0
0208      IF (J .EQ. 1) ICODE(4)=LFF4
0210      IF (J .EQ. 1) ICODE(5)=LFF5
0212      READ(8'NN,ERR=500)(ICODE(I),I=6,45)
0213      N1=(J-1)*3+1
0214      IF (NOREC .LE. N1) GOTO 250
0216      NN=NN+1
0217      READ(8'NN,ERR=500)(ICODE(I),I=46,85)
0218      N2=N1+1
0219      IF (NOREC .LE. N2) GOTO 250
0221      NN=NN+1
0222      READ(8'NN,ERR=500)(ICODE(I),I=86,125)
0223      NN=NN+1
0224      250 ICODE(45)=ICRLF
0225      ICODE(85)=ICRLF
0226      ICODE(125)=ICRLF
0227      ICODE(126)=IDC1
0228      IF (J .EQ. NWRTS) ICODE(126)=LHOME
0230      CALL WRLOOP(I1ST)
0231      I1ST=1
0232      310 CONTINUE
0233      710 IF (ICFLG .EQ. 0) GOTO 330
0235      DO 510 J=1,128
0236      510 ICODE(J)=ICON(J)
0237      CALL WRLOOP(I1ST)
0238      ICFLG=0
0239      330 IF (NOREC .EQ. 0) GOTO 25
C
0241      DO 70 J=1,NOREC
0242      NOR=NRCNO+J-1
0243      READ(8'NOR)(MOUT(I,J),I=1,10)
0244      70 WRITE(1,15)(MOUT(I,J),I=1,10)
0245      DO 80 J=1,11
0246      DO 80 I=1,10
0247      80 MOUT(I,J)=0
0248      GOTO 25
0249      500 CONTINUE
0250      END

```

FORTTRAN IV V01R-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 18:51:18
P00001.OBJ=P00001.FOR/NOSN/DE/LI:1

PAGE 001

```

C
C   PRELIM USER LANG PROG
C   WRITTEN IN FORTTRAN IV
C
0001   REAL*8 NDI,LID,TAB,NWD,MOUT
      1,SK8,CRTINMSG,ND
0002   REAL*8 Q2Q,Q4Q,KEYTYPEFM,OFIL,DI
0003   INTEGER ST
0004   INTEGER DEVST(6)
0005   DIMENSION OFIL(10,3),DI(100)
0006   DIMENSION MSGNO(3)
0007   COMMON /M1710/DEVST
0008   COMMON /LOOP/ ICODE(128),MSK,LSK
0009   DATA MSK,LSK/'77400','177/
0010   COMMON      NDI,LID,TAB,NWD,ST(3,9),IND
0011   COMMON /U000/ MOUT(10,11),NRCNO,NOREC
0012   COMMON /DSK/I1,I2,I3,I4,I5,I6,I16
0013   COMMON /MD04/NOCHARKEY(10),KEYTYPEFM(10)
0014   COMMON /F01/IFAC(3)
0015   COMMON /CPAC/ICON(128),ICFLG
0016   COMMON /S230/ ND
0017   DATA NDI,LID,TAB,NWD,SK8/'NDI      ',
      1'LID      ','TAB      ','NNWD      ','      '/
0018   DATA ISPC /'120240/
0019   DATA ICRLF /'106412/
0020   DATA LF,LHOME,LEDF,LFF4,LFF5,IDC1/'12','24','177777',
      1'14','5000','10400/
0021   DATA IC4H,IHC4,IC6H,IHC8/'2001','404','4001','410/
0022   DATA ILO/'51504/
0023   DATA IC9H,IHC9/'4401','411/
0024   DATA IH15/'405/
0025   DATA IH51/'2401/
0026   15  FORMAT(1X,10A8)
      C   START DIALOGUE
      C   IND IS THE PAIR INDEX NUMBER
      C   ST IS THE STATUS ROUTINE # ARRAY
0027   DO 20 IND=1,3
0028   MSGNO(IND)=0
0029   20  ST(IND,1)=0001
      C   THROW AWAY PACKET TO HAVE B7* INP BUF INIT EMPTY
0030   DO 130 J=1,129
0031   DO 140 I=1,70
0032   140 CONTINUE
0033   130 K=DEVST(1)
0034   I1ST=0
0035   ICFLG=0
0036   DO 22 J=1,11
0037   DO 22 I=1,10
0038   22  MOUT(I,J)=SK8
0039   CALL ASSIGN(1,'TTO:')
0040   CALL ASSIGN (2,'EFDIR')
0041   DEFINE FILE 2(10,100,U,I1)
0042   CALL ASSIGN(3,'EFLOCF')
0043   DEFINE FILE 3(100,120,U,I2)
0044   CALL ASSIGN(4,'EFCKTD')
0045   DEFINE FILE 4(100,120,U,I3)

```

FORTRAN IV VO1R-02 FRI 18-MAR-77 18:51:18 PAGE 002
 CORE=08K, UIC=[20,20] P00001.OBJ=P00001.FOR/NOSN/DE/LI:1

```

0046     CALL ASSIGN(5,'EFTRKD')
0047     DEFINE FILE 5(100,120,U,I4)
0048     CALL ASSIGN(6,'EFTERD')
0049     DEFINE FILE 6(100,120,U,I5)
0050     CALL ASSIGN(7,'INFO.DAT')
0051     DEFINE FILE 7(396,40,U,I6)
0052     CALL ASSIGN(8,'MSG.DAT')
0053     DEFINE FILE 8(211,40,U,I16)
0054     DEVST(5)=0
0055     DO 23 LU=3,6
0056 23  READ(LU'1,ERR=17)((OFIL(I,J),I=1,10),J=1,3)
0057 25  CALL RDLOOP
      C   CALC IND, FORM LIDS
0058  D   WRITE(1,16)(ICODE(J),J=1,4)
0059  D16  FORMAT(1X,'HEADER= ',408)
0060     IF (ICODE(4) .EQ. '41101 .AND. (ICODE(5) .EQ. '151317 .OR.
      1  ICODE(5) .EQ. '51117)) GOTO 500
0062     IND=4
0063     IF (ICODE(3) .EQ. IH51) GOTO 700
0065     IF (ICODE(3) .EQ. IC4H) IND=1
0067     IF (ICODE(3) .EQ. IC8H) IND=2
0069     IF (ICODE(3) .EQ. IC9H) IND=3
0071     IF (IND .EQ. 4) GOTO 25
0073     IF (ICODE(1) .EQ. MSGNO(IND)) GOTO 25
0075     MSGNO(IND)=ICODE(1)
0076     ICODE(3)=IHC4
0077     IF (IND .EQ. 2) ICODE(3)=IHC8
0079     IF (IND .EQ. 3) ICODE(3)=IHC9
0081     IF (ICODE(4) .NE. ILO) GOTO 630
0083     NRCNO=21
0084     NOREC=1
0085     ST(IND,1)=9999
0086     GOTO 12
0087 700  CALL HST
0088     IF (NOREC .EQ. 0) GOTO 25
0090     GOTO 12
0091 630  IF (ST(IND,1) .EQ. 0001) GOTO 51
0093     IF (ST(IND,1) .EQ. 0002) GOTO 52
0095     IF (ST(IND,1) .EQ. 0003) GOTO 53
0097     IF (ST(IND,1) .EQ. 0004) GOTO 54
0099     IF (ST(IND,1) .GE. 1000) GOTO 1
0101     GOTO 2
0102 1   IF (ST(IND,1) .LE. 1999) GOTO 61
0104 2   IF (ST(IND,1) .GE. 2000) GOTO 3
0106     GOTO 4
0107 3   IF (ST(IND,1) .LE. 2999) GOTO 62
0109 4   IF (ST(IND,1) .GE. 3000) GOTO 5
0111     GOTO 6
0112 5   IF (ST(IND,1) .LE. 3999) GOTO 63
0114 6   IF (ST(IND,1) .GE. 4000) GOTO 7
0116     GOTO 8
0117 7   IF (ST(IND,1) .LE. 4999) GOTO 64
0119 8   IF (ST(IND,1) .GE. 5000) GOTO 9
0121     GOTO 12
0122 9   IF (ST(IND,1) .LE. 5999) GOTO 65
0124     51 NRCNO=1

```

FORTRAN IV V01B-02
CORE=08K, UIC: E20,203FRI 18 MAR 77 18:51:10 PAGE 003
P00001.OBJ-P00001.FOR/NOSN/DE/LI:1

```
0125      NOREC=2
0126      ST(IND,1)=0002
0127      GOTO 12
0128      52 NRCNO=3
0129      NOREC=1
0130      ST(IND,1)=0003
0131      GOTO 12
0132      53 NRCNO=4
0133      NOREC=7
0134      ST(IND,1)=0004
0135      GOTO 12
0136      54 I=IAND(LSK, ICODE(4))
0137      I=I-48
0138      IF (I .EQ. 1) GOTO 101
0140      IF (I .EQ. 2) GOTO 102
0142      IF (I .EQ. 3) GOTO 103
0144      IF (I .EQ. 4) GOTO 104
0146      IF (I .EQ. 5) GOTO 105
0148      17 NRCNO=11
0149      NOREC=1
0150      GOTO 12
0151      101 NRCNO=12
0152      NOREC=2
0153      ST(IND,1)=1000
0154      GOTO 12
0155      102 NRCNO=21
0156      NOREC=1
0157      ST(IND,1)=9999
0158      GOTO 12
0159      103 NRCNO=21
0160      NOREC=1
0161      ST(IND,1)=9999
0162      GOTO 12
0163      104 NRCNO=127
0164      NOREC=6
0165      ST(IND,1)=4000
0166      GOTO 12
0167      105 NRCNO=108
0168      NOREC=4
0169      ST(IND,1)=5000
0170      GOTO 12
0171      61 CALL F1000
0172      GOTO 12
0173      62 CALL F2000
0174      GOTO 12
0175      63 CALL F3000
0176      GOTO 12
0177      64 CALL F4000
0178      GOTO 12
0179      65 CALL F5000
0180      12 IF (NOREC .EQ. 0) GOTO 710
C      WRITE TO LOOP
0182      IF (NRCNO .GE. 107) NRCNO=NRCNO+1
0184      DO 200 J=4,128
0185      200 ICODE(J)=0
0186      ICODE(127)=LEOP
```

FORTRAN IV V01B-02 FRI 18-MAR-77 18:51:18 PAGE 004
 CORE=08K, UIC=C20,20J P00001.OBJ=P00001.FOR/NOSN/DE/LI:1

```

0187      IF (NRCNO .NE. 11) GOTO 210
0189      ICODE(4)=LHOME
0190      DO 220 J=5,16
0191      220 ICODE(J)=LF
0192      READ(8'11)(ICODE(I),I=17,56)
0193      ICODE(57)=LHOME
0194      ICODE(58)=LEOP
0195      CALL WRLOOP(I1ST)
0196      GOTO 330
0197      210 NWRTS=4
0198      IF (NOREC .LE. 3) NWRTS=1
0200      IF (NOREC .GE. 4 .AND. NOREC .LE. 6) NWRTS=2
0202      IF (NOREC .GE. 7 .AND. NOREC .LE. 9) NWRTS=3
0204      NN=NRCNO
0205      DO 310 J=1,NWRTS
0206      JJJ=J
0207      DO 450 K=4,126
0208      450 ICODE(K)=0
0209      IF (J .EQ. 1) ICODE(4)=LFF4
0211      IF (J .EQ. 1) ICODE(5)=LFF5
0213      READ(8'NN,ERR=500)(ICODE(I),I=6,45)
0214      N1=(J-1)*3+1
0215      IF (NOREC .LE. N1) GOTO 250
0217      NN=NN+1
0218      READ(8'NN,ERR=500)(ICODE(I),I=46,85)
0219      N2=N1+1
0220      IF (NOREC .LE. N2) GOTO 250
0222      NN=NN+1
0223      READ(8'NN,ERR=500)(ICODE(I),I=86,125)
0224      NN=NN+1
0225      250 ICODE(45)=ICRLF
0226      ICODE(85)=ICRLF
0227      ICODE(125)=ICRLF
0228      ICODE(126)=IDC1
0229      IF (J .EQ. NWRTS) ICODE(126)=LHOME
0231      CALL WRLOOP(I1ST)
0232      I1ST=1
0233      310 CONTINUE
0234      710 IF (ICFLG .EQ. 0) GOTO 330
0236      DO 510 J=1,128
0237      510 ICODE(J)=ICON(J)
0238      CALL WRLOOP(I1ST)
0239      ICFLG=0
0240      330 IF (NOREC .EQ. 0) GOTO 25
C
0242      WRITE OUT SCREEN
0243      DO 70 J=1,NOREC
0244      NOR=NRCNO+J-1
0245      READ(8'NOR)(MOUT(I,J),I=1,10)
0245      70 WRITE(1,15)(MOUT(I,J),I=1,10)
0246      DO 80 J=1,11
0247      DO 80 I=1,10
0248      80 MOUT(I,J)=0
0249      GOTO 25
0250      500 CONTINUE
0251      END

```

P1000

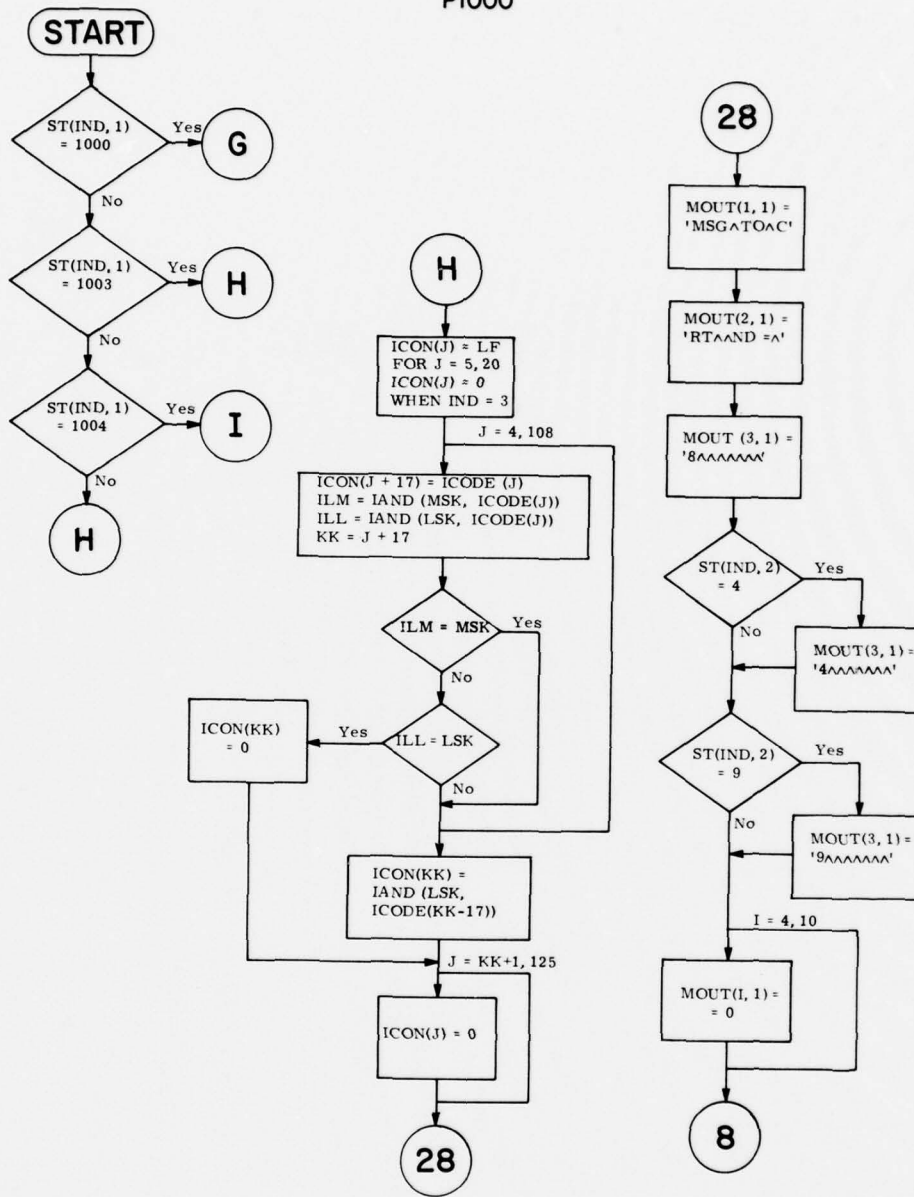


Figure 1-10. P1000

P1000 (cont.)

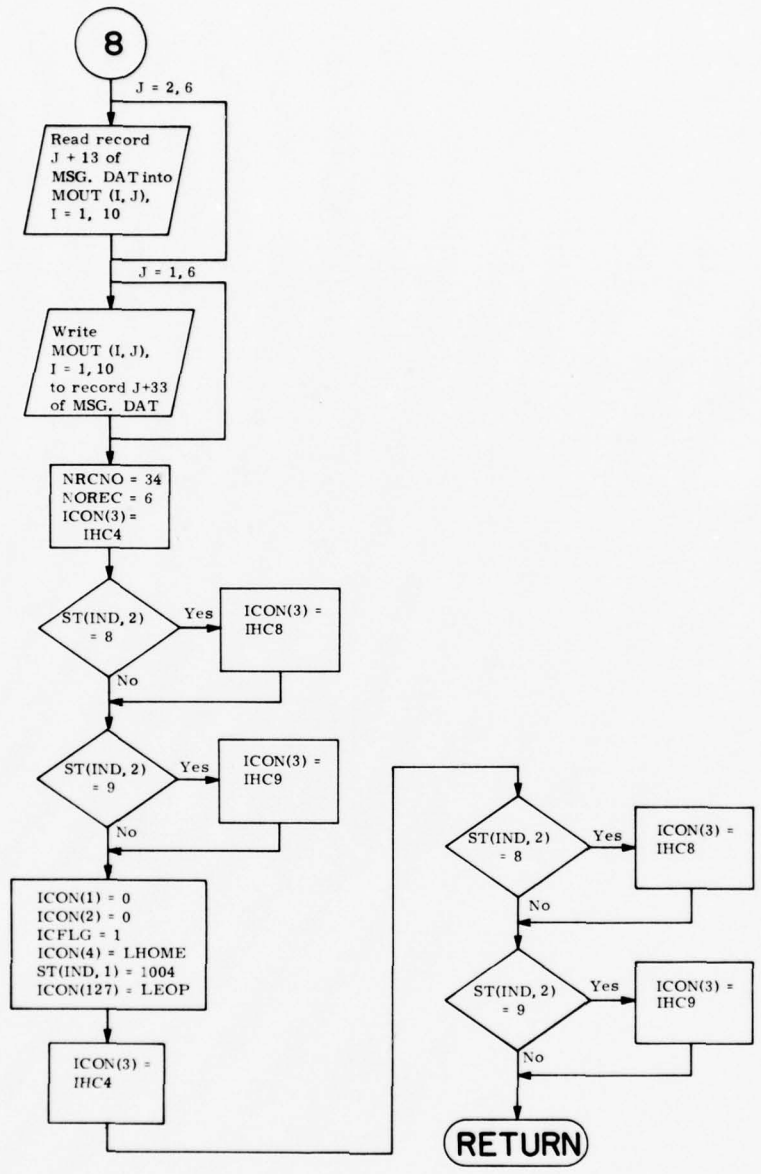


Figure 1-10. (Cont.)

P1000 (cont.)

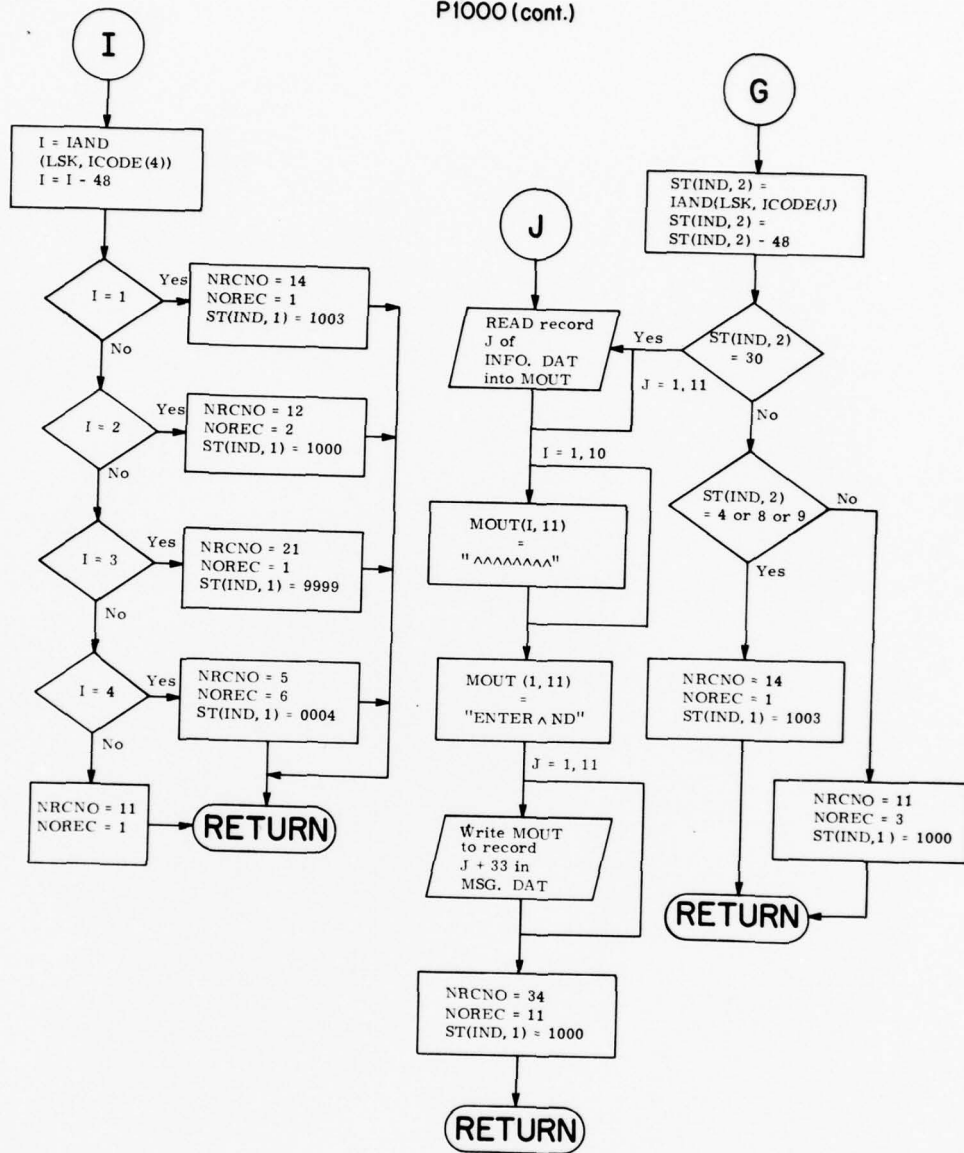


Figure 1-10. (Cont.)

FORTRAN IV V01B-02
CORE=0BK, UIC=[20,20]

TUE 01-MAR-77 15:52:27

PAGE 001

P1000.OBJ=P1000.FOR/NOSN/LI:1

```

C
0001 SUBROUTINE P1000
C
C CRT TO CRT MODE OF OPERATION
0002 REAL*8 MOUT,SK8,H8E,NDI,LID,TAB,NWD
0003 INTEGER ST
0004 COMMON NDI,LID,TAB,NWD,ST(3,9),IND
0005 COMMON /U000/ MOUT(10,11),NRCNO,NOREC
0006 COMMON /LODP/ ICODE(128),MSK,LSK
0007 COMMON /CPAC/ICON(128),ICFLG
0008 DATA IHC4,IHC8/'2404','2410/
0009 REAL*8 H34,H38 ,H39
0010 REAL*8 LIDPAIR,M1,M2
0011 DATA H38,H34,SK8,H8E/'8 ','4 ',' ' ','ENTER ND'/
0012 DATA M1,M2/'MSG TO C','RT ND= '/
0013 DATA ISFC,LF,LHOME,LEOP/'20040','12','24','177777/
0014 DATA IHC9,H39/'2411','9 '/
0015 IF (ST(IND,1) .EQ.1000) GOTO 14
0017 IF (ST(IND,1) .EQ.1003)GOTO 53
0019 IF (ST(IND,1) .EQ.1004)GOTO 54
0021 53 DO 500 J=5,20
0022 ICON(J)=LF
0023 500 IF (IND .EQ. 3) ICON(J)=0
0025 DO 510 J=4,108
0026 ICON(J+17)=ICODE(J)
0027 ILM=IAND(MSK,ICODE(J))
0028 ILL=IAND(LSK,ICODE(J))
0029 KK=J+17
0030 IF (ILM .EQ. MSK) GOTO 520
0032 510 IF (ILL .EQ. LSK) GOTO 530
0034 520 ICON(KK)=IAND(LSK,ICODE(KK-17))
0035 GOTO 540
0036 530 ICON(KK)=0
0037 540 DO 550 J=KK+1,125
0038 550 ICON(J)=0
0039 MOUT(1,1)=M1
0040 MOUT(2,1)=M2
0041 MOUT(3,1)=H38
0042 IF (ST(IND,2) .EQ. 4) MOUT(3,1)=H34
0044 IF (ST(IND,2) .EQ. 9) MOUT(3,1)=H39
0046 DO 20 I=4,10
0047 20 MOUT(I,1)=0
0048 DO 300 J=2,6
0049 300 READ(8'J+13)(MOUT(I,J),I=1,10)
0050 DO 400 J=1,6
0051 400 WRITE(8'J+33)(MOUT(I,J),I=1,10)
0052 NRCNO=34
0053 NOREC=6
0054 ICON(3)=IHC4
0055 IF (ST(IND,2) .EQ. 8) ICON(3)=IHC8
0057 IF (ST(IND,2) .EQ. 9) ICON(3)=IHC9
0059 ICON(1)=0
0060 ICON(2)=0
0061 ICON(4)=LHOME
0062 IF (IND .EQ. 3) ICON(4)=LF
0064 ICON(126)=LHOME

```

FORTRAN IV V01B-02
CORE=08K, UIC=C20,20J

TUE 01-MAR-77 15:52:27

PAGE 002

P1000.OBJ=P1000.FOR/NOSN/LI:1

```
0065     ICON(127)=LEOP
0066     ICFLG=1
0067     ST(IND,1)=1004
0068     RETURN
0069     54 I=IAND(LSK,ICODE(4))
0070     I=I-48
0071     IF (I .EQ. 1) GOTO 700
0073     IF (I .EQ. 2) GOTO 1
0075     IF (I .EQ. 3) GOTO 8
0077     IF (I .EQ. 4) GOTO 6
0079     19 NRCNO=11
0080     NOREC=1
0081     RETURN
0082     700 NRCNO=14
0083     NOREC=1
0084     ST(IND,1)=1003
0085     RETURN
0086     1 NRCNO=12
0087     NOREC=2
0088     ST(IND,1)=1000
0089     RETURN
0090     8 NRCNO=21
0091     NOREC=1
0092     ST(IND,1)=9999
0093     RETURN
0094     6 NRCNO=5
0095     NOREC=6
0096     ST(IND,1)=0004
0097     RETURN
C
0098     14 ST(IND,2)=IAND(LSK,ICODE(4))
0099     ST(IND,2)=ST(IND,2)-48
0100     IF (ST(IND,2) .EQ. 30) GOTO 30
C
C     CK FOR INVALID NODE DESIGNATOR
0102     IF (ST(IND,2) .EQ.4) GOTO 7
0104     IF (ST(IND,2) .EQ. 8) GOTO 7
0106     IF (ST(IND,2) .EQ. 9) GOTO 7
C
C     ELSE INVALID NODE DESIGNATOR
0108     17 NRCNO=11
0109     NOREC=3
0110     ST(IND,1)=1000
0111     RETURN
C
C     REQ ENTER MESS - MS121
0112     7 NRCNO=14
0113     NOREC=1
0114     ST(IND,1)=1003
0115     RETURN
0116     30 CONTINUE
0117     DO 100 J=1,11
0118     100 READ(7'J)(MOUT(I,J),I=1,10)
0119     DO 40 I=1,10
0120     40 MOUT(I,11)=SKB
0121     MOUT(1,11)=HBE
0122     DO 50 J=1,11
0123     50 WRITE(8'J+33)(MOUT(I,J),I=1,10)
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

TUE 01-MAR-77 15:52:27

PAGE 003

P1000.OBJ=P1000.FOR/NOSN/LI:1

0124 NRCND=34
0125 NOREC=11
0126 ST(IND,1)=1000
0127 RETURN
0128 END

FORTRAN IV DIAGNOSTICS

[WARNING] MSG #092 VARIABLE "LIDPAI" NAME EXCEEDS 6 CHARACTERS
[WARNING] MSG #094 NON-STANDARD STATEMENT ORDERING

FOR -- [P1000] ERRORS: 0, WARNINGS: 2

>

FOR P5000.OBJ=P5000.FOR/NOSN/LI:1

FORTRAN IV V018-02
 CORE=08K, UIC=(20,20)

FRI 18-MAR-77 19:01:25

PAGE 002

P10001 OBJ=P10001 FOR/N05N/LI:1

```

0065     ICON(127)=LEOP
0066     ICFLG=1
0067     ST(IND,1)=1004
0068     RETURN
0069     54 I=IAND(LSK,ICODE(4))
0070     I=I-48
0071     IF (I.EQ. 1) GOTO 700
0073     IF (I.EQ. 2) GOTO 1
0075     IF (I.EQ. 3) GOTO 8
0077     IF (I.EQ. 4) GOTO 6
0079     19 NRCNO=11
0080     NOREC=1
0081     RETURN
0082     700 NRCNO=14
0083     NOREC=1
0084     ST(IND,1)=1003
0085     RETURN
0086     1 NRCNO=12
0087     NOREC=2
0088     ST(IND,1)=1000
0089     RETURN
0090     8 NRCNO=21
0091     NOREC=1
0092     ST(IND,1)=9999
0093     RETURN
0094     6 NRCNO=5
0095     NOREC=6
0096     ST(IND,1)=0004
0097     RETURN
C      NODE DESIG OF DEST CRT
0098     14 ST(IND,2)=IAND(LSK,ICODE(4))
0099     ST(IND,2)=ST(IND,2)-48
0100     IF (ST(IND,2).EQ. 30) GOTO 30
C      ELSE DO
C      CK FOR INVALID NODE DESIGNATOR
0102     IF (ST(IND,2).EQ. 4) GOTO 7
0104     IF (ST(IND,2).EQ. 8) GOTO 7
0106     IF (ST(IND,2).EQ. 9) GOTO 7
C      ELSE INVALID NODE DESIGNATOR
0108     17 NRCNO=11
0109     NOREC=3
0110     ST(IND,1)=1000
0111     RETURN
C      REQ ENTER MESS - MS121
0112     7 NRCNO=14
0113     NOREC=1
0114     ST(IND,1)=1003
0115     RETURN
0116     30 CONTINUE
0117     DO 100 J=1,11
0118     100 READ(7,J)(MOUT(I,J),I=1,10)
0119     DO 40 I=1,10
0120     40 MOUT(I,11)=SK8
0121     MOUT(1,11)=HSE
0122     DO 50 J=1,11
0123     50 WRITE(8,J+22)(MOUT(I,J),I=1,10)

```

FORTRAN IV V01B-02
CORE=08K, UIC=(20,20)

FRI 18-MAR-77 19:01:25

PAGE 003

P10001.OBJ=P10001.FOR/NDSEN/LI.1

0124 NRCNO=34
0125 NOREC=11
0126 ST(IND,1)=1000
0127 RETURN
0128 END

FORTRAN IV DIAGNOSTICS

[WARNING] MSG #092 VARIABLE "LIDPAI" NAME EXCEEDS 6 CHARACTERS
[WARNING] MSG #094 NON-STANDARD STATEMENT ORDERING

FOR -- [P1000] ERRORS: 0, WARNINGS: 2

>

P2000

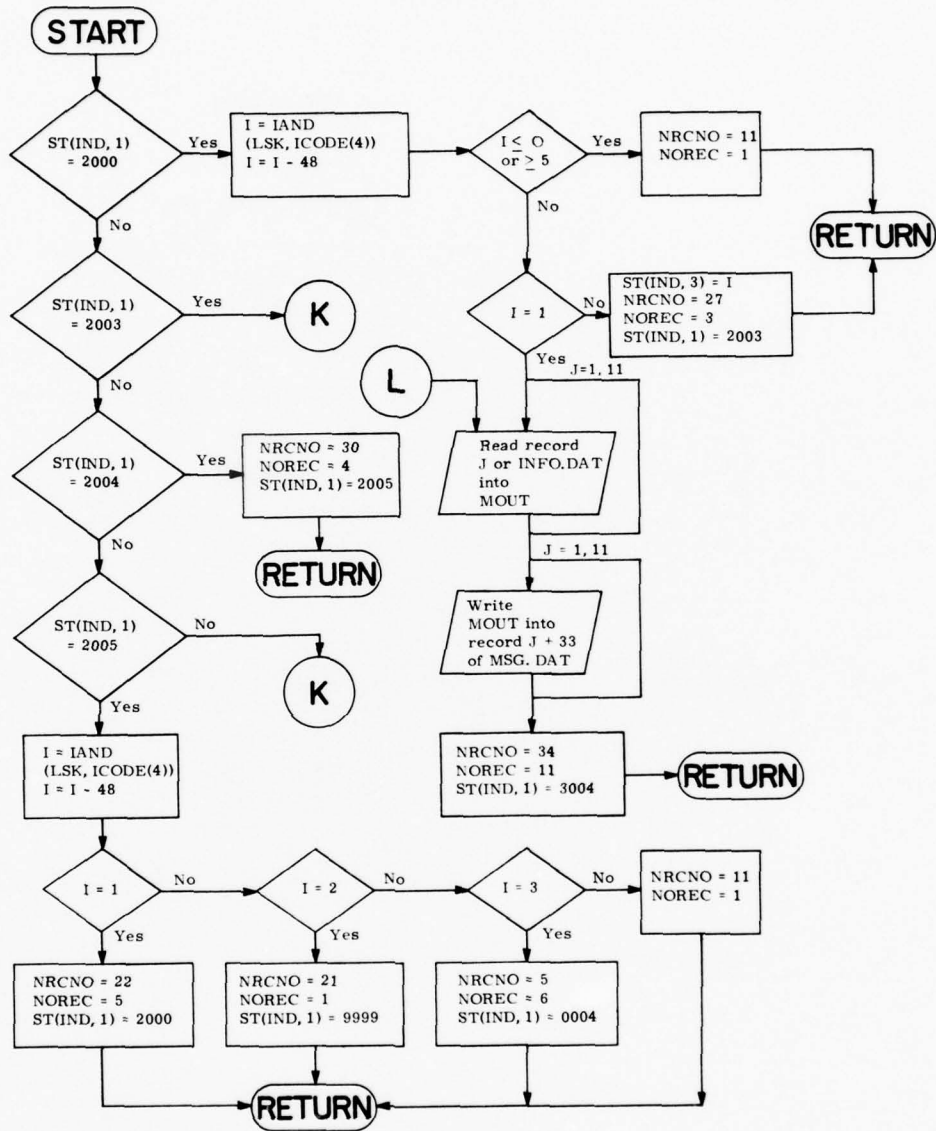


Figure 1-11. P2000

P2000 (cont.)

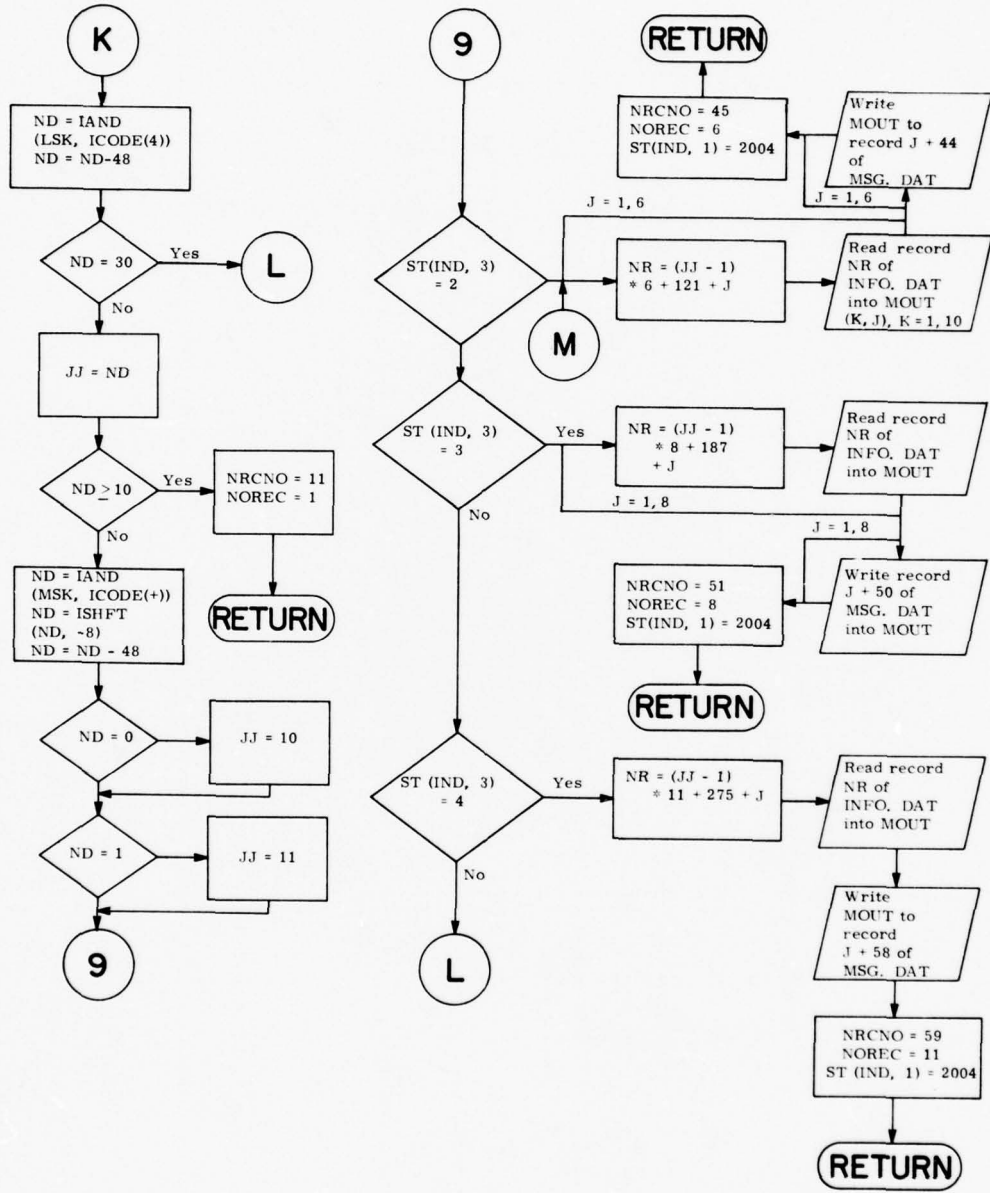


Figure 1-11. (Cont.)

