178
4/13/78
uncus NTB

Hq. 2016

MASTER

CONS/0385-3
NASA CR-135285
BCS 40180-3

A SIMULATION MODEL FOR WIND ENERGY STORAGE SYSTEMS

Volume III: Program Descriptions

A. W. Warren, R. W. Edsinger, J.D. Burroughs
ENERGY TECHNOLOGY APPLICATIONS DIVISION
BOEING COMPUTER SERVICES COMPANY
A Division of The Boeing Company

950 5505

August 1977

Prepared for the
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center
Cleveland, Ohio 44135

Contract NAS 3-20385

As a part of the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
Division of Energy Storage Systems

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

NOTICE

This report was prepared to document work sponsored by the United States Government. Neither the United States nor its agent, the United States Department of Energy, nor any Federal employees, nor any of their contractors, subcontractors or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

1. Report No. NASA CR-135285	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

A Simulation Model for Wind Energy Storage Systems
Volume III: Program Descriptions

5. Report Date

August, 1977

6. Performing Organization Code

7. Author(s)

A. W. Warren, R. W. Edsinger, J. D. Burroughs

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address

Energy Technology Applications Division of
Boeing Computer Services Company
Seattle, Washington 98124

11. Contract or Grant No.

NAS3-20385

13. Type of Report and Period Covered

Contractor Report

12. Sponsoring Agency Name and Address

Energy Research and Development Administration
Division of Energy Storage Systems
Washington, D.C. 20545

14. Sponsoring Agency ~~Code~~ Report No.

CONS/0385-3

15. Supplementary Notes

Final report. Prepared under Interagency Agreement E(49-28)-1026. Project Manager, Larry H. Gordon, Power Generation and Storage Division, NASA Lewis Research Center, Cleveland, Ohio 44135.

16. Abstract

The effort developed a comprehensive computer program for the modeling of wind energy/storage systems utilizing any combination of five types of storage (pumped hydro, battery, thermal, flywheel and pneumatic). An acronym for the program is SIMWEST (Simulation Model for Wind Energy Storage). The level of detail of SIMWEST is consistent with a role of evaluating the economic feasibility as well as the general performance of wind energy systems.

The software package consists of two basic programs and a library of system, environmental, and load components. The first program is a precompiler which generates computer models (in Fortran) of complex wind source/storage/application systems, from user specifications using the respective library components. The second program provides the techno-economic system analysis with the respective I/O, the integration of system dynamics, and the iteration for conveyance of variables. This SIMWEST program, as described, runs on the UNIVAC 1100 series computers.

This technical report contains three volumes. Volume I gives a brief overview of the SIMWEST program and describes the two NASA defined simulation studies. Volume II, the SIMWEST operation manual, describes the usage of the SIMWEST program, the design of the library components, and a number of simple example simulations intended to familiarize the user with the program's operation. Volume II also contains a listing of each SIMWEST library subroutine. Volume III, the SIMWEST program description contains program descriptions, flow charts and program listings for the SIMWEST Model Generation Program, the Simulation program, the File Maintenance program and the Printer Plotter program. Volume III generally would not be required by SIMWEST user.

17. Key Words (Suggested by Author(s)) Energy Storage, Computer Programs, System Simulation, Wind Energy	18. Distribution Statement Unclassified - unlimited STAR Category 61 ERDA Category UC-94b

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 238	22. Price*

* For sale by the National Technical Information Service, Springfield, Virginia 22161

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

FOREWARD

This report presents results of work conducted by Boeing Computer Services Company under NASA Contract NAS3-20385, "Wind Energy Storage Model Development." This program was conducted under the sponsorship of the Advanced Physical Methods Branch, Office of Conservation, ERDA, under the direction of Dr. G. C. Chang, and was administered by the NASA-Lewis Research Center Thermal and Mechanical Storage Section with Mr. L. H. Gordon as Project Manager. This report is in three volumes.

I. Technical Report
II. Operation Manual
III. Program Descriptions

The Boeing Program Manager for this work was R. W. Edsinger, and A. W. Warren was the principal investigator.

For completeness, the summary sections 1.1 and 1.2 of Volume I have been repeated in the Operation Manual, Volume II.

1.0 INTRODUCTION

This volume describes the computer programs for the simulation model for wind energy storage (SIMWEST). Each of the following sections contain a verbal program description with macro flow charts, and source code listings for each major program entity. Section 2.0 describes the model generation precompiler program which creates a Fortran model for the system to be simulated. Section 3.0 describes the simulation program. This is the executive program that exercises the Fortran model generated by the model generation program. Section 4.0 describes the file maintenance program (FILOAD). Section 5.0 describes the printer plotter program which is a post processor for the simulation program. All the source code to run a simulation is given in this volume, except for the library component source listings. The library source listings are given in Section 7.0 of Volume II, the User's Manual.

2.0 MODEL GENERATION PROGRAM DESCRIPTION

2.1 INTRODUCTION

The Model Generation program accepts program commands which describe the system model in terms of standard components. Each standard component is represented by a subroutine. The program then constructs a FORTRAN model which consists of a series of calls to these subroutines. In addition to generating the FORTRAN source code for the system model, the Model Generation program produces a line printer drawn schematic diagram of the system and a list of the input data required to complete the model description.

Upon completion of model generation, the FORTRAN source code is compiled and the resultant object code is available as input to the simulation program. The model source code may be punched onto cards for storage or manipulation by the system analyst. The model object code is also stored on a permanent file. In this way a given model can be used for several simulation runs without having to regenerate the model for each analysis.

2.2 PROGRAM STRUCTURE

Figure 2.2-1 contains a macro flow diagram of the Model Generation program. This flow diagram shows the principle tasks of the program. For each task, a statement number in the main program is given along with the name of the principle subroutine that accomplishes the task.

The first task upon starting program execution is to obtain the current list of all standard components. The SIMWEST program was designed to be independent of the number or type of standard components. All that is required of the standard components is that their inputs, outputs, and table quantities be arranged according to certain rules discussed in Section 6. Vol. 1.

The sequence of performing the subsequent tasks is very model dependent. As each task is identified and performed, data describing the system model are

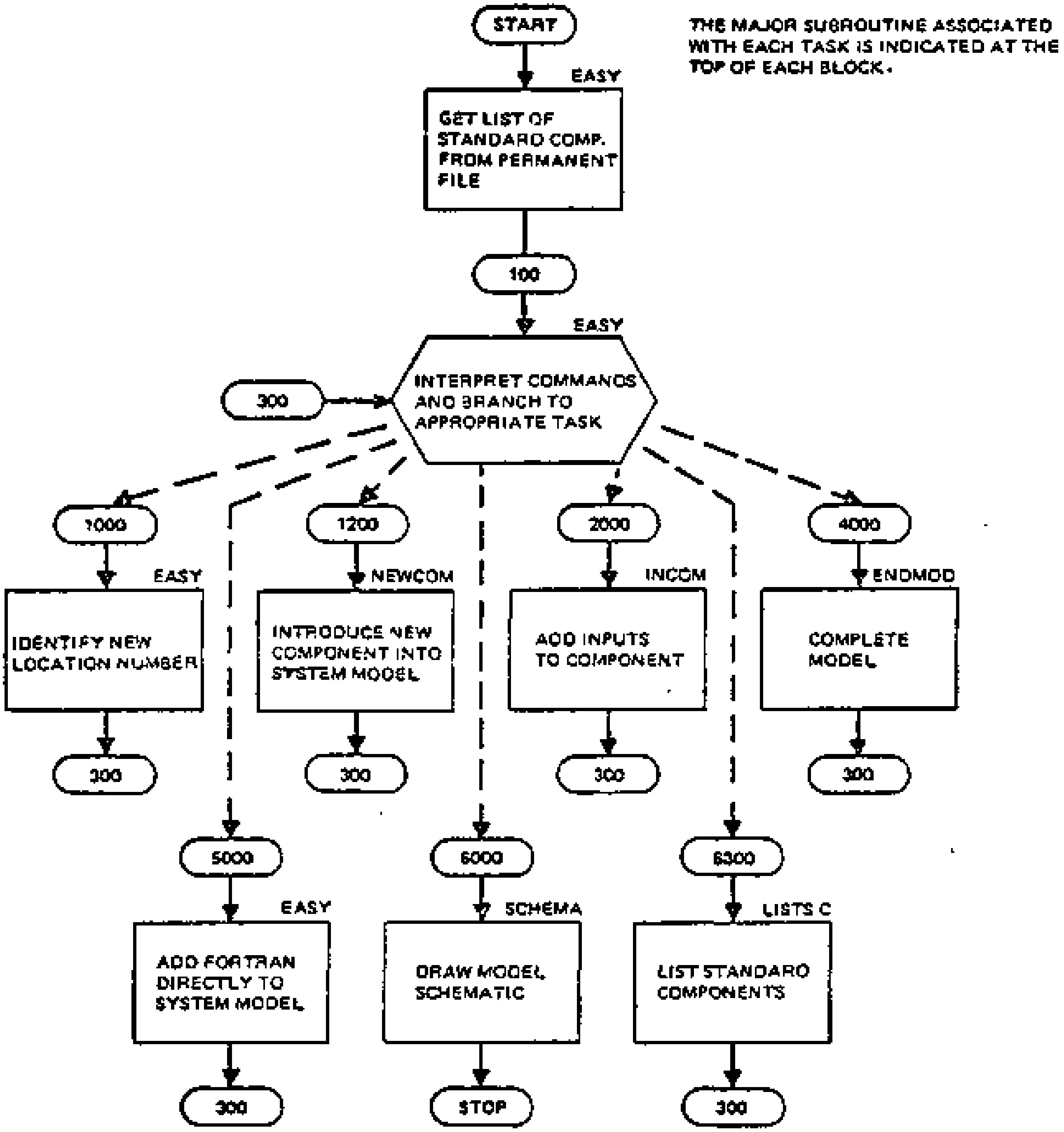

START
THE MAJOR SUBROUTINE ASSOCIATED WITH EACH TASK IS INDICATED AT THE TOP OF EACH BLOCK.
EASY
GET LIST OF STANDARD COMP. FROM PERMANENT FILE
100
EASY
INTERPRET COMMANDS AND BRANCH TO APPROPRIATE TASK
300
1000
EASY
IDENTIFY NEW LOCATION NUMBER
300
1200
NEWCOM
INTRODUCE NEW COMPONENT INTO SYSTEM MODEL
300
2000
INCOM
ADD INPUTS TO COMPONENT
300
4000
ENDMOD
COMPLETE MODEL
300
5000
EASY
ADD FORTRAN DIRECTLY TO SYSTEM MODEL
300
6000
SCHEMA
DRAW MODEL SCHEMATIC
STOP
6300
LISTS C
LIST STANDARD COMPONENTS
300

Figure 2.2-1 SIMWEST Model Generation Program - Macro Flow Diagram

accumulated on a random access temporary file. This file, M7, contains a list of inputs for each component in the system model. As inputs are satisfied by model connections their names are modified to indicate the source of the input information. A list of model component names, CMPMOD, is kept in core. In addition to the component name, this list contains codes indicating the location of the component on the model schematic, the symbol to be used for the component and the number of inputs the component requires.

Once the END OF MODEL command is received, the data accumulated for the model is processed to generate the model source code and the model schematic diagram.

The following sections describe each of the major tasks shown in Figure 2.2-1. Source listings for all subroutines are included in Section 2.3.

2.2.1 Command Interpretation

The second task performed by the program is to begin the interpretation of data cards which contain the system model description commands. Figure 2.2-2 contains a macro flow diagram of the command interpretation process.

As each command card is read it is printed to provide a record of progress through the model description. The model description is given as a series of "phrases." These phrases are identified in each card image by the routine, NXTPH, which locates one of the allowable phrase delimiters: comma, [,] , equals, [=], left or right parenthesis, [()], or three or more blanks. When the end of a card is reached, a blank phrase is returned by NXTPH which causes a new command card to be read.

Each phrase is first tested against the set of command phrases, shown in Table 2.2-1. If a match is obtained between the first ten characters of the input phrase and one of the command phrases the program branches to statement 400. At statement 400, tests are performed for unfinished tasks such as component definition that must be completed, or the end of the direct

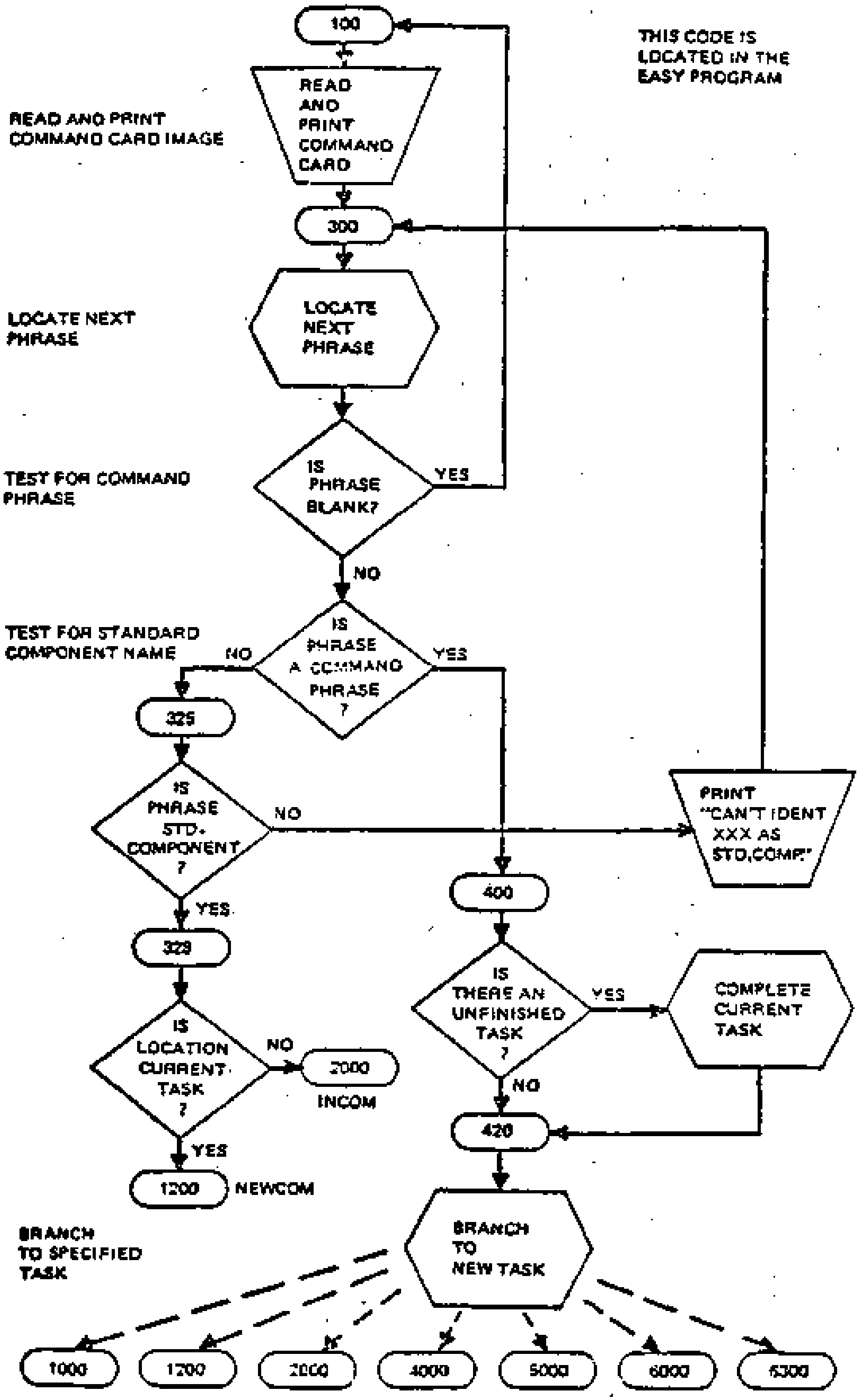

100
READ AND PRINT COMMAND CARD
THIS CODE IS LOCATED IN THE EASY PROGRAM
READ AND PRINT COMMAND CARD IMAGE
300
LOCATE NEXT PHRASE
LOCATE NEXT PHRASE
TEST FOR COMMAND PHRASE
IS PHRASE BLANK?
YES
NO
TEST FOR STANDARD COMPONENT NAME
IS PHRASE A COMMAND PHRASE ?
NO
YES
325
IS PHRASE STD. COMPONENT ?
NO
PRINT "CAN'T IDENT XXX AS STD.COMP."
YES
328
400
IS LOCATION CURRENT TASK ?
NO
2000
INCOM
YES
1200
NEWCOM
IS THERE AN UNFINISHED TASK ?
YES
COMPLETE CURRENT TASK
NO
420
BRANCH TO SPECIFIED TASK
BRANCH TO NEW TASK
1000
1200
2000
4000
5000
6000
6300

Figure 2.2-2. Model Generation Command Interpretation - Macro Flow Diagram

TABLE 2.2-1

MODEL GENERATION PROGRAM COMMAND PHRASES

PHRASE	USE
ADD PARAMETERS	Direct addition of parameters to model
ADD STATES	Direct addition of states to model
ADD TABLES	Direct addition of tables to model
ADD VARIABLES	Direct addition of variables to model
DIAGNOSTIC CONTROL	Control diagnostic printout to model
END OF MODEL	Specify end of model description
FORTRAN STATEMENTS	Specify start of FORTRAN statements
INPUTS	Specify input components
LIST STANDARD COMPONENTS	Request listing of standard components
LOCATION	Specify component location on schematic
MODEL DESCRIPTION	Specify start of model description
PRINT	Requested printed model output
PUNCH	Request printed and punched model output

FORTRAN input task. Once any unfinished task has been completed a branch is made at statement 420 to the new task.

If the input phrase is not identified as a command phrase, it's first two characters are compared to the list of standard component names, at statement 325. If the phrase is identified as a standard component, the program proceeds to either the new component routine, NEWCOM, or the component input routine, INCOM, depending on the current task.

If a particular command phrase requires additional modifying phrases, these phrases will be located on the command card and examined as to their suitability as a part of performing the requested task. For example the INPUTS task will check for modifying port numbers or physical quantity names associated with the input component. The "suitability" of a phrase will be determined by assuring that it is numeric, a physical quantity name, etc. depending on the specified task.

2.2.2 LOCATION Command Execution

The LOCATION command introduces the definition of a new component into the system model. This command must be followed by a numeric phrase that specifies the component location on the model schematic diagram. Failure to furnish a numeric location number causes a warning to be printed and the component will not appear on the model schematic.

If the previous command involved the specification of a component LOCATION, or INPUTS, the input quantity list for that component is stored before examining the next phrase as a valid location number.

2.2.3 New Component Name Examination

The next phrase following the location number phrase should contain the name of a standard component. When this occurs the subroutine NEWCOM is called.

If the name is not that of a standard component a warning message will be printed and the program will continue on with command card interpretation.

A flow diagram of the NEWCOM subroutine is shown in Figure 2.2-3. The main purpose of the NEWCOM subroutine is to get copies of the input and output lists for the specified component. Master copies of these lists are stored on permanent file, M18, for all standard components. However, if a component has already appeared in the model description, an input list for that component will be stored on local file, M7. This copy of the input list must be used since it may contain information regarding previous connections.

Additional tasks performed by NEWCOM include storing the symbol number, location number, and number of inputs, in the component name. These three integer numbers are stored in the last six characters of the component's name by means of the PUTCOD routine. The PUTCOD routine allows up to 5 integer values to be stored in a double precision word. These integers may assume values between ± 2047. The routine GETCOD is used to retrieve these values. Figure 2.2-4 shows how the ten characters of each model component's name are used.

The PUTCOD routine is also used to store each model component's identification number, IDCOMP, in the LOCATION sequence array, SEQA. Components are assigned consecutive identification numbers as they first appear in a model description. These numbers define the sequence of component names in the model component name list, CMPMOD, and are used as the record numbers for the component input lists on the mass storage file, M7. The sequence array, SEQA, stores the component identification numbers in the sequence that is specified by the components' LOCATION statements. In some cases this sequence may differ from that of first appearance in the model description. The LOCATION statement sequence specifies the sequence that each model component subroutine is to be called in the system model.

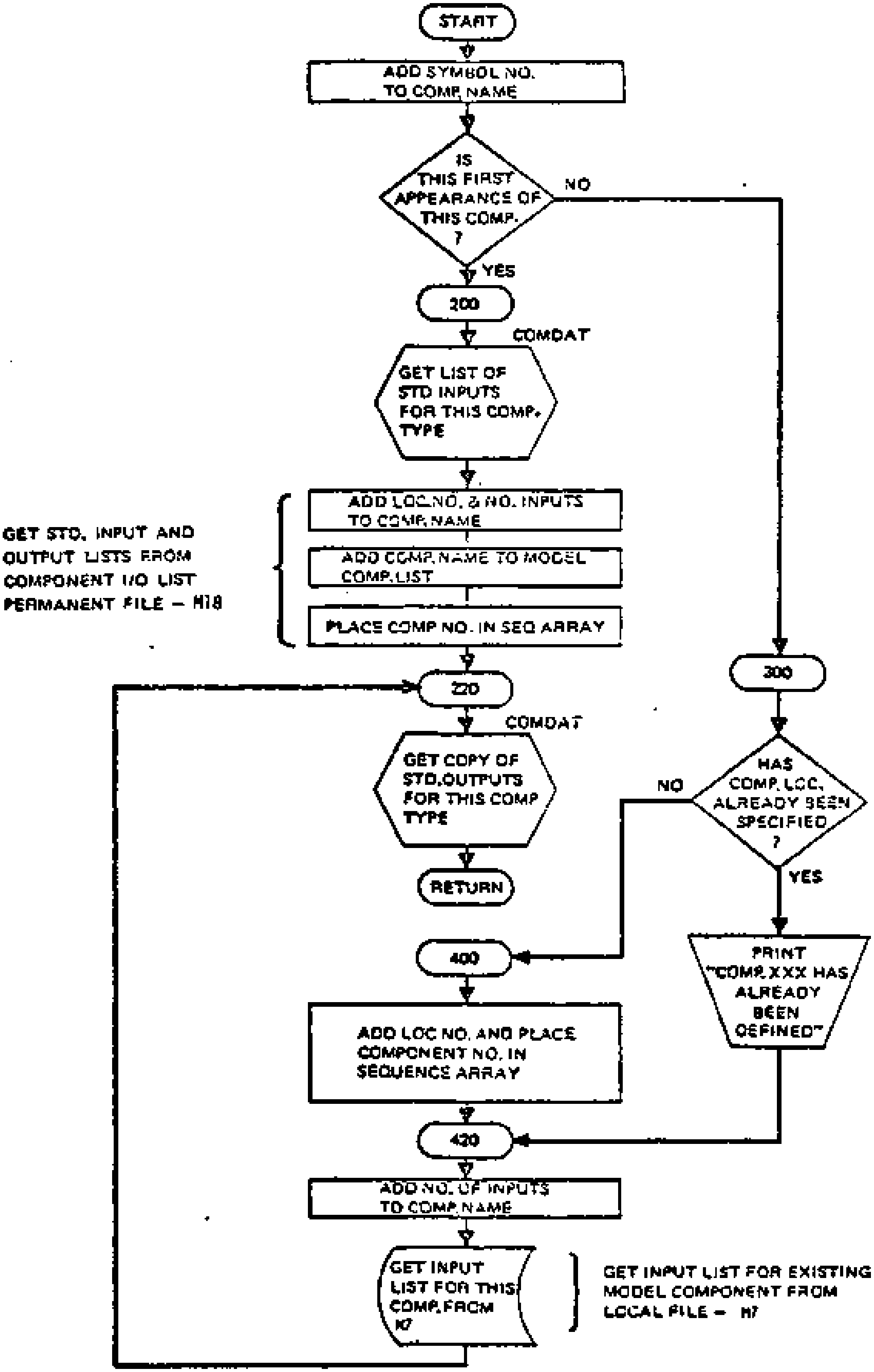

START
ADD SYMBOL NO. TO COMP. NAME
IS THIS FIRST APPEARANCE OF THIS COMP. ?
NO
YES
200
COMDAT
GET LIST OF STD INPUTS FOR THIS COMP. TYPE
ADD LOC NO. & NO. INPUTS TO COMP. NAME
ADD COMP. NAME TO MODEL COMP. LIST
PLACE COMP NO. IN SEQ ARRAY
GET STD. INPUT AND OUTPUT LISTS FROM COMPONENT I/O LIST PERMANENT FILE – H18
220
COMDAT
GET COPY OF STD. OUTPUTS FOR THIS COMP TYPE
RETURN
300
HAS COMP. LOC. ALREADY BEEN SPECIFIED ?
NO
YES
PRINT "COMP. XXX HAS ALREADY BEEN DEFINED"
400
ADD LOC NO. AND PLACE COMPONENT NO. IN SEQUENCE ARRAY
420
ADD NO. OF INPUTS TO COMP. NAME
GET INPUT LIST FOR THIS COMP. FROM H7
GET INPUT LIST FOR EXISTING MODEL COMPONENT FROM LOCAL FILE – H7

Figure 2.2-3. Subroutine NEWCOM - Macro Flow Diagram

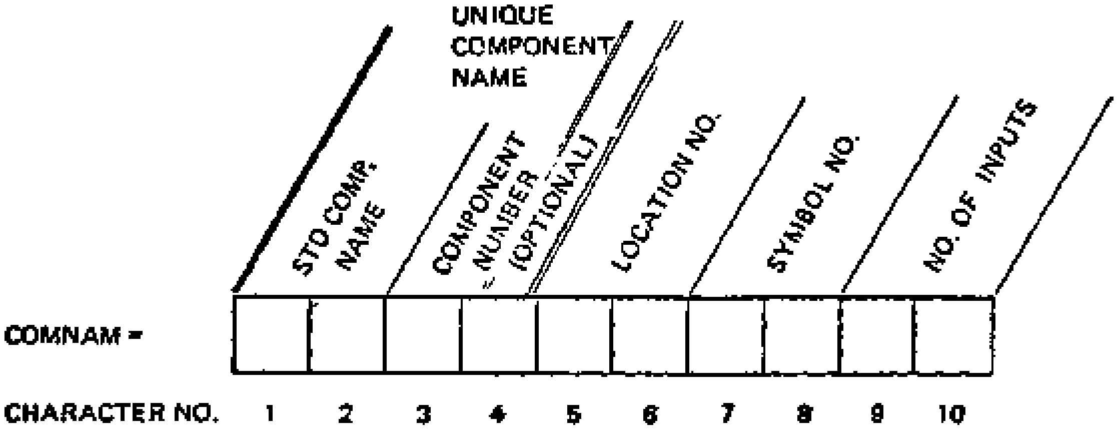


Figure 2.2-4. Use of Characters in Component Names

2.2.4 Inputs

The INPUTS command proceeds one or more instructions specifying those components which provide inputs to the component which has just been located. Component interconnections are made in the routine INCOM. Connections are recorded in the lists of inputs which are generated for each component as they are introduced into the model. The source of an input is indicated by replacing the standard physical quantity input name with the output quantity name of the source. Characters 4 through 6 of this name identifies the source component.

Figure 2.2-5 gives a macro-flow diagram of the INCOM routine. Upon entering the INCOM routine, input and output name lists are obtained for the upstream, i.e. input component. If this is the first appearance of this component the input list is obtained from the permanent file, M18, via the routine COMDAT. If the component had previously appeared in the model, it will have an input list on local file, M7, which will be used. The next phrase after the upstream component name is then examined. There are three valid possibilities for this phrase. It can be blank or another standard component name in which case the default option of connecting all matching physical quantities at a pair of ports is taken. If this phrase is numeric it is assumed that ports are being specified and all matching quantities at those ports are connected, via the routine PORTCN. If the phrase is alphanumeric and matches an output quantity of the upstream component, only the specified physical quantities are connected. Before returning from the INCOM routine the input list for the upstream component is stored on M7.

2.2.5 END OF MODEL Command Execution

The END OF MODEL command indicates the end of the model description. This command initiates the model generation process by the ENDMOD subroutine. The ENDMOD subroutine generates the FORTRAN source code for the system model routines EQMO, DATAIN, and BLOCK DATA MODEL and forms the model input requirements list. The principle sources of data for the ENDMOD routine are:

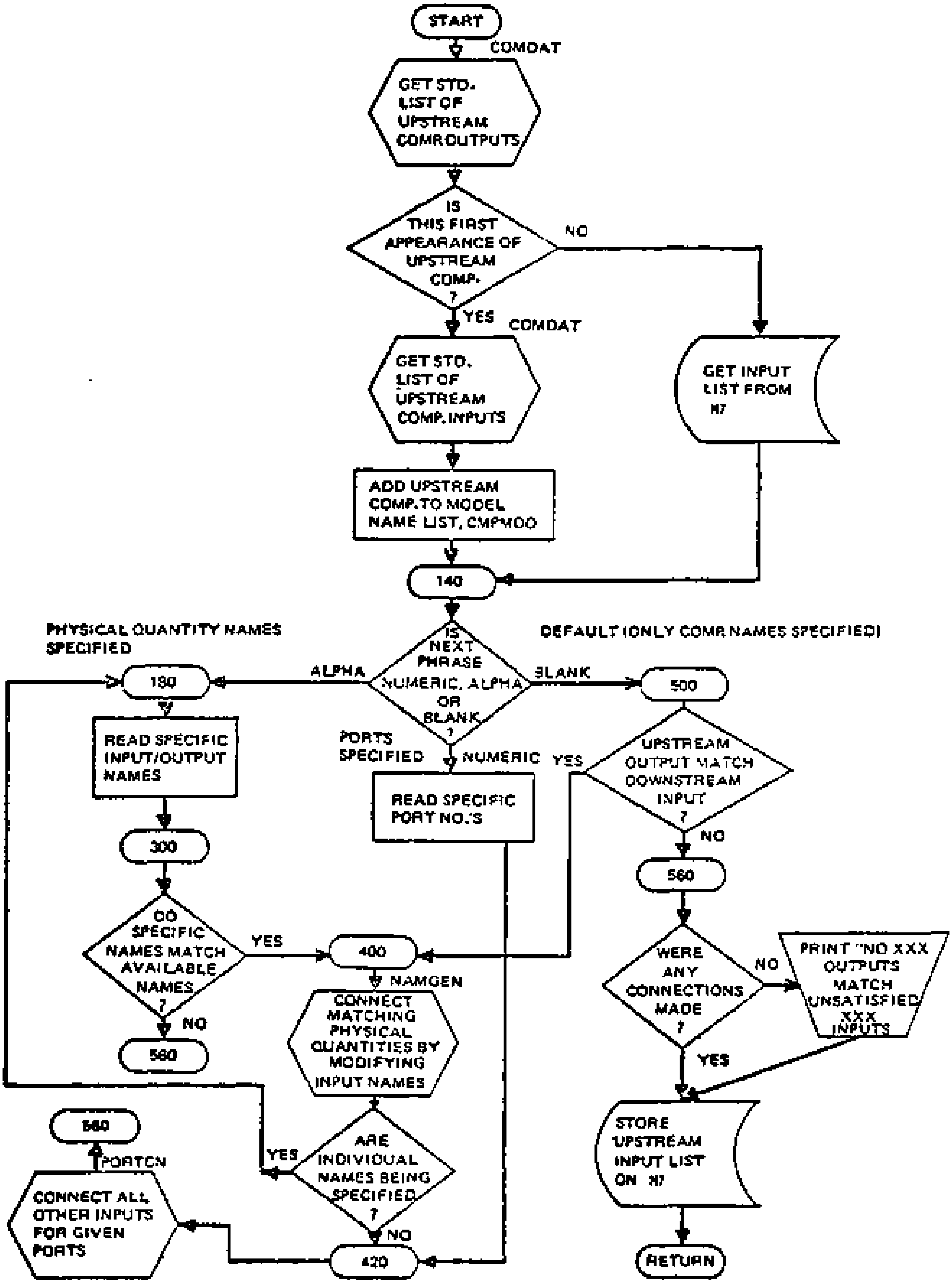

START
COMDAT
GET STD. LIST OF UPSTREAM COMP. OUTPUTS
IS THIS FIRST APPEARANCE OF UPSTREAM COMP. ?
NO
YES
COMDAT
GET STD. LIST OF UPSTREAM COMP. INPUTS
GET INPUT LIST FROM N7
ADD UPSTREAM COMP. TO MODEL NAME LIST, CMPMOD
140
PHYSICAL QUANTITY NAMES SPECIFIED
DEFAULT (ONLY COMP. NAMES SPECIFIED)
IS NEXT PHRASE NUMERIC, ALPHA OR BLANK ?
ALPHA
BLANK
180
500
READ SPECIFIC INPUT/OUTPUT NAMES
PORTS SPECIFIED
NUMERIC
READ SPECIFIC PORT NO.'S
YES
UPSTREAM OUTPUT MATCH DOWNSTREAM INPUT ?
NO
300
560
DO SPECIFIC NAMES MATCH AVAILABLE NAMES ?
YES
400
NAMGEN
NO
560
CONNECT MATCHING PHYSICAL QUANTITIES BY MODIFYING INPUT NAMES
WERE ANY CONNECTIONS MADE ?
NO
PRINT "NO XXX OUTPUTS MATCH UNSATISFIED XXX INPUTS"
YES
560
PORTCN
ARE INDIVIDUAL NAMES BEING SPECIFIED ?
YES
NO
STORE UPSTREAM INPUT LIST ON N7
CONNECT ALL OTHER INPUTS FOR GIVEN PORTS
420
RETURN

Figure 2.2-5. Subroutine INCOM - Macro Flow Diagram

(1) the collection of input name lists for each model component, stored on M7; (2) the list of model component names, CMPMOD; and (3) the location sequence of the model components, stored in SEQA. These lists describe all connections that have been made between standard components, the component names, and their location sequence in the model description. Figure 2.2-6 gives a macro flow diagram of the ENDMOD subroutine.

The source code for the subroutine calls is generated by the routines CALLCP and ENDCOM for standard components. This source code is temporarily stored on SCRTCH12. Lists of the state, variable, and parameter names contained in the model are also generated at this time and added to SCRTCH8, SCRTCH11, and SCRTCH10, respectively. These tasks for all system model components and any direct FORTRAN STATEMENTS, are completed when statement number 90 of ENDMOD is reached.

The source code statements for EQMO are next written onto SCRTCH9. The subroutines COMGEN and TABGEN are used to generate common statements for the model states, variables, parameters, and tables. The calls to standard components are transferred from SCRTCH12 to SCRTCH9 and the VARSET and RATSET entry point statements are added to SCRTCH9 to complete the source code for EQMO.

At ENDMOD statement number 700 the generation of subroutine DATAIN begins. The statements in DATAIN provide default values for the integrator error controls and the value of .99999 for all model parameters. If tables are present in the models, the routine TABDAT generates the common /CTABLE/ containing the single array TABLES which is used to load tabular data into the model. TABDAT also loads the arrays, TABNAM, MAXDIM, and LOCTAB with the table names, maximum dimensions, and pointers that are used in the table data input process.

At ENDMOD statement number 860, SCRTCH12 is rewound and the start of the Input Requirement List for the model is placed on it. Subroutine TABCAL is called to place the table information in this list.

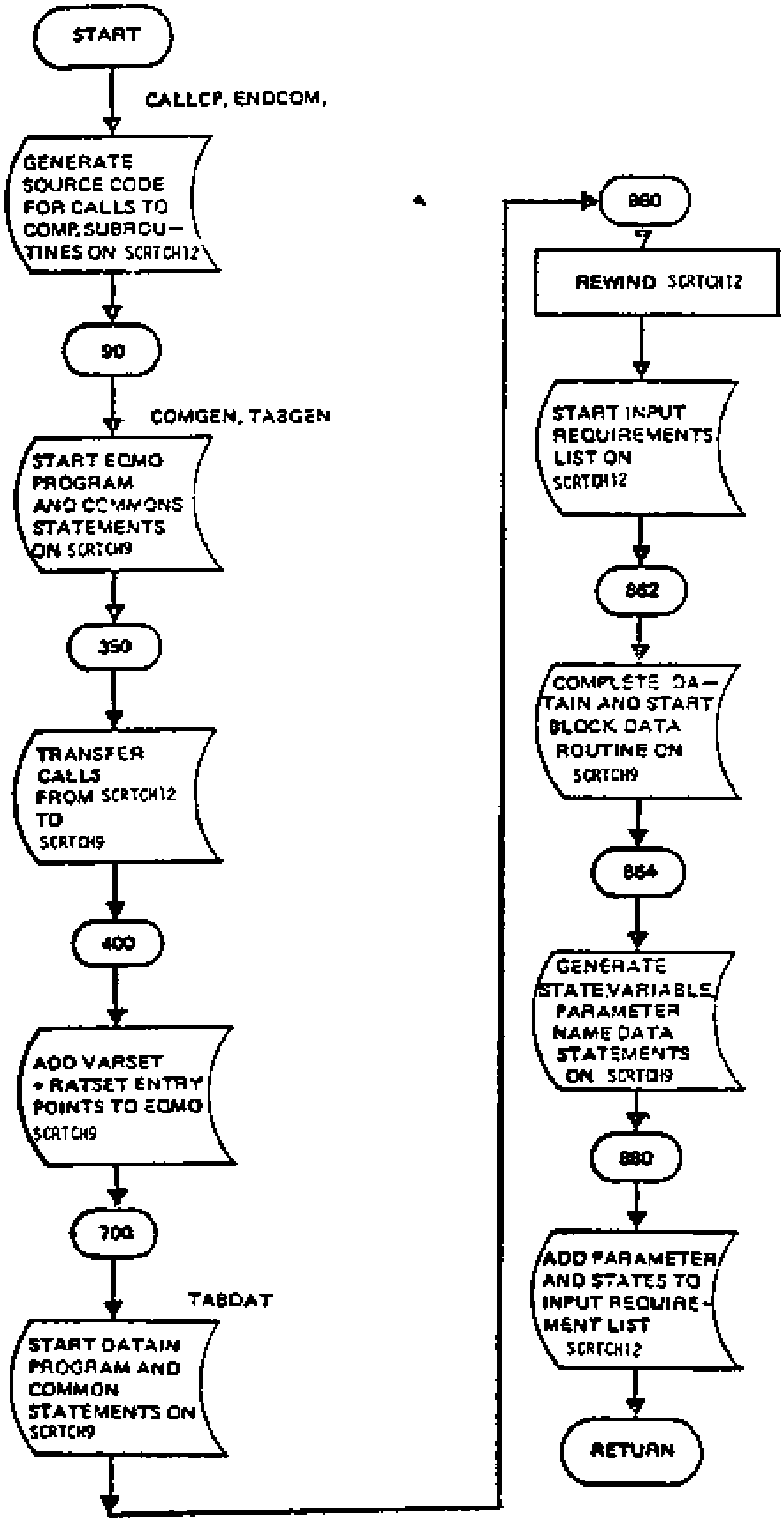

START
CALLCP, ENDCOM,
GENERATE SOURCE CODE FOR CALLS TO COMP. SUBROUTINES ON SCRTCH12
90
COMGEN, TABGEN
START EOMO PROGRAM AND COMMONS STATEMENTS ON SCRTCH9
350
TRANSFER CALLS FROM SCRTCH12 TO SCRTCH9
400
ADD VARSET + RATSET ENTRY POINTS TO EOMO SCRTCH9
700
TABDAT
START DATAIN PROGRAM AND COMMON STATEMENTS ON SCRTCH9
860
REWIND SCRTCH12
START INPUT REQUIREMENTS LIST ON SCRTCH12
862
COMPLETE DATAIN AND START BLOCK DATA ROUTINE ON SCRTCH9
864
GENERATE STATE VARIABLE, PARAMETER NAME DATA STATEMENTS ON SCRTCH9
880
ADD PARAMETER AND STATES TO INPUT REQUIREMENT LIST SCRTCH12
RETURN

Figure 2.2-6. Subroutine ENDMOD - Macro Flow Diagram

The BLOCK DATA MODEL routine source code is then added to SCRTCH9. The routine COMEQU is called once for each of the state, variable, and parameter name lists. This routine generates additional name arrays and equivalence statements whenever the number of names in a list exceeds 108. This is necessary to accommodate a compiler limitation of only 19 continuation cards in a single data statement. The NAMARY routine is used to transfer the state, variable, and parameter names from SCRTCHs 8, 11, and 10 into source code data statements on SCRTCH9. The final task of the ENDMOD subroutine is to add the parameter and state names of the model to the Input Requirement List on SCRTCH12.

2.2.6 FORTRAN STATEMENTS Command Execution

The FORTRAN STATEMENTS command allows FORTRAN source statements to be inserted directly into the system model. When this command phrase is encountered, a component name of FORT is added to the model component name list. Subsequent lines of instructions are then placed on the source file, SCRTCH9. The first phrase of each subsequent line of instruction is compared with the SIM-WEST command phrases. When a recognizable command is encountered, the direct FORTRAN mode terminates and the word FORT is written onto SCRTCH9 to mark the end of that block of FORTRAN statements. The recognized command is then executed.

Tests are included in the ENDMOD routine to provide special handling of any "FORT" components. If the ENDMOD routine encounters a FORT component while generating calls to standard components, it transfers the FORTRAN source statements from SCRTCH9 to SCRTCH12 thus placing them in the proper sequence in the model equation subroutine, EQMO.

2.3 MODEL GENERATION SOURCE LISTINGS

Compilation listings of the source code for the model generation program follows. One of the subroutines, COMORD is not currently used in the program. Several other subroutines such as NXTPH, KOMSTR and READMS are used in several of the programs and will be found in the source listings for the FILOAD program (Section 4.3). The names of the model generation routines, listed in alphabetical order, are:

BLKDAT
CALLCP
COMEQU
COMGEN
COMORD
CONNCT
EASY
ENDCOM
ENDMOD
HLINE
IJBIT
IJBIT1
INCOM
LINE
LISTSC
NAMARY
NAMGEN
NEWCOM
ORDER
PORTCN
SCHEMA
SYMBOL
TABCAL
TABDAT
TABGEN
VLINE

```
BLOCK DATA

STORAGE USED  CODE(1) 000000; DATA(0) 000000; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   COCINP 000012
 0004   COCOUT 000012
 0005   COCCRI 000012
 0006   COC    000010


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0006 I 000006 IOCAN      0006 I 000007 IXOC      0006 I 000005 LOCOC     0006 I 000002 NOC       0006 I 000004 NOCCR
 0006 I 000000 NOCIN      0006 I 000003 NOCMOD    0006 I 000001 NOCOUT    0005   000000 OCCRIT    0003   000000 OCINPT
 0004   000000 OCOUTP



00101       1*          BLOCK DATA                                                              000000
00102       2*          COMMON/COCINP/OCINPT(10)/COCOUT/OCOUTP(10)/COCCRI/OCCRIT(10)            000000
00103       3*          COMMON/COC/NOCIN,NOCOUT,NOC,NOCMOD,NOCCR,LOCOC,IOCAN,IXOC               000000
00103       4*     C    DATA OCINPT/10*(1H )/,OCOUTP/10*(1H )/                                  000000
00104       5*          DATA NOCIN/0/,NOCOUT/0/,NOC/-1/,NOCMOD/-1/,NOCCR/0/,LOCOC/-1/           000000
00104       6*         1,IOCAN/0/,IXOC/1/                                                       000000
00105       7*          END = BLKDAT  *****************************                             000000
```

```
SUBROUTINE CALLCP    ENTRY POINT 000445

STORAGE USED  CODE(1) 000514; DATA(0) 000122; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CIO     000003
 0004   CIAD    000003
 0005   COPCER  000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0006   PUTCOD
 0007   STRMOV
 0010   COMDAT
 0011   LINE
 0012   NAMGEN
 0013   GETI
 0014   NNCOD$
 0015   NWDU$
 0016   NIO2$
 0017   NIO1$
 0020   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000046 100L       0000     000040 101F       0001     000034 133G       0001     000112 154G       0001     000121 162G
 0001     000225 201G       0001     000234 207G       0001     000407 250G       0001     000170 300L       0000     000050 303F
 0000     000056 305F       0001     000270 320L       0001     000301 330L       0000     000057 340F       0001     000371 400L
 0000     000060 405F       0000     000026 71F        0001     000044 80L        0000 D 000020 ANAME      0000 D 000014 BLNK
 0000 D 000000 CALLS      0000 D 000012 COMMA      0000 I 000016 I          0003 I 000002 IOTAG      0000     000103 INJP$
 0003     000000 IREAD      0003 I 000001 IWRITE     0000 D 000010 NEWCMP     0000 I 000025 NO         0005     000002 NOP
 0004 I 000000 NOTAB      0003 I 000022 NOUT       0005 I 000001 NOV        0005 I 000000 NOX        0000 I 000017 NTAB
 0004 D 000001 TABNAM     0000 D 000023 TYPE       0000 D 000004 XDOT


00100      1*      CCALLCP                                                                         000000
00101      2*            SUBROUTINE CALLCP(COMNAM,NOCOMP,SOURCE,ISOUR,IVRSET,OUTPUT)                000000
00101      3*      C  VERSION 2.                   REVISED  DEC 15 1975                            000000
00101      4*      C  PURPOSE   TO INITIATE CALL GENERATION FOR STD. BCS COMPONENTS                000000
00101      5*      C      CALL SEQUENCE  COMNAM - COMPONENT NAME                                   000000
00101      6*      C                      NOCOMP - COMPONENT NUMBER                                000000
00101      7*      C                     SOURCE - SOURCE CODE ARRAY                                000000
00101      8*      C                     ISOUR  - SOURCE CODE ARRAY POINTER                        000000
00101      9*      C                     IVRSET - ARRAY CONTAINING VARSET,RATSET INFORMATIO        000000
00101     10*      C                      OUTPUT - WORK ARRAY FOR OUTPUT   TABLE NAMES             000000
00103     11*                       IMPLICIT DOUBLE PRECISION (A-Z)                                000000
00104     12*                       IMPLICIT INTEGER (I,J,K,L,M,N)                                 000000
```

```
00105          COMMON/CIO/IREAD,IWRITE,IDIAG/CTAB/NOTAB,TABNAM(1)                        000000
00106          COMMON /CORDER/NOX,NOV,NOP                                                000000
00107          DIMENSION IVRSET(1),SOURCE(8)                                             000000
00107         1,CALLS(2),OUTPUT(1),XDOT(2)                                               000000
00110          DOUBLE PRECISION NEWCMP, IVRSET                                           000000
00111          DATA NEWCMP/12HNEW COMPNT  /,COMMA/12H,                                   000000
00114          DATA BLNK/12H                                                             000000
00116          DATA CALLS/24H      CALL      (                                           000000
00120          DATA XDOT/24H,XDOT(    )   ,INT(                                          000000
00120    C --->      SAVE NO. OF VARIABLES AND STATES BEFORE COMPONENT IS FORMED         000000
00122          I=4*NOCOMP-3                                                              000000
00123          CALL PUTCOD(I,IVRSET,NOV)                                                 000004
00124          I=4*NOCOMP-1                                                              000011
00125          CALL PUTCOD(I,IVRSET,NOX)                                                 000014
00126          WRITE(12,71)COMNAM                                                        000021
00131    71    FORMAT('C'/'C',20X,'COMPONENT  ',A4/'C')                                  000034
00131    C --->      LOAD SOURCE WITH CALL XXX                                           000034
00132          DO 100 I=1,8                                                              000034
00135          IF(I.LE.2) GO TO 80                                                       000034
00137          SOURCE(I)=BLNK                                                            000040
00140          GO TO 100                                                                 000042
00141    80    SOURCE(I)=CALLS(I)                                                        000044
00142    100   CONTINUE                                                                  000051
00142    C --->      LOAD STANDARD COMPONENT SUBROUTINE NAME                             000051
00144          CALL STRMOV(COMNAM,1,2,SOURCE,12)                                         000051
00145          ISOUR=15                                                                  000060
00145    C --->      GET LIST OF TABLES FOR COMPONENT                                    000060
00146          CALL COMDAT(COMNAM,12HTABS        ,NTAB,OUTPUT)                           000062
00146    C --->      TEST IF TABLES ARE REQUIRED BY SUBROUTINE                           000062
00147          IF(NTAB.LE.0) GO TO 300                                                   000070
00147    C --->      ADD TABLE ARGUMENTS TO CALL SEQUENCE                                000070
00151          IF(IDIAG.GT.60)WRITE(IWRITE,101)(OUTPUT(I),I=1,NTAB)                      000073
00160    101   FORMAT(' CALLCP-TABLES'/(1X,6A10))                                        000121
00160    C --->      SCAN REQUIRED TABLES                                                000121
00161          DO 200 I=1,NTAB                                                           000121
00161    C --->      CONSTRUCT TABLE NAME                                                000121
00164          ANAME=OUTPUT(I)                                                           000121
00165          CALL STRMOV(COMNAM,1,4,ANAME,4)                                           000123
00165    C --->      ADD TABLE NAME TO TABLE LIST                                        000123
00166          NOTAB=NOTAB+1                                                             000132
00167          TABNAM(NOTAB)=ANAME                                                       000136
00170          IF(I.GT.1) CALL LINE(0,SOURCE,ISOUR,COMMA,1,12)                           000140
00172          CALL LINE(0,SOURCE,ISOUR,ANAME,6,12)                                      000154
00173    200   CONTINUE                                                                  000170
00173    C --->      GET LIST OF OUTPUT QUANTITIES FOR COMPONENT                         000170
00175    300   CALL COMDAT(COMNAM,12HOUTP        ,NOUT,OUTPUT)                           000170
00176          IF(IDIAG.GT.60)WRITE(IWRITE,303)(OUTPUT(I),I=1,NOUT)                      000175
00205    303   FORMAT(' CALLCP-OUTPUTS'/(1X,6A10))                                       000234
00205    C --->      SCAN OUTPUT QUANTITIES                                              000234
00206          DO 400 I=1,NOUT                                                           000234
00206    C --->      CONSTRUCT OUTPUT QUANTITY SPECIFIC NAME                             000234
00211          CALL MANGEN(OUTPUT(I),COMNAM,ANAME)                                       000234
00211    C --->      GET 10TH CHARACTER IN STD. NAME TO DETERMINE IF QUANTITY            000234
00211    C  IS A STATE OR A VARIABLE                                                     000234
00212          CALL GETT(OUTPUT(I),10,TYPE)                                              000243
00212    C --->      TEST FOR STATE OR VARIABLE                                          000243
00213          IF(TYPE.NE.BLNK) GO TO 320                                                000252
```

```
00213     70*     C --->       INCREMENT VARIABLE COUNTER                                            000252
00215     71*           NOV=NOV+1                                                                   000255
00216     72*           WRITE(11,305)ANAME                                                          000260
00221     73*     305   FORMAT(A10)                                                                 000266
00222     74*           GO TO 330                                                                   000266
00222     75*     C --->       INCREMENT STATE COUNTER                                               000266
00223     76*     320   NOX=NOX+1                                                                   000270
00224     77*           WRITE(8,305)ANAME                                                           000272
00227     78*     330   IF(NTAB.GT.0.OR.I.GT.1) CALL LINE(0,SOURCE,ISOUR,COMMA,1,12)                000301
00227     79*     C --->       ADD OUTPUT NAME TO CALL SEQUENCE                                      000301
00231     80*           CALL LINE(0,SOURCE,ISOUR,ANAME,6,12)                                        000320
00232     81*           IF(ITYPE.EQ.BLNK) GO TO 400                                                 000330
00232     82*     C --->       CONVERT CURRENT NO. OF STATE TO BCD                                   000330
00234     83*           ENCODE(3,340,NO)NOX                                                         000333
00237     84*     340   FORMAT(I3)                                                                  000342
00237     85*     C --->       LOAD CURRENT STATE NO. AS RATE SUBSCRIPT                              000342
00240     86*           CALL STRMOV(NO,1,3,XDOT,7)                                                  000342
00240     87*     C --->       LOAD CURENT STATE NO. AS INT SUBCRIPT                                 000342
00241     88*           CALL STRMOV(NO,1,3,YDOT,16)                                                 000351
00242     89*           CALL LINE(0,SOURCE,ISOUR,XDOT,19,12)                                        000360
00243     90*     400   CONTINUE                                                                    000373
00245     91*           IF(IDIAG.GE.50)WRITE(IWRITE,405)SOURCE                                      000373
00254     92*     405   FORMAT(' CALLCP-SOURCE'/(1X,6A10))                                          000412
00254     93*     C --->       SAVE NO. OF VARIABLES AND STATES AFTER COMPONENT IS FORMED            000412
00255     94*           I=4*NOCOMP-2                                                                000412
00256     95*           CALL PUTCOD(I,IVRSET,NOV)                                                   000415
00257     96*           I=4*NOCOMP                                                                  000422
00260     97*           CALL PUTCOD(I,IVRSET,NOX)                                                   000424
00261     98*     500   CONTINUE                                                                    000431
00262     99*           RETURN                                                                      000431
00263    100*           END $ CALLCP  ******************************                                000513
```

```
SUBROUTINE COMEQU    ENTRY POINT 000064

STORAGE USED  CODE(1) 000074; DATA(0) 000040; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NWDU$
 0004   NIO2$
 0005   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000017 112G      0000   000004 81F      0000 I 000001 I        0000   000026 INJP$     0000 I 000002 J
 0000 I 000003 K         0000 I 000000 NEXT


00100       CCOMEQU                                                          000000
00101            SUBROUTINE COMEQU(NAME,N)                                   000000
00101       C  VERSION 1.0                     REVISED  AUG 28 1975          000000
00101       C  PURPOSE   CREATE EQUIVALENT NAME ARRAYS TO ALLOW DATA STATEMENTS  000000
00101       C              TO LOAD NAME LISTS EXCEEDING 108 NAMES.          000000
00101       C  CALL SEQUENCE   NAME - NAME OF ARRAY TO BE EXTENDED           000000
00101       C                  N    - NUMBER OF NAMES IN LIST                000000
00101       C  DESIGNED BY   J.D. BURROUGHS               AUG 1975           000000
00103                          IMPLICIT DOUBLE PRECISION (A-Z)               000000
00104                          IMPLICIT INTEGER (I,J,K,L,M,N)                000000
00105                          DOUBLE PRECISION NAME                         000000
00105       C --->       CALCULATE NO. OF EXTENSIONS REQUIRED                000000
00106            NEXT=(N-1)/108                                              000000
00107            IF(NEXT.LE.0)RETURN                                         000004
00107       C --->       ADD AN EQUIVALENCE STATEMENT FOR EACH EXTENSION REQD.  000004
00111            DO 100 I=1,NEXT                                             000012
00114            J=108*I+1                                                   000017
00114       C ---        CALCULATE NO. OF WORDS IN EXTENSION                 000017
00115            K=N-J+1                                                     000023
00116            IF(K.GT.108)K=108                                           000026
00123            WRITE(9,81)NAME,I,K,NAME,J,NAME,I                           000034
00131       81   FORMAT(6X,'DOUBLE PRECISION ',A5,I2,'(',I3,')'/             000052
00131           1 6X,'EQUIVALENCE(',A5,'(',I5,'),',A5,I2,'(1)')              000052
00132       100  CONTINUE                                                    000052
00134            RETURN                                                      000052
00135            END $ COMEQU  ******************************                 000073
```

```
SUBROUTINE CONGEN    ENTRY POINT 000252

STORAGE USED  CODE(1) 000276; DATA(0) 001225; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003    STPPOV
 0004    ISCAN
 0005    LINE
 0006    NRFW$
 0007    NRDU$
 0010    NI02$
 0011    NWDU$
 0012    NI03$
 0013    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000   001172 105F      0001   000112 110L      0001   000017 125G      0001   000053 136G      0001   000062 144G
 0001   000166 274G      0001   000176 202G      0000 D 001162 ANAME     0000 D 001154 BLNCOM    0000 D 001156 BLNK
 0000 D 001152 COMMLT    0000 I 001170 I         0000   001212 INJP$     0000 I 000000 INT       0000 I 001151 INTEG
 0004 I 000000 ISCAN     0000 I 001171 ISOUR     0000 I 001165 J         0000 I 001166 K         0000 I 001167 NAMES
 0000 I 001164 NEXT      0000 D 001160 REALLT    0000 D 000021 RNAMES    0000 D 000001 SOURCE


00100        CCONGEN                                                                                000000
00101              SUBROUTINE CONGEN(N,CNAME,NUNIT,IUNIT)                                           000000
00101        C  VERSION 2.1 C  VERSION 2.                            REVISED  OCT 7 1976            000000
00101        C  PURPOSE  GENERATE COMMON STATEMENT GIVEN NAMES OF VARIABLES                         000000
00101        C              STORED IN THE COMMON                                                    000000
00101        C  CALL SEQUENCE  N       - NO. OF VARIABLES IN COMMON                                 000000
00101        C                 CNAME   - COMMON NAME. (2 CHARACTERS)                                000000
00101        C                 NUNIT   - FILE NO. CONTAINING NAMES                                  000000
00101        C                 IUNIT   - FILE NO. TO WHICH SOURCE CODE IS TO                        000000
00101        C                              BE WRITTEN.                                             000000
00103                              IMPLICIT DOUBLE PRECISION (A-Z)                                  000000
00104                              IMPLICIT INTEGER (I,J,K,L,M,N)                                   000000
00105              DIMENSION SOURCE(8),RNAMES(300)                                                  000000
00106              DATA INTEG/6HIJKLMN/                                                             000000
00106        C     LITERAL 'POOL' TO SATISFY DBLE PRECSN ASSGNMNT STMNTS                            000000
00110              DATA COMMLT/12H      COMM   /                                                    000000
00112              DATA BLNCOM/12HON /  /     /                                                     000000
00114              DATA BLNK/12H            /                                                       000000
00116              DATA REALLT/12H      REAL   /                                                    000000
00120              REWIND NUNIT                                                                     000000
00120        C ---   CALC. NO. OF EXTENSIONS TO COMMON STATEMENT REQ'D                              000000
00121              IEX=0                                                                            000002
00122              ANAME = BLNK                                                                     000003
```

```
00123      24*              NEXT=(N-1)/156+1                                                      000005
00124      25*              DO 400 J=1,NEXT                                                       000013
00124      26*        C ---    COMMON EXTENSION COUNTER                                           000013
00127      27*              K=J-1                                                                 000017
00127      28*        C ---    NUMBER OF NAMES PER EXTENSION                                      000017
00130      29*              NAMES=N-K*156                                                         000022
00130      30*        C ---    LIMIT NO. OF NAMES PER COMMON TO 156                               000022
00131      31*              NAMES=MIN0(NAMES,156)                                                 000025
00131      32*        C ---    GENERATE COMMON STATEMENT                                          000025
00131      33*        C --->       FORM COMMON NAME                                               000025
00132      34*              SOURCE(1)=COMPLT                                                      000032
00133      35*              SOURCE(2)=BLNCOM                                                      000034
00134      36*              CALL STRMOV(CNAME,1,2,SOURCE,15)                                      000036
00135      37*              DO 100 I=3,8                                                          000053
00140      38*        100   SOURCE(I)=BLNK                                                        000053
00142      39*              ISOUR=10                                                              000055
00142      40*        C --->       SCAN NAMES                                                     000055
00143      41*              DO 200 I=1,NAMES                                                      000062
00146      42*              READ(MUNIT,105)ANAME                                                  000062
00151      43*        105   FORMAT(8A10)                                                          000070
00151      44*        C ---    TEST FOR INTEGER NAMES                                             000070
00152      45*              IF(ISCAN(ANAME,1,1,INTEG,1,6,K).EQ.0)GO TO 110                        000070
00154      46*              INT=INT+1                                                             000103
00155      47*              RNAMES(INT)=ANAME                                                     000107
00156      48*        110   IF(I.GT.1) CALL LINE(0,SOURCE,ISOUR,12H,           ,1,IUNIT)          000112
00160      49*              CALL LINE(0,SOURCE,ISOUR,ANAME,6,JUNIT)                               000125
00161      50*        200   CONTINUE                                                              000137
00163      51*              WRITE(JUNIT,105)SOURCE                                                000137
00166      52*        400   CONTINUE                                                              000151
00166      53*        C ---    TEST IF INTEGER NAMES OCCURED                                      000151
00170      54*              IF(INT.EQ.0)RETURN                                                    000151
00172      55*              SOURCE(1)=REALLT                                                      000156
00173      56*              DO 500 I=2,8                                                          000160
00176      57*        500   SOURCE(I)=BLNK                                                        000166
00176      58*        C ---    SCAN INTEGER NAMES                                                 000166
00200      59*              ISOUR=12                                                              000170
00201      60*              DO 600 I=1,INT                                                        000176
00204      61*              IF(I.NE.1)CALL LINE(0,SOURCE,ISOUR,12H,          ,1,IUNIT)            000176
00206      62*              CALL LINE(0,SOURCE,ISOUR,RNAMES(I),6,JUNIT)                           000212
00207      63*        600   CONTINUE                                                              000226
00211      64*  /           WRITE(JUNIT,105)SOURCE                                                000226
00214      65*              RETURN                                                                000236
00215      66*              END & COMGEN  ******************************                          000275
```

```
SUBROUTINE COMORD     ENTRY POINT 000504

STORAGE USED  CODE(1) 000525; DATA(0) 004617; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CSEQ    000003
 0004   CIO     000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0005   GETCOD
 0006   PUTCOD
 0007   RCAIMS
 0010   KOMSTR
 0011   STPMOV
 0012   IJPITI
 0013   OKOER
 0014   NWDU$
 0015   NIO2$
 0016   NIOI$
 0017   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000045 100L      0001     000014 121G      0001     000065 136G      0001     000074 144G      0001     000143 157G
 0001     000254 172G      0000     003024 201F      0001     000260 214G      0001     000277 224G      0001     000333 235G
 0001     000191 249G      0001     000413 262G      0001     000430 272G      0001     000242 260L      0001     000250 300L
 0001     000461 303G      0001     000259 360L      0001     000271 400L      0001     000373 540L      0000     003045 551F
 0001     000422 600L      0003     003073 621F      0001     000042 ASL       0000 D 002776 BLNK      0000 D 003002 CHARS
 0000 D 003007 COMP        0000 D 003013 COMPS       0000 D 000000 CONARR      0000 D 003000 FORLT       0000 I 003004 I
 0000 I 003021 IM          0000 I 003011 ICOMP       0004     000002 INDIAG    0000 I 003022 IE          0000 I 003026 IERROR
 0000     004573 INJPS     0004     000000 IREAD     0000 I 003023 IRCARR      0000 I 002466 ISEQ        0004 I 000001 IWRITE
 0000 I 003120 INI         0000 I 003740 IW2         0000 I 003016 J           0000 I 003017 JCOMP       0000 I 003015 K
 0010 I 000000 KOMSTR      0000 I 003005 LOK         0000 I 003012 NINPUT      0003 I 000000 NSEQ        0000 I 003006 NWORDS
 0003 D 000001 SEQA        0000 D 003120 W1          0000 D 003743 W2


00100      1*     CCOMORD                                                                             000002
00101      2*           SUBROUTINE COMORD(CMPMOD,NOCOMP,INPUTS)                                       000002
00101      3*     C  VERSION 2.                        REVISED  SEPT 5 1975                             000002
00101      4*     C  PURPOSE   ORDER COMPONENTS SO THAT MODEL EQUATIONS ARE EXPLICIT                    000002
00101      5*     C  CALL SEQUENCE   CMPMOD - ARRAY CONTAINING NAMES OF MODEL COMPONENTS                000002
00101      6*     C                  NOCOMP - NUMBER OF COMPONENTS IN MODEL                             000002
00101      7*     C                  INPUT  - INPUT NAME ARRAY WORK SPACE                               000002
00101      8*     C  DESIGNED BY  J.D. BURROUGHS                     JULY 1975                          000002
00103      9*                     IMPLICIT DOUBLE PRECISION (A-Z)                                         000002
00104     10*                     IMPLICIT INTEGER (I,J,K,L,M,N)                                          000002
```

```
00105     11*                        DOUBLE PRECISION INPUTS                                       000002
00106     12*                  COMMON/CSEQ/NSEQ,SEQA(1)/CIO/IREAD,IWRITE,IDIAG                     000002
00107     13*                  DIMENSION CMPMOD(1),INPUTS(1),CONARR(667),ISCO(200),IW1(200)        000002
00107     14*                 1 ,IW2(200),W1(200),W2(200)                                          000002
00110     15*                  EQUIVALENCE(W1,IW1),(W2,IW2)                                        000002
00111     16*                  DATA BLNK/'             '/,FORLT/'FORT                              000002
00114     17*                  DATA CHARS/1205                                                     000002
00114     18*          C --->        TEST IF ALL COMPONENTS HAVE SEQUENCE NUMBERS                  000002
00116     19*                  IF(NSEQ.GE.NOCOMP)GO TO 100                                         000002
00116     20*          C ============  ASSIGN SEQUENCE NOS. TO UNSEQUENCED COMPONENTS              000002
00116     21*          C --->        SCAN ALL MODEL COMPONENTS                                     000002
00120     22*                  DO 85 I=1,NOCOMP                                                    000006
00120     23*          C --->        SKIP FORTRAN COMPONENTS                                       000006
00123     24*                  IF(CMPMOD(I).EQ.FORLT)GO TO 85                                      000014
00123     25*          C --->        GET LOCATION CODE                                             000014
00125     26*                  CALL GETCOD(3,CMPMOD(I),LOK)                                        000017
00126     27*                  IF(LOK.GT.0)GO TO 85                                                000026
00126     28*          C --->        INCREMENT SEQUENCE NO. COUNT                                  000026
00130     29*                  NSEQ=NSEQ+1                                                         000031
00131     30*                  CALL PUTCOD(NSEQ,SEQA,I)                                            000034
00132     31*          85      CONTINUE                                                            000045
00132     32*          C ============  ZERO CONNECTION ARRAY                                       000045
00134     33*          100     NWORDS=MIN0(NOCOMP*NOCOMP/60+1,60)                                  000045
00135     34*                  DO 120 I=1,NWORDS                                                   000056
00140     35*          120     CONARR(I)=0.D00                                                     000065
00140     36*          C ============  FORM CONNECTION ARRAY                                       000065
00140     37*          C --->        SCAN MODEL COMPONENTS IN CURRENT SEQUENCE                     000065
00142     38*                  COMP=BLNK                                                           000067
00143     39*                  DO 400 I=1,NSEQ                                                     000074
00143     40*          C --->        GET COMPONENT NUMBER                                          000074
00146     41*                  CALL GETCOD(I,SEQA,ICOMP)                                           000074
00146     42*          C --->        TEST FOR FORTRAN COMPONENTS                                   000074
00147     43*                  IF(CMPMOD(ICOMP).EQ.FORLT)GO TO 360                                 000104
00147     44*          C --->        GET NUMBER OF INPUTS TO ITH COMPONENT                         000104
00151     45*                  CALL GETCOD(5,CMPMOD(ICOMP),NINPUT)                                 000107
00151     46*          C --->        SKIP COMPONENTS WITH ZERO INPUTS                              000107
00152     47*                  IF(NINPUT.LE.0)GO TO 400                                            000116
00152     48*          C ============  GET INPUT LIST FOR ITH COMPONENT                            000116
00154     49*                  CALL READMS(7,INPUTS,NINPUT,ICOMP)                                  000121
00155     50*                  COMPS=BLNK                                                          000127
00155     51*          C --->        SCAN INPUTS                                                   000127
00156     52*                  DO 300 K=1,NINPUT                                                   000143
00156     53*          C --->        TEST TO IGNORE STATE INPUTS                                   000143
00161     54*                  IF(KOMPST(INPUTS(K),10,1,CHARS,1).EQ.0)GO TO 300                    000143
00161     55*          C --->        GET NAME OF COMPONENT PROVIDING INPUT                         000143
00163     56*                  CALL STRMOV(INPUTS(K),4,3,COMP,1)                                   000156
00163     57*          C ---         TEST TO SKIP PARAMETERS                                       000156
00164     58*                  IF(COMP.EQ.BLNK          )GO TO 300                                 000167
00164     59*          C --->        TEST TO SKIP SEARCH FOR SEQUENTIAL INPUTS FROM SAME COMPONEN  000167
00166     60*                  IF(COMP.EQ.COMPS)GO TO 300                                          000172
00173     61*                  COMPS=COMP                                                          000177
00173     62*          C ============  SCAN COMPONENTS TO LOCATE SEQUENCE NO. OF INPUT             000177
00173     63*                  DO 200 J=1,NSEQ                                                     000204
00174     64*                  CALL GETCOD(J,SEQA,JCOMP)                                           000204
00174     65*          C --->        COMPARE EACH COMPONENT WITH INPUT COMPONENT                   000204
00175     66*                  IF(KOMPST(COMP,1,3,CMPMOD(JCOMP),1).EQ.0)GO TO 280                  000211
00177     67*          200     CONTINUE                                                            000231
```

```
00204     68*           WRITE(IWRITE,201)COMP,CMPMOD(ICOMP)                                          000231
00205     69*     201   FORMAT(/5X,15H*** WARNING ***,5X,'CAN"T IDENTIFY   ',A4,'   AS A            000240
00205     70*          1VALID INPUT COMPONENT TO   ',A4/)                                             000240
00206     71*           GO TO 300                                                                     000240
00206     72*     C --->        SET I J BIT = 1                                                       000240
00207     73*     280   CALL IJBIT1(CONARR,I,J,NSEQ)                                                  000242
00210     74*     300   CONTINUE                                                                      000252
00212     75*           GO TO 400                                                                     000252
00212     76*     C ============  FOR FORTRAN COMPONENTS - REQUIRE ALL PREVIOUS COMPONENTS             000252
00213     77*     360   DO 380 J=1,I                                                                  000254
00216     78*           CALL IJBIT1(CONARR,I,J,NSEQ)                                                  000264
00217     79*     380   CONTINUE                                                                      000272
00221     80*     400   CONTINUE                                                                      000272
00221     81*     C ============  LOAD SEQUENCE VECTOR                                                000272
00223     82*           DO 420 I=1,NSEQ                                                               000272
00226     83*     420   ISEQ(I)=I                                                                     000277
00226     84*     C ============  ORDER COMPONENTS                                                    000277
00230     85*           CALL ORDER(NSEQ,ISEQ,CONARR,IW1,IW2,IERROR,IB,IE)                             000303
00231     86*           IF(IERROR.NE.0)GO TO 600                                                      000315
00231     87*     C --->        TEST FOR SUCCESSFUL ORDERING                                          000315
00233     88*           NWORDS=NSEQ/5+1                                                               000317
00233     89*     C ============  SAVE COPY OF SEQUENCE ARRAY                                         000317
00234     90*           DO 500 I=1,NWORDS                                                             000326
00237     91*           W1(I)=SEQA(I)                                                                 000333
00240     92*     500   CONTINUE                                                                      000335
00240     93*     C --->        SET REARRANGEMENT COUNTER                                             000335
00242     94*           IREARR=0                                                                      000335
00243     95*     C --->        SCAN COMPONENTS                                                       000335
00243     96*           DO 540 I=1,NSEQ                                                               000341
00243     97*     C --->        TEST IF SEQUENCE HAS BEEN MODIFIED                                    000341
00246     98*           IF(ISEQ(I).EQ.I)GO TO 540                                                     000341
00246     99*     C --->        INCREMENT REARRANGEMENT COUNTER                                       000341
00250    100*           IREARR=IREARR+1                                                               000344
00250    101*     C --->        GET COMPONENT NUMBER                                                  000344
00251    102*           CALL GETCOD(ISEQ(I),W1,JCOMP)                                                 000347
00251    103*     C --->        SAVE COMPONENT NAMES OF THOSE COMPONENTS WHOSE SEQUENCE HAS            000347
00252    104*           W2(IREARR)=CMPMOD(JCOMP)                                                      000363
00253    105*           CALL PUTCOD(I,SEQA,JCOMP)                                                     000365
00254    106*     540   CONTINUE                                                                      000374
00254    107*     C --->        TEST IF REARRANGEMENT OCCURED                                         000374
00256    108*           IF(IREARR.LE.0)RETURN                                                         000374
00260    109*           WRITE(IWRITE,551)(W2(I),I=1,IREARR)                                           000402
00266    110*     551   FORMAT(/5X,14H*** NOTICE ***,5X,'THE SEQUENCE OF THE FOLLOWING COM            000416
00266    111*          1PONENTS HAS BEEN ALTERED TO FORM AN EXPLICIT MODEL'//20(2X,A4)//)             000416
00267    112*           RETURN                                                                        000416
00267    113*     C ============  SCAN COMPONENTS THAT CAUSED IMPLICIT LOOP                           000416
00270    114*     600   J=0                                                                           000422
00271    115*           DO 620 I=IB,IE                                                                000422
00274    116*           CALL GETCOD(IW2(I),SEQA,JCOMP)                                                000430
00275    117*           J=J+1                                                                         000441
00275    118*     C --->        SAVE NAMES OF COMPONENTS IN IMPLICIT LOOP                             000441
00276    119*           W1(J)=CMPMOD(JCOMP)                                                           000445
00277    120*     620   CONTINUE                                                                      000450
00301    121*           WRITE(IWRITE,621)(W1(I),I=1,J)                                                000450
00307    122*     621   FORMAT(/5X,15H*** WARNING ***,5X,'THE FOLLOWING COMPONENTS FORM AN            000464
00307    123*          1 IMPLICIT LOOP.  MODEL RESULTS WILL BE INVALID.'//20(2X,A4)//)                000464
00310    124*           RETURN                                                                        000464
00311    125*           END @ COMORD  ******************************                                  000524
```

```
SUBROUTINE CONNCT    ENTRY POINT 000756


STORAGE USED  CODE(1) 001013; DATA(0) 000125; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CIO     000003


EXTERNAL REFERENCES (BLOCK, NAME)

 0004   KOMSTR
 0005   STPMOV
 0006   FCTCOD
 0007   HLINE
 0010   VLINE
 0011   PUTT
 0012   NRCOD$
 0013   NWDU$
 0014   NIO1$
 0015   NIO2$
 0016   NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000     000034 101F        0001     000120 120L        0001     000053 121G        0001     000062 127G        0000     000027 13F
 0001     000207 135L        0001     000223 140L        0001     000236 160L        0001     000247 200L        0001     000270 220L
 0001     000254 240L        0001     000275 260L        0001     000310 280L        0001     000321 300L        0001     000347 320L
 0001     000477 333G        0001     000375 340L        0001     000422 360L        0001     000446 400L        0001     000463 420L
 0000     000060 421F        0001     000610 440L        0001     000616 500L        0001     000741 540L        0000 D 000002 ASTRSK
 0000 D 000000 BLNK        0000 I 000006 I           0000 I 000023 ICO         0000 I 000013 ICOL        0003 I 000002 IDIAG
 0000 I 000014 IDS         0000 I 000012 ILIN        0000 I 000010 ILOC        0000 I 000015 INCOL       0000   000074 INJP$
 0000 I 000007 INLIN       0000 I 000011 IPAGE       0000 I 000016 IRCOL       0003   000000 IREAD       0000 I 000021 IRLIN
 0000 I 000017 ITC         0000 I 000020 ITL         0003 I 000001 IWRITE      0000 I 000026 K           0004 I 000000 KOMSTR
 0000 I 000022 LIN         0000 I 000005 LOCCOL      0000 I 000004 LOCLIN      0000 D 000024 PAG


00100          CCONNCT                                                                                    000016
00101                SUBROUTINE CONNCT(PAGE,NPAGE,LOK,INPUTS,NOIN,COMTAB,NOCOMP)                          000016
00101          C  VERSION 2.                           REVISED  DEC 15 1975                               000016
00101          C  PURPOSE   FORM CONNECTING LINE BETWEEN TWO SPECIFIED COMPONENT                              000016
00101          C                SYMBOLS AND LABEL INPUTS                                                  000016
00101          C  CALL SEQUENCE    PAGE    - 13X56 ARRAY CONTAINING HOLLORITH                                 000016
00101          C                                  REPRESENTATION OF A PAGE                                000016
00101          C                   NPAGE   - CURRENT PAGE NO.                                             000016
00101          C                    LOK      - LOCATION OF SYMBOL TO WHICH LINE IS                        000016
00101          C                                  TO BE DRAWN                                             000016
00101          C                   INPUTS - ARRAY OF INPUT QUANTITY NAMES                                 000016
00101          C                   NOIN    - NO. OF INPUT QUANTITY NAMES                                  000016
```

```
00101     C                   CONTAB - TABLE OF ALL COMPONENT NAMES AND          000016
00101     C                                THEIR LOCATIONS                       000016
00101     C                   NOCOMP - NO. OF COMPONENTS                         000016
00103                          IMPLICIT DOUBLE PRECISION (A-Z)                   000016
00104                          IMPLICIT INTEGER (I,J,K,L,M,N)                    000016
00105                          DOUBLE PRECISION INPUTS                           000016
00106           COMMON/CIO/IREAD,IWRITE,IDIAG                                    000016
00107           DIMENSION PAGE(13,56),INPUTS(1),CONTAB(1)                        000016
00110           DATA BLNK /12H            /                                      000016
00112           DATA ASTRSK /12H******/      /                                   000016
00112     C --->      RECEIVING COMPONENT LOCATION LINE NO.                      000016
00114           LOCLIN=7*((LOK-1)/10)+4                                          000016
00114     C --->      RECEIVING COMPONENT LOCATION COL. NO.                      000016
00115           LOCCOL=(MOD(LOK-1,10)+1)*13-6                                    000026
00116           IF(IDIAG.EQ.3)WRITE(IWRITE,13)(INPUTS(I),I=1,NOIN)               000035
00125     13    FORMAT(" COMPCT-INPUTS"/(1X,6A10))                               000042
00125     C --->      SCAN COMPONENTS LIST TO LOCATE INPUT COMP.                 000062
00126           DO 100 I=1,NOCOMP                                                000062
00131           IF(KOMPSTR(INPUTS,4,4,CONTAB(I),1).EQ.0)GO TO 120                000062
00133     100   CONTINUE                                                         000100
00135           CALL STRMOV(INPUTS,4,4,INLIN,1)                                  000100
00136           WRITE(IWRITE,101)INLIN,LOK                                       000107
00142     101   FORMAT(/5X,32H *** WARNING ***  CAN"T LOCATE  ,A4,               000116
00142          1 "  AS AN INPUT COMPONENT TO LOCATION ",I4)                      000116
00142     C --->      RETURN IF INPUT COMPONENT ISN"T IN CONTAB LIST             000116
00143           GO TO 540                                                        000116
00143     C --->      GET LOCATION OF INPUT COMPONENT                            000116
00144     120   CALL GETCOD(3,CONTAB(I),ILOC)                                    000120
00144     C --->      DETERMINE PAGE OF INPUT COMPONENT                          000120
00145           IPAGE=(ILOC/100)*100                                             000131
00145     C --->      COMPARE INPUT COMP. PAGE TO CURRENT PAGE                   000131
00146           IF(IPAGE.NE.NPAGE)GO TO 420                                      000136
00146     C --->      CONVERT GENERAL PAGE LOC TO LOCAL PAGE LOC                 000136
00150           ILOC=ILOC-IPAGE                                                  000140
00153     C --->      CALC. LOC. LINE AND COL. NO. FOR INPUT COMPONENT           000140
00151           ILIN=7*((ILOC-1)/10)+4                                           000143
00152           ICOL=(MOD(ILOC-1,10)+1)*13-6                                     000152
00152     C ---       TEST FOR INPUTS FROM DOWNSTREAM COMP.                      000152
00153           IDS=0                                                            000161
00154           IF(KOMPSTR(INPUTS,8,1,BLNK,1).NE.0)IDS=1                         000162
00154     C --->     TEST IF RECEIVING COMP. AND INPUT COMP. ARE ON SAME LINE    000162
00156           IF(ILIN-LOCLIN)200,130,220                                       000175
00156     C --->      SAME LINE.  TEST IF LEFT OR RIGHT                          000175
00161     130   IF(ICOL.GE.LOCCOL)GO TO 140                                      000202
00161     C --->      SAME LINE AND INPUT IS TO LEFT                             000202
00163     135   INCOL=ICOL+6                                                     000207
00164           IRCOL=LOCCOL-5                                                   000211
00165           ITC=IRCOL-7                                                      000214
00166           ITL=2-LOCLIN                                                     000216
00167           GO TO 160                                                        000221
00167     C --->      SAME LINE AND INPUT IS TO RIGHT                            000221
00170     140   INCOL=ICOL-5                                                     000223
00171           IRCOL=LOCCOL+6                                                   000225
00172           ITC=IRCOL+1                                                      000230
00173           ITL=LOCLIN+2                                                     000232
00173     C --->      ADD HORIZONTAL LINE                                        000232
00174     160   IF(IDS.NE.0)GO TO 500                                            000236
```

```
00176              CALL HLINE(PAGE,ILIN,INCOL,IRCOL)                000237
00177              GO TO 500                                        000245
00177        C --->       INPUT IS ABOVE.  TEST IF LEFT OR RIGHT    000245
00200        200   IF(ICOL-LOCCOL)300,240,320                       000247
00200        C --->       ABOVE AND SAME COLUMN                     000247
00203        240   INLIN=ILIN+3                                     000254
00204              IRLIN=LOCLIN-4                                   000256
00205              ITC=LOCCOL+3                                     000261
00206              ITL=1-IRLIN                                      000264
00207              GO TO 280                                        000266
00207        C --->       INPUT IS BELOW.  TEST IF LEFT OR RIGHT    000266
00210        220   IF(ICOL-LOCCOL)340,260,360                       000270
00210        C --->       BELOW AND SAME COLUMN                     000270
00213        260   INLIN=ILIN-4                                     000275
00214              IRLIN=LOCLIN+3                                   000277
00215              ITC=LOCCOL-8                                     000302
00216              ITL=IRLIN+1                                      000305
00216        C --->       ADD VERTICAL LINE                         000305
00217        280   IF(IDS.NE.0)GO TO 500                            000310
00221                 CALL VLINE(PAGE,ICOL,INLIN,IRLIN)             000311
00222              GO TO 500                                        000317
00222        C --->       INPUT IS IN UPPER LEFT QUAD.              000317
00223        300   IF(IDS.NE.0)GO TO 135                            000321
00225              LIN=ILIN+1                                       000322
00226              INCOL=ICOL+6                                     000325
00227              IRCOL=LOCCOL-1                                   000330
00230              ICO=IRCOL                                        000333
00231              INLIN=LIN                                        000334
00232              IRLIN=LOCLIN-4                                   000335
00233              ITC=LOCCOL-9                                     000340
00234              ITL=1-IRLIN                                      000343
00235              GO TO 400                                        000345
00235        C --->       INPUT IS IN UPPER RIGHT QUAD.             000345
00236        320   IF(IDS.NE.0)GO TO 240                            000347
00240              LIN=LOCLIN-1                                     000350
00241              INCOL=ICOL-1                                     000353
00242              IRCOL=LOCCOL+6                                   000356
00243              ICO=INCOL                                        000361
00244              INLIN=ILIN+3                                     000362
00245              IRLIN=LIN                                        000365
00246              ITC=LOCCOL+7                                     000366
00247              ITL=1-IRLIN                                      000371
00250              GO TO 400                                        000373
00250        C --->       INPUT IS IN LOWER LEFT QUAD.              000373
00251        340   IF(IDS.NE.0)GO TO 260                            000375
00253              LIN=LOCLIN+1                                     000376
00254              INCOL=ICOL+1                                     000401
00255              IRCOL=LOCCOL-5                                   000404
00256              ICO=INCOL                                        000407
00257              IRLIN=ILIN-4                                     000410
00260              INLIN=LIN                                        000413
00261              ITC=INCOL-6                                      000414
00262              ITL=IRLIN+1                                      000416
00263              GO TO 400                                        000420
00263        C --->       INPUT IS IN LOWER RIGHT QUAD.             000420
00264        360   IF(IDS.NE.0)GO TO 140                            000422
00266              LIN=ILIN-1                                       000423
```

```
00.      127*          INCOL=ICOL-5                                                    000426
00270    128*          IRCOL=LOCCOL+1                                                  000431
00271    129*          ICO=INCOL                                                       000434
00272    130*          INLIN=LIN                                                       000435
00273    131*          IRLIN=LOCLIN+3                                                  000436
00274    132*          ITC=IRCOL+2                                                     000441
00275    133*          ITL=IRLIN+1                                                     000443
00275    134*     C --->       ADD VERTICAL LINE SEGMENT                               000443
00276    135*     400  CALL VLINE(PAGE,ICO,INLIN,IRLIN)                                000446
00276    136*     C --->       ADD HORIZONTAL LINE SEGMENT                             000446
00277    137*          CALL HLINE(PAGE,LIN,INCOL,IRCOL)                                000453
00300    138*          GO TO 500                                                       000461
00300    139*     C --->       INPUT IS FROM ANOTHER PAGE                              000461
00300    140*     C ---        TEST TO PREVENT OFF PAGE SYMBOL FROM FALLING OFF PAGE   000461
00301    141*     420  IF(LOCLIN+7.GT.56.OR.LOCCOL-16.LT.1)GO TO 440                   000463
00301    142*     C --->       GENERATE EXTERNAL PAGE SYMBOL                           000463
00303    143*          CALL PUT1(PAGE(1,LOCLIN+3),LOCCOL-5,12H/                        000502
00304    144*          CALL PUT1(PAGE(1,LOCLIN+4),LOCCOL-7,1H/)                        000522
00305    145*          CALL STRMOV(ASTRSK,1,7,PAGE(1,LOCLIN+5),LOCCOL-15)              000533
00305    146*     C --->       PLACE EXTERNAL PAGE NO. IN EXTERNAL PAGE SYMBOL         000533
00306    147*          IPAGE=IPAGE/100                                                 000547
00307    148*          ENCODE(8,421,PAG)IPAGE                                          000553
00312    149*     421  FORMAT(5H*PAGE,I2,1H*)                                          000562
00313    150*          CALL STRMOV(PAG,1,8,PAGE(1,LOCLIN+6),LOCCOL-16)                 000562
00314    151*          CALL STRMOV(ASTRSK,1,6,PAGE(1,LOCLIN+7),LOCCOL-15)              000576
00315    152*     440  ITC=LOCCOL-16                                                   000610
00316    153*          ITL=LOCLIN+8                                                    000612
00316    154*     C --->       ADD TEXT TO INPUT LINE                                  000612
00317    155*     500  K=ISIGN(1,ITL)                                                  000616
00320    156*          ITL=IABS(ITL)                                                   000621
00321    157*          IF(NOIN.LT.1)GO TO 540                                          000623
00321    158*     C ---        PREVENT LABELS FROM FALLING OFF SIDES OF PAGE           000623
00323    159*          IF(ITC.LT.1)ITC=1                                               000627
00325    160*          IF(ITC.GT.123)ITC=123                                           000640
00325    161*     C ---        TEST FOR LABELS GOING OFF TOP OR BOTTOM OF PAGE         000640
00327    162*          IDS=ITL+K*(NOIN-1)                                              000646
00327    163*     C ---        REVERSE DIRECTION OF COLUMN TO PREVENT LOSS OF LABELS   000646
00330    164*          IF(IDS.LT.1.OR.IDS.GT.56)K=-K                                   000652
00330    165*     C --->       SCAN INPUTS FROM INPUT COMP.                            000652
00332    166*          DO 520 I=1,NOIN                                                 000677
00332    167*     C ---        TEST TO ASSURE THAT LABELS STAY ON PAGE                 000677
00335    168*          IF(ITL.LT.1.OR.ITL.GT.56)GO TO 540                              000677
00335    169*     C --->       ADD INPUT NAMES TO PAGE                                 000677
00337    170*          CALL STRMOV(INPUTS(I),1,7,PAGE(1,ITL),ITC)                      000715
00337    171*     C --->       INCREMENT PRINT LINE EITHER UP OR DOWN                  000715
00340    172*          ITL=ITL+K                                                       000733
00341    173*     520  CONTINUE                                                        000741
00343    174*     540  NOIN=0                                                          000741
00344    175*          RETURN                                                          000741
00345    176*          END @ CONNECT  *************************                        001012
```

```
MAIN PROGRAM    EASY

STORAGE USED  CODE(1) 001136; DATA(0) 002146; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003     C10      000003
 0004     COPOCR   000003
 0005     CTITLE   000016
 0006     CSEQ     000121
 0007     CTAB     000311
 0010     COCINP   000144
 0011     COCOUT   000144
 0012     COCCRI   000144
 0013     COC      000011


EXTERNAL REFERENCES (BLOCK, NAME)

 0014     READM$
 0015     NXTPH
 0016     LCMPH
 0017     STRMOV
 0020     NGMSTR
 0021     PUTCOD
 0022     NGMERC
 0023     NEWCOM
 0024     PCPLUB
 0025     INCOM
 0026     WRTIMS
 0027     ENDMOD
 0030     SCIBMA
 0031     LISTSC
 0032     NINTR$
 0033     NWDU$
 0034     NIO2$
 0035     NOFF$
 0036     NRDU$
 0037     NIO3$
 0040     NIOL$
 0041     NERR2$
 0042     NRCW$
 0043     NWEF$
 0044     NSTOP$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000066 120L       0001    000456 1000L      0000    001310 101F       0000    001275 11F        0001    000470 1100L
 0000    001352 1101F      0001    000510 1200L      0001    000535 1220L      0000    001366 1221F      0001    000547 1400L
 0001    000060 143G       0001    000127 164G       0001    000144 174G       0001    000573 2000L      0000    001311 201F
 0000    001317 205F       0000    001324 207F       0001    000237 216G       0001    000154 300L       0001    000614 3000L
 0001    000435 306G       0001    000161 320L       0001    000221 325L       0001    000260 328L       0001    000264 330L
 0000    001333 335F       0001    000301 400L       0001    000650 4000L      0000    001350 401F       0001    000355 410L
```

```
0001     000331 420L     0001     000361 500L     0001     000701 5000L    0001     000712 5100L    0001     000401 520L
0001     000724 5200L    0001     000736 5300L    0001     000750 5400L    0001     001007 5420L    0000     001413 5421F
0001     001021 5900L    0001     001023 6000L    0000     001435 6161F    0001     001044 6170L    0000     001436 6181F
0001     001067 6200L    0001     001100 6220L    0001     001123 6260L    0001     001125 6300L    0000 D 001237 PLNP
0000 D 000022 CHANDS   0000 D 000414 CMPMOD   0000 D 001440 CMPNTS   0000 D 001260 COMP     0000 D 001244 CAMIND
0000 D 001271 DCOMNM   0000 D 001250 DCPMAX   0000 D 000100 DINPUT   0000 D 000246 DOUT     0000 D 001242 FORLT
0000 I 001253 I        0000 I 001241 ICMMAX   0000 I 001440 ICMPNT   0000 D 000002 ICON     0000 I 001262 ICOMP
0000 I 001252 ICPMAX   0000 I 001273 IDCOMP   0003 I 000032 IDTAG    0000 I 001254 INDEX    0013 I 000006 IOCAN
0000 I 001236 IPUNCH   0003 I 000000 IREAD    0000 I 001235 ITASK    0013     000010 IUDC     0003 I 000001 IWRITE
0013 I 000007 IXOC     0000 I 001246 IIB      0000 I 001247 JJ       0000 I 001265 J        0000 I 000000 KONSTR
0000 D 000000 LOFNO    0013 I 000005 LOCOC    0000 I 001264 LTASK    0000 I 001266 MAINPT   0000 I 001267 NIOUT
0000 I 001263 NFLC     0013 I 000002 NOC      0013 I 000004 NOCCR    0013 I 000000 NOCIN    0013 I 000003 NOCPOD
0000 I 001234 NOCOMP   0013 I 000001 NOCOUT   0009 I 000002 NOP      0007 I 000000 NOTAR    0009 I 000001 NOV
0009 I 000002 NAX      0006 I 000000 NSEQ     0000 I 001257 NTASK    0012 D 000000 OCCRIT   0010 D 000000 OCINPT
0011 D 000000 OCOUTP   0000 D 001255 PHRS     0006 D 000001 SEQA     0000 D 000000 SOURCE   0000 D 001273 TABDIM
0007 D 000001 TAPNAM   0005 D 000000 TITLE
```

```
00100      1*     CEASY                                                                                    000000
00100      2*     C     PROGRAM CASY(INPUT=100,OUTPUT=100,TAPE5=INPUT,TAPE6=OUTPUT                        000000
00100      3*     C   1 ,TAPE7=100,TAPE8=100,TAPE9=100,TAPE10=100,TAPE11=100,TAPE12=100,                000000
00100      4*     C   2 TAPE4=100,TAPE78=100,PUNCH=100,TAPE3=PUNCH)                                       000000
00100      5*     C VERSION 2.1                          REVISED OCT 15 1976                               000000
00100      6*     C  PURPOSE    TO GENERATE FORTRAN SOURCE OF ECS MODEL IN THE                             000000
00100      7*     C                FORM REQUIRED BY THE NONSIM PROGRAM.                                    000000
00100      8*     C LIMITATIONS    ARRAY DIMENSIONS IMPOSE THE FOLLOWING LIMITS                            000000
00100      9*     C          LIMITED QUANTITY      CURRENT VALUE      ARRAYS IMPOSING THE LI              000000
00100     10*     C                                                                                        000000
00100     11*     C        STANDARD COMPONENTS          K = 150     NSI(I)      K=(I-3)/6                  000000
00100     12*     C              "          "           K = 150     CMPNTS(I)   K=I-1                      000000
00100     13*     C                                                                                        000000
00100     14*     C     STD. COMPONENTS PER MODEL       K =  100     IVRSC(I)  K=I+5/4   (SEE             000000
00100     15*     C              "          "           K = 100     CONARR(I)  K=(60+I)**.5 (S             000000
00100     16*     C              "          "           K = 100     ISEQ(I)    K=I    (SEE COMO            000000
00100     17*     C              "          "           K = 100     IV1(I)     K=I    (SEE COMO            000000
00100     18*     C              "          "           K = 100     IV2(I)     K=I    (SEE COMO            000000
00100     19*     C              "          "           K =  100     CMPMOD(I)  K=I                         000000
00100     20*     C              "          "           K = 100     INDEX(I)   K=I-1                       000000
00100     21*     C              "          "           K = 100     SEQA(I)    K=5*I                       000000
00100     22*     C                                                                                        000000
00100     23*     C      INPUTS FOR ANY STD. COMP.      K =  50     DINPUT(I)  K=I-1                       000000
00100     24*     C              "          "           K =  50     UINPUT(I)  K=I-1  (SEE INC             000000
00100     25*     C                                                                                        000000
00100     26*     C      OUTPUTS FOR ANY STD. COMP.     K =  50     OUTPUT(I)  K=I-1                       000000
00100     27*     C              "          "           K =  50     DOUT(I)    K=I-1                       000000
00100     28*     C                                                                                        000000
00100     29*     C       TABLES PER STD. COMP.         K =  9      TABLE(I)   K=I    (SEE COMO            000000
00100     30*     C                                                                                        000000
00100     31*     C       TABLES PER MODEL              K = 100     TAGNAM(I)  K=I    (SEE COMO            000000
00100     32*     C                                                                                        000000
00100     33*     C     OPTIMAL CONTROLLER INPUTS       K =  50     OCINPT(I)  K=I    (SEE COMO            000000
00100     34*     C                                                                                        000000
00100     35*     C     OPTIMAL CONTROLLER OUTPUTS      K =  50     OCOUTP(I)  K=I                         000000
00100     36*     C                                                                                        000000
00100     37*     C     OPTIMAL CONTROLLER CRITERIA     K =  50     OCCRIT(I)  K=I                         000000
```

```
00100   C                                                                        000000
00100   C    DESIGNED BY  J.D.BURROUGHS                          DATE  MAY 1974  000000
00101                          IMPLICIT DOUBLE PRECISION (A-Z)                    000000
00103                          IMPLICIT INTEGER (I,J,K,L,M,N)                     000001
00104                          DOUBLE PRECISION LOCNO,ICOM                        000001
00105         COMMON/CIO/IREAD,IWRITE,IDIAG/CORDER/NOX,NOV,NOP                   000001
00105        1/CTITLE/TITLE(7)/CSEQ/NSEQ,SEQ(40)/CTAB/NOTAB,TABNAM(100)          000001
00106         COMMON/COCINP/OCINPT(50)/COCOUT/OCOUTP(50)/COCCRI/OCCRIT(50)       000001
00107         COMMON/COC/NOCIN,NOCOUT,NOC,NOCMOD,NOCCR,LOCOC,IOCAN,IYOC,IUOC     000001
00110         DIMENSION ICOM(8),CMMNDS(15),SOURCE(6),DIGPUT(5),DOUT(5)           000001
00111         DIMENSION CMPNTS(151),CMMOD(200),ICMPNT(2,151)                     000001
00112         EQUIVALENCE(CMPNTS,ICMPNT)                                         000001
00113         DATA NOCOMP/0/,ITASK/6/,IPUNCH/0/                                  000001
00113   C           INACTIVATE O.C. PROCESSING                                   000001
00113   C     REDUCE NO. OF COMMANDS FROM 21 TO 15                               000001
00117         DATA BLANK/12H            /,ICMMAX/15/                             000001
00122         DATA CMMNDS/12HLOCATION    ,12HINPUTS      ,12HFORTRAN ST          000001
00122        112HEND OF MOD  ,12HXXXXXXXXXXX  ,12HMODEL DESC  ,                  000001
00122        212HPRINT       ,12HXXXXXXXXXXX  ,12HPUNCH       ,                  000001
00122        312HDIAGNOSTIC  ,12HADD STATES   ,12HADD VARIAB  ,                  000001
00122        412HADD PARAMT  ,12HADD TABLES   ,12HLIST STAND  /                  000001
00122   C           INACTIVATE O.C. COMMANDS (16 - 21)                           000001
00122   C    512HO.C. INPUT  ,12HO.C. OUTPU  ,12HO.C. ORDER  ,                   000001
00122   C    612HO.C. MODEL  ,12HO.C. CRITE  ,12HO.C. ANALY  /                   000001
00122   C     LITERAL 'POOL' TO SATISFY DBLE PRECSN ASSGNMT STMNTS               000001
00124         DATA FORLT/'FORT        '/,CMMUND/'-100        '/                  000001
00127         IDIAG=0                                                            000001
00130         IREAD=5                                                            000002
00131         IWRITE=6                                                           000004
00132         WRITE(IWRITE,11)                                                   000006
00134   11    FORMAT(1H1,10X,'INPUT COMMANDS'/)                                  000013
00134   C --->        OPEN STANDARD COMPONENT FILE                               000013
00135         DEFINE FILE 18(2810,302,U,I18),7(20),128,U,I7)                     000013
00135   C --->        OBTAIN STD. COMPONENT NAMES FROM PERMANENT FILE            000013
00137         CALL READMS(18,DCPMAX,1,12HCMPNTS                                  000031
00140         ICPMAX = DCPMAX                                                    000037
00141         CALL READMS(18,CMPNTS,ICPMAX,12HCMPNTS                             000045
00142         DO 20 I=2,ICPMAX                                                   000053
00145   20    CMPNTS(I-1)=CMPNTS(I)                                              000060
00147         ICPMAX=ICPMAX-1                                                    000062
00147   C --->        READ DATA CARD                                             000062
00150   100   CONTINUE                                                           000066
00151         READ(IREAD, 101, END = 6260, ERR = 6260)ICOM                       000066
00154   101   FORMAT(8A10)                                                       000077
00155   200   WRITE(IWRITE,201)ICOM                                              000077
00160   201   FORMAT(/' COMMAND CARD ---> ',8A10)                                000107
00160   C --->        DIAGNOSTIC PRINTS                                          000107
00161         IF(IDIAG.EQ.1)WRITE(IWRITE,205)(CMMNDS(I),I=1,ICMMAX)              000127
00170   205   FORMAT(' COMMANDS'/10(1X,A10))                                     000132
00171         IF(IDIAG.EQ.2)WRITE(IWRITE,207)(ICMPNT(1,I),                       000132
00171        1 ICMPNT(1,I),ICMPNT(2,I),I=1,ICPMAX)                               000132
00202   207   FORMAT(' STD. COMPONENTS'/ (1X,A6,2X,20I2/))                       000151
00202   C --->        INDEX FOR DATA CARD COLUMN                                 000151
00203         INDEX=1                                                            000151
00203   C --->        LOCATE NEXT PHRASE                                         000151
00204   300   CALL NXTPHR(ICOM,INDEX,PHRS)                                       000154
00205   320   IF(PHRS.EQ.BLANK) GO TO 100                                        000161
00205     95*     C --->        SEARCH COMMAND LIST                                                  000161
00207     96*           CALL LCMPH(PHRS,CMMNDS,ICMMAX,1,NTASK)                                       000163
00207     97*     C --->        NTASK = NEW TASK INDICATOR                                           000163
00210     98*           IF(NTASK.NE.0) GO TO 400                                                     000172
00210     99*     C --->        TEST FOR DIRECT MODEL MODES AND O.C. INPUTS                          000172
00212    100*           GO TO(325,325,5000,325,325,325,325,325,325,325,                              000174
00212    101*          1 5100,5200,5300,5400,325),ITASK                                              000174
00212    102*     C             INACTIVATE O.C. PROCESSING                                           000174
00212    103*     C    1 5100,5200,5300,5400,325,7000,7000,7000,7000,7000,                           000174
00212    104*     C    2 7000),ITASK                                                                 000174
00212    105*     C --->        SEPARATE STANDARD COMPONENT NAME FROM SPECIFIC COMPONENT NAM         000174
00213    106*     325   COMP=BLK                                                                     000221
00214    107*           CALL STPMOV(PHRS,1,2,COMP,1)                                                 000222
00214    108*     C --->        SEARCH COMPONENT NAME LIST                                           000222
00215    109*           DO 326 ICOMP=1,ICPMAX                                                        000237
00220    110*           IF(KOMSTR(CMPNMS(ICOMP),1,2,COMP,1).EQ.0)GO TO 328                           000237
00222    111*     326   CONTINUE                                                                     000255
00224    112*           ICOMP=0                                                                      000255
00225    113*           GO TO 330                                                                    000256
00226    114*     328   IF(ITASK.EQ.1) GO TO 1200                                                    000260
00230    115*           GO TO 2000                                                                   000262
00231    116*     330   WRITE(IWRITE,335)COMP                                                        000264
00234    117*     335   FORMAT(/5X,34H *** WARNING ***  CAN"T IDENTIFY  ,A10,'AS A STANDAR           000271
00234    118*          1D COMPONENT.')                                                               000271
00235    119*           IF(ITASK.EQ.2)GO TO 300                                                      000271
00237    120*           ITASK=6                                                                      000274
00240    121*           NEWC=0                                                                       000276
00241    122*           GO TO 300                                                                    000277
00241    123*     C --->        NEW COMMAND IDENTIFIED                                               000277
00242    124*     400   LTASK=ITASK                                                                  000301
00243    125*           ITASK=NTASK                                                                  000302
00244    126*           IF(LTASK.EQ.3)WRITE(9,401)                                                   000304
00247    127*     401   FORMAT('FORT')                                                               000314
00247    128*     C --->        TESTS FOR UNFINISHED BUSINESS                                        000314
00250    129*           IF(LTASK.EQ.1.OR.LTASK.EQ.2) GO TO 410                                       000314
00250    130*     C --->        BRANCH TO NEW TASK                                                   000314
00252    131*     420   GO TO(1200,2000,500,4000,4000,520,6000,100,5900,1400,                        000331
00252    132*          1 300,300,300,300,6300),ITASK                                                 000331
00252    133*     C             INACTIVATE O.C.PROCESSING                                            000331
00252    134*     C    1 300,300,300,300,6300,300,300,300,300,300,                                   000331
00252    135*     C    2 7100),ITASK                                                                 000331
00253    136*     410   IF(LTASK.EQ.2) GO TO 300                                                     000355
00255    137*           GO TO 3000                                                                   000357
00255    138*     C ====================  FORTRAN STATEMENTS   ITASK = 3                             000357
00256    139*     500   NOCOMP=NOCOMP+1                                                              000361
00256    140*     C ---         ADD COMP. NO. TO COMPONENT SEQUENCE LIST                             000361
00257    141*           NSEQ=NSEQ+1                                                                  000363
00260    142*           CALL PUTCOD(NSEQ,SEQA,NOCOMP)                                                000366
00261    143*           CMPMOD(NOCOMP)=FORLT                                                         000373
00262    144*           GO TO 100                                                                    000377
00262    145*     C ====================  MODEL DESCRIPTION    ITASK = 6                             000377
00263    146*     520   NEWC=0                                                                       000401
00264    147*           NCV=0                                                                        000401
00265    148*           NAX=0                                                                        000402
00266    149*           NCP=0                                                                        000403
00267    150*           NCOMP=0                                                                      000404
00270    151*           NSEQ=0                                                                       000405
```

```
00271               NOTAB=0                                                    000406
00272               NOCIN=0                                                    000407
00273               NOCOUT=0                                                   000410
00274               NOC=-1                                                     000411
00275               NOCMOD=-1                                                  000413
00276               NOCCR=0                                                    000414
00277               LOCOC=-1                                                   000415
00300               LOCAN=0                                                    000416
00301               IXOC=1                                                     000417
00302               REWIND 8                                                   000421
00303               REWIND 10                                                  000424
00304               REWIND 11                                                  000427
00304         C --->        LOAD TITLE                                         000427
00305               DO 530 I=1,7                                               000435
00310         530   TITLE(I)=BLNK                                              000435
00312               I=INDEX+1                                                  000437
00313               J=80-INDEX                                                 000442
00314               CALL STRMOV(ICOM,I,J,TITLE,1)                              000445
00315               GO TO 100                                                  000454
00315         C --->        INITIATE NEW COMPONENT                             000454
00315         C --->        GET COMPONENT LOCATION NUMBER                      000454
00315         C =====================  LOCATION      ITASK = 1                 000454
00316         1000  CALL NXTPH(ICOM,INDEX,LOCNO)                               000456
00317               CALL NUMERC(LOCNO,$1100)                                   000462
00320               GO TO 330                                                  000466
00321         1100  WRITE(IWRITE,1101)LOCNO                                    000470
00324         1101  FORMAT(/5X,18H *** WARNING ***  ,A10,'  IS NOT A VALID LOCATION NU   000475
00324              1MBER')                                                     000475
00325               CALL STRMOV(LOCNO,1,10,PHRS,1)                             000475
00326               LOCNO=CRNUMO                                               000504
00327               GO TO 320                                                  000506
00330         1200  IF(NEWC.EQ.1)GO TO 1220                                    000510
00332               CALL NEWCOM(PHRS,CMPNTS,ICOMP,LOCNO,CMPMOD,NOCOMP          000512
00332              1DINPUT,NOINPT,DOUT,NODOUT,IDCOMP)                          000512
00333               DCOMNM=PHRS                                                000527
00334               NEWC=1                                                     000531
00335               GO TO 320                                                  000533
00336         1220  WRITE(IWRITE,1221)DCOMNM,PHRS                              000535
00342         1221  FORMAT(/5X,28H *** WARNING *** COMPONENT  ,A10,'  DEFINITION WASN'   000543
00342              1T COMPLETED BEFORE STARTING THE DEFINITION OF COMPONENT  ',A10)      000543
00343               ITASK=6                                                    000543
00344               GO TO 3000                                                 000545
00344         C =====================  DIAGNOSTIC CONTROL    ITASK = 10        000545
00345         1400  CALL NXTPH(ICOM,INDEX,PHRS)                                000547
00345         C ---         CHECK FOR NUMERIC IPUT, SKIP INPUT IF NOT NUMERIC  000547
00346               CALL NUMERC(PHRS,$300)                                     000553
00346         C ---         CONVERT TO INTEGER                                 000553
00347               CALL ACDBUB(PHRS,PHRS)                                     000557
00350               IDIAG=PHRS                                                 000563
00351               GO TO 300                                                  000571
00351         C =====================  INPUTS     ITASK = 2                    000571
00351         C ---         TEST TO ASSURE THAT COMP. HAS BEEN IDENTIFIED.     000573
00352         2000  IF(LTASK.EQ.6)GO TO 300                                    000573
00352         C --->        ADD INPUTS TO COMPONENT                            000573
00354               CALL INCOM(ICOM,PHRS,INDEX,NOINPT,DINPUT,NDOUT,DOUT,       000575
00354              1 DCOMNM,CMPMOD,NOCOMP,ICOMP)                               000575
00355               GO TO 320                                                  000612
```

```
00356    C --->      STORE INPUT LIST FOR COMPONENT                                         000612
00356    3000  IF(IDCOMP.GE.1.AND.IDCOMP.LE.NOCOMP.AND.NDINPT.GT.0)                         000614
00356         1 CALL WRIMS(7,DINPUT,NDINPT,IDCOMP)                                          000614
00360          NEWC=0                                                                       000645
00361          GO TO 920                                                                    000646
00361    C ====================  END OF MODEL   COMPILE   ITASK = 4,5                        000646
00361    C --->      FORM MODEL SUBROUTINES                                                 000646
00362    4000  CALL FNDMOD(CMPMOD,NOCOMP,DOUT)                                              000650
00363          GO TO(300,300,300,300,6200,300,6000,100,5900,1400,                           000654
00363         1 300,300,300,300,300),ITASK                                                  000654
00363    C --->      WRITE FORTRAN ORID SOURCE FILE                                         000654
00364    5000  WRITE(9,101)ICOM                                                             000701
00367          GO TO 100                                                                    000710
00367    C ====================  ADD STATES     ITASK = 11                                  000710
00367    C --->      ADD STATES TO MODEL                                                    000710
00373    5100  WRITE(8,101)PHRS                                                             000712
00373          NOX=NOX+1                                                                    000717
00374          GO TO 300                                                                    000722
00374    C ====================  ADD VARIABLES   ITASK = 12                                 000722
00374    C --->      ADD VARIABLES TO MODEL                                                 000722
00375    5200  WRITE(11,101)PHRS                                                            000724
00400          NOV=NOV+1                                                                    000731
00401          GO TO 300                                                                    000734
00401    C ====================  ADD PARAMETERS   ITASK = 13                                000734
00401    C --->      ADD PARAMETERS TO MODEL                                                000734
00402    5300  WRITE(10,101)PHRS                                                            000736
00405          NOP=NOP+1                                                                    000743
00406          GO TO 300                                                                    000746
00406    C ====================  ADD TABLES    ITASK = 14                                   000746
00406    C --->      ADD TABLES TO MODEL                                                    000746
00406    C --->      GET TABLE DIMENSION IN NEXT PHRASE                                     000746
00407    5400  CALL NXTPH(ICOM,INDEX,TABDIM)                                                000750
00407    C --->      TEST TO ASSURE THAT TABLE DIMENSION IS NUMERIC                         000750
00410          CALL NUMERC(TABDIM,$5420)                                                    000754
00410    C --->      CONVERT TABLE DIMENSION TO INTEGER                                     000754
00411          CALL RCDOUB(TABDIM,TADIM)                                                    000760
00412          I=TABDIM                                                                     000764
00413          CALL PUTCOD(5,PHRS,1)                                                        000772
00414          NOTAB=NOTAB+1                                                                000777
00415          TABNAM(NOTAB)=PHRS                                                           001002
00416          GO TO 300                                                                    001005
00417    5420  WRITE(IWRITE,5421)PHRS,TABDIM                                                001007
00423    5421  FORMAT(/5X,29H *** WARNING *** TABLE NAME  ,A7,                              001015
00423         1' MUST BE FOLLOWED BY A NUMERIC DIMENSION RATHER THAN  ',A7)                 001015
00424          PHRS=TABDIM                                                                  001015
00425          GO TO 320                                                                    001017
00425    C --->      SET INDICATOR TO PUNCH SOURCE DECKS                                    001017