FORTTRAN IV V01B-02
CORE=08K, UIC=C20,20J

TUE 01-MAR-77 16:02:02

PAGE 001

P2000.OBJ=P2000.FOR/NOSN/LI:1

C SYSTEM INQUIRY MODE OF OPERATION

```
C
0001 SUBROUTINE P2000
0002 REAL*8 NDI,LID,TAB,NWD
0003 REAL*8 MOUT,CRTINMSG
0004 INTEGER ST
0005 COMMON NDI,LID,TAB,NWD,ST(3,9),IND
0006 COMMON /DSK/I1,I2,I3,I4,I5,I6,I16
0007 COMMON /U000/ MOUT(10,11),NRCNO,NOREC
0008 COMMON /LOOP/ ICODE(128),MSK,LSK
0009 IF (ST(IND,1) .EQ. 2000) GOTO 39
0011 IF (ST(IND,1) .EQ. 2003) GOTO 53
0013 IF (ST(IND,1) .EQ. 2004) GOTO 54
0015 IF (ST(IND,1) .EQ. 2005) GOTO 55
0017 53 ND=IAND(LSK,ICODE(4))
0018 ND=ND-48
0019 IF (ND .EQ. 30) GOTO 72
0021 JJ=ND
0022 IF (ND .GE. 10) GOTO 20
0024 ND=IAND(MSK,ICODE(4))
0025 ND=ISHFT(ND,-8)
0026 ND=ND-48
0027 IF (ND .EQ. 0) JJ=10
0029 IF (ND .EQ. 1) JJ=11
0031 GOTO 22
0032 20 NRCNO=11
0033 NOREC=1
0034 RETURN
0035 22 CONTINUE
0036 GOTO (72,51,52,530) ST(IND,3)
0037 51 DO 110 J=1,6
0038 NR=(JJ-1)*6+121+J
0039 110 READ(7'NR)(MOUT(K,J),K=1,10)
0040 DO 44 J=1,6
0041 44 WRITE(8'J+44,ERR=20)(MOUT(K,J),K=1,10)
0042 NRCNO=45
0043 NOREC=6
0044 ST(IND,1)=2004
0045 RETURN
0046 52 DO 120 J=1,8
0047 NR=(JJ-1)*8+187+J
0048 120 READ(7'NR)(MOUT(K,J),K=1,10)
0049 DO 45 J=1,8
0050 45 WRITE(8'J+50,ERR=20)(MOUT(K,J),K=1,10)
0051 NRCNO=51
0052 NOREC=8
0053 ST(IND,1)=2004
0054 RETURN
0055 530 DO 130 J=1,11
0056 NR=(JJ-1)*11+275+J
0057 READ(7'NR)(MOUT(K,J),K=1,10)
0058 130 WRITE(8'J+58)(MOUT(K,J),K=1,10)
0059 NRCNO=59
0060 NOREC=11
0061 ST(IND,1)=2004
0062 RETURN
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]TUE 01-MAR-77 16:02:02 PAGE 002
P2000.OBJ=P2000.FOR/NOSN/LI:1

```
0063 54 NRCNO=30
0064 NOREC=4
0065 ST(IND,1)=2005
0066 RETURN
0067 55 I=IAND(LSK,ICODE(4))
0068 I=I-48
0069 IF (I .EQ. 1) GOTO 2
0071 IF (I .EQ. 2) GOTO 8
0073 IF (I .EQ. 3) GOTO 6
0075 NRCNO=11
0076 NOREC=1
0077 RETURN
0078 2 NRCNO=22
0079 NOREC=5
0080 ST(IND,1)=2000
0081 RETURN
0082 8 NRCNO=21
0083 NOREC=1
0084 ST(IND,1)=9999
0085 RETURN
0086 6 NRCNO=5
0087 NOREC=6
0088 ST(IND,1)=0004
0089 RETURN
C GET INFOTYPE
0090 39 I=IAND(LSK,ICODE(4))
0091 I=I-48
0092 IF (I.LE.0) GOTO 71
0094 IF (I .GE. 5) GOTO 71
0096 IF (I.EQ. 1) GOTO 72
0098 ST(IND,3)=I
0099 25 NRCNO=27
0100 NOREC=3
0101 ST(IND,1)=2003
0102 RETURN
0103 72 DO 100 J=1,11
0104 100 READ(7'J)(MOUT(K,J),K=1,10)
0105 DO 43 J=1,11
0106 43 WRITE(8'J+33)(MOUT(K,J),K=1,10)
0107 NRCNO=34
0108 NOREC=11
0109 ST(IND,1)=2004
0110 RETURN
0111 71 NRCNO=11
0112 NOREC=1
0113 RETURN
0114 END
```

FORTRAN IV DIAGNOSTICS

[WARNING] MSG #092 VARIABLE "CRTINM" NAME EXCEEDS 6 CHARACTERS

FOR -- [P2000] ERRORS: 0, WARNINGS: 1

P3000

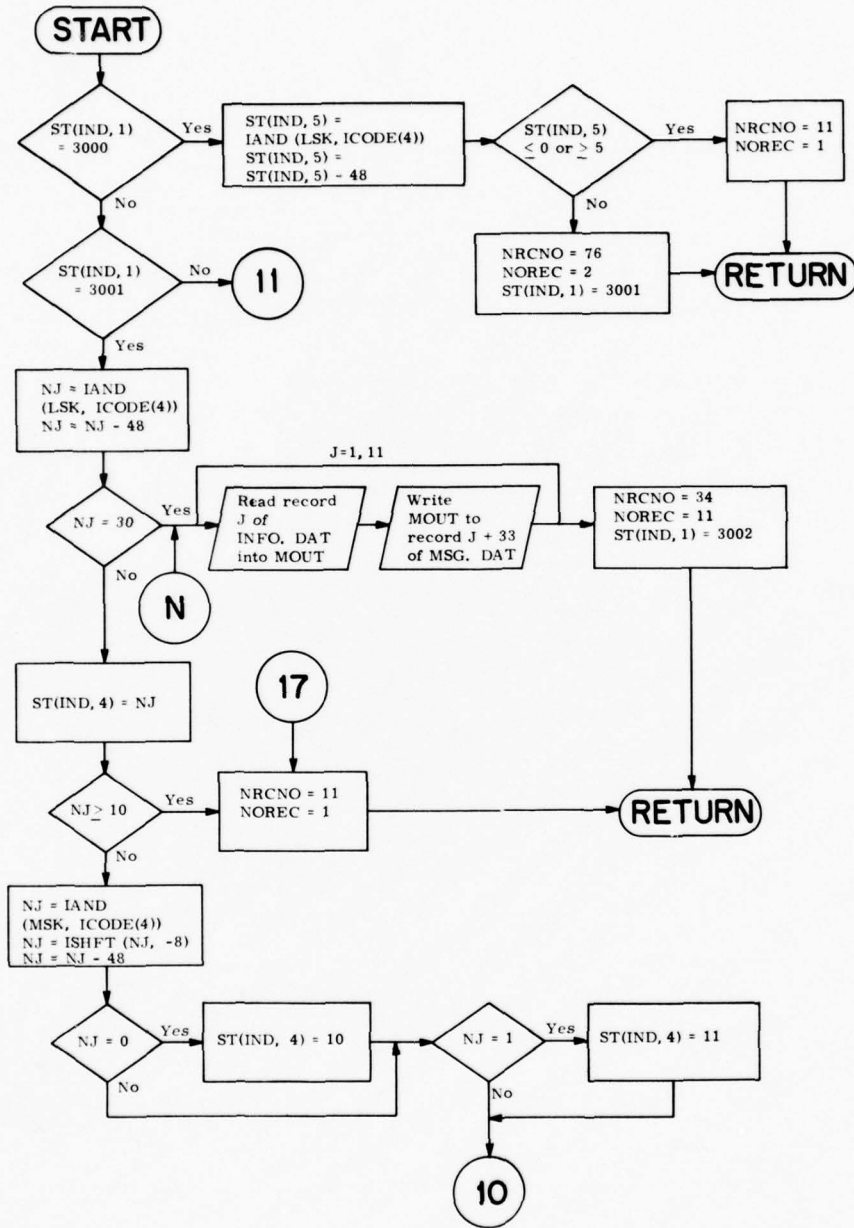


Figure 1-12. P3000

P3000(cont.)

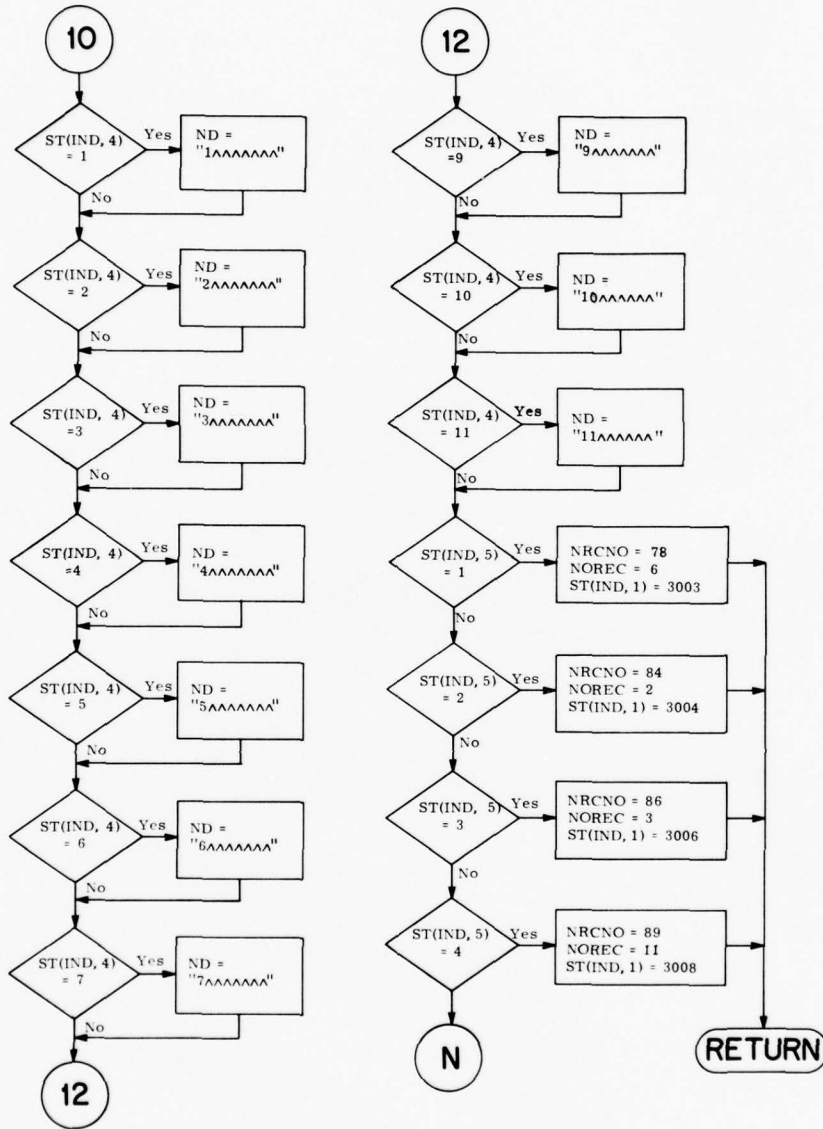


Figure 1-12. (Cont.)

P3000(cont.)

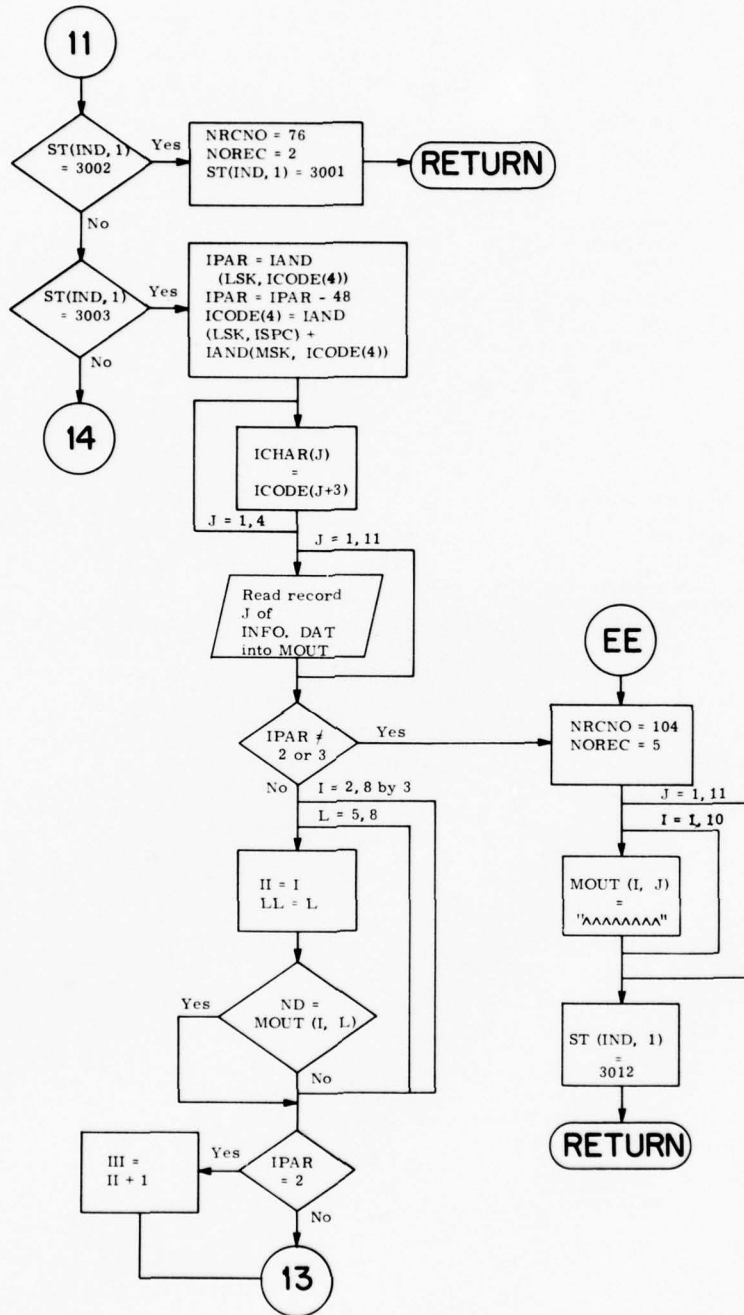


Figure 1-12. (Cont.)

P3000(cont.)

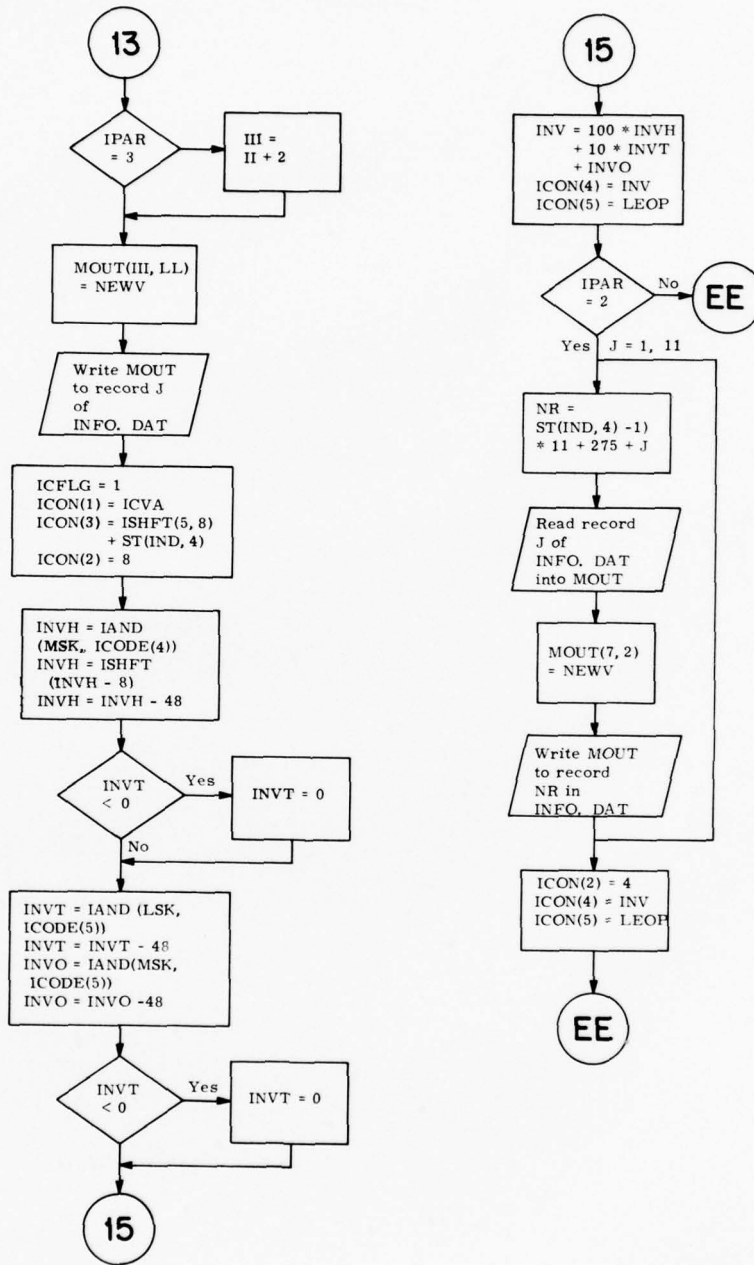


Figure 1-12. (Cont.)

P3000(cont.)

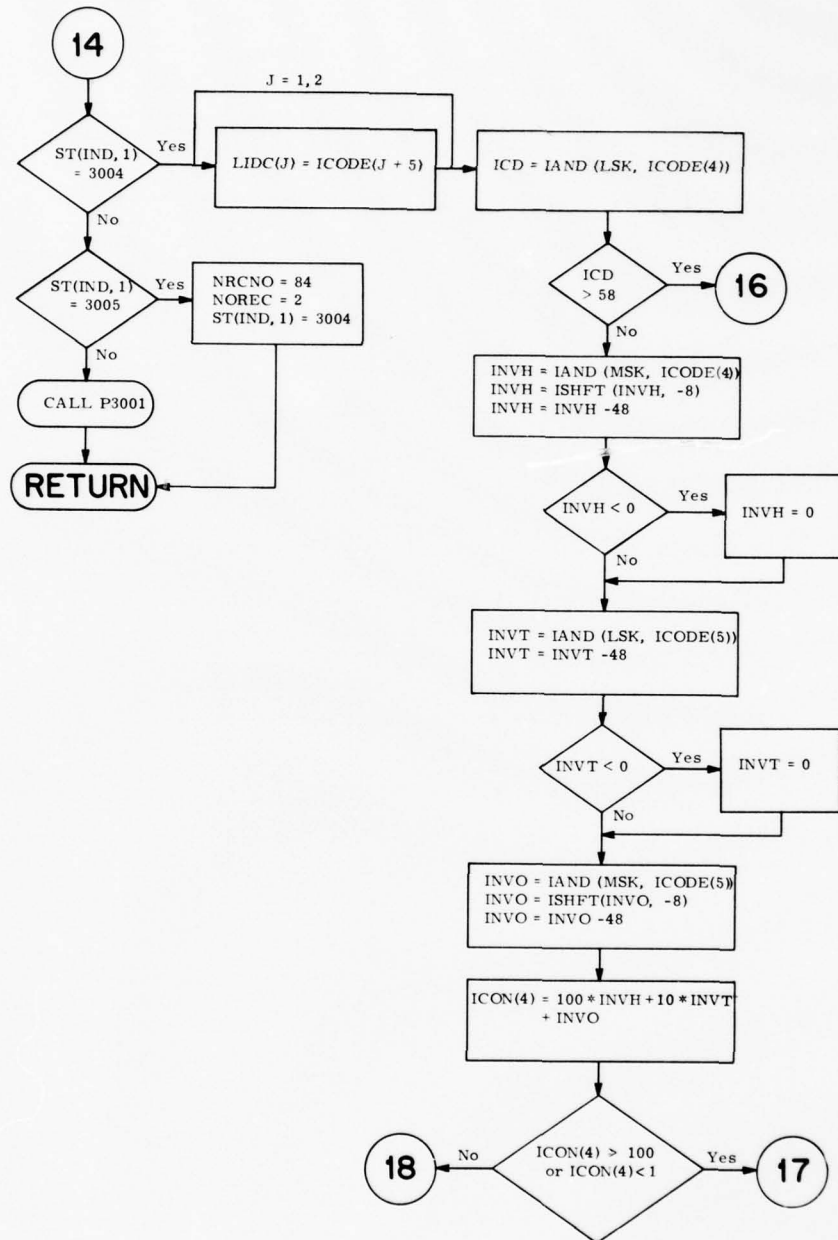


Figure 1-12. (Cont.)

3000(cont.)

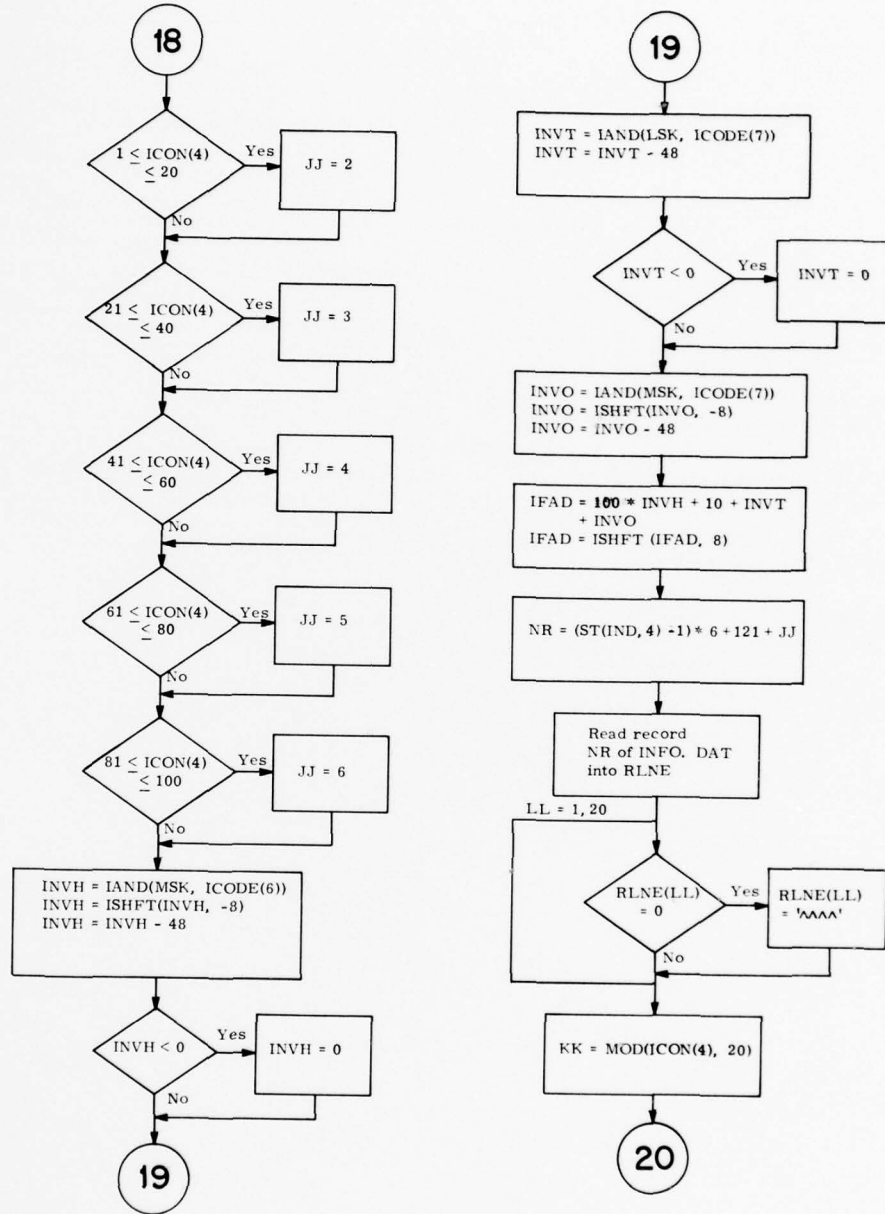


Figure 1-12. (Cont.)

P3000 (cont.)

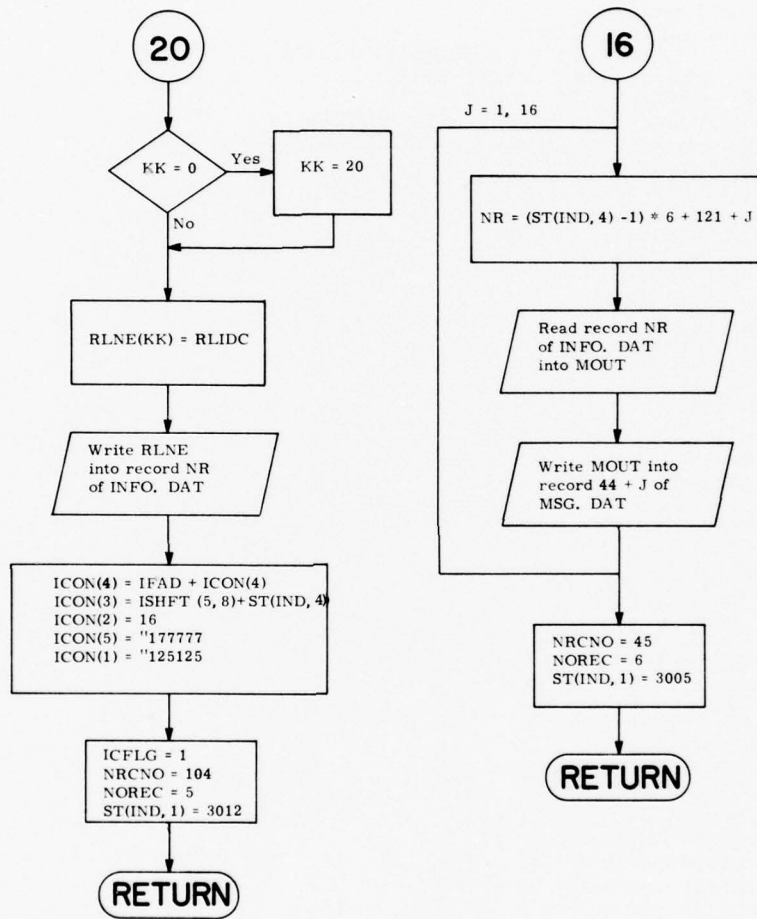


Figure 1-12. (Cont.)

FORTRAN IV V01B-02
CORE=0BK, UIC=[20,20]

FRI 18-MAR-77 19:04:40
F3000.OBJ=F3000.FOR/NOSN/LI:1

PAGE 001

```

C      SYSTEM CONTROL MODE OF OPERATION
C
0001  C      SUBROUTINE F3000
0002      REAL*8      NDI,LID,TAB,NWD
0003      INTEGER ST
0004      REAL*8 MOUT,ND,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,
1      NEWV,CND,SK8,Z
0005      REAL*4 RLIDC,RLNE,RSPC
0006      COMMON      NDI,LID,TAB,NWD,ST(3,9),IND
0007      COMMON /DSK/I1,I2,I3,I4,I5,I6,I16
0008      COMMON /S230/ ND
0009      COMMON /U000/ MOUT(10,11),NRCNO,NOREC
0010      COMMON /LOOP/ICODE(128),MSK,LSK
0011      COMMON /CFAC/ICON(128),ICFLG
0012      DIMENSION ICHAR(4),LIDC(2),RLNE(20)
0013      EQUIVALENCE (ICHAR,NEWV)
0014      EQUIVALENCE (LIDC,RLIDC)
0015      DATA N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,CND,SK8,Z/
1      '1      ','2      ','3      ','4      ','
2      '5      ','6      ','7      ','8      ','9      ','
3      '10     ','11     ','8HCAN'T DO','0      '//
0016      DATA ISPC/'120240/
0017      DATA ICVA,LEOP/'125125','177777/
0018      DATA RSPC // ''//
0019  20      IF (ST(IND,1) .EQ. 3000) GOTO 3
0021      IF (ST(IND,1) .EQ. 3001) GOTO 21
0023      IF (ST(IND,1) .EQ. 3002) GOTO 22
0025      IF (ST(IND,1) .EQ. 3003) GOTO 23
0027      IF (ST(IND,1) .EQ. 3004) GOTO 24
0029      IF (ST(IND,1) .EQ. 3005) GOTO 25
0031      CALL P3001
0032      RETURN
0033  21      NJ=IAND(LSK,ICODE(4))
0034      NJ=NJ-48
0035      IF (NJ .EQ. 30) GOTO 102
0037      ST(IND,4)=NJ
0038      IF (NJ .GE. 10) GOTO 120
0040      NJ=IAND(MSK,ICODE(4))
0041      NJ=ISHFT(NJ,-8)
0042      NJ=NJ-48
0043      IF (NJ .EQ. 0) ST(IND,4)=10
0045      IF (NJ .EQ. 1) ST(IND,4)=11
0047      IF (ST(IND,4) .EQ. 1) ND=N1
0049      IF (ST(IND,4) .EQ. 2) ND=N2
0051      IF (ST(IND,4) .EQ. 3) ND=N3
0053      IF (ST(IND,4) .EQ. 4) ND=N4
0055      IF (ST(IND,4) .EQ. 5) ND=N5
0057      IF (ST(IND,4) .EQ. 6) ND=N6
0059      IF (ST(IND,4) .EQ. 7) ND=N7
0061      IF (ST(IND,4) .EQ. 8) ND=N8
0063      IF (ST(IND,4) .EQ. 9) ND=N9
0065      IF (ST(IND,4) .EQ. 10) ND=N10
0067      IF (ST(IND,4) .EQ. 11) ND=N11
0069      GO TO 122
0070  120      NRCNO=11

```

FORTRAN IV V01B-02
 CORE=08K, UIC=C20,20J

FRI 18-MAR-77 19:04:40
 P3000.OBJ=P3000.FDR/NOSN/LI:1

PAGE 002

```

0071      NOREC=1
0072      RETURN
0073  122  IF (ST(IND,5) .EQ. 1) GO TO 161
0075      IF (ST(IND,5) .EQ. 2) GO TO 165
0077      IF (ST(IND,5) .EQ. 3) GO TO 166
0079      IF (ST(IND,5) .EQ. 4) GO TO 167
0081  102  DO 110 J=1,11
0082      READ(7,J)(MOUT(I,J),I=1,10)
0083  110  WRITE(8,J+33)(MOUT(I,J),I=1,10)
0084      NRCNO=34
0085      NOREC=11
0086      ST(IND,1)=3002
0087      RETURN
0088  161  NRCNO=78
0089      NOREC=6
0090      ST(IND,1)=3003
0091      RETURN
0092  165  NRCNO=84
0093      NOREC=2
0094      ST(IND,1)=3004
0095      RETURN
0096  166  NRCNO=86
0097      NOREC=3
0098      ST(IND,1)=3006
0099      RETURN
0100  167  NRCNO=89
0101      NOREC=11
0102      ST(IND,1)=3008
0103      RETURN
C      NDI RESPONSE
C      WAS SUBROUTINE 3002
0104  22  NRCNO=76
0105      NOREC=2
0106      ST(IND,1)=3001
0107      RETURN
C      NET DEVICE PARAMETERS
C      WAS SUBROUTINE 3003
0108  23  IPAR=IAND(LSK,ICODE(4))
0109      IPAR=IPAR-48
0110      ICODE(4)=IAND(LSK,ISFC)+IAND(MSK,ICODE(4))
0111      DO 600 J=1,4
0112  600  ICHAR(J)=ICODE(J+3)
0113      DO 200 J=1,11
0114  200  READ(7,J)(MOUT(I,J),I=1,10)
0115      IF (IPAR .EQ. 1) GO TO 250
0117      IF (IPAR .EQ. 2) GO TO 230
0119      IF (IPAR .EQ. 3) GO TO 230
0121      GO TO 250
0122  230  DO 260 I=2,8,3
0123      DO 260 L=5,8
0124      II=I
0125      LL=L
0126      IF (ND .EQ. MOUT(I,L)) GO TO 240
0128  260  CONTINUE
0129  240  IF (IPAR .EQ. 2) III=III+1
0131      IF (IPAR .EQ. 3) III=III+2

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:04:40

PAGE 003

P3000.OBJ=P3000.FOR/NOSN/LI:1

```
0133      MOUT(III,LL)=NEWV
0134      DO 241 J=1,11
0135 241   WRITE(7,J)(MOUT(I,J),I=1,10)
0136      ICFLG=1
0137      ICON(1)=ICVA
0138      ICON(3)=ISHFT(5,8)+ST(IND,4)
0139      ICON(2)=8
0140      INVH=IAND(MSK,ICODE(4))
0141      INVH=ISHFT(INVH,-8)
0142      INVH=INVH-48
0143      IF (INVH .LT. 0) INVH=0
0144      INVT=IAND(LSK,ICODE(5))
0145      INVT=INVT-48
0146      IF (INVT .LT. 0) INVT=0
0147      INVO=IAND(MSK,ICODE(5))
0148      INVO=ISHFT(INVO,-8)
0149      INVO=INVO-48
0150      INV=100*INVH+10*INVT+INVO
0151      ICON(4)=INV
0152      ICON(5)=LEOF
0153      IF (IPAR .EQ. 2) GOTO 247
0154      GO TO 250
0155 247   DO 248 J=1,11
0156      NR=(ST(IND,4)-1)*11+275+J
0157      READ(7,NR)(MOUT(I,J),I=1,10)
0158      MOUT(7,2)=NEWV
0159 248   WRITE(7,NR)(MOUT(I,J),I=1,10)
0160      ICON(2)=4
0161      ICON(4)=INV
0162      ICON(5)=LEOF
0163 250   NRCNO=104
0164      NOREC=5
0165      DO 280 J=1,11
0166      DO 280 I=1,10
0167 280   MOUT(I,J)=SK8
0168      ST(IND,1)=3012
0169      RETURN
0170      LIDS
C
0171 24   DO 700 J=1,2
0172 700   LIDC(J)=ICODE(J+5)
0173      ILD=IAND(LSK,ICODE(4))
0174      IF (ILD .GT. 58) GOTO 361
0175      INVH=IAND(MSK,ICODE(4))
0176      INVH=ISHFT(INVH,-8)
0177      INVH=INVH-48
0178      IF (INVH .LT. 0) INVH=0
0179      INVT=IAND(LSK,ICODE(5))
0180      INVT=INVT-48
0181      IF (INVT .LT. 0) INVT=0
0182      INVO=IAND(MSK,ICODE(5))
0183      INVO=ISHFT(INVO,-8)
0184      INVO=INVO-48
0185      ICON(4)=100*INVH+10*INVT+INVO
0186      IF (ICON(4) .GT. 100) GOTO 120
0187      IF (ICON(4) .LT. 1) GOTO 120
0188      JJ=2
```

FORTRAN IV V01R-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:04:40

PAGE 004

P3000.OBJ=P3000.FOR/NOSN/LI:1

```

0196     IF (ICON(4) .GE. 21) JJ=3
0198     IF (ICON(4) .GE. 41) JJ=4
0200     IF (ICON(4) .GE. 61) JJ=5
0202     IF (ICON(4) .GE. 81) JJ=6
0204     INVH=IAND(MSK,ICODE(6))
0205     INVH=ISHFT(INVH,-8)
0206     INVH=INVH-48
0207     IF (INVH .LT. 0) INVH=0
0209     INVT=IAND(LSK,ICODE(7))
0210     INVT=INVT-48
0211     IF (INVT .LT. 0) INVT=0
0213     INVO=IAND(MSK,ICODE(7))
0214     INVO=ISHFT(INVO,-8)
0215     INVO=INVO-48
0216     IFAD=100*INVH+10*INVT+INVO
0217     IFAD=ISHFT(IFAD,8)
0218     NR=(ST(IND,4)-1)*6+121+JJ
0219     READ(7'NR,ERR=120)(RLNE(I),I=1,20)
0220     DO 330 LL=1,20
0221 330 IF (RLNE(LL) .EQ. 0) RLNE(LL)=RSPC
0223     KK=MOD(ICON(4),20)
0224     IF (KK .EQ. 0) KK=20
0226     RLNE(KK)=RLIDC
0227     WRITE(7'NR,ERR=120)(RLNE(I),I=1,20)
0228 320 ICON(4)=IFAD+ICON(4)
0229     ICON(3)=ISHFT(5,8)+ST(IND,4)
0230     ICON(2)=16
0231     ICON(5)=LEOP
0232     ICON(1)=ICVA
0233     ICFLG=1
0234     NRCNO=104
0235     NOREC=5
0236     ST(IND,1)=3012
0237     RETURN
0238 361 DO 362 J=1,6
0239     NR=(ST(IND,4)-1)*6+121+J
0240     READ(7'NR,ERR=120)(MOUT(I,J),I=1,10)
0241 362 WRITE(8'44+J)(MOUT(I,J),I=1,10)
0242     NRCNO=45
0243     NOREC=6
0244     ST(IND,1)=3005
0245     RETURN
C     LID RESPONSE
C     WAS SUBROUTINE 3005
0246 25 NRCNO=84
0247     NOREC=2
0248     ST(IND,1)=3004
0249     RETURN
0250 3 ST(IND,5)=IAND(LSK,ICODE(4))
0251     ST(IND,5)=ST(IND,5)-48
0252     IF (ST(IND,5) .LE. 0) GO TO 120
0254     IF (ST(IND,5) .GE. 5) GO TO 120
0256     ST(IND,1)=3001
0257     NRCNO=76
0258     NOREC=2
0259     RETURN

```

Burroughs Corporation

FORTRAN IV V01B-02
CORE=0BK, UIC=[20,20]

FRI 18-MAR-77 19:04:40 PAGE 005
P3000.OBJ=P3000.FOR/NOSN/LI:1

0260 END

>

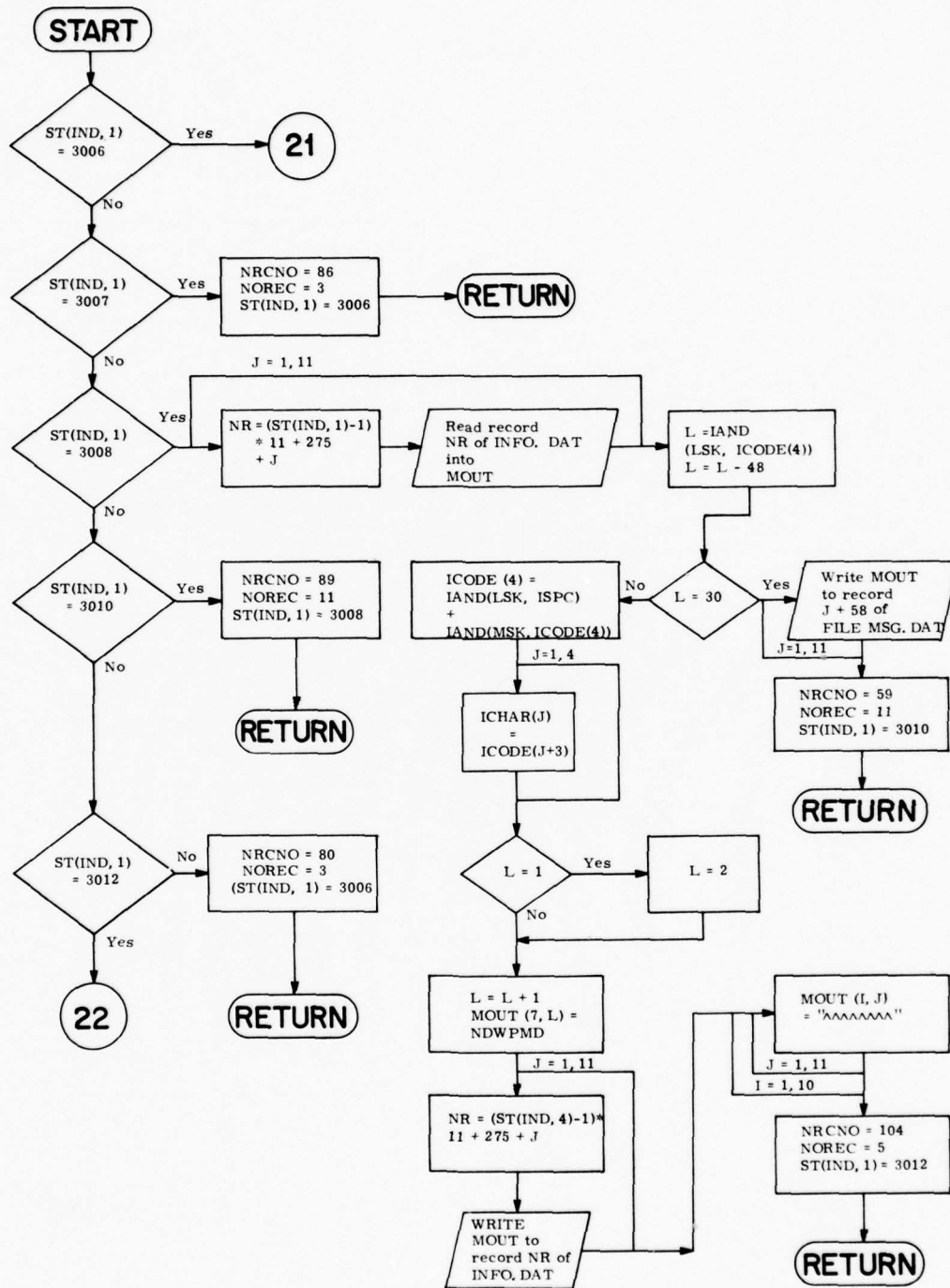


Figure 1-13. P3001

P300I (cont.)

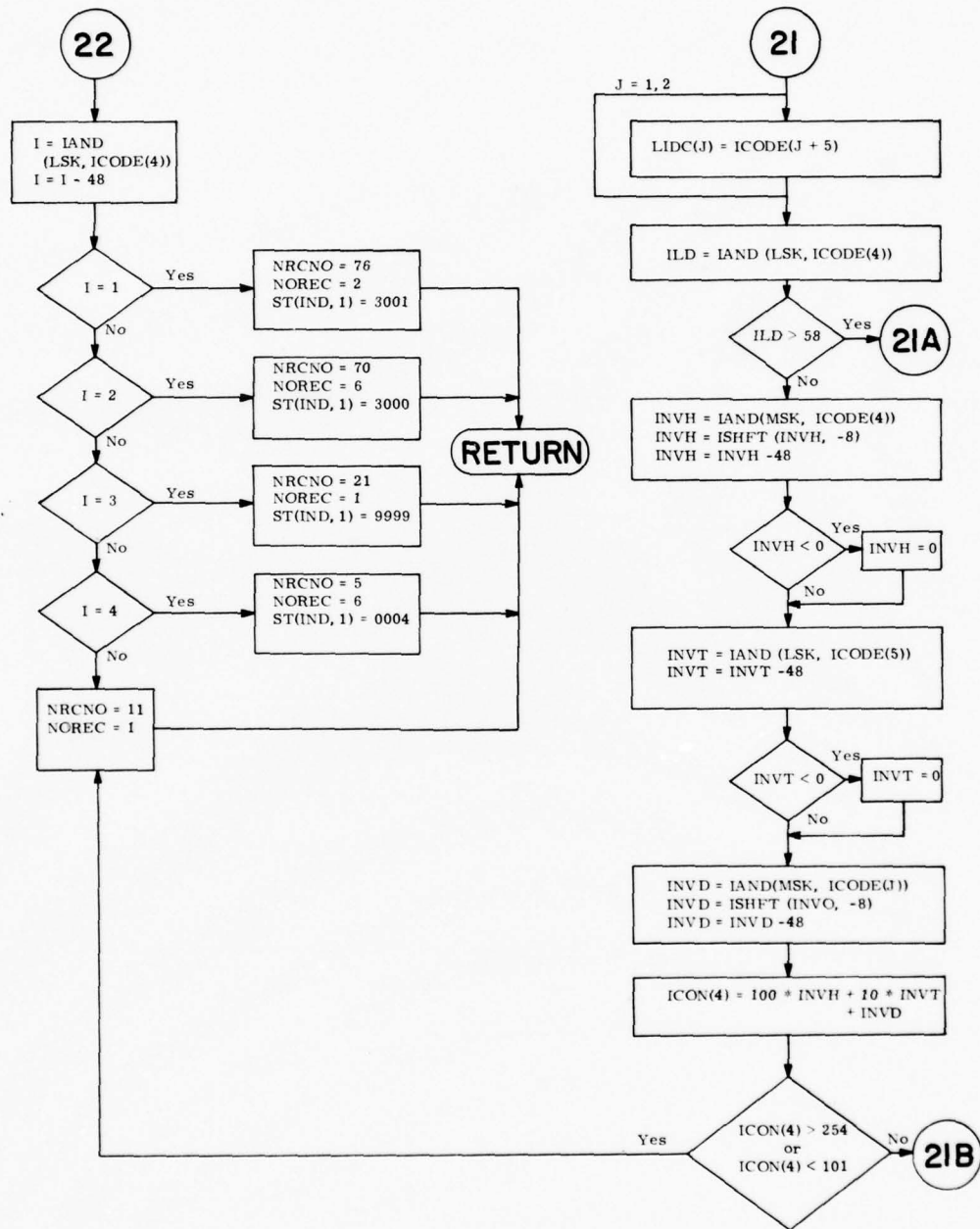


Figure 1-13. (Cont.)

P3001 (cont.)

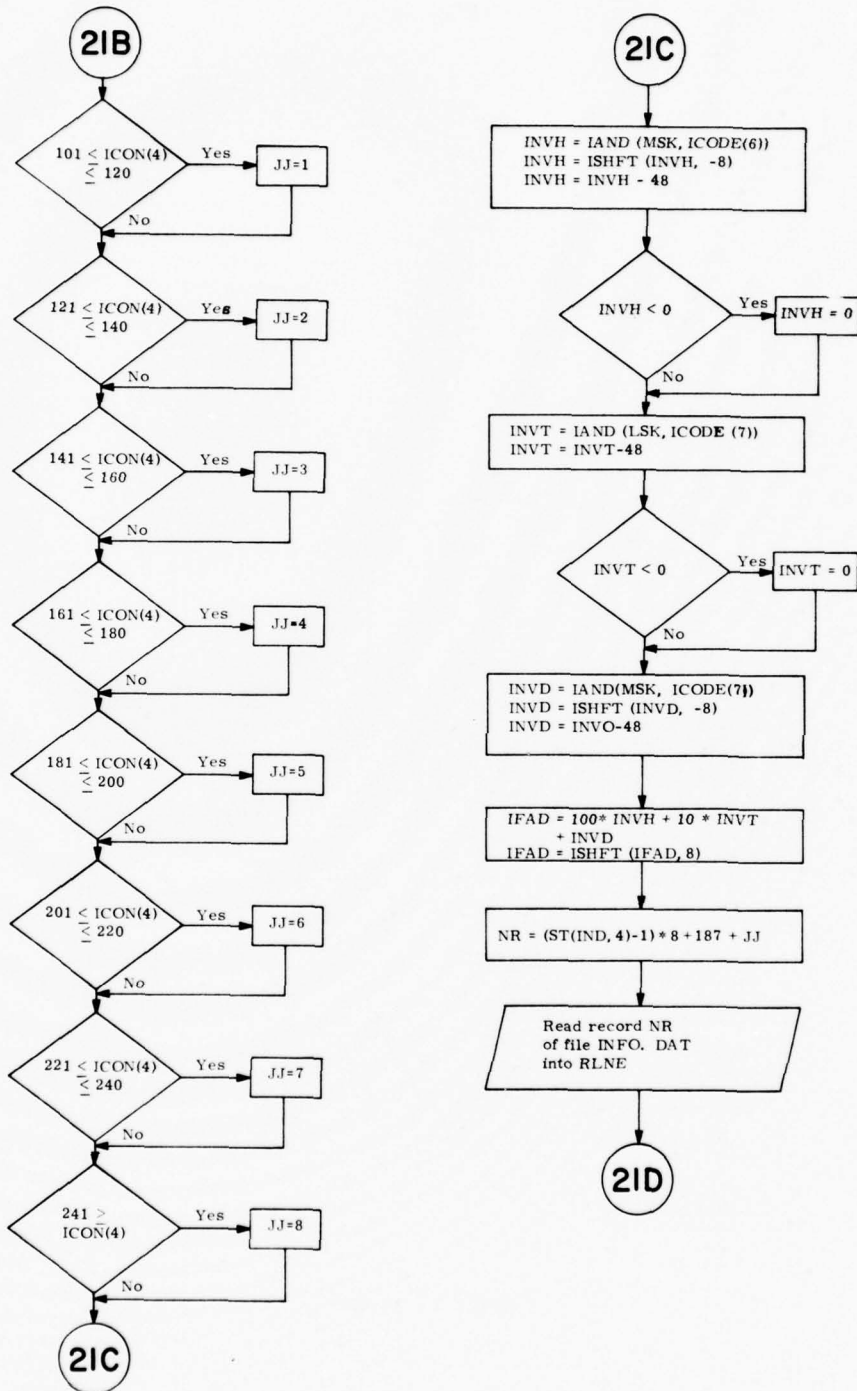


Figure 1-13. (Cont.)

P300I (cont.)

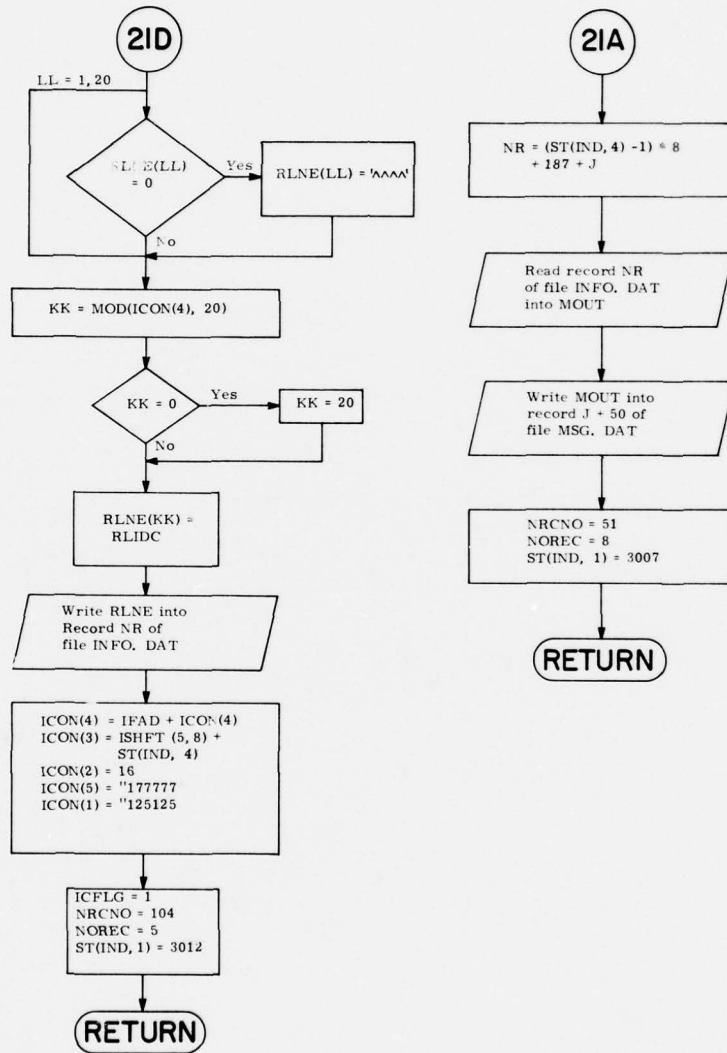


Figure 1-13. (Cont.)