00425    C ====================  PUNCH   ITASK =9                                           001017
00426    5900  IPUNCH=1                                                                     001021
00426    C ====================  PRINT   ITASK = 7                                          001021
00426    C --->      DRAW SCHEMATIC DIAGRAM                                                 001021
00427    6000  CALL SCHEMA(CMPMOD,NOCOMP,DINPUT,DOUT)                                       001023
00427    C --->      PRINT INPUT REQUIREMENTS LIST                                          001023
00430          END FILE 12                                                                  001030
00431          REWIND 12                                                                    001033
00432          WRITE(IWRITE,6161)                                                           001036
00434    6161  FORMAT(1H1)                                                                  001044
```

```
00435      6170  CONTINUE                                                    001044
00436            READ(12,101,END=6200,ERR=6260)SOURCE                        001044
00441      6180  WRITE(IWRITE,6181)SOURCE                                    001055
00444      6181  FORMAT(1X,7A10,A2)                                          001065
00445            GO TO 6170                                                  001065
00445      C --->       PUNCH SOURCE FILE                                    001065
00446      6200  IF(IPUNCH.NE.1)GO TO 100                                    001067
00450            END FILE 9                                                  001071
00451            REWIND 9                                                    001074
00452      6220  CONTINUE                                                    001100
00453            READ(9,101,END=100,ERR=6260)SOURCE                          001100
00456      6250  WRITE(3,101)SOURCE                                          001111
00461            GO TO 6220                                                  001121
00462      6260  CONTINUE                                                    001123
00463            STOP                                                        001123
00463      C ===================  LIST STANDARD COMPONENTS   ITASK = 15      001123
00464      6300  CALL LISTSC(ICPMAX,CMPNTS,DINPUT,DOUT)                      001125
00465            GO TO 300                                                   001132
00465      C           INACTIVATE O.C. PROCESSING                            001132
00465      C ===================  O.C. COMMANDS  ITASK = 16,17,18,19,20,22  001132
00465      C --->       INTERPRETE OPTIMAL CONTROLLER INPUTS                 001132
00465      C7000  CALL OCINTR(ITASK,PHRS)                                    001132
00465      C     GO TO 300                                                   001132
00465      C ===================  O.C. ANALYSIS ONLY   ITASK = 21            001132
00465      C --->       SET ANALYSIS ONLY FLAG                               001132
00465      C7100  IOCAN=1                                                    001132
00465      C     GO TO 300                                                   001132
00466            END @ EASY  ****************************                    001135
```

```
SUBROUTINE ENDCOM    ENTRY POINT 000166

STORAGE USED  CODE(1) 000215; DATA(0) 000050; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CIO    000003
 0004   CORDER 000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0005   GETCOD
 0006   READMS
 0007   GETT
 0010   MAPGEN
 0011   LINF
 0012   NGPUS
 0013   NLO2$
 0014   NIO1$
 0015   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000057 100L       0000     000014 101F       0000     000013 11F        0001     000102 110L       0001     000023 122G
 0001     000125 145G       0001     000140 154G       0000     000021 201F       0000     000022 205F       0000 D 000004 BLNK
 0000 D 000010 CHAR       0000 D 000000 COMMA      0000 I 000007 I          0003 I 000002 IDIAG      0000     000037 INJP
 0003     000000 IREAD      0003 I 000001 IWRITE     0000 I 000006 NINPUT     0000 I 000012 NO         0004 I 000002 NOP
 0004     000001 NOV        0004     000000 NOX        0000 D 000002 RPAR


00100          CENDCOM                                                                               000000
00101                SUBROUTINE ENDCOM(AINPUT,COMNAM,SOURCE,ISOUR,NOCOMP,NSEQ)                       000000
00101          C  VERSION 2.                             REVISED  DEC 15 1975                        000000
00101          C  PURPOSE   TO COMPLETE A COMPONENT DESCRIPTION IN THE ECS MODEL.                    000000
00101          C  CALL SEQUENCE  AINPUT - LIST OF INPUT QUANTITY NAMES                               000000
00101          C                   COMNAM - SPECIFIC COMPONENT NAME                                  000000
00101          C                   SOURCE - BUFFER ARRAY OF SOURCE CODE                              000000
00101          C                   ISOUR  - INDEX TO NEXT CHARACTER IN SOURCE BUFFER                 000000
00101          C                   NOCOMP - MODEL COMPONENT NO.                                      000000
00101          C                   NSEQ   - MODEL COMPONENT SEQUENCE NO.                             000000
00103                               IMPLICIT DOUBLE PRECISION (A-Z)                                  000000
00104                               IMPLICIT INTEGER (I,J,K,L,M,N)                                   000000
00105                DIMENSION AINPUT(1),SOURCE(8)                                                   000000
00106                COMMON/CIO/IREAD,IWRITE,IDIAG                                                   000000
00107                COMMON /CORDER/NOX,NOV,NOP                                                      000000
00110                DATA COMMA/12H,           /,RPAR/12H)                                           000000
00113                DATA BLNK/12H                                                                   000000
00115                CALL GETCOD(5,COMNAM,NINPUT)                                                    000000
```

```
00115     19*    C ---       TEST FOR COMPONENTS WITH NO INPUTS                                          000000
00116     20*          IF(NINPUT.LE.0)GO TO 110                                                      000004
00120     21*          CALL READMS(7,AINPUT,NINPUT,NOCOMP)                                           000007
00120     22*    C --->      SCAN INPUTS                                                             000007
00121     23*          DO 200 I=1,NINPUT                                                             000015
00121     24*    C --->      TEST 4TH CHARACTER TO DETERMINE IF INPUT SOURCE HAS BEEN SAT                000015
00124     25*          CALL GETT(AINPUT(I),4,CHAR)                                                   000023
00125     26*          IF(CHAR.NE.BLANK) GO TO 100                                                   000031
00125     27*    C --->      NOT STAISFIED - TYPE INPUT AS A PARAMETER                               000031
00125     28*    C --->      FORM UNIQUE NAME BY ADDING COMPONENT NAME                               000031
 0127     29*          CALL NAMGEN(AINPUT(I),COMNAM,AINPUT(I))                                       000034
00127     30*    C --->      INCREASE PARAMETER COUNTER                                              000034
00133     31*          NOP=NOP+1                                                                     000045
00133     32*    C --->      ADD NAME TO PARAMETER NAME LIST                                         000045
00131     33*          WRITE(15,11)AINPUT(I)                                                         000050
00134     34*    11    FORMAT(A10)                                                                   000057
00134     35*    C --->      ADD INPUT TO COMPONENT CALL SEQUENCE                                    000057
00135     36*    100   CALL LINE(0,SOURCE,ISOUR,COMMA,1,12)                                          000057
00136     37*          CALL LINE(0,SOURCE,ISOUR,AINPUT(I),6,12)                                      000066
00137     38*    200   CONTINUE                                                                      000102
00137     39*    C --->      COMPLETE CALL SEQUENCE WITH )                                           000102
00141     40*    110   CALL LINE(0,SOURCE,ISOUR,RPAR,1,12)                                           000102
00142     41*          IF(IDUMP.GE.50)WRITE(IWRITE,101)SOURCE                                        000111
00151     42*    101   FORMAT(' ENDCOM-SOURCE'/(1X,6A10))                                            000130
00151     43*    C --->      WRITE LINE ON SOURCE FILE                                               000130
00152     44*          WRITE(12,201)SOURCE                                                           000130
00163     45*    201   FORMAT(8A10)                                                                  000143
00163     46*    C --->      GENERATE STATEMENT NUMBER                                               000143
00161     47*          NO=NSEQ+9000                                                                  000143
00161     48*    C --->      WRITE CONTINUE STATEMENT ON SOURCE FILE                                 000143
00162     49*          WRITE(12,205)NO                                                               000146
00165     50*    205   FORMAT(1X,I4,1X,'CONTINUE')                                                   000154
00166     51*          RETURN                                                                        000154
00167     52*          END @ ENDCOM  ******************************                                  000214
```

```
SUBROUTINE ENDMOD    ENTRY POINT 001306

STORAGE USED  CODE(1) 001326; DATA(0) 001722; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CORDER 000003
 0004    CTITLE 000016
 0005    CSEQ   000003
 0006    CTAB   000003
 0007    CCC    000010
 0010    CIO    000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0011    READKS
 0012    GETCOD
 0013    CALLCP
 0014    ENDCOM
 0015    MONSTR
 0016    COMGEN
 0017    TAPGEN
 0020    FACGO$
 0021    LINE
 0022    TAPCAL
 0023    COMFOU
 0024    NAMARY
 0025    TABDAT
 0026    STEPOV
 0027    NREW$
 0030    NWDU$
 0031    NIO1$
 0032    NIO2$
 0033    NRDU$
 0034    NIO3$
 0035    NWEF$
 0036    NERR3$

 STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000250 105L     0000    000760 111F     0001    000267 120L     0000    000767 121F     0001    000306 140L
 0001    000037 142G     0001    000045 150G     0000    000774 151F     0000    000612 23F      0001    000400 302G
 0000    000617 31F      0001    000410 311G     0001    000464 331G     0001    000323 350L     0001    000526 351G
 0001    000346 400L     0000    001003 401F     0000    001033 411F     0001    000652 417G     0001    000372 420L
 0000    001042 501F     0000    001043 511F     0001    000534 515L     0001    001025 517G     0001    001233 573G
 0001    000123 60L      0001    000560 600L     0000    001054 601F     0000    000656 61F      0001    000605 620L
 0001    000631 700L     0000    001063 701F     0000    001135 711F     0000    001220 719F     0001    000157 74L
 0001    000677 740L     0000    001227 741F     0001    000712 780L     0000    001247 781F     0001    000167 80L
 0001    000725 800L     0000    000657 81F      0000    001267 821F     0000    001316 831F     0000    001330 833F
 0000    001351 841F     0001    000762 850L     0000    001355 851F     0001    000772 860L     0000    001404 861F
```

```
0001    001040 864L       0000    001495 865F       0000    001534 867F       0000    001575 869F       0000    001605 881F
0001    001173 900L       0000    001625 901F       0000    000660 91F        0000    001626 911F       0001    001232 920L
0000    001627 921F       0000    000752 93F        0001    001242 960L       0000    001632 961F       0001    001270 999L
0000 D 000603 ANAME       0000 D 000530 BLNK        0000 D 000536 COMMA       0000 D 000605 COMP        0000 D 000601 COMPS
0000 D 000546 CYCLES      0000 D 000550 DLINES      0000 D 000524 ECS         0000 D 000526 FORLT       0000 D 000520 GT
0000 D 000534 HCP         0000 D 000532 HCX         0000 D 000542 HP          0000 D 000540 HT          0000 I 000556 I
0000 I 000557 ICOMP       0010 I 000002 IDIAG       0000    001677 INJP$      0007    000006 IOCAN      0010    000000 IREAD
0000 I 000565 ISN         0000 I 000560 ISOUR       0000 I 000562 IVR         0000 D 000600 IVRSET      0010 I 000001 IWRITE
0007    000007 IXOC       0000 I 000563 IXSOUR      0000 I 000566 J           0000 I 000571 K           0015 I 000000 KOMSTR
0007    000005 LOCCOC     0000 I 000572 MAXI        0000 I 000576 MAXTP       0000 I 000573 N           0000 I 000564 NCT
0007    000002 NOC        0007    000004 NOCCR      0007    000000 NOCIN      0007    000003 NOCMOD     0007    000001 NOCOUT
0003 I 000002 NOP         0000 I 000575 NOPP        0006 I 000000 NOTAB       0000 I 000577 NOTPBP      0003 I 000001 NOV
0000 I 000574 NOVP        0003 I 000000 NOX         0000 I 000561 NOXP        0005 I 000000 NSEQ        0000 I 000603 KUNIT
0000 I 000567 NP          0000 I 000570 N1          0000 D 000554 PFNAME      0000 D 000552 RESET       0000 D 000544 PPAR
0005 R 000001 SEQA        0006 D 000001 TABNAM      0004 D 000000 TITLE       0000 D 000500 XSOUR
```

```
00100         CENDMOD                                                                              000005
00101               SUBROUTINE ENDMOD(CMPMOD,NOCOMP,OUTPUT)                                       000005
00101         C  VERSION 3.                              REVISED  JUNE 7 1976                     000005
00101         C  PURPOSE   COMPLETE THE GENERATION OF ECS MODEL SUBROUTINES EDMO   DAT            000005
00101         C  CALL SEQUENCE   CMPMOD - ARRAY CONTAINING NAMES OF MODEL COMPS.                  000005
00101         C                  NOCOMP - COMPONENT COUNTER                                       000005
00101         C                  OUTPUT - INPUT-OUTPUT-TABLE NAME ARRAY WORK SPACE                000005
00101         C   DESIGNED BY  J.D.BURROUGHS                          DATE  JULY 1974             000005
00103                           IMPLICIT DOUBLE PRECISION (A-Z)                                   000005
00104                           IMPLICIT INTEGER (I,J,K,L,M,N)                                    000005
00105                           DOUBLE PRECISION IVRSET                                           000005
00106               COMMON/COPDFA/NOX,NOV,NOP/CTITLE/TITLE(7)/CSEQ/NSEQ,SEQA(1)                   000005
00106              1 /CTAB/NOTAB,TABNAM(1)/COC/NOCIN,NOCOUT,NOC,NOCMOD,NOCCR,LOCCOC,              000005
00106              2 IOCAN,IXOC                                                                   000005
00107               COMMON/CIO/IREAD,IWRITE,IDIAG                                                 000005
00110               DIMENSION IVRSET(163),XSOUR(8),GT(2),CMPMOD(1),OUTPUT(1)                      000005
00111               DATA GT/20H      GO T  0(          /,ECS/12HECS          /                    000005
00113         C     LITERAL 'FOOL' TO SATISFY DBLE PRECSN ASSGNMNT STMNTS                         000005
00114               DATA FORLT/'FORT         '/,BLNK/'               '/                           000005
00117               DATA HCX/12HCX           /,HCP/12HCP             /                            000005
00122               DATA COMMA/12H,           /,HT/12HT            /                              000005
00125               DATA HP/12HP            /,PPAR/12H),INDP       /                              000005
00130               DATA CYCLES/'CYCLES       '/,DLINES/'DLINES       '/,                          000005
00130              1 RESET/'RESET                                                                 000005
00134               REWIND 12                                                                     000005
00135               REWIND 9                                                                      000010
00135         C ---          GET PERMANENT FILE NAME                                              000010
00136               CALL READPS(10,PFNAME,1,12HPFNAME)                                            000013
00137               IF(IDIAG.EQ.2)WRITE(IWRITE,21)(CMPMOD(I),I=1,NOCOMP)                          000021
00146         21    FORMAT('  CMPMOD '/10(1X,A10))                                                000045
00146         C            INACTIVATE O.C. PROCESSING                                             000045
00146         C --->        COMPLETE OPTIMAL CONTROLLER SPECIFICATION                             000045
00146         C     CALL OCCMD(NOCOMP,CMPMOD,OUTPUT)                                              000045
00146         C     IF(NOCOMP.LE.0)GO TO 95                                                       000045
00146         C --->        CHECK COMPONENT SEQUENCE FOR IMPLICIT EQUATIONS                       000045
00146         C       CALL COPORD(CMPMOD,NOCOMP,OUTPUT)                                           000045
00146         C --->        SCAN MODEL COMPONENTS IN SEQUENCE OF LOCATION STATEMENTS              000045
00147               DO 80 I=1,NSEQ                                                                000045
```

```
00147     39*     C --->       GET COMPONENT NO. IN LOCATION SEQUENCE                                        000045
00152     40*           CALL GETCOO(I,SEQA,ICOMP)                                                            000045
00152     41*     C --->       TEST FOR DIRECT FORTRAN COMPONENTS                                            000045
00153     42*           IF(CMPMOD(ICOMP).EQ.FORLT)GO TO 60                                                   000055
00155     43*           IF(I.EQ.1)WRITE(12,31)                                                               000061
00160     44*        31 FORMAT(6X,'IF(CPUS.EQ.CPUSEC) GO TO 1'                                               000071
00160     45*          1 /6X,'IF(CYCLES.LE.0.) DLINES=0.'                                                    000071
00160     46*          2 /6X,'ITEST=0'/6X,'IF(NRSET.GT.0.) ITEST=1'                                          000071
00160     47*          3 /6X,'CPUS=CPUSEC'/6X,'ICNT=0'/6X,'IMPL=0'                                           000071
00160     48*          4 /'    1  CONTINUE')                                                                 000071
00160     49*     C            INACTIVATE O.C. PROCESSING                                                    000071
00160     50*     C --->       TEST FOR O.C. IF YES CALL OCCALL                                              000071
00160     51*     C     IF(KOMSTR(CMPMOD(ICOMP),1,2,HOC,1).EQ.0)GO TO 72                                     000071
00160     52*     C --->       INITIATE COMPONENT SUBROUTINE CALL GENERATION                                 000071
00161     53*           CALL CALLCP(CMPMOD(ICOMP),ICOMP,XSOUR,ISOUR,IVRSET,OUTPUT)                           000071
00161     54*     C --->       COMPLETE COMPONENT SUBROUTINE CALL GENERATION                                 000071
00162     55*           CALL ENDCOM(OUTPUT,CMPMOD(ICOMP),XSOUR,ISOUR,ICOMP,I)                                000104
00163     56*           GO TO 80                                                                             000121
00163     57*     C --->       TRANSFER DIRECT FORTRAN FROM FILE 9 TO FILE 12                                000121
00164     58*     60    CONTINUE                                                                             000123
00165     59*           READ(9,61,END=80,ERR=999)XSOUR                                                       000123
00170     60*     61    FORMAT(A10)                                                                          000134
00171     61*     70    IF(KOMSTR(XSOUR,1,4,FORLT,1).EQ.0)GO TO 74                                           000134
00173     62*           WRITE(12,61)XSOUR                                                                    000145
00176     63*           GO TO 60                                                                             000155
00176     64*     C          INACTIVATE O.C. PROCESSING                                                      000155
00176     65*     C72    CALL OCCALL(CMPMOD,NOCOMP,I,IVRSET,OUTPUT)                                          000155
00177     66*        74 IF(I.EQ.1)WRITE(12,31)                                                               000157
00202     67*     80    CONTINUE                                                                             000170
00204     68*     90    REWIND 9                                                                             000170
00204     69*     C-----             ADD PARAMETERS CYCLES,DLINES,RESET                                      000170
00204     70*     C                                                                                          000170
00205     71*           WRITE(10,81) CYCLES,DLINES,RESET                                                     000173
00212     72*        81 FORMAT(A10)                                                                          000203
00213     73*           NOP=NOP+3                                                                            000203
00213     74*     C =========>               FORM SUBROUTINE COMO                                            000203
00214     75*           NOXP=MAX0(NOX,1)                                                                     000206
00215     76*           WRITE(9,91)TITLE,PFNAME,NOXP,NOXP                                                    000214
00223     77*     91    FORMAT('@FOR,IS ASSI.COMO,ASRO.COMO'/                                                000230
00223     78*          16X,'SUBROUTINE COMO(TIME,TMAX,INOP)'/'C'/'C',9X,7A10/'C'/                            000230
00223     79*          2'C --->   THIS SUBROUTINE WAS PREPARED BY THE SIMWEST PRECOMPILER                    000230
00223     80*          3/'C',25X,'USING  ',A10,' COMPONENTS'                                                 000230
00223     81*          4/6X,'COMMON/CXDOT/XDOT(',I4,')/CINT/INT(',I4,')'                                     000230
00223     82*          5 /6X,'COMMON/CIMPL/IMPL,ICNT,ITEST/COVRLY/DUM(3),CPUSEC'                             000230
00223     83*          6 /6X,'COMMON/COST/CC(19)')                                                           000230
00224     84*           IF(NOX.LT.1) GO TO 105                                                               000230
00224     85*     C --->       FORM /CX/ COMMON                                                              000230
00226     86*           WRITE(9,93)                                                                          000234
00230     87*     93    FORMAT('C --->        STATE VARIABLES')                                              000241
00231     88*           CALL COMGEN(NOX,HCX,8,9)                                                             000241
00232     89*     105   IF(NOV.LT.1) GO TO 120                                                               000250
00232     90*     C --->       FORM /CV/ COMMON                                                              000250
00234     91*           WRITE(9,111)                                                                         000253
00236     92*     111   FORMAT('C --->        VARIABLES')                                                    000260
00237     93*           CALL COMGEN(NOV,2HCV,11,9)                                                           000260
00240     94*     120   IF(NOP.LT.1) GO TO 140                                                               000267
00240     95*     C --->       FORM /CP/ COMMON                                                              000267
```

```
00242         WRITE(9,121)                                                    000272
00244    121  FORMAT("C --->        PARAMETERS")                              000277
00245    130  CALL COMGEN(NOP,NCP,IP,9)                                       000277
00245    C --->      GENERATE TABLE COMMON IN COMO                            000277
00246    140  CALL TABGEN                                                     000306
00246    C           INACTIVATE O.C. PROCESSING                               000306
00246    C --->      GENERATE O.C. COMMONS                                    000306
00246    C    IF(IOCAN.GT.0)CALL OCCOM                                        000306
00247         WRITE(9,151)                                                    000307
00251    151  FORMAT("C --->                 MODEL EQUATIONS")                000314
00251    C --->      TRANSFER CALL SEQUENCE FILE ONTO PROGRAM FILE            000314
00252         END FILE 12                                                     000314
00253         REWIND 12                                                       000317
00254    350  CONTINUE                                                        000323
00255         READ(12,61,END=400,ERR=999)XSOUR                                000323
00260    370  WRITE(9,61)XSOUR                                                000334
00263         GO TO 350                                                       000344
00263    C --->      WRITE RETURN AND ENTRY VARSET AT END OF SUBROUTINE       000344
00264    400  WRITE(9,401)                                                    000346
00266    401  FORMAT(6X,"CALL IMPLIC(CYCLES,DLINES)"                          000352
00266        1 /6X,"IF(IMPL.LT.4)GO TO 1"                                     000352
00266        2 /6X,"IF(CYCLES.GT.0.)IMPL=1"                                   000352
00266        3/6X,"RETURN"/6X,"ENTRY VARSET(TIME,TMAX,INOPT)")                000352
00266    C --->      IVR = 2 FOR VARIABLES.  IVR = 0 FOR STATES.              000352
00267         IVR=2                                                           000352
00267    C --->      TEST THAT THERE ARE VARIABLES IN MODEL                   000352
00270         IF(NOV.LE.0) GO TO 620                                          000354
00272    C ---        TEST FOR MORE THAN 244 VARIABLES                        000354
00272         IF(NOV.GT.244) WRITE(9,411)IVR                                  000357
00276    411   FORMAT(6X,"IF(INOP.GT.244)GO TO 1000",I1)                      000372
00276    C --->      LOAD XSOUR WITH GO TO(                                   000372
00277    420  XSOUR(1)=GT(1)                                                  000372
00300         XSOUR(2)=GT(2)                                                  000373
00301         DO 500 I=3,8                                                    000400
00304    500  XSOUR(I)=BLNK                                                   000400
00306         IXSOUR=11                                                       000402
00307         NGT=0                                                           000404
00307    C --->      SCAN COMPONENTS                                          000404
00310         DO 600 I=1,NOCOMP                                               000410
00310    C --->      GENERATE STATEMENT NO. CORRESPONDING TO EACH COMPONENT   000410
00313         ISN=9000+I                                                      000410
00313    C --->      CONVERT ISN TO BCD FORMAT                                000410
00314         ENCODE(4,501,ISN)ISN                                            000413
00317    501  FORMAT(I4)                                                      000422
00317    C --->      INDEX FOR THE NO. OF VARIABLES (STATES) BEFORE COMPONENT WAS   000422
00320         CALL GETCOD(I,SEQA,ICOMP)                                       000422
00321         J=4*ICOMP-IVR-1                                                 000427
00322         CALL GETCOD(J,IVRSET,NO)                                        000434
00322    C --->      INDEX FOR THE NO. OF VARIABLES (STATES) AFTER COMPONENT WAS    000434
00323         J=4*ICOMP-IVR                                                   000441
00324         CALL GETCOD(J,IVRSET,N1)                                        000445
00324    C --->      TEST TO DETERMINE IF ANY VARIABLES (STATES) WERE FORMED  000445
00325         IF(N1.LE.NO) GO TO 600                                          000452
00327         NO=NO+1                                                         000456
00327    C --->      SCAN THE NO. OF VARIABLES (STATES) FOR THIS COMPONENT    000456
00330         DO 520 J=NO,N1                                                  000461
[illegible]   NGT= NGT+1                                                      000464
```

```
00333          C ---         TEST IF 2ND LEVEL OF GO TO IS REQUIRED                        000464
00334                IF(NGT.LE.244)GO TO 515                                               000466
00336                CALL LINE(0,XSOUR,IXSOUR,RPAR,6,9)                                    000471
00337                WRITE(9,61)XSOUR                                                      000501
00342                WRITE(9,511)IVR                                                       000511
00345          511   FORMAT('1000',I1,' INOP= INOP-244')                                   000517
00346                XSOUR(1)= GT(1)                                                       000517
00347                XSOUR(2)= GT(2)                                                       000521
00350                DO 505 K=3,8                                                          000526
00353          505   XSOUR(K)= BLNK                                                        000526
00355                IXSOUR= 13                                                            000530
00356                NF=0                                                                  000532
00357          515   IF(IXSOUR.NE.13) CALL LINE(0,XSOUR,IXSOUR,COMMA,1,9)                  000534
00357          C --->       PLACE STATEMENT NO. IN COMPUTER GO TO STATEMENT                000534
00361                CALL LINE(0,XSOUR,IXSOUR,ISN,4,9)                                     000546
00362          520   CONTINUE                                                              000561
00364          600   CONTINUE                                                              000561
00364          C --->       COMPLETE GO TO(  STATEMENT                                     000561
00366                CALL LINE(0,XSOUR,IXSOUR,12H),INOP       ,6,9)                        000561
00367                WRITE(9,61)XSOUR                                                      000571
00372                IF(IVR.LE.0) GO TO 700                                                000601
00374          620   IVR=0                                                                 000605
00375                WRITE(9,601)                                                          000605
00377          601   FORMAT(6X,'ENTRY RATSET(TIME,TMAX,INOP)')                             000612
00377          C --->       TEST THAT THERE ARE STATES IN THE MODEL                        000612
00400                IF(NOX.LE.0) GO TO 700                                                000612
00400          C ---        TEST IF 2ND LEVEL OF GO TO IS REQUIRED                         000612
00402                IF(NOX.GT.244) WRITE(9,411)IVR                                        000615
00406                GO TO 425                                                             000627
00406          C =========>       FORM SUBROUTINE DATAIN       ==================          000627
00406          C --->       COMMON AND DIMENSION STATEMENTS                                000627
00407          700   WRITE(9,701)TITLE                                                     000631
00412          701   FORMAT(6X,'END'/'@FOR,IS ASS1.DATAIN,ASRD.DATAIN'/                    000640
00412               16X,'SUBROUTINE DATAIN'/'C'/'C',9X,7A12/'C'/                           000640
00412               2'C --->     THIS SUBROUTINE WAS PREPARED BY THE EASY PRECOMPILER'/    000640
00412               36X,'DOUBLE PRECISION NAMEX,NAMER,NAMEV,NAMEP'/                        000640
00412               46X,'COMMON/COPDER/NOX,NOV,NOP')                                       000640
00412          C --->       TEST IF STATES ARE PRESENT IN MODEL                            000640
00413                IF(NOX.LT.1) GO TO 740                                                000640
00413          C --->       FORM STATE RELATED COMMONS                                     000640
00415                WRITE(9,711)(NOX,I=1,10)                                              000644
00423          711   FORMAT('C --->       STATE RELATED COMMONS'/                          000655
00423               16X,'COMMON/CX/X(',I4,')/CXDOT/XDOT(',I4,')/CXIC/XIC(',I4,')'/         000655
00423               25X,'1 /CXIC1/XIC1(',I4,')/CXIC2/XIC2(',I4,')/CXIC3/XIC3(',I4,')'/     000655
00423               35X,'2 /CINT/INT(',I4,')/CNAMEX/NAMEX(',I4,')/CNAMER/NAMER(',I4,')'    000655
00423               4/5X,'3 /CNTRLS/AN,IPRNT,MODE,ERROR(',I4,')')                          000655
00423          C --->       CALCULATE THE AMOUNT OF WORK SPACE REQ'D.                      000655
00424                NO=NOX*(2*NOX+7)                                                      000655
00425                IF(NO.LT.1000)NO=1000                                                 000662
00427                WRITE(9,719)NO                                                        000670
00432          719   FORMAT(6X,'COMMON/CWORK/CWORK(',I5,')')                               000677
00432          C --->       TEST IF VARIABLES ARE PRESENT IN MODEL                         000677
00433          740   IF(NOV.LT.1) GO TO 780                                                000677
00435                WRITE(9,741)NOV,NOV                                                   000702
00441          741   FORMAT('C --->       VARIABLE RELATED COMMONS'/                       000712
00441               16X,'COMMON /CV/V(',I4,')/CNAMEV/NAMEV(',I4,')')                       000712
00441          C --->       TEST IF PARAMETERS ARE PRESENT IN MODEL                        000712
```

```
00442     780   IF(NOP.LT.1) GO TO 800                                                            000712
00444           WRITE(9,781)NOP,NOP                                                               000715
00450     781   FORMAT('C --->      PARAMETER RELATED COMMONS'/                                   000725
00450          16X,'COMMON /CP/P(',I4,')/CNAMEP/NAMEP(',I4,')')                                   000725
00453     C --->      LOAD NO. OF STATE,VARIABLE, AND PARAMETERS INTO COMMONS                     000725
00453     800   WRITE(9,821)NOX,NOV,NOP                                                           000725
00456     821   FORMAT('C --->      SET NO. OF STATES,VARIABLES, AND PARAMETERS'/                 000734
00456          16X,'NOX=',I4/6X,'NOV=',I4/6X,'NOP=',I4)                                           000734
00457           IF(NOX.LE.0) GO TO 850                                                            000734
00457     C --->      LOAD STATE ERROR AND PARAMETER DEFAULT VALUES INTO COMMONS                  000734
00461           WRITE(9,831)                                                                      000737
00463     831   FORMAT('C --->      LOAD STATE ERROR DEFAULT VALUES'/                             000744
00463          16X,'DO 100 I=1,NOX'/6X,'ERROR(I)=.1')                                             000744
00464           IF(DFNAME.EQ.ECS)WRITE(9,833)                                                     000744
00467     833   FORMAT(6X,'CALL GETT(NAMEX(I),1,KAR)'/6X,'IF(KAR.EQ.HT)ERROR(I)=1                 000754
00467          3.'/6X,'IF(KAR.EQ.HP)ERROR(I)=.005')                                               000754
00470           WRITE(9,841)                                                                      000754
00472     841   FORMAT('100   CONTINUE')                                                          000762
00473     850   IF(NOP.LE.0) GO TO 860                                                            000762
00475           WRITE(9,851)                                                                      000764
00477     851   FORMAT('C --->      LOAD PARAMETER DEFAULT VALUES'/                               000772
00477          16X,'DO 300 I=1,NOP'/'300    P(I)=.99999'/                                         000772
00477          26X,'WRITE(6,301)'/'301    FORMAT(1H1)')                                           000772
00500     860   REWIND 12                                                                         000772
00500     C --->      START FORMATION OF INPUT REQUIREMENTS LIST                                  000772
00501           WRITE(12,861)TITLE,NOCOMP,NOTAB,NOP,NOX,NOV                                       000774
00511     861   FORMAT(//10X,7A10//5X,'THIS MODEL CONTAINS ',I4,' COMPONENTS'/                    001012
00511          15X,'WITH  ',I4,' TABLES',2X,I4,' PARAMETERS',2X,I4,' STATES AND'                  001012
00511          22X,I4,' VARIABLES.'                                                               001012
00511          2//10X,'INPUT DATA REQUIREMENTS LIST'/)                                            001012
00512           MAXT=0                                                                            001012
00513           IF(NOTAB.LE.0)GO TO 864                                                           001013
00515           CALL TABCAL                                                                       001016
00515     C ==============      COMPLETE DATAIN SUBROUTINE  ==  START BLOCK DATA MODEL  ==        001016
00515     C                                                                                       001016
00515     C ---       CALCULATE TOTAL STORAGE REQUIRED BY MODEL TABLES                            001016
00516           DO 862 I=1,NOTAB                                                                  001020
00521           CALL GETCOD(5,TABNAM(I),N)                                                        001025
00522           MAXT=MAXT+IABS(N)                                                                 001033
00523     862   CONTINUE                                                                          001040
00523     C ---       TESTS TO PREVENT DIMENSIONS < 1                                             001040
00525     864   NOVP=MAX0(NOV,1)                                                                  001044
00526           NOPP=MAX0(NOP,1)                                                                  001045
00527           MAXTP=MAX0(MAXT,1)                                                                001053
00530           NOTABP=MAX0(NOTAB,1)                                                              001061
00531           WRITE(9,865)NOXP,NOVP,NOPP,MAXTP,NOTABP,NOTABP,NOTABP                             001067
00542     865   FORMAT(6X,'RETURN'/6X,'END'/'@FOR,IS ASST.MODEL,ASRO.MODEL'/                      001103
00542          16X,'BLOCK DATA MODEL'/'C --->      MODEL NAME COMMONS'/                           001103
00542          26X,'DOUBLE PRECISION NAMEX,NAMEV,NAMEP,TABNAM'/                                   001103
00542          36X,'COMMON/CNAMEX/NAMEX(',I4,')/CNAMEV/NAMEV(',I4,                                001103
00542          4')/CNAMEP/NAMEP(',I4,')'/5X,'1/CTABLE/TABLES(',I4,')/CTABNA/TABNAM                001103
00542          5(',I3,')'/                                                                        001103
00542          65X,'2/CMAXDT/NOTAB,MAXDIM(',I3,')/CLOCTA/LOCTAB(',I3,')')                         001103
00542     C --->      CREATE EQUIVALENCE STATEMENTS IF NEEDED TO ALLOW DATA                       001103
00542     C --->      STATEMENTS TO LOAD NAME LISTS EXCEEDING 130 NAMES                           001103
0               CALL CONEQU(2HNAMEX      ,NOX)                                                    001103
0               CALL CONEQU(2HNAMEV      ,NOV)                                                    001107
```

```
00545         CALL COMCRU(12HNAMEP       ,NOP)                                 001113
00545   C --->      TEST FOR O.C.  IF YES CALL OCBLKD                          001113
00545   C         INACTIVATE O.C. PROCESSING                                   001113
00545   C     IF(IOCAN.GT.0)CALL OCBLKD                                        001113
00545   C --->      GENERATE NAME DATA STATEMENTS                              001113
00546         WRITE(9,867)                                                     001117
00550   867   FORMAT('C --->                 MODEL DATA STATEMENTS')          001124
00550   C --->      GENERATE STATE, VARIABLE, AND PARAMETER NAME DATA STATEMENTS  001124
00551         CALL NAMARY(12HNAMEX       ,5,NOX,8)                             001124
00552         CALL NAMARY(12HNAMEV       ,5,NOV,11)                            001132
00553         CALL NAMARY(12HNAMEP       ,5,NOP,10)                            001140
00553   C --->      CALCULATE NO. OF WORDS IN TABLES (LESS FLIGHT TABLES)      001140
00553   C ---       GENERATE TABLE NAMES, MAX DIMENSIONS,   LOCATIONS          001140
00554         CALL TABDAT                                                      001146
00554   C ==========  TABLE INITIATION  ==========                             001146
00555         WRITE(9,869)MAXTP                                                001155
00560   869   FORMAT(6X,'DATA TABLES/',I5,9H*1.99999//6X,'END')                001156
00561   880   IF(NOP.LE.0) GO TO 960                                           001156
00561   C --->      ADD PARAMETERS AND STATES TO INPUT REQUIREMENTS LIST       001156
00563         NUNIT=10                                                         001161
00564         NI=NOP                                                           001163
00565         WRITE(12,881)                                                    001165
00567   881   FORMAT(///14X,'PARAMETERS REQUIRED'//                            001173
00567        111X,'COMPONENT',5X,'PARAMETER'/                                  001173
00567        215X,'NAME',10X,'NAME')                                           001173
00570   900   REWIND NUNIT                                                     001173
00571         COMPS=BLNK                                                       001175
00572         DO 940 I=1,NI                                                    001203
00572   C --->      SCAN PARAMETER (STATE) LIST                                001203
00575         READ(NUNIT,901)ANAME                                             001203
00600   901   FORMAT(A7)                                                       001210
00601         CALL STRMOV(ANAME,4,4,COMP,1)                                    001212
00601   C --->      COMPARE CURRENT COMPONENT NAME WITH PREVIOUS NAME          001215
00602         IF(COMPS.EQ.COMP) GO TO 920                                      001217
00604         WRITE(12,911)                                                    001222
00606   911   FORMAT(1H )                                                      001227
00607         COMPS=COMP                                                       001227
00610   920   WRITE(12,921)COMP,ANAME                                          001232
00614   921   FORMAT(15X,A4,9X,A7)                                             001242
00615   940   CONTINUE                                                         001242
00617   960   CONTINUE                                                         001242
00620         IF(NOX.LE.0) RETURN                                              001242
00622         IF(NUNIT.EQ.8) RETURN                                            001247
00624         NUNIT=8                                                          001255
00625         NI=NOX                                                           001257
00626         WRITE(12,961)                                                    001261
00630   961   FORMAT(///18X,'STATES'/                                          001266
00630        12X,'(INITIAL CONDITIONS AND ERROR CONTROLS REQUIRED)'//          001266
00630        213X,'COMPONENT',6X,'STATE'/15X,'NAME',10X,'NAME')                001266
00631         GO TO 900                                                        001266
00632     999 RETURN                                                           001270
00633         END @ ENGMOD  ******************************                     001325
```

```
SUBROUTINE HLINE     ENTRY POINT 000114

STORAGE USED  CODE(1) 000130; DATA(0) 000026; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   PUTT
 0004   KOMSTR
 0005   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000021 100L      0001     000042 124G      0001     000027 200L      0001     000100 300L      0000 D 000002 HGT
 0000 D 000003 HLT       0000 I 000010 I         0000     000015 INJP$     0000 I 000006 I1        0000 I 000007 I2
 0004 I 000000 KOMSTR    0000 D 000004 POINT


00100      CHLINE                                                                        000005
00101            SUBROUTINE HLINE(PAGE,LINE,IN,IR)                                       000005
00101      C  PURPOSE   ADD A HORIZONTAL CONNECTION LINE TO ECS SCHEMATIC                000005
00101      C  CALL SEQUENCE  PAGE - 13X56 ARRAY CONTAINING HOLLORITH                     000005
00101      C                           REPRESENTATION OF A PAGE                          000005
00101      C                    LINE - LINE NO. FOR HORIZONTAL LINE                      000005
00101      C                    IN   - INPUT COMPONENT COL. LOCATION                     000005
00101      C                    IR   - RECEIVING COMPONENT COL. LOCATION                 000005
00103                              IMPLICIT DOUBLE PRECISION (A-Z)                       000005
00104                              IMPLICIT INTEGER (I,J,K,L,M,N)                        000005
00105            DIMENSION PAGE(13,56)                                                   000005
00105      C     LITERAL "POOL" TO SATISFY DBLE PRECSN ASSGNMNT STMNTS                   000005
00106            DATA HLT/"<              "/,HGT/">                                      000005
00106      C --->       IS INPUT COMP. ON LEFT OR RIGHT                                  000005
00111            IF(IN.GE.IR)GO TO 100                                                   000005
00113            POINT=HGT                                                               000011
00114            I1=IN                                                                   000013
00115            I2=IR                                                                   000015
00116            GO TO 200                                                               000017
00116      C --->       INPUT IS ON RIGHT                                                000017
00117      100   POINT=HLT                                                               000021
00120            I1=IR                                                                   000022
00121            I2=IN                                                                   000024
00121      C --->       PLACE POINT ON RECEIVING END OF LINE                             000024
00122      200   CALL PUTT(PAGE(1,LINE),IR,POINT)                                        000027
00122      C --->       ADD NO. OF SYMBOLS REQ"D. TO SPAN COLUMNS                        000027
00123            DO 300 I=I1,I2                                                          000035
00123      C --->       TEST TO PREVENT OVERWRITING POINTS                               000035
00126            IF(KOMSTR(PAGE(1,LINE),I,1,HLT,1).EQ.0)GO TO 300                        000042
00133            IF(KOMSTR(PAGE(1,LINE),I,1,HGT,1).EQ.0)GO TO 300                        000055
00         C --->       ADD HORIZONTAL LINE SYMBOL                                       000055
```

```
00132      32*              CALL PUTT(PAGE(I,LINE),T,12H=                                                                      000070
00133      33*        300   CONTINUE                                                                                           000101
00135      34*              RETURN                                                                                             000101
00136      35*              END $ HLINE  ******************************                                                        000127
```

```
FUNCTION IJBIT       ENTRY POINT 000044

STORAGE USED  CODE(1) 000050; DATA(0) 000015; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000 I 000001 IBIT      0000 I 000000 IJBIT     0000   000005 INJP$     0000 I 000002 IWORD     0000 I 000003 LBIT


00100       1*      CIJBIT                                                                      000002
00101       2*           FUNCTION IJBIT(A,I,J,N)                                                000002
00101       3*      C  VERSION 1.                        REVISED  AUG 7 1975                    000002
00101       4*      C  PURPOSE   SET IJBIT EQUAL TO THE I J ELEMENT IN BINARY ARRAY A           000002
00101       5*      C  CALL SEQUENCE    A - N X N BINARY ARRAY                                  000002
00101       6*      C                   I - ROW INDEX                                           000002
00101       7*      C                   J - COLUMN INDEX                                        000002
00101       8*      C                   N - COLUMN DIMENSION OF ARRAY                           000002
00101       9*      C  DESIGNED BY  J.D. BURROUGHS                 JULY 1975                    000002
00103      10*           DIMENSION A(1)                                                         000002
00104      11*           IBIT=I+(J-1)*N-1                                                       000002
00105      12*           IWORD=IBIT/36 + 1                                                      000010
00106      13*           LBIT=MOD(IBIT,36)                                                      000014
00107      14*           IJBIT = 0                                                              000020
00110      15*           FLD(35,1,IJBIT) = FLD(LBIT,1,A(IWORD))                                 000021
00111      16*           RETURN                                                                 000032
00112      17*           END   IJBIT  ******************************                             000047
```

```
SUBROUTINE IJBIT1    ENTRY POINT 000050

STORAGE USED  CODE(1) 000055; DATA(0) 000017; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000 I 000000 IBIT      0000   000005 INJP$    0000 I 000001 IWORD     0000 I 000003 I11       0000 I 000002 LBIT


00100        CIJBIT1                                                                                000002
00101               SUBROUTINE IJBIT1(A,I,J,N)                                                      000002
00101        C  VERSION 1.                     REVISED  AUG 7 1975                                  000002
00101        C  PURPOSE   LOAD 1 IN I J LOCATION OF N BY N BINARY ARRAY A.                          000002
00101        C  CALL SEQUENCE    A - N X N BINARY ARRAY                                             000002
00101        C                   I - ROW INDEX                                                      000002
00101        C                   J - COLUMN INDEX                                                   000002
00101        C                   N - COLUMN DIMENSION OF ARRAY                                      000002
00101        C  DESIGNED BY  J.D. BURROUGHS                JULY 1975                                000002
00103               DIMENSION A(1)                                                                  000002
00104               IBIT=I+(J-1)*N-1                                                                000002
00105               IWORD=IBIT/36 + 1                                                               000010
00106               LBIT=MOD(IBIT,36)                                                               000014
00107               I11 = 1                                                                         000020
00110               FLD(LBIT,1,A(IWORD)) = FLD(35,1,I11)                                            000022
00111               RETURN                                                                          000036
00112               END @ IJBIT1  ************************                                          000054
```

```
SUBROUTINE INCOM      ENTRY POINT 001040

STORAGE USED  CODE(1) 001152; DATA(0) 000521; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CIO     000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0004    COMOAT
 0005    KOMSTR
 0006    PUTCOD
 0007    CLTCOD
 0010    FEAOMS
 0011    NXTFM
 0012    MLMERC
 0013    CLTT
 0014    NAMBCN
 0015    STPMOV
 0016    FORICN
 0017    WATIMS
 0020    NWDPS
 0021    NI02$
 0022    NI01$
 0023    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000075 120L      0001    000024 122G      0001    000135 130L      0001    000196 140L      0001    000143 150G
 0001    000231 160L      0000    000346 161F      0001    000241 180L      0001    000246 204G      0001    000266 220L
 0001    000320 222G      0001    000313 240L      0001    000400 244G      0001    000435 255G      0000    000374 261F
 0001    000347 280L      0001    000370 300L      0001    000633 316G      0001    000432 320L      0001    000452 323G
 0001    000750 347F      0001    000762 356G      0001    000467 363L      0001    000775 365G      0001    001013 374G
 0001    000472 390L      0001    000476 400L      0001    000553 420L      0001    000602 440L      0001    000620 500L
 0001    000673 540L      0001    000677 560L      0000    000421 571F      0001    001014 600L      0000    000437 801F
 0000    000444 803F      0000    000451 805F      0000    000456 807F      0000 D  000316 BLNK      0000 D  000343 CINFAM
 0000 R  000335 OPORT     0000 D  000323 HINPT     0000 D  000322 HOUTP     0000 D  000314 H777      0000 I  000330 I
 0003 I  000002 IOTAG     0003    000475 INJP$     0003 I  000340 IPHAS     0003    000000 IPEAD     0000 I  000332 JUCOMP
 0003 I  000001 IWRITE    0000 I  000345 J         0005 I  000000 KOMSTR    0000 I  000337 MODE      0000 I  000324 NOCON
 0000 I  000331 NDINPT    0000 I  000327 NUOUT     0000 D  000325 UCOMNM    0000 D  000000 UINPUT    0000 D  000146 UOUT
 0000 D  000341 UOUTNM    0000 D  000333 UPORT


00100       1*      CINCOM                                                                                       000005
00101       2*          SUBROUTINE INCOM(ICOM,PHAS,INDEX,NDINPT,DINPUT,NDOUT,                                    000005
            3*         1DOUT,DCONNM,CMPMOD,NOCOMP,ICOMP)                                                         000005
            4*      C  VERSION 2.                               REVISED  DEC 16 1975                             000005
```

```
00101   C  PURPOSE   PERFORM INPUT-OUTPUT CONNECTIONS BETWEEN STD. COMPS.           000005
00101   C  CALL SEQUENCE   ICOM    - COMMAND STRING ARRAY                           000005
00101   C                  PHRS    - CURRENT PHRASE (UPSTREAM COMP. NAME UPON ENT   000005
00101   C                  INDEX   - COMMAND STRING CHARACTER INDEX                 000005
00101   C                  NDINPT- NO. OF INPUTS FOR DOWNSTREAM COMP.               000005
00101   C                  DINPUT - LIST OF INPUT QUANTITY NAMES FOR DOWNSTREAM     000005
00101   C                            COMPONENT                                      000005
00101   C                  NDOUT   - NO. OF OUTPUTS FOR DOWNSTREAM COMP.            000005
00101   C                  DOUT    - LIST OF OUTPUT QUANTITY NAMES FOR DOWNSTREAM   000005
00101   C                  DCOMNM- SPECIFIC COMPONENT NAME OF DOWNSTREAM COMP.      000005
00101   C                  CMPMOD - LIST OF COMPONENTS IN CURRENT MODEL             000005
00101   C                  NOCOMP - NO. OF COMP. IN CURRENT MODEL                   000005
00101   C                  ICOMP   - UPSTREAM COMP. TYPE                            000005
00103                       IMPLICIT DOUBLE PRECISION (A-Z)                         000005
00104                       IMPLICIT INTEGER (I,J,K,L,M,N)                          000005
00105                       DOUBLE PRECISION ICOM,ICOMP                             000005
00106         COMMON /CIO/IREAD,IWRITE,IDIAG                                        000005
00107         DIMENSION ICOM(8),DINPUT(1),DOUT(1),UINPUT(51),UOUT(51),CMPMOD(1)     000005
00107   C     LITERAL "POOL" TO SATISFY DBLE PRECSN ASSGNMNT STMNTS                 000005
00110         DATA HZZZ/'ZZZ         '/,BLNK/'                                      000005
00113         DATA HINPT/'INPT         '/,HOUTP/'OUTP                               000005
00113   C --->       NO. OF CONNECTIONS INDICATOR                                   000005
00116         NCCON=0                                                               000005
00116   C --->       SAVE UPSTREAM COMPONENT NAME                                   000005
00117         UCOMNM=PHRS                                                           000006
00117   C --->       GET LIST OF UPSTREAM COMP. OUTPUTS                             000006
00120         CALL COMDAT(UCOMNM,HOUTP,NUOUT,UOUT)                                  000010
00120   C --->       SCAN COMP. IN CURRENT MODEL                                    000010
00121         DO 100 I=1,NOCOMP                                                     000024
00121   C --->       TEST TO SEE IF UPSTREAM COMP. HAS BEEN DEFINED                 000024
00124         IF(KOMSTR(CMPMOD(I),1,4,UCOMNM,1).EQ.0)GO TO 120                      000024
00126   100   CONTINUE                                                              000042
00126   C --->       GET STD. INPUT LIST FOR UPSTREAM COMP.                         000042
00130         CALL COMDAT(UCOMNM,HINPT,NUINPT,UINPUT)                               000042
00130   C --->       STORE COMP. LOC.=-100, COMP TYPE, NO. INPUTS FOR UPSTREAM CO   000042
00131         CALL PUTCOD(3,UCOMNM,-100)                                            000050
00132         CALL PUTCOD(5,UCOMNM,NUINPT)                                          000055
00132   C --->       INCREMENT MODEL COMP. COUNT                                    000055
00133         NOCOMP=NOCOMP+1                                                       000062
00133   C --->       ADD COMP. NAME TO CURRENT MODEL LIST                           000062
00134         CMPMOD(NOCOMP)=UCOMNM                                                 000065
00135         IUCOMP=NOCOMP                                                         000071
00136         GO TO 140                                                             000073
00136   C --->       GET INPUT LIST FOR EXISTING COMP.                              000073
00137   120   IUCOMP=I                                                              000075
00140         CALL GETCOD(5,CMPMOD(I),NUINPT)                                       000076
00140   C --->       TEST FOR COMPONENT DRIVING ITSELF                              000076
00141         IF(KOMSTR(UCOMNM,1,4,DCOMNM,1).EQ.0)GO TO 130                         000107
00141   C --->       GET INPUT LIST FROM FILE 7                                     000107
00143         UINPUT(1)=HZZZ                                                        000120
00144         IF(NUINPT.GT.0)CALL READMS(7,UINPUT,NUINPT,IUCOMP)                    000122
00146         GO TO 140                                                             000133
00146   C --->       LOAD UPSTREAM INPUTS FROM DOWNSTREAM INPUTS LIST               000133
00147   130   DO 135 I=1,NUINPT                                                     000135
00152   135   UINPUT(I)=DINPUT(I)                                                   000143
00152   C --->       DEFAULT ON PORT DESIGNATION IS BLANK (UNIVERSAL PORT)          000143
00154   140   UPORT=BLNK                                                            000146
```

```
00155          DPORT=BLNK                                                      000147
00156          MODE=1                                                          000150
00157          CALL NXTPH(ICOM,INDEX,PHRS)                                     000152
00160          IPHRS=1                                                         000157
00161          IF(KOMSTR(PHRS,1,1,BLNK,1).EQ.0)GO TO 500                       000161
00161    C --->      TEST FOR NUMERIC, I.E. PORT NUMBER                        000161
00163          CALL NUMERC(PHRS,$180)                                          000172
00163    C --->      SAVE NUMERIC PORT NO.                                     000172
00164          MODE=1                                                          000176
00165          UPORT=PHRS                                                      000200
00166          CALL NXTPH(ICOM,INDEX,PHRS)                                     000202
00167          IF(KOMSTR(PHRS,1,1,BLNK,1).EQ.0)GO TO 160                       000207
00167    C --->      TEST FOR NUMERIC, I.E. PORT NUMBER                        000207
00171          CALL NUMERC(PHRS,$160)                                          000220
00171    C --->      SAVE DOWNSTREAM PORT NO.                                  000220
00172          DPORT=PHRS                                                      000224
00173          IPHRS=0                                                         000226
00174          GO TO 420                                                       000227
00175    160   WRITE(IWRITE,161)PHRS,UCOMNM                                    000231
00201    161 - FORMAT(/5X,18H *** WARNING ***  ,A10,'IS NOT A VALID PORT DESIGNAT 000237
00201         1ION FOR INPUT COMPONENT  ',A4,'.   ERRONEOUS CONNECTIONS MAY OCCUR 000237
00201         2')                                                              000237
00202          GO TO 420                                                       000237
00202    C --->      SCAN UPSTREAM OUTPUTS                                     000237
00203    180   DO 200 I=1,NUOUT                                                000241
00206          IF(KOMSTR(UOUT(I),1,3,PHRS,1).EQ.0)GO TO 220                    000246
00210    200   CONTINUE                                                        000264
00212          GO TO 500                                                       000264
00212    C --->      SAVE OUTPUT NAME                                          000264
00213    220   UOUTNM=UOUT(I)                                                  000266
00214          MODE=0                                                          000271
00215          CALL NXTPH(ICOM,INDEX,PHRS)                                     000272
00216          CALL NUMERC(PHRS,$240)                                          000277
00216    C --->      SAVE UPSTREAM PORT NO.                                    000277
00217          UPORT=PHRS                                                      000303
00220          CALL NXTPH(ICOM,INDEX,PHRS)                                     000305
00220    C --->      SCAN DOWNSTREAM INPUTS                                    000305
00221    240   DO 260 I=1,NDINPT                                               000313
00224          IF(KOMSTR(DINPUT(I),1,3,PHRS,1).EQ.0)GO TO 280                  000320
00226    260   CONTINUE                                                        000336
00230          WRITE(IWRITE,261)PHRS,DCOMNM                                    000336
00234    261   FORMAT(/5X,18H *** WARNING ***  ,A10,'IS NOT A VALID INPUT QUANTIT 000345
00234         1Y OR PORT DESIGNATION FOR COMPONENT  ',A4)                      000345
00235          GO TO 500                                                       000345
00236    280   DINNAM=DINPUT(I)                                                000347
00237          CALL NXTPH(ICOM,INDEX,PHRS)                                     000353
00240          CALL NUMERC(PHRS,$300)                                          000360
00241          DPORT=PHRS                                                      000364
00242          IPHRS=0                                                         000366
00242    C --->      SEARCH FOR MATCH BETWEEN NAMES   PORT NO. GIVEN ABOVE     000366
00243    300   DO 380 I=1,NDINPT                                               000370
00243    C --->      TEST FOR NAME MATCH                                       000370
00246          IF(KOMSTR(DINPUT(I),1,3,DINNAM,1).NE.0)GO TO 380                000400
00246    C --->      BYPASS PORT TEST IF PORT NOT SPECIFIED                    000400
00250          IF(DPORT.EQ.BLNK )GO TO 320                                     000413
00250    C --->      DOWNSTREAM PORT TEST                                      000413
[illegible]    IF(KOMSTR(DINPUT(I),9,1,DPORT,1).NE.0)GO TO 380                 000416
```

```
00252    C --->       SCAN UPSTREAM OUTPUTS                                                000416
00254    320   DO 360 J=1,NUOUT                                                            000435
00254    C --->       TEST FOR NAME MATCH                                                  000435
00257          IF(KOMSTR(UOUT(J),1,3,UOUTNM,1).NE.0)GO TO 360                              000435
00257    C --->       TEST IF PORT IS SPECIFIED                                            000435
00261          IF(UPORT.EQ.BLNK )GO TO 400                                                 000450
00261    C --->       TEST FOR PORT MATCH                                                  000450
00263          IF(KOMSTR(UOUT(J),9,1,UPORT,1).EQ.0)GO TO 400                               000453
00265    360   CONTINUE                                                                    000474
00267    380   CONTINUE                                                                    000474
00271          GO TO 560                                                                   000474
00271    C --->       SATISFY SPECIFIC INPUT                                               000474
00271    C --->       GET UPSTREAM AND DOWNSTREAM PORT NOS.                                000474
00272    400   CALL GETI(UOUT(J),9,UPORT)                                                  000476
00273          CALL GETI(DINPUT(I),9,DPORT)                                                000507
00274          CALL NAMGEN(UOUT(J),UCONNM,DINPUT(I))                                       000522
00274    C --->       TAG INPUT AS FROM AN UPSTREAM SOURCE                                 000522
00275          CALL STRMOV(BLNK,1,1,DINPUT(I),8)                                           000535
00276          NOCON=1                                                                     000546
00277          IF(MODE.EQ.0)GO TO 440                                                      000550
00277    C --->       SATISFY ALL OTHER INPUTS USING OUTPUTS OF SPECIFIED PORTS            000550
00301    420   CALL PORTCN(DINPUT,NDINPT,UOUT,NUOUT,DPORT,UPORT,UCONNM,NOCON,              000553
00301        1 BLNK)                                                                       000553
00301    C --->       SATISFY UPSTREAM INPUTS                                              000553
00302          CALL PORTCN(UINPUT,NUINPT,DOUT,NDOUT,UPORT,DPORT,DCONNM,NOCON,              000565
00302        1 12HD)                                                                       000565
00303          GO TO 560                                                                   000600
00304    440   UPORT=BLNK                                                                  000602
00305          DPORT=BLNK                                                                  000603
00306          IF(IPHRS.EQ.1)GO TO 180                                                     000604
00310          CALL MATPH(ICOM,INDEX,PHRS)                                                 000607
00311          IPHRS=1                                                                     000614
00312          GO TO 181                                                                   000616
00313    500   IF(MODE.EQ.0)GO TO 560                                                      000620
00313    C --->       REGULAR CONNECTION ROUTINE                                           000620
00313    C --->       SCAN DOWNSTREAM INPUTS                                               000620
00315          DO 540 I=1,NDINPT                                                           000621
00315    C --->       TEST IF INPUT IS SATISFIED                                           000621
00320          IF(KOMSTR(DINPUT(I),9,1,BLNK,1).NE.0)GO TO 540                              000633
00322    C --->       SCAN UPSTREAM OUTPUTS                                                000633
00322          DO 520 J=1,NUOUT                                                            000652
00322    C --->       TEST FOR NAME MATCH                                                  000652
00325          IF(KOMSTR(DINPUT(I),1,3,UOUT(J),1).EQ.0)GO TO 400                           000652
00327    520   CONTINUE                                                                    000677
00331    540   CONTINUE                                                                    000677
00333    560   IF(NOCON.LE.0)WRITE(IWRITE,571)UCONNM,DCONNM                                000677
00340    571   FORMAT(/5X,21H *** WARNING *** NO  ,A4,"  OUTPUTS MATCH UNSATISF            000710
00340       1IED",A4,"  INPUTS")                                                           000710
00340    C --->       STORE UPSTREAM INPUT LIST                                            000710
00341          IF(NUINPT.GT.0)CALL WRITMS(7,UINPUT,NUINPT,IUCOMP)                          000710
00343          IF(IDIAG.LE.70)GO TO 600                                                    000732
00345          WRITE(IWRITE,801)(UINPUT(I),I=1,NUINPT)                                     000740
00353    801   FORMAT(" INCOM-UINPUTS"/(1X,6A10))                                          000753
00354          WRITE(IWRITE,803)(UOUT(I),I=1,NUOUT)                                        000753
00362    803   FORMAT(" INCOM-UOUT"/(1X,6A10))                                             000765
00363          WRITE(IWRITE,805)(DINPUT(I),I=1,NDINPT)                                     000765
00371    805   FORMAT(" INCOM-DINPUT"/(1X,6A10))                                           001000
```

```
00372     176*              WRITE(IWRITE,807)(DOUT(I),I=1,NDOUT)                        001003
00403     177*        807   FORMAT(' INCOM-DOUT'/(1X,6A10))                             001014
00400     178*        C --->       TEST IF NEXT PHRASE HAS BEEN USED                    001014
00401     179*        600   IF(IPHRS.EQ.0)CALL NXTPH(ICOM,INDEX,PHRS)                   001014
00403     180*              RETURN                                                      001022
00404     181*              END @ INCOM  *****************************                  001151
```

```
SUBROUTINE LINE      ENTRY POINT 000144


STORAGE USED  CODE(1) 000203; DATA(0) 000024; BLANK COMMON(2) 000000


EXTERNAL REFERENCES (BLOCK, NAME)

 0003    STRMOV
 0004    NERR3$
 0005    NIO1$
 0006    NIO2$
 0007    NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000    000007 101F       0001    000022 117G       0001    000033 126G       0001    000076 142G       0001    000107 150G
 0001    000040 300L       0001    000055 400L       0000 D 000002 BLNK       0000 I 000004 I          0000    000013 INJP$
 0000 I 000006 L          0000 I 000005 NO         0000 D 000000 X



00100       1*      CLINE                                                                              000003
00101       2*            SUBROUTINE LINE(MODE,SOURCE,ISOUR,TEXT,N,NTAPE)                              000003
00101       3*      C  PURPOSE   TO CONTROL THE FLOW OF SOURCE TEXT AND GENERATE                       000003
00101       4*      C               CONTINUES AS NEEDED TO STAY WITHIN COLUMNS 1 - 72                  000003
00101       5*      C  CALL SEQUENCE   MODE   - MODE=0 -> NEW LINE IS STARTED BEGINING WITH            000003
00101       6*      C                            MODE=1 -> TEXT IS SPLIT TO FIT EXACTLY 7-72           000003
00101       7*      C                  ISOUR  - NEXT CHARACTER FOR WRITING                             000003
00101       8*      C                  TEXT   - NEW TEXT STRING                                        000003
00101       9*      C                  N      - NO. OF CHARACTERS TO ADD                               000003
00101      10*      C                  NTAPE  - FILE TO WRITTEN TO                                     000003
00103      11*                          IMPLICIT DOUBLE PRECISION (A-Z)                                000003
00104      12*                          IMPLICIT INTEGER (I,J,K,L,M,N)                                 000003
00105      13*            DIMENSION SOURCE(8)                                                          000003
00106      14*            DATA X/12H      X       /,BLNK/12H                                           000003
00106      15*      C --->       TEST FOR END OF LINE                                                  000003
00111      16*            IF(ISOUR+N.LE.73) GO TO 300                                                  000003
00113      17*            IF(MODE.NE.0) GO TO 400                                                      000010
00113      18*      C --->       NEW LINE REQUIRED                                                     000010
00113      19*      C --->       WRITE CURRENT LINE                                                    000010
00115      20*            WRITE(NTAPE,101)SOURCE                                                       000012
00123      21*      101   FORMAT(8A10)                                                                 000025
00123      22*      C --->       GENERATE CONTINUE SYMBOL                                              000025
00124      23*            SOURCE(1)=X                                                                  000025
00125      24*            DO 200 I=2,8                                                                 000033
00130      25*      200   SOURCE(I)=BLNK                                                               000033
00132      26*            ISOUR=7                                                                      000035
00133      27*      300   CALL STRMOV(TEXT,1,N,SOURCE,ISOUR)                                           000040
00134      28*            ISOUR=ISOUR+N                                                                000046
00135      29*            RETURN                                                                       000054
```

```
00135      C --->       MODE=1 SPLIT TEXT BETWEEN CURRENT AND NEXT LINE                    000051
00136      400    NO=73-ISOUR                                                               000055
00136      C --->       COMPLETE CURRENT LINE                                               000055
00137             CALL STRMOV(TEXT,1,NO,SOURCE,ISOUR)                                       000057
00140             WRITE(NTAPE,101)SOURCE                                                    000066
00146             SOURCE(1)=X                                                               000101
00147             DO 420 I=2,8                                                              000107
00152      420    SOURCE(I)=BLNK                                                            000107
00152      C --->       NO. CHARACTERS LEFT IN TEXT                                         000107
00154             L=N-NO                                                                    000111
00154      C --->       NEXT CHARACTER IN TEXT TO MOVE                                      000111
00155             NO=NO+1                                                                   000114
00156             CALL STRMOV(TEXT,NO,L,SOURCE,7)                                           000117
00157             ISOUR=L+7                                                                 000126
00160             RETURN                                                                    000131
00161             END a LINE   ******************************                               000202
```

```
SUBROUTINE LISTSC    ENTRY POINT 000244

STORAGE USED  CODE(1) 000265; DATA(0) 000177; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   C10    000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0004   READMS
 0005   COMPAT
 0006   GETT
 0007   GETCOD
 0010   NRDU$
 0011   NIO2$
 0012   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000    000066 301F       0001    000322 125G       0001    000102 141G       0000    000076 521F       0001    000130 530L
 0001    000150 535L       0001    000174 540L       0000    000140 543F       0000    000150 563F       0000 D 000047 AIN
 0000 D 000032 BLNK       0000 D 000034 HONE       0000 D 000036 HTWO       0000 I 000042 I          0000 I 000031 IBLNK
 0000 I 000045 ID         0003   000002 IDIAG      0000   000157 INJP$      0000 D 000001 IP         0003   000000 IREAD
 0000 D 000003 IV         0003 I 000001 IWRITE     0000 I 000046 J          0000 I 000000 MAX        0000 I 000043 NI
 0000 I 000044 NO         0000 I 000045 NT         0000 D 000056 OP         0000 D 000051 OUT        0000 D 000040 PFNAME
 0000 D 000060 ST         0000 D 000053 TAB        0000 D 000005 TABLE


00100         CLISTSC                                                                      000002
00101               SUBROUTINE LISTSC(ICPMAX,CMPNTS,AINPUT,OUTPUT)                         000002
00101         C  VERSION 2.                       REVISED  OCT 8 1976                      000002
00101         C  PURPOSE   PROVIDE A LIST OF STANDARD COMPONENTS AND THEIR                 000002
00101         C                  INPUTS, OUTPUTS, AND TABLES                               000002
00101         C  CALL SEQUENCE   ICPMAX - NO. OF STANDARD COMPONENTS                       000002
00101         C                  CMPNTS - LIST OF STANDARD COMPONENT NAMES                 000002
00101         C                  AINPUT - WORK SPACE FOR INPUT NAMES                       000002
00101         C                  OUTPUT - WORK SPACE FOR OUTPUT NAMES                      000002
00103                             IMPLICIT DOUBLE PRECISION (A-Z)                          000002
00104                             IMPLICIT INTEGER (I,J,K,L,M,N)                           000002
00105                             DOUBLE PRECISION IP,IV                                   000002
00106               COMMON/C10/IREAD,IWRITE,IDIAG                                          000002
00107               DIMENSION CMPNTS(1),AINPUT(1),OUTPUT(1),TABLE(10)                      000002
00113               DATA IBLNK/'      '/                                                   000002
00113         C     LITERAL 'POOL' TO SATISFY DBLE PRECSN ASSGNMNT STMNTS                  000002
00112               DATA BLNK/'            '/                                              000002
00114               DATA HONE/'1           '/,HTWO/'2                                      000002
00117               CALL READMS(10,PFNAME,1,12HPFNAME                                      000002
00120                WRITE(IWRITE,101)PFNAME                                                  000010
00123          101   FORMAT(1H1,14X,"LIST OF STANDARD  ",A10," COMPONENTS")                   000022
00123          C --->       SCAN STD. COMPONENTS                                              000022
00124                DO 560 I=1,ICPMAX                                                        000022
00127                WRITE(6,521)I,CMPNTS(I)                                                  000022
00133          521   FORMAT(///15X,"COMPONENT NO.",I3,"   NAME = ",A2//                       000031
00133               13X,"INPUTS",8X,"OUTPUTS",16X,"TABLES"/                                   000031
00133               22(" NAME  PORT     "),"  NAME   INDP. VAR.  MAX. DATA")                  000031
00133          C --->       GET INPUT,OUTPUT,AND TABLE NAMES                                  000031
00134                CALL  CONDAT(CMPNTS(I),L2HINPT          ,NI,AINPUT)                      000031
00135                CALL  CONDAT(CMPNTS(I),L2HOUTP          ,NO,OUTPUT)                      000041
00136                CALL  CONDAT(CMPNTS(I),L2HTABS          ,NT,TABLE)                       000051
00137                MAX=MAX0(NI,NO,NT)                                                       000061
00137          C --->       SCAN LONGEST LIST OF NAMES                                        000061
00140                DO 560 J=1,MAX                                                           000073
00140          C --->       BLANK NAMES                                                       000073
00143                AIN=BLNK                                                                 000102
00144                OUT=BLNK                                                                 000104
00145                TAB=BLNK                                                                 000105
00146                ID=BLNK                                                                  000106
00147                IP=BLNK                                                                  000110
00150                OP=BLNK                                                                  000111
00151                IV=BLNK                                                                  000112
00152                ST=BLNK                                                                  000113
00153                IF(J.GT.NI)GO TO 530                                                     000114
00155                AIN=AINPUT(J)                                                            000120
00156                CALL GETT(AIN,9,IP)                                        01010         000122
00157          530   IF(J.GT.NO)GO TO 535                                                     000130
00161                OUT=OUTPUT(J)                                                            000133
00162                CALL GETT(OUT,9,OP)                                        01040         000135
00163                CALL GETT(OUT,10,ST)                                       01050         000147
00164          535   IF(J.GT.NT)GO TO 540                                                     000150
00166                TAG=TABLE(J)                                                             000153
00166          C --->       GET TABLE DIMENSION                                               000153
00167                CALL GETCOD(5,TAG,ID)                                                    000155
00170                IV=HIVO                                                                  000162
00171                IF(ID.GT.0)GO TO 540                                                     000164
00173                IV=NONE                                                                  000167
00174                ID=TABS(ID)                                                              000171
00175          540   WRITE(IWRITE,541)AIN,IP,OUT,OP,ST,TAG,IV,ID                              000174
00207          541   FORMAT(2X,A6,A1,8X,A6,A1,1X,A1,7X,A6,5X,A1,9X,I3)                        000220
00210          560   CONTINUE                                                                 000220
00213                WRITE(IWRITE,563)                                                        000220
00215          563   FORMAT(1H1)                                                              000225
00216                RETURN                                                                   000225
00217                END $ LISTSC  ******************************                             000264
```

```
SUBROUTINE NAMARY     ENTRY POINT 000237


STORAGE USED  CODE(1) 000256; DATA(0) 000100; BLANK COMMON(2) 000000


EXTERNAL REFERENCES (BLOCK, NAME)

 0003    LINE
 0004    NNCOD$
 0005    NREW$
 0006    NI02$
 0007    NRDU$
 0010    NWDU$
 0011    NI03$
 0012    NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000    000040 105F       0001    000115 111G       0000    000044 121F       0001    000022 121G       0000    000050 125F
 0001    000060 134G       0001    000163 162G       0000    000054 201F       0000 D 00002C ANAME      0000 D 000024 BLNK
 0000 D 000026 HDATA      0000 I 000037 I          0000    000045 INJP$      0000 I 000036 ISOUR      0000 I 000034 ISTART
 0000 I 000035 ISTOP      0000 I 000031 J          0000 I 000032 K          0000 I 000030 NEXT       0000 I 000033 N10
 0000 D 000000 SOURCE



00100        1*      CNAMARY                                                                             000000
00101        2*             SUBROUTINE NAMARY(CNAME,NCHAR,N,NUNIT)                                       000000
00101        3*      C  VERSION 1.2                         REVISED  AUG 22 1975                         000000
00101        4*      C  PURPOSE  FORM A DATA STATEMENT THAT CONTAINS A GIVEN LIST OF NAMES               000000
00101        5*      C  CALL SEQUENCE    CNAME  - NAME OF THE ARRAY TO BE INITIALIZED                     000000
00101        6*      C                   NCHAR  - NO. OF CHARACTERS IN ARRAY NAME                         000000
00101        7*      C                   N      - NO. OF NAMES TO BE PLACED IN DATA STATEMENT             000000
00101        8*      C                   NUNIT  - UNIT CONTAINING LIST OF NAMES                           000000
00101        9*      C  DESIGNED BY  J.D. BURROUGHS                  MAY 1974                              000000
00103       10*                        IMPLICIT DOUBLE PRECISION (A-Z)                                     000000
00104       11*                        IMPLICIT INTEGER (I,J,K,L,M,N)                                      000000
00105       12*             DIMENSION SOURCE(16)                                                           000000
00106       13*             DIMENSION ANAME(2)                                                             000000
00107       14*             DATA ANAME /24H                                                                000000
00111       15*             DATA BLNK/12H            ,HDATA/12H      DATA                                  000000
00111       16*      C --->       TEST FOR EMPTY SET                                                       000000
00114       17*             IF(N.LE.0) RETURN                                                              000002
00116       18*             REWIND NUNIT                                                                   000005
00116       19*      C ---        CALCULATE THE NO. OF DATA STATEMENT EXTENSIONS REQD.                     000005
00117       20*             NEXT=(N-1)/108+1                                                               000010
00117       21*      C ---        SCAN DATA STATEMENT EXTENSIONS                                           000010
00120       22*             DO 400 J=1,NEXT                                                                000016
00123       23*      C ---        EXTENSION COUNTER                                                        000016
00123       24*             K=J-1                                                                          000022
```

```
00123      25*     C ---        NO. OF CHARACTERS PER EXTENSION                          000022
00124      26*           N10=12*(N-K*108)                                                000025
00124      27*     C ---        LIMIT NO. OF CHARACTERS PER DATA STATEMENT TO 1296       000025
00125      28*           IF(N10.GT.1296)N10=1296                                         000031
00125      29*     C ---        CALC. FIRST AND LAST WORD IN LIST OF DATA STATEMENT      000031
00127      30*           ISTART=K*108+1                                                  000037
00130      31*           ISTOP=ISTART+N10/12-1                                           000043
00130      32*     C --->       GENERATE DATA STATEMENT                                  000043
00131      33*           SOURCE(1)=HDATA                                                 000051
00132      34*           ISOUR=12                                                        000051
00133      35*           DO 100 I=2,8                                                    000060
00136      36*     100   SOURCE(I)=BLNK                                                  000060
00136      37*     C --->       LOAD ARRAY NAME                                          000060
00140      38*           CALL LINE(0,SOURCE,ISOUR,CNAME,NCHAR,9)                         000062
00140      39*     C ---        TEST IF DATA STATEMENT EXTENSION IS REQUIRED             000062
00141      40*           IF(K.LE.0)GO TO 110                                             000072
00141      41*     C ---        ENCODE DATA EXTENSION NO.                                000072
00143      42*           ENCODE(2,105,NEK)                                               000075
00146      43*     105   FORMAT(I2)                                                      000104
00146      44*     C ---        ADD EXTENSION NO. TO DUMMY ARRAY NAME                    000104
00147      45*           CALL LINE(0,SOURCE,ISOUR,N,2,9)                                 000104
00150      46*     110   CALL LINE(0,SOURCE,ISOUR,12H/            ,1,9)                  000115
00151      47*           ENCODE(4,121,N10)N10                                            000126
00154      48*     121   FORMAT(I4)                                                      000136
00154      49*     C --->       LOAD NO. OF CHARACTERS IN DATA STATEMENT                 000136
00155      50*           CALL LINE(0,SOURCE,ISOUR,N10,4,9)                               000136
00156      51*           CALL LINE(0,SOURCE,ISOUR,12H)            ,1,9)                  000146
00156      52*     C --->       SCAN NAMES                                               000146
00157      53*           ANAME(1) = BLNK                                                 000156
00160      54*           ANAME(2) = BLNK                                                 000160
00161      55*           DO 200 I=ISTART,ISTOP                                           000163
00164      56*           READ(NUNIT,125)ANAME(1)                                         000163
00167      57*     125   FORMAT(A6)                                                      000170
00167      58*     C --->       LOAD NAMES INTO DATA STATEMENT                           000170
00170      59*           CALL LINE(1,SOURCE,ISOUR,ANAME,12,9)                            000175
00171      60*     200   CONTINUE                                                        000201
00175      61*           CALL LINE(1,SOURCE,ISOUR,12H/            ,1,9)                  000201
00174      62*           WRITE(9,201)SOURCE                                              000211
00177      63*     201   FORMAT(8A10)                                                    000223
00200      64*     400   CONTINUE                                                        000223
00202      65*           RETURN                                                          000223
00203      66*           END a NAMARY  *************************                         000255
```

```
SUBROUTINE NAMGEN     ENTRY POINT 000060

STORAGE USED  CODE(1) 000073; DATA(0) 000014; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   STRMOV
 0004   KOMSTR
 0005   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000 D 000000 BLNK      0000 I 000002 I         0000   000010 INJPS     0004 I 000000 KOMSTR


00100      1*     CNAMGEN                                                                                  000000
00101      2*           SUBROUTINE NAMGEN(SOURNM,COMNAM,QUANAM)                                            000000
00101      3*     C  PURPOSE   GENERATE UNIQUE NAMES FOR ALL MODEL VARIABLES PARAMETERS                    000000
00101      4*     C  CALL SEQUENCE    SOURNM - SOURCE NAME                                                 000000
00101      5*     C                   COMNAM - COMPONENT NAME                                              000000
00101      6*     C                   QUANAM - QUANTITY NAME                                               000000
00103      7*                          IMPLICIT DOUBLE PRECISION (A-Z)                                     000000
00104      8*                          IMPLICIT INTEGER (I,J,K,L,M,N)                                      000000
00105      9*           DATA PLNK/12H           /                                                          000000
00105     10*     C --->      TRANSFER SOURCE NAME TO QUANTITY NAME                                        000000
00107     11*           QUANAM=SOURNM                                                                      000000
00107     12*     C --->      ADD COMP. NAME TO COL. 4 TO 6                                                000000
00110     13*           CALL STRMOV(COMNAM,1,3,QUANAM,4)                                                   000001
00110     14*     C ---       TEST COL. 9 FOR PORT NUMBER                                                  000001
00111     15*           IF(KOMSTR(QUANAM,9,1,PLNK,1).EQ.0)RETURN                                          000010
00111     16*     C --->      TEST IF COL. 2 OR COL. 3 IS TO BE USED FOR PORT NO.                          000016
00113     17*           I=3                                                                                000024
00114     18*           IF(KOMSTR(QUANAM,2,1,PLNK,1).EQ.0)I=2                                              000026
00114     19*     C --->      PLACE PORT NO. IN COL. I                                                     000026
00116     20*           CALL STRMOV(QUANAM,9,1,QUANAM,I)                                                   000041
00117     21*           RETURN                                                                             000050
00120     22*           END & NAMGEN  ************************                                             000072
```

```
SUBROUTINE NEWCOM    ENTRY POINT 000266

STORAGE USED  CODE(1) 000344; DATA(0) 000052; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CIO     000003
 0004    CSEQ    000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0005    FCPDUB
 0006    GETCOD
 0007    PUTCOD
 0010    KOMSTR
 0011    COMDAT
 0012    RLAUN$
 0013    NWDU$
 0014    NIO2$
 0015    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000050 123G      0001    000067 200L      0001    000130 220L      0001    000143 300L      0000    000012 301F
 0001    000167 400L      0001    000222 420L      0000 D 000000 MINPT     0000 D 000002 MOUTP     0000 D 000004 MZZZ
 0000 I 000010 I          0003    000002 IFLAG     0000    000035 INJP$     0003    000000 IREAD     0000 I 000007 ISYMB
 0003 I 000001 IWRITE     0010 I 000000 KOMSTR     0000 I 000011 LN        0000 I 000006 LOCNO     0004 I 000000 NSEQ
 0004 D 000001 SEQA


00100      1*      CNEWCOM                                                                              000011
00101      2*            SUBROUTINE NEWCOM(COMNAM,CMPNTS,ICOMP,ALOC,CMPMOD,NOCOMP,                      000011
00101      3*           1AINPUT,NINPUT,AOUT,NOUT,IDCOMP)                                                 000011
00101      4*      C  VERSION 2.                          REVISED  JAN 12 1976                          000011
00101      5*      C  PURPOSE   INTRODUCE NEW COMPONENT INTO ECS MODEL                                  000011
00101      6*      C  CALL SEQUENCE   COMNAM - COMPONENT NAME                                           000011
00101      7*      C                  CMPNTS - LIST OF STD. COMP. NAMES                                 000011
00101      8*      C                  ICOMP  - LOCATION OF STD. COMP. NAME IN LIST                      000011
00101      9*      C                  ALOC   - COMPONENT LOCATION NO.                                   000011
00101     10*      C                  CMPMOD - LIST OF COMP. IN CURRENT MODEL                           000011
00101     11*      C                  NOCOMP - NO. OF COMP. IN CURRENT MODEL                            000011
00101     12*      C                  AINPUT - STD. NAMES OF INPUTS FOR COMP.                           000011
00101     13*      C                  NINPUT - NO. OF INPUTS TO COMP.                                   000011
00101     14*      C                  AOUT   - STD. NAMES OF OUTPUTS FOR COMP.                          000011
00101     15*      C                  NOUT   - NO. OF OUTPUTS FOR COMP.                                 000011
00101     16*      C                  IDCOMP - COMP. NO. IN CURRENT MODEL                               000011
00101     17*      C    DESIGNED BY  J.D.BURROUGHS                        DATE  JULY 1974               000011
          18*                     IMPLICIT DOUBLE PRECISION (A-Z)                                       000011
```

```
00104              IMPLICIT INTEGER (I,J,K,L,M,N)                          000011
00105        COMMON /CIO/IREAD,IWRITE,IDIAG/CSEQ/NSEQ,SEQA(1)              000011
00106        DIMENSION CMPNTS(1),CMPMOD(1),AINPUT(1),AOUT(1)               000011
00106  C     LITERAL "POOL" TO SATISFY ODLF PRECSN                         000011
00107        DATA HINPT/12HINPT        /,HOUTP/12HOUTP                     000011
00112        DATA HZZZ/12HZZZ                                              000011
00112  C --->      CONVERT LOCATION NO. FROM HOLLORITH TO INTEGER          000011
00114        CALL PCODUB(ALOC,ALOC)                                        000011
00115        LOCNO=ALOC                                                    000015
00115  C --->      GET SYMBOL NO. FOR COMPONENT AND PUT IN LOCATION 4      000015
00116        CALL GETCOD(5,CMPNTS(ICOMP),ISYMB)                            000023
00117        CALL PUTCOD(4,COMNAM,ISYMB)                                   000032
00117  C --->      TEST THAT 1 OR MORE COMP. EXIST IN MODEL                000032
00120        IF(NOCOMP.LT.1)GO TO 200                                      000037
00120  C --->      SCAN EXISTING COMPS. IN MODEL                           000037
00122        DO 100 I=1,NOCOMP                                             000042
00122  C --->      TEST THAT NEW COMP. NAME IS UNIQUE                      000042
00125        IF(KOMSTR(CMPMOD(I),1,3,COMNAM,1).EQ.0)GO TO 300              000056
00127  100   CONTINUE                                                      000067
00127  C --->      NEW NAME IS UNIQUE                                      000067
00127  C --->      GET STD. INPUT LIST FOR COMP.                           000067
00131  200   CALL CONCAT(COMNAM,HINPT,NINPUT,AINPUT)                       000067
00131  C --->      ADD LOC. NO. AND NO. OF INPUTS TO COMP. NAME            000067
00132        CALL PUTCOD(3,COMNAM,LOCNO)                                   000074
00133        CALL PUTCOD(5,COMNAM,NINPUT)                                  000101
00133  C --->      ADVANCE COMP. COUNT                                     000101
00134        NOCOMP=NOCOMP+1                                               000106
00134  C --->      ADD NEW NAME TO MODEL COMP. NAME LIST                   000106
00135        CMPMOD(NOCOMP)=COMNAM                                         000111
00135  C --->      ADD COMP. NO. TO COMPONENT SEQUENCE LIST                000111
00136        NSEQ=NSEQ+1                                                   000115
00137        CALL PUTCOD(NSEQ,SEQA,NOCOMP)                                 000120
00140        IDCOMP=NOCOMP                                                 000125
00140  C --->      GET LIST OF STD. OUTPUTS                                000125
00141  220   CALL CONCAT(CMPNTS(ICOMP),HOUTP,NOUT,AOUT)                    000130
00142        RETURN                                                        000137
00142  C --->      TEST LOCATION NO. FOR COMP. THAT HAVE RECEIVED INPUTS BUT HA    000137
00142  C           BEEN DEFINED.                                           000137
00143  300   CALL GETCOD(3,CMPMOD(I),LN)                                   000143
00144        IF(LN.LE.0)GO TO 400                                          000154
00146        WRITE(IWRITE,301)COMNAM                                       000157
00151  301   FORMAT(/5X,29H * * WARNING *** COMPONENT  ,A4,'  HAS ALREADY BEEN    000165
00151       1 DEFINED')                                                    000165
00152        GO TO 420                                                     000165
00152  C --->      ADD LOCATION NO. TO COMP. NAME                          000165
00153  400   CALL PUTCOD(3,CMPMOD(I),LOCNO)                                000167
00153  C ---       ADD SYMBOL NUMBER TO COMPONENT NAME                     000167
00154        CALL PUTCOD(4,CMPMOD(I),ISYMB)                                000201
00154  C --->      ADD COMP. NO. TO COMPONENT SEQUENCE LIST                000201
00155        NSEQ=NSEQ+1                                                   000211
00156        CALL PUTCOD(NSEQ,SEQA,I)                                      000214
00157  420   COMNAM=CMPMOD(I)                                              000222
00157  C --->      GET NO. OF INPUTS                                       000222
00160        CALL GETCOD(5,COMNAM,NINPUT)                                  000226
00160  C --->      GET INPUT LIST FROM FILE 7                              000226
00161        AINPUT(1)=HZZZ                                                000233
00162        IF(NINPUT.GT.0)CALL READMS(7,AINPUT,NINPUT,1)                 000235
```

```
00164      76*        INCOMP=I                                                      000246
00165      77*        GO TO 220                                                     000250
00166      78*        END @ NEWCOM  ***************************                     000343
```

```
SUBROUTINE ORDER      ENTRY POINT 000203

STORAGE USED  CODE(1) 000234; DATA(0) 000025; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003    IJ*IT
 0004    NEFR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000021 117G       0001    000031 117G       0001    000114 147G       0001    000024 15L        0001    000043 30L
 0001    000056 40L        0001    000105 60L        0001    000123 66L        0001    000132 70L        0000 I 000002 I
 0000 I 000000 IJAIT       0000    000006 INJP$      0000 I 000003 JS          0000 I 000004 N           0000 I 000000 NCO
 0000 I 000001 NTW2


00100       1*       CORDER                                                                                     000010
00101       2*             SUBROUTINE ORDER(NV,ICO,A,IW1,IW2,IERROR,IB,IE)                                      000010
00101       3*       C   VERSION 1.                      REVISED  AUG 4 1975                                   000010
00101       4*       C   PURPOSE  GENERATE A SEQUENCE VECTOR THAT REORDERS VARIABLES                            000010
00101       5*       C             SO THAT CONNECTION MATRIX IS LOWER TRIANGULAR.                               000010
00101       6*       C   CALL SEQUENCE    NV      - SYSTEM ORDER                                                000010
00101       7*       C                    ICO     - SEQUENCE VECTOR                                             000010
00101       8*       C                    A       - SYSTEM CONNECTION MATRIX                                    000010
00101       9*       C                    IW1     - NTH ORDER VECTOR - PROCESS CODE                             000010
00101      10*       C                    IW2     - NTH ORDER VECTOR - PROCESS SEQUENCE                         000010
00101      11*       C                    IERROR - ERROR FLAG  0 = SYSTEM WAS REDUCED TO LOWER                  000010
00101      12*       C                                                TRIANGULAR FORM.                          000010
00101      13*       C                                              1 = SYSTEM CAN NOT BE REDUCED T             000010
00101      14*       C                                                 TRIANGULAR FORM                          000010
00101      15*       C                    IB      - FIRST WORD IN IW2 POINTING TO LOOP COMP.                    000010
00101      16*       C                    IE      - LAST WORD IN IW2 POINTING TO LOOP COMP.                     000010
00101      17*       C   DESIGNED BY   F FATH                               JULY 1975                           000010
00103      18*                         IMPLICIT DOUBLE PRECISION (A-Z)                                          000010
00104      19*                         IMPLICIT INTEGER (I,J,K,L,M,N)                                           000010
00105      20*             DIMENSION ICO(1),IW1(1),IW2(1),A(1)                                                  000010
00106      21*             NCO=0                                                                                000010
00107      22*             IERROR=0                                                                             000011
00107      23*       C     SET ELEMENT COUNT IN PROCESS SEQUENCE VECTOR TO ZERO                                 000011
00110      24*             NTW2=0                                                                               000012
00110      25*       C     INITIALIZE PROCESS CODE FOR EACH ELEMENT TO -1 (NO PROCESS)                          000012
00111      26*             DO 10 I=1,NV                                                                         000021
00114      27*        10   IW1(I)=-1                                                                            000021
00114      28*       C     FIND FIRST NON-PROCESSED ELEMENT                                                     000021
00116      29*        15   DO 20 I=1,NV                                                                         000024
00121      30*             IF(IW1(I).LT.0)GO TO 30                                                              000031
00123      31*        20   CONTINUE                                                                             000037
```

```
00125     32*     C     IF ALL ELEMENTS PROCESSED, RETURN                                      000037
00125     33*           RETURN                                                                 000037
00125     34*     C     PUT NON-PROCESSED ELEMENT INTO PROCESS SEQUENCE VECTOR AT BOTTOM       000037
00126     35*     30    NIW2=NIW2+1                                                            000043
00127     36*           IW2(NIW2)=I                                                            000045
00127     37*     C     SET PROCESS CODE TO 0 (PARTIAL PROCESS)                                000045
00130     38*           IW1(I)=0                                                               000051
00130     39*     C     CHECK FOR DEPENDANCE ON OTHER ELEMENTS                                 000051
00131     40*           JS=0                                                                   000054
00132     41*     40    JS=JS+1                                                                000056
00132     42*     C     IF ALL ELEMENT DEPENDANCIES CHECKED, PROCESS IS COMPLETE               000056
00133     43*           IF(JS.GT.NV)GO TO 70                                                   000060
00135     44*           K=IJOTI(A,I,JS,NV)                                                     000063
00135     45*     C     IF NO DEPENDANCE (K=0) KEEP LOOKING                                    000063
00136     46*           IF(K.EQ.0)GO TO 40                                                     000072
00136     47*     C     IF DEPENDANT ON ELEMENT ALREADY PROCESSED (CODE=1) KEEP LOOKING        000072
00136     48*     C     IF DEPENDANT ON ELEMENT NOT PROCESSED (CODE=-1) START PROCESSING       000072
00136     49*     C     ON THAT ELEMENT.                                                       000072
00136     50*     C     IF DEPENDANT ON ELEMENT PARTIALLY PROCESSED (CODE=0) SEQUENCING        000072
00136     51*     C     IS IMPOSSIBLE.  SET ERROR FLAG AND START ERROR REPORT.                 000072
00140     52*           IF(IW1(JS))50,60,40                                                    000074
00143     53*     50    I=JS                                                                   000101
00144     54*           GO TO 30                                                               000103
00145     55*     60    IERROR=1                                                               000105
00145     56*     C     LOOK FOR JS IN IW2.  THIS IS BEGINING OF DEPENDANT LOOP                000105
00146     57*           DO 65 K=1,NIW2                                                         000106
00151     58*           IF(IW2(K).EQ.JS)GO TO 66                                               000114
00153     59*     65    CONTINUE                                                               000123
00155     60*     66    IB=K                                                                   000123
00155     61*     C     SET END OF LOOP POINTER                                                000123
00156     62*           IE=NIW2                                                                000124
00156     63*     C     RETURN DUE TO ERROR                                                    000124
00157     64*           RETURN                                                                 000126
00157     65*     C     PROCESS FOR ELEMENT COMPLETE - UPDATE PROCESSED ELEMENT COUNT          000126
00160     66*     70    NCO=NCO+1                                                              000132
00160     67*     C     SET SEQUENCE VECTOR POSITION TO INDICATE ELEMENT                       000132
00161     68*           ICO(NCO)=I                                                             000134
00161     69*     C     SET PROCESS CODE FOR ELEMENT TO COMPLETE (CODE=1)                      000134
00162     70*           IW1(I)=1                                                               000140
00162     71*     C     DECREMENT PROCESS SEQUENCE POINTER                                     000140
00163     72*           NIW2=NIW2-1                                                            000144
00163     73*     C     IF ALL PROCESSED - RETURN                                              000144
00164     74*           IF(NCO.EQ.NV)RETURN                                                    000147
00164     75*     C     IF NO ELEMENT LEFT IN PROCESS SEQUENCE VECTOR, GO LOOK FOR FIRST       000147
00164     76*     C     NON-PROCESSED ELEMENT.                                                 000147
00166     77*           IF(NIW2.LT.1)GO TO 15                                                  000155
00166     78*     C     CONTINUE PROCESSING BOTTOM ELEMENT IN PROCESS SEQUENCE VECTOR          000155
00166     79*     C     WHERE IT WAS INTERRUPTED.                                              000155
00170     80*           JS=I                                                                   000160
00171     81*           I=IW2(NIW2)                                                            000162
00172     82*           GO TO 40                                                               000166
00173     83*           END @ ORDER  **************************                                000233
```

```
SUBROUTINE PORTCN    ENTRY POINT 000213


STORAGE USED  CODE(1) 000240; DATA(0) 000031; BLANK COMMON(2) 000000


EXTERNAL REFERENCES (BLOCK, NAME)

 0003   KOMSTR
 0004   NAMCON
 0005   STRMOV
 0006   NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000062 100L      0001   000015 112G      0001   000134 120L      0001   000065 125G      0001   000140 140L
 0001    000167 200L      0000 D 000000 BLNK      0000 I 000002 I         0000   000011 INJP$     0000 I 000003 J
 0003 I 000000 KOMSTR



00100      1*      CPORTCN                                                                             000015
00101      2*            SUBROUTINE PORTCN(AINPUT,NINPUT,OUTPUT,NOUT,IPORT,OPORT,OUTNAM,                000015
00101      3*           1 NOCON,STREAM)                                                                000015
00101      4*      C  PURPOSE   CONNECT ALL MATCHING PHYSICAL QUANTITIES AT SPECIFIED                      000015
00101      5*      C            PORTS ON TWO COMPONENTS.                                                   000015
00101      6*      C  CALL SEQUENCE   AINPUT - INPUT QUANTITY NAME LIST                                    000015
00101      7*      C                  NINPUT - NO. OF INPUT QUANTITIES                                     000015
00101      8*      C                  OUTPUT - OUTPUT QUANTITY NAME LIST                                   000015
00101      9*      C                  NOUT   - NO. OF OUTPUT QUANTITIES                                    000015
00101     10*      C                  IPORT  - INPUT PORT NO.                                              000015
00101     11*      C                  OPORT  - OUTPUT PORT NO.                                             000015
00101     12*      C                  OUTNAM - OUTPUT COMP. NAME                                           000015
00101     13*      C                  NOCON  - NO CONNECTION FLAG                                          000015
00101     14*      C                  STREAM - SOURCE INDICATOR. BLANK = UPSTREAM SOURCE                   000015
00101     15*      C                                 D = DOWNSTREAM SOURCE                                 000015
00103     16*                          IMPLICIT DOUBLE PRECISION (A-Z)                                     000015
00104     17*                          IMPLICIT INTEGER (I,J,K,L,M,N)                                      000015
00105     18*                          DOUBLE PRECISION IPORT                                              000015
00106     19*            DIMENSION AINPUT(1),OUTPUT(1)                                                     000015
00107     20*            DATA BLNK/12H                                                                     000015
00107     21*      C --->       SCAN INPUT LIST                                                            000015
00111     22*            DO 200 I=1,NINPUT                                                                 000015
00111     23*      C --->       TEST IF INPUT IS SATISFIED                                                 000015
00114     24*            IF(KOMSTR(AINPUT(I),9,1,BLNK,1).NE.0)GO TO 200                                    000015
00114     25*      C --->       BYPASS PORT TEST IF INPUT IS UNIVERSAL PORT                                000015
00116     26*            IF(KOMSTR(AINPUT(I),9,1,BLNK,1).EQ.0)GO TO 100                                    000030
00116     27*      C --->       BYPASS TEST IF SPECIFIED PORT IS UNIVERSAL PORT                            000030
00120     28*            IF(IPORT.EQ.BLNK)GO TO 100                                                        000043
00120     29*      C --->       COMPARE PORTS                                                              000043
00122     30*            IF(KOMSTR(AINPUT(I),9,1,IPORT,1).NE.0)GO TO 200                                   000046
```

```
00122      C --->      SCAN OUTPUTS                                                        000046
00124      100   DO 120 J=1,NOUT                                                           000065
00124      C --->      TEST FOR PHYSICAL QUANTITY MATCH                                    000065
00127            IF(KONSTR(AINPUT(I),1,3,OUTPUT(J),1).NE.0)GO TO 120                       000065
00127      C --->      BYPASS PORT TEST IF SPECIFIED PORT IS UNIVERSAL PORT                000065
00131            IF(OPORT.EQ.BLNK)GO TO 140                                                000102
00131      C --->      BYPASS PORT TEST IF OUTPUT IS UNIVERSAL PORT                        000102
00133            IF(KONSTR(OUTPUT(J),9,1,BLNK,1).EQ.0)GO TO 140                            000105
00133      C --->      TEST FOR PORT MATCH                                                 000105
00135            IF(KONSTR(OUTPUT(J),9,1,OPORT,1).EQ.0)GO TO 140                           000120
00137      120   CONTINUE                                                                  000136
00141            GO TO 200                                                                 000136
00141      C --->      SATISFY INPUT                                                       000136
00142      140   CALL NAMGEN(OUTPUT(J),OUTNAM,AINPUT(I))                                   000140
00142      C --->      PLACE SOURCE INDICATOR IN NAME                                      000140
00143            CALL STRMOV(STREAM,1,1,AINPUT(I),8)                                       000153
00144            NOCON=1                                                                   000164
00145      200   CONTINUE                                                                  000172
00147            RETURN                                                                    000172
00153            END A PORTCN  ***************************                                 000237
```

```
SUBROUTINE SCHEMA    ENTRY POINT 000512


STORAGE USED  CODE(1) 000536; DATA(0) 003026; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CIO     000003
 0004    CTITLE  000016


EXTERNAL REFERENCES (BLOCK, NAME)

 0005    NMCOD$
 0006    STRMOV
 0007    NOMSTR
 0010    FLTCOD
 0011    SYMBOL
 0012    RCADM$
 0013    CONACT
 0014    NI02$
 0015    NWDU$
 0016    NI01$
 0017    NI03$
 0020    NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000005 100L       0001     000015 122G       0001     000017 125G       0001     000034 136G       0000     002710 139F
 0001     000076 156G       0001     000063 160L       0001     000220 216G       0001     000337 244G       0000     002711 251F
 0001     000345 252G       0001     000202 260L       0000     002716 261F       0001     000213 300L       0001     000315 420L
 0000     002760 423F       0001     000401 440L       0001     000413 460L       0001     000415 480L       0001     000436 500L
 0001     000441 602L       0000     002767 605F       0000 D 002666 ASTRSK     0000 D 002662 BLNK       0000 D 002702 COMMA
 0000 D 002664 FORLT      0000 I 002674 I          0003 I 000002 IDIAG      0000 D 000000 INCOM      0000     003006 INUFS
 0003     000000 IREAD      0000 I 002733 ISYMB      0003 I 000001 IWRITE     0000 I 002675 J          0007 I 000000 NOMSTR
 0000 I 002676 LOCCOL     0000 I 002673 LOCL       0000 I 002677 LOCNO      0000 I 002702 LOCPAG     0000 I 002672 LPK
 0000 I 002670 MAXPAG     0000 I 002706 MORE       0000 I 002707 NAME       0000 I 002704 NINPUT     0000 I 002705 NOIN
 0000 I 002671 NPAGE      0000 D 003002 PAGE       0004 D 000000 TITLE
```

```
00100        1*      CSCHEMA                                                                           000002
00101        2*            SUBROUTINE SCHEMA(CMPMOD,NOCOMP,INPUTS,NAMES)                               000002
00101        3*      C  VERSION 2.                        REVISED  SEPT 10 1975                        000002
00101        4*      C  PURPOSE   PRODUCE A SCHEMATIC DIAGRAM ON THE LINEPRINTER                       000002
00101        5*      C            OF THE ECS MODEL                                                     000002
00101        6*      C  CALL SEQUENCE    CMPMOD - LIST OF COMPONENTS IN MODEL                          000002
00101        7*      C                   NOCOMP - NO. OF COMP. IN MODEL                                000002
00101        8*      C                   INPUTS - WORK ARRAY FOR INPUT NAMES                           000002
00101        9*      C                   NAMES  - WORK ARRAY FOR LABEL NAMES                           000002
00101       10*      C  DESIGNED BY  J.D. BURROUGHS                    JUNE 1974                       000002
```

```
00103                       IMPLICIT DOUBLE PRECISION (A-Z)                  000002
00104                       IMPLICIT INTEGER (I,J,K,L,M,N)                   000002
00105                       DOUBLE PRECISION INPUTS,NAMES,INCOM              000002
00106               COMMON /CIO/IREAD,IWRITE,IDIAG/CTITLE/TITLE(7)           000002
00107               DIMENSION PAGE(13,56),CMPMOD(1),INPUTS(1),NAMES(1)       000002
00110               DATA BLNK/12H            /,FORLT/12HFORT                 000002
00113               DATA ASTRSK/12H********     /                            000002
00115               MAXPAG=0                                                 000002
00116               NPAGE=0                                                  000003
00116       C --->       BLANK PAGE AND LOAD LOCATION NUMBERS                000003
00117       100     LOK=NPAGE                                                000005
00117       C --->       LOCATION NO. LINE COUNTER                           000005
00120               LOCL=4                                                   000006
00120       C --->       SCAN ALL LINES ON PAGE                              000006
00121               DO 160 I=1,56                                            000017
00121       C --->       BLANK ENTIRE LINE                                   000017
00124               DO 120 J=1,13                                            000017
00127       120     PAGE(J,I)=BLNK                                           000017
00127       C --->       TEST IF LINE CONTAINS LOCATION NUMBERS.             000017
00131               IF(I.LT.LOCL)GO TO 160                                   000021
00131       C --->       INCREMENT LOCATION NO. LINE COUNTER                 000021
00133               LOCL=LOCL+7                                              000025
00134               LOCCOL=-8                                                000030
00134       C --->       SCAN COLS. AND LOAD LOCATION NOS.                   000030
00135               DO 140 J=1,10                                            000034
00135       C --->       INCREMENT LOCATION NO.                              000034
00140               LOK=LOK+1                                                000034
00141               LOCCOL=LOCCOL+13                                         000036
00142               ENCODE(4,139,LOCNO)LOK                                   000041
00145       139     FORMAT(I4)                                               000050
00146               CALL STRMOV(LOCNO,1,4,PAGE(1,I),LOCCOL)                  000052
00147       140     CONTINUE                                                 000065
00151       160     CONTINUE                                                 000065
00151       C --->       PLACE COMPONENT SYMBOLS ON PAGE                     000065
00151       C --->       TEST THAT MORE THAN 0 COMP. EXIST IN MODEL          000065
00153               IF(NOCOMP.LE.0)GO TO 602                                 000065
00153       C ---        SCAN COMPS. IN MODEL                                000065
00155               DO 300 I=1,NOCOMP                                        000070
00160               COMNAM=CMPMOD(I)                                         000076
00160       C ---        SKIP FORTRAN COMPONENTS                             000076
00161               IF(KOMSTR(COMNAM,1,4,FORLT,1).EQ.0)GO TO 300             000077
00161       C ---        GET LOCATION NO. FROM COMP. NAME                    000077
00163               CALL GETCOD(3,COMNAM,LOK)                                000110
00163       C --->       DETERMINE PAGE NO.                                  000110
00164               LOCPAG=(LOK/100)*100                                     000115
00164       C --->       DETERMINE MAX. NO. OF PAGES REQ'D.                  000115
00165               MAXPAG=MAX0(MAXPAG,LOCPAG)                               000122
00165       C --->       TEST IF COMPONENT IS ON CURRENT PAGE                000122
00166               IF(LOCPAG.NE.NPAGE)GO TO 300                             000127
00166       C --->       CONVERT GENERAL PAGE LOCATION TO LOCAL PAGE LOCATION 000127
00170               LOCPAG=LOK-LOCPAG                                        000132
00170       C --->       TEST TO ASSURE LOC NO. IS ON PAGE                   000132
00171               IF(LOCPAG.LT.1.OR.LOCPAG.GT.80)GO TO 260                 000135
00171       C --->       ADD SYMBOL TO CURRENT PAGE FOR COMPONENT             000135
00173               CALL GETCOD(4,COMNAM,ISYMB)                              000152
00[illegible]       IF(IDIAG.EQ.22)WRITE(IWRITE,251)COMNAM,COMNAM,ISYMB      000157
00[illegible] 251   FORMAT(' SCHEMA ',A10,1X,O20,I10)                        000172
```

```
00203                    CALL SYMBOL(PAGE,COMNAM,ISYMB,LOCPAG)                                      000172
00203          C --->          FORM TABLE OF COMPONENT NAMES (ON ONLY FIRST PASS)                   000172
00204                    GO TO 300                                                                  000200
00205          260       WRITE(IWRITE,261)LOK,COMNAM                                                000202
00211          261       FORMAT(/5X,31H *** WARNING ***  LOCATION NO. ,I4,                          000210
00211                  1 ' FOR COMPONENT  ',A9,'  HAS LAST TWO DIGITS OUTSIDE THE ALLOWABL          000210
00211                  2E RANGE OF 1 TO 80.'/10X,                                                   000210
00211                  3'NO SYMBOL WILL BE PLACED IN SCHEMATIC FOR THIS COMPONENT.')                000210
00212                    LOK=-100                                                                   000210
00213          300       CONTINUE                                                                   000220
00213          C --->          ADD CONNECTING LINES AND NAMES TO SCHEMATIC                          000220
00213          C --->          SCAN MODEL COMPONENTS                                                000220
00215          400       DO 500 I=1,NOCOMP                                                          000226
00215          C --->          BYPASS DIRECT FORTRAN INPUT COMPONENTS                               000226
00220                    IF(KOMSTR(CMPMOD(I),1,4,FORLT,1).EQ.0)GO TO 500                            000226
00220          C --->          GET LOCATION NO.                                                     000226
00222                    CALL GETCOD(3,CMPMOD(I),LOK)                                               000233
00222          C --->          DETERMINE PAGE NO.                                                   000233
00223                    LOCPAG=(LOK/100)*100                                                       000242
00223          C --->          CONVERT LOC TO LOCAL PAGE LOCATION                                   000242
00224                    LOK=LOK-LOCPAG                                                             000247
00224          C --->          TEST TO ASSURE LOC NO. IS ON PAGE                                    000247
00225                    IF(LOK.LT.1.OR.LOK.GT.80)LOCPAG=-1                                         000251
00225          C --->          SKIP INPUTS TO QUANTITIES ON OTHER PAGES                             000251
00227                    IF(LOCPAG.NE.NPAGE)GO TO 500                                               000271
00227          C --->          GET NO. OF INPUTS TO COMP.                                           000271
00231                    CALL GETCOD(5,CMPMOD(I),NINPUT)                                            000274
00231          C --->          BYPASS COMP. WITH NO INPUTS                                          000274
00232                    IF(NINPUT.LE.0)GO TO 500                                                   000303
00232          C --->          GET INPUTS LIST                                                      000303
00234                    CALL CLADHS(7,INPUTS,NINPUT,I)                                             000306
00234          C --->          INITIALIZE NO. INPUTS COUNTER    CURRENT INPUT COMP. NAME            000306
00235          420       NOIN=0                                                                     000315
00236                    MORE=0                                                                     000317
00237                    INCOM=ASTRSK                                                               000320
00240                    IF(IDIAG.EQ.30)WRITE(IWRITE,423)CMPMOD(I),(INPUTS(J),J=1,NINPUT)           000323
00253          423       FORMAT(' SCHEMA-INPUTS  ',A10/10(1X,A10))                                  000345
00253          C --->          SCAN INPUTS                                                          000345
00253                    DO 480 J=1,NINPUT                                                          000345
00253          C --->          TEST IF INPUT IS FROM CURRENT COMP. I.E. PARAMETER                   000345
00254                    IF(KOMSTR(INPUTS(J),4,1,BLNK,1).EQ.0)GO TO 480                             000345
00254          C --->          IS THIS A NEW INPUT SOURCE'                                          000345
00256                    IF(KOMSTR(INCOM,4,4,INPUTS(J),4).EQ.0)GO TO 440                            000357
00256          C ---           BYPASS NAME LOAD IF 2ND COMPONENT APPEARS                            000357
00260                    IF(MORE.NE.0)GO TO 460                                                     000372
00260          C --->          SAVE NEW SOURCE NAME                                                 000372
00262                    INCOM=INPUTS(J)                                                            000374
00263                    MORE=1                                                                     000376
00263          C --->          ADVANCE INPUT COUNT                                                  000376
00264          440       NOIN=NOIN+1                                                                000401
00265                    NAMES(NOIN)=INPUTS(J)                                                      000405
00266                    INPUTS(J)=BLNK                                                             000407
00267                    GO TO 480                                                                  000411
00270          460       MORE=2                                                                     000413
00271          480       CONTINUE                                                                   000416
00273          C --->          IS THERE A CURRENT INPUT COMPONENT                                   000416
00273                    IF(NOIN.LE.0)GO TO 500                                                     000416
```

```
00275     125*          CALL CONNCT(PAGE,NPAGE,LOK,NAMES,NDIN,CNPROD,NDCOMP)           000421
00275     126*    C ---        DO MORE COMPONENTS PROVIDE INPUTS                       000421
00276     127*          IF(MORE.EQ.2)GO TO 420                                         000432
00300     128*    500   CONTINUE                                                       000441
00300     129*    C --->       PRINT PAGE                                              000441
00302     130*    602   NAME=NPAGE/100                                                 000441
00303     131*          WRITE(IWRITE,605)TITLE,NAME,PAGE                               000444
00310     132*    605   FORMAT((1H1,29X,7A10,24X,*PAGE *,I3/(2X,13A10))                000461
00313     133*    C --->       TEST FOR LAST PAGE                                      000461
00311     134*          IF(NPAGE.GE.MAXPAG)RETURN                                      000461
00313     135*          NPAGE=NPAGE+100                                                000470
00314     136*          GO TO 100                                                      000473
00315     137*          END @ SCHEMA  ***************************                      000535
```

```
SUBROUTINE SYMBOL    ENTRY POINT 000475

STORAGE USED  CODE(1) 000513; DATA(0) 000102; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CIO    000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0004   STRMOV
 0005   PUTT
 0006   NWDU1$
 0007   NI02$
 0010   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000127 137G      0001    000167 153G      0001    000154 200L      0001    000162 205L      0001    000325 207G
 0001    000234 206L      0001    000243 215L      0000    000010 227F      0001    000245 220L      0001    000441 231G
 0001    000252 300L      0001    000262 400L      0001    000371 415L      0001    000373 420L      0001    000400 500L
 0000 D 000000 ASTRSK     0000 I 000004 I        0000 I 000007 ICOL     0003 I 000002 IDIAG    0000    000045 INJP$
 0003    000000 IREAD     0003 I 000001 IWRITE   0000 I 000006 K        0000 I 000005 L        0000 I 000003 LOCCOL
 0000 I 000002 LOCLIN


00100      CSYMBOL                                                                000002
00101            SUBROUTINE SYMBOL(PAGE,CONNAM,ISYMB,LOK)                         000002
00101      C  VERSION 1.2                       REVISED  OCT 17 1975              000002
00101      C  PURPOSE   ADD COMPONENT SYMBOLS AND NAMES TO ECS MODEL SCHEMATIC     000002
00101      C  CALL SEQUENCE    PAGE    - 13X56 ARRAY CONTAINING HOLLERITH         000002
00101      C                             REPRESENTATION OF A PAGE                 000002
00101      C                   CONNAM - NAME OF COMPONENT TO BE ADDED TO PAGE     000002
00101      C                   ISYMB  - SYMBOL TYPE NO.                           000002
00101      C                   LOK    - LOCATION OF SYMBOL ON PAGE                000002
00101      C  DESIGNED BY  J.D.BURROUGHS                   JUNE 1974              000002
00103                            IMPLICIT DOUBLE PRECISION (A-Z)                  000002
00104                            IMPLICIT INTEGER (I,J,K,L,M,N)                   000002
00105            COMMON/CIO/IREAD,IWRITE,IDIAG                                    000002
00106            DIMENSION PAGE(13,56)                                            000002
00106      C     LITERAL 'POOL' TO SATISFY DBLE PRECSN                            000002
00107            DATA ASTRSK/12H************  /                                   000002
00107      C --->      LOCATION LINE NO.                                          000002
00111            LOCLIN=7*((LOK-1)/10)+3                                          000002
00111      C --->      LOCATION COLUMN  NO.                                       000002
00112            LOCCOL=(MOD(LOK-1,10)+1)*13-10                                   000012
00112      C --->      ADD COMPONENT NAME TO PAGE                                 000012
00113            CALL STRMOV(CONNAM,1,3,PAGE(1,LOCLIN),LOCCOL+3)                  000021
```

```
00114          IF(IDIAG.EQ.22)WRITE(IWRITE,22)COMNAM,ISYMB,LOK                    000036
00122    22    FORMAT(' SYMBOL ',A10,2I10)                                        000051
00122    C --->      TEST FOR SYMBOL TYPE                                         000051
00122    C                                                                        000051
00122    C           SYMBOL NUMBERS LESS THAN 64 SHOULD NOT BE USED DUE TO        000051
00122    C           CSORT REPLACING 00B WITH 55B WHEN CALLED BY FILOAD.          000051
00122    C                                                                        000051
00123          IF(ISYMB.EQ.100)GO TO 200                                          000051
00125          IF(ISYMB.EQ.200)GO TO 400                                          000054
00127          IF(ISYMB.EQ.300) GO TO 300                                         000057
00131          IF(ISYMB.EQ.400)GO TO 500                                          000062
00131    C --->      DEFAULT SYMBOL - SQUARE                                      000062
00133          LOCLIN=LOCLIN-2                                                    000065
00133    C --->      TOP AND BOTTOM LINES                                         000065
00134          CALL STRMOV(ASTRSK,1,10,PAGE(1,LOCLIN),LOCCOL)                     000070
00135          CALL STRMOV(ASTRSK,1,10,PAGE(1,LOCLIN+5),LOCCOL)                   000105
00135    C --->      SIDES                                                        000105
00136          DO 100 I=1,4                                                       000127
00141          CALL PUT(PAGE(1,LOCLIN+I),LOCCOL,12H*                              000127
00142          CALL PUT(PAGE(1,LOCLIN+I),LOCCOL+9,12H*                            000135
00145    100   CONTINUE                                                           000150
00145          RETURN                                                             000150
00145    C --->      COMPRESSOR SYMBOL                                            000150
00146    200   L=LOCCOL                                                           000154
00147          K=2                                                                000155
00150          ICOL=L+1                                                           000157
00151    205   LOCLIN=LOCLIN-5                                                    000162
00152          DO 220 I=1,10                                                      000167
00155          LOCLIN=LOCLIN+1                                                    000167
00155    C --->      TEST TO PREVENT TOP OF SYMBOL FROM GOING OFF TOP OF PAGE     000167
00156          IF(LOCLIN.LT.1)GO TO 208                                           000172
00156    C --->      TEST TO PREVENT BOTTOM OF SYMBOL FROM GOING OFF PAGE         000172
00160          IF(LOCLIN.GT.56)RETURN                                             000175
00160    C --->      STRAIGHT EDGE OF SYMBOL                                      000175
00162          CALL STRMOV(12H*           ,1,1,PAGE(1,LOCLIN),L)                  000204
00162    C --->      SLOPING EDGE OF SYMBOL                                       000204
00163          CALL STRMOV(12H*           ,1,1,PAGE(1,LOCLIN),ICOL)               000221
00163    C ==--->       TEST TO REVERSE SLOPE OF RIGHT EDGE                       000221
00164    208   IF(I.EQ.5)GO TO 215                                                000234
00166          ICOL=ICOL+K                                                        000236
00167          GO TO 220                                                          000241
00170    215   K=-K                                                               000243
00171    220   CONTINUE                                                           000246
00173          RETURN                                                             000246
00173    C --->      TURBINE SYMBOL                                               000246
00174    300   L=LOCCOL+9                                                         000252
00175          K=-2                                                               000254
00176          ICOL=L-1                                                           000256
00177          GO TO 205                                                          000260
00177    C --->      CIRCLE SYMBOL                                                000260
00200    400   LOCLIN=LOCLIN-2                                                    000262
00201          CALL STRMOV(12H  ******    ,1,10,PAGE(1,LOCLIN),LOCCOL)            000264
00202          CALL STRMOV(12H  ******    ,1,10,PAGE(1,LOCLIN+5),LOCCOL)          000301
00203          K=1                                                                000314
00204          L=LOCCOL+1                                                         000316
00205          ICOL=L+7                                                           000320
00.      C --->      ADD SIDES TO SYMBOL                                          000320
```

```
00206               DO 420 I=1,4                                                               000325
00211               LOCLIN=LOCLIN+1                                                            000325
00211         C --->        LEFT EDGE OF SYMROL                                                000325
00212               CALL STRMOV(12H*              ,1,1,PAGE(1,LOCLIN),L)                       000330
00212         C --->        RIGHT EDGE OF SYMROL                                               000330
00213               CALL STRMOV(12H*              ,1,1,PAGE(1,LOCLIN),ICOL)                    000344
00213         C --->        REVERSE SLOPE OF EDGES                                             000344
00214               IF(I.EQ.2)GO TO 415                                                        000356
00216               L=L-K                                                                      000361
00217               ICOL=ICOL+K                                                                000364
00220               GO TO 420                                                                  000367
00221         415   K=-K                                                                       000371
00222         420   CONTINUE                                                                   000374
00224               RETURN                                                                     000374
00224         C ---         OPTIMAL CONTROLLER SYMBOL                                          000374
00225         500   LOCLIN=LOCLIN-2                                                            000400
00225         C --->        TOP AND BOTTOM LINES                                               000400
00226               CALL STRMOV(12H 00000000    ,1,10,PAGE(1,LOCLIN),LOCCOL)                   000402
00227               CALL STRMOV(12H 00000000    ,1,10,PAGE(1,LOCLIN+5),LOCCOL)                 000417
00227         C --->        SIDES                                                              000417
00230               DO 520 I=1,4                                                               000441
00233               CALL PUT(PAGE(1,LOCLIN+I),LOCCOL,12HO                                      000441
00234               CALL PUT(PAGE(1,LOCLIN+I),LOCCOL+9,12HO                                    000447
00235         520   CONTINUE                                                                   000462
00237               RETURN                                                                     000462
00243               END $ SYMBOL  ************************                                     000512
```

```
SUBROUTINE TABCAL     ENTRY POINT 000116


STORAGE USED  CODE(1) 000122; DATA(0) 000062; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CTAB    000003


EXTERNAL REFERENCES (BLOCK, NAME)

 0004    STRMOV
 0005    GETCOD
 0006    NWDU$
 0007    NI02$
 0010    NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000013 11F        0001    000014 116G       0000    000041 51F        0001    000057 60L        0001    000070 70L
 0001    000072 80L        0000    000042 61F        0000 D 000007 ANAME      0000 D 000000 BLNK       0000 D 000004 COMP
 0000 D 000002 COMPS      0000 I 000006 I      .    0000    000054 INJP$      0000 I 000011 N          0003 I 000000 NOTAB
 0000 I 000012 NI         0003 D 000001 TABNAM



00100      1*      CTABCAL                                                                                    000000
00101      2*            SUBROUTINE TABCAL                                                                    000000
00101      3*      C  PURPOSE  GENERATE TABLE INPUT REQUIREMENTS LIST ON FILE 12                                000000
00103      4*                        IMPLICIT DOUBLE PRECISION (A-Z)                                          000000
00104      5*                        IMPLICIT INTEGER (I,J,K,L,M,N)                                           000000
00105      6*            COMMON/CTAB/NOTAB,TABNAM(1)                                                          000000
00106      7*            DATA BLNK /12H                                                                       000000
00110      8*            WRITE(12,11)                                                                         000000
00112      9*      11    FORMAT(16X,'TABLES REQUIRED'//                                                       000004
00112     10*           12X,'COMPONENT   TABLE    NO. INDEP.   MAX. DATA'/                                    000004
00112     11*           24X,'NAME',7X,'NAME',5X,'VARIABLES    ALLOWED')                                       000004
00113     12*            COMPS=BLNK                                                                           000004
00114     13*            COMP=COMPS                                                                           000006
00114     14*      C --->      SCAN TABLES.                                                                   000006
00115     15*            DO 100 I=1,NOTAB                                                                     000014
00115     16*      C --->      GET TABLE NAME                                                                 000014
00120     17*            CALL STRMOV(TABNAM(I),1,7,ANAME,1)                                                   000014
00120     18*      C --->      GET MAXIMUM DIMENSION FOR TABLE                                                000014
00121     19*            CALL GETCOD(5,TABNAM(I),N)                                                           000024
00122     20*            NI=IABS(N)                                                                           000033
00122     21*      C --->      GET SPECIFIC COMPONENT NAME                                                    000033
00123     22*            CALL STRMOV(ANAME,4,4,COMP,1)                                                        000035
00        23*            IF(COMP.EQ.COMPS) GO TO 60                                                           000044
00        24*            WRITE(12,5)                                                                          000047
```

```
00130     25*     51     FORMAT(1H )                                                      000054
00131     26*            COMPS=COMP                                                       000054
00132     27*     60     N1=N1-3                                                          000057
00132     28*     C --->      TEST FOR SINGLE OR DOUBLE INDEP. VARIABLE TABLE             000057
00133     29*            IF(N.GT.1) GO TO 70                                              000061
00135     30*            N=1                                                              000064
00136     31*            GO TO 80                                                         000066
00137     32*     70     N=2                                                              000070
00140     33*     80     WRITE(12,81)COMP,ANAME,N,N1                                      000072
00146     34*     81     FORMAT(6X,A4,5X,A7,6X,I1,10X,I4)                                 000104
00147     35*     100    CONTINUE                                                         000104
00151     36*            RETURN                                                           000104
00152     37*            END @ TABCAL  **************************                         000121
```

```
SUBROUTINE TABDAT     ENTRY POINT 000337


STORAGE USED  CODE(1) 000345; DATA(0) 000105; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CTAB    000003


EXTERNAL REFERENCES (BLOCK, NAME)

 0004    NKCOD$
 0005    LINL
 0006    STRMOV
 0007    FETCOD
 0010    MVBU$
 0011    NIO2$
 0012    NIO3$
 0013    NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000     000054 101F       0001     000024 140G       0001     000060 155G       0001     000135 174G       0000     000056 201F
 0001     000143 201G       0001     000235 225G       0000     000257 231F       0001     000245 233G       0000     000047 91F
 0000 D 000044 AN          0000 D 000020 ANAME       0000 D 000024 BLNK        0000 D 000026 HDATA       0000 D 000036 HLOCTB
 0000 D 000034 HMAXDM      0000 D 000032 HSLASH      0000 D 000030 HTABNM      0000 I 000041 I           0000   000072 INJP$
 0000 I 000040 ISOUR       0000 I 000046 LOK         0000 I 000043 N           0003 I 000000 NOTAB       0000 I 000042 NID
 0000 D 000000 SOURCE      0003 D 000001 TABNAM



000100       1*      CTABDAT                                                                                      000000
000101       2*            SUBROUTINE TABDAT                                                                      000000
000101       3*      C  VERSION 3.                       REVISED MAY 4 1976                                       000000
000101       4*      C  PURPOSE   GENERATE DATA STATEMENTS FOR MODEL TABLE DATA INPUT CONTROL                     000000
000101       5*      C   DESIGNED BY    J.D.BURROUGHS                        DATE  MARCH 1975                     000000
000103       6*                          IMPLICIT DOUBLE PRECISION (A-Z)                                          000000
000104       7*                          IMPLICIT INTEGER (I,J,K,L,M,N)                                           000000
000105       8*            COMMON/CTAB/NOTAB,TABNAM(1)                                                            000000
000106       9*            DIMENSION SOURCE(8)                                                                    000000
000107      10*            DIMENSION ANAME(2)                                                                     000000
000110      11*            DATA ANAME /24H                                                                        000000
000112      12*            DATA BLNK /12H            /        .                                                   000000
000114      13*            DATA HDATA /12H       DATA  /                                                          000000
000116      14*            DATA HTABNM /12H TABNAM/    /                                                          000000
000120      15*            DATA HSLASH /12H/           /                                                          000000
000122      16*            DATA HMAXDM /12H MAXDIM/    /                                                          000000
000124      17*            DATA HLOCTB /12H LOCTAB/    /                                                          000000
001         18*      C ============  SET NUMBER OF TABLES IN MODEL                                                000000
001         19*            WRITE(9,91)NOTAB                                                                       000000
```

```
00131      91    FORMAT(6X,'DATA NOTAB/',I3,'/')                          000005
00132            IF(NOTAB.LE.0)RETURN                                     000005
00132      C ---------->      LOAD TABLE NAME DATA                        000005
00134            SOURCE(1)=HDATA                                          000013
00135            SOURCE(2)=HTABNM                                         000015
00136            ISOUR=19                                                 000017
00137            DO 100 I=3,8                                             000024
00142      100   SOURCE(I)=BLNK                                           000024
00142      C --->      CALC. NO. OF CHARACTERS IN TABLE NAME LIST         000024
00144            NID=12*NOTAB                                             000026
00145            ENCODE(4,101,NID)NID                                     000031
00153      101   FORMAT(I3,1HH)                                           000040
00153      C --->      ADD NO. OF CHARACTERS TO DATA STATEMENT LINE       000040
00151            CALL LINE(0,SOURCE,ISOUR,NID,4,9)                        000040
00152            ANAME(1)=BLNK                                            000050
00153            ANAME(2)=BLNK                                            000052
00153      C --->      SCAN TABLES                                        000052
00154            DO 200 I=1,NOTAB                                         000060
00157            CALL STRMOV(TABNAM(I),1,6,ANAME,1)                       000060
00157      C --->      ADD TABLE NAME TO LINE                             000060
00160            CALL LINE(1,SOURCE,ISOUR,ANAME,12,9)                     000070
00161      200   CONTINUE                                                 000101
00163            CALL LINE(1,SOURCE,ISOUR,HSLASH,1,9)                     000101
00164            WRITE(9,201)SOURCE                                       000111
00167      201   FORMAT(8A10)                                             000121
00167      C ---------->      LOAD TABLE DIMENSION DATA                   000121
00170            SOURCE(1)=HDATA                                          000121
00171            SOURCE(2)=HMAXDM                                         000123
00172            ISOUR=19                                                 000125
00173            DO 220 I=3,8                                             000135
00176      220   SOURCE(I)=BLNK                                           000135
00176      C --->      SCAN TABLES                                        000135
00200            DO 240 I=1,NOTAB                                         000143
00200      C --->      GET MAX. TABLE DIMENSION                           000143
00203            CALL GETCOD(5,TABNAM(I),N)                               000143
00204            N=IABS(N)                                                000152
00204      C --->      CONVERT TO DISPLAY CODE                            000152
00205            ENCODE(5,231,AN)N                                        000154
00210      231   FORMAT(I4,1H,)                                           000163
00211            IF(I.EQ.NOTAB)CALL STRMOV(HSLASH,1,1,AN,5)               000163
00211      C --->      ADD MAX. DIMENSION TO LINE                         000163
00213            CALL LINE(0,SOURCE,ISOUR,AN,5,9)                         000176
00214      240   CONTINUE                                                 000211
00216            WRITE(9,201)SOURCE                                       000211
00216      C ---------->      LOAD TABLE LOCATION DATA                    000211
00221            SOURCE(1)=HDATA                                          000221
00222            SOURCE(2)=HLOCTB                                         000223
00223            ISOUR=19                                                 000225
00224            DO 300 I=3,8                                             000235
00227      300   SOURCE(I)=BLNK                                           000235
00231            LOK=1                                                    000237
00231      C --->      SCAN TABLES                                        000237
00232            DO 325 I=1,NOTAB                                         000245
00232      C --->      CONVERT TO DISPLAY CODE                            000245
00235            ENCODE(5,231,AN)LOK                                      000245
00243            IF(I.EQ.NOTAB)CALL STRMOV(HSLASH,1,1,AN,5)               000254
00243      C --->      ADD TABLE LOCATION NO. TO LINE                     000254
```

```
00242      77*            CALL LINE(0,SOURCE,ISOUR,AN,5,9)                                      000267
00242      78*      C --->       GET MAX. DIMENSION OF TABLE                                    000267
00243      79*            CALL GETCOD(S,TABNAM(I),N)                                            000277
00243      80*      C --->       CALC. THE NEXT TABLE STARTING LOCATION                         000277
00244      81*            LOK=LOK+IABS(N)                                                       000306
00245      82*      320   CONTINUE                                                              000313
00247      83*            WRITE(9,20)ISOURCE                                                    000313
00252      84*            RETURN                                                                000323
00253      85*            END A TABDAT  *************************                               000344
```

```
SUBROUTINE TABGEN    ENTRY POINT 000145

STORAGE USED  CODE(1) 000152; DATA(0) 000071; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CTAB   000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0004   GETCOD
 0005   STRMOV
 0006   NDCON$
 0007   LINE
 0010   NWDU$
 0011   NIO2$
 0012   NIO3$
 0013   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000   000037 10F        0000   000043 105F       0001   000024 125G       0001   000034 133G       0000   000046 201F
 0000 D 000035 AN         0000 D 000024 BLNK       0000 D 000030 HBLNCM     0000 D 000026 HCOMM      0000 I 000032 I
 0000   000060 INJP$      0000 I 000033 ISOUR      0000 I 000034 N          0003 I 000000 NOTAB      0000 D 000020 SOTAB
 0000 D 000000 SOURCE     0003 D 000001 TABNAM


00100      1*      CTABGEN                                                                              000000
00101      2*            SUBROUTINE TABGEN                                                              000000
00101      3*      C  PURPOSE   GENERATE THE TABLE COMMON FOR ECS MODEL                                     000000
00101      4*      C  CALL SEQUENCE   NTAB - TOTAL NO. OF TABLES REQ"D BY MODEL                         000000
00101      5*      C  METHOD   THE NAMES OF THE TABLES AND THEIR DIMENSIONS ARE STORED                      000000
00101      6*      C           IN TABNAM.  THE NAME IS STORED IN THE FIRST 7 CHARACTERS                     000000
00101      7*      C           OF EACH WORD AND THE DIMENSION IS STORED IN THE LAST 2                       000000
00101      8*      C           CHARACTERS VIA THE ROUTINE PUTCOD.                                           000000
00103      9*                        IMPLICIT DOUBLE PRECISION (A-Z)                                        000000
00104     10*                        IMPLICIT INTEGER (I,J,K,L,M,N)                                         000000
00105     11*            COMMON/CTAB/NOTAB,TABNAM(1)                                                        000000
00106     12*            DIMENSION SOURCE(8),SOTAB(2)                                                       000000
00107     13*            DATA BLNK  /12H            /                                                       000000
00111     14*            DATA HCOMM /12H      COMM  /                                                       000000
00113     15*            DATA HBLNCM /12HON/CTABLE/  /                                                      000000
00115     16*            IF(NOTAB.LE.0)RETURN                                                               000000
00117     17*            WRITE(6,10)                                                                        000005
00121     18*      10    FORMAT('0 --->      TABLES')                                                       000012
00122     19*            SOURCE(1)=HCOMM                                                                    000012
00123     20*            SOURCE(2)=HBLNCM                                                                   000014
00124     21*            DO 100 I=3,8                                                                       000024
```

```
00127      22*      100   SOURCE(I)=BLNK                                                    000024
00131      23*            ISOUR=22                                                          000026
00131      24*      C ---        SCAN ALL TABLES IN THE MODEL                               000026
00132      25*            DO 200 I=1,NOTAB                                                  000034
00132      26*      C --->       GET TABLE DIMENSION                                        000034
00135      27*            CALL GETCOD(5,TABNAM(I),N)                                        000034
00136      28*            N=IABS(N)                                                         000043
00136      29*      C ---        GET TABLE NAME                                             000043
00137      30*            CALL STRMOV(TABNAM(I),1,7,SOTAB,1)                                000045
00137      31*      C --->       CONVERT DIMENSION TO BCD                                   000045
00140      32*            ENCODE(6,105,AN)N                                                 000056
00143      33*      105   FORMAT(1H(,I3,2H),)                                               000065
00143      34*      C ---        REMOVE COMMA IF LAST TABLE                                 000065
00144      35*            IF(I.GE.NOTAB)CALL STRMOV(BLNK,1,1,AN,6)                          000065
00146      36*            CALL STRMOV(AN,1,6,SOTAB,8)                                       000100
00146      37*      C ---        ADD TABLE NAME TO SOURCE LINE                              000100
00147      38*            CALL LINE(0,SOURCE,ISOUR,SOTAB,13,9)                              000103
00150      39*      200   CONTINUE                                                          000122
00152      40*            WRITE(9,201)SOURCE                                                000122
00155      41*      201   FORMAT(8A10)                                                      000132
00156      42*            RETURN                                                            000132
00157      43*            END @ TABGEN  *************************                           000151
```

```
SUBROUTINE VLINE     ENTRY POINT 000115

STORAGE USED  CODE(1) 000132; DATA(0) 000025; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   PUTT
 0004   KOMSTR
 0005   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000016 100L       0001    000045 127G       0001    000024 200L       0001    000102 300L       0000 D 000000 HA
 0000 D 000034 HI          0000 D 000002 HV          0000 I 000012 I           0000    000014 INJP$      0000 I 000010 I1
 0000 I 000011 I2          0004 I 000000 KOMSTR      0000 D 000006 POINT



00100      1*      CVLINE                                                                               000002
00101      2*            SUBROUTINE VLINE(PAGE,ICOL,IN,IR)                                              000002
00101      3*      C  PURPOSE   PAGE - 13X56 ARRAY CONTAINING HOLLORITH                                 000002
00101      4*      C                     REPRESENTATION OF A PAGE                                       000002
00101      5*      C               ICOL - COLUMN NO. OF LINE                                            000002
00101      6*      C               IN   - LINE NO. OF INPUT COMPONENT                                   000002
00101      7*      C               IR   - LINE NO. OF RECEIVING COMPONENT                               000002
00103      8*                         IMPLICIT DOUBLE PRECISION (A-Z)                                   000002
00104      9*                         IMPLICIT INTEGER (I,J,K,L,M,N)                                    000002
00105     10*            DIMENSION PAGE(13,56)                                                          000002
00106     11*            DATA HA /12HA          /                                                       000002
00110     12*            DATA HV /12HV          /                                                       000002
00112     13*            DATA HI /12HI          /                                                       000002
00112     14*      C --->       IS INPUT ABOVE OR BELOW                                                 000002
00114     15*            IF(IN.GE.IR)GO TO 100                                                          000002
00114     16*      C --->       INPUT IS ABOVE                                                          000002
00116     17*            POINT=HV                                                                       000006
00117     18*            I1=IN                                                                          000010
00120     19*            I2=IR                                                                          000012
00121     20*            GO TO 200                                                                      000014
00121     21*      C --->       INPUT IS BELOW                                                          000014
00122     22*      100   POINT=HA                                                                       000016
00123     23*            I1=IR                                                                          000017
00124     24*            I2=IN                                                                          000021
00124     25*      C --->       PLACE POINT ON RECEIVING END OF LINE                                    000021
00125     26*      200   CALL PUTT(PAGE(1,IR),ICOL,POINT)                                               000024
00125     27*      C --->       ADD NO. OF SYMBOLS REQ'D. TO SPAN LINES                                 000024
00126     28*            DO 300 I=I1,I2                                                                 000035
00126     29*      C --->       TEST TO PREVENT OVERWRITING POINTS                                      000035
00131     30*            IF(KOMSTR(PAGE(2,I),ICOL,I,HA,1).EQ.0)GO TO 300                                000045
00133     31*            IF(KOMSTR(PAGE(1,I),ICOL,I,HV,1).EQ.0)GO TO 300                                000051
```

```
00135      32*              CALL PUTT(PAGE(1,I),ICOL,MI)                                                          000072
00136      33*       300    CONTINUE                                                                              000103
00140      34*              RETURN                                                                                000103
00141      35*              END @ VLINE  ********************************                                         000131
```