FORTRAN IV V01B-02
 CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:10:34

PAGE 001

P3001.OBJ=P3001.FOR/NOSN/LI:1

```

0001      SUBROUTINE P3001
0002      REAL*8 NDWPM, NWD, MOUT, NIM, NDI, LID, TAB, ND, SKB
0003      REAL*4 RLIDC, RLNE, RSFC
0004      INTEGER ST
C
0005      SUBS 3007 THRU 3012
0006      COMMON NDI, LID, TAB, NWD, ST(3,9), IND
0007      COMMON /DSK/ I1, I2, I3, I4, I5, I6, I16
0008      COMMON /S230/ ND
0009      COMMON /S301/ I TYP
0010      COMMON /U000/ MOUT(10,11), NRCNO, NOREC
0011      COMMON /LOOP/ ICODE(128), MSK, LSK
0012      COMMON /CPAC/ ICON(128), ICFLG
0013      DIMENSION ICHAR(4), LIDC(2), RLNE(20)
0014      EQUIVALENCE (ICAR, NDWPM)
0015      EQUIVALENCE (LIDC, RLIDC)
0016      DATA SKB/' '
0017      DATA NIM/'NOT IMPL'/
0018      DATA ISFC/'120240'/
0019      DATA ICVA, LEOP/'125125,'177777'/
0020      DATA RSFC/' '
C
0021      PICK UP WHERE P3000 LEFT OFF
0022      IF (ST(IND,1) .EQ. 3006) GOTO 26
0023      IF (ST(IND,1) .EQ. 3007) GO TO 27
0024      IF (ST(IND,1) .EQ. 3008) GO TO 28
0025      IF (ST(IND,1) .EQ. 3010) GO TO 30
0026      IF (ST(IND,1) .EQ. 3012) GO TO 32
0027      TAB RESPONSE - WAS SUBROUTINE 3007
C
0030      41 FORMAT(A8)
0031      27 NRCNO=86
0032      NOREC=3
0033      ST(IND,1)=3006
0034      RETURN
C
0035      WORKPAGE PARAMETERS - WAS 3008
0036      28 DO 100 J=1,11
0037      NR=(ST(IND,4)-1)*11+275+J
0038      100 READ(7,NR)(MOUT(I,J), I=1,10)
0039      L=IAND(LSK, ICODE(4))
0040      L=L-48
0041      IF (L .EQ. 30) GOTO 75
0042      ICODE(4)=IAND(LSK, ISFC)+IAND(MSK, ICODE(4))
0043      DO 200 J=1,4
0044      200 ICHAR(J)=ICODE(J+3)
0045      IF (L .EQ. 1) L=2
0046      L=L+1
0047      MOUT(7,L)=NDWPM
0048      DO 110 J=1,11
0049      NR=(ST(IND,4)-1)*11+275+J
0050      110 WRITE(7,NR)(MOUT(I,J), I=1,10)
0051      DO 120 J=1,11
0052      DO 120 I=1,10
0053      120 MOUT(I,J)=SKB
0054      NRCNO=104
0055      NOREC=5
0056      ST(IND,1)=3012
0057      RETURN
0058
0059      75 DO 76 J=1,11

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]FRI 18-MAR-77 19:10:34
P3001.OBJ=P3001.FOR/NOSN/LI:1

PAGE 002

```
0060 76 WRITE(8'J+58,ERR=999)(MOUT(I,J),I=1,10)
0061 NRCNO=59
0062 NOREC=11
0063 ST(IND,1)=3010
0064 RETURN
C
0065 30 NWD RESPONSE - WAS 3010
0066 NRCNO=89
0067 NOREC=11
0068 ST(IND,1)=3008
0069 RETURN
C
0069 32 I=IAND(LSK,ICODE(4))
0070 I=I-48
0071 IF (I .EQ. 1) GO TO 149
0073 IF (I .EQ. 2) GO TO 103
0075 IF (I .EQ. 3) GO TO 108
0077 IF (I .EQ. 4) GO TO 106
0079 999 NRCNO=11
0080 NOREC=1
0081 RETURN
0082 149 NRCNO=76
0083 NOREC = 2
0084 ST(IND,1)=3001
0085 RETURN
0086 103 NRCNO=70
0087 NOREC=6
0088 ST(IND,1)=3000
0089 RETURN
0090 108 NRCNO=21
0091 NOREC=1
0092 ST(IND,1)=9999
0093 RETURN
0094 106 NRCNO=5
0095 NOREC=6
0096 ST(IND,1)=0004
0097 RETURN
C
0098 26 LID/FAD TABLE
0099 810 DO 810 J=1,2
0100 LIDC(J)=ICODE(J+5)
0101 ILD=IAND(LSK,ICODE(4))
0103 IF (ILD .GT. 58) GOTO 472
0103 INVH=IAND(MSK,ICODE(4))
0104 INVH=ISHFT(INVH,-8)
0105 INVH=INVH-48
0106 IF (INVH .LT. 0) INVH=0
0108 INVT=IAND(LSK,ICODE(5))
0109 INVT=INVT-48
0110 IF (INVT .LT. 0) INVT=0
0112 INVO=IAND(MSK,ICODE(5))
0113 INVO=ISHFT(INVO,-8)
0114 INVO=INVO-48
0115 ICON(4)=100*INVH+10*INVT+INVO
0116 IF (ICON(4) .GT. 254) GOTO 999
0118 IF (ICON(4) .LT. 101) GOTO 999
0120 JJ=1
0121 IF (ICON(4) .GE. 121) JJ=2
```

FORTRAN IV V01B-02 FRI 18-MAR-77 19:10:34 PAGE 003
CORE=08K, UIC=[20,20] P3001.0BJ=P3001.FOR/NOSN/LI:1

```
0123     IF (ICON(4) .GE. 141) JJ=3
0125     IF (ICON(4) .GE. 161) JJ=4
0127     IF (ICON(4) .GE. 181) JJ=5
0129     IF (ICON(4) .GE. 201) JJ=6
0131     IF (ICON(4) .GE. 221) JJ=7
0133     IF (ICON(4) .GE. 241) JJ=8
0135     INVH=IAND(MSK,ICOD(6))
0136     INVH=ISHFT(INVH,-8)
0137     INVH=INVH-48
0138     IF (INVH .LT. 0) INVH=0
0140     INVT=IAND(LSK,ICOD(7))
0141     INVT=INVT-48
0142     IF (INVT .LT. 0) INVT=0
0144     INVO=IAND(MSK,ICOD(7))
0145     INVO=ISHFT(INVO,-8)
0146     INVO=INVO-48
0147     IFAD=100*INVH+10*INVT+INVO
0148     IFAD=ISHFT(IFAD,8)
0149     NR=(ST(IND,4)-1)*8+187+JJ
0150     READ(7'NR,ERR=999)(RLNE(I),I=1,20)
0151     DO 430 LL=1,20
0152 430 IF (RLNE(LL) .EQ. 0) RLNE(LL)=RSPC
0154     KK=MOD(ICON(4),20)
0155     IF (KK .EQ. 0) KK=20
0157     RLNE(KK)=RLIDC
0158     WRITE(7'NR,ERR=999)(RLNE(I),I=1,20)
0159 420 ICON(4)=IFAD+ICON(4)
0160     ICON(3)=ISHFT(5,8)+ST(IND,4)
0161     ICON(2)=16
0162     ICON(5)=LEOP
0163     ICON(1)=ICVA
0164     ICFLG=1
0165     NRCNO=104
0166     NOREC=5
0167     ST(IND,1)=3012
0168     RETURN
0169 472 DO 471 J=1,8
0170     NR=(ST(IND,4)-1)*8+187+J
0171     READ(7'NR,ERR=999)(MOUT(I,J),I=1,10)
0172 471 WRITE(8'J+50)(MOUT(I,J),I=1,10)
0173     NRCNO=51
0174     NOREC=8
0175     ST(IND,1)=3007
0176     RETURN
0177     END
```

>
*** MTO: -- SELECT ERROR

P4000

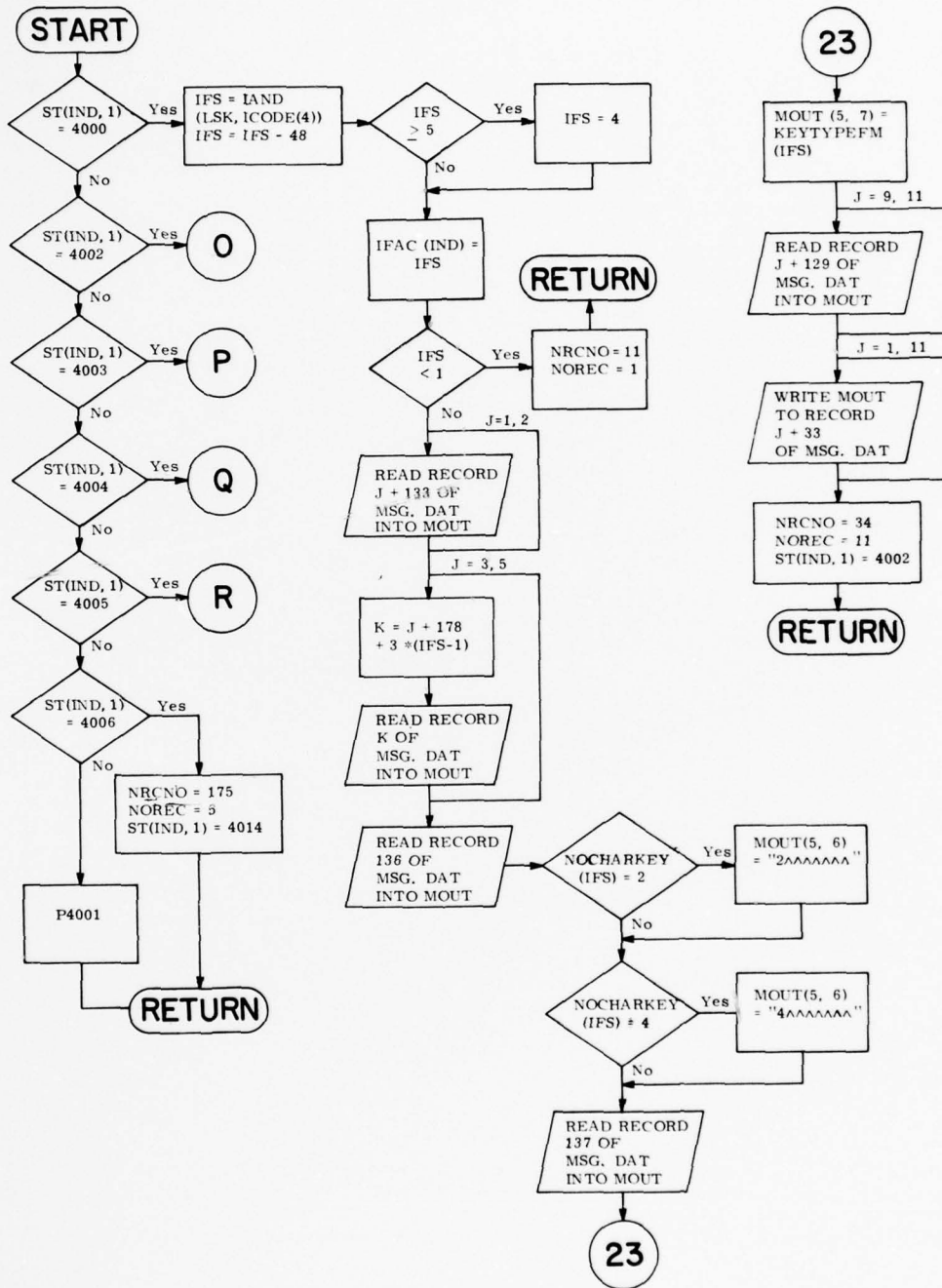


Figure 1-14. P4000

P4000 (cont.)

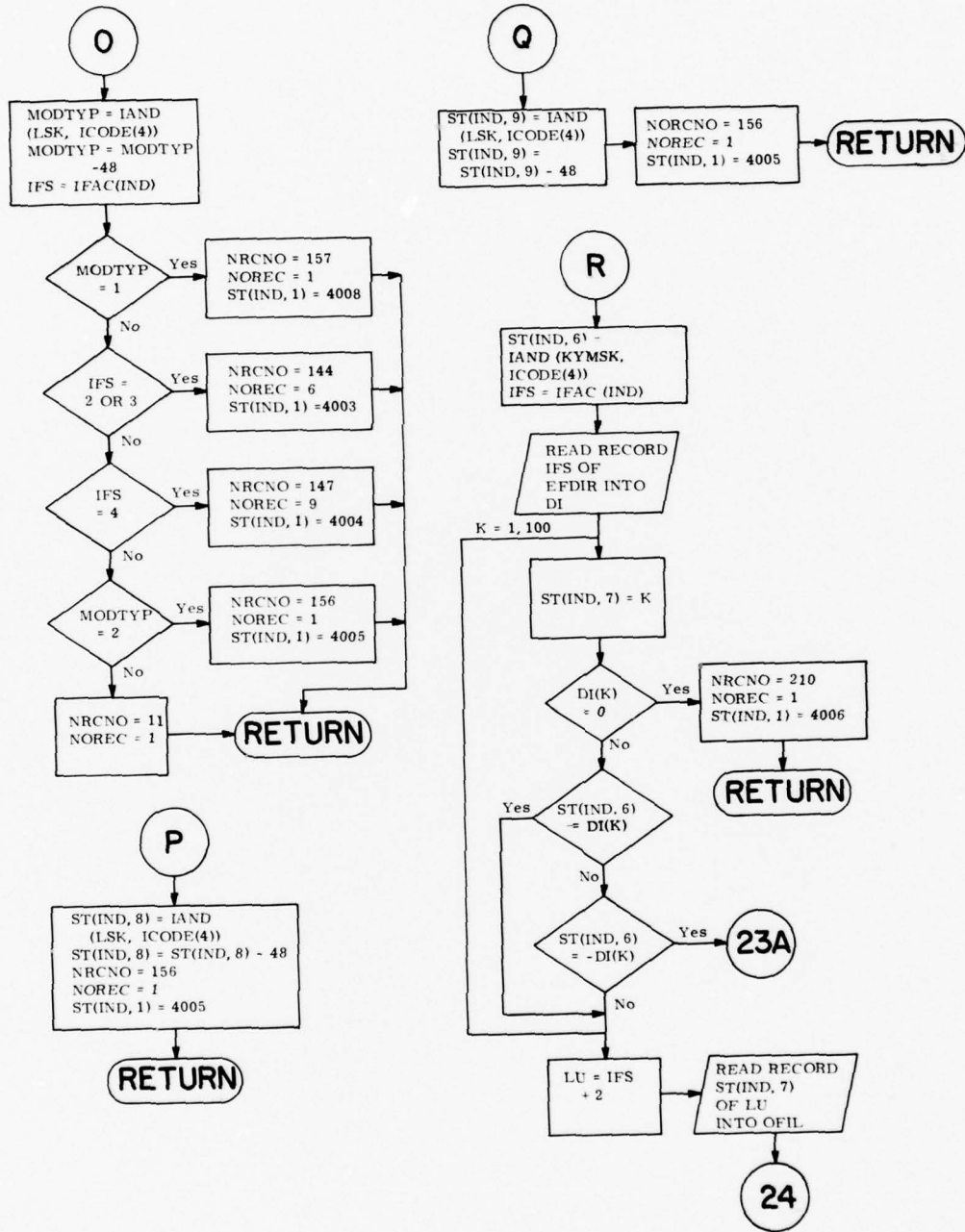


Figure 1-14. (Cont.)

P4000(cont.)

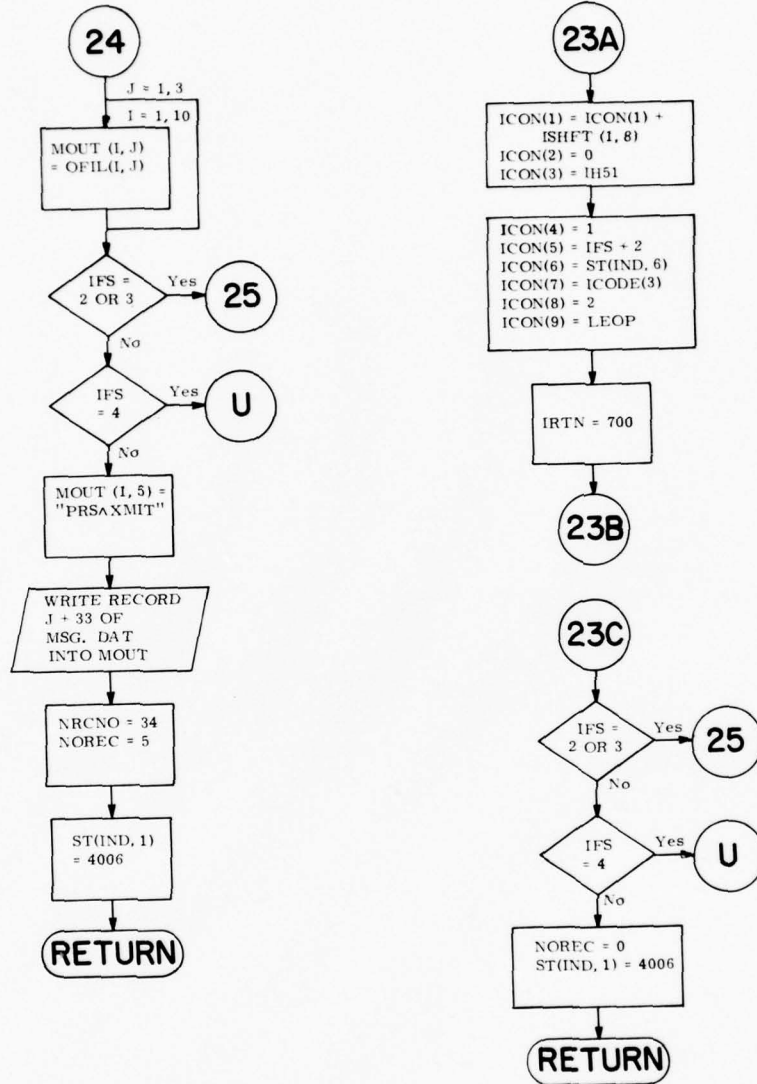


Figure 1-14. (Cont.)

P4000(cont.)

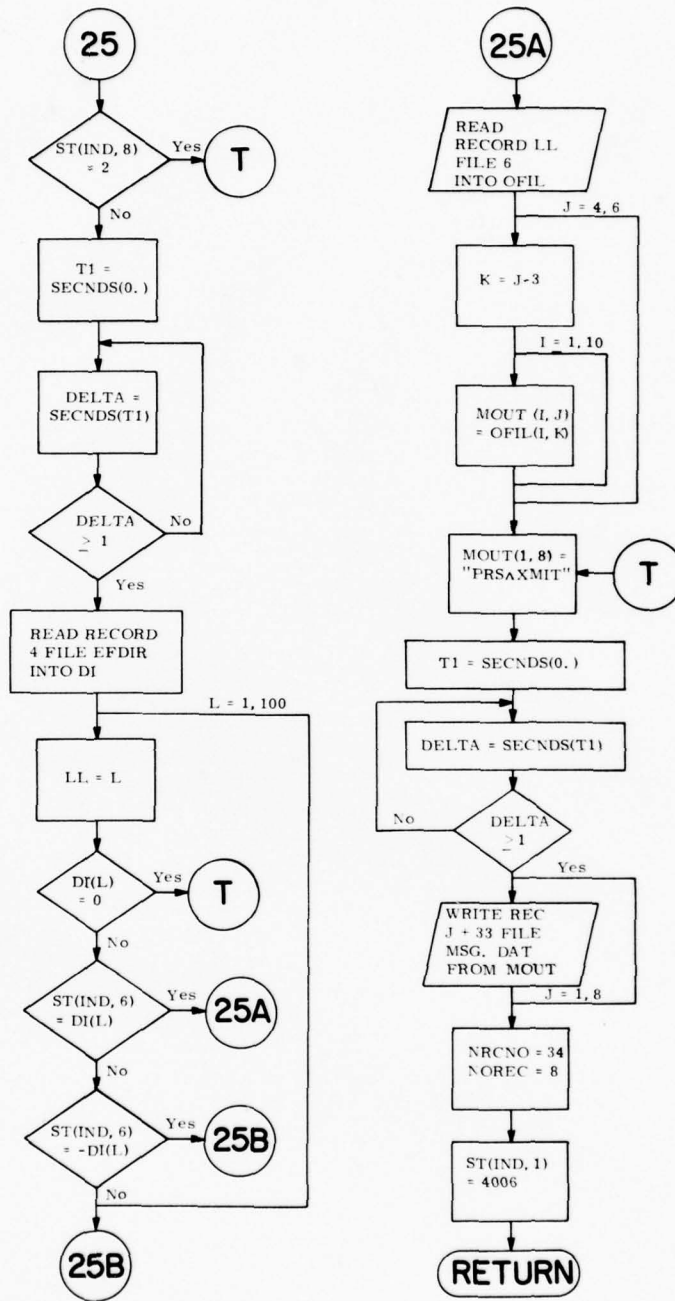


Figure 1-14. (Cont.)

P4000(cont.)

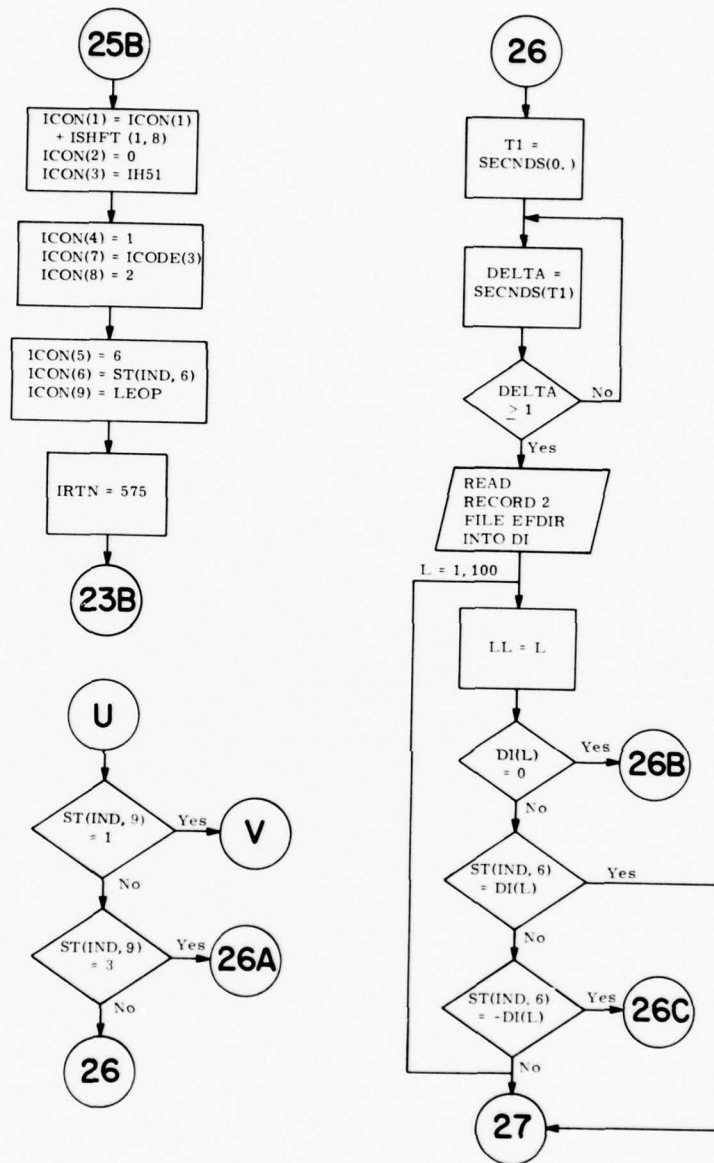


Figure 1-14. (Cont.)

P4000 (cont.)

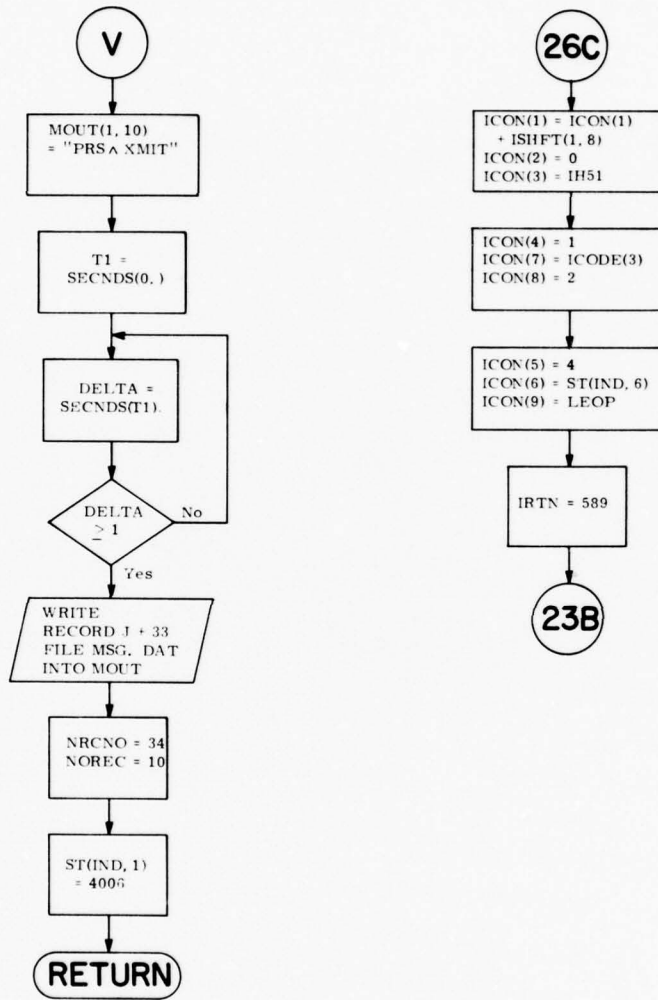


Figure 1-14. (Cont.)

P4000(cont.)

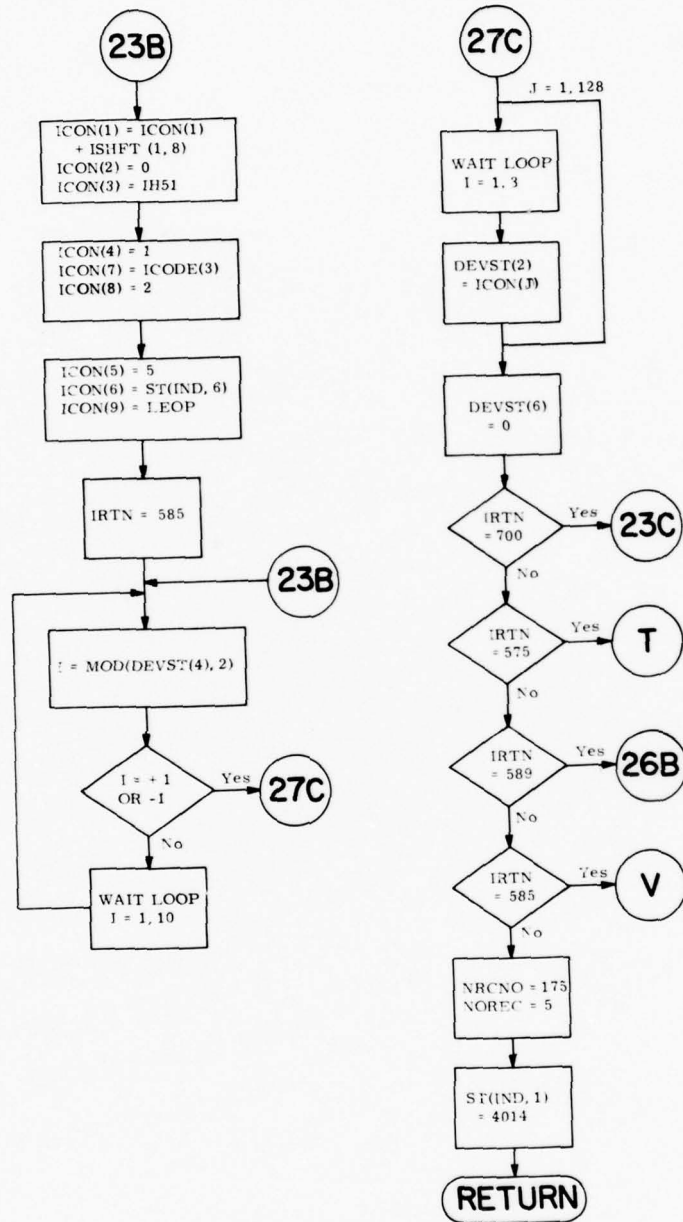


Figure 1-14. (Cont.)

P4000(cont.)

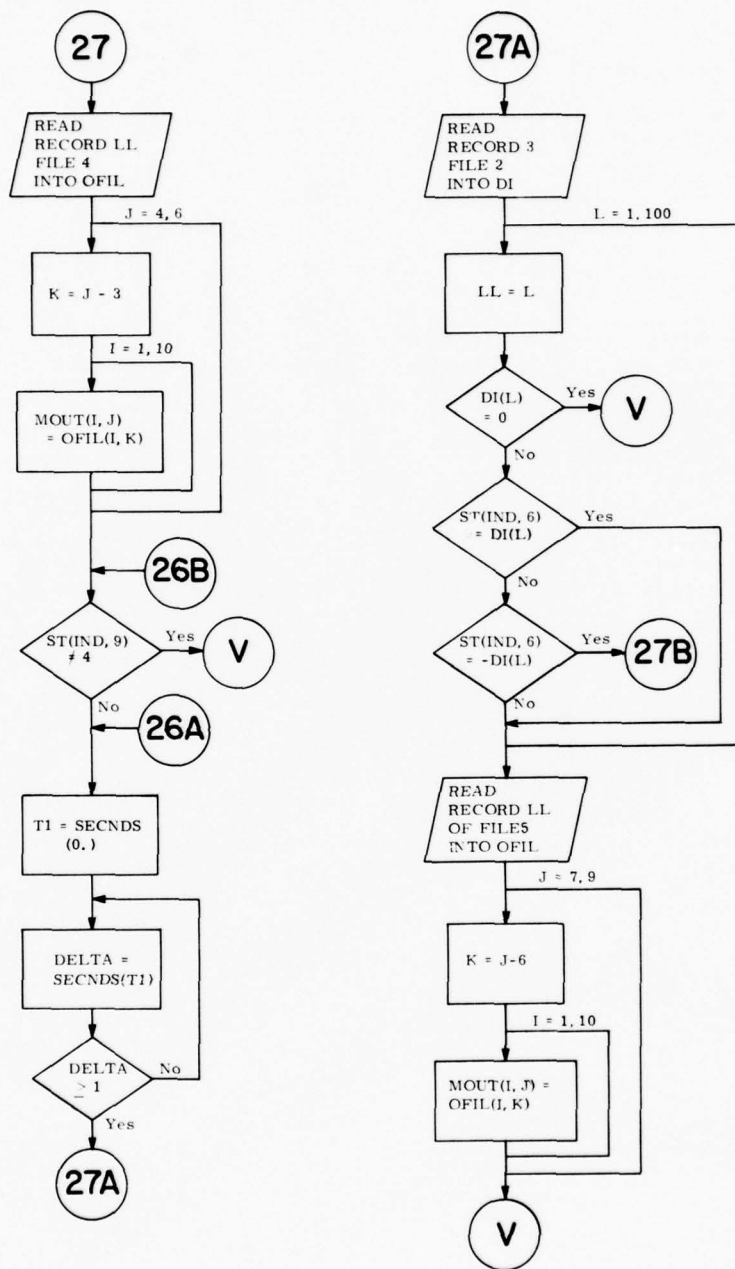


Figure 1-14. (Cont.)

FORTRAN IV V01B-02 FRI 18-MAR-77 19:09:18 PAGE 001
 CORE=08K, UIC=(20,20) P4000, OBJ=P4000, FOR/NOSN/LI:1

```

0001      SUBROUTINE P4000
          C
          C
          C      FILE ACCESS MODE OF OPERATION
          C
0002      REAL*8 MOUT, Q20, Q40, KEYTYPEFM, XMT, OFIL,
          1      ST1, STD, ADE, UPD, NDI, LID, TRB, NWD
0003      INTEGER ST, DI
0004      INTEGER DEVST(6)
0005      COMMON NDI, LID, TRB, NWD, ST(3,9), IND
0006      DIMENSION OFIL(10,3), DI(100)
0007      COMMON /DSK/ I1, I2, I3, I4, I5, I6, I16
0008      COMMON /U000/ MOUT(10,11), NRCNO, NOREC
0009      COMMON /M004/ NOCHARKEY(10), KEYTYPEFM(10)
0010      COMMON /F01/ IFAC(3)
0011      COMMON /LOOP/ ICODE(128), MSK, LSK
0012      COMMON /CPAC/ ICON(128), ICFLG
0013      COMMON /M1710/ DEVST
0014      DATA Q20, Q40 /8H2                    , 8H4            /
0015      DATA XMT, ST1, STD/
          1'PRS XMIT', 'ST.            /, 'ST. DO Y'/
0016      DATA NOCHARKEY/2, 2, 2, 2, 4, 2, 4, 2, 2, 2/
0017      DATA KEYTYPEFM/8H LET-DIG, 8H LET-DIG, 8H LET-DIG, 8H LET-DIG,
          18HALPHANUM, 8H LET-DIG, 8HALPHANUM,
          28HLET-DIG , 8H LET-DIG, 8H    DIGIT/
0018      DATA ADE/    ' ADDED'/
0019      DATA UPD/8H UPDATED/
0020      DATA IH51, LEOP, KYMSK/"2401, "177777, "775777/
0021      IF (ST(IND,1) .EQ. 4000) GOTO 72
0023      IF (ST(IND,1) .EQ. 4002) GOTO 22
0025      IF (ST(IND,1) .EQ. 4003) GOTO 23
0027      IF (ST(IND,1) .EQ. 4004) GOTO 24
0029      IF (ST(IND,1) .EQ. 4005) GOTO 25
0031      IF (ST(IND,1) .EQ. 4006) GOTO 26
0033      CALL P4001
0034      RETURN
0035      22 MODTYP=IAND(LSK, ICODE(4))
0036      MODTYP=MODTYP-48
0037      IFS=IFAC(IND)
          C      RD TYPE OF MOD
          C      NO MODIFICATION - ACCESS ONLY
          C      CK FOR CROSS-REFERENCING
0038      IF (MODTYP .EQ. 1) GOTO 260
0040      IF (IFS .EQ. 2) GOTO 278
0042      IF (IFS .EQ. 3) GOTO 278
0044      IF (IFS .EQ. 4) GOTO 279
0046      IF (MODTYP .EQ. 2) GOTO 263
0048      GOTO 73
0049      278 NRCNO=140
0050      NOREC=6
0051      ST(IND,1)=4003
0052      RETURN
0053      279 NRCNO=147
0054      NOREC=9
0055      ST(IND,1)=4004
0056      RETURN

```

```

FORTRAN IV      V01B-02      FRI 18-MAR-77 19:03:18      PAGE 002
CORE=08K, UIC=(20,20)      P4000,00J=P4000, FOR/NOSN/LI:1

0057   260  NRCNO=157
0058       NOREC=1
0059       ST(IND,1)=4003
0060       RETURN
0061   263  NRCNO=156
0062       ST(IND,1)=4005
0063       NOREC=1
0064       RETURN
0065   23   ST(IND,8)=IAND(LSK,ICODE(4))
0066       ST(IND,8)=ST(IND,8)-48
C
0067       READ CROSS-REF 1
0068       NRCNO=156
0069       NOREC=1
0069       ST(IND,1)=4005
0070       RETURN
0071   24   ST(IND,9)=IAND(LSK,ICODE(4))
0072       ST(IND,9)=ST(IND,9)-48
C
0073       RD C-R 2
0074       NRCNO=156
0075       NOREC=1
0075       ST(IND,1)=4005
0076       RETURN
0077   25   ST(IND,6)=IAND(KYMSK,ICODE(4))
0078       IFS=IFAC(IND)
C
0079       READ(2,IFS,ERR=73)(DI(I),I=1,100)
0080       DO 550 K=1,100
0081       ST(IND,7)=K
0082       IF (DI(K) .EQ. 0) GOTO 565
0084       IF (ST(IND,6) .EQ. DI(K)) GOTO 560
0086       IF (ST(IND,6) .EQ. -DI(K)) GOTO 600
0088   550  CONTINUE
C
0089       KEY EXISTS
0089   560  LU = IFS+2
0090       READ(LU,ST(IND,7),ERR=73)((OFIL(I,J),I=1,10),J=1,3)
0091       DO 582 J=1,3
0092       DO 582 I=1,10
0093   582  MOUT(I,J)=OFIL(I,J)
0094       IF (IFS .EQ. 2) GOTO 570
0096       IF (IFS .EQ. 3) GOTO 570
0098       IF (IFS .EQ. 4) GOTO 580
C
0100       MOUT(1,5)=XMT
0101       DO 5100 J=1,5
0102   5100 WRITE(8,J+33)(MOUT(I,J),I=1,10)
0103       NRCNO=34
0104       NOREC=5
0105       ST(IND,1)=4006
0106       RETURN
0107   600  ICON(1)=ICON(1)+ISHFT(1,8)
0108       ICON(2)=0
0109       ICON(3)=IH51
0110       ICON(4)=1
0111       ICON(5)=IFS+2
0112       ICON(6)=ST(IND,6)
0113       ICON(7)=ICODE(3)

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]FRI 18-MAR-77 19:08:18 PAGE 003
P4000. OBJ=P4000. FOR/NOSN/LI:1

```
0114            ICON(8)=2
0115            ICON(9)=LEOP
0116            IRTN=700
0117            GOTO 710
0118        700 IF (IFS .EQ. 2) GOTO 570
0120            IF (IFS .EQ. 3) GOTO 570
0122            IF (IFS .EQ. 4) GOTO 580
0124            NOREC=0
0125            ST(IND,1)=4006
0126            RETURN
          C        RECORD NOT EXIST(NK)
0127        565 NRCNO=210
0128            NOREC=1
0129            ST(IND,1)=4006
0130            RETURN
          C        C-R IFS=04
0131        570 IF (ST(IND,8) .EQ. 2) GOTO 575
0133            T1=SECNDS(0.)
0134        1000 DELTA=SECNDS(T1)
0135            IF (DELTA .GE. 1) GOTO 640
0137            GOTO 1000
          C        ELSE LOCATE RECORD
0138        640 READ(2,4,ERR=73)((DI(I), I=1,100)
0139            DO 571 L=1,100
0140            LL=L
0141            IF (DI(L) .EQ. 0) GOTO 575
0143            IF (ST(IND,6) .EQ. DI(L)) GOTO 572
0145            IF (ST(IND,6) .EQ. -DI(L)) GOTO 660
0147        571 CONTINUE
0148        572 READ(6,LL,ERR=73)((OFIL(I,J), I=1,10), J=1,3)
0149            DO 573 J=4,6
0150            K=J-3
0151            DO 573 I=1,10
0152        573 MOUT(I,J)=OFIL(I,K)
0153        575 MOUT(1,8)=XMT
0154            T1=SECNDS(0.)
0155        1010 DELTA=SECNDS(T1)
0156            IF (DELTA .GE. 1) GOTO 1020
0158            GOTO 1010
0159        1020 DO 5110 J=1,8
0160        5110 WRITE(8,J+33)(MOUT(I,J), I=1,10)
0161        661        NRCNO=34
0162            NOREC=8
0163            ST(IND,1)=4006
0164            RETURN
0165        660        ICON(1)=ICON(1)+ISHFT(1,8)
0166            ICON(2)=0
0167            ICON(3)=IHS1
0168            ICON(4)=1
0169            ICON(7)=ICDCE(2)
0170            ICON(8)=2
0171            ICON(5)=6
0172            ICON(6)=ST(IND,6)
0173            ICON(9)=LEOP
0174            IRTN=575
0175            GOTO 710
```

FORTRAN IV V01B-02
 CORE=08K, UIC=[20, 20]

FRI 18-MAR-77 19:08:18

PAGE 004

P4000. OBJ=P4000. FOR/NOSN/LI:1

```

C      C-R IFS=2,3
0176   580 IF (ST(IND,9) .EQ. 1) GOTO 585
0178     IF (ST(IND,9) .EQ. 3) GOTO 583
0180     T1=SECNDS(0)
0181  1030 DELTA=SECNDS(T1)
0182     IF (DELTA .GE. 1) GOTO 650
0184     GOTO 1030
0185   650 READ(2,2,ERR=73)(DI(I), I=1,100)
0186     DO 584 L=1,100
0187     LL=L
0188     IF (DI(L) .EQ. 0) GOTO 589
0190     IF (ST(IND,6) .EQ. DI(L)) GOTO 586
0192     IF (ST(IND,6) .EQ. -DI(L)) GOTO 670
0194   584 CONTINUE
0195   586 READ(4,LL,ERR=73)((OFIL(I,J), I=1,10), J=1,3)
0196     DO 87 J=4,6
0197     K=J-3
0198     DO 87 I=1,10
0199     87 MOUT(I,J)=OFIL(I,K)
0200   589 IF (ST(IND,9) .NE. 4) GOTO 585
0202   583 T1=SECNDS(0)
0203  1040 DELTA=SECNDS(T1)
0204     IF (DELTA .GE. 1) GOTO 1050
0206     GOTO 1040
0207  1050 READ(2,2,ERR=73)(DI(I), I=1,100)
0208     DO 588 L=1,100
0209     LL=L
0210     IF (DI(L) .EQ. 0) GOTO 585
0212     IF (ST(IND,6) .EQ. DI(L)) GOTO 591
0214     IF (ST(IND,6) .EQ. -DI(L)) GOTO 680
0216   588 CONTINUE
0217   591 READ(5,LL,ERR=73)((OFIL(I,J), I=1,10), J=1,3)
0218     DO 592 J=7,9
0219     K=J-6
0220     DO 592 I=1,10
0221     592 MOUT(I,J)=OFIL(I,K)
0222   585 MOUT(1,10)=XMT
0223     T1=SECNDS(0)
0224  1060 DELTA=SECNDS(T1)
0225     IF (DELTA .GE. 1) GOTO 1070
0227     GOTO 1060
0228  1070 DO 5120 J=1,10
0229  5120 WRITE(8,J+35)(MOUT(I,J), I=1,10)
0230     NRCNO=34
0231     NOREC=10
0232     ST(IND,1)=4006
0233     RETURN
0234   670 ICON(1)=ICON(1)+ISHFT(1,8)
0235     ICON(2)=0
0236     ICON(3)=IHS1
0237     ICON(4)=1
0238     ICON(7)=IC0DE(3)
0239     ICON(8)=2
0240     ICON(5)=4
0241     ICON(6)=ST(IND,6)
0242     ICON(9)=LEOP
  
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:08:18

PAGE 005

P4000.06J=P4000.FOR/NOSN/LI:1

```
0243      IRTN=589
0244      GOTO 710
0245      680 ICON(1)=ICON(1)+ISHFT(1,8)
0246      ICON(2)=0
0247      ICON(3)=IH51
0248      ICON(4)=1
0249      ICON(7)=IC00E(3)
0250      ICON(8)=2
0251      ICON(5)=5
0252      ICON(6)=ST(IND,6)
0253      ICON(9)=LEQP
0254      IRTN=585
0255      710 I=MOD(DEVST(4),2)
0256      IF (I .EQ. 1) GOTO 40
0258      IF (I .EQ. -1) GOTO 40
0260      DO 60 I=1,10
0261      60 CONTINUE
0262      GOTO 710
0263      40 DO 71 J=1,128
0264      DO 80 I=1,3
0265      80 CONTINUE
0266      71 DEVST(2)=ICON(J)
0267      DEVST(6)=0
0268      IF (IRTN .EQ. 700) GOTO 700
0270      IF (IRTN .EQ. 575) GOTO 575
0272      IF (IRTN .EQ. 589) GOTO 589
0274      IF (IRTN .EQ. 585) GOTO 585
0276      26 NRCNO=175
0277      NOREC=5
0278      ST(IND,1)=4014
0279      RETURN
C      GET FILE SELECTED
0280      72 IFS =IAND(LSK,IC00E(4))
0281      IFS=IFS-48
C      CK FOR INVALID ENTRY
0282      IF (IFS .GE. 9) IFS=4
0284      IFAC(IND)=IFS
0285      IF (IFS .GE. 1) GOTO 70
0287      73 NRCNO=11
0288      NOREC=1
0289      RETURN
70      DO 75 J=1,2
0291      75 READ(8'133+J)(MOUT(I,J),I=1,10)
0292      DO 76 J=3,5
0293      K=J+178+3*(IFS-1)
0294      76 READ(8'K)(MOUT(I,J),I=1,10)
0295      READ(8'136)(MOUT(I,6),I=1,10)
0296      IF (NOCHARKEY(IFS) .EQ. 2) MOUT(5,6)=020
0298      IF (NOCHARKEY(IFS) .EQ. 4) MOUT(5,6)=040
0300      READ(8'137)(MOUT(I,7),I=1,10)
0301      MOUT(5,7)=KEYTYPEFM(IFS)
0302      DO 77 J=9,11
0303      77 READ(8'129+J)(MOUT(I,J),I=1,10)
0304      DO 95 J=1,11
0305      95 WRITE(8'J+22)(MOUT(I,J),I=1,10)
0306      NRCNO=34
```

FORTRAN IV V010-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:08:18

PAGE 006

P4000.OBJ=P4000.FOR./NOEN/LI:1

0307 NOREC=11
0308 ST(IND,1)=4002
0309 RETURN
0310 END

FORTRAN IV DIAGNOSTICS

[WARNING] MSG #092 VARIABLE "NOCHAR" NAME EXCEEDS 6 CHARACTERS
[WARNING] MSG #092 VARIABLE "KEYTYP" NAME EXCEEDS 6 CHARACTERS

FOR -- [P4000] ERRORS: 0, WARNINGS: 2
>

FORTRAN IV V01B-02 FRI 18-MAR-77 19:20:00 PAGE 001
 CORE=03K, UIC=(20,20) 40001.0BJ=P40001.FOR/NOSN/LI:1