3.0 SIMULATION PROGRAM DESCRIPTION

3.1 INTRODUCTION

The Simulation program accepts program commands which describe analyses to be performed on the given system model. Each analysis is then performed on the nonlinear system model that was created by the Model Generation program. The Simulation program core requirements vary as a function of model size, growing as the square of the number of states in the model.

3.2 PROGRAM STRUCTURE

Figure 3.2-1 contains a macro flow diagram of the SIMWEST Analysis program. This flow diagram shows the principle tasks of the program. For each task, a statement number of the main, (NONSIM), program is given along with the name of the principle program that accomplishes that task.

The sequence of performing the various tasks depends on the analysis and data requests. As each analysis is performed it's outputs are generated on the lineprinter.

3.2.1 Command Interpretation

Figure 3.2-2 contains a macro flow diagram of the Simulation program command interpretation process. Each input data card is read and printed to provide a record of the progress through the analysis requests. Phrases are identified on each card by the routine NXTPH. When a blank phrase is encountered a new card is read. Each phrase is tested against the three types: command phrases, program names, and program values. If one of these types is recognized the proper action is taken. If the phrase is not one of these types a test is made for an outstanding task. An outstanding task consists of such multiphrase tasks as defining state names, inputing parameter values, specifying initial conditions, etc. If there is no outstanding task the warning message "CAN'T INTERPRET xxxxx" is printed and the program goes on to the next phrase.