```

0001      SUBROUTINE P4000
          C
          C
          C      FILE ACCESS MODE OF OPERATION
          C
0002      REAL*8 MOUT, Q20, Q40, KEYTYPEFM, XMT, OFIL,
          1      ST1, STD, ADE, UPD, NDI, LID, TRB, NMD
0003      INTEGER ST, DI
0004      INTEGER DEVST(6)
0005      COMMON NDI, LID, TRB, NMD, ST(3,9), IND
0006      DIMENSION OFIL(10,3), DI(100)
0007      COMMON /DSK/ I1, I2, I3, I4, I5, I6, I16
0008      COMMON /U000/ MOUT(10,11), NRCNO, NOREC
0009      COMMON /M004/ NOCHARKEY(10), KEYTYPEFM(10)
0010      COMMON /F01/ IFAC(3)
0011      COMMON /LOOP/ ICODE(128), MSK, LSK
0012      COMMON /CPAC/ ICON(128), ICFLG
0013      COMMON /M1710/ DEVST
0014      DATA Q20, Q40 /8H2                    , 8H4            /
0015      DATA XMT, ST1, STD/
          1 'PRS XMIT', 'ST.            ', 'ST. DO Y' /
0016      DATA NOCHARKEY/2, 2, 2, 2, 4, 2, 4, 2, 2, 2/
0017      DATA KEYTYPEFM/8H LET-DIG, 8H LET-DIG, 8H LET-DIG, 8H LET-DIG,
          13HALPHANUM, 8H LET-DIG, 8HALPHANUM,
          28HLET-DIG, 8H LET-DIG, 8H    DIGIT/
0018      DATA ADE/    ' ADDED'/
0019      DATA UPD/8H UPDATED/
0020      DATA IH51, LEOP, KYMSK/"405, "177777, "775777.
0021      IF (ST(IND, 1) .EQ. 4000) GOTO 72
0023      IF (ST(IND, 1) .EQ. 4002) GOTO 22
0025      IF (ST(IND, 1) .EQ. 4003) GOTO 23
0027      IF (ST(IND, 1) .EQ. 4004) GOTO 24
0029      IF (ST(IND, 1) .EQ. 4005) GOTO 25
0031      IF (ST(IND, 1) .EQ. 4006) GOTO 26
0033      CALL P4001
0034      RETURN
0035      22 MOOTYP=IAND(LSK, ICODE(4))
0036      MOOTYP=MOOTYP-48
0037      IFS=IFAC(IND)
          C      RD TYPE OF MOD
          C      NO MODIFICATION - ACCESS ONLY
          C      CK FOR CROSS-REFERENCING
0038      IF (MOOTYP .EQ. 1) GOTO 260
0040      IF (IFS .EQ. 2) GOTO 278
0042      IF (IFS .EQ. 3) GOTO 278
0044      IF (IFS .EQ. 4) GOTO 279
0046      IF (MOOTYP .EQ. 2) GOTO 263
0048      GOTO 73
0049      278 NRCNO=140
0050      NOREC=6
0051      ST(IND, 1)=4003
0052      RETURN
0053      279 NRCNO=147
0054      NOREC=9
0055      ST(IND, 1)=4004
0056      RETURN

```

FORTRAN IV V018-02
CORE=03K, UIC=[20,20]

FRI 18-MAR-77 19:20:00 PAGE 002
40001.08J=P40001.FOR/NOSN/LI:1

```

0057 260 NRCNO=157
0058     NOREC=1
0059     ST(IND,1)=4008
0060     RETURN
0061 263 NRCNO=156
0062     ST(IND,1)=4005
0063     NOREC=1
0064     RETURN
0065 23  ST(IND,8)=IAND(LSK,IC0DE(4))
0066     ST(IND,8)=ST(IND,8)-48
C     READ CROSS-REF 1
0067     NRCNO=156
0068     NOREC=1
0069     ST(IND,1)=4005
0070     RETURN
0071 24  ST(IND,9)=IAND(LSK,IC0DE(4))
0072     ST(IND,9)=ST(IND,9)-48
C     RD C-R 2
0073     NRCNO=156
0074     NOREC=1
0075     ST(IND,1)=4005
0076     RETURN
0077 25  ST(IND,6)=IAND(KYMSK,IC0DE(4))
0078     IFS=IFAC(IND)
C     GET ST(IND,6)
0079     READ(2,IFS,ERR=73)((O(I),I=1,100)
0080     DO 550 K=1,100
0081     ST(IND,7)=K
0082     IF (D(K) .EQ. 0) GOTO 565
0084     IF (ST(IND,6) .EQ. D(K)) GOTO 560
0086     IF (ST(IND,6) .EQ. -D(K)) GOTO 600
0088     550 CONTINUE
C     KEY EXISTS
0089     560 LU = IFS+2
0090     READ(LU,ST(IND,7),ERR=73)((OFIL(I,J),I=1,10),J=1,3)
0091     DO 582 J=1,3
0092     DO 582 I=1,10
0093     582 MOUT(I,J)=OFIL(I,J)
0094     IF (IFS .EQ. 2) GOTO 570
0096     IF (IFS .EQ. 3) GOTO 570
0098     IF (IFS .EQ. 4) GOTO 580
C     ELSE NO C-R
0100     MOUT(1,5)=XMT
0101     DO 5100 J=1,5
0102     5100 WRITE(8,J+23)(MOUT(I,J),I=1,10)
0103     NRCNO=24
0104     NOREC=5
0105     ST(IND,1)=4006
0106     RETURN
0107 600  ICON(1)=ICON(1)+ISHFT(1,8)
0108     ICON(2)=0
0109     ICON(3)=IH51
0110     ICON(4)=1
0111     ICON(5)=IFS+2
0112     ICON(6)=ST(IND,6)
0113     ICON(7)=IC0DE(3)

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:20:00

PAGE 003

40001.0BJ=P40001.FOR/NOSN/LI:1

```
0114            ICON(8)=2
0115            ICON(9)=LEOP
0116            IRTN=700
0117            GOTO 710
0118        700 IF (IFS .EQ. 2) GOTO 570
0120            IF (IFS .EQ. 3) GOTO 570
0122            IF (IFS .EQ. 4) GOTO 580
0124            NOREC=0
0125            ST(IND,1)=4006
0126            RETURN
          C        RECORD NOT EXIST(NK)
0127        565 NRCNO=210
0128            NOREC=1
0129            ST(IND,1)=4006
0130            RETURN
          C        C-R IFS=04
0131        570 IF (ST(IND,8) .EQ. 2) GOTO 575
0133            T1=SECNDS(0.)
0134        1000 DELTA=SECNDS(T1)
0135            IF (DELTA .GE. 1) GOTO 640
0137            GOTO 1000
          C        ELSE LOCATE RECORD
0138        640 READ(2,4,ERR=73)((DI(I), I=1,100)
0139            DO 571 L=1,100
0140            LL=L
0141            IF (DI(L) .EQ. 0) GOTO 575
0143            IF (ST(IND,6) .EQ. DI(L)) GOTO 572
0145            IF (ST(IND,6) .EQ. -DI(L)) GOTO 660
0147        571 CONTINUE
0148        572 READ(6,LL,ERR=73)((OFIL(I,J), I=1,10), J=1,3)
0149            DO 573 J=4,6
0150            K=J-3
0151            DO 573 I=1,10
0152            MOUT(I,J)=OFIL(I,K)
0153        575 MOUT(1,8)=XMT
0154            T1=SECNDS(0.)
0155        1010 DELTA=SECNDS(T1)
0156            IF (DELTA .GE. 1) GOTO 1020
0158            GOTO 1010
0159        1020 DO 5110 J=1,8
0160        5110 WRITE(8,J+2)((MOUT(I,J), I=1,10)
0161            661        NRCNO=34
0162            NOREC=8
0163            ST(IND,1)=4006
0164            RETURN
0165        660        ICON(1)=ICON(1)+ISHFT(1,8)
0166            ICON(2)=0
0167            ICON(3)=IHS1
0168            ICON(4)=1
0169            ICON(7)=IC00E(3)
0170            ICON(8)=2
0171            ICON(5)=6
0172            ICON(6)=ST(IND,6)
0173            ICON(9)=LEOP
0174            IRTN=575
0175            GOTO 710
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:20:00

PAGE 004
40001.06J=P40001.FOR/NOSN/LI:1

```

C      C-R IFS=2.3
0176    580 IF (ST(IND,9) .EQ. 1) GOTO 585
0178      IF (ST(IND,9) .EQ. 2) GOTO 583
0180      T1=SECNDS(0)
0181    1030 DELTA=SECNDS(T1)
0182      IF (DELTA .GE. 1) GOTO 650
0184      GOTO 1030
0185    650 READ(2,2,ERR=73)((DI(I), I=1,100)
0186      DO 584 L=1,100
0187      LL=L
0188      IF (DI(L) .EQ. 0) GOTO 589
0190      IF (ST(IND,6) .EQ. DI(L)) GOTO 586
0192      IF (ST(IND,6) .EQ. -DI(L)) GOTO 670
0194    584 CONTINUE
0195    586 READ(4,LL,ERR=73)((OFIL(I,J), I=1,10), J=1,3)
0196      DO 87 J=4,6
0197      K=J-3
0198      DO 87 I=1,10
0199      87 MOUT(I,J)=OFIL(I,K)
0200    589 IF (ST(IND,9) .NE. 4) GOTO 585
0202      T1=SECNDS(0)
0203    1040 DELTA=SECNDS(T1)
0204      IF (DELTA .GE. 1) GOTO 1050
0206      GOTO 1040
0207    1050 READ(2,3,ERR=73)((DI(I), I=1,100)
0208      DO 588 L=1,100
0209      LL=L
0210      IF (DI(L) .EQ. 0) GOTO 585
0212      IF (ST(IND,6) .EQ. DI(L)) GOTO 591
0214      IF (ST(IND,6) .EQ. -DI(L)) GOTO 680
0216    588 CONTINUE
0217    591 READ(5,LL,ERR=73)((OFIL(I,J), I=1,10), J=1,3)
0218      DO 592 J=7,9
0219      K=J-6
0220      DO 592 I=1,10
0221      592 MOUT(I,J)=OFIL(I,K)
0222    585 MOUT(1,10)=XMT
0223      T1=SECNDS(0)
0224    1060 DELTA=SECNDS(T1)
0225      IF (DELTA .GE. 1) GOTO 1070
0227      GOTO 1060
0228    1070 DO 5120 J=1,10
0229    5120 WRITE(8,J+23)((MOUT(I,J), I=1,10)
0230      NRCND=24
0231      NOREC=10
0232      ST(IND,1)=4006
0233      RETURN
0234    670 ICON(1)=ICON(1)+ISHFT(1,8)
0235      ICON(2)=0
0236      ICON(3)=IH51
0237      ICON(4)=1
0238      ICON(7)=ICODE(3)
0239      ICON(8)=2
0240      ICON(5)=4
0241      ICON(6)=ST(IND,6)
0242      ICON(9)=LEOP

```

FORTRAN IV V01B-02
CORE=00K, UIC=[20,20]

FRI 18-MAR-77 19:20:00

PAGE 005

40001.00J=P40001.FOR/NOSN/LI:1

```

0243      IRTN=589
0244      GOTO 710
0245      680 ICON(1)=ICON(1)+ISHFT(1,8)
0246      ICON(2)=0
0247      ICON(3)=IH51
0248      ICON(4)=1
0249      ICON(7)=ICODE(2)
0250      ICON(8)=2
0251      ICON(5)=5
0252      ICON(6)=ST(IND,6)
0253      ICON(9)=LEOP
0254      IRTN=585
0255      710 I=MOD(DEVST(4),2)
0256      IF (I .EQ. 1) GOTO 40
0258      IF (I .EQ. -1) GOTO 40
0260      DO 60 I=1,10
0261      60 CONTINUE
0262      GOTO 710
0263      40 DO 71 J=1,128
0264      DO 80 I=1,3
0265      80 CONTINUE
0266      71 DEVST(2)=ICON(J)
0267      DEVST(6)=0
0268      IF (IRTN .EQ. 700) GOTO 700
0270      IF (IRTN .EQ. 575) GOTO 575
0272      IF (IRTN .EQ. 589) GOTO 589
0274      IF (IRTN .EQ. 585) GOTO 585
0276      26 NRCND=175
0277      NOREC=5
0278      ST(IND,1)=4014
0279      RETURN
C
0280      72 IFS =IAND(LSK,ICODE(4))
0281      IFS=IFS-48
C
0282      CK FOR INVALID ENTRY
0284      IF (IFS .GE. 5) IFS=4
0284      IFAC(IND)=IFS
0285      IF (IFS .GE. 1) GOTO 70
0287      73 NRCND=11
0288      NOREC=1
0289      RETURN
70
0290      DO 75 J=1,2
0291      75 READ(8'132+J')(MOUT(I,J), I=1,10)
0292      DO 76 J=3,5
0293      K=J+178+3*(IFS-1)
0294      76 READ(8'K')(MOUT(I,J), I=1,10)
0295      READ(8'136')(MOUT(I,6), I=1,10)
0296      IF (NOCARKEY(IFS) .EQ. 2) MOUT(5,6)=020
0298      IF (NOCARKEY(IFS) .EQ. 4) MOUT(5,6)=040
0300      READ(8'137')(MOUT(I,7), I=1,10)
0301      MOUT(5,7)=KEYTYPEFM(IFS)
0302      DO 77 J=9,11
0303      77 READ(8'129+J')(MOUT(I,J), I=1,10)
0304      DO 95 J=1,11
0305      95 WRITE(8'J+23')(MOUT(I,J), I=1,10)
0306      NRCND=34

```

FORTRAN IV V01B-02
CORE=03K, UIC=[20,20]

FRI 18-MAR-77 19:20:00 PAGE 006
40001.OBJ=P40001.FOR/NOSN/LI:1

0307 NOREC=11
0308 ST(IND,1)=4002
0309 RETURN
0310 END

FORTRAN IV DIAGNOSTICS

[WARNING] MSG #092 VARIABLE "NOCHAR" NAME EXCEEDS 6 CHARACTERS
[WARNING] MSG #092 VARIABLE "KEYTYP" NAME EXCEEDS 6 CHARACTERS

FOR -- [P4000] ERRORS: 0, WARNINGS: 2
>

FOR HST.OBJ=HST.FOR/NOSN/LI:1

P4001

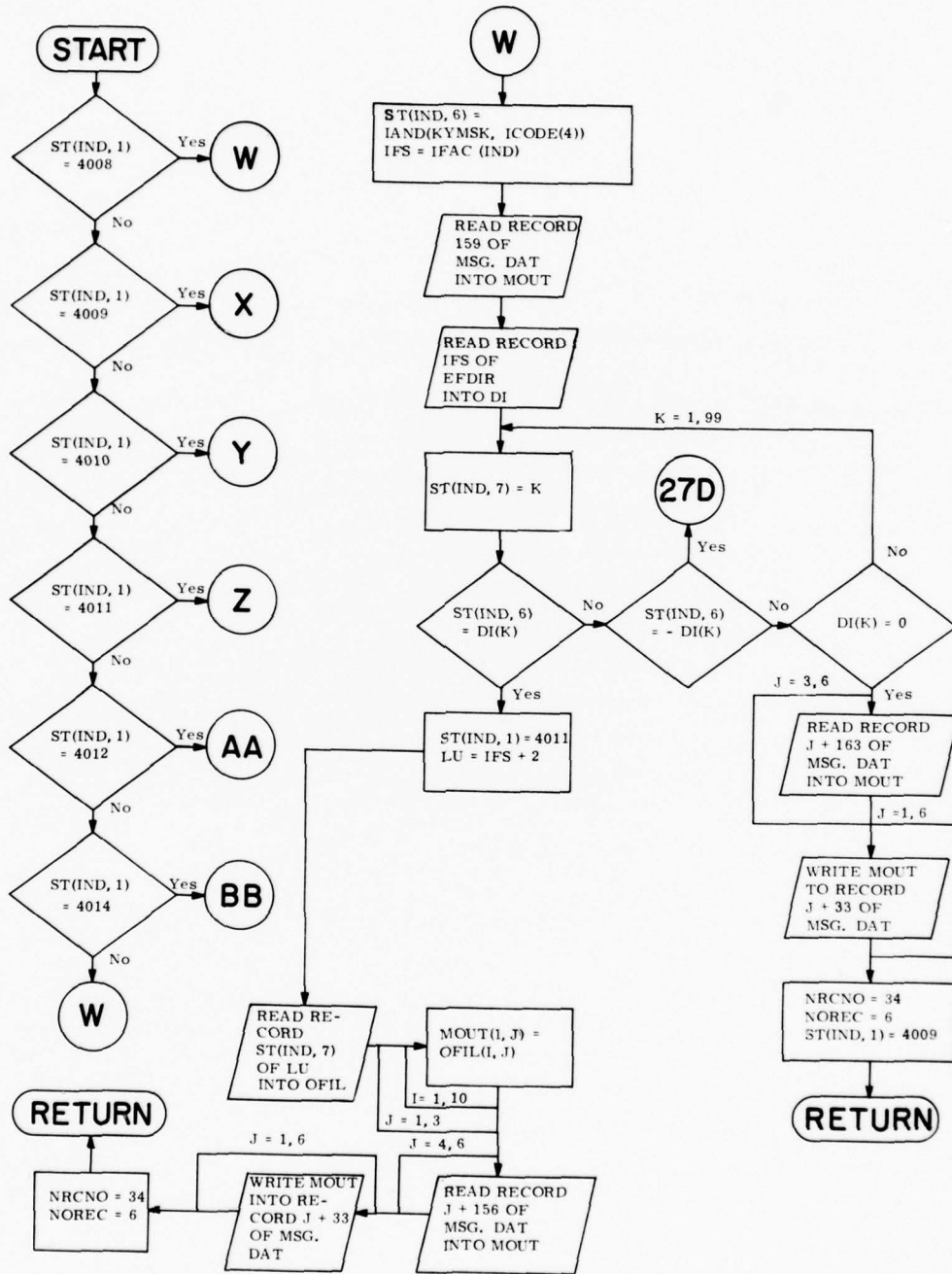


Figure 1-15. P4001

P4001(cont.)

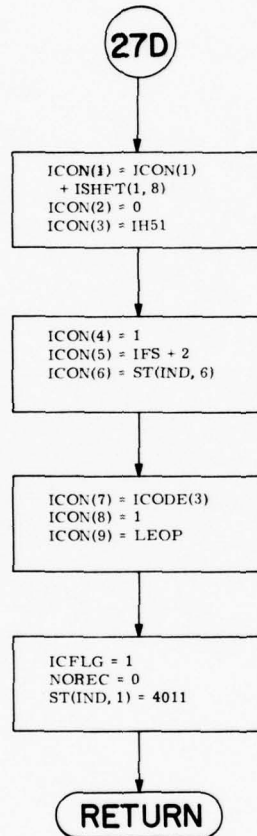


Figure 1-15. (Cont.)

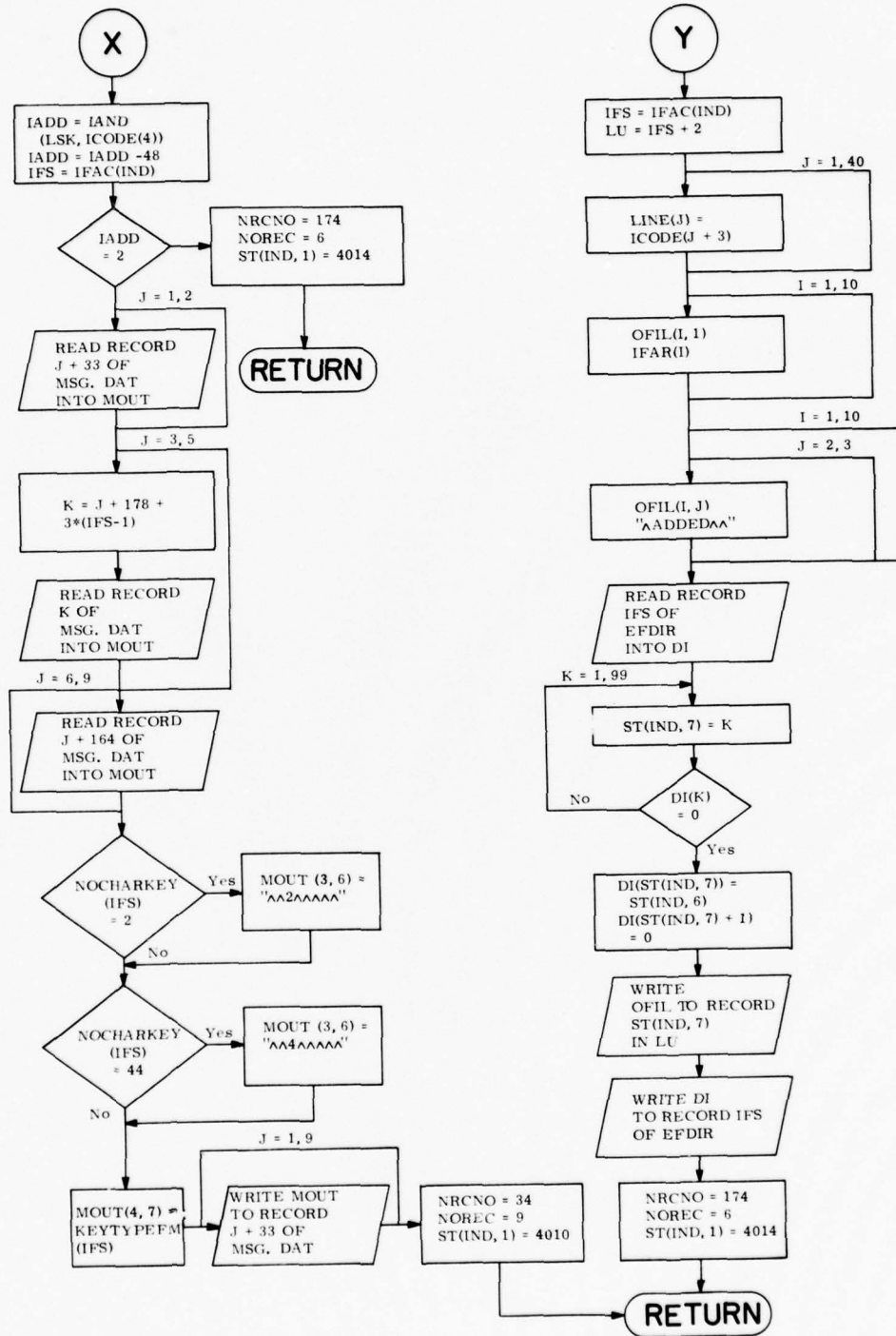


Figure 1-15. (Cont.)

P400I (cont.)

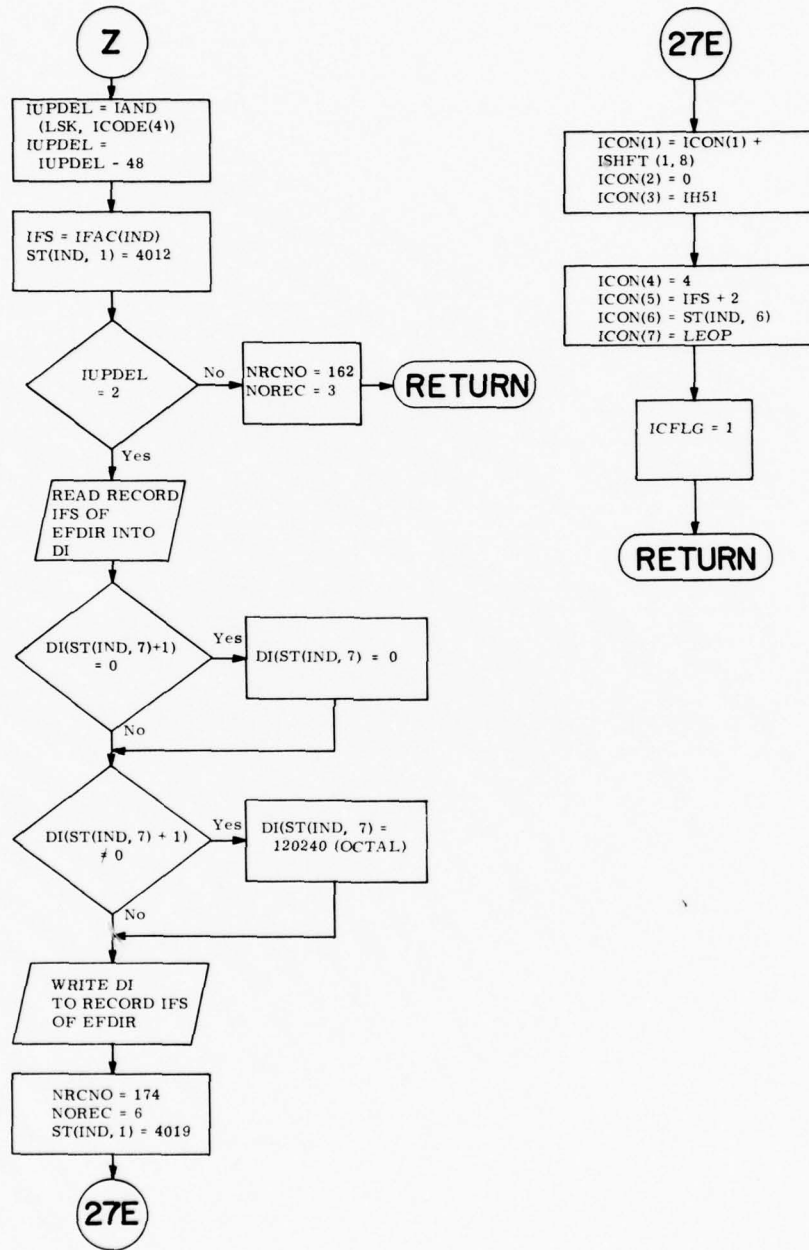


Figure 1-15. (Cont.)

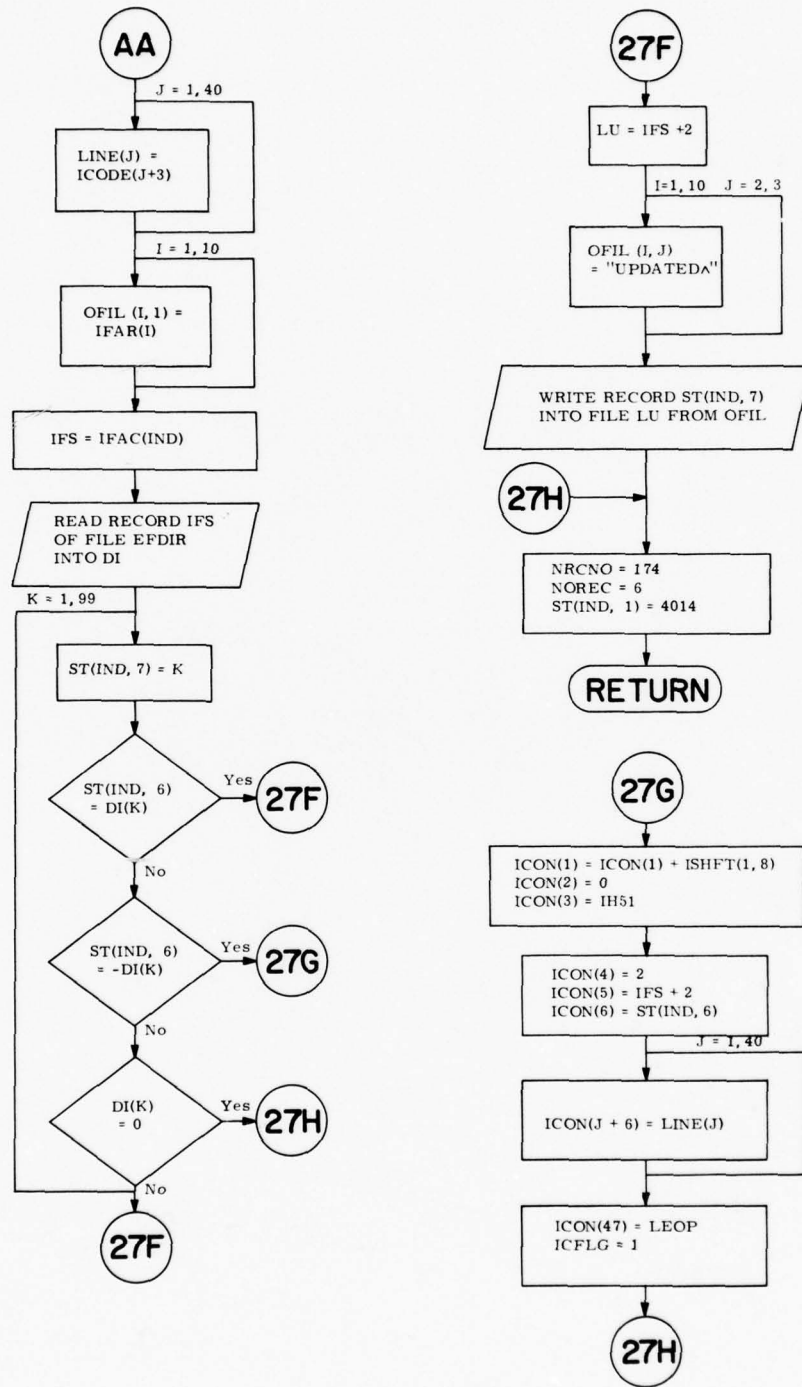


Figure 1-15. (Cont.)

P4001 (cont.)

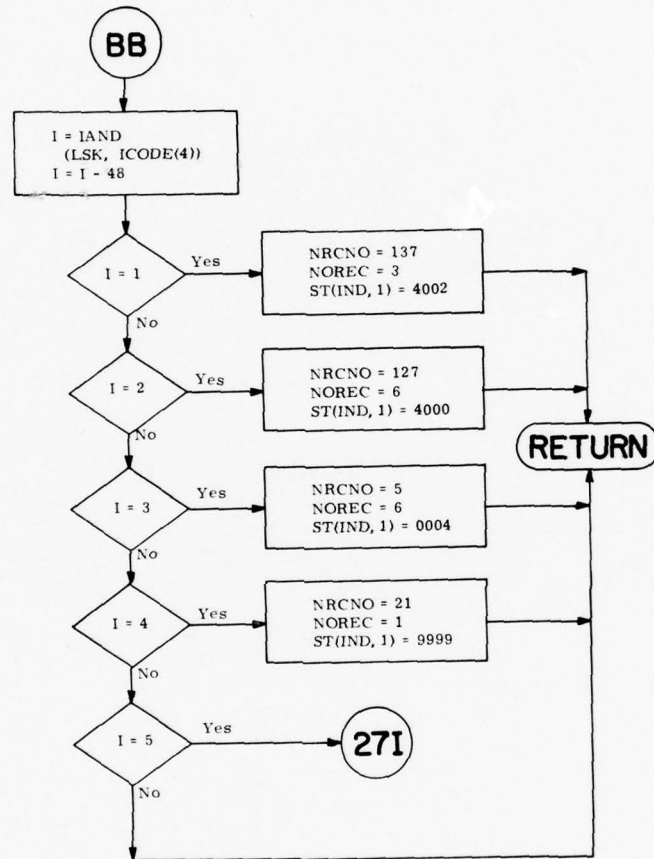


Figure 1-15. (Cont.)

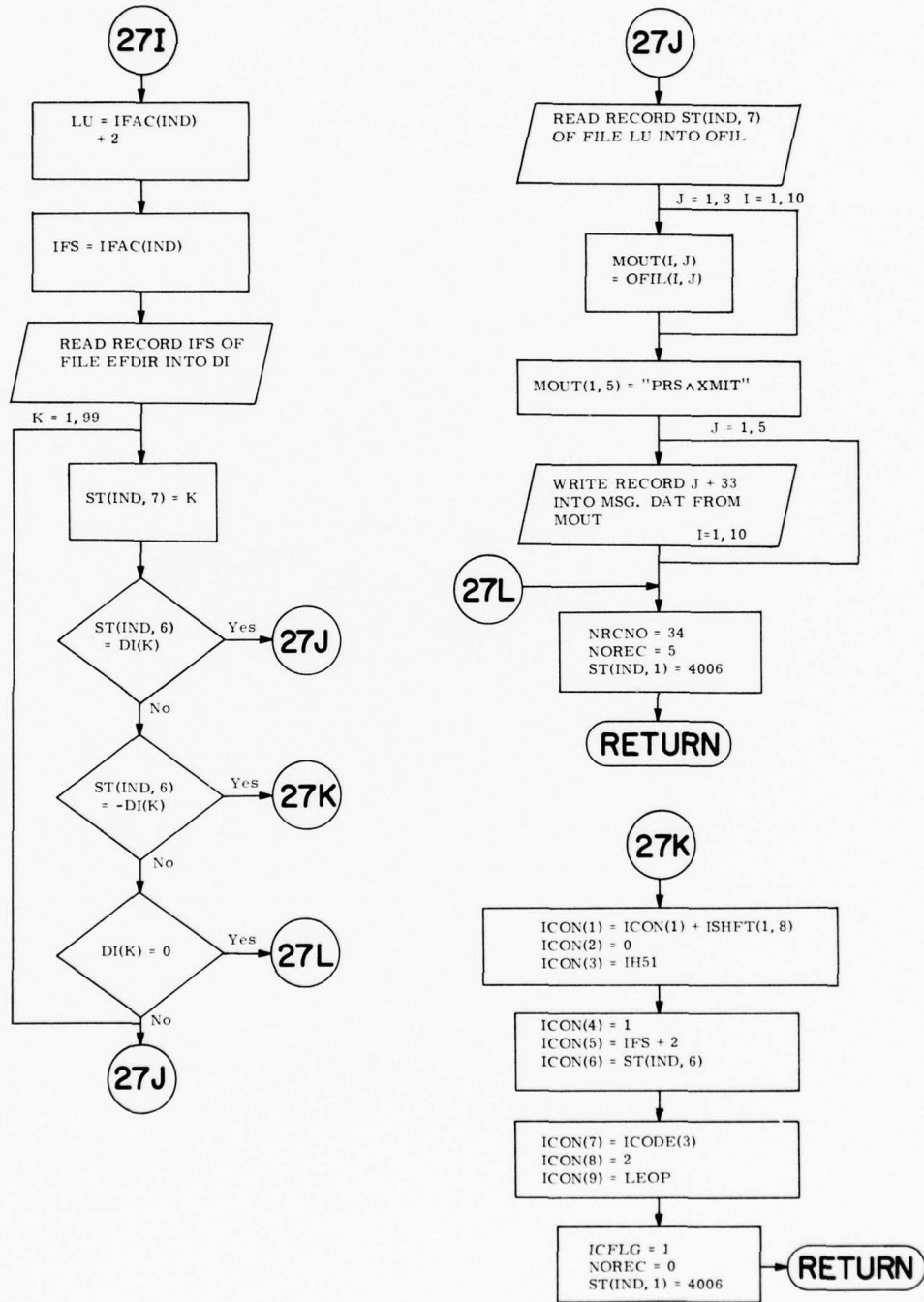


Figure 1-15. (Cont.)

FORTRAN IV V01B-02 FRI 18-MAR-77 19:15:58 PAGE 001
 CORE=0BK, UIC=E20,20J P4001.OBJ=P4001.FOR/NOSN/LI:1

```

0001      SUBROUTINE P4001
          C
          C
          C      FILE ACCESS MODE OF OPERATION
          C
0002      REAL*8 MOUT,Q2Q,Q4Q,KEYTYPEFM,XMT,OFIL,
          1      ST1,STD,ADE,UFD,NDI,LID,TAB,NWD
0003      INTEGER ST,DI
0004      DIMENSION OFIL(10,3),DI(100)
0005      COMMON NDI,LID,TAB,NWD,ST(3,9),IND
0006      COMMON /DSK/ I1,I2,I3,I4,I5,I6,I16
0007      COMMON /U000/ MOUT(10,11),NRCNO,NOREC
0008      COMMON /MD04/ NOCHARKEY(10),KEYTYPEFM(10)
0009      COMMON /F01/IFAC(3)
0010      COMMON /LOOP/ICODE(128),MSK,LSK
0011      COMMON /CPAC/ICDN(128),ICFLG
0012      REAL*8 IFAR
0013      DIMENSION LINE(40),IFAR(10)
0014      EQUIVALENCE(LINE,IFAR)
0015      DATA Q2Q,Q4Q/' 2      ',' 4      '//
0016      DATA XMT,ST1,STD/
          1'PRS XMIT','ST.      ','ST. DO Y'//
0017      DATA ADE/' ADDED '//
0018      DATA UFD/'UPDATED '//
0019      DATA ISPC/'120240/
0020      DATA IH51,LEOF,KYMSK/'2401','177777','77577/
0021      IF (ST(IND,1) .EQ. 4008) GOTO 28
0023      IF (ST(IND,1) .EQ. 4009) GOTO 29
0025      IF (ST(IND,1) .EQ. 4010) GOTO 30
0027      IF (ST(IND,1) .EQ. 4011) GOTO 31
0029      IF (ST(IND,1) .EQ. 4012) GOTO 32
0031      IF (ST(IND,1) .EQ. 4014) GOTO 34
0033      28 ST(IND,6)=IAND(KYMSK,ICODE(4))
0034      IFS=IFAC(IND)
          C      FOR RECORD MODIFICATION
          C      RECORD LOCK OCCURS HERE
0035      READ(8'159)(MOUT(I,1),I=1,10)
0036      READ(2'IFS,ERR=1415)(DI(I),I=1,100)
0037      DO 850 K=1,99
0038      ST(IND,7)=K
0039      IF (ST(IND,6) .EQ. DI(K)) GOTO 864
0041      IF (ST(IND,6) .EQ. -DI(K)) GOTO 500
0043      850 IF (DI(K) .EQ. 0) GOTO 860
          C      RECORD DOES NOT EXIST
0045      860 CONTINUE
0046      DO 886 J=3,6
0047      886 READ(8'J+163)(MOUT(I,J),I=1,10)
0048      DO 890 J=1,6
0049      890 WRITE(8'33+J)(MOUT(I,J),I=1,10)
0050      NRCNO=34
0051      NOREC=6
0052      ST(IND,1)=4009
0053      RETURN
          C      RECORD EXISTS
0054      864 ST(IND,1)=4011
0055      LU=IFS+2

```

FORTRAN IV V01B-02
CORE=08K, UIC=C20,20J

FRI 18-MAR-77 19:15:58

PAGE 002

P4001,OBJ=P4001.FOR/NOSN/LI:1

```
0056      READ(LU'ST(IND,7),ERR=1415)((OFIL(I,J),I=1,10),J=1,3)
0057      DO 883 J=1,3
0058      DO 883 I=1,10
0059      883 MOUT(I,J)=OFIL(I,J)
0060      DO 884 J=4,6
0061      884 READ(8'J+156)(MOUT(I,J),I=1,10)
0062      DO 8100 J=1,6
0063      8100 WRITE(8'J+33)(MOUT(I,J),I=1,10)
0064      NRCNO=34
0065      NOREC=6
0066      RETURN
0067      500 ICON(1)=ICON(1)+ISHFT(1,8)
0068      ICON(2)=0
0069      ICON(3)=IH51
0070      ICON(4)=1
0071      ICON(5)=IFS+2
0072      ICON(6)=ST(IND,6)
0073      ICON(7)=ICODE(3)
0074      ICON(8)=1
0075      ICON(9)=LEOF
0076      ICFLG=1
0077      NOREC=0
0078      ST(IND,1)=4011
0079      RETURN
0080      29 IADD=IAND(LSK,ICODE(4))
0081      IADD=IADD-48
0082      IFS=IFAC(IND)
0083      IF (IADD .EQ. 2) GOTO 961
C      ADD A RECORD
0085      DO 930 J=1,2
0086      930 READ(8'133+J)(MOUT(I,J),I=1,10)
0087      DO 935 J=3,5
0088      K=J+178+3*(IFS-1)
0089      935 READ(8'K)(MOUT(I,J),I=1,10)
0090      DO 940 J=6,9
0091      940 READ(8'J+164)(MOUT(I,J),I=1,10)
0092      IF (NOCHARKEY(IFS) .EQ. 2) MOUT(3,6)=Q2Q
0094      IF (NOCHARKEY(IFS) .EQ. 4) MOUT(3,6)=Q4Q
0096      MOUT(4,7)=KEYTYPEFM(IFS)
0097      DO 990 J=1,9
0098      990 WRITE(8'J+33)(MOUT(I,J),I=1,10)
0099      NRCNO=34
0100      NOREC=9
0101      ST(IND,1)=4010
0102      RETURN
C      DO NOT ADD A RECORD
0103      961 NRCNO=174
0104      NOREC=6
C      RECORD UNLOCK SHOULD OCCUR HERE
0105      ST(IND,1)=4014
0106      RETURN
0107      30 IFS=IFAC(IND)
0108      LU=IFS+2
0109      DO 300 J=1,40
0110      300 LINE(J)=ICODE(J+3)
0111      DO 310 I=1,10
```

FORTRAN IV V01B-02 FRI 18-MAR-77 19:15:58 PAGE 003
 CORE=08K, UIC=[20,20] P4001.OBJ=P4001.FOR/NOSN/LI:1

```

0112    310 OFIL(I,1)=IFAR(I)
      C    ADD A NEW RECORD
0113    DO 1050 I=1,10
0114    DO 1050 J=2,3
0115    1050 OFIL(I,J)=ADE
0116    READ(2'IFS,ERR=1415)(DI(I),I=1,100)
0117    DO 1055 K=1,99
0118    ST(IND,7)=K
0119    IF (DI(K) .EQ. 0) GOTO 1060
0121    1055 CONTINUE
0122    1060 DI(ST(IND,7))=ST(IND,6)
0123    DI(ST(IND,7)+1)=0
0124    WRITE(LU'ST(IND,7),ERR=1415)((OFIL(I,J),I=1,10),J=1,3)
0125    WRITE(2'IFS,ERR=1415)(DI(I),I=1,100)
0126    NRCNO=174
0127    NOREC=6
0128    ST(IND,1)=4014
0129    ICON(1)=ICON(1)+ISHFT(1,8)
0130    ICON(2)=0
0131    ICON(3)=IH51
0132    ICON(4)=3
0133    ICON(5)=IFS+2
0134    ICON(6)=ST(IND,6)
0135    ICON(7)=LEOF
0136    ICFLG=1
0137    RETURN
0138    31 IUPDEL=IAND(LSK,ICODE(4))
0139    IUPDEL=IUPDEL-48
0140    IFS=IFAC(IND)
0141    ST(IND,1)=4012
0142    IF (IUPDEL .EQ. 2) GOTO 1161
      C    UPDATE RECORD
0144    NRCNO=162
0145    NOREC=3
0146    RETURN
      C    DELETE RECORD
0147    1161 CONTINUE
0148    READ(2'IFS,ERR=1415)(DI(I),I=1,100)
0149    IF (DI(ST(IND,7)+1) .EQ. 0) DI(ST(IND,7))=0
0151    IF (DI(ST(IND,7)+1) .NE. 0) DI(ST(IND,7))=ISPC
0153    WRITE(2'IFS,ERR=1415)(DI(I),I=1,100)
0154    NRCNO=174
0155    NOREC=6
0156    ST(IND,1)=4014
0157    ICON(1)=ICON(1)+ISHFT(1,8)
0158    ICON(2)=0
0159    ICON(3)=IH51
0160    ICON(4)=4
0161    ICON(5)=IFS+2
0162    ICON(6)=ST(IND,6)
0163    ICON(7)=LEOF
0164    ICFLG=1
0165    RETURN
0166    32 DO 400 J=1,40
0167    400 LINE(J)=ICODE(J+3)
0168    DO 410 I=1,10

```

FORTRAN IV V01B-02
CORE=08K, UIC=C20,20J

FRI 18-MAR-77 19:15:58

PAGE 004

P4001.0BJ=P4001.FOR/NOSN/LI:1

```
0169 410 OFIL(I,1)=IFAR(I)
0170 IFS=IFAC(IND)
0171 READ(2,IFS,ERR=1415)(DI(I),I=1,100)
0172 DO 1500 K=1,99
0173 ST(IND,7)=K
0174 IF (ST(IND,6) .EQ. DI(K)) GOTO 1510
0176 IF (ST(IND,6) .EQ. -DI(K)) GOTO 510
0178 1500 IF (DI(K) .EQ. 0) GOTO 530
0180 1510 LU=IFS+2
0181 DO 1250 I=1,10
0182 DO 1250 J=2,3
0183 1250 OFIL(I,J)=UPD
0184 WRITE(LU,ST(IND,7),ERR=1415)((OFIL(I,J),I=1,10),J=1,3)
0185 530 NRCNO=174
0186 NOREC=6
0187 ST(IND,1)=4014
0188 RETURN
0189 510 ICON(1)=ICON(1)+ISHFT(1,8)
0190 ICON(2)=0
0191 ICON(3)=IH51
0192 ICON(4)=2
0193 ICON(5)=IFS+2
0194 ICON(6)=ST(IND,6)
0195 DO 520 J=1,40
0196 520 ICON(J+6)=LINE(J)
0197 ICON(47)=LEDP
0198 ICFLG=1
0199 GOTO 530
0200 34 I=IAND(LSK,ICODE(4))
0201 I=I-48
C MODE 4 SWITCH OUT
0202 IF (I .EQ. 1) GOTO 1470
0204 IF (I .EQ. 2) GOTO 144
0206 IF (I .EQ. 3) GOTO 146
0208 IF (I .EQ. 4) GOTO 148
0210 IF (I .EQ. 5) GOTO 149
0212 1415 NRCNO=11
0213 NOREC=1
0214 RETURN
0215 1470 NRCNO=137
0216 NOREC=3
0217 ST(IND,1)=4002
0218 RETURN
0219 144 NRCNO=127
0220 NOREC=6
0221 ST(IND,1)=4000
0222 RETURN
0223 146 NRCNO=5
0224 NOREC=6
0225 ST(IND,1)=0004
0226 RETURN
0227 148 NRCNO=21
0228 NOREC=1
0229 ST(IND,1)=9999
0230 RETURN
0231 149 LU=IFAC(IND)+2
```

FORTRAN IV V01B-02 FRI 18-MAR-77 19:15:58 PAGE 005
CORE=08K, UIC=[20,20] P4001.OBJ=P4001.FOR/NOSN/LI:1

```
0232        IFS=IFAC(IND)
0233        READ(2'IFS,ERR=1415)(DI(I),I=1,100)
0234        DO 1520 K=1,99
0235        ST(IND,7)=K
0236        IF (ST(IND,6) .EQ. DI(K)) GOTO 1530
0238        IF (ST(IND,6) .EQ. -DI(K)) GOTO 540
0240        1520 IF (DI(K) .EQ. 0) GOTO 550
0242        1530 READ(LU'ST(IND,7),ERR=1415)((OFIL(I,J),I=1,10),J=1,3)
0243        DO 150 J=1,3
0244        DO 150 I=1,10
0245        150 MOUT(I,J)=OFIL(I,J)
0246        MOUT(1,5)=XMT
0247        DO 151 J=1,5
0248        151 WRITE(8'J+33)(MOUT(I,J),I=1,10)
0249        550 NRCNO=34
0250        NOREC=5
0251        ST(IND,1)=4006
0252        RETURN
0253        540 ICON(1)=ICON(1)+ISHFT(1,8)
0254        ICON(2)=0
0255        ICON(3)=IH51
0256        ICON(4)=1
0257        ICON(5)=IFS+2
0258        ICON(6)=ST(IND,6)
0259        ICON(7)=ICDRE(3)
0260        ICON(8)=2
0261        ICON(9)=LEOF
0262        ICFLG=1
0263        NOREC=0
0264        ST(IND,1)=4006
0265        RETURN
0266        END
```

FORTRAN IV DIAGNOSTICS

[WARNING] MSG #092 VARIABLE "NOCHAR" NAME EXCEEDS 6 CHARACTERS
[WARNING] MSG #092 VARIABLE "KEYTYP" NAME EXCEEDS 6 CHARACTERS

FOR -- [P4001] ERRORS: 0, WARNINGS: 2-
>

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:22:21

PAGE 001

P40011.OBJ=P40011.FOR/NOSN/LI:1

```

0001      SUBROUTINE P4001
          C
          C
          C      FILE ACCESS MODE OF OPERATION
          C
0002      REAL*8 MOUT,Q2Q,Q4Q,KEYTYPEFM,XMT,OFIL,
          1      ST1,STD,ADE,UPD,NDI,LID,TAB,NWD
0003      INTEGER ST,DI
0004      DIMENSION OFIL(10,3),DI(100)
0005      COMMON NDI,LID,TAB,NWD,ST(3,9),IND
0006      COMMON /DSK/ I1,I2,I3,I4,I5,I6,I16
0007      COMMON /U000/ MOUT(10,11),NRCNO,NOREC
0008      COMMON /MD04/ NOCHARKEY(10),KEYTYPEFM(10)
0009      COMMON /FO1/IFAC(3)
0010      COMMON /LOOF/ICODE(128),MSK,LSK
0011      COMMON /CFAC/ICON(128),ICFLG
0012      REAL*8 IFAR
0013      DIMENSION LINE(40),IFAR(10)
0014      EQUIVALENCE(LINE,IFAR)
0015      DATA Q2Q,Q4Q/' 2      ',' 4      '/
0016      DATA XMT,ST1,STD/
          1'PRS XMIT','ST.      ','ST. DO Y'/
0017      DATA ADE/' ADDED  '//
0018      DATA UPD/'UPDATED  '//
0019      DATA ISPC/'120240/
0020      DATA IH51,LEOP,KYMSK/'405','177777','77577/
0021      IF (ST(IND,1) .EQ. 4008) GOTO 28
0023      IF (ST(IND,1) .EQ. 4009) GOTO 29
0025      IF (ST(IND,1) .EQ. 4010) GOTO 30
0027      IF (ST(IND,1) .EQ. 4011) GOTO 31
0029      IF (ST(IND,1) .EQ. 4012) GOTO 32
0031      IF (ST(IND,1) .EQ. 4014) GOTO 34
0033      28 ST(IND,6)=IAND(KYMSK,ICODE(4))
0034      IFS=IFAC(IND)
          C      FOR RECORD MODIFICATION.
          C      RECORD LOCK OCCURS HERE
0035      READ(8'159)(MOUT(I,1),I=1,10)
0036      READ(2'IFS,ERR=1415)(DI(I),I=1,100)
0037      DO 850 K=1,99
0038      ST(IND,7)=K
0039      IF (ST(IND,6) .EQ. DI(K)) GOTO 864
0041      IF (ST(IND,6) .EQ. -DI(K)) GOTO 500
0043      850 IF (DI(K) .EQ. 0) GOTO 860
          C      RECORD DOES NOT EXIST
0045      860 CONTINUE
0046      DO 886 J=3,6
0047      886 READ(8'J+163)(MOUT(I,J),I=1,10)
0048      DO 890 J=1,6
0049      890 WRITE(8'33+J)(MOUT(I,J),I=1,10)
0050      NRCNO=34
0051      NOREC=6
0052      ST(IND,1)=4009
0053      RETURN
          C      RECORD EXISTS
0054      864 ST(IND,1)=4011
0055      LU=IFS+2

```

FORTRAN IV V01B-02
 CORE=08K, UID=C20,20J

FRI 18-MAR-77 19:22:21

PAGE 002

P40011.OBJ=P40011.FOR/NOSN/LI:1

```

0056      READ(LU'ST(IND,7),ERR=1415)((OFIL(I,J),I=1,10),J=1,3)
0057      DO 883 J=1,3
0058      DO 883 I=1,10
0059      883 MOUT(I,J)=OFIL(I,J)
0060      DO 884 J=4,6
0061      884 READ(8'J+156)(MOUT(I,J),I=1,10)
0062      DO 8100 J=1,6
0063      8100 WRITE(8'J+33)(MOUT(I,J),I=1,10)
0064      NRCNO=34
0065      NOREC=6
0066      RETURN
0067      500 ICON(1)=ICON(1)+ISHFT(1,8)
0068      ICON(2)=0
0069      ICON(3)=IHS1
0070      ICON(4)=1
0071      ICON(5)=IFS+2
0072      ICON(6)=ST(IND,6)
0073      ICON(7)=ICODE(3)
0074      ICON(8)=1
0075      ICON(9)=LEQP
0076      ICFLG=1
0077      NOREC=0
0078      ST(IND,1)=4011
0079      RETURN
0080      29 IADD=IAND(LSK,ICODE(4))
0081      IADD=IADD-48
0082      IFS=IFAC(IND)
0083      IF (IADD .EQ. 2) GOTO 961
C      ADD A RECORD
0085      DO 930 J=1,2
0086      930 READ(8'133+J)(MOUT(I,J),I=1,10)
0087      DO 935 J=3,5
0088      K=J+178+3*(IFS-1)
0089      935 READ(8'K)(MOUT(I,J),I=1,10)
0090      DO 940 J=6,9
0091      940 READ(8'J+164)(MOUT(I,J),I=1,10)
0092      IF (NOCHARKEY(IFS) .EQ. 2) MOUT(3,6)=Q2Q
0094      IF (NOCHARKEY(IFS) .EQ. 4) MOUT(3,6)=Q4Q
0096      MOUT(4,7)=KEYTYPEFM(IFS)
0097      DO 990 J=1,9
0098      990 WRITE(8'J+33)(MOUT(I,J),I=1,10)
0099      NRCNO=34
0100      NOREC=9
0101      ST(IND,1)=4010
0102      RETURN
C      DO NOT ADD A RECORD
0103      961 NRCNO=174
0104      NOREC=6
C      RECORD UNLOCK SHOULD OCCUR HERE
0105      ST(IND,1)=4014
0106      RETURN
0107      30 IFS=IFAC(IND)
0108      LU=IFS+2
0109      DO 300 J=1,40
0110      300 LINE(J)=ICODE(J+3)
0111      DO 310 I=1,10

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:22:21

PAGE 003

P40011.OBJ=P40011.FOR/N06N/LI:1

```
0112    310 OFIL(I,1)=IFAR(I)
C        ADD A NEW RECORD
0113        DO 1050 I=1,10
0114        DO 1050 J=2,3
0115    1050 OFIL(I,J)=ADE
0116        READ(2'IFS,ERR=1415)(DI(I),I=1,100)
0117        DO 1055 K=1,99
0118        ST(IND,7)=K
0119        IF (DI(K) .EQ. 0) GOTO 1060
0121    1055 CONTINUE
0122    1060 DI(ST(IND,7))=ST(IND,6)
0123        DI(ST(IND,7)+1)=0
0124        WRITE(LU'ST(IND,7),ERR=1415)((OFIL(I,J),I=1,10),J=1,3)
0125        WRITE(2'IFS,ERR=1415)(DI(I),I=1,100)
0126        NRCNO=174
0127        NOREC=6
0128        ST(IND,1)=4014
0129        ICON(1)=ICON(1)+ISHFT(1,8)
0130        ICON(2)=0
0131        ICON(3)=IH51
0132        ICON(4)=3
0133        ICON(5)=IFS+2
0134        ICON(6)=ST(IND,6)
0135        ICON(7)=LEOP
0136        ICFLG=1
0137        RETURN
0138    31 IUPDEL=IAND(LSK,ICODE(4))
0139        IUPDEL=IUPDEL-48
0140        IFS=IFAC(IND)
0141        ST(IND,1)=4012
0142        IF (IUPDEL .EQ. 2) GOTO 1161
C
0144        UPDATE RECDR
0145        NRCNO=162
0146        NOREC=3
0146        RETURN
C
0147    1161 DELETE RECORD
0147    1161 CONTINUE
0148        READ(2'IFS,ERR=1415)(DI(I),I=1,100)
0149        IF (DI(ST(IND,7)+1) .EQ. 0) DI(ST(IND,7))=0
0151        IF (DI(ST(IND,7)+1) .NE. 0) DI(ST(IND,7))=ISPC
0153        WRITE(2'IFS,ERR=1415)(DI(I),I=1,100)
0154        NRCNO=174
0155        NOREC=6
0156        ST(IND,1)=4014
0157        ICON(1)=ICON(1)+ISHFT(1,8)
0158        ICON(2)=0
0159        ICON(3)=IH51
0160        ICON(4)=4
0161        ICON(5)=IFS+2
0162        ICON(6)=ST(IND,6)
0163        ICON(7)=LEOP
0164        ICFLG=1
0165        RETURN
0166    32 DO 400 J=1,40
0167    400 LINE(J)=ICODE(J+3)
0168        DO 410 I=1,10
```

FORTRAN IV V01B-02
 CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:22:21
 P40011.OBJ=P40011.FOR/NOSN/LI:1

PAGE 004

```

0169   410 OFIL(I,1)=IFAR(I)
0170     IFS=IFAC(IND)
0171     READ(2'IFS,ERR=1415)(DI(I),I=1,100)
0172     DO 1500 K=1,99
0173       ST(IND,7)=K
0174       IF (ST(IND,6) .EQ. DI(K)) GOTO 1510
0176       IF (ST(IND,6) .EQ. -DI(K)) GOTO 510
0178   1500 IF (DI(K) .EQ. 0) GOTO 530
0180   1510 LU=IFS+2
0181     DO 1250 I=1,10
0182     DO 1250 J=2,3
0183   1250 OFIL(I,J)=UPD
0184     WRITE(LU'ST(IND,7),ERR=1415)((OFIL(I,J),I=1,10),J=1,3)
0185   530  NRCNO=174
0186     NOREC=6
0187     ST(IND,1)=4014
0188     RETURN
0189   510  ICON(1)=ICON(1)+ISHFT(1,8)
0190     ICON(2)=0
0191     ICON(3)=IH51
0192     ICON(4)=2
0193     ICON(5)=IFS+2
0194     ICON(6)=ST(IND,6)
0195     DO 520 J=1,40
0196   520  ICON(J+6)=LINE(J)
0197     ICON(47)=LEDP
0198     ICFLG=1
0199     GOTO 530
0200   34  I=IAND(LSK,ICODE(4))
0201     I=I-4B
C
0202     MODE 4 SWITCH OUT
0204     IF (I .EQ. 1) GOTO 1470
0206     IF (I .EQ. 2) GOTO 144
0208     IF (I .EQ. 3) GOTO 146
0210     IF (I .EQ. 4) GOTO 148
0212     IF (I .EQ. 5) GOTO 149
0212   1415 NRCNO=11
0213     NOREC=1
0214     RETURN
0215   1470 NRCNO=137
0216     NOREC=3
0217     ST(IND,1)=4002
0218     RETURN
0219   144  NRCNO=127
0220     NOREC=6
0221     ST(IND,1)=4000
0222     RETURN
0223   146  NRCNO=5
0224     NOREC=6
0225     ST(IND,1)=0004
0226     RETURN
0227   148  NRCNO=21
0228     NOREC=1
0229     ST(IND,1)=9999
0230     RETURN
0231   149  LU=IFAC(IND)+2

```

FORTRAN IV V01B-02
CORE=08K, UIC=C20,20J

FRI 18-MAR-77 19:22:21

PAGE 005
P40011.OBJ=P40011.FDR/NOSN/LI:1

```
0232     IFS=IFAC(IND)
0233     READ(2'IFS,ERR=1415)(DI(I),I=1,100)
0234     DO 1520 K=1,99
0235     ST(IND,7)=K
0236     IF (ST(IND,6) .EQ. DI(K)) GOTO 1530
0238     IF (ST(IND,6) .EQ. -DI(K)) GOTO 540
0240 1520 IF (DI(K) .EQ. 0) GOTO 550
0242 1530 READ(LU'ST(IND,7),ERR=1415)((OFIL(I,J),I=1,10),J=1,3)
0243     DO 150 J=1,3
0244     DO 150 I=1,10
0245 150 MOUT(I,J)=OFIL(I,J)
0246     MOUT(1,5)=XMT
0247     DO 151 J=1,5
0248 151 WRITE(8'J+33)(MOUT(I,J),I=1,10)
0249 550 NRCND=34
0250     NOREC=5
0251     ST(IND,1)=4006
0252     RETURN
0253 540 ICON(1)=ICON(1)+ISHFT(1,8)
0254     ICON(2)=0
0255     ICON(3)=IH51
0256     ICON(4)=1
0257     ICON(5)=IFS+2
0258     ICON(6)=ST(IND,6)
0259     ICON(7)=ICODE(3)
0260     ICON(8)=2
0261     ICON(9)=LEDP
0262     ICFLG=1
0263     NOREC=0
0264     ST(IND,1)=4006
0265     RETURN
0266     END
```

FORTRAN IV DIAGNOSTICS

```
[ WARNING ] MSG #092    VARIABLE "NOCHAR" NAME EXCEEDS 6 CHARACTERS
[ WARNING ] MSG #092    VARIABLE "KEYTYP" NAME EXCEEDS 6 CHARACTERS
```

```
FOR -- [P4001 ] ERRORS: 0, WARNINGS: 2
>
```

P5000

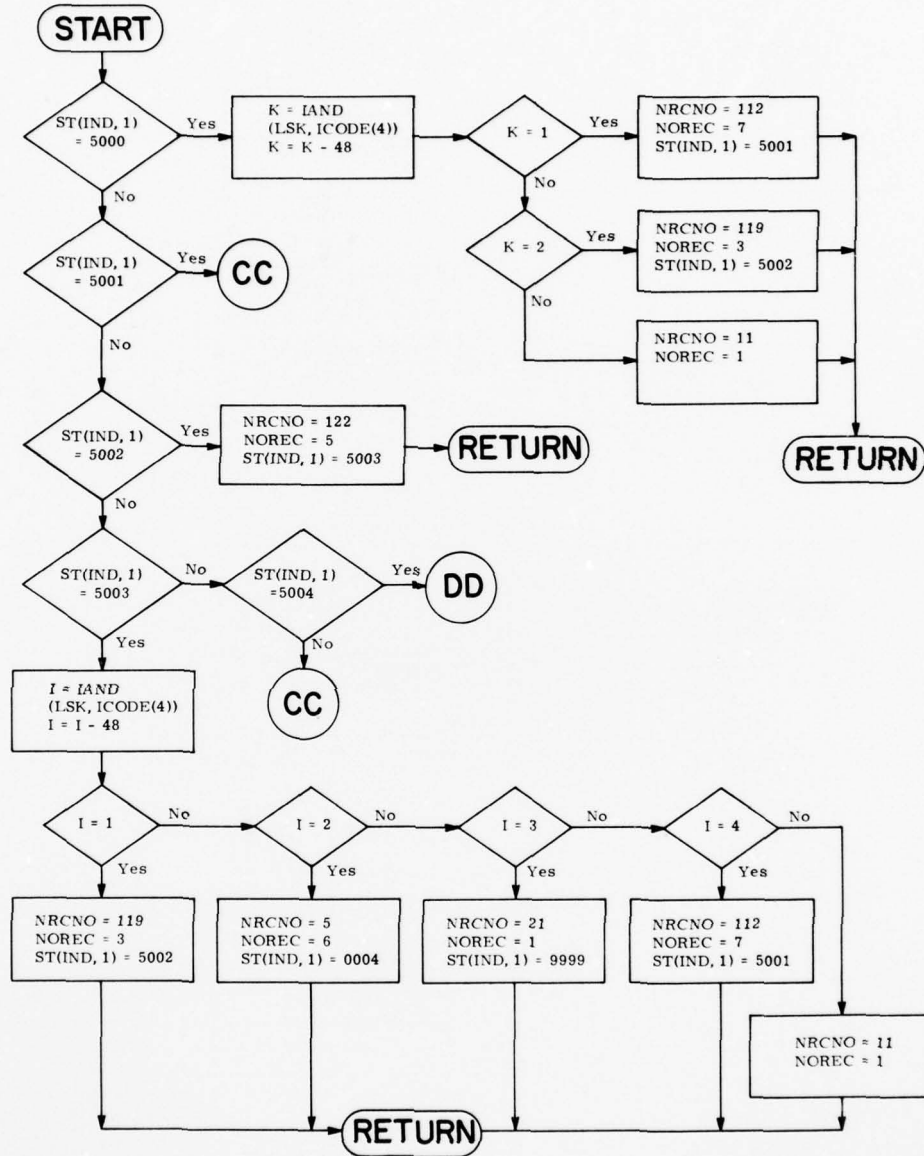


Figure 1-16. P5000

P5000(cont.)

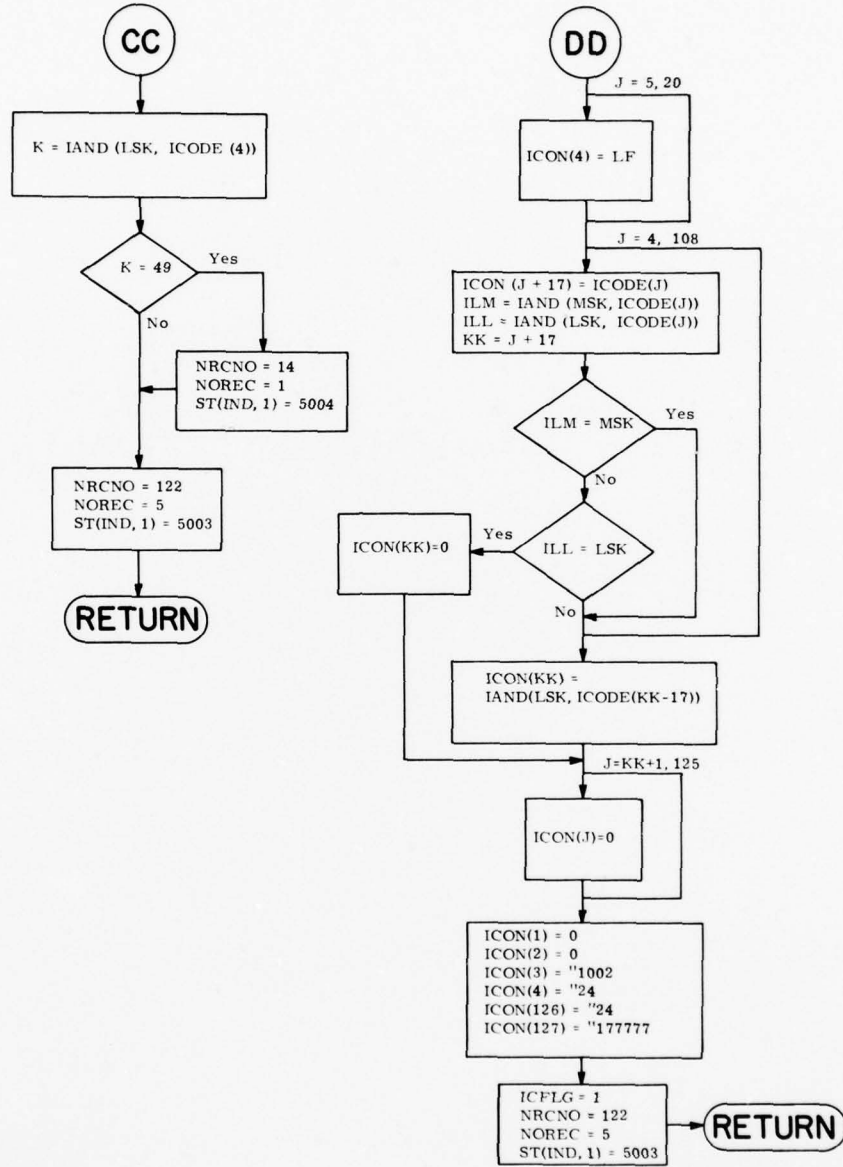


Figure 1-16. (Cont.)

FORTRAN IV VO1R-02 TUE 01-MAR-77 15:55:40 PAGE 001
 CORE=0BK, UIC=[20,20] P5000.OBJ=P5000.FOR/NOSN/LI:1

```

0001            SUBROUTINE P5000
          C
0002            REAL*8 MOUT,NDI,LID,TAB,NWD
0003            INTEGER ST
0004            COMMON NDI,LID,TAB,NWD,ST(3,9),IND
0005            COMMON /U000/ MOUT(10,11),NRCNO,NOREC
0006            COMMON /CFAC/ICON(128),ICFLG
0007            DATA LF,LHOME,LEOP,IBRCST,IHB,'12','24','177777','1002','2577/
0008            COMMON /LOOP/ICODE(128),MSK,LSK
          C
          C            CARD FORMAT MODE OF OPERATION
          C
0009            IF(ST(IND,1) .EQ. 5000)GOTO 85
0011            IF (ST(IND,1) .EQ. 5001) GOTO 21
0013            IF (ST(IND,1) .EQ. 5002) GOTO 22
0015            IF (ST(IND,1) .EQ. 5003)GOTO 23
0017            IF (ST(IND,1) .EQ. 5004) GOTO 24
0019            21 K=IAND(LSK,ICODE(4))
0020            IF (K .EQ. 49) GOTO 100
0022            NRCNO=122
0023            NOREC=5
0024            ST(IND,1)=5003
0025            RETURN
0026            100 NRCNO=14
0027            NOREC=1
0028            ST(IND,1)=5004
0029            RETURN
0030            24 DO 110 J=5,20
0031            110 ICON(J)=LF
0032            DO 120 J=4,108
0033            ICON(J+17)=ICODE(J)
0034            ILM=IAND(MSK,ICODE(J))
0035            ILL=IAND(LSK,ICODE(J))
0036            KK=J+17
0037            IF (ILM .EQ. MSK) GOTO 130
0039            120 IF (ILL .EQ. LSK) GOTO 140
0041            130 ICON(KK)=IAND(LSK,ICODE(KK-17))
0042            GOTO 150
0043            140 ICON(KK)=0
0044            150 DO 160 J=KK+1,125
0045            160 ICON(J)=0
0046            ICON(1)=0
0047            ICON(2)=IBRCST
0048            ICON(3)=IHB
0049            ICON(4)=LHOME
0050            ICON(126)=LHOME
0051            ICON(127)=LEOP
0052            ICFLG=1
0053            NRCNO=122
0054            NOREC=5
0055            ST(IND,1)=5003
0056            RETURN
0057            22 NRCNO=122
0058            NOREC=5
0059            ST(IND,1)=5003
0060            RETURN

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

TUE 01-MAR-77 15:55:40

PAGE 002

P5000.OBJ=P5000.FOR/NOSN/LI:1

```
0061      23 I=IAND(LSK,ICODE(4))
0062          I=I-48
0063          IF (I .EQ. 1) GOTO 830
0065          IF (I .EQ. 2) GOTO 6
0067          IF (I .EQ. 3) GOTO 8
0069          IF (I .EQ. 4) GOTO 820
0071          NRCNO=11
0072          NOREC=1
0073          RETURN
0074      830 NRCNO=119
0075          NOREC=3
0076          ST(IND,1)=5002
0077          RETURN
0078      6 NRCNO=5
0079          NOREC=6
0080          ST(IND,1)=0004
0081          RETURN
0082      8 NRCNO=21
0083          NOREC=1
0084          ST(IND,1)=9999
0085          RETURN
0086      820 NRCNO=112
0087          NOREC=7
0088          ST(IND,1)=5001
0089          RETURN
0090      85 K=IAND(LSK,ICODE(4))
0091          K=K-48
0092          IF (K .EQ. 1) GOTO 82
0094          IF (K .EQ. 2) GOTO 83
0096      20 NRCNO=11
0097          NOREC=1
0098          RETURN
0099      82 NRCNO=112
0100          NOREC=7
0101          ST(IND,1)=5001
0102          RETURN
0103      83 NRCNO=119
0104          NOREC=3
0105          ST(IND,1)=5002
0106          RETURN
0107          END
```

FORTRAN IV DIAGNOSTICS

[WARNING] MSG #094 NON-STANDARD STATEMENT ORDERING

FOR -- [P5000] - ERRORS: 0, WARNINGS: 1

>

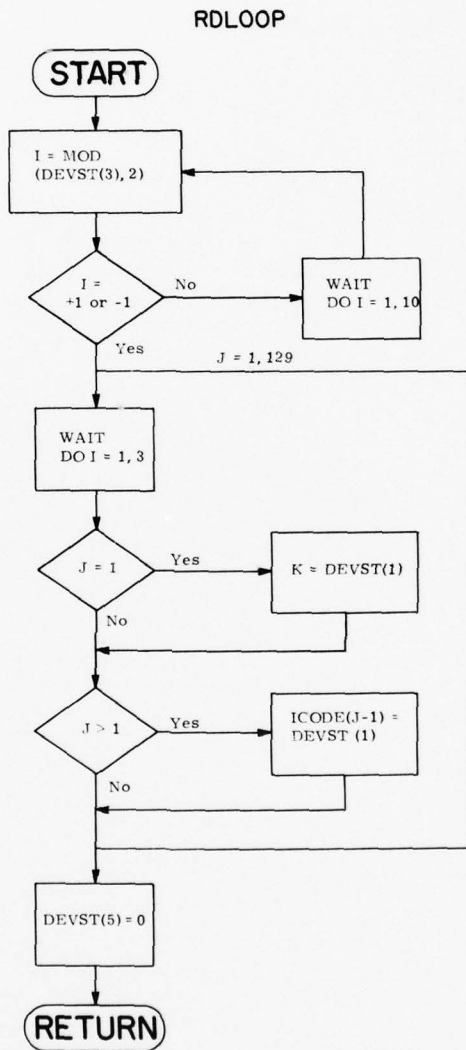


Figure 1-17. RDLOOP

AD-A063 407

BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 17/2
EXPLORATORY SYSTEMS CONTROL MODEL (ESM). BOOK 1. FORTRAN. SOFTW--ETC(U)
APR 77 DCA100-75-C-0054
NL

UNCLASSIFIED

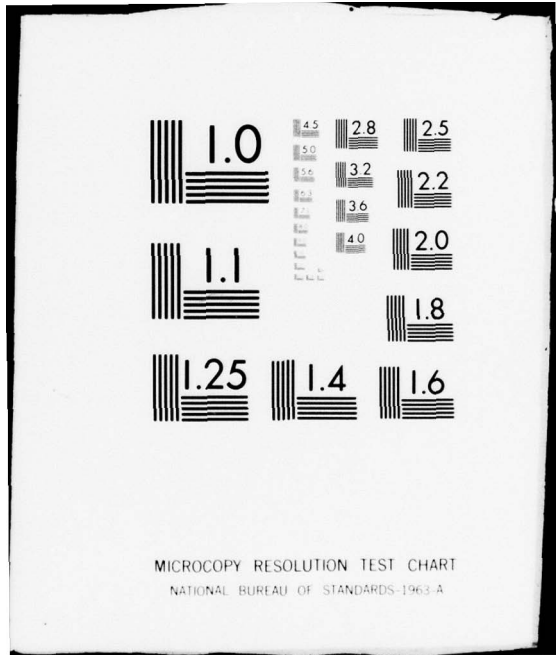
66143-3

3 OF 3

AD A063407



END
DATE
FILMED
3-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963 A

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

TUE 01-MAR-77 16:16:24

PAGE 001

RDLOOP.OBJ=RDLOOP.FOR/NOSN/LI:1

```
0001       SUBROUTINE RDLOOP
0002       INTEGER DEVST(6)
0003       COMMON /M1710/DEVST
0004       COMMON /LOOP/ ICODE(128),MSK,LSK
0005       90 I=MOD(DEVST(3),2)
0006       IF (I .EQ. 1) GOTO 120
0008       IF (I .EQ. -1) GOTO 120
0010       DO 110 I=1,10
0011       110 CONTINUE
0012       GOTO 90
0013       120 DO 130 J=1,129
0014       DO 140 I=1,3
0015       140 CONTINUE
0016       IF (J .EQ. 1) K=DEVST(1)
0018       IF (J .GT. 1) ICODE(J-1)=DEVST(1)
0020       130 CONTINUE
0021       DEVST(5)=0
0022       RETURN
0023       END
```

>

WRLOOP
(I1ST)

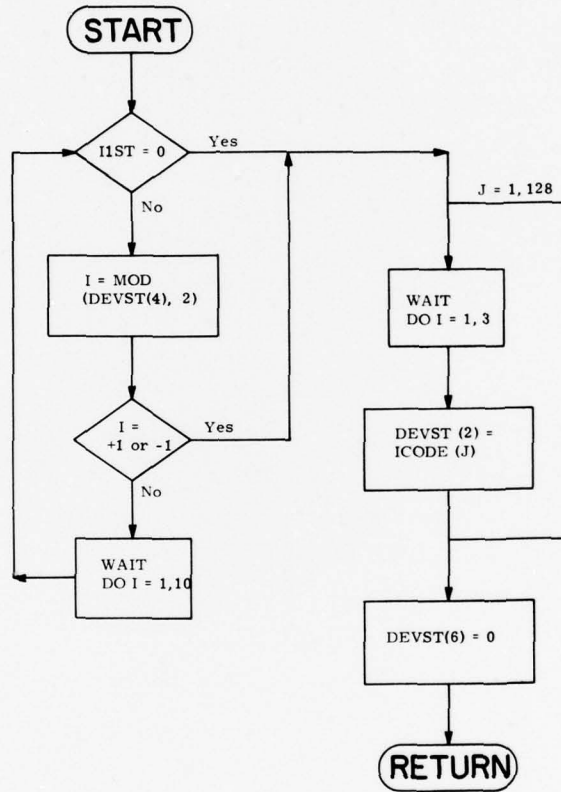


Figure 1-18. WRLOOP

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

TUE 01-MAR-77 16:17:24

PAGE 001

WRLOOP.OBJ=WRLOOP.FOR/NOSN/LI:1

```
0001     SUBROUTINE WRLOOP(I1ST)
0002     INTEGER DEVST(6)
0003     COMMON /M1710/ DEVST
0004     COMMON /LOOP/ICODE(128),MSK,LSK
0005     IF (I1ST .EQ. 0) GOTO 40
0007     50 I=MOD(DEVST(4),2)
0008     IF (I .EQ. 1) GOTO 40
0010     IF (I .EQ. -1) GOTO 40
0012     DO 60 I=1,10
0013     60 CONTINUE
0014     GOTO 50
0015     40 DO 70 J=1,128
0016     DO 80 I=1,3
0017     80 CONTINUE
0018     70 DEVST(2)=ICODE(J)
0019     DEVST(6)=0
0020     RETURN
0021     END
```

> 31

HST

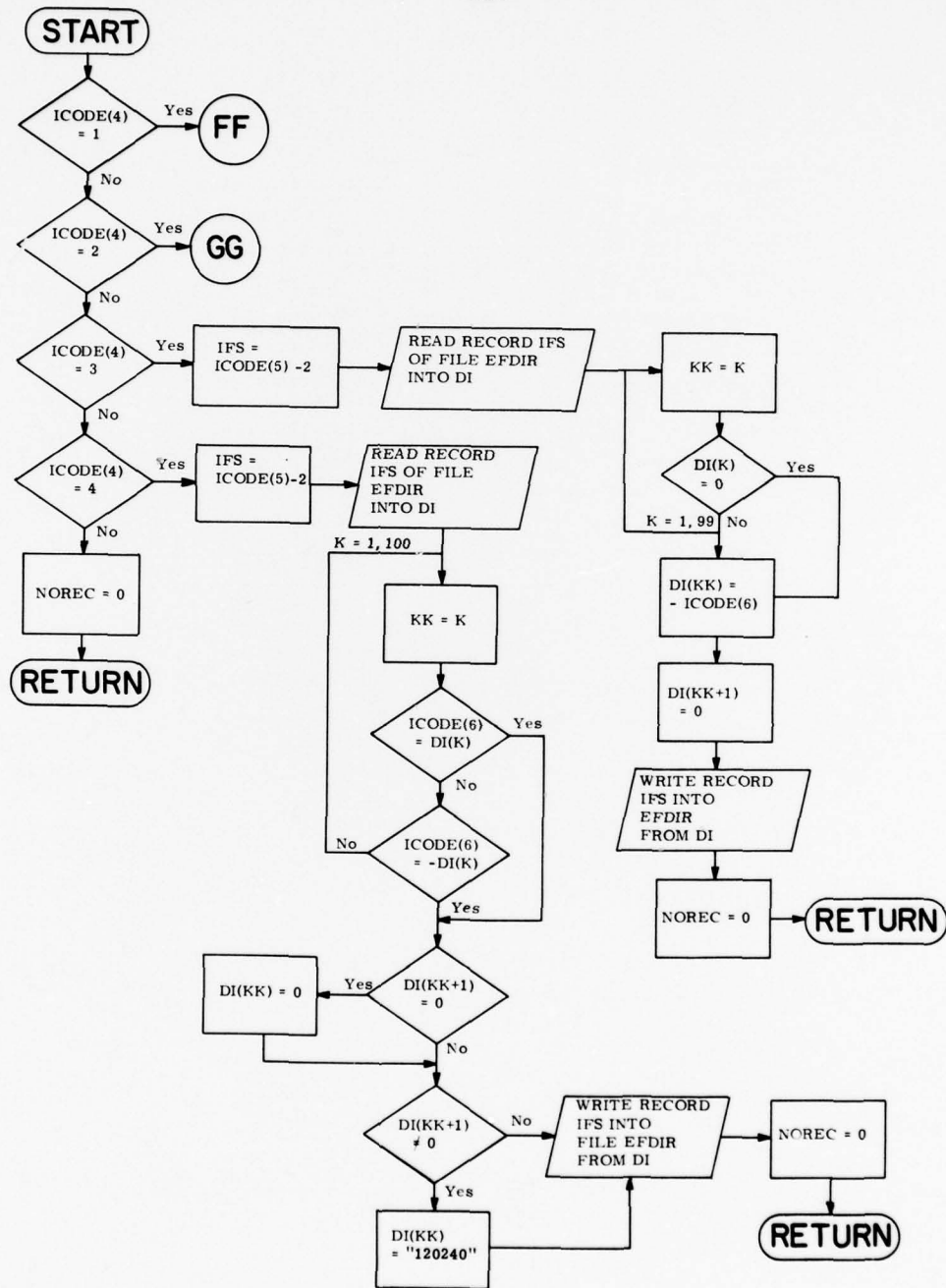


Figure 1-19. HST

HST (cont.)

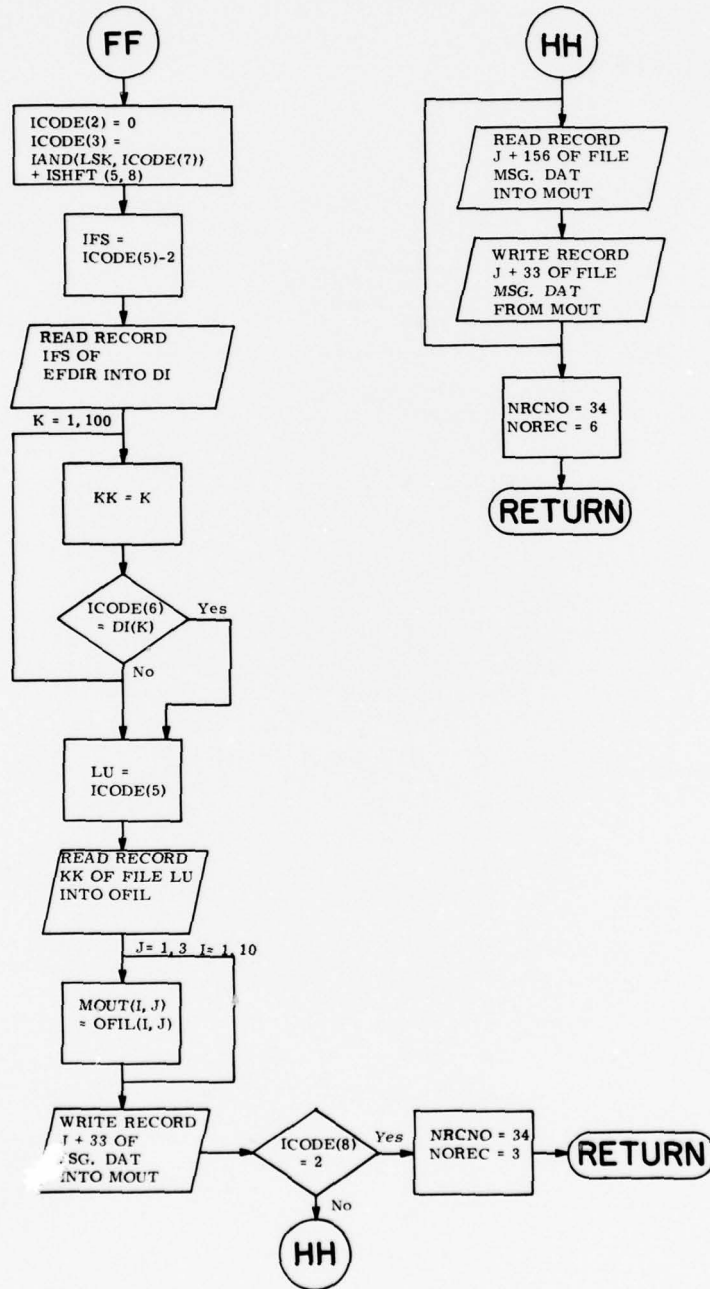


Figure 1-19. (Cont.)

HST (cont.)

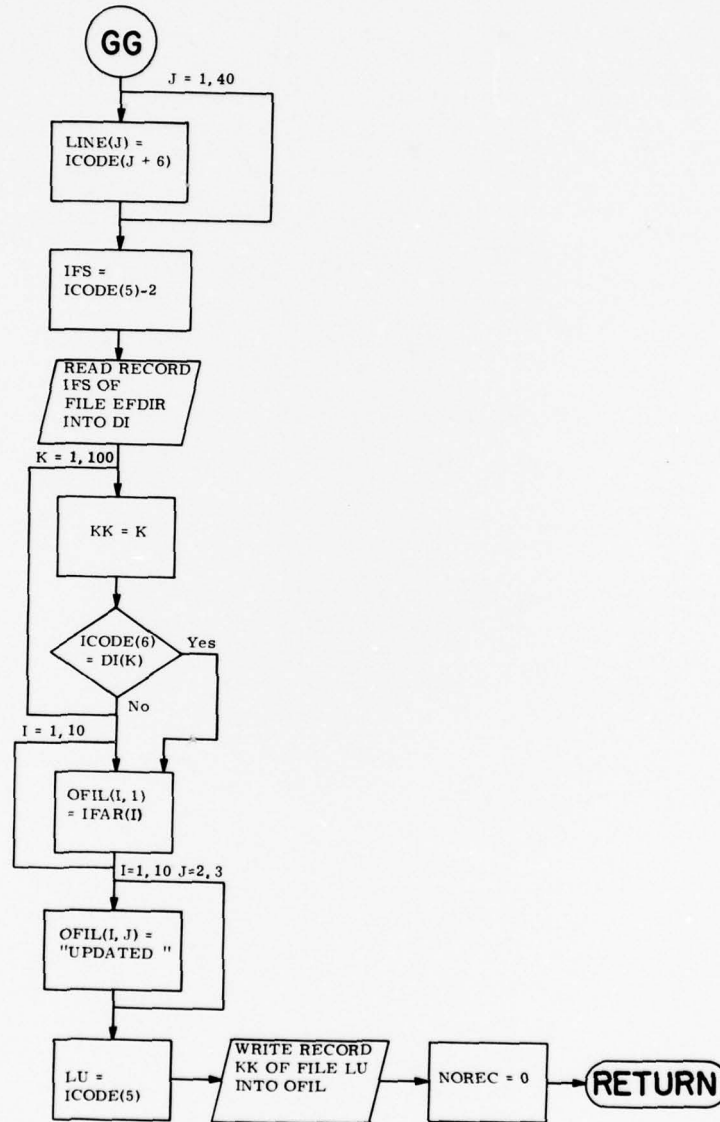


Figure 1-19. (Cont.)

FORTRAN IV V01B-02
 CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:27:27

PAGE 001

HST.OBJ=HST.FOR/NDSN/LI:1

```

0001       SUBROUTINE HST
0002       REAL*8 MOUT, OFIL, XMT, IFAR, UPD
0003       INTEGER ST, OI
0004       COMMON /LOOP/ICODE(128), MSK, LSK
0005       COMMON /U000/MOUT(10, 11), NRCNO, NOREC
0006       COMMON /DSK/I1, I2, I3, I4, I5, I6, I16
0007       DIMENSION OFIL(10, 3), OI(100)
0008       DIMENSION LINE(40), IFAR(10)
0009       EQUIVALENCE (LINE, IFAR)
0010       DATA XMT, UPD, /'PRS XMIT', /'UPDATED' /
0011       DATA ISPC/'120240'/
0012       IF (ICODE(4) .EQ. 1) GOTO 21
0014       IF (ICODE(4) .EQ. 2) GOTO 22
0016       IF (ICODE(4) .EQ. 3) GOTO 23
0018       IF (ICODE(4) .EQ. 4) GOTO 24
0020       NOREC=0
0021       RETURN
0022       21 ICODE(2)=0
0023       ICODE(3)=IAND(LSK, ICODE(7))+ISHFT(5, 8)
0024       IFS=ICODE(5)-2
0025       READ(2' IFS, ERR=99)(OI(I), I=1, 100)
0026       DO 550 K=1, 100
0027       KK=K
0028       IF (ICODE(6) .EQ. OI(K)) GOTO 560
0030       550 CONTINUE
0031       560 LU=ICODE(5)
0032       READ(LU, KK, ERR=99)((OFIL(I, J), I=1, 10), J=1, 3)
0033       DO 582 J=1, 3
0034       DO 582 I=1, 10
0035       582 MOUT(I, J)=OFIL(I, J)
0036       DO 682 J=1, 3
0037       682 WRITE(8' J+33)(MOUT(I, J), I=1, 10)
0038       IF (ICODE(8) .EQ. 2) GOTO 30
0040       DO 650 J=4, 6
0041       READ(8' J+156)(MOUT(I, J), I=1, 10)
0042       650 WRITE(8' J+33)(MOUT(I, J), I=1, 10)
0043       NRCNO=34
0044       NOREC=6
0045       RETURN
0046       30 NRCNO=34
0047       NOREC=3
0048       RETURN
0049       99 NRCNO=11
0050       NOREC=1
0051       RETURN
0052       22 DO 400 J=1, 40
0053       400 LINE(J)=ICODE(J+6)
0054       IFS=ICODE(5)-2
0055       READ(2' IFS, ERR=99)(OI(I), I=1, 100)
0056       DO 401 K=1, 100
0057       KK=K
0058       IF (ICODE(6) .EQ. OI(K)) GOTO 402
0060       401 CONTINUE
0061       402 DO 410 I=1, 10
0062       410 OFIL(I, 1)=IFAR(I)
0063       DO 1250 I=1, 10 .