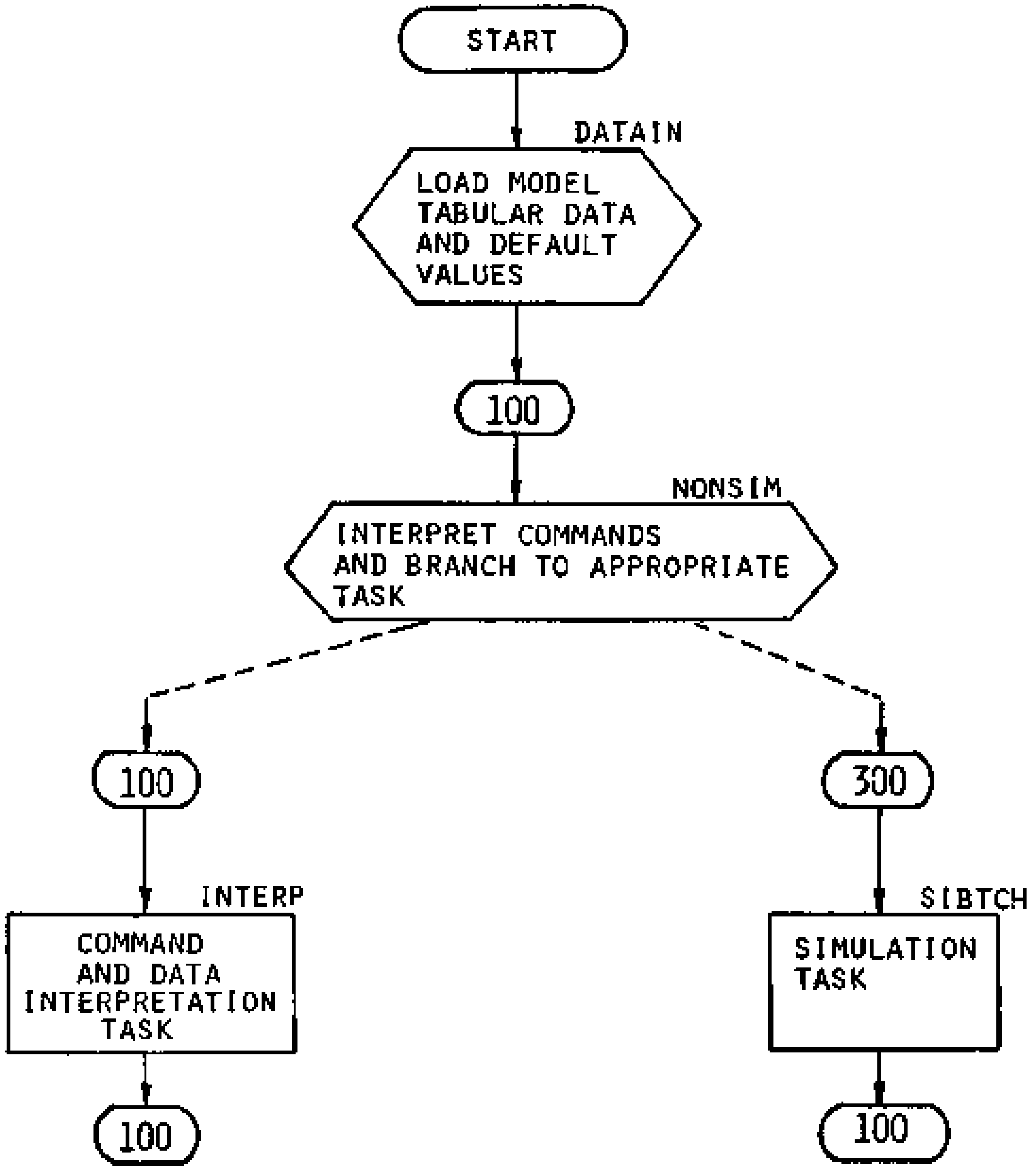
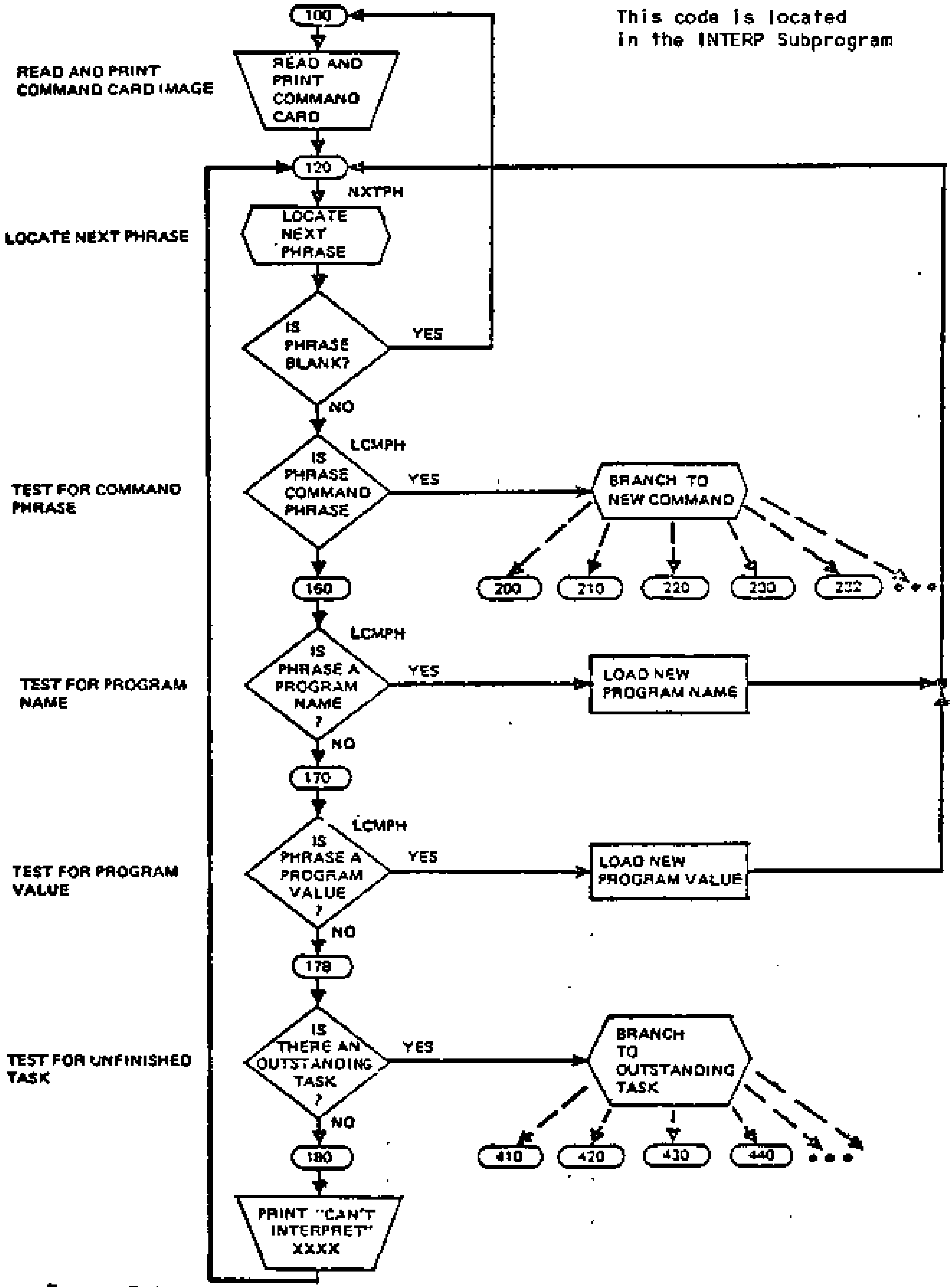

START
DATAIN
LOAD MODEL TABULAR DATA AND DEFAULT VALUES
100
NONSIM
INTERPRET COMMANDS AND BRANCH TO APPROPRIATE TASK
100
300
INTERP
COMMAND AND DATA INTERPRETATION TASK
SIBTCH
SIMULATION TASK
100
100

FIGURE 3.2-1. SIMWEST ANALYSIS PROGRAM - MACRO FLOW DIAGRAM



This code is located
in the INTERP Subprogram

FIGURE 3.2-2. ANALYSIS PROGRAM COMMAND INTERPRETATION - MACRO FLOW DIAGRAM

3.2.2 Temporary Files

Two temporary files SCRTCH25 and SCRTCH26 are used by the Simulation program. SCRTCH25 serves as a temporary buffer for simulation plot data. The plot data for each report interval is stored on SCRTCH25 until all report intervals for the simulation analysis have been completed. Upon completion of the simulation analysis, information describing the number of plots, report intervals, and plot scales are placed on SCRTCH26 and the plot data itself is transferred from SCRTCH25 to SCRTCH26.

Upon completion of all analyses for a particular run, SCRTCH26 is processed by a separate program (NSMPPT) to generate lineprinter plots.

3.3 SIMULATION PROGRAM SOURCE LISTINGS

Compilation listings for the simulation program follows. Some subroutines such as NXTPH and LCMPH are used in several of the programs and will be found in the source listings for the FILOAD program (Section 4.3). There are five subroutines which are only called by the model EQMO or the library components. These are listed after the simulation program source. The names of the simulation routines, in order of appearance, are:

BLOCKDA
CODGEN
CODLOD
DISPLA
DTTIM
FPCT
FSHELL
INIT
INPUTS
INTERP
LPRINT
NAMES
NONSIM
PLINIT
SETIN
SHELLX
SIBTCH
STEP1
TABIN
TITLE
VALUES
VARMOD
VAROUT
XFR
CUBIC
IMPLIC
TBLU1
TBLU2
UNIF

```
BLOCK DATA

STORAGE USED  CODE(1) 000000; DATA(0) 000000; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CSIMUL 000022
 0004   CPRGN  000020
 0005   CPROV  000033
 0006   CSMPAR 000026
 0007   CCOMM  000012
 0010   CSCALE 000366
 0011   CIO    000003
 0012   CPRINT 000036
 0013   COVRLY 000004


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0013 R 000003 CPUSEC    0006 I 000024 ICIND     0007   000000 ICOM      0003 D 000012 IDENT     0011 I 000002 IDIAG
 0003 I 000010 INDEX     0007 I 000011 INDEXX    0003 I 000005 INDMAX    0013 I 000000 INST      0003 I 000002 IOUT
 0007   000010 IPHRS     0003 I 000011 IPLOT     0003 I 000001 IPRATE    0003 I 000000 IPRIN     0011 I 000000 IREAD
 0011 I 000001 IWRITE    0013 I 000002 LOKSIM    0013 I 000001 LOKSS     0012 I 000024 LPRT      0010 I 000360 NPLTS
 0003 I 000004 NPTMAX    0003   000003 NPTS      0010 D 000170 NVAR      0004 D 000000 PRONAM    0012 D 000000 PRINAM
 0005 R 000000 PVALUE    0010 R 000000 SCALE     0006 D 000000 SMPAR     0003 R 000006 TINC      0003 R 000007 TMAX


00100      1*       CBLOCKDA                                                                              000000
00101      2*             BLOCK DATA                                                                      000000
00101      3*       C  VERSION 3.                         REVISED  APRIL 30 1976                          000000
00102      4*             COMMON /CSIMUL/IPRIN,IPRATE,IOUT,NPTS,NPTMAX,INDMAX,TINC,TMAX                   000000
00102      5*            1 ,INDEX,IPLOT,IDENT(4)                                                          000000
00103      6*             COMMON/CPRGN/PRONAM(8)/CPROV/PVALUE(27)/CSMPAR/SMPAR(10),ICIND(2)               000000
00104      7*             COMMON/CCOMM/ICOM(8),IPHRS,INDEXX                                               000000
00105      8*             COMMON /CSCALE/SCALE(5,4,6),NVAR(5 ,2,6),NPLTS(6)                               000000
00106      9*             COMMON/CIO/IREAD,IWRITE,IDIAG                                                   000000
00107     10*             COMMON/CPRINT/PRINAM(10),LPRT(10)                                               000000
00110     11*             COMMON/COVRLY/INST,LOKSS,LOKSIM,CPUSEC                                          000000
00111     12*             DOUBLE PRECISION    IDENT,NVAR,PRONAM,SMPAR,PRINAM                              000000
00112     13*             DATA IPRIN,IPRATE,IOUT,TINC,TMAX/1,2,1,.1,1./                                   000000
00120     14*             DATA IDENT/4*12HTIME         /                                                  000000
00122     15*             DATA INDEX,IPLOT/2,1/,INDEXX/0/                                                 000000
00126     16*             DATA PRONAM/8*12H            /,SMPAR/10*12H                                     000000
00131     17*             DATA PVALUE/-1.,1.,-1.,0.,1.,0.,1.,2.,3.,.1,1.,.1,100.,-1.,1.,5.,               000000
00131     18*            11.,0.,1.,0.,-10.,0.,0.,10.,0.,0.,5./                                            000000
00133     19*             DATA NPTMAX/1/,INDMAX/505/                                                      000000
00136     20*             DATA NPLTS/3*1,3*0/,NVAR/60*12H            /,SCALE/120*0./                      000000
00142     21*             DATA ICIND/2*0/                                                                 000000
00144     22*             DATA IREAD,IWRITE,IDIAG/5,6,0/                                                  000000
00150     23*             DATA PRINAM/10*12H            /,LPRT/10*0/                                      000000
00153     24*             DATA INST,LOKSS,LOKSIM/3*1/,CPUSEC/0./                                          000000
00160     25*             END $  BLOCKDA  ******************************                                  000000
```

```
SUBROUTINE CODGEN    ENTRY POINT 000137

STORAGE USED  CODE(1) 000200; DATA(0) 000016; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CNAMEX 000002
 0004   CNAMER 000002
 0005   CNAMEV 000002
 0006   CNAMEP 000002
 0007   CORDER 000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0010   LCMPH
 0011   NERR4$
 0012   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000072 225L      0001     000100 235L      0001     000106 245L      0001     000114 255L      0001     000122 260L
 0001     000012 80L       0001     000031 90L       0000 D 000002 BLANK      0000     000011 INJP$     0006 D 000000 NAMEP
 0004 D 000000 NAMER       0005 D 000000 NAMEV      0003 D 000000 NAMEX      0007 I 000002 NOP        0007 I 000001 NOV
 0007 I 000000 NOX         0000 D 000000 NTIME


00100      1*      CCODGEN                                                                          000000
00101      2*            SUBROUTINE CODGEN(IDENT,IC,ICODE,$)                                        000000
00101      3*      C  PURPOSE   GENERATE INTEGER IDENTIFICATION CODES GIVEN ALPHANUMERIC            000000
00101      4*      C  CALL SEQUENCE   IDENT = ALPHANUMERIC IDENTIFIER                               000000
00101      5*      C                  IC    = INITIAL CONDITION INDICATOR                           000000
00101      6*      C                  ICODE = INTEGER CODE NUMBER                                   000000
00101      7*      C                  R1    = RETURN TAKEN WHEN IDENTIFIER CAN*T BE FOUND           000000
00101      8*      C                  $     = UNIVAC ARG FOR VARIABLE RETURN                        000000
00101      9*      C  CODE SCHEME   THE SEVENTH COLUMN IS USED TO DESIGNATE WHICH GROUP             000000
00101     10*      C                THE QUANTITY BELONGS.  THE FOLLOWING CODE IS USED               000000
00101     11*      C                  STATE VARIABLES      = 0                                      000000
00101     12*      C                  STATE DERIVATIVES    = 1                                      000000
00101     13*      C                  STATE I.C.*S         = 2                                      000000
00101     14*      C                  VARIABLES            = 3                                      000000
00101     15*      C                  PARAMETERS           = 4                                      000000
00101     16*      C                              ICODE    = 0 IS USED FOR TIME                     000000
00103     17*            COMMON/CNAMEX/NAMEX(1)/CNAMER/NAMER(1)/CNAMEV/NAMEV(1)/CNAMEP/             000000
00103     18*           1 NAMEP(1)                                                                  000000
00104     19*            COMMON/CORDER/NOX,NOV,NOP                                                  000000
00105     20*            DOUBLE PRECISION IDENT,NAMEX,NAMER,NAMEV,NAMEP,NTIME,BLANK                 000000
0         21*            DATA NTIME/12HTIME        /,BLANK/12H                                      000000
0         22*            IF(IDENT.EQ.BLANK)GO TO 260                                                000000
```

```
00111      23*     C   TEST FOR TIME CODE                                    000000
00113      24*            IF(IDENT.NE.NTIME) GO TO 80                        000002
00115      25*            ICODE=0                                            000005
00116      26*            RETURN                                             000006
00116      27*     C   SEARCH STATE NAMELIST                                 000006
00117      28*     80     CALL LCMPH(IDENT,NAMEX,NOX,1,ICODE)                000012
00120      29*            IF(ICODE.EQ.0) GO TO 90                            000020
00122      30*            IF(IC.EQ.0) RETURN                                 000022
00124      31*            GO TO 255                                          000027
00124      32*     C   SEARCH VARIABLES NAMELIST                             000027
00125      33*     90     CALL LCMPH(IDENT,NAMEV,NOV,1,ICODE)                000031
00126      34*            IF(ICODE.NE.0) GO TO 225                           000037
00126      35*     C   SEARCH RATES NAMELIST                                 000037
00130      36*            CALL LCMPH(IDENT,NAMEP,NOX,1,ICODE)                000041
00131      37*            IF(ICODE.NE.0) GO TO 235                           000050
00131      38*     C   SEARCH PARAMETER NAMELIST                             000050
00133      39*            CALL LCMPH(IDENT,NAMEP,NOP,1,ICODE)                000052
00134      40*            IF(ICODE.NE.0) GO TO 245                           000061
00134      41*     C   IDENTIFIER CAN*T BE RECOGNIZED.                       000061
00136      42*            ICODE=-1                                           000063
00137      43*            RETURN 4                                           000065
00140      44*     225    ICODE=ICODE+3000000                                000072
00141      45*            RETURN                                             000074
00142      46*     235    ICODE=ICODE+1000000                                000100
00143      47*            RETURN                                             000102
00144      48*     245    ICODE=ICODE+4000000                                000106
00145      49*            RETURN                                             000110
00146      50*     255    ICODE=ICODE+2000000                                000114
00147      51*            RETURN                                             000116
00150      52*        260 ICODE=-1                                           000122
00151      53*            RETURN                                             000123
00152      54*            END J  CONGEN  **************************          000177
```

```
SUBROUTINE COOLOD    ENTRY POINT 000101


STORAGE USED  CODE(1) 000114; DATA(0) 000030; BLANK COMMON(2) 000000


EXTERNAL REFERENCES (BLOCK, NAME)

 0003    PUTT
 0004    NNCOD$
 0005    GETT
 0006    NIO2$
 0007    NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000    000011 11F        0001    000007 110G      0001    000034 122G       0001    000057 50L        0000 D 000000 BLNK
 0000 I 000006 I           0000    000014 INJP$     0000 I 000010 J           0000 I 000007 K          0000 D 000002 KAR
 0000 D 000004 NUM



00101       1*      CCOOLOD                                                                                       000007
00101       2*            SUBROUTINE CODLOD(NAME,N,INITAL)                                                        000007
00101       3*      C  PURPOSE   LOAD NAME ARRAYS WITH DEFAULT NAMES.                                             000007
00101       4*      C  CALL SEQUENCE    NAME   = N X 1 NAME ARRAY.                                                000007
00101       5*      C                   N      = NO. OF NAMES IN ARRAY.                                           000007
00101       6*      C                   INITAL = INITIAL CHARACTER WORD.                                          000007
00103       7*            DOUBLE PRECISION NAME(N)                                                                000007
00104       8*            DOUBLE PRECISION BLNK,INITAL,KAR,NUM                                                    000007
00105       9*            DATA BLNK/12H                                                                           000007
00105      10*      C   SCAN NAMES.                                                                               000007
00107      11*            DO 100 I=1,N                                                                            000007
00107   .  12*      C  BLANK OUT NAME.                                                                            000007
00112      13*            NAM( I)=BLNK                                                                            000007
00112      14*      C  PUT INITIAL CHARACTER IN 1ST CHARACTER OF NAME.                                            000007
00113      15*            CALL PUTT(NAME(I),1,INITAL)                                                             000011
00113      16*      C  CONVERT I TO BCD.                                                                          000011
00114      17*            ENCODE(10,11,NUM)I                                                                      000023
00117      18*      11    FORMAT(I10)                                                                             000027
00120      19*            K=2                                                                                     000027
00120      20*      C  SCAN CHARACTERS OF NUM FOR NUMERIC VALUE.                                                  000027
00121      21*            DO 50 J=1,10                                                                            000034
00121      22*      C  GET JTH CHARACTER OF NUM.                                                                  000034
00124      23*            CALL GETT(NUM,J,KAR)                                                                    000034
00124      24*      C  TEST FOR BLANK CHARACTERS AND SKIP THESE.                                                  000034
00125      25*            IF(KAR.EQ.BLNK) GO TO 50                                                                000041
00125      26*      C  LOAD NON-BLANK CHARACTERS CONTAINING NUMERIC INTO NAME.                                    000041
00127      27*            CALL PUTT(NAME(I),K,KAR)                                                                000044
00         28*            K=K+1                                                                                   000053
00         29*         50 CONTINUE                                                                                000054
```

```
00133     30*        100 CONTINUE                                                                   000069
00135     31*            RETURN                                                                     000069
00136     32*            END @  CODLOD  ****************************                                 000113
```

```
SUBROUTINE DISPLA    ENTRY POINT 000151

STORAGE USED  CODE(1) 000177; DATA(0) 000020; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CSCALE 000366

EXTERNAL REFERENCES (BLOCK, NAME)

 0004    LCMPH
 0005    NUMERC
 0006    BCDREL
 0007    NERR2$
 0010    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000127 100L       0001    000032 20L        0001    000132 200L       0001    000137 300L       0001    000061 40L
 0001    000106 60L        0001    000120 80L        0001    000125 90L        0000 I 000007 ICODE       0000    000012 INJP$
 0003 D 000000 LIST        0000 I 000006 NPLT        0003 I 000360 NPLTS       0003 D 000170 NVAR        0003 R 000000 SCALE


00100         CDISPLA                                                                           000003
00101              SUBROUTINE DISPLA(IDSPLY,IPHRS,MODE,ICOL)                                    000003
00101         C  PURPOSE   INTERPRETS INPUT DATA PHRASES THAT DESCRIBE GRAPHIC DISPLAY          000003
00101         C  CALL SEQUENCE    IDSPLY = DISPLAY NUMBER.                                      000003
00101         C                   IPHRS  = PHRASE TO BE INTERPRETED.                            000003
00101         C                   MODE   = MODE = 1,2,3 INDICATES THAT VS,YRANGE,OR             000003
00101         C                                   XRANGE RESPECTIVELY WAS THE LAST INTRUCTION.  000003
00101         C                   ICOL   = SET EQUAL TO THE COLUMN NUMBER IN SCALE.             000003
00103              COMMON/CSCALE/SCALE(5,4,6),NVAR(5,2,6),NPLTS(6)                              000003
00104              DOUBLE PRECISION NVAR,IPHRS,LIST(3)                                          000003
00105              DATA LIST/36HVS          YRANGE       XRANGE                                 000003
00105         C  CURRENT NUMBER OF PLOTS/DISPLAY.                                               000003
00107              NPLT=NPLTS(IDSPLY)                                                           000003
00107         C  SEARCH FOR COMMAND WORD.                                                       000003
00110              CALL LCMPH(IPHRS,LIST,3,1,ICODE)                                             000005
00111              IF(ICODE.LE.0) GO TO 20                                                      000014
00113         C  SAVE ICODE IN MODE AND BRANCH TO SET ICOL IF REQUIRED.                         000014
00113              MODE=ICODE                                                                   000017
00114              GO TO (100,200,300),ICODE                                                    000021
00114         C  TEST FOR NUMERIC PHRASE.                                                       000021
00115         20   CALL NUMERC(IPHRS,$60)                                                       000032
00116              IF(MODE.LE.2) GO TO 40                                                       000035
00116         C  CONVERT X SCALE FROM A TO G FORMAT.                                            000035
00122              CALL BCDREL(SCALE(NPLT,ICOL,IDSPLY),IPHRS$)                                  000041
                   ICOL=9                                                                       000053
```

```
00122          RETURN                                             000055
00122    C  CONVERT Y SCALE FROM A TO G FORMAT.                   000055
00123    40    CALL BCDREL(SCALE(NPLT,ICOL,IDSPLY),IPHRS)         000061
00124          ICOL=2                                             000072
00125          RETURN                                             000074
00126    60    IF(MODE.EQ.1) GO TO 80                             000100
00130          NPLT=MIN0(NPLT+1,5)                                000102
00131          NPLTS(IDSPLY)=NPLT                                 000111
00131    C  LOAD Y AXIS NAME.                                     000111
00132          NVAR(NPLT,1,IDSPLY)=IPHRS                          000112
00133          GO TO 90                                           000116
00133    C  LOAD X AXIS NAME.                                     000116
00134    80    NVAR(NPLT,2,IDSPLY)=IPHRS                          000120
00135    90    MODE=-1                                            000125
00136    100   RETURN                                             000127
00136    C  SET COLUMN INDICATOR TO 1 FOR YRANGE.                 000127
00137    200   ICOL=1                                             000132
00140          RETURN                                             000133
00140    C  SET COLUMN INDICATOR TO 3 FOR XRANGE.                 000133
00141    300   ICOL=3                                             000137
00142          RETURN                                             000140
00143          END $  DISPLA  ******************************      000176
```

```
SUBROUTINE DTTIM     ENTRY POINT 000006

STORAGE USED  CODE(1) 000010; DATA(0) 000004; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000   000000 INJP$


00100      1*      CDTTIM                                                          000000
00101      2*            SUBROUTINE DTTIM (A)                                      000006
00101      3*      C                                                               000006
00101      4*      C     GET THE CURRENT DATE AND TIME                             000000
00101      5*      C                                                               000000
00103      6*            DIMENSION A(1)                                            000000
00103      7*      C     A(1) = DATE(I)                                            000000
00103      8*      C     A(2) = TIME(I)                                            000000
00104      9*            RETURN                                                    000006
00105     10*            END OF  DTTIM  ******************************             000007
```

```
 1.                                                         *   ASSEMBLER ROUTINE FPCT
 2.                                                         *
 3.                                                         *   FROM FORTRAN-  CALL FPCT(ICPU,IO,IER,IWAIT,ISUP)
 4.                                                         *   WHERE ICPU,IO, ETC. ARE TIME UNITS FOR CURRENT
 5.                                                         *   JOB IN 200 USEC UNITS.
 6.                                                     AXR$
 7.                                           $(1)
 8.      01   000000   72 01 00 00 0 000024   FPCT*     SLJ   SAVEREG
 9.           000001   10 00 00 00 0 000213             LA    A0,(0210,PCTBFR)
10. U         000002   72 11 00 00 0 000000             ER    PCT$
11.           000003   10 00 00 00 0 000031             LA    A0,CPU
12.           000004   01 00 00 13 1 000000             SA    A0,*0,X11
13.           000005   27 00 01 00 0 000214             LX    X1,(1,0)
14.           000006   10 16 01 00 000000               LA    A1,0,,U
15.           000007   10 00 00 01 0 000177             LA    A0,IO,X1
16.           000010   14 00 01 00 0 000014             AA    A1,A0
17.           000011   47 16 01 00 000011               TLEM  X1,9,,U
18.           000012   74 04 00 00 0 000007             J     $-3
19.           000013   01 00 01 13 1 000001             SA    A1,*1,X11
20.           000014   10 00 00 00 0 000211             LA    A0,ER
21.           000015   01 00 00 13 1 000002             SA    A0,*2,X11
22.           000016   10 00 00 00 0 000212             LA    A0,WAIT
23.           000017   01 00 00 13 1 000003             SA    A0,*3,X11
24.           000020   10 00 00 00 0 000016             LA    A0,SUP
25.           000021   01 00 00 13 1 000004             SA    A0,*4,X11
26.           000022   72 01 00 00 0 000030             SLJ   RESTORE
27.           000023   74 04 00 13 0 000006   RETURN    J     6,X11
28.                                                                                    *
29.                                                                                    *
30.           000024   000000000000           SAVEREG   +     0
31.           000025   06 00 01 00 0 000000             SX    X1,SAVEX1
32.           000026   71 12 00 00 0 000001             DS    A0,SAVEA0
33.           000027   74 04 00 00 1 000024             J     *SAVEREG
34.                                                                                    *
35.                                                                                    *
36.           000030   000000000000           RESTORE   +     0
37.           000031   27 00 01 00 0 000000             LX    X1,SAVEX1
38.           000032   71 13 00 00 0 000001             DL    A0,SAVEA0
39.           000033   74 04 00 00 1 000030             J     *RESTORE
40.                                                                                    *
41.                                                                                    *
42.                                           $(0)
43.                                                                                    *
44.      00   000000   000000000000           SAVEX1    +     0
45.           000001   000000000000           SAVEA0    +     0
46.           000002   000000000000           SAVEA1    +     0
47.                                                                                    *
48.                                                                                    *
49.           000003                          PCTBFR    RES   0210
50.                    000000000016           SUP       EQU   PCTBFR+11
51.                    000000000031           CPU       EQU   PCTBFR+22
52.                    000000000177           IO        EQU   PCTBFR+124
53.                    000000000211           ER        EQU   PCTBFR+134
```

```
     54.                  000000000212          WAIT     EQU    PCTBFR+135
     55.                                                 END
               000213     000210 200003
               000214     000001 500000

UNDEFINED SYMBOLS
PCTS

END ASM. ERRORS   NONE
```

```
SUBROUTINE FSHELL     ENTRY POINT 000124

STORAGE USED  CODE(1) 000141; DATA(0) 000031; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000015 105G      0001    000042 121G      0001    000024 20L       0001    000034 40L       0001    000045 50L
 0001    000103 70L       0000 I 000000 I         0000 I 000004 II        0000    000007 INJP$     0000 I 000003 J
 0000 I 000002 K         0000 I 000005 LIMBO     0000 I 000001 M


00100         CFSHELL                                                                000015
00101               SUBROUTINE FSHELL (IARRAY,KEY,N)                                 000015
00101            C  PURPOSE   ORDER AN ARRAY TO HAVE INCREASING MAGNITUDE AND        000015
00101            C                FORM KEY FOR ORDERING RELATED ARRAY.               000015
00101            C  CALL SEQUENCE   IARRAY - N X 1 ARRAY OF VALUES TO BE SORTED      000015
00101            C                       KEY    - N X 1 ARRAY OF KEYS FOR SORTING DEPENDENT   000015
00101            C                                   ARRAY                          000015
00101            C                       N      - NUMBER OF ELEMENTS TO BE SORTED.  000015
00103               DIMENSION IARRAY(1),KEY(1)                                       000015
00104               DO 10 I=1,N                                                      000015
00107            10 KEY(I)=I                                                         000015
00111               M=N                                                              000021
00112            20 M=M/2                                                            000024
00113               IF(M)30,30,40                                                    000026
00116            30 RETURN                                                           000030
00117            40 K=N-M                                                            000034
00120               DO 70 J=1,K                                                      000036
00123               I=J                                                              000042
00124            50 II=I+M                                                           000045
00125               IF(IARRAY(I)-IARRAY(II))70,70,60                                 000057
00130            60 LIMBO=IARRAY(I)                                                  000063
00131               IARRAY(I)=IARRAY(II)                                             000065
00132               IARRAY(II)=LIMBO                                                 000067
00133               LIMBO=KEY(I)                                                     000070
00134               KEY(I)=KEY(II)                                                   000072
00135               KEY(II)=LIMBO                                                    000074
00136               I=I-M                                                            000015
00137               IF(I)70,70,50                                                    000100
00142            70 CONTINUE                                                         000104
00144               GO TO 20                                                         000104
00145               END OF FSHELL  *********************************                 000140
```

```
SUBROUTINE INIT        ENTRY POINT 000055

STORAGE USED  CODE(1) 000061; DATA(0) 000023; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CORDER 000003
 0004    CINT   000001
 0005    CNAMEX 000002
 0006    CNAMER 000002
 0007    CNAMEV 000002
 0010    CNAMEP 000002
 0011    CXIC   000001
 0012    CNTRLS 000004
 0013    CWORKN 000010

EXTERNAL REFERENCES (BLOCK, NAME)

 0014    CONLOD
 0015    PLINIT
 0016    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000004 121G       0001     000036 135G       0012     000000 ANTYPE      0012 R 000003 ERROR      0000 D 000004 HP
 0000 D 000002 HR          0000 D 000000 HS          0000 D 000006 HV          0000 I 000010 I          0000     000014 INJP$
 0004 I 000000 INT         0012     000001 IPRINT     0012     000002 MODE       0013 I 000001 N          0010 D 000000 NAMEP
 0006 D 000000 NAMER       0007 D 000000 NAMEV       0005 D 000000 NAMEX       0013 I 000000 NN         0003     000002 NOP
 0003     000001 NOV       0003 I 000000 NOX         0011 R 000000 XIC


00100        1*      CINIT                                                                                      000000
00100        2*      C       OVERLAY(INIT,1,0)                                                                  000000
00100        3*      C        PROGRAM INIT                                                                      000000
00101        4*             SUBROUTINE INIT                                                                     000000
00101        5*      C  VERSION 1.2                          REVISED  MAY 15 1975                               000000
00101        6*      C  PURPOSE   TO INITIALIZE INTEGRATOR CONTROL,PARAMETER NAME,STATE                          000000
00101        7*      C            NAME, RATE NAME, VARIABLE NAME ARRAYS TO DEFAULT VALUES                       000000
00101        8*      C  DESIGNED BY  J.O. BURROUGHS                       FEB 1974                              000000
00103        9*             COMMON /CORDER/NOX,NOV,NOP/CINT/INT(1)                                              000000
00104       10*             COMMON/CNAMEX/NAMEX(1)/CNAMER/NAMER(1)/CNAMEV/NAMEV(1)/CNAMEP/                      000000
00104       11*            1 NAMEP(1)/CXIC/XIC(1)                                                               000000
00105       12*             COMMON/CNTRLS/ANTYPE,IPRINT,MODE,ERROR(1)                                           000000
00106       13*             COMMON/CWORKN/NN,N(7)                                                               000000
00107       14*             DOUBLE PRECISION    NAMEX,NAMER,NAMEV,NAMEP                                         000000
00110       15*             DOUBLE PRECISION HS/1?HS                                                            000000
00112       16*             DOUBLE PRECISION HR/1?HR                                                            000000
0.          17*             DOUBLE PRECISION HP/1?HP                                                            000000
```

```
00116          DOUBLE PRECISION NV/12HV                                        000006
00116    C  INITIALIZE INT ARRAY                                               000000
00120          DO 10 I=1,NOX                                                   000000
00123          EPROR(I)=.1                                                     000004
00124          XIC(I)=0.                                                       000005
00125       10 INT(I)=1                                                        000006
00125    C  LOAD STATE NAME ARRAY WITH S001,S002,....                          000006
00125    C     CALL CODLOD(NAMEX,NOX,HS)                                       000006
00125    C  LOAD RATE NAME ARRAY WITH R001,R002,....                           000006
00127          CALL CODLOD(NAMER,NOX,HR)                                       000011
00127    C  LOAD PARAMETER NAME ARRAY WITH P001,P002,....                       000011
00127    C     CALL CODLOD(NAMEP,NOP,HP)                                       000011
00127    C  LOAD VARABLE NAME ARRAY WITH V001,V002,....                        000011
00127    C     CALL CODLOD(NAMEV,NOV,HV)                                       000011
00127    C  CALCULATE INDICES FOR WORK  STORAGE                                000011
00130          NN=NOX*NOX+1                                                    000016
00131          N(1)=NN+NOX*NOX                                                 000023
00132          IF(N(1).LT.168)N(1)=168                                         000025
00134          DO 100 I=2,7                                                    000036
00137     100 N(I)=N(I-1)+NOX                                                  000036
00141          CALL PLINIT                                                     000041
00142          RETURN                                                          000043
00143          END @  INIT  *******************************                    000063
```

```
SUBROUTINE INPUTS    ENTRY POINT 000234

STORAGE USED  CODE(1) 000250; DATA(0) 000056; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CCOMM  000023

EXTERNAL REFERENCES (BLOCK, NAME)

 0004   NXTPH
 0005   NUMERC
 0006   PCPKEL
 0007   LCPCH
 0010   NRDU$
 0011   NI03$
 0012   NI02$
 0013   NWDU$
 0014   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000012 100L      0000    000025 121F      0000    000026 141F      0001    000203 175G      0001    000050 200L
 0001     000215 204G      0001    000075 220L      0001    000106 240L      0001    000146 270L      0001    000151 280L
 0001     000155 300L      0001    000174 340L      0001    000206 360L      0001    000221 500L      0001    000223 520L
 0000 D 000014 BLNK       0000 I 000017 I         0003 D 000000 ICOM      0000 D 000000 ICOML     0003 I 000022 INDEX
 0000 I 000027 INDEXS     0000   000043 INJP$     0003 D 000020 IPHRS     0000 I 000021 ISTAT     0000 I 000020 J
 0000 I 000023 K          0000 I 000016 MODE      0000 I 000024 NM        0000 D 000012 VALUE


00100        CINPUTS                                                                          000002
00101              SUBROUTINE INPUTS(A,N,M,NMAX)                                              000002
00101        C  VERSION 1.                         REVISED  MAY 22 1975                       000002
00101        C  PURPOSE   ALLOW FREE FIELD INPUT OF ARRAY DATA                                    000002
00101        C  CALL SEQUENCE    A       --ARRAY TO RECEIVE DATA                                  000002
00101        C                   N       - NUMBER OF ROWS IN ARRAY                                000002
00101        C                   M       - NUMBER OF COLUMNS IN ARRAY                             000002
00101        C                   NMAX    - ROW DIMENSION OF ARRAY A                               000002
00101        C  METHOD   THE FOLLOWING COMMANDS ARE RECOGNIZED                                    000002
00101        C          Z = ZERO ALL ELEMENTS OF ARRAY                                            000002
00101        C          I = SET ALL ELEMENTS OF ARRAY TO 1.E36 (INFINITY)                         000002
00101        C          C = INPUT DATA TO BE GIVEN BY COLUMN                                      000002
00101        C          R = INPUT DATA TO BE GIVEN BY ROW                                         000002
00101        C          D = INPUT DATA TO BE GIVEN BY DIAGONAL                                    000002
00101        C      FOLLOWING THE COL, ROW, DIAG, COMMANDS THE ROW AND COLUMN LOCATION            000002
00101        C      AT WHICH DATA LOADING IS TO START MUST BE GIVEN.  THESE VALUES                000002
00101        C      ARE FOLLOWED BY ELEMENT VALUES.  EACH COMMAND, ROW NO., COL. NO.,             000002
00101        C      OR ELEMENT VALUE MUST BE SEPERATED BY ONE OF THE STANDARD DELIMITE            000002
```

```
00101      19*    C      STANDARD DELIMITERS ARE  THREE OR MORE SPACES; COMMA; EQUAL SIGN;       000002
00101      20*    C      LEFT OR RIGHT PARENTHESIS.                                              000002
00101      21*    C  DESIGNED BY  J.D.BURROUGHS                      MAY 1975                    000002
00103      22*           COMMON/CCOMM/ICOM(8),IPHRS,INDEX                                        000002
00104      23*           DIMENSION ICOML(5),A(1)                                                 000002
00105      24*           DOUBLE PRECISION ICOM,ICOML,IPHRS,VALUE                                 000002
00106      25*           DATA ICOML/'I            I          C                                   000002
00106      26*         1              'R           D          '                                  000002
00110      27*           DOUBLE PRECISION BLNK/12H                                               000002
00110      28*    C --->         SET DEFAULT MODE TO COLUMN INPUT                                000002
00112      29*           MODE=3                                                                  000002
00112      30*    C ======= MODE = MODE OF INPUT INDICATOR. 1 = ZERO ARRAY                       000002
00112      31*    C            2 = SET ARRAY TO 1.E36, 3 = COLUMN INPUT, 4 = ROW INPUT,           000002
00112      32*    C            5 = DIAGONAL INPUT.                                               000002
00113      33*           I=1                                                                     000004
00114      34*           J=1                                                                     000006
00115      35*           ISTAT=2                                                                 000007
00115      36*    C ======== ISTAT = INPUT STATUS INDICATOR.  0 = ROW NO. NEEDED; 1 = COL        000007
00115      37*    C                                           2 = READY FOR DATA VALUES          000007
00116      38*    100    INDEXS=INDEX                                                            000012
00116      39*    C --->         LOCATE NEXT PHRASE                                              000012
00117      40*           CALL NXTPH(ICOM,INDEX,IPHRS)                                            000013
00120      41*           IF(IPHRS.NE.BLNK)GO TO 200                                              000020
00120      42*    C --->         READ NEXT CARD                                                  000020
00122      43*           READ(5,121,END=520)ICOM                                                 000023
00125      44*    121    FORMAT(8A10)                                                            000034
00126      45*    140    WRITE(6,141)ICOM                                                        000034
00131      46*    141    FORMAT(/20H COMMAND CARD ----->,5X,8A10)                                000049
00132      47*           INDEX=1                                                                 000044
00133      48*           GO TO 100                                                               000046
00133      49*    C --->         TEST FOR NUMERIC PHRASE                                         000046
00134      50*    200    CALL NUMERC(IPHRS,$300)                                                 000050
00134      51*    C --->         NUMERIC PHRASE DETECTED                                         000050
00135      52*           CALL FCOPEL(VALUE,IPHRS)                                                000053
00136      53*           IF(ISTAT-1)210,220,240                                                  000057
00141      54*    210    I=VALUE                                                                 000063
00142      55*           ISTAT=1                                                                 000071
00143      56*           GO TO 100                                                               000073
00144      57*    220    J=VALUE                                                                 000075
00145      58*           ISTAT=2                                                                 000102
00146      59*           GO TO 100                                                               000104
00146      60*    C --->         TESTS TO LIMIT INPUT TO GIVEN ROW AND COLUMN DIMENSIONS         000104
00147      61*    240    IF(I.GT.N.OR.J.GT.M)GO TO 100                                           000126
00151      62*           K=I+NMAX*(J-1)                                                          000123
00152      63*           A(K)=VALUE                                                              000130
00152      64*    C --->         INCREASE INDICES DEPENDING ON INPUT MODE                        000130
00153      65*           IF(MODE-4)280,260,270                                                   000134
00156      66*    260    J=J+1                                                                   000141
00157      67*           GO TO 100                                                               000144
00160      68*    270    J=J+1                                                                   000146
00161      69*    280    I=I+1                                                                   000151
00162      70*           GO TO 100                                                               000153
00162      71*    C --->         ALPHA PHRASE DETECTED                                           000153
00163      72*    300    CALL ICMPH(IPHRS,ICOML,5,1,MODE)                                        000155
00164      73*           IF(MODE.EQ.0)GO TO 500                                                  000163
00164      74*    C --->         RESTORE INDEX TO PREVIOUS PHRASE SINCE ALPHA PHRASE IS NOT R    000163
00166      75*           IF(MODE-2)390,380,310                                                   000165
```

```
00171      76*     310   ISTAT=0                                          000171
00172      77*           GO TO 100                                        000172
00172      78*     C --->       ZERO ARRAY MODE                           000172
00173      79*     340   NM=NMAX*M                                        000174
00174      80*           DO 360 I=1,NM                                    000176
00177      81*     360   A(I)=0.                                          000203
00201      82*           GO TO 100                                        000204
00201      83*     C --->       SET ARRAY TO 1.E36 (INFINITY)             000204
00202      84*     380   NM=NMAX*M                                        000206
00203      85*           DO 400 I=1,NM                                    000210
00206      86*     400   A(I)=1.E36                                       000215
00210      87*           GO TO 100                                        000217
00211      88*     500   INDEX=INDEXS                                     000221
00212      89*     520   RETURN                                           000223
00213      90*           END 3  INPUTS  *****************************     000247
```

```
SUBROUTINE INTERP     ENTRY POINT 001477

STORAGE USED  CODE(1) 001510; DATA(0) 000577; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CNTRLS 000004
 0004    COVPLY 000004
 0005    CIO    000003
 0006    CXIC   000001
 0007    CWORK  000002
 0010    CP     000001
 0011    CINT   000001
 0012    CX     000001
 0013    CXIC1  000001
 0014    CXIC2  000001
 0015    CXIC3  000001
 0016    CNAMEX 000002
 0017    CNAMER 000002
 0020    CNAMEV 000002
 0021    CNAMEP 000002
 0022    CUNITX 000001
 0023    CUNITR 000001
 0024    CUNITV 000001
 0025    CUNITP 000001
 0026    CSCALE 000272
 0027    CSHPAR 000026
 0030    COFFER 000001
 0031    CTITC  000001
 0032    CPRINT 000036
 0033    CPPGN  000026
 0034    CPPOV  000033
 0035    CPLOTS 000134
 0036    CCOMM  000023
 0037    CTALNA 000032
 0040    CHAKDI 000002
 0041    CLOCTA 000001
 0042    CTABLE 000001

EXTERNAL REFERENCES (BLOCK, NAME)

 0043    FPCT
 0044    NXTPH
 0045    LCMPH
 0046    NUMERC
 0047    RCDKEL
 0050    NER
 0051    NAMES
 0052    VALUES
 0053    DISPLA
 0054     IONGEN
```

```
0055     TITLE
0056     TAPIN
0057     NWDU$
0060     NIO2$
0061     NRDU$
0062     NIO3$
0063     NCPP2$
0064     NIO1$
0065     ALPR1$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001     000055 100L     0000     000342 101F     0000     000343 105F     0001     000066 111L     0001     000100 120L
0001     000105 140L     0001     000222 160L     0001     000306 165L     0001     000310 168L     0001     000312 170L
0001     000345 176L     0000     000351 177F     0001     000361 178L     0001     000464 180L     0000     000371 181F
0001     000473 200L     0001     000500 210L     0001     000502 215L     0001     000505 220L     0001     000510 230L
0001     000513 232L     0001     000516 234L     0001     000521 236L     0001     000524 240L     0001     000533 245L
0000     000423 247F     0001     000562 250L     0001     000570 260L     0000     000402 2630F    0001     000576 270L
0001     000544 275G     0001     000604 280L     0001     000612 290L     0001     000620 300L     0001     000626 310L
0001     000636 320L     0001     000632 325G     0001     000645 330L     0001     000642 333G     0001     000652 334L
0001     000656 340L     0001     000647 341G     0001     000663 352G     0001     000660 360L     0001     000671 410L
0001     000732 470L     0001     000713 430L     0001     000724 445L     0001     000735 450L     0001     000746 460L
0001     000757 480L     0001     000766 500L     0001     001273 5000L    0001     001275 5005L    0000     000467 501CF
0000     000476 502CF    0000     000504 503CF    0000     000513 505CF    0001     001230 507G     0001     001414 510CL
0000     000524 510IF    0001     000773 520L     0001     001020 530L     0001     001315 540G     0001     001031 541L
0001     001041 542L     0001     001045 545L     0001     001076 546L     0000     000425 547F     0001     001133 55CG
0001     001111 550L     0001     001351 560G     0001     001120 560L     0001     001126 562L     0001     001367 570G
0001     001130 570L     0001     001133 580L     0001     001135 590L     0001     001137 600L     0001     001436 6000L
0000     000541 600IF    0001     001403 601G     0001     001142 610L     0001     001432 614G     0001     001145 620L
0001     001147 630L     0001     001152 640L     0001     001154 650L     0001     001157 660L     0001     001161 670L
0001     001164 680L     0001     001166 690L     0001     001171 700L     0000     000326 71F      0001     001174 710L
0001     001177 720L     0001     000326 80L      0001     001204 800L     0001     001205 820L     0001     000317 90L
0001     001211 910L     0001     001212 920L     0000     000443 921F     0000     000447 922F     0001     001236 940L
0001     001255 945L     0001     001251 950L     0000     000454 951F     0001     001260 960L     0001     001263 980L
0011  R  000000 AINT     0000  R  000313 CPOCL    0000  R  000312 CPSEC    0004  R  000303 CPUSEC   0003  R  000303 ERROR
0000  I  000317 I        0000  D  000170 IBLNK    0000  D  000172 IC       0027  I  000024 ICIND    0000  I  000302 ICLMAX
0000  I  000321 ICOL     0036  D  000000 ICOM     0000  D  000000 ICOML    0000  I  000305 ICPSEC   0035     000002 IDIAG
0000  I  000316 IDSPLY   0000  I  000307 IER      0033  I  000002 IMODE    0036  I  000022 INDEX    0035  I  000000 INDPLT
0035     000003 INDWR    0000     000555 INJPS    0004  I  000000 INST     0003  I  000000 INSTO    0011  I  000000 INT
0000  I  000326 IO       0000  I  000323 IOCAN    0035  I  000002 IOPT     0036  D  000020 IPHRS    0000  I  000303 IPKMAX
0035  I  000072 IPOPT    0003     000001 IPRINT   0000  D  000174 IPROGN   0000  D  000214 IPROGV   0000  I  000304 IPVMAX
0005  I  000000 IREAD    0000  I  000311 ISUB     0000  I  000320 ITNO     0000  I  000310 IWAIT    0005     000001 IWRITE
0000  I  000325 J        0041  I  000073 LOCTAB   0004  I  000002 LOKSIM   0004  I  000001 LONSS    0032  I  000024 LPRT
0040  I  000001 MAXDIM   0000  I  000315 MODE     0021  D  000000 NAMEP    0017  D  000000 NAMER    0020  D  000000 NAMEV
0016  D  000000 NAMEX    0003  I  000314 NAMPRT   0005  D  000166 NOM      0030  I  000002 NOP      0040  I  000000 NOTAB
0033  I  000001 NOV      0030  I  000000 NOX      0026  I  000264 NPLTS    0000  I  000324 NTAB     0020  I  000322 NUNIT
0025  I  000000 NUNTIP   0023  I  000000 NUNITR   0024  I  000000 NUNITV   0022  I  000000 NUNITX   0026  R  000173 NVAR
0010  R  000000 P        0035  D  000040 PLOTID   0033  D  000000 PRONAM   0032  D  000000 PRTNAM   0035  D  000052 PTITLE
0034  R  000000 PVALUE   0026     000303 SCALE    0027  D  000000 SMPAR    0042  R  000000 TABLES   0037  D  000000 TAPNAM
0031  R  000000 TIME     0007  D  000000 WORK     0012  R  000000 X        0006  R  000000 XIC      0013  R  000000 XIC1
0014  R  000000 XIC2     0015  R  000000 XIC3



0C          1*      CINTERP                                                                                            C0060C
```

```
00100  C      OVERLAY(INTERP,2,0)                                                     000000
00100  C       PROGRAM INTERP                                                         000000
00101        SUBROUTINE INTERP                                                        000000
00101  C VERSION 3.1                          REVISED  OCT 11 1976                    000000
00101  C       PURPOSE                                                                000000
00101  C      READS,PRINTS AND INTERPRETS INSTRUCTIONS FROM DATA CARDS                000000
00101  C       CALL SEQUENCE                                                          000000
00101  C           IREAD - READ UNIT NUMBER                                           000000
00101  C           INST  - INSTRUCTION NUMBER                                         000000
00101  C DESIGNED BY  J.D. BURROUGHS                  FEB 1974                        000000
00103        DIMENSION AINT(1)                                                        000000
00104        COMMON /CNTRLS/INSTO ,IPRINT,INDOF,ERROR(1)                              000000
00105        COMMON /COVRLY/INST,LOKSS,LOKSIM,CPUSEC/CIO/IREAD,IWRITE,IDIAG           000000
00106        COMMON/CXIC/XIC(1)/CWORK/WORK(1)/CP/P(1)/CINT/INT(1)/CX/X(1)             000000
00107        COMMON /CXIC1/XIC1(1)/CXIC2/XIC2(1)/CXIC3/XIC3(1)                        000000
00110        COMMON/CNAMEX/NAMEX(1)/CNAMER/NAMER(1)/CNAMEV/NAMEV(1)/CNAMEP/           000000
00110       1 NAMEP(1)                                                                000000
00111        COMMON/CUNITX/NUNITX(1)/CUNITR/NUNITR(1)/CUNITV/NUNITV(1)/CUNITP/        000000
00111       1 NUNITP(1)                                                               000000
00112        COMMON/CSCALE/SCALE(5,4,6),NVAR(5,2,6),NPLTS(6)                          000000
00113        COMMON /CSHPAR/SHPAR(10),ICNVD(2)                                        000000
00114        COMMON/CORDER/NOX,NOV,NOP/CTIME/TIME                                     000000
00115        COMMON/CPRINT/PRTNAM(10),LPRT(10)                                        000000
00116        COMMON /CPRON/PRONAM(8)/CPROV/PVALUE(27)                                 000000
00117        COMMON /CPLOTS/ INOPL1,INOWR,IOPT(30),PLOTID( 5),PTITLE( 8),             000000
00117       *                    IPOPT(10)                                            000000
00123        COMMON/CCOMM/ICOM(6),IPHRS,INDEX                                         000000
00120  C       COMMON/COCDIM/NX,NU,NS,NC,NRS,NRC,IXOC,IUOC,IOCAN,IPOINT(25)           000000
00121        COMMON/CTABNA/TABNAM(1)/CMAXDI/NOTAB,MAXDIM(1)/CLOCTA/LOCTAB(1)          000000
00122        COMMON/CTABLE/TABLE(1)                                                   000000
00123        DOUBLE PRECISION IPHRS,ICOML(59),NAMEX,NAMER,NAMEV,NAMEP,WORK            000000
00124        DOUBLE PRECISION NONE,PRONAM,PRTNAM,ICOM,SHPAR,                          000000
00124       1 TABNAM,IBLNK,IC,IPROGN(8),IPROGV(27),PLOTID,PTITLE                      000000
00125        REAL NVAR                                                                000000
00126        EQUIVALENCE (AINT,INT)                                                   000000
00127        DATA ICLMAX/59/,NONE/12HNONE                                             000000
00132        DATA IPNMAX/8/,IPVMAX/27/                                                000000
00132  C ==================  PROGRAM COMMANDS  =====================                  000000
00135        DATA ICOML /  *DEFINE STA  *,*DEFINE RAT  *,*DEFINE PAR  *,              000000
00135       1*DEFINE VAR  *,*INITIAL CO  *,*PARAMETER   *,*DISPLAY1    *,             000000
00135       2*DISPLAY2    *,*DISPLAY3    *,*DISPLAY4    *,*DISPLAY5    *,             000000
00135       3*DISPLAY6    *,*SCAN1       *,*SCAN2       *,*XIC-X       *,             000000
00135       4*XIC-XIC1    *,*XIC-XIC2    *,*XIC-XIC3    *,*XIC1-XIC    *,             000000
00135       5*XIC2-XIC    *,*XIC3-XIC    *,*ALL STATES  *,*NO STATES   *,             000000
00135       6*INT CONTRO  *,*ERROR CONT  *,*SIMULATE    *,*LINEAR ANA  *,             000000
00135       7*EIGEN SENS  *,*STABILITY   *,*TRANSFER F  *,*STEADY STA  *,             000000
00135       8*ROOT LOCUS  *,*PUNCH X     *,*SH PARAMET  *,*PLOT TABLE  *,             000000
00135       9*PRINT VART  *,*TITLE       *,*PLOT ID     *,*PLOT ON     *,             000000
00135       A*PLOT OFF    *,*SC4020      *,*CALCOMP     *,*RL MANUAL   *,             000000
00135       B*RL AUTO SC  *,*SI MANUAL   *,*SI AUTO SC  *,*SS MANUAL   *,             000000
00135       C*SS AUTO SC  *,*TF MANUAL   *,*TF AUTO SC  *,*MODE        *,             000000
00135       D*NICHOLS     *,*NYQUIST     *,*PRINTER PL  *,*DESIGN O.C  *,             000000
00135       E*O.C. DATA   *,*SAVE O.C.   *,*PLOT ALL T  *,*TABLE       */             000000
00137        DATA IBLNK/12H            /,IC/12HIC                                     000000
00137  C ==================  PROGRAM NAMES  =====================                     000000
00137  C     DATA IPROGN /  *OCPEN       *,*INDEP1      *,*INDEP2      *,             000000
00137  C    1*EIGEN PARA  *,*TF INPUT    *,*TF OUTPUT   *,*SS PARAMET  *,             000000
```

```
00137   C      2*RL PARAMET  */                                                   000000
00137   C ==================  PROGRAM VALUES  ====================                000000
00142         DATA IPROGY/   *START1      *,*STOP1       *,*START2      *,        000000
00142        1*DELTA2      *,*CURVES2     *,*PRINT CONT  *,*PRATE        *,       000000
00142        2*OUTRATE     *,*INT MODE    *,*TINC        *,*TMAX         *,       000000
00142        3*FREQ MAX    *,*FREQ MIN    *,*SS START    *,*SS STOP      *,       000000
00142        4*SS POINTS   *,*SS ITERATI  *,*RL START    *,*RL STOP      *,       000000
00142        5*RL POINTS   *,*REAL MIN    *,*REAL MAX    *,*IMAG MIN     *,       000000
00142        6*IMAG MAX    *,*O.C. MODEL  *,*O.C. ORDER  *,*INITIAL TI   */       000000
00142   C -----  TEST FOR CPU SECOND MEASURE                                      000000
00144         IF(CPUSEC.EQ.0.)GO TO 80                                            000000
00146         CALL FPCT(ICPSEC,IO,IER,IWAIT,ISUP)                                 000001
00147         CPSEC= ICPSEC/5000.                                                 000010
00150         CPDEL=CPSEC-CPUSEC                                                  000015
00151         WRITE(6,71)CPDEL                                                    000017
00154   71    FORMAT(10X,F13.6,*SECONDS WERE REQUIRED FOR THE PREVIOUS*,          000026
00154        1* ANALYSIS*/)                                                       000026
00155   80    MAPPRT=INST                                                         000026
00156         IMODE=PVALUE(9)                                                     000027
00157   90    INSTO=0                                                             000037
00160         IF(INDEX.GT.0.AND.INDEX.LT.81)GO TO 120                             000037
00160   C==============  READ AND WRITE ONE CARD  =========                       000037
00162   100   READ(IREAD,101,END=5000)ICOM                                        000055
00165   101   FORMAT(8A10)                                                        000066
00166    111  WRITE(6,105) ICOM                                                   000066
00171   105   FORMAT(/20H COMMAND CARD ----->,5X,8A10)                            000075
00171   C------>SET CHARACTER SCAN INDEX                                          000075
00172         INDEX=1                                                             000075
00172   C------>LOCATE NEXT PHRASE                                                000075
00173   120   CALL NXTPH(ICOM,INDEX,IPHRS)                                        000100
00173   C------>READ NEXT CARD IF BLANK PHRASE                                    000100
00174   140   IF(IPHRS.EQ.IBLNK) GO TO 100                                        000105
00174   C------>SEARCH COMMAND LIST                                               000105
00176         CALL LCMPH(IPHRS,ICOML,ICLMAX,1,INST)                               000107
00176   C------>COMMAND IDENTIFIED                                                000107
00177         IF(INST.LE.0) GO TO 160                                             000116
00177   C===========  BRANCH TO NEW COMMAND  ==========                           000116
00201         GO TO (200,200,200,205,230,230,210,220,230,232,                     000121
00201        1      234,236,500,500,240,250,260,270,280,290,                      000121
00201        2      300,310,320,200,200,500,500,500,500,500,                      000121
00201        3      500,500,920,330,200,360,550,560,570,580,                      000121
00201        4      590,600,610,620,630,640,650,660,670,680,                      000121
00201        5      690,700,710,720,820,830,950,960,980)           ,INST          000121
00201   C============  SEARCH PROGRAM NAME LIST  ==========                       000121
00202     160 CALL LCMPH(IPHRS,IPROGN,IPNMAX,1,INST)                              000122
00202   C------>PHASE NOT PROGRAM NAME                                            000122
00203         IF(INST.LE.0) GO TO 170                                             000230
00203   C------>GET NEXT PHRASE                                                   000230
00205         CALL NXTPH(ICOM,INDEX,IPHPS)                                        000233
00205   C------>LOAD PROGRAM NAME                                                 000233
00206         PRONAM(INST)=IPHRS                                                  000240
00207         IF(INST.NE.7.AND.INST.NE.8) GO TO 165                               000244
00211         IF(IPHRS.EQ.NONE)PRONAM(INST)=IBLNK                                 000260
00211   C------>GET NEXT PHRASE                                                   000260
00213         CALL NXTPH(ICOM,INDEX,IPHRS)                                        000271
00214         ICINO(INST-6)=0                                                     000276
002           IF(IPHRS.NE.IC) GO TO 168                                           000300
```

```
00215        C----->SET INDICATOR .EQ. 1                                                  000300
00217               ICIND(INST-6)=1                                                       000303
00220          165  INSTO=0                                                               000306
00221               GO TO 120                                                             000306
00222          168  INSTO=0                                                               000310
00223               GO TO 140                                                             000310
00223        C=========  SEARCH PROGRAM VALUE LIST  =========                            000310
00224          170  CALL LCMPH(IPHRS,IPROGV,IPVMAX,1,INST)                                000312
00224        C----->PHRASE NOT PROGRAM VALUE                                              000312
00225               IF(INST.LE.0) GO TO 178                                               000320
00225        C----->GET NEXT PHRASE                                                       000320
00227               CALL TXTPH(ICOM,INDEX,IPHRS)                                          000323
00227        C----->TEST 1ST CHARACTER FOR NUMERIC                                        000323
00230               CALL NUMERC(IPHRS,$176)                                               000330
00230        C----->CONVERT A TO G FORMAT                                                 000330
00231               CALL ACDFFL(IPVALUE(INST),IPHRS)                                      000334
00232               GO TO 165                                                             000343
00233          176  WRITE(6,177) IPROGV(INST),IPHRS                                       000345
00237          177  FORMAT(//10X,15H*** WARNING ***,3X,A10,22HCAN*T BE SET EQUAL TO       000357
00237             1 A10,23H  VALUE MUST BE NUMERIC //)                                    000357
00240               GO TO 168                                                             000357
00240        C----->CHECK FOR OUTSTANDING COMMAND                                         000357
00241          178  IF(INSTO.LE.0) GO TO 180                                              000361
00241        C=========  BRANCH TO OUTSTANDING COMMAND  =========                        000361
00243               GO TO (410,420,430,440,450,460,480,480,480,490,                       000363
00243             1        440,480,500,500,240,250,260,270,280,290,                       000363
00243             2        300,310,320,520,530,500,500,500,500,500,                       000363
00243             3        500,500,500,540,240,545,550,560,500,500,                       000363
00243             4        500,500,500,500,500,530,500,500,500,500,                       000363
00243             5        500,500,500,500,500,830,830,960,980)        ,INSTO             000363
00244        180    WRITE(6,181)IPHRS                                                     000464
00247        181    FORMAT(//15X,34H*** WARNING ***  CAN*T INTERPRET  ,A10//)             000471
00250               GO TO 120                                                             000471
00250        C----->SET INSTO TO INDICATE A NEW OUTSTANDING TASK                          000471
00251        200    INSTO=INST                                                            000473
00252               MODE=-1                                                               000474
00253               GO TO 120                                                             000476
00254        210    IDSPLY=1                                                              000500
00255        215    NPLTS(IDSPLY)=0                                                       000502
00256               GO TO 200                                                             000503
00257        220    IDSPLY=2                                                              000505
00260               GO TO 215                                                             000506
00261        230    IDSPLY=3                                                              000510
00262               GO TO 215                                                             000511
00263          232  IDSPLY = 4                                                            000513
00264               GO TO 215                                                             000514
00265          234  IDSPLY = 5                                                            000516
00266               GO TO 215                                                             000517
00267          236  IDSPLY = 6                                                            000521
00270               GO TO 215                                                             000522
00270        C----->TRANSFER X TO XIC                                                     000522
00271          240  CALL XFR(X,XIC,NOX)                                                   000524
00272               LCNSIM=LCNSS                                                          000530
00273          245  WRITE(6,2630)(I,NAMEX(I),XIC(I),I=1,NOX)                              000533
00303         2630  FORMAT(1H1,NOX,7H/*/*/*/,3X,*INITIAL CONDITIONS/OPERATING POINT*,     000553
00303             1 3X,7H/*/*/*/,//5(I4,1H ,A8,3H = ,G10.4))                              000553
00304               WRITE(6,247)                                                          000563
```

```
00306     173*      247    FORMAT(/////)                                                000560
00307     174*             GO TO 165                                                    000560
00307     175*      C ===========  TRANSFER XIC1 TO XIC  ==========                     000560
00310     176*        250 CALL XFR(XIC1,XIC,NOX)                                        000562
00311     177*             GO TO 245                                                    000566
00311     178*      C ===========  TRANSFER XIC2 TO XIC  ==========                     000566
00312     179*        260 CALL XFR(XIC2,XIC,NOX)                                        000570
00313     180*             GO TO 245                                                    000574
00313     181*      C ===========  TRANSFER XIC3 TO XIC  ==========                     000574
00314     182*        270 CALL XFR(XIC3,XIC,NOX)                                        000576
00315     183*             GO TO 245                                                    000602
00315     184*      C ===========  TRANSFER XIC TO XIC1  ==========                     000602
00316     185*        280 CALL XFR(XIC,XIC1,NOX)                                        000604
00317     186*             GO TO 165                                                    000610
00317     187*      C ===========  TRANSFER XIC TO XIC2  ==========                     000610
00320     188*        290 CALL XFR(XIC,XIC2,NOX)                                        000612
00321     189*             GO TO 165                                                    000616
00321     190*      C ===========  TRANSFER XIC TO XIC3  ==========                     000616
00322     191*        300 CALL XFR(XIC,XIC3,NOX)                                        000620
00323     192*             GO TO 165                                                    000624
00323     193*      C ===========  ALL STATES  ===========                              000624
00324     194*        310 DO 315 I=1,NOX                                                000626
00327     195*        315 INI(I)=1                                                      000632
00331     196*             GO TO 165                                                    000634
00331     197*      C ===========  NO STATES  ===========                               000634
00332     198*        320 DO 325 I=1,NOX                                                000636
00335     199*        325 INI(I)=0                                                      000642
00337     200*             GO TO 165                                                    000643
00337     201*      C----->LOAD SHPAR WITH BLANKS                                       000643
00340     202*        330 DO 335 I=1,10                                                 000647
00343     203*        335 SHPAR(I)=IBLNK                                                000647
00345     204*      338    INSTO=INST                                                   000652
00346     205*             ITNO=1                                                       000652
00347     206*        340 MODE=0                                                        000656
00350     207*             GO TO 120                                                    000656
00350     208*      C-----LOAD PRTNAM WITH BLANKS                                       000656
00351     209*      360    DO 365 I=1,10                                                000661
00354     210*             IPRT(I)=-1                                                   000663
00355     211*      365    PRTNAM(I)=IBLNK                                              000664
00357     212*             GO TO 3**                                                    000667
00357     213*      C----->DEFINE STATES TASK                                           000667
00360     214*      410    CALL NAMES(IPHRS,NAMEX,NUNITX,NOX,ITNO,MODE)                 000671
00361     215*             GO TO 120                                                    000700
00361     216*      C----->DEFINE RATES TASK                                            000700
00362     217*      420    CALL NAMES(IPHRS,NAMER,NUNITR,NOX,ITNO,MODE)                 000702
00363     218*             GO TO 120                                                    000711
00363     219*      C----->DEFINE PARAMETERS TASK                                       000711
00364     220*      430    CALL NAMES(IPHRS,NAMEP,NUNITP,NOP,ITNO,MODE)                 000713
00365     221*             GO TO 120                                                    000722
00365     222*      C----->DEFINE VARIABLES TASK                                        000722
00366     223*      440    CALL NAMES(IPHRS,NAMEV,NUNITV,NOV,ITNO,MODE)                 000724
00367     224*             GO TO 120                                                    000733
00367     225*      C----->INITIAL CONDITIONS TASK                                      000733
00370     226*      450    CALL VALUES(IPHRS,NAMEX,NOX,XIC,ITNO,MODE)                   000735
00371     227*             GO TO 120                                                    000744
00371     228*      C----->PARAMETER INPUT TASK                                         000744
00        229*      460    CALL VALUES(IPHRS,NAMEP,NOP,P,ITNO,MODE)                     000746
```

```
00373    230*        GO TO 120                                                  000755
00373    231*    C----->DISPLAY TASK                                            000755
00374    232*    480   CALL DISPLA(IDSPLY,IPHRS,MODE,ICOL)                      000757
00375    233*          GO TO 120                                                000764
00375    234*    C---->RETURN TO MAIN PROGRAM WITH INST SET TO INDICATED TASK   000764
00376    235*     500  INST=TC                                                  000766
00377    236*          IF(NAMPRT.EQ.1)GO TO 5005                                000766
00401    237*          GO TO 6000                                               000771
00401    238*    C----->LOAD INTEGRATOR CONTROLS                                000771
00402    239*      520 CALL VALUES(IPHRS,NAMEX,NOX,AINT,ITNO,MODE)              000773
00402    240*    C----->CONVERT REAL TO INTEGER                                 000773
00403    241*          IF(MODE.EQ.2) INT(ITNO)=AINT(ITNO)                       001002
00405    242*          GO TO 120                                                001016
00405    243*    C----->LOAD ERROR CONTROLS                                     001016
00406    244*      530 CALL VALUES(IPHRS,NAMEX,NOX,ERROR,ITNO,MODE)             001020
00407    245*          GO TO 120                                                001027
00407    246*    C----->LOAD STABILITY MARGIN PARAMETER NAME                    001027
00410    247*      540 CALL NAMES(IPHRS,SMPAR,NUNIT,10,ITNO,MODE)               001031
00411    248*    542   ITNO=ITNO+1                                              001041
00412    249*          GO TO 340                                                001043
00412    250*    C----- LOAD PRINT VARIABLE NAMES                               001043
00413    251*      545 CALL NAMES(IPHRS,PRTNAM,NUNIT,10,ITNO,MODE)              001045
00413    252*    C-----  DETERMINE I.D. CODES FOR PRINT QUANTITIES              001045
00414    253*          IF(MODE.NE.1)GO TO 542                                   001054
00416    254*          CALL CODGEN(PRTNAM(ITNO),0,LPRT(ITNO),$546)              001057
00417    255*          GO TO 542                                                001074
00420    256*    546   WRITE(6,547)PRTNAM(ITNO)                                 001076
00423    257*    547   FORMAT(//20X,31H*** WARNING ***  CAN'T IDENTIFY,3X,A10   001107
00423    258*         1,'AS A VALID PRINT VARIABLE'//)                          001107
00424    259*          GO TO 542                                                001107
00424    260*    C                                                              001107
00424    261*    C     SET PLOTTING OPTIONS                                     001107
00424    262*    C                                                              001107
00424    263*    C ============  TITLE  ==============                          001107
00425    264*    550   CALL TITLE (ICON,INDEX,PTITLE,80)                        001111
00426    265*          GO TO 562                                                001116
00426    266*    C ============  PLOT ID  ==============                        001116
00427    267*    560   CALL TITLE (ICON,INDEX,PLOTID,48)                        001126
00430    268*    562   INDEX=0                                                  001126
00431    269*          GOTO 90                                                  001126
00431    270*    C ==========  PLOT ON  ===========                             001126
00432    271*      570 INDPLT = 1                                               001130
00432    272*    C     CALL ONSW(1)                                             001130
00433    273*          GO TO 165                                                001131
00433    274*    C ==========  PLOT OFF  ===========                            001131
00434    275*      580 INDPLT = 0                                               001133
00435    276*          GO TO 165                                                001133
00435    277*    C ==========  SC4020  =========                                001133
00436    278*      590 IOPT(29) = 0                                             001135
00437    279*          GO TO 165                                                001135
00437    280*    C ===========  CALCOMP  =============                          001135
00440    281*      600 IOPT(29) = 1                                             001137
00441    282*          GO TO 165                                                001140
00441    283*    C ===========  RL MANUAL SCALES  ==========                    001140
00442    284*      610 IOPT(11) = 1                                             001142
00443    285*          GO TO 165                                                001143
00443    286*    C =========  RL AUTO SCALES  ========                          001143
```

```
00444     287*        620 IPOPT(1) = 0                                                          001145
00445     288*             GO TO 165                                                            001145
00445     289*       C ============  SI MANUAL SCALES  ============                              001145
00446     290*        630 IPOPT(2) = 1                                                          001147
00447     291*             GO TO 165                                                            001150
00447     292*       C ===========  SI AUTO SCALES  ===========                                 001150
00450     293*        640 IPOPT(2) = 0                                                          001152
00451     294*             GO TO 165                                                            001152
00451     295*       C ==========  SS MANUAL SCALES  =========                                  001152
00452     296*        650 IPOPT(3) = 1                                                          001154
00453     297*             GO TO 165                                                            001155
00453     298*       C =========  SS AUTO SCALES  ==========                                    001155
00454     299*        660 IPOPT(3) = 0                                                          001157
00455     300*             GO TO 165                                                            001157
00455     301*       C ==========  TF MANUAL SCALES  =========                                  001157
00456     302*        670 IPOPT(4) = 1                                                          001161
00457     303*             GO TO 165                                                            001162
00457     304*       C ========  TF AUTO SCALES  =========                                      001162
00460     305*        680 IPOPT(4) = 0                                                          001164
00461     306*             GO TO 165                                                            001164
00461     307*       C ==========  BODE  =========                                              001164
00462     308*        690 IPOPT(5) = 1                                                          001166
00463     309*             GO TO 165                                                            001167
00463     310*       C ==========  NICHOLS  ==========                                          001167
00464     311*        700 IPOPT(6) = 1                                                          001171
00465     312*             GO TO 165                                                            001172
00465     313*       C =========  NYQUIST  ========                                             001172
00466     314*        710 IPOPT(7) = 1                                                          001174
00467     315*             GO TO 165                                                            001175
00467     316*       C ==========  PRINTER PLOTS  ==========                                    001175
00470     317*        720 IOPT(30) = 1                                                          001177
00470     318*       C     CALL ONSW(2)                                                         001177
00471     319*             INOPLT=1                                                             001200
00472     320*             GO TO 165                                                            001202
00472     321*       C ------       READ O.C. DATA TASK                                         001202
00472     322*       C800   CALL OCDATA                                                         001202
00473     323*       800   CONTINUE                                                             001204
00474     324*             GO TO 165                                                            001204
00474     325*       C =============  DESIGN O.C. TASK  =============                           001204
00474     326*       C ---        TEST THAT MODEL IS DIMENSIONED FOR O.C. DESIGN                001204
00475     327*       820   IF(IOCAN.EQ.2)GO TO 500                                              001205
00475     328*       C      WRITE(6,825)                                                        001205
00475     329*       C825  FORMAT(//15X,15H*** WARNING ***,3X,'WORK SPACE WAS NOT PROVIDED IN     001205
00475     330*       C    1 MODEL FOR OPTIMAL CONTROLLER DESIGN'//)                             001205
00477     331*             GO TO 165                                                            001207
00477     332*       C ==============  SAVE O.C. TASK =============                             001207
00477     333*       C900   CALL OCSAVE                                                         001207
00500     334*       900   CONTINUE                                                             001211
00501     335*             GO TO 165                                                            001211
00501     336*       C ===============  PUNCH X TASK  ================                           001211
00502     337*       920   WRITE(3,921)                                                         001212
00504     338*       921   FORMAT('INITIAL CONDITIONS')                                         001216
00505     339*             WRITE(3,922)(NAMEX(I),X(I),I=1,NOX)                                  001216
00514     340*       922   FORMAT(4(A7,'=',G10.4,','))                                          001234
00515     341*             GO TO 165                                                            001234
0         342*       C ==============  PLOT TABLES TASK  ============                           001234
0         343*       940   CALL ECMPN(IPHRS,TABNAM,NOTAB,I,NTAB)                                001236
```

```
00517            IF(NTAB.LE.0)GO TO 950                                                001244
00517      C-------      CALL TABLE PLOTTING ROUTINE                                   001244
00517      C995   CALL PLOTAB(NTAB)                                                    001244
00521      945    CONTINUE                                                             001250
00521      C      CALL ONSW(1)                                                         001250
00522             GO TO 170                                                            001250
00523      950    WRITE(6,951)IPHRS                                                    001251
00526      951    FORMAT(//15X,15H*** WARNING ***,3X,A10,' IS NOT VALID TABLE NAME'    001256
00526           1//)                                                                   001256
00527             GO TO 120                                                            001256
00527      C ===============   PLOT ALL TABLES   TASK   =============                  001256
00530      960    NTAB=-1                                                              001260
00531             GO TO 945                                                            001261
00531      C ==============   TABLE    TASK    ================                        001261
00532      980    CONTINUE                                                             001263
00533             CALL TABIN(TABLES,TABNAM,MAXDIM,LOCTAB,NOTAB)                        001263
00534             GO TO 111                                                            001271
00534      C------END OF FILE ENCOUNTERED                                              001271
00535      5000   IFST=-1                                                              001273
00536      5005   WRITE(6,5010)(I,NAMEX(I),I=1,NOX)                                    001275
00545       5010 FORMAT(//1H1,50X,11HSTATE NAMES//10(I4,1X,A8))                        001323
00546             WRITE(6,5020)(I,NAMER(I),I=1,NOX)                                    001323
00555       5020 FORMAT(//50X,10HRATE NAMES//10(I4,1X,A8))                             001341
00556             WRITE(6,5030)(I,NAMEV(I),I=1,NOV)                                    001341
00565       5030 FORMAT(//50X,14HVARIABLE NAMES//10(I4,1X,A8))                         001357
00566             WRITE(6,5050)(I,NAMEP(I),P(I),I=1,NOP)                               001357
00576      5050   FORMAT(//49X,'PARAMETER VALUES'//5(I4,1X,A8,                         001376
00576           12H= ,G11.5))                                                          001376
00576      C =============   SCAN FOR UNINITIALIZED PARAMETERS                         001376
00577             J=0                                                                  001376
00603             DO 5100 I=1,NOP                                                      001403
00603             IF(P(I).NE..99999)GO TO 5100                                         001403
00605             J=J+1                                                                001405
00606             WORK(J)=NAMEP(I)                                                     001411
00607      5100   CONTINUE                                                             001416
00611             IF(J.GT.0)WRITE(6,5101)(WORK(I),I=1,J)                               001416
00620      5101   FORMAT(/////15X,15H*** WARNING ***,15X,'UNINITIALIZED PARAMETERS'    001436
00620           1 //10(3X,A8,2X))                                                      001436
00621      6000   CONTINUE                                                             001436
00622             TIME=PVALUE(27)                                                      001436
00623             WRITE(6,6001)                                                        001437
00625      6001   FORMAT(1H1)                                                          001444
00625      C -----          GET CURRENT CPU TIME                                       001444
00626             CALL FPCTI(ICPSEC,IO,ICA,IWAIT,ISUB)                                 001444
00627             CPUSEC= ICPSEC/5000.                                                 001453
00630             RETURN                                                               001460
00631             END # INTERP  *****************************                          001507
```

```
SUBROUTINE LPRINT     ENTRY POINT 000255


STORAGE USED  CODE(1) 000271; DATA(0) 000133; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CNAMEX  000002
 0004    CNAMER  000002
 0005    CNAMEV  000002
 0006    CNAMEP  000002
 0007    CX      000001
 0010    CXDOT   000001
 0011    CV      000001
 0012    CP      000001
 0013    CORDER  000003
 0014    CPRINT  000036
 0015    CTIES   000003


EXTERNAL REFERENCES (BLOCK, NAME)

 0016    VAPOUT
 0017    NWDU$
 0020    NIO1$
 0021    NIO2$
 0022    NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000    000014 11F        0001    000024 117G       0000    000027 13F        0001    000043 130G       0001    000071 143G
 0000    000036 15F        0001    000137 160G       0000    000046 17F        0001    000155 172G       0001    000221 211G
 0001    000234 221G       0001    000152 300L       0001    000174 310L       0001    000175 320L       0000    000056 343F
 0000    000063 363F       0000 I 000012 I           0000    000071 INJP$      0015    000000 JSTART     0015 I 000001 KINIT
 0014 I 000024 LPRT        0000 I 000013 N           0006 D 000000 NAMEP       0004 D 000000 NAMER       0005 D 000000 NAMEV
 0003 D 000000 NAMEX       0013 I 000002 NOP         0013 I 000001 NOV         0013 I 000000 NOX         0000 R 000000 OUTPUT
 0012 R 000000 P           0014 D 000000 PRINAM      0015    000002 TP         0011 R 000000 V           0007 R 000000 X
 0010 R 000000 XDOT


 00100       1*     CLPPINT                                                                          000000
 00101       2*          SUBROUTINE LPRINT(IPRINT,TIME)                                              000000
 00101       3*     C  VERSION 3.                         REVISED  MAY 5 1976                        000000
 00101       4*     C  PURPOSE   PROVIDE GENERAL LINEPRINTER OUTPUTS.                                000000
 00101       5*     C  CALL SEQUENCE   IPRINT - PRINT CONTROL VARIABLE.                              000000
 00101       6*     C                  TIME   - CURRENT TIME.                                        000000
 00101       7*     C   IPRINT VALUE    QUANTITIES PRINTED                                           000000
 00101       8*     C      0 OR 1       STATES, RATES, AND TIME                                      000000
 001         9*     C        2          STATES, RATES, VARIABLES, AND TIME                           000000
 001        10*     C        3          STATES, RATES, VARIABLES, (PARAMETERS AT TIME=0 ONLY)        000000