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:27:27

PAGE 002

HST.OBJ=HST.FOR/NOSH/LI.1

```
0064      DO 1250 J=2,3
0065 1250  OFIL(I,J)=UPD
0066      LU=ICODE(5)
0067      WRITE(LU, KK, ERR=99)((OFIL(I,J), I=1,10), J=1,3)
0068      NOREC=0
0069      RETURN
0070      23 IFS=ICODE(5)-2
0071      READ(2' IFS, ERR=99)(DI(I), I=1,100)
0072      DO 1055 K=1,99
0073      KK=K
0074      IF (DI(K) .EQ. 0) GOTO 1060
0076 1055  CONTINUE
0077 1060  DI(KK)=-ICODE(6)
0078      DI(KK+1)=0
0079      WRITE(2' IFS, ERR=99)(DI(I), I=1,100)
0080      NOREC=0
0081      RETURN
0082      24 IFS=ICODE(5)-2
0083      READ(2' IFS, ERR=99)(DI(I), I=1,100)
0084      DO 1070 K=1,100
0085      KK=K
0086      IF (ICODE(6) .EQ. DI(K))GOTO 1075
0088      IF (ICODE(6) .EQ. -DI(K)) GOTO 1075
0090 1070  CONTINUE
0091 1075  IF (DI(KK+1) .EQ. 0) DI(KK)=0
0093      IF (DI(KK+1) .NE. 0) DI(KK)=ISPC
0095      WRITE(2' IFS, ERR=99)(DI(I), I=1,100)
0096      NOREC=0
0097      RETURN
0098      END
```

>

```
0001       SUBROUTINE HST
0002       REAL*8 MOUT,OFIL,XMT,IFAR,UPD
0003       INTEGER ST,DI
0004       COMMON /LOOP/ICODE(128),MSK,LSK
0005       COMMON /U000/MOUT(10,11),NRCNO,NOREC
0006       COMMON /DSK/I1,I2,I3,I4,I5,I6,I16
0007       DIMENSION OFIL(10,3),DI(100)
0008       DIMENSION LINE(40),IFAR(10)
0009       EQUIVALENCE (LINE,IFAR)
0010       DATA XMT,UPD/'FRS XMIT','UPDATED'/'
0011       DATA ISPC/'120240/'
0012       IF (ICODE(4) .EQ. 1) GOTO 21
0014       IF (ICODE(4) .EQ. 2) GOTO 22
0016       IF (ICODE(4) .EQ. 3) GOTO 23
0018       IF (ICODE(4) .EQ. 4) GOTO 24
0020       NOREC=0
0021       RETURN
0022       21 ICODE(2)=0
0023       ICODE(3)=IAND(LSK,ICODE(7))+ISHFT(1,8)
0024       IFS=ICODE(5)-2
0025       READ(2'IFS,ERR=99)(DI(I),I=1,100)
0026       DO 550 K=1,100
0027       KK=K
0028       IF (ICODE(6) .EQ. DI(K)) GOTO 560
0030       550 CONTINUE
0031       560 LU=ICODE(5)
0032       READ(LU'KK,ERR=99)((OFIL(I,J),I=1,10),J=1,3)
0033       DO 582 J=1,3
0034       DO 582 I=1,10
0035       582 MOUT(I,J)=OFIL(I,J)
0036       DO 682 J=1,3
0037       682 WRITE(8'J+33)(MOUT(I,J),I=1,10)
0038       IF (ICODE(8) .EQ. 2) GOTO 30
0040       DO 650 J=4,6
0041       READ(8'J+156)(MOUT(I,J),I=1,10)
0042       650 WRITE(8'J+33)(MOUT(I,J),I=1,10)
0043       NRCNO=34
0044       NOREC=6
0045       RETURN
0046       30 NRCNO=34
0047       NOREC=3
0048       RETURN
0049       99 NRCNO=11
0050       NOREC=1
0051       RETURN
0052       22 DO 400 J=1,40
0053       400 LINE(J)=ICODE(J+6)
0054       IFS=ICODE(5)-2
0055       READ(2'IFS,ERR=99)(DI(I),I=1,100)
0056       DO 401 K=1,100
0057       KK=K
0058       IF (ICODE(6) .EQ. DI(K)) GOTO 402
0060       401 CONTINUE
0061       402 DO 410 I=1,10
0062       410 OFIL(I,1)=IFAR(I)
0063       DO 1250 I=1,10
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 19:28:42

PAGE 002

HST1.OBJ=HST1.FOR/NOSN/LI:1

```
0064      DO 1250 J=2,3
0065 1250 OFIL(I,J)=UPD
0066      LU=ICODE(5)
0067      WRITE(LU'KK,ERR=99)((OFIL(I,J),I=1,10),J=1,3)
0068      NOREC=0
0069      RETURN
0070      23 IFS=ICODE(5)-2
0071      READ(2'IFS,ERR=99)(DI(I),I=1,100)
0072      DO 1055 K=1,99
0073      KK=K
0074      IF (DI(K) .EQ. 0) GOTO 1060
0076 1055 CONTINUE
0077 1060 DI(KK)=-ICODE(6)
0078      DI(KK+1)=0
0079      WRITE(2'IFS,ERR=99)(DI(I),I=1,100)
0080      NOREC=0
0081      RETURN
0082      24 IFS=ICODE(5)-2
0083      READ(2'IFS,ERR=99)(DI(I),I=1,100)
0084      DO 1070 K=1,100
0085      KK=K
0086      IF (ICODE(6) .EQ. DI(K))GOTO 1075
0088      IF (ICODE(6) .EQ. -DI(K)) GOTO 1075
0090 1070 CONTINUE
0091 1075 IF (DI(KK+1) .EQ. 0) DI(KK)=0
0093      IF (DI(KK+1) .NE. 0) DI(KK)=ISPC
0095      WRITE(2'IFS,ERR=99)(DI(I),I=1,100)
0096      NOREC=0
0097      RETURN
0098      END
```

>

EDI MSG.DAT
[00080 LINES READ IN]
[PAGE 03
*BLOCK OFF
*T

*LI
THIS IS THE ESM - (EXPLORATORY SYSCON MODEL)
ENTER USERCODE PLEASE
ENTER PASSWORD PLEASE
YOU ARE NOW LOGGED IN - (TO LOGOUT, ENTER 'DS')
PLEASE SELECT ONE MODE OF OPERATION

- 1. CRT TO CRT
- 2. SYSTEM INQUIRY
- 3. SYSTEM CONTROL
- 4. FILE ACCESS
- 5. CARD FORMAT

INVALID ENTRY - PLEASE TRY AGAIN (OR ENTER DS TO LOGOUT)
ENTER DEST CRT NODE DESIGNATOR(ND) - 4 FOR LP#2, 8 FOR LP#3
IF NOT KNOWN ENTER 'NDI'

PLEASE TYPE IN MESSAGE AND TRANSMIT
PLEASE SELECT ONE MODE OF OPERATION

- 1. NEW MESSAGE TO SAME CRT
- 2. NEW MESSAGE TO ANOTHER CRT
- 3. LOGOUT
- 4. NEW MODE OF OPERATION.

020 NOT YET IMPLEMENTED-PLEASE RESELECT
YOU ARE LOGGED OUT FROM ESM

PLEASE SELECT TYPE OF SYSTEM INFORMATION

- 1. NETWORK DEVICE INFORMATION
- 2. LID/FAD CONVERSION TABLE (LID'S 1-100)
- 3. LID/FAD CONVERSION TABLE (LID'S 101-254)
- 4. WORKPAGE PARAMETERS OF NODE.

PLEASE ENTER NODE DESIGNATOR (ND).
IF ND IS NOT KNOWN, ENTER NDI
FOR NETWORK DEVICE INFORMATION.

PLEASE SELECT ONE OF THE FOLLOWING:

- 1. NEW SYSTEM INQUIRY.
- 2. LOGOUT.
- 3. ANOTHER MODE OF OPERATION.

MSG TO CRT ND= 8

PLEASE SELECT ONE MODE OF OPERATION

- 1. NEW MESSAGE TO SAME CRT
- 2. NEW MESSAGE TO ANOTHER CRT
- 3. LOGOUT
- 4. NEW MODE OF OPERATION.

GATE 2 2 1 6 1 3 10 2 1
GATE 3 3 2 7 3 4 11 1 4

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, RDA IS FUNCTIONAL ADDRESS
AND WTD IS WRITE TOKEN DESTINATION.
PRESS 'T' KEY FOR NEXT INSTRUCTION.

LID/FAD CONVERSION TABLE

1 1 1 2 2 2 2 4 3 2 1
0
0
0
0
0
0 0 0 0 0 0
0
0
0

```

0
0
0 0 0 0 0 0 0 0 0 0 0 0 0 1
      NODE WORKPAGE PARAMETERS
CRT  NODE HAS DESIGNATOR      8      RDA      4 IN LOOP      3
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . 1, 2
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . 8
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . 8
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 12
TIMEOUT FOR PACKET RETRANSMISSION . . . . . 41
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 4
PRESS 'T' KEY FOR NEXT INSTRUCTION
PLEASE SELECT TYPE OF SYSTEM PARAMETERS
TO BE CHANGED.
  1. NETWORK DEVICE PARAMETERS.
  2. LID/FAD CONVERSION TABLE (LID'S 1-100)
  3. LID/FAD CONVERSION TABLE (LID'S 101-254)
  4. WORKPAGE PARAMETERS. (NOT YET IMPLEMENTED IN CIE MEMORY)
PLEASE SELECT NODE DESIGNATOR (ND) & ENTER
IF ND IS NOT KNOWN, ENTER 'NDI' FOR DISPLAY.
PLEASE SELECT PARAMETER TO BE CHANGED, FOLLOWED
BY THE NEW VALUE. (FORMAT I1,I3,5X)
  1. NODE DESIGNATOR -- NOT IMPLEMENTED
  2. FUNCTIONAL ADDRESS.
  3. WRITE TOKEN DESTINATION.
  4. NO CHANGE.
PLEASE ENTER LID FOLLOWED BY NEW FAD (FORMAT I4,I4).
FOR TABLE DISPLAY ENTER 'LID'.
PLEASE ENTER LID FOLLOWED BY NEW FAD
(FORMAT I4,I4)
FOR TABLE PAGE DISPLAY, ENTER 'TAB'.
PLEASE ENTER ONE OF THE FOLLOWING FOLLOWED BY NEW
VALUE (FORMAT I1,A7). FOR NODE WKPG DISPLAY, ENTER 'NNWD'.
  1. ALTERNATE GATEWAY FUNCTIONAL ADDRESS .
  2. ALTERNATE GATEWAY FUNCTIONAL ADDRESS .
  3. MAXIMUM INPUT QUEUE SIZE (EXTERNAL).
  4. MAXIMUM OUTPUT QUEUE SIZE (BITSTREAM).
  5. MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE.
  6. TIMEOUT FOR WRITE TOKEN REGENERATION.
  7. TIMEOUT FOR PACKET RETRANSMISSION.
  8. NUMBER OF NODES IN SYSTEM.
  9. NUMBER OF NODES IN LOOP.
100 PLEASE SELECT ONE OF THE FOLLOWING:
101 1. SYSTEM UPDATE OF CHANGE
102 2. LOOP UPDATE OF CHANGE.
103 3. NO ACTION.
PLEASE SELECT ONE OF THE FOLLOWING:
  1. NEW SYSTEM CONTROL OF SAME TYPE
  2. NEW SYSTEM CONTROL OF DIFFERENT TYPE
  3. LOGOUT.
  4. ANOTHER MODE OF OPERATION.
DO YOU WISH TO EXECUTE AN ESM
DEMONSTRATION PROGRAM?
  1. YES
  2. NO
PLEASE SELECT PROGRAM TO BE EXECUTED.
USE CRT AS AN I/O DEVICE FOR THE PROGRAM.
1. CRT BROADCAST.

```

2. RECORD MOVE.
3. INTERPROCESS COMMUNICATION.--ABORT USRLNG, RUN PROC

.....
.....

YOUR CRT WILL ACT AS A USER TERMINAL.
IT WILL ATTACH TO A HOST COMPUTER. PLEASE ADHERE
TO STANDARD MCR AND UTILITIES CONTROL FORMATS. PRESS XMIT

PLEASE SELECT ONE OF THE FOLLOWING:

1. NEW CARD FORMAT RUN.
2. NEW MODE OF OPERATION.
3. LOGOUT.

4. NEW DEMO PROGRAM.

PLEASE SELECT FILE TO BE ACCESSED:(ONLY #1-4 ON DISK)

- | | |
|-------------------------|--------------------------|
| 01. LOCATION FILE | 06. SUBSTANDARD CIRCUITS |
| 02. CIRCUIT DIRECTORY | 07. MESSAGE FILE |
| 03. TRUNK DIRECTORY | 08. SUBSCRIBER LIST |
| 04. TERMINAL DIRECTORY | 09. INSTALLATION LIST |
| 05. SATELLITE DIRECTORY | 10. TRAFFIC REPORT FILE |

A RECORD OF THE FILE YOU HAVE SELECTED

HAS THE FOLLOWING FORMAT:

THE KEY HAS A
CHARACTER CODE IN FORM

DO YOU WISH TO MODIFY THIS FILE?

1. YES.
2. NO.

THE 2 BYTE ALPHANUMERIC LOCATION KEY OF THE CIRCUIT OR
TRUNK DIRECTORY FILES MAY BE USED AS A KEY TO CROSS-
REFERENCE THE TERMINAL DIRECTORY FILE. DO YOU WISH
TO CROSS-REFERENCE?

1. YES.
2. NO.

PLEASE ENTER ONE OF THE ABOVE INTEGER VALUES ON CRT DISPLAY.
THE 2 BYTE ALPHANUMERIC LOCATION KEY OF THE TERMINAL
DIRECTORY FILE MAY BE USED AS A KEY TO CROSS-REFERENCE
THE CIRCUIT DIRECTORY AND/OR TRUNK DIRECTORY FILES.

PLEASE SELECT MODE OF ACCESS.

1. NO CROSS-REFERENCE.
2. CROSS-REFERENCE CIRCUIT DIRECTORY.
3. CROSS-REFERENCE TRUNK DIRECTORY.
4. CROSS-REFERENCE BOTH.

PLEASE ENTER ONE OF THE ABOVE INTEGER VALUES ON CRT DISPLAY.

PLEASE ENTER ACCESS KEY.

PLEASE ENTER KEY OF RECORD TO BE MODIFIED

RECORD MAY BE LOCKED. *NOT YET IMPLEMENTED*

FOR THIS RECORD PLEASE SELECT TYPE OF DESIRED CHANGE

1. UPDATE.
2. DELETE.

MAKE ANY CHANGES YOU WISH USING CRT KEYBOARD.
WHEN CHANGES ARE COMPLETE, PRESS XMIT KEY.

ENTER UPDATED RECORD ON FIRST LINE OF CRT

THE RECORD DOES NOT EXIST. DO YOU WISH
TO ADD A RECORD TO THE FILE?

1. YES
2. NO

KEY IS

CHARACTERS OF TYPE

ENTER THE RECORD ACCORDING TO THE ABOVE FORMAT.

WHEN RECORD IS COMPLETE, PRESS XMIT KEY.

ENTER NEW RECORD ON FIRST LINE OF CRT.

** MODIFICATION COMPLETE **

PLEASE SELECT ONE OF THE FOLLOWING:

1. NEW RECORD OF FILE.
 2. NEW FILE.
 3. NEW MODE OF OPERATION.
 4. LOGOUT
 5. DISPLAY SAME RECORD.
1000 RECORDS EACH 8 BYTES LONG
FACILITY DESCRIPTION(2)-KEY
DESCRIPTIVE INFO(6).
LOCATION(2) KEY
TYPE OPERATION(1),TYPE SERV(1),SUBSCRIB. RT.(2),MOD RT(2),TECH SPEC(2),
COMMER CO(4), DCA AREA(1),AVAILABILITY(1), AGENCY CODE(2).
LOCATION(2)-KEY
BANDWIDTH(3),CHAN NO(1),ROUTE NO(2),
DCA AREA(1),AVAIL(1),CAPACITY(2),TRNK MI(2),SUPERGROUP(2).
TERM EQUIP(2)-KEY
OTHER TERM EQUIP(2),LOCATION(2),-KEY FOR CR,
TRUNK INFO(2),CIRCUIT INFO(2).
- 192 NAME/CODE(4)-KEY
193 CONDITION(2),CAPACITY(2),
194 POWER(2),BANDWIDTH(2),AUTHORIZATION(2).
195 LOCATION(2)-KEY
196 TYPE OP(1),TYP SERV(1),SUB RT(2),MOD RT(2),TECH SPEC(2),
197 COMM CO(4),DCA AREA(1),AVAIL(1),AGENCY CODE(2),CIRC NO(2).
198 SOURCE CODE/DEST CODE(4)-KEY
199 LENGTH(2),CLASS(2),SEG NO(1),NO ADD SEGMENTS(1),
200 MESSAGE(40).
201 AGENCY CODE(2)-KEY
202 NAME(4),AUTHORIZATION(2),
203 DCA AREA(1),LOCATION # (1)
204 AGENCY CODE(2)-KEY
205 NAME(6),LOCATION # (1),DCA AREA(1),
206 AUTH CIV(2),AUTH MIL(2),BRANCH.
207 50 RECORDS EACH 84 BYTES LONG
208 SEGMENT NO(2)-KEY
209 LENGTH(2),REPORT(80).
THE RECORD DOES NOT EXIST. PRESS TRANSMIT PLEASE.
*ED
EDI -- DEVICE FULL
[EXIT]

>

EDI JUNK
[CREATING NEW FILE]
INPUT

*SIZE 50
*KILL
EDI>INFOPM.DAT
[00050 LINES READ IN]
[PAGE 03
*BLOCK OFF
*T
*LI

NETWORK DEVICE INFORMATION

LOCAL LOOP		LOCAL LOOP						LOCAL LOOP	
LOOP #1		LOOP #1		LOOP #2		LOOP #2		LOOP #3	
NT	ND	RDA	WTD	ND	RDA	WTD	ND	RDA	WTD
CRT	--	--	--	4	4	2	8	4	3
HOST	1	1	3	5	2	1	9	3	2
GATE	2	2	1	6	1	3	10	2	1
GATE	3	3	2	7	3	4	11	1	4

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, RDA IS FUNCTIONAL ADDRESS
AND WTD IS WRITE TOKEN DESTINATION.
PRESS 'T' KEY FOR NEXT INSTRUCTION.

NETWORK DEVICE INFORMATION

LOCAL LOOP		LOCAL LOOP						LOCAL LOOP	
LOOP #1		LOOP #1		LOOP #2		LOOP #2		LOOP #3	
NT	ND	FAD	WTD	ND	FAD	WTD	ND	FAD	WTD
CRT	--	--	--	4	4	3	8	4	3
HOST	1	1	2	5	2	4	9	3	1
GATE	2	2	3	6	1	2	10	1	2
GATE	3	3	1	7	3	1	11	2	4

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, FAD IS FUNCTIONAL ADDRESS
AND WTD IS WRITE TOKEN DESTINATION.
PRESS 'T' KEY FOR NEXT INSTRUCTION.

NETWORK DEVICE INFORMATION

LOCAL LOOP		LOCAL LOOP						LOCAL LOOP	
LOOP #1		LOOP #1		LOOP #2		LOOP #2		LOOP #3	
NT	ND	FAD	WTD	ND	FAD	WTD	ND	FAD	WTD
CRT	--	--	--	4	4	3	8	4	3
HOST	1	1	2	5	2	4	9	3	1
GATE	2	2	3	6	1	2	10	1	2
GATE	3	3	1	7	3	1	11	2	4

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, FAD IS FUNCTIONAL ADDRESS
AND WTD IS WRITE TOKEN DESTINATION.
PRESS 'T' KEY FOR NEXT INSTRUCTION.

NETWORK DEVICE INFORMATION

LOCAL LOOP		LOCAL LOOP						LOCAL LOOP	
LOOP #1		LOOP #1		LOOP #2		LOOP #2		LOOP #3	
NT	ND	FAD	WTD	ND	FAD	WTD	ND	FAD	WTD
CRT	--	--	--	4	4	3	8	4	3
HOST	1	1	2	5	2	4	9	3	1
GATE	2	2	3	6	1	2	10	1	2
GATE	3	3	1	7	3	1	11	2	4

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, FAD IS FUNCTIONAL ADDRESS
AND WTD IS WRITE TOKEN DESTINATION.
PRESS 'T' KEY FOR NEXT INSTRUCTION.

NETWORK DEVICE INFORMATION

LOCAL LOOP LOOP #1				LOOP #2				LOOP #3		
NT	ND	FAD	WTD	ND	FAD	WTD	ND	FAD	WTD	
CRT	--	--	--	4	4	3	8	4	3	
HOST	1	1	2	5	2	4	9	3	1	
GATE	2	2	3	6	1	2	10	1	2	
GATE	3	3	1	7	3	1	11	2	4	

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, FAD IS FUNCTIONAL ADDRESS AND WTD IS WRITE TOKEN DESTINATION. PRESS 'T' KEY FOR NEXT INSTRUCTION.

NETWORK DEVICE INFORMATION

LOCAL LOOP LOOP #1				LOOP #2				LOOP #3		
NT	ND	FAD	WTD	ND	FAD	WTD	ND	FAD	WTD	
CRT	--	--	--	4	4	3	8	4	3	
HOST	1	1	2	5	2	4	9	3	1	
GATE	2	2	3	6	1	2	10	1	2	
GATE	3	3	1	7	3	1	11	2	4	

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, FAD IS FUNCTIONAL ADDRESS AND WTD IS WRITE TOKEN DESTINATION. PRESS 'T' KEY FOR NEXT INSTRUCTION.

NETWORK DEVICE INFORMATION

LOCAL LOOP LOOP #1				LOOP #2				LOOP #3		
NT	ND	FAD	WTD	ND	FAD	WTD	ND	FAD	WTD	
CRT	--	--	--	4	4	3	8	4	3	
HOST	1	1	2	5	2	4	9	3	1	
GATE	2	2	3	6	1	2	10	1	2	
GATE	3	3	1	7	3	1	11	2	4	

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, FAD IS FUNCTIONAL ADDRESS AND WTD IS WRITE TOKEN DESTINATION. PRESS 'T' KEY FOR NEXT INSTRUCTION.

NETWORK DEVICE INFORMATION

LOCAL LOOP LOOP #1				LOOP #2				LOOP #3		
NT	ND	FAD	WTD	ND	FAD	WTD	ND	FAD	WTD	
CRT	--	--	--	4	4	3	8	4	3	
HOST	1	1	2	5	2	4	9	3	1	
GATE	2	2	3	6	1	2	10	1	2	
GATE	3	3	1	7	3	1	11	2	4	

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, FAD IS FUNCTIONAL ADDRESS AND WTD IS WRITE TOKEN DESTINATION. PRESS 'T' KEY FOR NEXT INSTRUCTION.

NETWORK DEVICE INFORMATION

LOCAL LOOP LOOP #1				LOOP #2				LOOP #3		
NT	ND	FAD	WTD	ND	FAD	WTD	ND	FAD	WTD	
CRT	--	--	--	4	4	3	8	4	3	
HOST	1	1	2	5	2	4	9	3	1	
GATE	2	2	3	6	1	2	10	1	2	
GATE	3	3	1	7	3	1	11	2	4	

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, FAD IS FUNCTIONAL ADDRESS AND WTD IS WRITE TOKEN DESTINATION. PRESS 'T' KEY FOR NEXT INSTRUCTION.

NETWORK DEVICE INFORMATION

LOCAL LOOP LOOP #1			LOOP #2				LOOP #3		
NT	ND	FAD	WTD	ND	FAD	WTD	ND	FAD	WTD
CRT	--	--	--	4	4	3	8	4	3
HOST	1	1	2	5	2	4	9	3	1
GATE	2	2	3	6	1	2	10	1	2
GATE	3	3	1	7	3	1	11	2	4

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, FAD IS FUNCTIONAL ADDRESS AND WTD IS WRITE TOKEN DESTINATION. PRESS 'T' KEY FOR NEXT INSTRUCTION.

NETWORK DEVICE INFORMATION

LOCAL LOOP LOOP #1			LOOP #2				LOOP #3		
NT	ND	FAD	WTD	ND	FAD	WTD	ND	FAD	WTD
CRT	--	--	--	4	4	3	8	4	3
HOST	1	1	2	5	2	4	9	3	1
GATE	2	2	3	6	1	2	10	1	2
GATE	3	3	1	7	3	1	11	2	4

NOTE: NT IS NODE TYPE ND IS NODE DESIGNATOR, FAD IS FUNCTIONAL ADDRESS AND WTD IS WRITE TOKEN DESTINATION. PRESS 'T' KEY FOR NEXT INSTRUCTION.

LID/FAD CONVERSION TABLE

1	2	3	2	2	2	2	3	3	3	3
0										
0										
0										

LID/FAD CONVERSION TABLE

1	2	3	2	2	2	2	3	3	3	3
0										
0										
0										

LID/FAD CONVERSION TABLE

1	2	3	2	2	2	2	3	3	3	3
0										
0										
0										

LID/FAD CONVERSION TABLE

1	1	1	4	2	3	1	3	3	3	3
0										
0										
0										

LID/FAD CONVERSION TABLE

1	1	1	4	2	3	1	3	3	3	3
0										
0										
0										

LID/FAD CONVERSION TABLE

1	1	1	4	2	3	1	3	3	3	3
0										
0										
0										

LID/FAD CONVERSION TABLE
1 1 1 4 2 3 1 3 3 3 3

0
0
0
0

LID/FAD CONVERSION TABLE
1 1 1 2 2 2 2 4 3 2 1

0
0
0
0

LID/FAD CONVERSION TABLE
1 1 1 2 2 2 2 4 3 2 1

0
0
0
0

LID/FAD CONVERSION TABLE
1 1 1 2 2 2 2 4 3 2 1

0
0
0
0

LID/FAD CONVERSION TABLE
1 1 1 2 2 2 2 4 3 2 1

0
0
0
0

0 0 0 0 0 0

0
0
0
0

0 0 0 0 0 0 0 0 0 0 0 0 2

0
0
0
0

0 0 0 0 0 0

0
0
0
0

0 0 0 0 0 0 0 0 0 0 0 0 3

0
0
0
0

0 0 0 0 0 0

0
0
0
0

0 0 0 0 0 0 0 0 0 0 0 1

0
0
0
0

0 0 0 0 0 2

0
0
0
0

```
0 0 0 0 0 0 0 0 0 0 0 0 0 3
0
0 0 0 0 0 4
0
0
0
0
0
0 0 0 0 0 0 0 0 0 0 0 0 4
0 0 0 0 0 0
0
0
0 0 0 0 0 0 0 0 0 0 0 0 2
0 0 0 0 0 4
0
0
0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 2
0
0
0 0 0 0 0 0 0 0 0 0 0 0 3
0 0 0 0 0 0
0
0
0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 4
0
0
0 0 0 0 0 0 0 0 0 0 0 0 2
0 0 0 0 0 0
0
0
0
0
0 0 0 0 0 0 0 0 0 0 0 0 4
```

 NODE WORKPAGE PARAMETERS
MOST NODE HAS DESIGNATOR 1 RDA 1 IN LOOP 1
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES 2, 3

```

MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . . 4
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . . 6
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 4
TIMEOUT FOR PACKET RETRANSMISSION . . . . . 41
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 3
PRESS 'T' KEY FOR NEXT INSTRUCTION
      NODE WORKPAGE PARAMETERS
GATE#2  NODE HAS DESIGNATOR      2      RDA      2 IN LOOP      1
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . . 2, 3
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . . 10
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE MSG TERM. . . . . NA
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 7
TIMEOUT FOR PACKET RETRANSMISSION . . . . . NA
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 3
PRESS 'T' KEY FOR NEXT INSTRUCTION
      NODE WORKPAGE PARAMETERS
GATE#3  NODE HAS DESIGNATOR      3      RDA      3 IN LOOP      1
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . . 2, 3
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . . 10
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . . NA
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 12
TIMEOUT FOR PACKET RETRANSMISSION . . . . . NA
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 3
PRESS 'T' KEY FOR NEXT INSTRUCTION
      NODE WORKPAGE PARAMETERS
CRT     NODE HAS DESIGNATOR      4      RDA      4 IN LOOP      2
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . . 1, 3
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . . 8
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . . 8
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 12
TIMEOUT FOR PACKET RETRANSMISSION . . . . . 41
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 4
PRESS 'T' KEY FOR NEXT INSTRUCTION
      NODE WORKPAGE PARAMETERS
HOST   NODE HAS DESIGNATOR      5      RDA      2 IN LOOP      2
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . . 1, 3
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . . 4
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . . 6
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 4
TIMEOUT FOR PACKET RETRANSMISSION . . . . . 41
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 4
PRESS 'T' KEY FOR NEXT INSTRUCTION
      NODE WORKPAGE PARAMETERS
GATE#1  NODE HAS DESIGNATOR      6      RDA      1 IN LOOP      2
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . . 1, 3
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . . 10
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . . NA
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 7
TIMEOUT FOR PACKET RETRANSMISSION . . . . . NA

```

```
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 4
PRESS 'T' KEY FOR NEXT INSTRUCTION
      NODE WORKPAGE PARAMETERS
GATE#3 NODE HAS DESIGNATOR      7      RDA      3 IN LOOP      2
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . 1, 3
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . 10
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . NA
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 6
TIMEOUT FOR PACKET RETRANSMISSION . . . . . NA
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 4
PRESS 'T' KEY FOR NEXT INSTRUCTION
      NODE WORKPAGE PARAMETERS
CRT NODE HAS DESIGNATOR      8      RDA      4 IN LOOP      3
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . 1, 2
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . 8
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . 8
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 12
TIMEOUT FOR PACKET RETRANSMISSION . . . . . 41
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 4
PRESS 'T' KEY FOR NEXT INSTRUCTION
      NODE WORKPAGE PARAMETERS
HOST NODE HAS DESIGNATOR      9      RDA      3 IN LOOP      3
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . 1, 2
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . 8
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . 8
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 14
TIMEOUT FOR PACKET RETRANSMISSION . . . . . 41
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 4
PRESS 'T' KEY FOR NEXT INSTRUCTION
      NODE WORKPAGE PARAMETERS
GATE#1 NODE HAS DESIGNATOR      10     RDA      1 IN LOOP      3
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . 1, 2
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . 10
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . NA
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 4
TIMEOUT FOR PACKET RETRANSMISSION . . . . . NA
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 4
PRESS 'T' KEY FOR NEXT INSTRUCTION
      NODE WORKPAGE PARAMETERS
GATE#2 NODE HAS DESIGNATOR      11     RDA 2      IN LOOP      3
ALTERNATE GATEWAY FUNCTIONAL ADDRESSES . . . . 1, 2
MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) . . . . 10
MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) . . . . 1
MAXIMUM PACKET XMISSIONS BEFORE ERROR MESSAGE . . NA
TIMEOUT FOR WRITE TOKEN REGENERATION . . . . . 7
TIMEOUT FOR PACKET RETRANSMISSION . . . . . NA
NUMBER OF NODES IN SYSTEM . . . . . 11
NUMBER OF NODES IN LOOP . . . . . 4
PRESS 'T' KEY FOR NEXT INSTRUCTION
#ED ?
EDI -- DEVICE FULL
[EXIT]
```

1.4 ESM Loader Utility

The ESM Loader Utility (ESMLDR) is used for loading the RAM control memories of the ESM B7* CIE microprocessors. Loading procedures are described in Section 4.3 of the ESM User Manual. Microcode object files normally reside in UIC [1,20]. The user must be in the UIC of the object file to be loaded when running ESMLDR. The object file must consist of 256 byte records. Sixteen bit instruction words resulting from the MDMP L Assembler are loaded into sequential control memory locations consisting of 12 bit instructions where the (left) most significant four bits of the sixteen bit word are dropped. The full 4K of control memory is loaded so that when an end-of-file error condition arises the remainder of memory is filled with STEP (octal 607) instructions. The end-of-file error message printed on the terminal is the normal display. The third and fourth instruction word is used for a recovery mode GOTO instruction for the case of hardware failures. This GOTO instruction is loaded at the last two instruction words so that hardware failures which cause jumps to non-programmed control memory result in STEP's being executed until the error recovery GOTO instruction at the last two words of control memory.

ESMLDR runs on host processor B only since loading hardware exists for that machine. ESM Tape #2 contains the source file (ESMLDR.FOR), object file (ESMLDR.OBJ), and task (ESMLDR.TSK). Task Builder (TKB) options include:

```
UNITS=2
ACTFIL=2
COMMON=M1710:RW
MAXBUF=256
ASG=TT1:1, SYO:2
```

ESMLDR

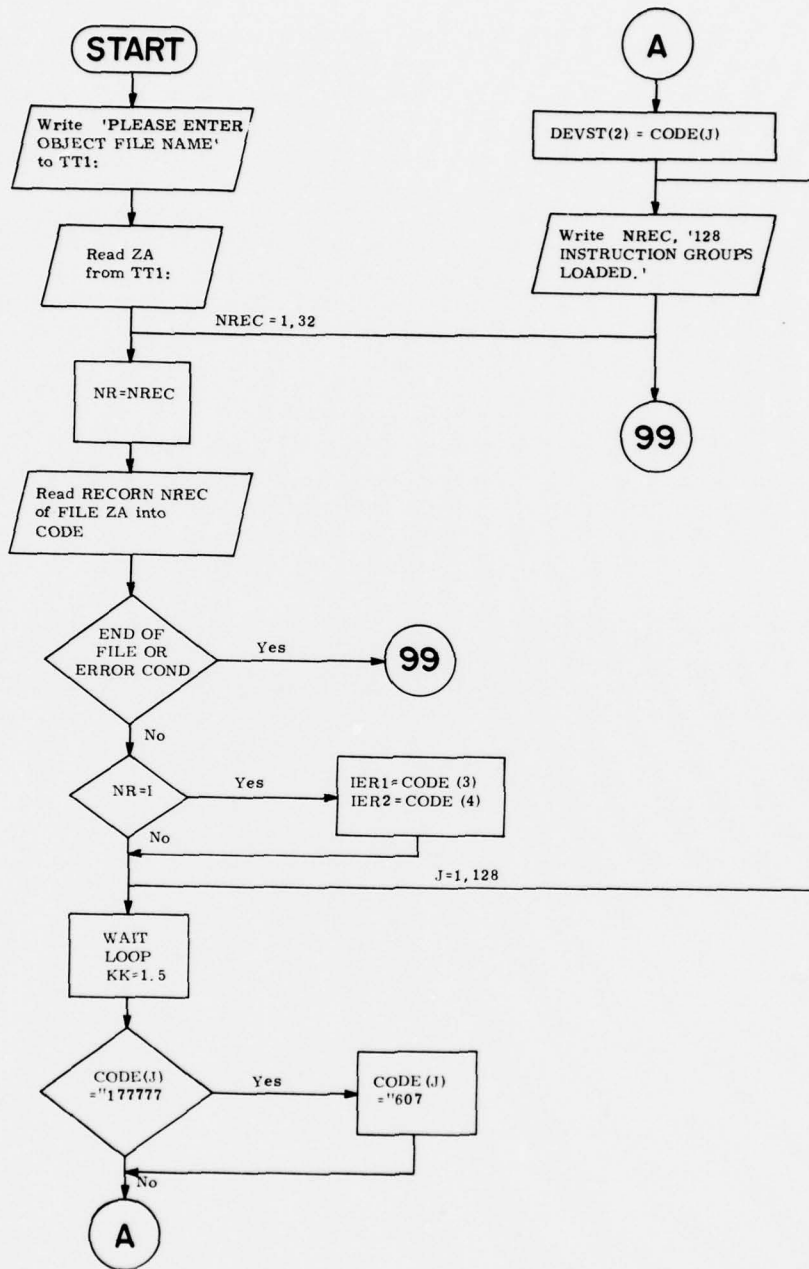


Figure 1-20. ESMLDR

ESMLDR (cont.)

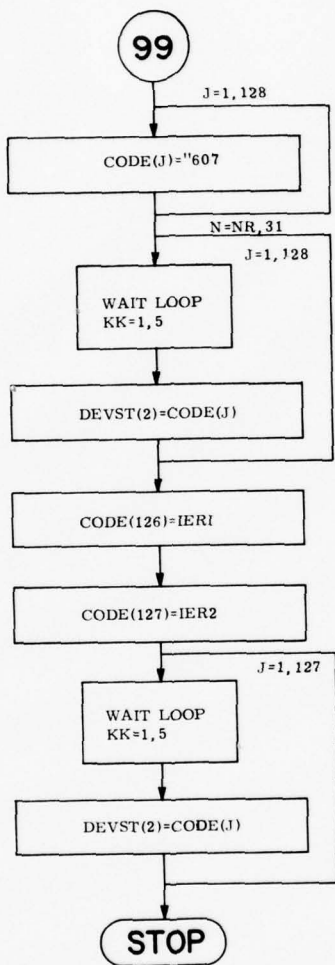


Figure 1-20. (Cont.)

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

SAT 26-FEB-77 14:52:10

PAGE 001

ESMLDR.OBJ=ESMLDR.FOR/NOSN/LI:1

```
0001        INTEGER DEVST(6),CODE(128)
0002        REAL*8 ZA(3)
0003        COMMON /M1710/DEVST
0004        DATA ISTEP/'607/
0005        DATA IAON/'177777/
0006        CALL ASSIGN(1,'TT1:')
0007        WRITE(1,11)
0008        11 FORMAT(1X,'PLEASE ENTER OBJECT FILE NAME')
0009        READ(1,12)ZA
0010        12 FORMAT(3A8)
0011        CALL ASSIGN(2,ZA)
0012        DEFINE FILE 2(32,128,U,I1)
0013        DO 20 NREC=1,32
0014        NR=NREC
0015        READ(2'NREC,END=99,ERR=99) CODE
0016        IF (NR .EQ. 1) IER1=CODE(3)
0018        IF (NR .EQ. 1) IER2=CODE(4)
0020        DO 18 J=1,128
0021        DO 16 KK=1,5
0022        16 CONTINUE
0023        IF (CODE(J) .EQ. IAON) CODE(J)=ISTEP
0025        DEVST(2)=CODE(J)
0026        18 CONTINUE
0027        WRITE (1,13) NREC
0028        13        FORMAT(1X,I3,' 128 INSTRUCTION GROUPS LOADED.')
0029        20 CONTINUE
0030        99 DO 30 J=1,128
0031        30 CODE(J)=ISTEP
0032        DO 40 N=NR,31
0033        DO 40 J=1,128
0034        DO 50 KK=1,5
0035        50 CONTINUE
0036        DEVST(2)=CODE(J)
0037        40 CONTINUE
0038        CODE(126)=IER1
0039        CODE(127)=IER2
0040        DO 60 J=1,127
0041        DO 70 KK=1,5
0042        70 CONTINUE
0043        DEVST(2)=CODE(J)
0044        60 CONTINUE
0045        END
```

>

1.5 Record Move Utility

The ESM Record Move Utility is used for moving records of the ATEC simulation files (EFLOCF, EFTRKD, EFCKTD, EFTERD) between host processors A and B to maintain the directory file (EFDIR) necessary for the distributed file system of mode 4 of the ESM User Language. The utility exists in two forms, RCMV1 for processor A, and RCMV5 for processor B. The two programs differ only in LID pair addresses (ICODE(3) of message header), and logging DECSCOPE terminal definition. ESM Tape #1 contains the source (.FOR), object (.OBJ), task (.TSK), and overlay description language files (.ODL) for the two programs.

The overlay structure for the utility consists of the main program and two overlaid routines RDLOOP and WRLOOP which are listed in Section 1.3.

The program allows records to be moved from the host processor which is the primary dialogue director for the ESM terminal running the utility to the other host processor. The user is given the choice of file and access key. If the key exists on the machine, the record is displayed and moved to the other machine by means of host-host interprocess communication control messages. The utility may be used for building a consistent distributed file system so that multiple copies of records are eliminated. To terminate the utility, enter "DS" on the ESM terminal. Note that both processors must be running the utility for successful record transfers.

Task Builder (TKB) Commands for building the utility task are:

```
TKB   RCMV5.TSK = RCMV5.ODL/MP,[1,1] SYSLIB/LB:$SHORT
```

Options include:

```
UNITS=8  
ACTFIL=8  
COMMON=M1710:RW  
MAXBUF=240  
ASG=TI:1, SYO:2:3:4:5:6:7:8
```

RCMV5

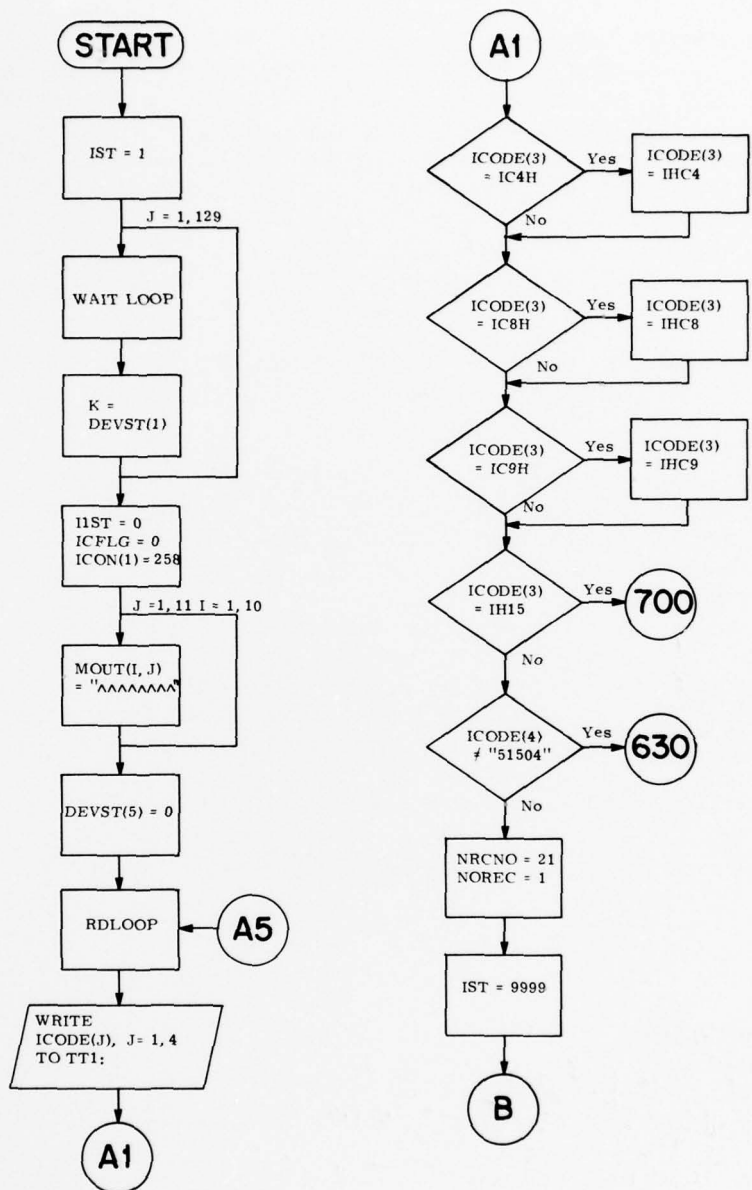


Figure 1-21. RCMV5

RCMV5(cont.)

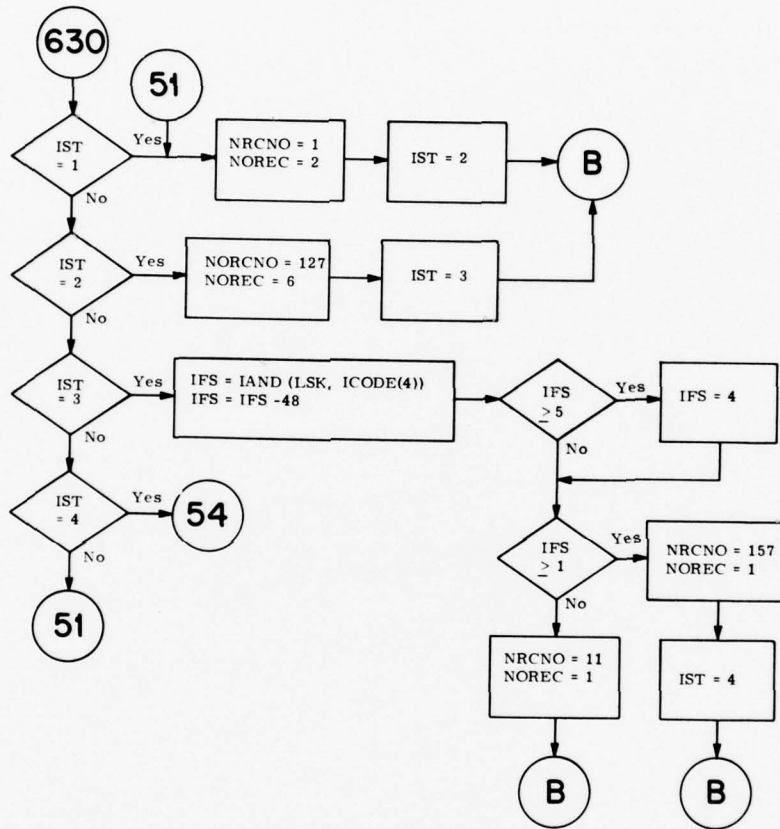


Figure 1-21. (Cont.)

RCMV5(cont.)

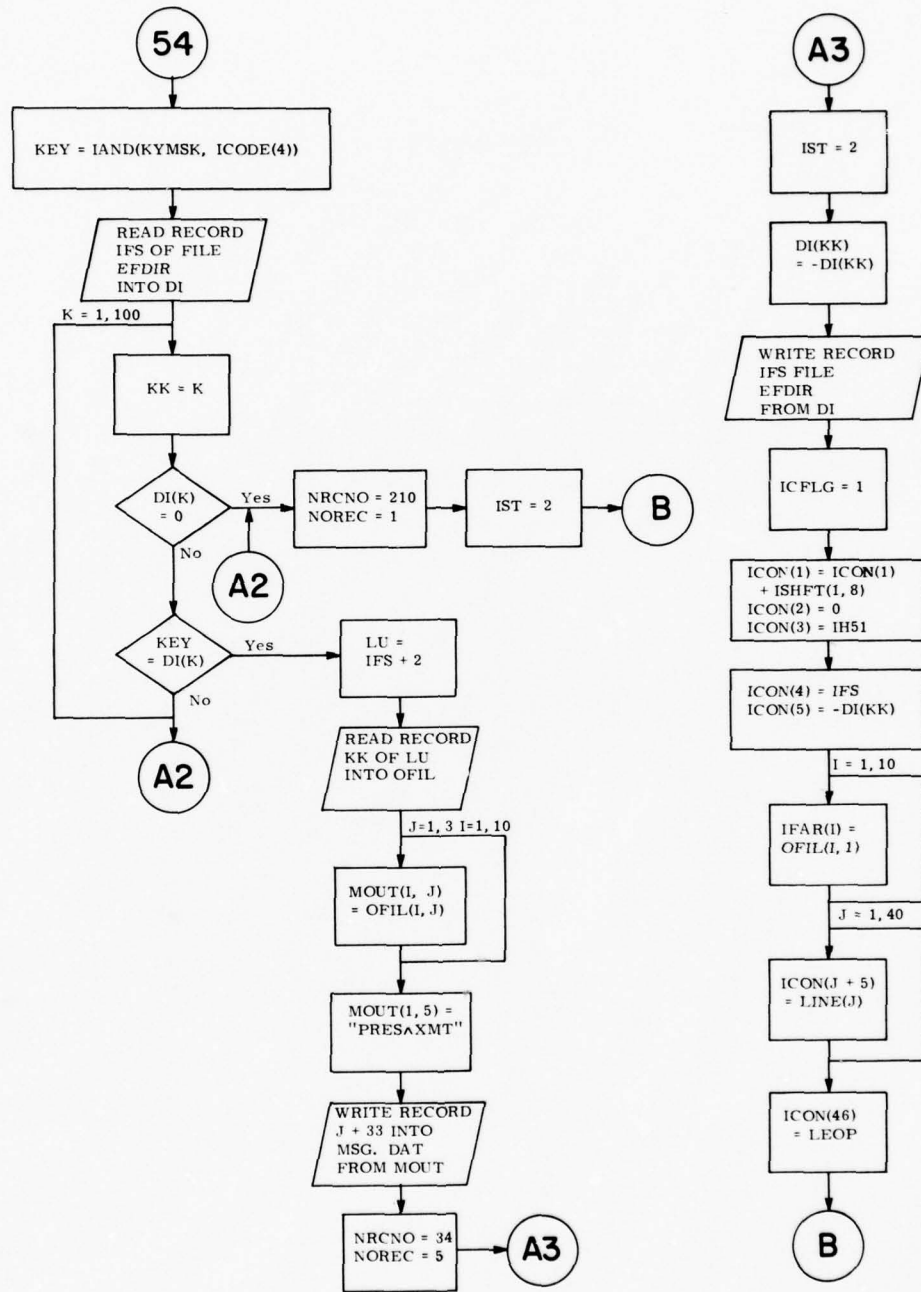


Figure 1-21. (Cont.)

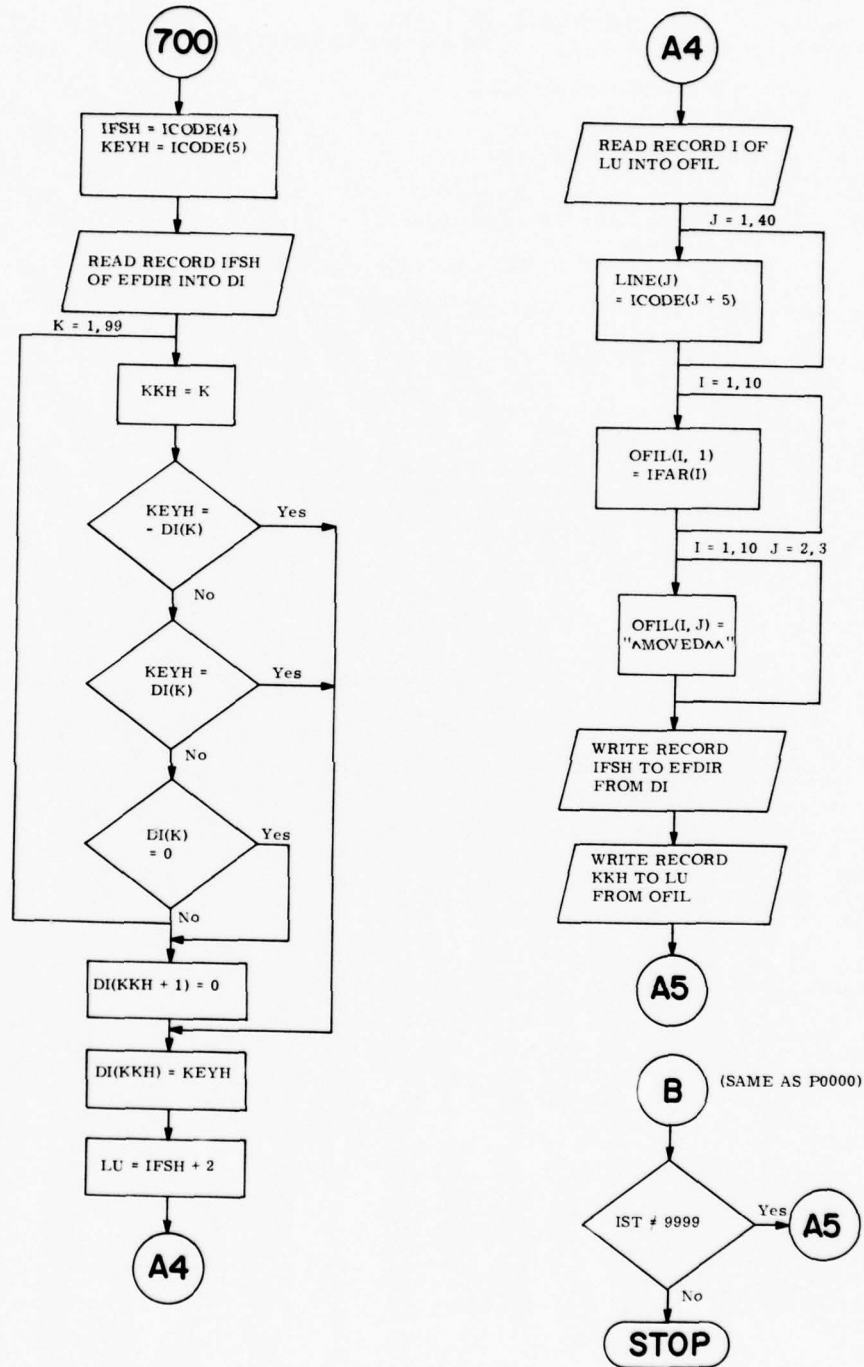


Figure 1-21. (Cont.)

FORTRAN IV V01R-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 11:24:14

PAGE 001

RCHV5.OBJ=RCHV5.FOR/NOSN/LI:1

```
0001 REAL*8 MOUT,OFIL,IFAR,MVD,XMT,SKB
0002 INTEGER DEVST(6)
0003 INTEGER DI
0004 COMMON /LOOP/ICODE(128),MSK,LSK
0005 COMMON /M1710/DEVST
0006 DIMENSION OFIL(10,3),DI(100),LINE(40)
0007 DIMENSION IFAR(10),ICON(128),MOUT(10,11)
0008 EQUIVALENCE(LINE,IFAR)
0009 DATA MVD,KYMSK,XMT,IH51/' MOVED ',*77577,'PRES XMT',*2401/
0010 DATA LEOP/'177777/
0011 DATA LHOME,ICRLF,IDC1/'24,*106412,*10400/
0012 DATA SKB/' /
0013 DATA LFF4,LFF5/'14,*5000/
0014 DATA LF/'12/
0015 DATA ILO/'51504/
0016 DATA IH15/'405/
0017 DATA LSK,MSK/'177,*77400/
0018 DATA IC9H,IHC9/'4405,*2411/
0019 DATA IC4H,IHC4,IC8H,IHC8/'2005,*2404,*4005,*2410/
0020 15 FORMAT(1X,10A8)
0021 IST=1
0022 DO 130 J=1,129
0023 DO 140 I=1,70
0024 140 CONTINUE
0025 130 K=DEVST(1)
0026 I1ST=0
0027 ICFLG=0
0028 ICON(1)=258
0029 DO 22 J=1,11
0030 DO 22 I=1,10
0031 22 MOUT(I,J)=SKB
0032 CALL ASSIGN(1,'TT1:')
0033 CALL ASSIGN(2,'EFDIR')
0034 DEFINE FILE 2(10,100,U,I1)
0035 CALL ASSIGN(3,'EFLOCF')
0036 DEFINE FILE 3(100,120,U,I2)
0037 CALL ASSIGN(4,'EFCKTD')
0038 DEFINE FILE 4(100,120,U,I3)
0039 CALL ASSIGN(5,'EFTRKD')
0040 DEFINE FILE 5(100,120,U,I4)
0041 CALL ASSIGN(6,'EFTERD')
0042 DEFINE FILE 6(100,120,U,I5)
0043 CALL ASSIGN(7,'INFO.DAT')
0044 DEFINE FILE 7(396,40,U,I6)
0045 CALL ASSIGN(8,'MSG.DAT')
0046 DEFINE FILE 8(211,40,U,I16)
0047 DEVST(5)=0
0048 25 CALL RDLOOP
0049 WRITE(1,16)(ICODE(J),J=1,4)
0050 16 FORMAT(1X,'HEADER=',408)
0051 IF (ICODE(3) .EQ. IC4H) ICODE(3)=IHC4
0053 IF (ICODE(3) .EQ. IC8H) ICODE(3)=IHC8
0055 IF (ICODE(3) .EQ. IC9H) ICODE(3)=IHC9
0057 IF (ICODE(3) .EQ. IH15) GOTO 700
0059 IF (ICODE(4) .NE. ILO) GOTO 630
0061 NRCNO=21
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]FRI 18-MAR-77 11:24:14 PAGE 002
RCMV5.OBJ=RCMV5.FOR/NOSN/LI:1

```
0062        NOREC=1
0063        IST=9999
0064        GOTO 12
0065      630 IF (IST .EQ. 1) GOTO 51
0067        IF (IST .EQ. 2) GOTO 52
0069        IF (IST .EQ. 3) GOTO 53
0071        IF (IST .EQ. 4) GOTO 54
0073      51 NRCNO=1
0074        NOREC=2
0075        IST=2
0076        GOTO 12
0077      52 NRCNO=127
0078        NOREC=6
0079        IST=3
0080        GOTO 12
0081      53 IFS=IAND(LSK,ICODE(4))
0082        IFS=IFS-48
0083        IF (IFS .GE. 5) IFS=4
0085        IF (IFS .GE. 1) GOTO 90
0087      99 NRCNO=11
0088        NOREC=1
0089        GOTO 12
0090      90 NRCNO=157
0091        NOREC=1
0092        IST=4
0093        GOTO 12
0094      54 KEY=IAND(KYMSK,ICODE(4))
0095        READ(2'IFS,ERR=99)(DI(I),I=1,100)
0096        DO 550 K=1,100
0097        KK=K
0098        IF (DI(K) .EQ. 0) GOTO 565
0100        IF (KEY .EQ. DI(K)) GOTO 560
0102      550 CONTINUE
0103      565        NRCNO=210
0104        NOREC=1
0105        IST=2
0106        GOTO 12
0107      560        LU=IFS+2
0108        READ(LU'KK,ERR=99)((OFIL(I,J),I=1,10),J=1,3)
0109        DO 582 J=1,3
0110        DO 582 I=1,10
0111      582        MOUT(I,J)=OFIL(I,J)
0112        MOUT(1,5)=XMT
0113        DO 5100 J=1,5
0114      5100        WRITE(8'J+33)(MOUT(I,J),I=1,10)
0115        NRCNO=34
0116        NOREC=5
0117        IST=2
0118        DI(KK)=-DI(KK)
0119        WRITE(2'IFS,ERR=99)(DI(I),I=1,100)
0120        ICFLG=1
0121        ICON(1)=ICON(1)+ISHFT(1,8)
0122        ICON(2)=0
0123        ICON(3)=IH51
0124        ICON(4)=IFS
0125        ICON(5)=-DI(KK)
```

FORTRAN IV V01B-02
CORE=08K, UIC=C20,20J

FRI 18-MAR-77 11:24:14

PAGE 003

RCMV5.OBJ=RCMV5.FOR/NOSN/LI:1

```
0126      DO 600 I=1,10
0127      600 IFAR(I)=OFIL(I,1)
0128      DO 610 J=1,40
0129      610 ICON(J+5)=LINE(J)
0130      ICON(46)=LEOP
0131      GOTO 12
0132      700 IFSH=ICODE(4)
0133      KEYH=ICODE(5)
0134      READ(2'IFSH,ERR=99)(DI(I),I=1,100)
0135      DO 1055 K=1,99
0136      KKH=K
0137      IF (KEYH .EQ. -DI(K)) GOTO 1060
0139      IF (KEYH .EQ. DI(K)) GOTO 1060
0141      IF (DI(K) .EQ. 0) GOTO 1070
0143      1055 CONTINUE
0144      1070 DI(KKH+1)=0
0145      1060 DI(KKH)=KEYH
0146      LU=IFSH+2
0147      READ(LU'1,ERR=99)((OFIL(I,J),I=1,10),J=1,3)
0148      DO 300 J=1,40
0149      300 LINE(J)=ICODE(J+5)
0150      DO 305 I=1,10
0151      305 OFIL(I,1)=IFAR(I)
0152      DO 1050 I=1,10
0153      DO 1050 J=2,3
0154      1050 OFIL(I,J)=MVD
0155      WRITE(2'IFSH,ERR=99)(DI(I),I=1,100)
0156      WRITE(LU'KKH,ERR=99)((OFIL(I,J),I=1,10),J=1,3)
0157      GOTO 25
0158      12 CONTINUE
C      WRITE TO LOOP
0159      IF (NRCNO .GE. 107) NRCNO=NRCNO+1
0161      DO 200 J=4,128
0162      200 ICODE(J)=0
0163      ICODE(127)=LEOP
0164      IF (NRCNO .NE. 11) GOTO 210
0166      ICODE(4)=LHOME
0167      DO 220 J=5,16
0168      220 ICODE(J)=LF
0169      READ(8'11)(ICODE(I),I=17,56)
0170      ICODE(57)=LHOME
0171      ICODE(58)=LEOP
0172      CALL WRLOOP(I1ST)
0173      GOTO 330
0174      210 NWRTS=4
0175      IF (NOREC .LE. 3) NWRTS=1
0177      IF (NOREC .GE. 4 .AND. NOREC .LE. 6) NWRTS=2
0179      IF (NOREC .GE. 7 .AND. NOREC .LE. 9) NWRTS=3
0181      NN=NRCNO
0182      DO 310 J=1,NWRTS
0183      JJJ=J
0184      DO 450 K=4,126
0185      450 ICODE(K)=0
0186      IF (J .EQ. 1) ICODE(4)=LFF4
0188      IF (J .EQ. 1) ICODE(5)=LFF5
0190      READ(8'NN,ERR=500)(ICODE(I),I=6,45)
```

FORTRAN IV V01B-02
CORE=0BK, UIC=[20,20]

FRI 18-MAR-77 11:24:14

PAGE 004

RCMV5.OBJ=RCMV5.FOR/NOSN/LI:1

```
0191      N1=(J-1)*3+1
0192      IF (NOREC .LE. N1) GOTO 250
0194      NN=NN+1
0195      READ(8'NN,ERR=500)(ICODE(I),I=46,85)
0196      N2=N1+1
0197      IF (NOREC .LE. N2) GOTO 250
0199      NN=NN+1
0200      READ(8'NN,ERR=500)(ICODE(I),I=86,125)
0201      NN=NN+1
0202 250  ICODE(45)=ICRLF
0203      ICODE(85)=ICRLF
0204      ICODE(125)=ICRLF
0205      ICODE(126)=IDC1
0206      IF (J .EQ. NWRTS) ICODE(126)=LHOME
0208      CALL WRLOOP(I1ST)
0209      I1ST=1
0210 310  CONTINUE
0211      IF (ICFLG .EQ. 0) GOTO 330
0213      DO 510 J=1,128
0214 510  ICODE(J)=ICON(J)
0215      CALL WRLOOP(I1ST)
0216      ICFLG=0
0217 330  CONTINUE
C      WRITE OUT SCREEN
0218      DO 70 J=1,NOREC
0219      NOR=NRCNO+J-1
0220      READ(8'NOR)(MOUT(I,J),I=1,10)
0221 70  WRITE(1,15)(MOUT(I,J),I=1,10)
0222      DO 80 J=1,11
0223      DO 80 I=1,10
0224 80  MOUT(I,J)=0
0225 500  CONTINUE
0226      IF (IST .NE. 9999) GOTO 25
0228      END
```

>

FORTRAN IV V01R-02
 CORE=08K, UIC=[20,20]

FRI 18-MAR-77 11:14:18

PAGE 001

RCMV1.OBJ=RCMV1.FOR/NOSN/LI:1

```

0001     REAL*8 MOUT,OFIL,IFAR,MVD,XMT,SKB
0002     INTEGER DEVST(6)
0003     INTEGER DI
0004     COMMON /LOOP/ICODE(128),MSK,LSK
0005     COMMON /M1710/DEVST
0006     DIMENSION OFIL(10,3),DI(100),LINE(40)
0007     DIMENSION IFAR(10),ICON(128),MOUT(10,11)
0008     EQUIVALENCE(LINE,IFAR)
0009     DATA MVD,KYMSK,XMT,IH51/' MOVED ', '77577','PRES XMT','2401/
0010     DATA LEOP/'177777/
0011     DATA LHOME,ICRLF,IDC1/'24','106412','10400/
0012     DATA SK8/' '/
0013     DATA LFF4,LFF5/'14','5000/
0014     DATA LF/'12/
0015     DATA ILO/'51504/
0016     DATA IH15/'405/
0017     DATA LSK,MSK/'177','77400/
0018     DATA IC9H,IHC9/'4401','411/
0019     DATA IC4H,IHC4,IC8H,IHC8/'2001','404','4001','410/
0020     15  FORMAT(1X,10A8)
0021     IST=1
0022     DO 130 J=1,129
0023     DO 140 I=1,70
0024     140 CONTINUE
0025     130 K=DEVST(1)
0026     I1ST=0
0027     ICFLG=0
0028     ICON(1)=258
0029     DO 22 J=1,11
0030     DO 22 I=1,10
0031     22 MOUT(I,J)=SK8
0032     CALL ASSIGN(1,'TTO:')
0033     CALL ASSIGN(2,'EFDIR')
0034     DEFINE FILE 2(10,100,U,I1)
0035     CALL ASSIGN(3,'EFLOCF')
0036     DEFINE FILE 3(100,120,U,I2)
0037     CALL ASSIGN(4,'EFCKTB')
0038     DEFINE FILE 4(100,120,U,I3)
0039     CALL ASSIGN(5,'EFTRKD')
0040     DEFINE FILE 5(100,120,U,I4)
0041     CALL ASSIGN(6,'EFTERD')
0042     DEFINE FILE 6(100,120,U,I5)
0043     CALL ASSIGN(7,'INFO.DAT')
0044     DEFINE FILE 7(396,40,U,I6)
0045     CALL ASSIGN(8,'MSG.DAT')
0046     DEFINE FILE 8(211,40,U,I16)
0047     DEVST(5)=0
0048     25  CALL RDLOOP
0049     WRITE(1,16)(ICODE(J),J=1,4)
0050     16  FORMAT(1X,'HEADER=',408)
0051     IF (ICODE(3) .EQ. IC4H) ICODE(3)=IHC4
0053     IF (ICODE(3) .EQ. IC8H) ICODE(3)=IHC8
0055     IF (ICODE(3) .EQ. IC9H) ICODE(3)=IHC9
0057     IF (ICODE(3) .EQ. IH51) GOTO 700
0059     IF (ICODE(4) .NE. ILO) GOTO 630
0061     NRCNO=21

```

FORTRAN IV V01B-02
CORE=08K, UIC=C20,20JFRI 18-MAR-77 11:14:18 PAGE 002
RCMV1.OBJ=RCMV1.FOR/NOSN/LI:1

```
0062        NOREC=1
0063        IST=9999
0064        GOTO 12
0065        630 IF (IST .EQ. 1) GOTO 51
0067        IF (IST .EQ. 2) GOTO 52
0069        IF (IST .EQ. 3) GOTO 53
0071        IF (IST .EQ. 4) GOTO 54
0073        51 NRCNO=1
0074        NOREC=2
0075        IST=2
0076        GOTO 12
0077        52 NRCNO=127
0078        NOREC=6
0079        IST=3
0080        GOTO 12
0081        53 IFS=IAND(LSK,ICODE(4))
0082        IFS=IFS-48
0083        IF (IFS .GE. 5) IFS=4
0085        IF (IFS .GE. 1) GOTO 90
0087        99 NRCNO=11
0088        NOREC=1
0089        GOTO 12
0090        90 NRCNO=157
0091        NOREC=1
0092        IST=4
0093        GOTO 12
0094        54 KEY=IAND(KYMSK,ICODE(4))
0095        READ(2'IFS,ERR=99')(DI(I),I=1,100)
0096        DO 550 K=1,100
0097        KK=K
0098        IF (DI(K) .EQ. 0) GOTO 565
0100        IF (KEY .EQ. DI(K)) GOTO 560
0102        550 CONTINUE
0103        565        NRCNO=210
0104        NOREC=1
0105        IST=2
0106        GOTO 12
0107        560        LU=IFS+2
0108        READ(LU'KK,ERR=99')((OFIL(I,J),I=1,10),J=1,3)
0109        DO 582 J=1,3
0110        DO 582 I=1,10
0111        582        MOUT(I,J)=OFIL(I,J)
0112        MOUT(1,5)=XMT
0113        DO 5100 J=1,5
0114        5100 WRITE(8'J+33')(MOUT(I,J),I=1,10)
0115        NRCNO=34
0116        NOREC=5
0117        IST=2
0118        DI(KK)=-DI(KK)
0119        WRITE(2'IFS,ERR=99')(DI(I),I=1,100)
0120        ICFLG=1
0121        ICON(1)=ICON(1)+ISHFT(1,8)
0122        ICON(2)=0
0123        ICON(3)=IH15
0124        ICON(4)=IFS
0125        ICON(5)=-DI(KK)
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 11:14:18

PAGE 003

RCMV1.OBJ=RCMV1.FOR/NOSN/LI:1

```
0126      DO 600 I=1,10
0127      600 IFAR(I)=OFIL(I,1)
0128      DO 610 J=1,40
0129      610 ICON(J+5)=LINE(J)
0130      ICON(46)=LEOF
0131      GOTO 12
0132      700 IFSH=ICODE(4)
0133      KEYH=ICODE(5)
0134      READ(2'IFSH,ERR=99)(DI(I),I=1,100)
0135      DO 1055 K=1,99
0136      KKH=K
0137      IF (KEYH .EQ. -DI(K)) GOTO 1060
0139      IF (KEYH .EQ. DI(K)) GOTO 1060
0141      IF (DI(K) .EQ. 0) GOTO 1070
0143      1055 CONTINUE
0144      1070 DI(KKH+1)=0
0145      1060 DI(KKH)=KEYH
0146      LU=IFSH+2
0147      READ(LU'1,ERR=99)((OFIL(I,J),I=1,10),J=1,3)
0148      DO 300 J=1,40
0149      300 LINE(J)=ICODE(J+5)
0150      DO 305 I=1,10
0151      305 OFIL(I,1)=IFAR(I)
0152      DO 1050 I=1,10
0153      DO 1050 J=2,3
0154      1050 OFIL(I,J)=MVD
0155      WRITE(2'IFSH,ERR=99)(DI(I),I=1,100)
0156      WRITE(LU'KKH,ERR=99)((OFIL(I,J),I=1,10),J=1,3)
0157      GOTO 25
0158      12 CONTINUE
C      WRITE TO LOOP
0159      IF (NRCNO .GE. 107) NRCNO=NRCNO+1
0161      DO 200 J=4,128
0162      200 ICODE(J)=0
0163      ICODE(127)=LEOF
0164      IF (NRCNO .NE. 11) GOTO 210
0166      ICODE(4)=LHOME
0167      DO 220 J=5,16
0168      220 ICODE(J)=LF
0169      READ(8'11)(ICODE(I),I=17,56)
0170      ICODE(57)=LHOME
0171      ICODE(58)=LEOF
0172      CALL WRLOOP(I1ST)
0173      GOTO 330
0174      210 NWRTS=4
0175      IF (NOREC .LE. 3) NWRTS=1
0177      IF (NOREC .GE. 4 .AND. NOREC .LE. 6) NWRTS=2
0179      IF (NOREC .GE. 7 .AND. NOREC .LE. 9) NWRTS=3
0181      NN=NRCNO
0182      DO 310 J=1,NWRTS
0183      JJJ=J
0184      DO 450 K=4,126
0185      450 ICODE(K)=0
0186      IF (J .EQ. 1) ICODE(4)=LFF4
0188      IF (J .EQ. 1) ICODE(5)=LFF5
0190      READ(8'NN,ERR=500)(ICODE(I),I=6,45)
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 11:14:18

PAGE 004

RCMV1.OBJ=RCMV1.FOR/NOSN/LI:1

```
0191            N1=(J-1)*3+1
0192            IF (NOREC .LE. N1) GOTO 250
0194            NN=NN+1
0195            READ(8'NN,ERR=500)(ICODE(I),I=46,85)
0196            N2=N1+1
0197            IF (NOREC .LE. N2) GOTO 250
0199            NN=NN+1
0200            READ(8'NN,ERR=500)(ICODE(I),I=86,125)
0201            NN=NN+1
0202            250 ICODE(45)=ICRLF
0203            ICODE(85)=ICRLF
0204            ICODE(125)=ICRLF
0205            ICODE(126)=IDC1
0206            IF (J .EQ. NWRTS) ICODE(126)=LHOME
0208            CALL WRLOOP(I1ST)
0209            I1ST=1
0210            310 CONTINUE
0211            IF (ICFLG .EQ. 0) GOTO 330
0213            DO 510 J=1,128
0214            510 ICODE(J)=ICON(J)
0215            CALL WRLOOP(I1ST)
0216            ICFLG=0
0217            330 CONTINUE
          C        WRITE OUT SCREEN
0218            DO 70 J=1,NOREC
0219            NOR=NRCNO+J-1
0220            READ(8'NOR)(MOUT(I,J),I=1,10)
0221            70 WRITE(1,15)(MOUT(I,J),I=1,10)
0222            DO 80 J=1,11
0223            DO 80 I=1,10
0224            80 MOUT(I,J)=0
0225            500 CONTINUE
0226            IF (IST .NE. 9999) GOTO 25
0228            END
```

>

1.6 Interprocess Communication and Resource Sharing Demonstration Program

This program is used to illustrate how the ESM may be used to perform interprocess communication and resource sharing experiments. After LOGON, the Logical ID/Functional Address (LID/FAD) Table of LID's 1-100 is displayed for the dialogue director host on an ESM terminal. The same table for the other host is then displayed after the operator transmits a character(s). After the next CRT transmission two new logical ID's are created in the system. Logical ID's 81 and 82 are created such that 81 is owned by processor B loop 2 (FAD 2) and 82 is owned by processor A loop 1 (FAD 1). Special control packets are sent to the two host nodes and two gateway nodes connecting loops 1 and 2, and the system control disk file (INFO.DAT) is updated to reflect the change. After all tables are updated and a system lock is terminated the updated LID/FAD table for the dialogue director host is displayed. After the next CRT transmission the updated table for the other host is displayed. After the next CRT transmission, the dialogue director host sends a bid request to the other host using the new LID's. The other host responds to the bid and the dialogue director host displays the message BID followed by a request for a message. When the other host node is placed in a do not execute state the dialogue director performs a 20 second timeout and responds with NC BID. If the bid has been acknowledged a message entered on the ESM terminal and transmitted is displayed on the other host's DECSCOPE implemented by host-host messages using the new LID's. The tables are then returned to their original state and displayed thus destroying the LID's that were created for the interprocess communication. Entering "DS" at an ESM terminal terminates the program.

The program exists in two forms, PROC1 for host processor A, and PROC5 for host processor B. ESM Tape #1 contains the source (.FOR), object (.OBJ), task (.TSK), and overlay description language files (.ODL) for the program. The overlay structure consists of the main program and two overlaid routines RDLOOP and WRLOOP which are listed in Section 1.3.

The Task Builder (TKB) commands for the program are:

```
TKB PROC5.TSK=PROC5.ODL/MP, [1,1] SYSLIB/LB:$SHORT.
```

Options include:

```
UNITS=3  
ACTFIL=3  
MAXBUF=80  
COMMON=M1710:RW  
ASG=TI:1, SYO:2:3
```

Both host processors must be running the demonstration program for interprocess communication. Before running the program the STESM indirect command file must be activated (by entering @ STESM at the DECSCOPE) and the ESM loops cleared to initialize the system control file (INFO.DAT) and nodal LID/FAD tables.

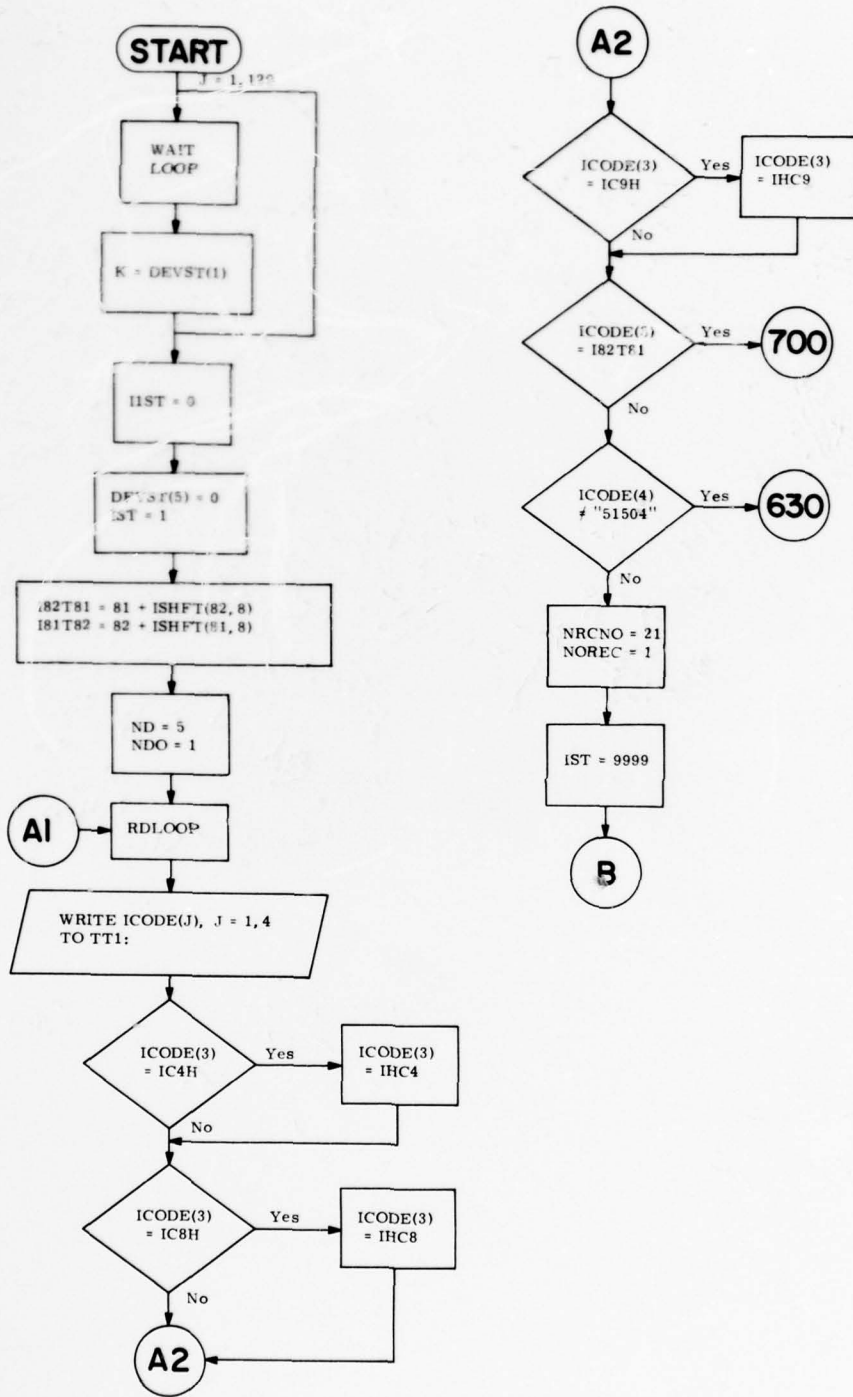


Figure 1-22. PROC5

PROC5 (cont.)

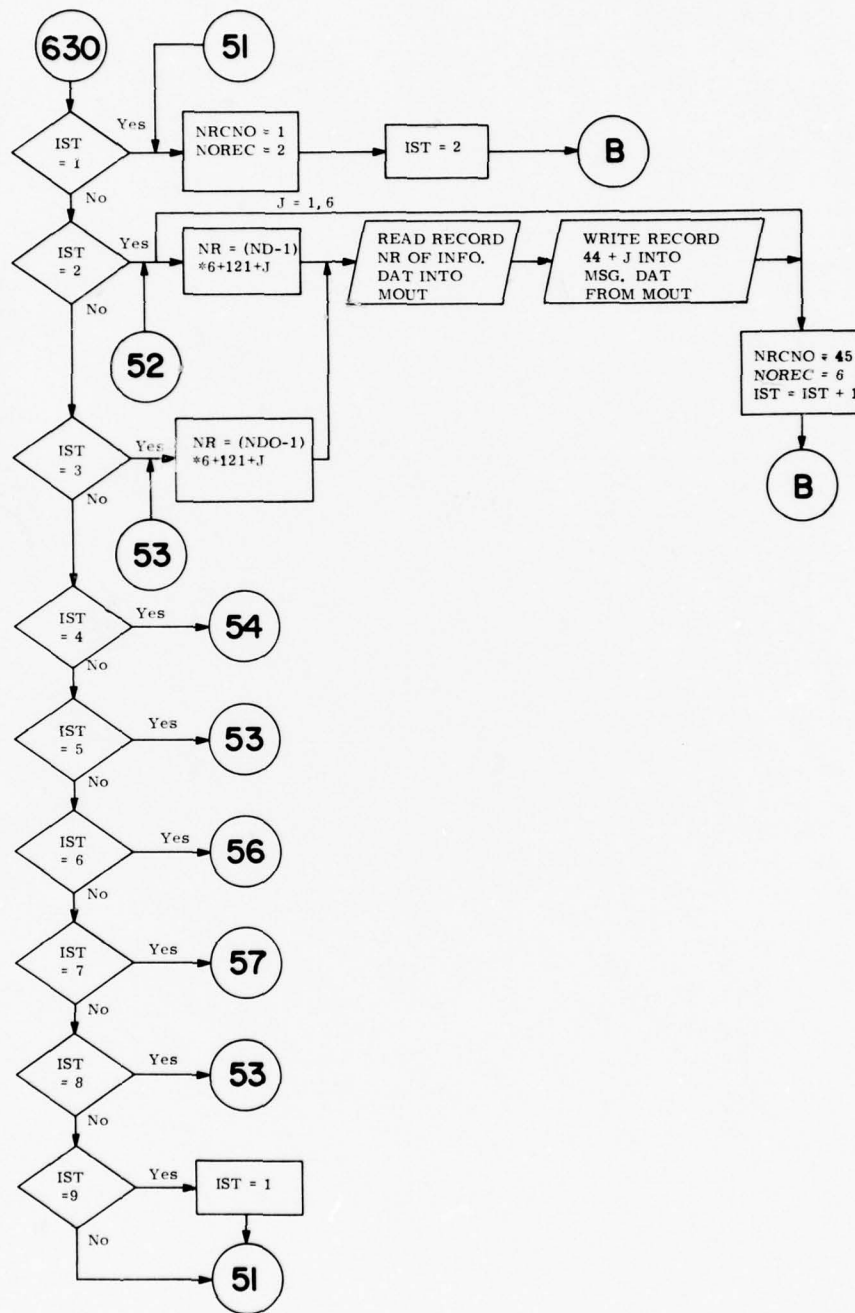


Figure 1-22. (Cont.)

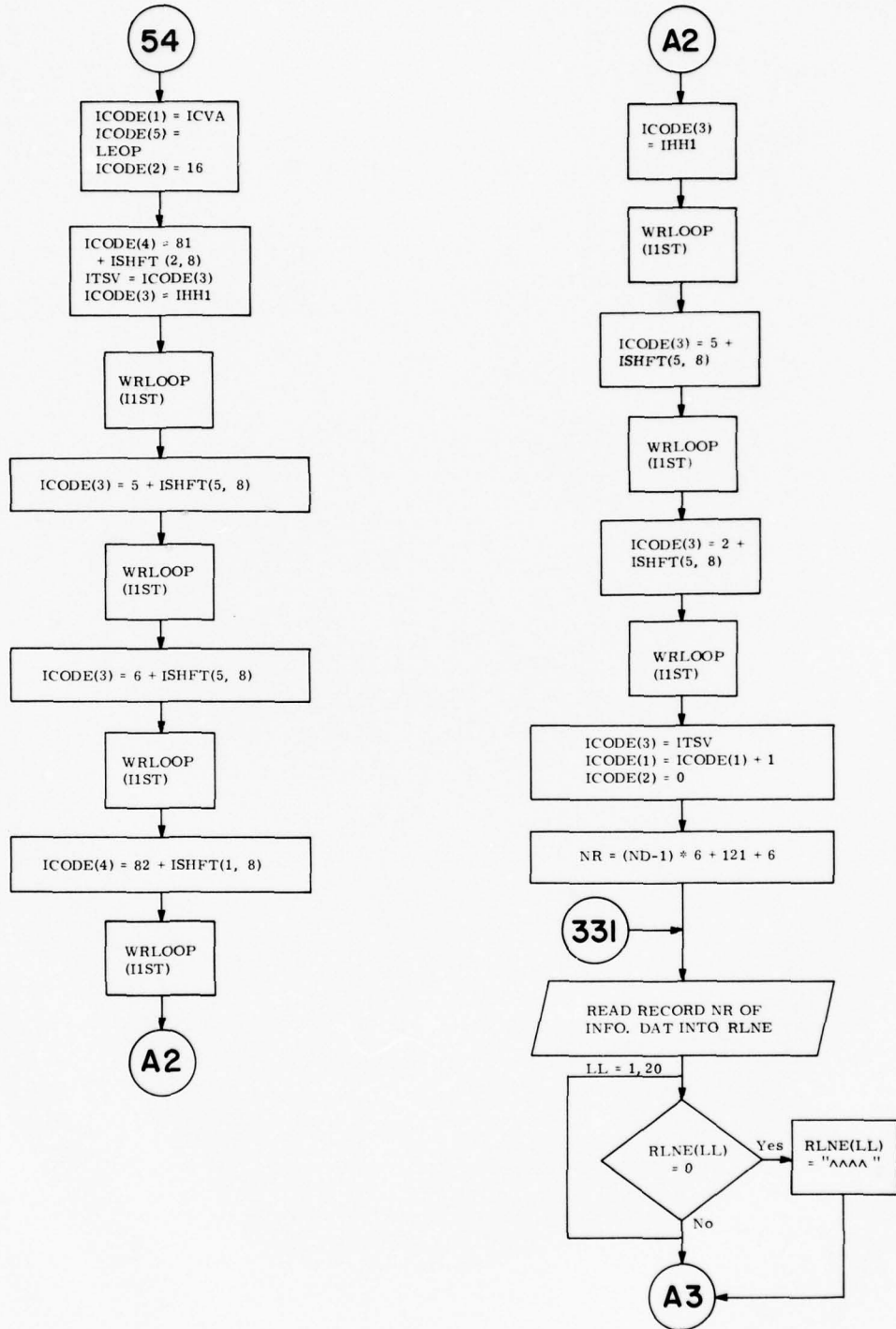


Figure 1-22. (Cont.)

PROC5 (cont.)

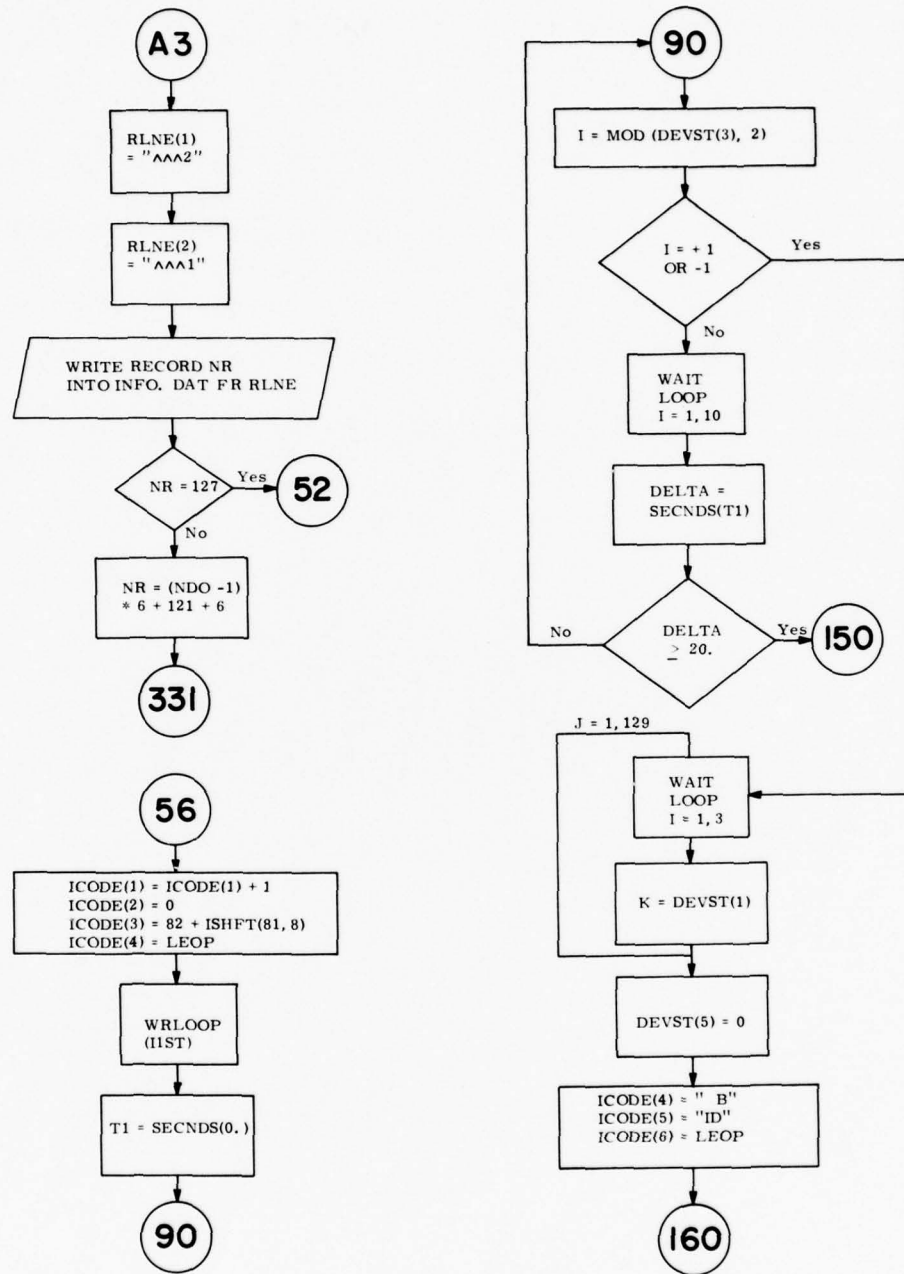


Figure 1-22. (Cont.)

PROC5 (cont.)

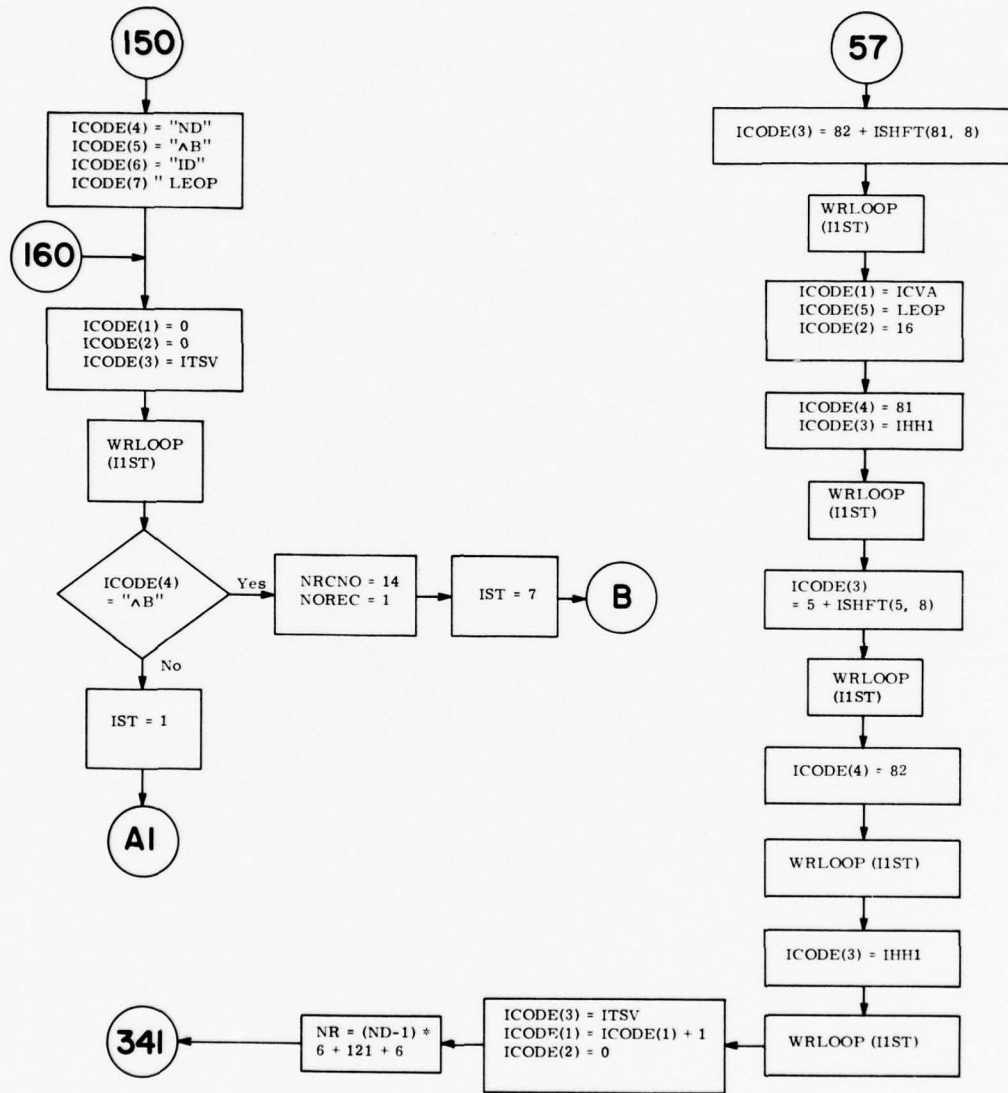


Figure 1-22. (Cont.)

PROC5 (cont.)

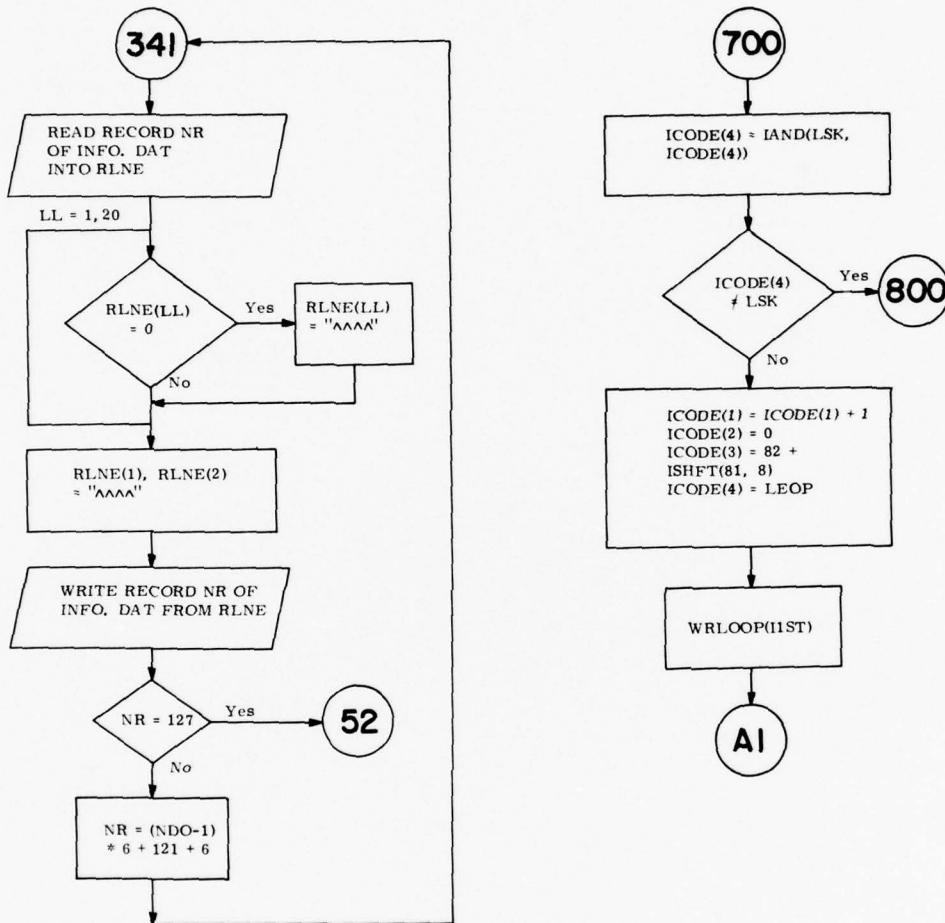


Figure 1-22. (Cont.)

PROC5 (cont.)

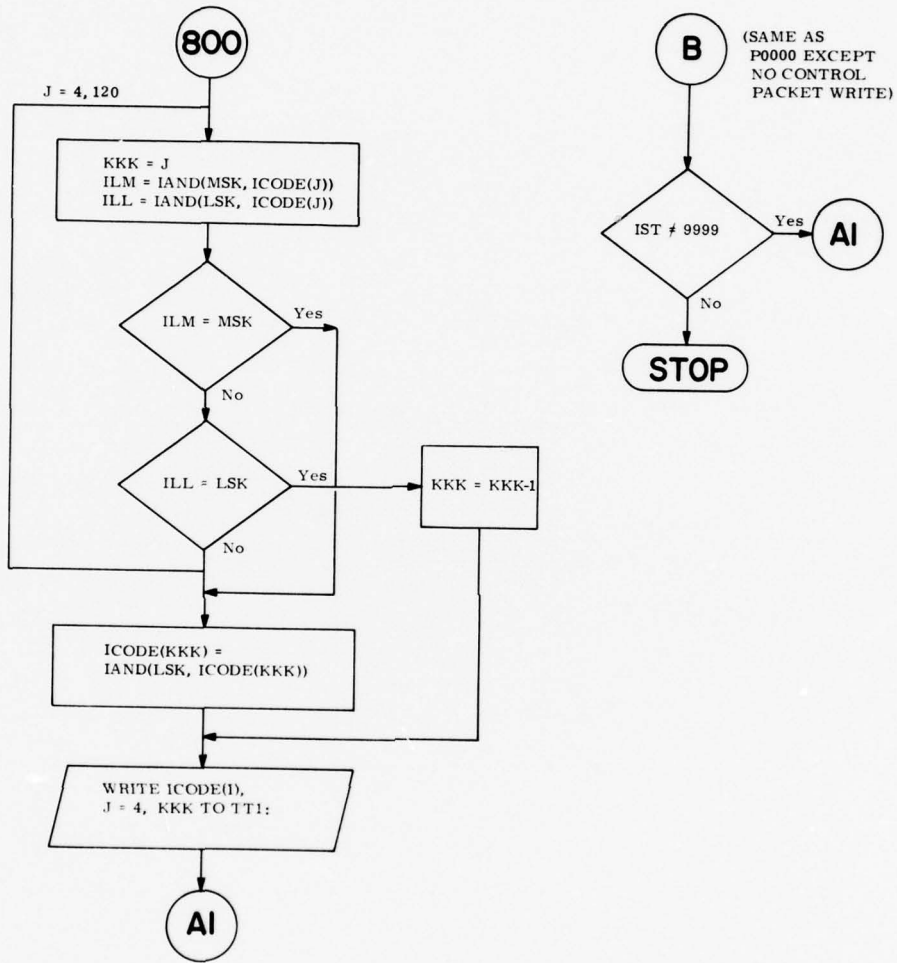


Figure 1-22. (Cont.)

FORTRAN IV V01B-02
 CORE=0BK, UIC=E20,20J

FRI 18-MAR-77 13:01:02 PAGE 001
 PROC5.OBJ=PROC5.FOR/NOSN/LI:1

```

C      INTERPROCESS COMM & RESOURCE SHARING DEMO
C
0001      INTEGER DEVST(6)
0002      COMMON /LOOP/ICODE(128),MSK,LSK
0003      COMMON /M1710/DEVST
0004      REAL*4 RONE,RTWO,RLNE(20),RSPC
0005      REAL*8 MOUT(10,11)
0006      DATA IH1H,IHH1/'405','2401/
0007      DATA IC9H,IHC9/'4405','2411/
0008      DATA IC4H,IHC4,IC8H,IHC8/'2005','2404','4005','2410/
0009      DATA ILO,ICVA,LEOP/'51504','125125','177777/
0010      DATA RONE,RTWO,RSPC/' 1',' 2','  ' /
0011      DATA NO,IB,ID/'NO',' B','ID'/
0012      DATA LSK/'377/
0013      DATA LHOME,ICRLF,IDC1/'24','106412','10400/
0014      DATA LFF4,LFF5/'14','5000/
0015      DATA LF/'12/
0016      DATA MSK/'177400/
0017      15  FORMAT(1X,10A8)
0018      DO 130 J=1,129
0019      DO 140 I=1,70
0020      140 CONTINUE
0021      130 K=DEVST(1)
0022      IIST=0
0023      CALL ASSIGN(1,'TT1:')
0024      CALL ASSIGN(2,'INFO.DAT')
0025      DEFINE FILE 2(396,40,U,I1)
0026      CALL ASSIGN(3,'MSG.DAT')
0027      DEFINE FILE 3(211,40,U,I2)
0028      DEVST(5)=0
0029      IST=1
0030      I82T81=81+ISHFT(82,8)
0031      I81T82=82+ISHFT(81,8)
0032      ND=5
0033      NDO=1
0034      25 CALL RDLLOOP
0035      WRITE(1,16)(ICODE(J),J=1,4)
0036      16  FORMAT(1X,'HEADER=',408)
0037      IF (ICODE(3) .EQ. IC4H) ICODE(3)=IHC4
0039      IF (ICODE(3) .EQ. IC8H) ICODE(3)=IHC8
0041      IF (ICODE(3) .EQ. IC9H) ICODE(3)=IHC9
0043      IF (ICODE(3) .EQ. I82T81) GOTO 700
0045      IF (ICODE(4) .NE. ILO) GOTO 630
0047      NRCND=21
0048      NOREC=1
0049      IST=9999
0050      GOTO 12
0051      630 IF (IST .EQ. 1) GOTO 51
0053      IF (IST .EQ. 2) GOTO 52
0055      IF (IST .EQ. 3) GOTO 53
0057      IF (IST .EQ. 4) GOTO 54
0059      IF (IST .EQ. 5) GOTO 55
0061      IF (IST .EQ. 6) GOTO 56
0063      IF (IST .EQ. 7) GOTO 57
0065      IF (IST .EQ. 8) GOTO 58
0067      IF (IST .EQ. 9) IST=1

```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]FRI 18-MAR-77 13:01:02 PAGE 002
PROC5.OBJ=PROC5.FOR/NOSN/LI:1

```
0069      GOTO 51
0070      51 NRCNO=1
0071          NOREC=2
0072          IST=2
0073      GOTO 12
0074      52 DO 362 J=1,6
0075          NR=(ND-1)*6+121+J
0076          READ(2'NR')(MOUT(I,J),I=1,10)
0077      362 WRITE(3'44+J')(MOUT(I,J),I=1,10)
0078          NRCNO=45
0079          NOREC=6
0080          IST=IST+1
0081          GOTO 12
0082      53 DO 363 J=1,6
0083          NR=(NDO-1)*6+121+J
0084          READ(2'NR')(MOUT(I,J),I=1,10)
0085      363 WRITE(3'44+J')(MOUT(I,J),I=1,10)
0086          NRCNO=45
0087          NOREC=6
0088          IST=IST+1
0089          GOTO 12
0090      54 ICODE(1)=ICVA
0091          ICODE(5)=LEOP
0092          ICODE(2)=16
0093          ICODE(4)=81+ISHFT(2,8)
0094          ITSV=ICODÉ(3)
0095          ICODE(3)=IHH1
0096          CALL WRLOOP(I1ST)
0097          ICODE(3)=5+ISHFT(5,8)
0098          CALL WRLOOP(I1ST)
0099          ICODE(3)=6+ISHFT(5,8)
0100          CALL WRLOOP(I1ST)
0101          ICODE(4)=82+ISHFT(1,8)
0102          CALL WRLOOP(I1ST)
0103          ICODE(3)=IHH1
0104          CALL WRLOOP(I1ST)
0105          ICODE(3)=5+ISHFT(5,8)
0106          CALL WRLOOP(I1ST)
0107          ICODE(3)=2+ISHFT(5,8)
0108          CALL WRLOOP(I1ST)
0109          ICODE(3)=ITSV
0110          ICODE(1)=ICODÉ(1)+1
0111          ICODE(2)=0
0112          NR=(ND-1)*6+121+6
0113      331 READ(2'NR')(RLNE(I),I=1,20)
0114          DO 332 LL=1,20
0115      332 IF (RLNE(LL) .EQ. 0) RLNE(LL)=RSFC
0116          RLNE(1)=RTWO
0117          RLNE(2)=RONE
0118          WRITE(2'NR')(RLNE(I),I=1,20)
0119          IF (NR .EQ. 127) GOTO 52
0120          NR=(NDO-1)*6+121+6
0121          GOTO 331
0122      55 GOTO 53
0123      56 ICODE(1)=ICODÉ(1)+1
0124          ICODE(2)=0
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 13:01:02

PAGE 003

PROC5.OBJ=PROC5.FOR/NOSN/LI:1

```
0127      ICODE(3)=82+ISHFT(81,8)
0128      ICODE(4)=LEOF
0129      CALL WRLOOP(I1ST)
0130      T1=SECNDS(0.)
0131      90 I=MOD(DEVST(3),2)
0132      IF (I .EQ. 1) GOTO 120
0134      IF (I .EQ. -1) GOTO 120
0136      DO 110 I=1,10
0137      110 CONTINUE
0138      DELTA=SECNDS(T1)
0139      IF (DELTA .GE. 20.) GOTO 150
0141      GOTO 90
0142      120 DO 131 J=1,129
0143      DO 141 I=1,3
0144      141 CONTINUE
0145      K=DEVST(1)
0146      131 CONTINUE
0147      DEVST(5)=0
0148      ICODE(4)=IB
0149      ICODE(5)=ID
0150      ICODE(6)=LEOF
0151      GOTO 160
0152      150 ICODE(4)=NO
0153      ICODE(5)=IB
0154      ICODE(6)=ID
0155      ICODE(7)=LEOF
0156      160 ICODE(1)=0
0157      ICODE(2)=0
0158      ICODE(3)=ITSV
0159      CALL WRLOOP(I1ST)
0160      IF (ICODE(4) .EQ. IB) GOTO 170.
0162      IST=1
0163      GOTO 25
0164      170 NRCNO=14
0165      NOREC=1
0166      IST=7
0167      GOTO 12
0168      57 ICODE(3)=82+ISHFT(81,8)
0169      CALL WRLOOP(I1ST)
0170      ICODE(1)=ICVA
0171      ICODE(5)=LEOF
0172      ICODE(2)=16
0173      ICODE(4)=81
0174      ICODE(3)=IHH1
0175      CALL WRLOOP(I1ST)
0176      ICODE(3)=5+ISHFT(5,8)
0177      CALL WRLOOP(I1ST)
0178      ICODE(4)=82
0179      CALL WRLOOP(I1ST)
0180      ICODE(3)=IHH1
0181      CALL WRLOOP(I1ST)
0182      ICODE(3)=ITSV
0183      ICODE(1)=ICODE(1)+1
0184      ICODE(2)=0
0185      NR=(NR-1)*6+121+6
0186      341 READ(2,NR)(RLNE(I),I=1,20)
```

FORTRAN IV V01B-02
CORE=0BK, UIC=C20,20J

FRI 18-MAR-77 13:01:02

PAGE 004

PROC5.OBJ=PROC5.FOR/NOSN/LI:1

```
0187 DO 342 LL=1,20
0188 342 IF (RLNE(LL) .EQ. 0) RLNE(LL)=RSFC
0190 RLNE(1)=RSFC
0191 RLNE(2)=RSFC
0192 WRITE(2'NR')(RLNE(I),I=1,20)
0193 IF (NR .EQ. 127) GOTO 52
0195 NR=(NDO-1)*6+121+6
0196 GOTO 341
0197 58 GOTO 53
0198 700 ICODE(4)=IAND(LSK,ICODE(4))
0199 IF (ICODE(4) .NE. LSK) GOTO 800
0201 ICODE(1)=ICODE(1)+1
0202 ICODE(2)=0
0203 ICODE(3)=82+ISHFT(81,8)
0204 ICODE(4)=LEOP
0205 CALL WRLOOP(I1ST)
0206 GOTO 25
0207 800 DO 810 J=4,120
0208 KKK=J
0209 ILM=IAND(MSK,ICODE(J))
0210 ILL=IAND(LSK,ICODE(J))
0211 IF (ILM .EQ. MSK) GOTO 820
0213 810 IF (ILL .EQ. LSK) GOTO 830
0215 820 ICODE(KKK)=IAND(LSK,ICODE(KKK))
0216 GOTO 840
0217 830 KKK=KKK-1
0218 840 WRITE(1,20)(ICODE(J),J=4,KKK)
0219 20 FORMAT(1X,39A2/1X,39A2/1X,39A2/)
0220 GOTO 25
0221 12 CONTINUE
0222 IF (NRCNO .GE. 107) NRCNO=NRCNO+1
0224 DO 200 J=4,128
0225 200 ICODE(J)=0
0226 ICODE(127)=LEOP
0227 IF (NRCNO .NE. 11) GOTO 210
0229 ICODE(4)=LHOME
0230 DO 220 J=5,16
0231 220 ICODE(J)=LF
0232 READ(8'11)(ICODE(I),I=17,56)
0233 ICODE(57)=LHOME
0234 ICODE(58)=LEOP
0235 CALL WRLOOP(I1ST)
0236 GOTO 330
0237 210 NWRTS=4
0238 IF (NOREC .LE. 3) NWRTS=1
0240 IF (NOREC .GE. 4 .AND. NOREC .LE. 6) NWRTS=2
0242 IF (NOREC .GE. 7 .AND. NOREC .LE. 9) NWRTS=3
0244 NN=NRCNO
0245 DO 310 J=1,NWRTS
0246 JJJ=J
0247 DO 450 K=4,126
0248 450 ICODE(K)=0
0249 IF (J .EQ. 1) ICODE(4)=LFF4
0251 IF (J .EQ. 1) ICODE(5)=LFF5
0253 READ(3'NN,ERR=500)(ICODE(I),I=6,45)
0254 N1=(J-1)*3+1
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 13:01:02

PAGE 005

PROC5.OBJ=PROC5.FOR/NOSN/LI:1

```
0255     IF (NOREC .LE. N1) GOTO 250
0257     NN=NN+1
0258     READ(3'NN,ERR=500)(ICODE(I),I=46,85)
0259     N2=N1+1
0260     IF (NOREC .LE. N2) GOTO 250
0262     NN=NN+1
0263     READ(3'NN,ERR=500)(ICODE(I),I=86,125)
0264     NN=NN+1
0265     250 ICODE(45)=ICRLF
0266         ICODE(85)=ICRLF
0267         ICODE(125)=ICRLF
0268         ICODE(126)=IDC1
0269         IF (J .EQ. NWRTS) ICODE(126)=LHOME
0271         CALL WRLOOP(I1ST)
0272         I1ST=1
0273     310 CONTINUE
0274     330 CONTINUE
0275         DO 70 J=1,NOREC
0276             NOR=NRCND+J-1
0277             READ(3'NOR)(MOUT(I,J),I=1,10)
0278     70 WRITE(1,15)(MOUT(I,J),I=1,10)
0279             DO 80 J=1,11
0280                 DO 80 I=1,10
0281     80 MOUT(I,J)=0
0282     500 CONTINUE
0283         IF (I1ST .NE. 9999) GOTO 25
0285         END
```

>

```
      C      INTERPROCESS COMM & RESOURCE SHARING DEMO
      C
0001      INTEGER DEVST(6)
0002      COMMON /LOOP/ICODE(128),MSK,LSK
0003      COMMON /M1710/DEVST
0004      REAL*4 RONE,RTWO,RLNE(20),RSPC
0005      REAL*8 MOUT(10,11)
0006      DATA IH1H,IHH1/'405','2401/
0007      DATA IC9H,IHC9/'4401','411/
0008      DATA IC4H,IHC4,IC8H,IHC8/'2001','404','4001','410/
0009      DATA ILO,ICVA,LEOP/'51504','125125','177777/
0010      DATA RONE,RTWO,RSPC/' 1',' 2','  ' //
0011      DATA NO,IB,ID/'ND',' B',' ID'/
0012      DATA LSK/'377/
0013      DATA LHOME,ICRLF,IDC1/'24','106412','10400/
0014      DATA LFF4,LFF5/'14','5000/
0015      DATA LF/'12/
0016      DATA MSK/'177400/
0017      15  FORMAT(1X,10A8)
0018      DO 130 J=1,129
0019      DO 140 I=1,70
0020      140 CONTINUE
0021      130 K=DEVST(1)
0022      I1ST=0
0023      CALL ASSIGN(1,'TTO:')
0024      CALL ASSIGN(2,'INFO.DAT')
0025      DEFINE FILE 2(396,40,U,I1)
0026      CALL ASSIGN(3,'MSC.DAT')
0027      DEFINE FILE 3(211,40,U,I2)
0028      DEVST(5)=0
0029      IST=1
0030      I82T81=81+ISHFT(82,8)
0031      I81T82=82+ISHFT(81,8)
0032      ND=1
0033      NDO=5
0034      25  CALL RDLOOP
0035      WRITE(1,16)(ICODE(J),J=1,4)
0036      16  FORMAT(1X,'HEADER=',408)
0037      IF (ICODE(3) .EQ. IC4H) ICODE(3)=IHC4
0039      IF (ICODE(3) .EQ. IC8H) ICODE(3)=IHC8
0041      IF (ICODE(3) .EQ. IC9H) ICODE(3)=IHC9
0043      IF (ICODE(3) .EQ. I81T82) GOTO 700
0045      IF (ICODE(4) .NE. ILO) GOTO 630
0047      NRCNO=21
0048      NOREC=1
0049      IST=9999
0050      GOTO 12
0051      630 IF (IST .EQ. 1) GOTO 51
0053      IF (IST .EQ. 2) GOTO 52
0055      IF (IST .EQ. 3) GOTO 53
0057      IF (IST .EQ. 4) GOTO 54
0059      IF (IST .EQ. 5) GOTO 55
0061      IF (IST .EQ. 6) GOTO 56
0063      IF (IST .EQ. 7) GOTO 57
0065      IF (IST .EQ. 8) GOTO 58
0067      IF (IST .EQ. 9) IST=1
```

FORTRAN IV U01B-02
CORE=08K, UIC=[20,20]FRI 18-MAR-77 12:51:31 PAGE 002
PROC1.OBJ=PROC1.FOR/NOSN/LI:1

```
0069        GOTO 51
0070        51 NRCNO=1
0071        NOREC=2
0072        IST=2
0073        GOTO 12
0074        52 DO 362 J=1,6
0075        NR=(ND-1)*6+121+J
0076        READ(2'NR)(MOUT(I,J),I=1,10)
0077        362 WRITE(3'44+J)(MOUT(I,J),I=1,10)
0078        NRCNO=45
0079        NOREC=6
0080        IST=IST+1
0081        GOTO 12
0082        53 DO 363 J=1,6
0083        NR=(NDD-1)*6+121+J
0084        READ(2'NR)(MOUT(I,J),I=1,10)
0085        363 WRITE(3'44+J)(MOUT(I,J),I=1,10)
0086        NRCNO=45
0087        NOREC=6
0088        IST=IST+1
0089        GOTO 12
0090        54 ICODE(1)=ICV,
0091        ICODE(5)=LEOP
0092        ICODE(2)=16
0093        ICODE(4)=81+ISHFT(2,8)
0094        ITSV=ICODE(3)
0095        ICODE(3)=IHH1
0096        CALL WRLOOP(I1ST)
0097        ICODE(3)=5+ISHFT(5,8)
0098        CALL WRLOOP(I1ST)
0099        ICODE(3)=6+ISHFT(5,8)
0100        CALL WRLOOP(I1ST)
0101        ICODE(4)=82+ISHFT(1,8)
0102        CALL WRLOOP(I1ST)
0103        ICODE(3)=IHH1
0104        CALL WRLOOP(I1ST)
0105        ICODE(3)=5+ISHFT(5,8)
0106        CALL WRLOOP(I1ST)
0107        ICODE(3)=2+ISHFT(5,8)
0108        CALL WRLOOP(I1ST)
0109        ICODE(3)=ITSV
0110        ICODE(1)=ICODE(1)+1
0111        ICODE(2)=0
0112        NR=(ND-1)*6+121+6
0113        331 READ(2'NR)(RLNE(I),I=1,20)
0114        DO 332 LL=1,20
0115        332 IF (RLNE(LL) .EQ. 0) RLNE(LL)=RSPC
0117        RLNE(1)=RTWD
0118        RLNE(2)=RONE
0119        WRITE(2'NR)(RLNE(I),I=1,20)
0120        IF (NR .EQ. 151) GOTO 52
0122        NR=(NDD-1)*6+121+6
0123        GOTO 331
0124        55 GOTO 53
0125        56 ICODE(1)=ICODE(1)+1
0126        ICODE(2)=0
```

FORTRAN IV V01B-02
CORE=08K, UIC=[20,20]

FRI 18-MAR-77 12:51:31

PAGE 003

PROC1.OBJ=PROC1.FOR/NOSN/LI:1

```
0127      ICODE(3)=I82T81
0128      ICODE(4)=LEOP
0129      CALL WRLOOP(I1ST)
0130      T1=SECNDS(0.)
0131      90 I=MOD(DEVST(3),2)
0132      IF (I .EQ. 1) GOTO 120
0134      IF (I .EQ. -1) GOTO 120
0136      DO 110 I=1,10
0137      110 CONTINUE
0138      DELTA=SECNDS(T1)
0139      IF (DELTA .GE. 20.) GOTO 150
0141      GOTO 90
0142      120 DO 131 J=1,129
0143      DO 141 I=1,3
0144      141 CONTINUE
0145      K=DEVST(1)
0146      131 CONTINUE
0147      DEVST(5)=0
0148      ICODE(4)=IB
0149      ICODE(5)=ID
0150      ICODE(6)=LEOP
0151      GOTO 160
0152      150 ICODE(4)=NO
0153      ICODE(5)=IB
0154      ICODE(6)=ID
0155      ICODE(7)=LEOP
0156      160 ICODE(1)=0
0157      ICODE(2)=0
0158      ICODE(3)=ITSV
0159      CALL WRLOOP(I1ST)
0160      IF (ICODE(4) .EQ. IB) GOTO 170
0162      IST=1
0163      GOTO 25
0164      170 NRCNO=14
0165      NOREC=1
0166      IST=7
0167      GOTO 12
0168      57 ICODE(3)=I82T81
0169      CALL WRLOOP(I1ST)
0170      ICODE(1)=ICVA
0171      ICODE(5)=LEOP
0172      ICODE(2)=16
0173      ICODE(4)=81
0174      ICODE(3)=IHH1
0175      CALL WRLOOP(I1ST)
0176      ICODE(3)=5+ISHFT(5,8)
0177      CALL WRLOOP(I1ST)
0178      ICODE(4)=82
0179      CALL WRLOOP(I1ST)
0180      ICODE(3)=IHH1
0181      CALL WRLOOP(I1ST)
0182      ICODE(3)=ITSV
0183      ICODE(1)=ICODE(1)+1
0184      ICODE(2)=0
0185      NR=(ND-1)*6+121+6
0186      341 READ(2'NR)(RLNE(I),I=1,20)
```

FORTRAN IV V01B-02
 CORE=08K, UIC=[20,20]

FRI 18-MAR-77 12:51:31

PAGE 004

PROC1.OBJ=PROC1.FOR/NOSN/LI:1

```

0187      DO 342 LL=1,20
0188  342 IF (RLNE(LL) .EQ. 0) RLNE(LL)=RSPC
0190      RLNE(1)=RSPC
0191      RLNE(2)=RSPC
0192      WRITE(2,NR)(RLNE(I),I=1,20)
0193      IF (NR .EQ. 151) GOTO 52
0195      NR=(NDO-1)*6+121+6
0196      GOTO 341
0197      58 GOTO 53
0198  700 ICODE(4)=IAND(LSK,ICODE(4))
0199      IF (ICODE(4) .NE. LSK) GOTO 800
0201      ICODE(1)=ICODE(1)+1
0202      ICODE(2)=0
0203      ICODE(3)=I82TB1
0204      ICODE(4)=LEOP
0205      CALL WRLOOP(I1ST)
0206      GOTO 25
0207  800 DO 810 J=4,120
0208      KKK=J
0209      ILM=IAND(MSK,ICODE(J))
0210      ILL=IAND(LSK,ICODE(J))
0211      IF (ILM .EQ. MSK) GOTO 820
0213  810 IF (ILL .EQ. LSK) GOTO 830
0215  820 ICODE(KKK)=IAND(LSK,ICODE(KKK))
0216      GOTO 840
0217  830 KKK=KKK-1
0218  840 WRITE(1,20)(ICODE(J),J=4,KKK)
0219      20 FORMAT(1X,39A2/1X,39A2/1X,39A2/)
0220      GOTO 25
0221      12 CONTINUE
0222      IF (NRCNO .GE. 107) NRCNO=NRCNO+1
0224      DO 200 J=4,128
0225  200 ICODE(J)=0
0226      ICODE(127)=LEOP
0227      IF (NRCNO .NE. 11) GOTO 210
0229      ICODE(4)=LHOME
0230      DO 220 J=5,16
0231  220 ICODE(J)=LF
0232      READ(8'11)(ICODE(I),I=17,56)
0233      ICODE(57)=LHOME
0234      ICODE(58)=LEOP
0235      CALL WRLOOP(I1ST)
0236      GOTO 330
0237  210      NWRTS=4
0238      IF (NOREC .LE. 3) NWRTS=1
0240      IF (NOREC .GE. 4 .AND. NOREC .LE. 6) NWRTS=2
0242      IF (NOREC .GE. 7 .AND. NOREC .LE. 9) NWRTS=3
0244      NN=NRCNO
0245      DO 310 J=1,NWRTS
0246      JJJ=J
0247      DO 450 K=4,126
0248  450 ICODE(K)=0
0249      IF (J .EQ. 1) ICODE(4)=LFF4
0251      IF (J .EQ. 1) ICODE(5)=LFF5
0253      READ(3'NN,ERR=500)(ICODE(I),I=6,45)
0254      N1=(J-1)*3+1

```

FORTRAN IV V01B-02
CORE=08K, UIC=L20,20J

FRI 18-MAR-77 12:51:31

PAGE 005

PROC1.OBJ=PROC1.FOR/NOSN/LI:1

```
0255     IF (NOREC .LE. N1) GOTO 250
0257     NN=NN+1
0258     READ(3'NN,ERR=500)(ICODE(I),I=46,85)
0259     N2=N1+1
0260     IF (NOREC .LE. N2) GOTO 250
0262     NN=NN+1
0263     READ(3'NN,ERR=500)(ICODE(I),I=86,125)
0264     NN=NN+1
0265     250 ICODE(45)=ICRLF
0266         ICODE(85)=ICRLF
0267         ICODE(125)=ICRLF
0268         ICODE(126)=IDC1
0269         IF (J .EQ. NWRTS) ICODE(126)=LHOME
0271         CALL WRLOOP(I1ST)
0272         I1ST=1
0273     310 CONTINUE
0274     330 CONTINUE
0275         DO 70 J=1,NOREC
0276             NOR=NRCNO+J-1
0277             READ(3'NOR)(MOUT(I,J),I=1,10)
0278     70 WRITE(1,15)(MOUT(I,J),I=1,10)
0279             DO 80 J=1,11
0280                 DO 80 I=1,10
0281     80 MOUT(I,J)=0
0282     500 CONTINUE
0283         IF (IST .NE. 9999) GOTO 25
0285     END
```

>

1.7 Diagnostics

1.7.1 PDP-11 Interface (PDP)

This diagnostic program checks the M1710 PDP-11 interface and the Host Interface Board. The diagnostic runs on the host machine to be tested and the microcode object file PDPO.OBJ runs on the B7* CIE microprocessor connected to that host. The source (.FOR), object (.OBJ), and the task (.TSK) files reside on ESM Tape #4. The program uses terminal TTO: for I/O. A timing parameter is requested which is entered in I3 format, and which must be greater or equal to 3. This timing parameter controls the rate at which words are sent across the interface. The program then provides a time for packet transmission printout. A three line packet can then be entered on the terminal. Proper interface operation results in the message being printed on the terminal after the CIE program is run from location WRTB (see PDPO description in Section 2.3).

Task Builder (TKB) options are:

```
UNITS=1
ACTFIL=1
MAXBUF=80
COMMON=M1710:RW
ASG=TTO:1
```

1.7.2 Control Memory - CONMEM

ESM Tape #4 contains the source (.FOR), object (.OBJ), and task (.TSK) files for the control memory checking program (CONMEM). The program is a variation of the ESMLDR utility described in Section 1.4 in which a halt is inserted at the last word of control memory. The program is useful in debugging certain types of control memory and loading problems.

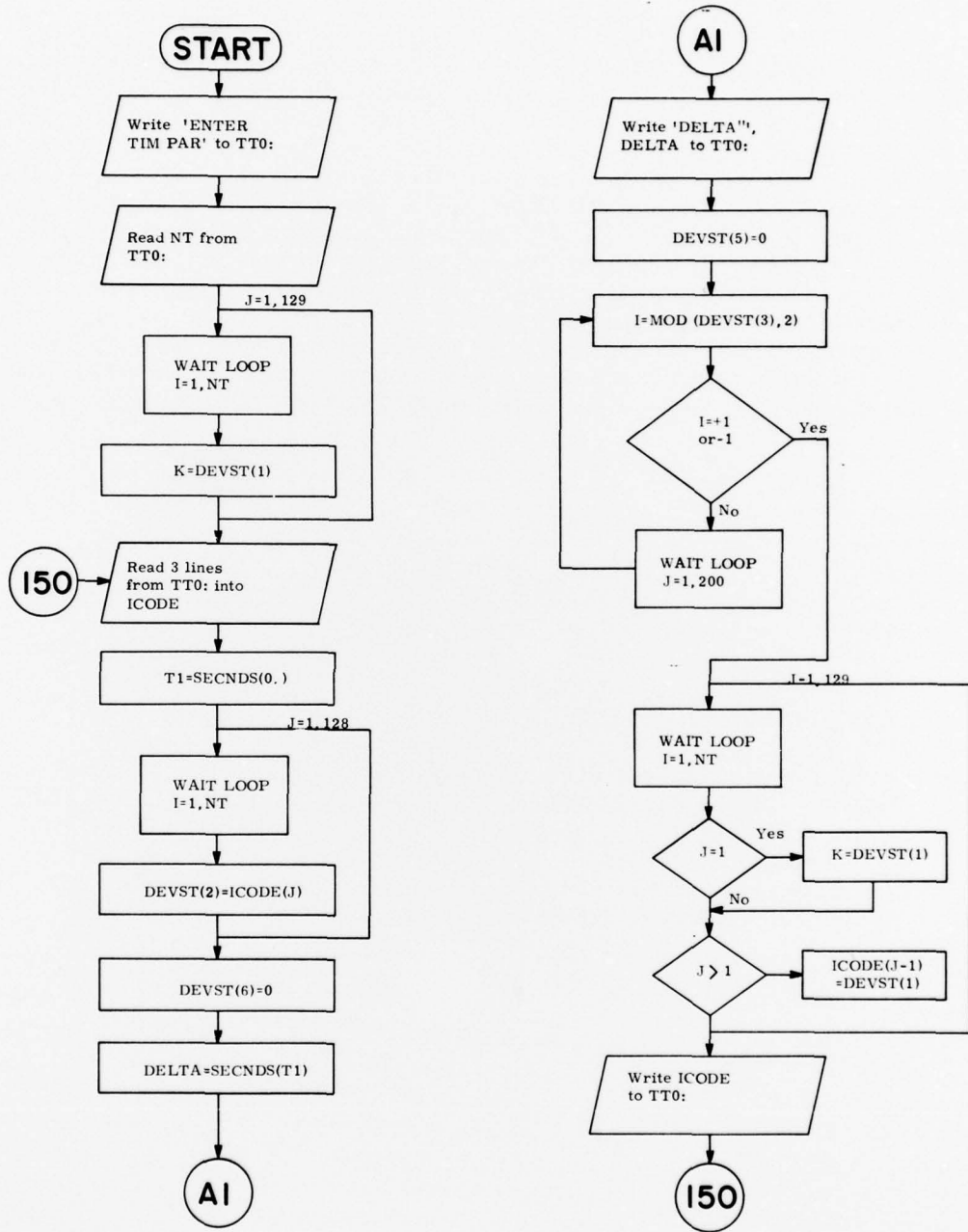


Figure 1-23. PDP

FORTRAN IV V01B-02 TUE 01-MAR-77 16:34:58 PAGE 001
CORE=08K, UIC=[1,4] PDP.OBJ=PDP.FOR/NOSN/LI:1

```
      C      DIAG PROG FOR CKING M1710-CIE INT
0001      INTEGER DEVST(6)
0002      COMMON /M1710/DEVST
0003      DIMENSION ICODE(128)
0004      CALL ASSIGN(1,'TT0:')
0005      NT=70
      D      WRITE(1,300)
D 300     FORMAT(1X,'ENTER TIM PAR')
      D      READ(1,310) NT
D 310     FORMAT(I3)
      C      THROW AWAY PACKET FOR INP BUF INIT EMPTY
0006      DO 130 J=1,129
0007      DO 140 I=1,NT
0008      140 CONTINUE
0009      130 K=DEVST(1)
0010      150 READ(1,30)(ICODE(I),I=1,40)
0011      READ(1,30)(ICODE(I),I=41,80)
0012      READ(1,30)(ICODE(I),I=81,120)
0013      30 FORMAT(40A2)
      D      T1=SECNDS(0.)
D 70      DO 70 J=1,128
0015      DO 80 I=1,NT
0016      80 CONTINUE
0017      70 DEVST(2)=ICODE(J)
0018      DEVST(6)=0
      D      DELTA=SECNDS(T1)
D 400     WRITE(1,400) DELTA
D 400     FORMAT(1X,'DELTA=',F9.5)
0019      DEVST(5)=0
0020      90 I=MOD(DEVST(3),2)
0021      IF (I .EQ. 1) GOTO 120
0023      IF (I .EQ. -1) GOTO 120
0025      DO 110 J=1,200
0026      110 CONTINUE
0027      GOTO 90
0028      120 DO 230 J=1,129
0029      DO 240 I=1,NT
0030      240 CONTINUE
0031      IF (J .EQ. 1) K=DEVST(1)
0033      IF (J .GT. 1) ICODE(J-1)=DEVST(1)
0035      230 CONTINUE
0036      WRITE(1,160)(ICODE(I),I=1,39)
0037      WRITE(1,160)(ICODE(I),I=41,79)
0038      WRITE(1,160)(ICODE(I),I=81,119)
0039      160 FORMAT(1X,39A2/)
0040      GOTO 150
0041      END
```

>

FORTRAN IV V01B-02
CORE=08K, UIC=L1,4J

TUE 01-MAR-77 16:33:29

PAGE 001

CONMEM.OBJ=CONMEM.FOR/NOSN/LI:1

```
0001     INTEGER DEVST(6),CODE(128)
0002     REAL*8 ZA(3)
0003     COMMON /M1710/DEVST
0004     DATA ISTEP/'607/
0005     DATA IAON/'177777/
0006     DATA IER1,IER2/'363,'7776/
0007     CALL ASSIGN(1,'TT1:')
0008     WRITE(1,11)
0009     11 FORMAT(1X,'PLEASE ENTER OBJECT FILE NAME')
0010     READ(1,12)ZA
0011     12 FORMAT(3A8)
0012     CALL ASSIGN(2,ZA)
0013     DEFINE FILE 2(32,128,U,I1)
0014     DO 20 NREC=1,32
0015     NR=NREC
0016     READ(2,NREC,END=99,ERR=99) CODE
0017     DO 18 J=1,128
0018     DO 16 KK=1,5
0019     16 CONTINUE
0020     IF (CODE(J) .EQ. IAON) CODE(J)=ISTEP
0022     DEVST(2)=CODE(J)
0023     18 CONTINUE
0024     WRITE (1,13) NREC
0025     13 FORMAT(1X,I3,' 128 INSTRUCTION GROUPS LOADED.')
0026     20 CONTINUE
0027     99 DO 30 J=1,128
0028     30 CODE(J)=ISTEP
0029     DO 40 N=NR,31
0030     DO 40 J=1,128
0031     DO 50 KK=1,5
0032     50 CONTINUE
0033     DEVST(2)=CODE(J)
0034     40 CONTINUE
0035     CODE(126)=IER1
0036     CODE(127)=IER2
0037     DO 60 J=1,127
0038     DO 70 KK=1,5
0039     70 CONTINUE
0040     DEVST(2)=CODE(J)
0041     60 CONTINUE
0042     END
```

>

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 66143-3	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Software Maintenance Manual, Exploratory Systems Control Model (ESM), Book 1 Fortran		5. TYPE OF REPORT & PERIOD COVERED Final Report July 75 - April 77
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) DCA 100-75-C-0054
9. PERFORMING ORGANIZATION NAME AND ADDRESS Burroughs Corporation Federal and Special Systems Group Paoli, PA 19301		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task 15203 P.E. 33143K
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Engineering Center 1860 Wiehle Avenue Reston, VA 22090		12. REPORT DATE April 77
		13. NUMBER OF PAGES 242
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same as 11		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same as 16		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) loops, rings, System Control, Defense Communications		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This publication is the Software Maintenance Manual for the Exploratory Systems Control Model (ESM). The software described is contained on four system tapes. Book 1 contains description, flowcharts, and listings for programs written in FORTRAN. Book 2 contains description, flowcharts, and listings for programs written in MDMPL Assemble Language. This manual was prepared by the Burroughs Corporation and is submitted in accordance with the requirements of contract DCA 100-75-C-0054.		



Burroughs Corporation

Federal and Special Systems Group

Paoli, Pennsylvania 19301