```

```
00101     C          4          STATES, RATES, VARIABLES, PARAMETERS, AND TIME          000000
00101     C          5          VARIABLES SPECIFIED IN  PRTNAM ARRAY                    000000
00103           COMMON/CNAMEX/NAMEX(1)/CNAMER/NAMER(1)/CNAMEV/NAMEV(1)                  000000
00104           COMMON/CNAMEP/NAMEP(1)                                                  000000
00105           COMMON/CX/X(1)/CXDOT/XDOT(1)/CV/V(1)/CP/P(1)                            000000
00106           COMMON/CORDER/NOX,NOV,NOP                                               000000
00107           COMMON/CPRINT/PRTNAM(10),LPRT(10)/CDIFS/JSTART,KINIT,IP                 000000
00110           DOUBLE PRECISION PRTNAM,NAMEX,NAMER,NAMEV,NAMEP                         000000
00111           DIMENSION OUTPUT(10)                                                    000000
00111     C ------>       TEST FOR LIST OPTION                                          000000
00112           IF(IPRINT.EQ.5)GO TO 300                                                000000
00112     C ------>       PRINT STATES                                                  000000
00114           WRITE(6,11)TIME,(I,NAMEX(I),X(I),I=1,NOX)                               000022
00125     11    FORMAT(//10X,'TIME = ',G10.4,30X,'STATES'/5(I4,1X,A8,2H= ,G11.5))       000033
00125     C ------>       PRINT RATES.                                                  000033
00126           WRITE(6,13)(I,NAMER(I),XDOT(I),I=1,NOX)                                 000033
00136     13    FORMAT(/57X,'RATES'/5(I4,1X,A8,2H= ,G11.5))                             000052
00136     C ------>       TEST FOR VARIABLES OPTION.                                    000052
00137           IF(IPRINT.LE.1)RETURN                                                   000052
00137     C ------>       PRINT VARIABLES.                                              000052
00141           WRITE(6,15)(I,NAMEV(I),V(I),I=1,NOV)                                    000061
00151     15    FORMAT(/57X,'VARIABLES'/5(I4,1X,A8,2H= ,G11.5))                         000100
00151     C ------         TEST FOR PARAMETER PRINT OPTIONS                             000100
00152           IF(IPRINT.LE.2)RETURN                                                   000100
00154           IF(IPRINT.LE.3.AND.TIME.GT.0.)RETURN                                    000107
00156           WRITE(6,17)(I,NAMEP(I),P(I),I=1,NOP)                                    000127
00166     17    FORMAT(/57X,'PARAMETERS'/5(I4,1X,A8,2H= ,G11.5))                        000146
00167           RETURN                                                                  000146
00167     C ------>       SCAN CODES AND GET CURRENT VALUES.                            000146
00170     300   N=0                                                                     000152
00171           DO 320 I=1,10                                                           000155
00171     C ------>       TEST FOR LAST VARIABLE                                        000155
00174           IF(LPRT(I).EQ.-1)GO TO 310                                              000155
00176           CALL VAROUT(LPRT(I),OUTPUT(I))                                          000165
00177           N=I                                                                     000170
00200           GO TO 320                                                               000172
00201     310   OUTPUT(I)=0.                                                            000174
00202     320   CONTINUE                                                                000176
00202     C ------>       TEST FOR NO LIST QUANTITIES IDENTIFIED                        000176
00204           IF(N.LT.1)RETURN                                                        000176
00204     C ------>       PRINT HEADING WHEN KINIT = 0.                                 000176
00206           IF(KINIT.EQ.0)WRITE(6,343)(PRTNAM(I),I=1,N)                             000205
00215     343   FORMAT(/4X,'TIME',3X,10(3X,A8,1X))                                      000224
00215     C ------>       PRINT LIST VALUES.                                            000224
00216     360   WRITE(6,363)TIME,(OUTPUT(I),I=1,N)                                      000224
00225     363   FORMAT(1X,G10.4,10G12.5)                                                000237
00226           RETURN                                                                  000237
00227           END OF LPRINT  ****************************                             000270
```

```
SUBROUTINE NAMES     ENTRY POINT 000106


STORAGE USED  CODE(1) 000133; DATA(0) 000034; BLANK COMMON(2) 000000


EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NUMERC
 0004   RCPFCL
 0005   NWDU$
 0006   NIO2$
 0007   NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000011 100L      0001   000043 110L      0001   000050 120L      0000   000001 121F      0001   000062 200L
 0000 R 000030 FLNO      0000   000025 INJP$



00100      1*     CNAMES                                                                              000003
00101      2*           SUBROUTINE NAMES(IPHRS,NAME,NUNIT,NO,ITNO,MODE)                               000003
00101      3*     C  PURPOSE   LOADS ALPHANUMERIC NAMES OF QUANTITIES IDENTIFIED BY                   000003
00101      4*     C            DEFINE STATEMENTS.                                                     000003
00101      5*     C  CALL SEQUENCE   IPHRS  = ARRAY CONTAINING NEXT PHRASE TO BE EXAMINED.            000003
00101      6*     C                  NAME   = ARRAY TO BE LOADED WITH NAMES OF                        000003
00101      7*     C                             DEFINED QUANTITIES.                                   000003
00101      8*     C                  NUNIT  = ARRAY, TO BE LOADED WITH UNIT NAMES                     000003
00101      9*     C                             OF DEFINED QUANTITIES.                                000003
00101     10*     C                  NO     = NUMBER OF DEFINED QUANTITIES.                           000003
00101     11*     C                  ITNO   = POSITION OF GIVEN QUANTITY IN NAME ARRAY.               000003
00101     12*     C                  MODE   = MODE OF OPERATION INDICATOR.                            000003
00101     13*     C                      MODE = 0 WHEN ITNO HAS BEEN LOADED.                          000003
00101     14*     C                      MODE = 1 WHEN NAME HAS BEEN LOADED.                          000003
00103     15*           DOUBLE PRECISION NAME(NO),IPHRS                                               000003
00104     16*           REAL NUNIT(NO)                                                                000003
00104     17*     C  TEST FOR NUMERIC FIRST CHARACTER.                                                000003
00105     18*           CALL NUMERC(IPHRS,$100)                                                       000003
00106     19*           GO TO 200                                                                     000007
00106     20*     C  TEST THAT ITNO IS WITHIN ALLOWABLE RANGE.                                        000007
00107     21*     100   IF(ITNO.LT.1.OR.ITNO.GT.NO) GO TO 120                                         000011
00111     22*           IF(MODE.NE.0) GO TO 110                                                       000026
00111     23*     C  LOAD NAME                                                                        000026
00113     24*           NAME(ITNO)=IPHRS                                                              000033
00114     25*           MODE=1                                                                        000035
00115     26*           RETURN                                                                        000037
00115     27*     C  LOAD UNITS NAME.    (ALL NAMES WILL BE PUT IN WORD 1 FOR NOW.)                   000037
00116     28*     110   NUNIT(1    )=IPHRS                                                            000043
00         29*           RETURN                                                                        000044
00         30*     120   WRITE(6,121) ITNO,IPHRS                                                       000050
```

```
00124      31*     121   FORMAT(15X,15H*** WARNING ***,I4,40H EXCEEDS THE ALLOWABLE INDEX R                    000056
00124      32*          1ANGE FOR  ,A10,34H THIS QUANTITY WILL NOT BE DEFINED)                                 000056
00125      33*           RETURN                                                                                000056
00125      34*     C  CONVERT IPHRS TO I FORMAT.                                                               000056
00126      35*     200   CALL FCOREL(FLNO,IPHRS)                                                               000062
00127      36*           ITNO=FLNO                                                                             000065
00130      37*           MODE=0                                                                                000074
00131      38*           RETURN                                                                                000075
00132      39*           END 4  NAMES  ****************************                                            000132
```

```
MAIN PROGRAM   NONSIM

STORAGE USED  CODE(1) 000176; DATA(0) 000025; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    COVRLY 000004
 0004    CPPOV  000033
 0005    CPRGN  000020
 0006    CSMPAR 000026
 0007    CORDER 000003
 0010    CUNKN  000010
 0011    CSIMUL 000022
 0012    CPLOTS 000067

EXTERNAL REFERENCES (BLOCK, NAME)

 0013    DATASN
 0014    INIT
 0015    INTERP
 0016    SWITCH
 0017    NINTP$
 0020    NSTOP$
 0021    NLPP2$
 0022    NWDUS
 0023    NIO2$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000010 100L        0001     000173 1000L       0001     000115 200L        0001     000116 300L        0000     000000 301F
 0001     000127 310L        0001     000165 400L        0001     000166 420L        0001     000167 500L        0001     000170 600L
 0001     000171 700L        0001     000172 800L        0004     000010 AMODE       0003     000003 CPUSEC      0004     000004 CURVES
 0004     000003 DELTA2      0005 D 000000 DEPEN       0005 D 000006 ESPAR       0004     000013 FMAX        0004     000014 FMIN
 0006     000024 ICIND       0011 0 000012 IPCNT       0004     000027 IMMAX       0004     000026 IMMIN       0015 D 000014 INDEP
 0005 D 000002 INDEP1      0005 D 000004 INDEP2      0011     000010 INDEX       0011     000005 INDMAX      0012     000000 INDPLT
 0012     000001 INOVR       0003 I 000000 INST        0012     000002 IOPT        0011 I 000002 IOUT        0011     000011 IPLOT
 0012     000065 IPOPT       0011 I 000001 IPRATE      0011 I 000000 IPRIN       0003 I 000002 LOKSIM      0003 I 000001 LOKSS
 0010     000001 N           0005 D 000010 NINPUT      0010     000000 NN          0007     000002 NOP         0005 D 000012 NOUT
 0007     000001 NOV         0007     000000 NOX         0011     000004 NPIMAX      0011     000003 NPTS        0007     000003 NSIM
 0004     000030 OCMOD       0004     000031 OCOPD       0004 R 000007 OUTRAT      0012     000040 PLOTID      0004 R 000006 PRATE
 0004 R 000005 PRINT       0012     000045 PTITLE      0004     000025 RLMAX       0004     000024 RLMIN       0005 R 000016 PLPAR
 0004     000023 RPOINT      0004     000021 RSTART      0004     000022 RSTOP       0006 D 000000 SHPAR       0004     000017 SPOINT
 0004     000020 SSLIM       0004 R 000011 TINC        0011 R 000006 TINC2       0004 R 000012 TMAX        0011 R 000007 TMAX2
 0004     000032 TZERO       0004     000001 XMAX1       0004     000000 XMIN1       0004     000002 XMIN2       0004     000015 XSTART
 0004     000016 XSTOP


00100       1*      CNONSIM                                                                          000000
00100       2*      C       OVERLAY(NONSIM,0,0)                                                      000000
00100       3*      C       PROGRAM NONSIM(INPUT=100,OUTPUT=200,TAPE5=INPUT,TAPE6=OUTPUT,            000000
00100     C     1 PUNCH=100,TAPE3=PUNCH,TAPE30,TAPE25)                                  000000
00100     C  VERSION 3.                    REVISED  APRIL 30 1976                       000000
00100     C  PURPOSE    MAIN PROGRAM FOR THE BATCH VERSION OF NONSIM.                   000000
00101           COMMON/COVRLY/INST,LOKSS,LOKSIM,CPUSEC                                  000000
00103           COMMON/CPROV/XMIN1,XMAX1,XMIN2,DELTA2,CURVES,PRINT,PRATE,OUTRAT,        000001
00103          1 AMODE,TINC,TMAX,FMAX,FMIN,XSTART,XSTOP,SPOINT,SSLIM,RSTART,RSTOP,      000001
00103          2.RPOINT,RLMIN,RLMAX,IMMIN,IMMAX,OCMOD,OCOPD,TZERO                       000001
00104           COMMON/CPRON/DEPEN,INDEP1,INDEP2,FSPAR,NINPUT,NOUT,INDEP,RLPAR          000001
00105           COMMON/CSMPAR/SMPAR(10),ICIND(2)                                        000001
00106           COMMON/CORDER/NSIM,NOX,NOP/CWORKN/NN,N(7)                               000001
00107           COMMON/CSIMUL/IPRIN,IPRATE,IOUT,NPTS,NPTMAX,INDMAX,TINC2,TMAX2,         000001
00107          1 INDEX,IPLOT,IDENT(4)                                                   000001
00110           DOUBLE PRECISION IDENT,SMPAR,DEPEN,INDEP1,INDEP2,ESPAR,                 000001
00110          1  NINPUT,NOUT,INDEP,RLPAR                                               000001
00111           EQUIVALENCE(NSIM,NOX)                                                   000001
00112           COMMON /CPLOTS/ INDPLT,INOUR,IOPT(30),PLOTID( 5),PTITLE( 8),            000001
00112          *                IPOPT(10)                                               000001
00112     C  CALL USER FURNISHED INPUT ROUTINE.                                         000001
00113           CALL DATAIN                                                             000001
00113     C       CALL OVERLAY(6HINIT,1,0)                                              000001
00114           CALL INIT                                                               000003
00115           INST=1                                                                  000005
00115     C  INTERPRETATION ROUTINE TO READ INSTRUCTIONS.                               000005
00115     C100    CALL OVERLAY(6HINTERP,2,0,6HRECALL)                                   000005
00116     100   CALL INTERP                                                             000010
00117           IF(INST.LE.0) STOP                                                      000011
00117     C  BRANCH TO SPECIFIED ANALYSIS.                                              000011
00121           GO TO (100,100,100,100,100,100,100,100,100,100,                         000016
00121          1        100,100,200,200,100,100,100,100,100,100,                        000016
00121          2        100,100,100,100,100,300,400,400,500,600,                        000016
00121          3        700,800,420,100,800,100,100,100,100,100,                        000016
00121          4        100,100,100,100,100,100,100,100,100,100,                        000016
00121          5        100,100,100,100,1000,100,100)            ,INST                  000016
00121     C  GENERAL FUNCTION OF ONE INDEPENDENT VARIABLE.                              000016
00121     C200    CALL OVERLAY(6HGFBTCH,3,0)                                            000016
00122     200   CONTINUE                                                                000115
00123           GO TO 100                                                               000115
00124       300 IF(LOKSIM.EQ.1) GO TO 310                                               000116
00126           WRITE(6,301)                                                            000120
00130     301   FORMAT(//15X,15H*** WARNING ***,5X,*SIMULATION WILL NOT BE RUN DUE      000125
00130          1 TO FAILURE TO REACH VALID STEADY STATE*//)                             000125
00131           GO TO 100                                                               000125
00132       310 IPRIN=PRINT                                                             000127
00133           IPRATE=PRATE                                                            000135
00134           IOUT=OUTRAT                                                             000144
00135           TINC2=TINC                                                              000153
00136           TMAX2=TMAX                                                              000155
00136     C       CALL OVERLAY(6HSIBTCH,4,0)                                            000155
00137           CALL SIBTCH                                                             000157
00140           LOKSS=1                                                                 000161
00141           GO TO 100                                                               000163
00141     C400    CALL OVERLAY(6HLABTCH,5,0)                                            000163
00142     400   CONTINUE                                                                000165
00143           GO TO 100                                                               000165
00144     420   CONTINUE                                                                000166
00145           GO TO 100                                                               000166
00145     C500    CALL OVERLAY(6HSMBTCH,6,0)                                            000166
```

```
00146      61*     500   CONTINUE                                                                          000167
00147      62*           GO TO 100                                                                         000167
00147      63*     C600   CALL OVERLAY(6HTFBTCH,7,0)                                                       000167
00150      64*     600   CONTINUE                                                                          000170
00151      65*           GO TO 100                                                                         000170
00151      66*     C700   CALL OVERLAY(6HSSBTCH,10B,0)                                                     000170
00152      67*     700   CONTINUE                                                                          000171
00153      68*           GO TO 100                                                                         000171
00153      69*     C800   CALL OVERLAY(6HRLBTCH,11B,0)                                                     000171
00154      70*     800   CONTINUE                                                                          000172
00155      71*           GO TO 100                                                                         000172
00155      72*     C ====================  DESIGN O.C.  ==================                                  000172
00155      73*     C -------------------------  GENERATE LINEAR SYSTEM MODEL  -- PROGRAM O                  000172
00155      74*     C1000  CALL OVERLAY(6HNONSIM,12B,0)                                                     000172
00156      75*     1000  CONTINUE                                                                          000173
00156      76*     C -------------------------  GENERATE OPTIMAL CONTROLLER  -- PROGRAM OC                  000173
00156      77*     C      CALL OVERLAY(6HNONSIM,13B,0)                                                     000173
00157      78*           GO TO 100                                                                         000173
00160      79*           END $  NONSIM  ****************************                                       000175
```

```
SUBROUTINE PLINIT    ENTRY POINT 000046

STORAGE USED  CODE(1) 000052; DATA(0) 000026; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CPLOTS 000104
 0004   CSCALE 000366

EXTERNAL REFERENCES (BLOCK, NAME)

 0005   NREW$
 0006   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000007 157L      0001    000013 124L      0001    000020 132L      0001    000025 134L      0001    000033 144L
 0000 D 000012 BLNK      0000 D 000000 DFLTID    0000 I 000014 I         0003 I 000000 INDPLT    0003 I 000001 INDWR
 0000   000022 INJP$     0003 I 000002 IOPT      0003 I 000072 IPOPT     0004 I 000360 NPLTS     0004 D 000170 NVAR
 0003 D 000040 PLOTID    0003 D 000052 PTITLE    0004    000000 SCALE



00100      1*      CPLINIT                                                                          000000
00101      2*            SUBROUTINE PLINIT                                                          000000
00101      3*      C                                                                                000000
00101      4*      C     INITIALIZE FOR PLOTTING                                                    000000
00101      5*      C                                                                                000000
00103      6*            COMMON /CPLOTS/ INDPLT,INDWR,IOPT(30),PLOTID( 5),PTITLE( 8),               000000
00103      7*           +                IPOPT(10)                                                  000000
00104      8*            COMMON /CSCALE/ SCALE(5,4,6),NVAR(5,2,6),NPLTS(6)                          000000
00105      9*            DOUBLE PRECISION PLOTID,PTITLE,DFLTID,NVAR,BLNK                            000000
00106     10*            DIMENSION DFLTID(5)                                                        000000
00107     11*            DATA BLNK /12H                                                             000000
00111     12*            DATA DFLTID /60H ANALYSIS PLOTS                                            000000
00111     13*           +                                                                           000000
00111     14*      C                                                                                000000
00113     15*             REWIND 26                                                                 000000
00114     16*            INDPLT = 0                                                                 000002
00115     17*            INDWR = 0                                                                  000003
00116     18*            DO 10 I=1,30                                                               000007
00121     19*         10 IOPT(I) = 0                                                                000007
00123     20*            DO 20 I=1,5                                                                000013
00126     21*         20 PLOTID(I) = DFLTID(I)                                                      000013
00130     22*            DO 30 I=1,8                                                                000020
00133     23*         30 PTITLE(I) = BLNK                                                           000020
00135     24*            DO 40 I=1,10                                                               000025
00141     25*         40 IPOPT(I) = 0                                                               000025
00142     26*            IPOPT(5) = 1                                                               000026
```

```
00143                DO 50 I=1,6                                                     000033
00146             50 NPLTS(I) = 0                                                    000033
00150                RETURN                                                          000034
00151                END $  PLINIT  ************************                         000051
```

```
SUBROUTINE SETIN     ENTRY POINT 000164

STORAGE USED  CODE(1) 000173; DATA(0) 000020; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CX      000001
 0004    CXDOT   000001
 0005    CV      000001
 0006    CP      000001
 0007    CXTC    000001
 0010    CTIME   000001

EXTERNAL REFERENCES (BLOCK, NAME)

 0011    EQMO
 0012    VARSET
 0013    RATSET
 0014    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000014 10L        0001    000036 20L        0001    000072 30L        0001    000126 40L        0001    000004 5L
 0000     000011 INJP$      0000 I 000000 J           0006 R 000000 P           0010 R 000000 TIME        0005 R 000000 V
 0003 R 000000 X            0004 R 000000 XDOT        0007    000000 XTC


00100       1*      CSETIN                                                                                    000000
00101       2*            SUBROUTINE SETIN(I,VAR)                                                             000000
00101       3*      C  PURPOSE   TO MODIFY THE CURRENT VALUE OF A STATE VARIABLE,PARAMETER,                   000000
00101       4*      C            ETC. AND TO EXECUTE THE MODEL TO OBSERVE THE RESULTS OF                      000000
00101       5*      C            THE MODIFICATION.                                                            000000
00101       6*      C  CALL SEQUENCE   I   = IDENTIFICATION CODE.                                             000000
00101       7*      C                  VAR = NEW NUMERIC VALUE OF QUANTITY IDENTIFIED BY COD                  000000
00103       8*            COMMON/CX/X(1)/CXDOT/XDOT(1)/CV/V(1)/CP/P(1)/CXTC/XTC(1)                            000000
00104       9*            COMMON/CTIME/TIME                                                                   000000
00104      10*      C  TEST FOR TIME                                                                          000000
00105      11*            IF(I.NE.0) GO TO 10                                                                 000000
00107      12*            TIME=VAR                                                                            000001
00110      13*      5     CALL EQMO(0.,0.,0)                                                                  000004
00111      14*            RETURN                                                                              000010
00111      15*      C  TEST FOR STATES                                                                        000010
00112      16*      10    IF(I.LT.1.OR.I.GT.1000000) GO TO 20                                                 000014
00114      17*            X(I)=VAR                                                                            000031
00115      18*            GO TO 5                                                                             000034
00115      19*      C  TEST FOR VARIABLES                                                                     000034
00116      20*      20    IF(I.LE.3000000.OR.I.GT.4000000) GO TO 30                                           000036
00120      21*            J=I-3000000                                                                         000053
```

```
00121              V(J)=VAR                                           000057
00122              CALL VARSET(0.,0.,J)                               000061
00123              RETURN                                             000066
00123        C  TEST FOR RATES                                        000066
00124        30    IF(I.LE.1000000.OR.I.GT.2000000) GO TO 40          000072
00126              J=I-1000000                                        000107
00127              XPOT(J)=VAR                                        000113
00130              CALL RATSET(0.,0.,J)                               000115
00131              RETURN                                             000122
00131        C  TEST FOR PARAMETERS                                   000122
00132        40    IF(I.LE.4000000.OR.I.GT.5000000) RETURN            000126
00134              P(I-4000000)=VAR                                   000146
00135              GO TO 5                                            000152
00136              END *  SETIN  ************************             000172
```

```
SUBROUTINE SHELLX    ENTRY POINT 000110


STORAGE USED  CODE(1) 000122; DATA(0) 000023; BLANK COMMON(2) 000000


EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000013 10L        0001   000022 106G       0001   000034 116G      0001   000042 40L        0001   000051 50L
 0001   000061 60L        0000 I 000001 I          0000 I 000000 IFIRST    0000 I 000003 IK         0000   000005 INJP$
 0000 R 000002 TEMP


00100       1*        CSHELLX                                                                            000010
00101       2*              SUBROUTINE SHELLX(DARRAY,KEY,N)                                              000010
00101       3*        C  PURPOSE   REORDER ELEMENTS OF SINGLE DIMENSION ARRAY                            000010
00101       4*        C            BASED ON THE INDEX ARRAY KEY.                                         000010
00101       5*        C  CALL SEQUENCE   DARRAY - ARRAY TO BE REORDERED                                  000010
00101       6*        C                  KEY    - INDEX ARRAY                                            000010
00101       7*        C                  N      - NUMBER OF ELEMENTS IN ARRAY                            000010
00103       8*              DIMENSION DARRAY(1),KEY(1)                                                   000010
00104       9*              IFIRST=1                                                                     000010
00105      10*           10 DO 20 I=IFIRST,N                                                             000013
00110      11*              IF(KEY(I))20,20,40                                                           000022
00113      12*           20 CONTINUE                                                                     000034
00115      13*              DO 30 I=1,N                                                                  000034
00120      14*           30 KEY(I)=-KEY(I)                                                               000034
00122      15*              RETURN                                                                       000036
00123      16*           40 IFIRST=I                                                                     000042
00124      17*              TEMP=DARRAY(I)                                                               000043
00125      18*              GO TO 60                                                                     000047
00126      19*           50 DARRAY(I)=DARRAY(IK)                                                         000051
00127      20*              I=IK                                                                         000056
00130      21*           60 IK=KEY(I)                                                                    000061
00131      22*              KEY(I)=-IK                                                                   000064
00132      23*              IF(IK-IFIRST)50,70,50                                                        000065
00135      24*           70 DARRAY(I)=TEMP                                                               000067
00136      25*              GO TO 10                                                                     000073
00137      26*              END OF SHELLX  *************************                                     000121
```

```
SUBROUTINE SIBTCH    ENTRY POINT 000521

STORAGE USED  CODE(1) 000535; DATA(0) 000300; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    COTOFR 000003
 0004    CPROV  000036
 0005    CX     000051
 0006    CXPOT  000001
 0007    CINT   000001
 0010    CXIC   000001
 0011    CNTKLS 000054
 0012    CSIMUL 000022
 0013    CPRINT 000036
 0014    CDIFS  000003
 0015    CTIME  000001
 0016    CMMESS 000032
 0017    CWORK  000001
 0020    CSCALE 000366
 0021    CPLOTS 000104

EXTERNAL REFERENCES (BLOCK, NAME)

 0022    CTTIM
 0023    CODGEN
 0024    EG40
 0025    LPRINT
 0026    VAROUT
 0027    STEPI
 0030    NWDU$
 0031    NI03$
 0032    NI02$
 0033    NPRU$
 0034    NREW$
 0035    NRDU$
 0036    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000123 10L       0001     000341 125L      0001     000014 126G      0001     000355 130L      0001     000065 162G
 0001     000112 171G      0001     000137 177G      0001     000133 20L       0001     000500 200L      0001     000163 214G
 0001     000221 235G      0001     000252 254G      0000     000207 2710F     0001     000321 277G      0000     000246 2941F
 0001     000146 30L       0001     000404 331G      0001     000454 356G      0001     000157 40L       0001     000172 50L
 0001     000203 60L       0001     000210 65L       0001     000212 67L       0001     000272 77L       0001     000272 80L
 0004 R 000010 AMODE     0011     000000 ANTYPE    0017     000000 DSPLY     0004     000000 DUM1      0004     000011 DUM2
 0011     000003 ERROR     0000 I 000174 I         0000 I 000172 IBLK      0000 I 000173 ICOUNT    0012 I 000012 IPCNT
 0016 I 000001 IERR      0016     000000 IFATAL    0000 I 000203 IMAX      0011 I 000002 IMODE     0012 I 000010 INDEX
 0012     000005 INOMAX    0021 I 000000 INOPLT    0021 I 000001 INPWR     0000     000256 INJP$     0007     000000 INT
 0021 I 000002 IOPT      0012 I 000002 IOUT      0012 I 000011 IPLOT     0021 I 000072 IPORT     0006 I 000017 IPOUT
```

```
0012 I 000001 IPRATE    0000 I 000176 IPRCNT    0012 I 000003 IPRIN     0011   000001 IPRINT    0000 I 000177 ISET
0000 I 000000 IVAR      0000 I 000074 IVRCOD    0000 I 000204 IV1       0000 I 000205 IV2       0000 I 000202 J
0014 I 000000 JSTART    0000 I 000206 K         0014 I 000001 KINIT     0013   000024 LPRT      0000 I 000200 NCODES
0000 I 000201 NDISP     0003   000002 NOP       0003   000001 NOV       0020 I 000360 NPLTS     0012   000004 NPTMAX
0012   000003 NPTS      0003 I 000000 NSIM      0020 D 000170 NVAR      0021 D 000040 PLOTID    0013 D 000000 PRTNAM
0021 D 000052 PTITLE    0020 R 000000 SCALE     0015 R 000000 TIME      0012 R 000006 TINC      0012 R 000007 TMAX
0014   000002 TP        0000 R 000133 VRCOD     0005 R 000000 X         0006 R 000000 XDOT      0010 R 000000 XIC
```

```
00100      CSIBTCH                                                             000000
00100      C       OVERLAY(SIBTCH,4,0)                                         000000
00100      C       PROGRAM SIBTCH                                              000000
00101            SUBROUTINE SIBTCH                                             000000
00101      C  VERSION 3.1                      REVISED  OCT 7 1976             000000
00103            COMMON/CORDER/NSIM,NOV,NOP                                    000000
00104            COMMON/CPROV/DUM1(8),AMODE,DUM2(15)                           000000
00105            COMMON /CX/X(1)/CXDOT/XDOT(1)/CINT/INT(1)/CXIC/XIC(1)         000000
00106            COMMON /CNTRLS/ANTYPE,IPRINT,IMODF,ERROR(1)                   000000
00107            COMMON /CSIMUL/IPRIN,IPRATE,IOUT,NPTS,NPTMAX,INDMAX,TINC,TMAX 000000
00107           1 ,INDEX,IPLOT,IDENT(4)                                        000000
00110            COMMON/CPRINT/PRINAM(10),LPRT(10)/CDIFS/JSTART,KINIT,TP       000000
00111            DOUBLE PRECISION IDENT,PTITLE,PLOTID,PRTNAM,NVAR              000000
00112            COMMON/CTIME/TIME/ERMESS/IFATAL,IERR                          000000
00113            COMMON /CWORK/ DSPLY(1)                                       000000
00114            COMMON /CSCALE/ SCALE(5,4,6),NVAR( 5,2,6),NPLTS(6)            000000
00115            COMMON /CPLOTS/ INOPLT,INDWR,IOPT(30),PLOTID( 5),PTITLE( 8),  000000
00115           +                   IPOPT(10)                                  000000
00116            DIMENSION IVAR(5,2,6),IVRCOD(31)                              000000
00117            DIMENSION VRCOD(31)                                           000000
00120            DATA IBLK /6H                                                 000000
00122            IMODE=AMODE                                                   000000
00123            IPLOT=1                                                       000006
00124            ICOUNT=0                                                      000010
00125            DO 5 I=1,31                                                   000014
00130          5 VRCOD(I) = 0.0                                                000014
00132            IOPT(1) = IBLK                                                000015
00133            IOPT(4) = IBLK                                                000017
00134            CALL CTTIM (IOPT(3))                                          000020
00135            IOPT(2)=IOPT(2)+1                                             000023
00136            WRITE(6,2700)IPRATE,IOUT,IMODE,TINC,TMAX,PTITLE,(IOPT(I),I=2,4)  000026
00147       2700 FORMAT(45X,41H/*/*/*/   SIMULATION ANALYSIS   /*/*/*/ //20X,  000046
00147           1 11HPRINT RATE=,I3,3X,13HDISPLAY RATE=,I3,3X, 5HMODE=,        000046
00147           2 I3,3X, 5HTINC=, G12.5,3X, 5HTMAX=, G12.5//26X,8A10//         000046
00147           3 15X,'CASE NO.',I4,27X,2A10/)                                 000046
00150            IPOUT=IOUT*IPRATE                                             000046
00151            IPRCNT=0                                                      000051
00152            INDEX=1                                                       000052
00153            ISET=0                                                        000054
00154            IF ( INOPLT .EQ. 0 ) GO TO 67                                 000055
00154      C                                                                   000055
00154      C     FIND CODE NUMBERS FOR THIS SIMULATION.                        000055
00154      C                                                                   000055
00154      C          NVAR - PARAMETER NAMES FOR EACH PLOT                     000055
00154      C          IVAR - POINTERS INTO IVRCOD FOR EACH PARAMETER           000055
00154      C        IVRCOD - UNIQUE CODE NUMBERS USED IN THIS SIMULATION       000055
```

```
00154      47*     C                                                              000055
00156      48*             NCODES = 1                                             000057
00157      49*             NDISP = 0                                              000060
00160      50*             IVRCOD(1) = 0                                          000061
00161      51*             DO 65 J=1,6                                            000065
00164      52*             IMAX = NPCTS(J)                                        000075
00165      53*             IF ( IMAX .EQ. 0 ) GO TO 65                            000077
00167      54*             NDISP = J                                              000101
00170      55*             DO 60 I=1,IMAX                                         000103
00173      56*             CALL CODGEN (NVAR( I,1,J),0,IV1,$10)                   000112
00174      57*          10 CALL CODGEN (NVAR( I,2,J),0,IV2,$20)                   000123
00175      58*          20 CONTINUE                                               000131
00176      59*             DO 30 K=1,NCODES                                       000133
00201      60*             IF ( IVRCOD(K) .NE. IV1 ) GO TO 30                     000137
00203      61*             IVAR(I,1,J) = K                                        000142
00204      62*             GO TO 40                                               000144
00205      63*          30 CONTINUE                                               000147
00207      64*             NCODES = NCODES + 1                                    000147
00210      65*             IVRCOD(NCODES) = IV1                                   000153
00211      66*             IVAR(I,1,J) = NCODES                                   000155
00212      67*          40 CONTINUE                                               000157
00213      68*             DO 50 K=1,NCODES                                       000157
00216      69*             IF ( IVRCOD(K) .NE. IV2 ) GO TO 50                     000163
00220      70*             IVAR(I,2,J) = K                                        000166
00221      71*             GO TO 60                                               000170
00222      72*          50 CONTINUE                                               000173
00224      73*             NCODES = NCODES + 1                                    000173
00225      74*             IVRCOD(NCODES) = IV2                                   000177
00226      75*             IVAR(I,2,J) = NCODES                                   000201
00227      76*          60 CONTINUE                                               000212
00231      77*          65 CONTINUE                                               000212
00233      78*          67 CONTINUE                                               000212
00233      79*     C                                                              000212
00233      80*     C       INITIALIZE FOR SIMULATION                              000212
00233      81*     C                                                              000212
00234      82*             DO 70 I=1,NSIM                                         000212
00237      83*             X(I)=XIC(I)                                            000221
00240      84*          70 XDOT(I)=0.                                             000222
00242      85*             JSTART=0                                               000224
00243      86*             KINIT=0                                                000225
00243      87*     C ----------  TURN ON ERROR MESSAGES IN MODEL                  000225
00244      88*             IERR=1                                                 000226
00245      89*             CALL FGMO(TIME,TMAX,ISCT)                              000230
00245      90*     C ----------  TURN OFF ERROR MESSAGES IN MODEL                 000230
00246      91*             IERR=0                                                 000235
00247      92*             IF(IPRIN.GT.0)CALL LPRINT(IPRIN,TIME)                  000236
00251      93*             IF ( INOFLT .EQ. 0 ) GO TO 77                          000245
00253      94*             DO 75 K=1,NCODES                                       000252
00256      95*             CALL VAROUT (IVRCOD(K),VRCOD(K))                       000252
00257      96*          75 CONTINUE                                               000262
00261      97*             WRITE (75) VRCOD                                       000262
00264      98*          77 CONTINUE                                               000272
00264      99*     C                                                              000272
00264     100*     C       INCREMENT COUNTERS AND SAVE PARAMETER VALUES IF REQUIRED.  000272
00264     101*     C                                                              000272
00265     102*          80 CALL STEP1(TIME,TINC)                                  000272
00266     103*             ICOUNT=ICOUNT+1                                        000275
```

```
00267          IPRCNT=IPRCNT+1                                                  000300
00270          IF(ICOUNT.LT.IOUT) GO TO 130                                     000303
00272          ICOUNT=0                                                         000306
00273          IF ( INDPLT .EQ. 0 ) GO TO 105                                   000307
00275          INDEX=INDEX+1                                                    000313
00276          DO 100 K=1,NCODES                                                000321
00301          CALL VAROUT (IVRCOD(K),VRCOD(K))                                 000321
00302      100 CONTINUE                                                         000331
00304          WRITE (25) VRCOD                                                 000331
00307      105 CONTINUE                                                         000341
00310          IF(IPRCNT.LT.IPOUT) GO TO 130                                    000341
00312          IPRCNT=0                                                         000344
00313          IF ( IPRIN .GT. 0 ) CALL LPRINT (IPRIN,TIME)                     000345
00313    C     GO TO 130                                                        000345
00313    C 110 CONTINUE                                                         000345
00313    C     WRITE (6,120)                                                    000345
00313    C 120 FORMAT (///1H ,10(1H*),7HWARNING, 10(1H*),66H THE NUMBER OF DATA P  000345
00313    C    +OINTS EXCEEDS AVAILABLE STORAGE FOR ONE RUN. ,20(1H*)//          000345
00313    C    +7PX,40H THE DATA TO THIS POINT WILL BE PLOTTED.///)              000345
00313    C     INDEX = INDEX - 1                                                000345
00313    C     GO TO 140                                                        000345
00315      130 CONTINUE                                                         000355
00316          IF(TIME.LT.TMAX -.00001) GO TO 80                                000355
00320      140 CONTINUE                                                         000361
00321          WRITE(6,2941)                                                    000361
00323    2941  FORMAT(/////)                                                    000366
00323    C                                                                      000366
00323    C     WRITE PLOT DATA.                                                 000366
00323    C                                                                      000366
00324          IF ( INDPLT .EQ. 0 ) GO TO 200                                   000366
00326          IOPT(1) = 2                                                      000374
00327          IOPT(5) = NDISP                                                  000376
00330          DO 150 I=1,NDISP                                                 000404
00333          IOPT(5+I) = NPLTS(I)                                             000404
00334      150 CONTINUE                                                         000406
00336          IOPT(12) = INDEX                                                 000406
00337          IOPT(13) = NCODES                                                000410
00340          IOPT(14) = IPOPT(2)                                              000412
00341          IOPT(15) = 0                                                     000414
00342          WRITE (26) IOPT,PLOTID,PTITLE                                    000415
00347          WRITE (26) SCALE,NVAR,IVAR                                       000432
00354          REWIND 25                                                        000447
00355          DO 180 I=1,INDEX                                                 000454
00360          READ (25) VRCOD                                                  000454
00363          WRITE (26) VRCOD                                                 000462
00366      180 CONTINUE                                                         000472
00370          REWIND 26                                                        000472
00371          INDWR = 1                                                        000475
00372      200 CONTINUE                                                         000534
00373          END & SIBTCH  ************************                           000534
```

```
SUBROUTINE STEP1     ENTRY POINT 000162

STORAGE USED  CODE(1) 000206; DATA(0) 000035; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CORDER 000003
 0004    CX     000001
 0005    CXDOT  000001
 0006    CNTRLS 000004
 0007    CWORK  000001
 0010    CWORKN 000010
 0011    CTIME  000001
 0012    CSIMUL 000010
 0013    CNAMEX 000001
 0014    COIFS  000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0015    EOMD
 0016    NERR2$
 0017    NWDU$
 0020    NI01$
 0021    NI02$
 0022    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000014 100L      0000    000002 IDIF      0001    000036 117G      0001    000116 144G      0001    000143 154G
 0001    000076 500L      0001    000072 535L      0001    000100 600L      0007 R  000000 A         0000 R  000001 DT2
 0012    000005 DUM       0006    000003 ERROR     0000 I  000000 I         0000    000074 INJP$     0006    000000 INSTR
 0006    000001 IPRINT    0014    000000 JSTART    0014 I  000001 KINIT     0006 I  000002 MODE      0003    000002 NOP
 0003    000001 NOV       0013 I  000000 NAMEX     0010    000000 NN        0003 I  000000 NSIM      0010    000001 N1
 0010    000002 N2        0010    000003 N3        0010    000004 N4        0010    000005 N5        0010    000006 N6
 0010 I  000007 N7        0011    000000 TIM       0012 R  000007 TMAX      0014 R  000002 TP        0004 R  000000 X
 0005 R  000000 XDOT


00100       1*      CSTEP1                                                                          000000
00101       2*            SUBROUTINE STEP1(TIME,TINC)                                               000000
00101       3*      C  VERSION 4.                       REVISED  SEPT 17 1976                       000000
00101       4*      C       PURPOSE    CALL INTEGRATION SCHEME SELECTED BY MODE VARIABLE            000000
00101       5*      C  CALL SEQUENCE    TIME   - CURRENT TIME                                       000000
00101       6*      C                   TINC   - TIME STEP TO BE TAKEN TO NEXT REPORT INTERVAL      000000
00101       7*      C  DESIGNED BY  J.D. BURROUGHS                 FEB 1974                         000000
00103       8*            COMMON/CORDER/NSIM,NOV,NOP/CX/X(1)/CXDOT/XDOT(1)                          000000
00104       9*            COMMON/CNTRLS/INSTR,IPRINT,MODE,ERROR(1)                                  000000
00105      10*           COMMON/CWORK/A(1)/CWORKN/NN,N1,N2,N3,N4,N5,N6,N7                           000000
```

```
00106     11*           COMMON/CTIME/TIM/CSIMUL/DUM(3),TMAX/CNAMEX/NAMEX(1)                   000000
00107     12*           COMMON/COEFS/JSTART,KINIT,TP                                          000000
00107     13*     C ===============  SET NEXT PRINT TIME                                      000000
00110     14*          TP=TIME+TINC                                                           000000
00111     15*          GO TO 600                                                              000002
00112  *DIAGNOSTIC*  CONTROL CAN NEVER REACH THE NEXT STATEMENT
00112     16*       5   GO TO(500,100,600),MODE                                               000003
00112     17*     C =======================  NRKVS INTEGRATOR  ====================           000003
00112     18*     C 100   CALL OVERLAY(5HNRKVS,4,1,6HRECALL)                                  000003
00113     19*     100   CONTINUE                                                              000014
00114     20*           IF(TIME.GT.TMAX) WRITE(6,101) (I,NAMEX(I),A(N7+I-1),I=1,NSIM)         000014
00125     21*      101   FORMAT(//47X,*INTEGRATOR STEP SIZE LIMITING COUNTS*/                 000045
00125     22*         1 5(I4,1X,A8,2H= ,G11.5))                                               000045
00126     23*           KINIT= 1                                                              000045
00127     24*          IF(MODE.EQ.1.AND.TIME.LT.TP-.00001)GO TO 505                           000047
00131     25*           RETURN                                                                000064
00131     26*     C -----          START GEAR INTEGRATION WITH INITIAL CALL TO NRKVS          000064
00132     27*      500   IF(KINIT.EQ.0) GO TO 100                                             000070
00132     28*     C =====================  GEAR INTEGRATOR  ====================             000070
00132     29*     C505    CALL OVERLAY(6HNOHSIM,4,2,6HRECALL)                                 000070
00134     30*     505   CONTINUE                                                              000072
00135     31*           IF(KINIT.NE.0) RETURN                                                 000072
00137     32*          GO TO 100                                                              000076
00137     33*     C =================  FIXED STEP INTEGRATOR  ==================              000076
00140     34*       600 DT2=TINC*.5                                                           000100
00141     35*          KINIT=1                                                                000102
00142     36*          CALL EOMO(TIME,TINC,0)                                                 000104
00143     37*          DO 601 I=1,NSIM                                                        000111
00146     38*          A(I)=X(I)+DT2*XDOT(I)                                                  000116
00147     39*     601  X(I)=X(I)+TINC*XDOT(I)                                                 000121
00151     40*          TIME=TIME+TINC                                                         000126
00152     41*          CALL EOMO(TIME,TINC,0)                                                 000131
00153     42*          DO 602 I=1,NSIM                                                        000136
00156     43*     602  X(I)=A(I)+DT2*XDOT(I)                                                  000143
00160     44*          RETURN                                                                 000147
00161     45*          END Q  STEPI  ************************                                 000205
```

```
SUBROUTINE TABIN     ENTRY POINT 000666

STORAGE USED  CODE(1) 000710; DATA(0) 000245; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CIO     000003
 0004    CCOMM   000023

EXTERNAL REFERENCES (BLOCK, NAME)

 0005    NXTPW
 0006    NUMERC
 0007    BCDREL
 0010    LCMPH
 0011    STRMOV
 0012    KONSTR
 0013    NLPUT
 0014    NIO2$
 0015    NRPUT
 0016    NIO3$
 0017    NERR2$
 0020    NIO1$
 0021    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000021 10L        0001    000032 100L       0001    000130 1000L      0000    000024 101F       0001    000171 1100L
 0000    000025 1101F      0001    000045 120L       0001    000066 122L       0001    000074 130L       0001    000105 140L
 0001    000115 160L       0000    000023 22F        0001    000200 2000L      0001    000434 2656       0001    000465 3006
 0001    000223 3000L      0001    000247 3020L      0001    000252 3030L      0000    000245 3041F      0001    000517 3136
 0001    000636 3536       0001    000306 4000L      0001    000314 4020L      0001    000330 4040L      0001    000332 5000L
 0001    000344 6000L      0001    000374 6020L      0000    000135 6021F      0000    000111 6031F      0000    000121 6041F
 0001    000440 6100L      0000    000124 6101F      0000    000134 6121F      0001    000503 6140L      0001    000535 6200L
 0000    000142 6201F      0001    000553 6300L      0000    000161 6301F      0001    000561 6400L      0001    000577 6500L
 0001    000601 6520L      0000    000200 6531F      0001    000646 6540L      0000 0  000003 BLNK       0004 0  000000 CARD
 0000 C  000005 HTABLE     0000 I  000022 I          0003    000002 IDIAG      0004 I  000022 INDEX      0000 I  000012 INDEXS
 0000    000220 INJP$      0003 I  000000 IREAD      0000 I  000020 ITAG       0003 I  000001 IWRITE     0012 I  000000 KONSTR
 0000 I  000021 LIN        0000 I  000015 LOK        0000 I  000000 MAX        0000 I  000011 MODE       0000 I  000014 NTAB
 0000 I  000007 NX         0000 I  000016 NXMAX      0000 I  000013 NZ         0000 I  000017 NZMAX      0004 R  000023 PHRS
 0000 0  000001 TABN       0000 R  000013 VALUE


00100        1*      CTABIN                                                                                 000016
00101        2*            SUBROUTINE TABIN(ITAP,TAPNAM,MAXDIM,LOCTAB,NOTAB)                               000016
00101        3*      C  VERSION 2.1                            REVISED  JAN 7 1976                         000016
00101        4*      C  PURPOSE   PROVIDE FREE FIELD READ OF TABULAR DATA FOR EITHER                       000016
00101        5*      C                  SINGLE OR DOUBLE TABLE LOOKUPS                                     000016
```

```
00101   C   CALL SEQUENCE  TAB    - ARRAY INTO WHICH DATA WILL BE LOADED          000016
00101   C                  TABNAM - ARRAY OF ALLOWABLE TABLE NAMES                000016
00101   C                  MAXDIM - ARRAY OF MAX. DIMENSIONS FOR TABLES           000016
00101   C                  LOCTAB - ARRAY OF TABLE LOCATIONS IN ARRAY TAB         000016
00101   C                  NOTAB  - NO. OF TABLES IN MODEL                        000016
00101   C   METHOD   TABLE DESCRIPTION IS IN THE FOLLOWING FORMAT                 000016
00101   C        CARD 1   TABLE  *TABLE NAME*   NX   NZ                           000016
00101   C        CARD 2*  SECONDARY INDEPENDENT VARIABLE TABLE                    000016
00101   C        CARD 3*  PRIMARY INDEPENDENT VARIABLE TABLE                      000016
00101   C        CARD 4*  DEPENDENT VARIABLE TABLE                                000016
00101   C           *USE AS MANY CARDS AS DESIRED. MUST START TABLE WITH          000016
00101   C            A NEW CARD. MUST GIVE NZ,NX, AND NX*NZ POINTS RESPECTIVELY   000016
00101   C            IN EACH TABLE.                                               000016
00101   C        NX - NO. OF POINTS IN PRIMARY IND. VAR. TABLE                    000016
00101   C        NZ - NO. OF POINTS IN SECONDARY IND. VAR. TABLE                  000016
00101   C        DATA ITEMS ARE FREE FIELD. ITEMS MUST BE SEPERATED BY EITHER     000016
00101   C        2 OR MORE BLANKS,COMMA,EQUALS, OR LEFT OR RIGHT PARENTHESIS      000016
00103         COMMON/CIO/IREAD,IWRITE,IDIAG                                       000016
00104         COMMON/CCOMM/CARD(8),PHRS,INDEX                                     000016
00105         DIMENSION TAB(1),TABNAM(1),MAXDIM(1),LOCTAB(1)                      000016
00106         DOUBLE PRECISION TABNAM,CARD,PHRS,TABN                              000016
00107         DOUBLE PRECISION BLNK/12H                                           000016
00111         DOUBLE PRECISION MTABLE/12HTABLE                                    000016
00113         TABN=BLNK                                                           000016
00114   10    NX=0                                                                000021
00115         NZ=0                                                                000021
00116         MODE=0                                                              000022
00117         WRITE(IWRITE,20)                                                    000023
00121   20    FORMAT(////)                                                        000030
00122         GO TO 122                                                           000030
00122   C --->      READ DATA CARD                                                000030
00123   100   READ(IREAD,101,END=6520)CARD                                        000032
00126   101   FORMAT(8A10)                                                        000042
00126   C --->      SET CHARACTER INDEX                                           000042
00127         INDEX=1                                                             000042
00127   C --->      LOCATE NEXT PHRASE                                            000042
00130   120   INDEXS=INDEX                                                        000045
00131         CALL NXTPH(CARD,INDEX,PHRS)                                         000046
00131   C --->      TEST FOR BLANK PHRASE                                         000046
00132         IF(PHRS.EQ.BLNK)GO TO 100                                           000053
00132   C --->      TEST OPERATING MODE                                           000053
00134         IF(MODE.NE.0)GO TO 130                                              000056
00134   C ========================= MODE=0 == CHECK FOR *TABLE*                   000056
00136         CALL NUMERC(PHRS,$122)                                              000060
00137         GO TO 100                                                           000064
00140   122   IF(PHRS.NE.MTABLE)GO TO 6530                                        000066
00142         MODE=1                                                              000070
00143         GO TO 120                                                           000072
00144   130   IF(MODE.GT.1)GO TO 140                                              000074
00144   C ========================= MODE=1 == STORE TABLE NAME                    000074
00146         CALL NUMERC(PHRS,$160)                                              000077
00146   C --->      NUMERIC PHRS                                                  000077
00147         GO TO 6540                                                          000103
00147   C --->      CONVERT BCD TO REAL                                           000103
00147   C ========================= MODE .GT. 1                                   000103
00150   140   CALL NUMERC(PHRS,$6200)                                             000105
00151         CALL BCDFTL(VALUE,PHRS)                                             000110
```

```
00151    C --->       BRANCH TO TASK INDICATED BY MODE                           000110
00152    100   GO TO(1000,2000,3000,4000,5000,6000),MODE                         000115
00152    C ========================= MODE=1 == STORE TABLE NAME                  000115
00153    1000  CALL LCMPH(PHRS,TABNAM,NOTAB,1,NTAB)                              000130
00154          IF(NTAB.LE.0)GO TO 1100                                           000136
00154    C --->       STARTING LOCATION FOR TABLE DATA                           000136
00156          LOK=LOCTAB(NTAB)                                                  000141
00156    C --->       LAST WORD ADDRESS FOR TABLE DATA                           000141
00157          MAX=MAXDIM(NTAB)+LOK-1                                            000145
00160          CALL STRMOV(PHRS,1,6,TAB(LOK),1)                                  000152
00161          MODE=2                                                            000165
00162          GO TO 120                                                         000167
00163    1100  WRITE(IWRITE,1101)PHRS                                            000171
00166    1101  FORMAT(17H *** WARNING *** ,A10,                                  000176
00166         1'IS NOT A VALID TABLE NAME FOR THIS MODEL.  DATA WILL BE IGNORED')  000176
00167          GO TO 10                                                          000176
00167    C ======================== MODE=2 == STORE NO. POINTS IN PRI. IND. TABL  000176
00170    2000  TAB(LOK+1)=VALUE                                                  000200
00171          NXMAX=VALUE                                                       000203
00172          MODE=3                                                            000212
00173          CALL NXTFH(CARD,INDEX,PHRS)                                       000214
00174          GO TO 140                                                         000221
00174    C ======================== MODE=3 == STORE NO. POINTS IN SEC. IND. TABLE 000221
00175    3000  LOK=LOK+2                                                         000223
00176          TAB(LOK)=VALUE                                                    000225
00177          NZMAX=VALUE                                                       000231
00177    C --->       TEST IF THERE IS A SECONDAY INDEPENDENT VAR. TABLE         000231
00200          IF(NZMAX.LE.1) GO TO 3020                                         000240
00202          MODE=4                                                            000243
00203          GO TO 3040                                                        000245
00204    3020  MODE=5                                                            000247
00205          NZMAX=0                                                           000250
00206    3040  ITAB=LOK                                                          000252
00207          IF(LOK+NXMAX+NZMAX+NXMAX*MAX0(1,NZMAX).LE.MAX)GO TO 100          000253
00211          LIM=MAXDIM(NTAB)-3                                                000267
00212          WRITE(IWRITE,3041)NXMAX,NZMAX,LIM                                 000274
00217    3041  FORMAT(17H *** WARNING *** ,I4,' PRIMARY AND ',I4,                000304
00217         1' SECONDARY INDEPENDENT VARIABLE POINTS EXCEEDS THE ',            000304
00217         2I4,' WORD STORAGE LIMIT FOR THE'/21X,                             000304
00217         3'FOLLOWING TABLE.    SOME DATA WILL BE LOST.'/)                   000304
00220          GO TO 100                                                         000304
00220    C ================= MODE=4 ==     STORE SECONDARY IND. VAR. TABLE       000304
00221    4000  NZ=NZ+1                                                           000306
00222          IF(NZ.GT.NZMAX)GO TO 4040                                         000310
00224    4020  ITAB=ITAB+1                                                       000314
00224    C --->       LIMIT DATA TO TAB ARRAY MAX.                               000314
00225          IF(ITAB.LE.MAX)TAB(ITAB)=VALUE                                    000316
00227          GO TO 120                                                         000326
00230    4040  MODE=5                                                            000330
00230    C ======================= MODE=5 == STORE PRI. IND. VAR. TABLE          000330
00231    5000  NX=NX+1                                                           000332
00232          IF(NX.LE.NXMAX)GO TO 4020                                         000334
00234          MODE=6                                                            000337
00235          NX=0                                                              000341
00236          NZ=0                                                              000342
00236    C ======================= MODE=6 == STORE DEPENDENT VAR. TABLE          000342
00237    6000  ITAB=ITAB+1                                                       000344
```

```
00240               IF(ITAB.LE.MAXITAB)TAB(ITAB)=VALUE                              000346
00242               NX=NX+1                                                         000356
00243               IF(NX.LT.NXMAX)GO TO 120                                        000361
00245               NX=0                                                            000364
00246               NZ=NZ+1                                                         000365
00247               IF(NZ.LT.NZMAX)GO TO 120                                        000370
00247         C --->       TABLE READ IN COMPLETE - PRINT                           000370
00251         6020  WRITE(IWRITE,6021)TAB(LOK-2)                                    000374
00254         6021  FORMAT(20X,'TABLE ',A6/)                                        000404
00254         C --->       TEST IF THERE ARE 2 INDEPENDENT VAR.                     000404
00255               IF(NZMAX.LE.0)GO TO 6100                                        000404
00257               WRITE(IWRITE,6031)                                              000407
00261         6031  FORMAT(10X,'SECONDARY INDEPENDENT VARIABLE TABLE'/)             000414
00262               ITAB=LOK                                                        000414
00263               WRITE(IWRITE,6041)(TAB(ITAB+I),I=1,NZMAX)                       000420
00271         6041  FORMAT(10(3X,G10.4))                                            000440
00272         6100  WRITE(IWRITE,6101)                                              000440
00274         6101  FORMAT(/10X,'PRIMARY INDEPENDENT VARIABLE TABLE'/)              000444
00275               ITAB=LOK+NZMAX                                                  000444
00276               WRITE(IWRITE,6041)(TAB(ITAB+I),I=1,NXMAX)                       000451
00304               ITAB=LOK+NXMAX+NZMAX                                            000470
00305               NZ=0                                                            000474
00306               WRITE(IWRITE,6121)                                              000475
00310         6121  FORMAT(/10X,'DEPENDENT VARIABLE TABLE'/)                        000503
00311         6140  WRITE(IWRITE,6041)(TAB(ITAB+I),I=1,NXMAX)                       000503
00317               NZ=NZ+1                                                         000522
00320               IF(NZ.GE.NZMAX) GO TO 6400                                      000525
00322               ITAB=ITAB+NXMAX                                                 000530
00323               GO TO 6140                                                      000533
00324         6200  CONTINUE                                                        000535
00325               INDEX=INDEKS                                                    000535
00326               WRITE(IWRITE,6201)CARD                                          000536
00331         6201  FORMAT(45H *** WARNING ***  NON-NUMERIC DATA ON THIS CARD-->,8A10   000546
00331              1/17X,'WILL READ NEXT TABLE'/)                                   000546
00332               GO TO 6070                                                      000546
00333         6300  WRITE(IWRITE,6301)CARD                                          000550
00336         6301  FORMAT(46H *** WARNING ***  NON-ALPHA NAME ON THIS CARD-->,     000557
00336              18A10/17X,'WILL IGNORE THIS CARD'/)                              000557
00337               GO TO 100                                                       000557
00340         6400  WRITE(IWRITE,20)                                                000561
00342               NX=0                                                            000565
00343               NZ=0                                                            000566
00344               MODE=0                                                          000567
00345               WRITE(IWRITE,20)                                                000572
00347               GO TO 100                                                       000575
00350         6500  CONTINUE                                                        000577
00351               INDEX=INDEKS                                                    000577
00351         C --->       CHECK THAT ALL TABLES HAVE BEEN INPUT                    000577
00352         6520  DO 6540 I=1,NOTAB                                               000601
00355               LOK=LOCTAB(I)                                                   000606
00356               CALL STRMOV(TAB(LOK),1,6,TABN,1)                                000607
00357               IF(NCPSTR(TABNAM(I),1,7,TAB(LOK),1).EQ.0)GO TO 6540             000622
00361               WRITE(IWRITE,6531)TABNAM(I)                                     000637
00364         6531  FORMAT(//35H *** WARNING ***   DATA FOR TABLE  ,A6,             000650
00364              1' HAS NOT BEEN INPUT'/)                                         000650
00365         6540  CONTINUE                                                        000650
00367               RETURN                                                          000650
00370               END @  TABIN  ****************************                      000707
```

```
SUBROUTINE TITLE      ENTRY POINT 000144

STORAGE USED  CODE(1) 000164  DATA(0) 000036; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003    GETT
 0004    PUTT
 0005    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000026 10L       0001    000005 112G      0001    000037 132G      0001    000067 145G      0001    000112 157G
 0001     000033 20L       0001    000055 40L       0000 D 000000 BLNK     0000 D 000006 CHAR     0000 D 000002 COMMA
 0000 D 000004 EQUAL       0000 I 000010 I          0000    000022 INJP$    0000 I 000011 I1       0000 I 000012 I2
 0000 I 000015 J1          0000 I 000016 J2         0000 I 000017 K         0000 I 000014 NC       0000 I 000013 NW


00100        CTITLE                                                                             000005
00101              SUBROUTINE TITLE (CARD,IN,TITL,NT)                                           000005
00101        C  VERSION 1.                            REVISED  MAY 15 1975                      000005
00101        C                                                                                  000005
00101        C     PURPOSE - TO LOCATE AND CENTER A TEXTUAL TITLE.                              000005
00101        C                                                                                  000005
00101        C        CARD - INPUT CARD IMAGE                                                   000005
00101        C          IN - CHARACTER AT WHICH TO START SEARCH                                 000005
00101        C        TITL - RESULTING TITLE                                                    000005
00101        C          NT - NUMBER OF CHARACTERS IN TITLE FIELD                                000005
00101        C                                                                                  000005
00103              DOUBLE PRECISION CARD(1),TITL(1),BLNK,COMMA,EQUAL,CHAR                       000005
00104              DATA BLNK /1H /                                                              000005
00106              DATA COMMA /1H,/,               EQUAL /1H=/                                  000005
00106        C                                                                                  000005
00106        C     FIND FIRST NON-BLANK CHARACTER.                                              000005
00106        C                                                                                  000005
00111              DO 10 I=IN,80                                                                000005
00114              I1 = I                                                                       000005
00115              CALL GETT(CARD,I,CHAR)                                                       000007
00116              IF ( CHAR .EQ. COMMA ) GO TO 10                                              000014
00120              IF ( CHAR .EQ. EQUAL ) GO TO 10                                              000017
00122              IF ( CHAR .NE. BLNK ) GO TO 20                                               000022
00124           10 CONTINUE                                                                     000027
00126              RETURN                                                                       000027
00127           20 CONTINUE                                                                     000033
00127        C                                                                                  000033
00127        C     FIND LAST CHARCTER.                                                          000033
00127        C                                                                                  000033
00130              I2 = 81                                                                      000033
```

```
00131              DO 30 I=IN,80                                   000037
00134              I2 = I2 + 1                                     000037
00135              CALL GETT(CARD,I2,CHAR)                         000042
00136              IF ( CHAR .NE. BLNK ) GO TO 40                  000047
00140           30 CONTINUE                                        000055
00142           40 CONTINUE                                        000055
00142        C                                                     000055
00142        C     MOVE TITLE INTO TITL ARRAY.                     000055
00142        C                                                     000055
00143              NW = (NT+1) / 10 + 1                            000055
00144              DO 50 I=1,NW                                    000062
00147              TITL(I) = BLNK                                  000067
00150           50 CONTINUE                                        000071
00152              NC = I2 - I1 + 1                                000071
00153              J1 = (NT-NC) / 2 + 1                            000075
00154              J2 = J1 + NC - 1                                000101
00155              K = I1                                          000104
00156              DO 60 I=J1,J2                                   000112
00161              CALL GETT(CARD,K,CHAR)                          000112
00162              CALL PUTT(TITL,I,CHAR)                          000117
00163              K = K + 1                                       000124
00164           60 CONTINUE                                        000131
00166              RETURN                                          000131
00167              END @  TITLE  ************************          000163
```

```
SUBROUTINE VALUES     ENTRY POINT 000077

STORAGE USED  CODE(1) 000125; DATA(0) 000043; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003    NUMERC
 0004    LCMPH
 0005    PCDREL
 0006    NWDU$
 0007    NI02$
 0010    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000027 100L      0000    000000 101F      0001    000042 200L      0001    000061 300L      0000    000015 301F
 0001    000010 50L       0000    000035 INJP$



00100        CVALUES                                                                        000002
00101               SUBROUTINE VALUES(IPHRS,NAME,NO,VALUE,ITNO,MODE)                        000002
00101        C  PURPOSE   LOADS NUMERIC VALUES OF QUANTITIES IDENTIFIED BY DEFINE           000002
00101        C                 STATEMENTS.                                                  000002
00101        C  CALL SEQUENCE    IPHRS = ARRAY CONTAINING NEXT PHRASE TO BE EXAMINED        000002
00101        C                   NAME  = ARRAY CONTAINING NAMES OF DEFINED QUANTITIES.      000002
00101        C                   NO      = NUMBER OF DEFINED QUANTITIES.                    000002
00101        C                   VALUE = ARRAY INTO WHICH NUMERIC VALUES ARE TO BE LOA      000002
00101        C                   ITNO  = POSITION OF GIVEN QUANTITY IN NAME ARRAY.          000002
00101        C                   MODE  = MODE OF OPERATION.                                 000002
00101        C                   MODE  = 0  A NAME CANNOT BE IDENTIFIED.                    000002
00101        C                   MODE  = 2  NAME HAS BEEN IDENTIFIED.                       000002
00103               DIMENSION NAME(NO),VALUE(NO)                                            000002
00104               DOUBLE PRECISION IPHRS,NAME                                             000002
00104        C  TEST FOR NUMERIC FIRST CHARACTER.                                           000002
00105               CALL NUMERC(IPHRS,$50)                                                  000002
00106               GO TO 200                                                               000006
00106        C  SEARCH NAMELIST FOR NAME CONTAINED IN IPHRS.                                000006
00107        50     CALL LCMPH(IPHRS,NAME,NO,1,ITNO)                                        000010
00110               IF(ITNO.LE.0) GO TO 100                                                 000016
00113        C  NAME FOUND AT LOCATION ITNO.                                                000016
00112               MODE=2                                                                  000021
00113               RETURN                                                                  000023
00113        C  NAME NOT FOUND.                                                             000023
00114        100    WRITE(6,101) IPHRS                                                      000027
00117        101    FORMAT(15X,33H*** WARNING ***  CAN*T IDENTIFY  ,A10,                    000034
00117             1 23H  VALUE WILL BE IGNORED)                                             000034
00120               MODE=-1                                                                 000034
00121               RETURN                                                                  000036
```

```
C  TEST MODE TO ASSURE THAT NAME HAS BEEN IDENTIFIED.                  000036
200   IF(MODE.NE.2) GO TO 300                                          000042
C  CONVERT NUMERIC VALUE CONTAINED IN IPHRS FROM A TO G FORMAT.        000042
      CALL DCDREL(VALUE(ITNO),IPHRS)                                   000044
      MODE=0                                                           000054
      RETURN                                                           000055
300   WRITE(6,301)IPHRS                                                000061
301   FORMAT(15X,71H*** WARNING ***  A VALID PARAMETER NAME MUST PRECEDE  000066
     1 THE NUMERIC VALUE    ,A10)                                      000066
      RETURN                                                           000066
      END A  VALUES  ******************************                    000124
```

```
SUBROUTINE VARMOD    ENTRY POINT 000164

STORAGE USED  CODE(1) 000173; DATA(0) 000020; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CX      000001
 0004    CXDOT   000001
 0005    CV      000001
 0006    CP      000001
 0007    CXIC    000001
 0010    CTIME   000001

EXTERNAL REFERENCES (BLOCK, NAME)

 0011    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000025 10L       0001    000052 20L       0001    000077 30L       0001    000123 40L       0001    000156 50L
 0000    *000011 INJP$     0006 R 000000 P         0010 R 000000 TIME      0005 R 000000 V         0003 R 000000 X
 0004 R 000000 XDOT        0007 R 000000 XIC



00100      CVARMOD                                                                  000000
00101           SUBROUTINE VARMOD(I,VAR)                                            000000
00101      C  PURPOSE   TO MODIFY THE CURRENT VALUE OF A STATE,VARIABLE,            000000
00101      C              PARAMETER, ETC. GIVEN THE INTEGER IDENTIFICATION CODE     000000
00101      C              FOR THE QUANTITY.                                         000000
00101      C  CALL SEQUENCE    I   = IDENTIFICATION CODE.                           000000
00101      C                   VAR = NEW NUMERIC VALUE BEING INPUT.                 000000
00103           COMMON/CX/X(1)/CXDOT/XDOT(1)/CV/V(1)/CP/P(1)/CXIC/XIC(1)            000000
00104           COMMON/CTIME/TIME                                                   000000
00104      C  TEST FOR PARAMETER CODE                                               000000
00105           IF(I.LE.4000000.OR.I.GT.5000000) GO TO 10                           000000
00107           P(I-4000000)=VAR                                                    000015
00110           RETURN                                                              000021
00110      C  TEST FOR IC CODE                                                      000021
00111      10   IF(I.LE.2000000.OR.I.GT.3000000) GO TO 20                           000025
00113           XIC(I-2000000)=VAR                                                  000042
00114           RETURN                                                              000046
00114      C  TEST FOR VARIABLE CODE                                                000046
00115      20   IF(I.LE.3000000.OR.I.GT.4000000) GO TO 30                           000052
00117           V(I-3000000)=VAR                                                    000067
00120           RETURN                                                              000073
00120      C  TEST FOR STATE CODE                                                   000073
00121      30   IF(I.LT.1.OR.I.GT.1000000) GO TO 40                                 000077
00123           X(I)=VAR                                                            000114
```

```
00124      25*          RETURN                                                            000117
00124      26*     C  TEST FOR RATE CODE                                                  000117
00125      27*     40   IF(I.LE.1000000.OR.I.GT.2000000) GO TO 50                         000123
00127      28*          XDOT(I-1000000)=VAR                                               000140
00130      29*          RETURN                                                            000144
00130      30*     C  TEST FOR TIME CODE                                                  000144
00131      31*     50   IF(I.EQ.0) TIME=VAR                                               000150
00133      32*          RETURN                                                            000153
00134      33*          END @  VARMOD  **************************                         000172
```

```
SUBROUTINE VAROUT    ENTRY POINT 000171

STORAGE USED  CODE(1) 000206; DATA(0) 000020; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CX       000001
 0004    CXDOT    000001
 0005    CV       000001
 0006    CP       000001
 0007    CXIC     000001
 0010    CTIME    000001


EXTERNAL REFERENCES (BLOCK, NAME)

 0011    NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000007 10L        0001    000033 20L        0001    000060 30L        0001    000105 40L        0001    000132 50L
 0001    000157 60L        0000    000012 INJP$      0006 R 000000 P           0010 R 000000 TIME        0005 R 000000 V
 0003 R 000000 X           0004 R 000000 XDOT        0007 R 000000 XIC


00100      1*      CVAROUT                                                                        000000
00101      2*            SUBROUTINE VAROUT(I,VAR)                                                 000000
00101      3*      C  PURPOSE   TO RETRIEVE THE NUMERIC VALUES OF STATES,VARIABLES,               000000
00101      4*      C            PARAMETERS,ETC. GIVEN THE INTEGER IDENTIFICATION CODE             000000
00101      5*      C            FOR THE QUANTITY DESIRED.                                         000000
00101      6*      C  CALL SEQUENCE   I    = IDENTIFICATION CODE.                                 000000
00101      7*      C                  VAR = NUMERIC VALUE RETURNED.                               000000
00103      8*            COMMON/CX/X(1)/CXDOT/XDOT(1)/CV/V(1)/CP/P(1)/CXIC/XIC(1)                 000000
00104      9*            COMMON/CTIME/TIME                                                        000000
00104     10*      C  TEST FOR TIME CODE                                                          000000
00105     11*            IF(I.NE.0) GO TO 10                                                      000003
00107     12*            VAR=TIME                                                                 000001
00110     13*            RETURN                                                                   000003
00110     14*      C  TEST FOR STATE CODE                                                         000003
00111     15*      10    IF(I.LT.1.OR.I.GT.1000000) GO TO 20                                      000007
00113     16*            VAR=X(I)                                                                 000024
00114     17*            RETURN                                                                   000027
00114     18*      C  TEST FOR VARIABLE CODE                                                      000027
00115     19*      20    IF(I.LE.3000000.OR.I.GT.4000000) GO TO 30                                000033
00117     20*            VAR=V(I-3000000)                                                         000050
00120     21*            RETURN                                                                   000054
00120     22*      C   TEST FOR RATE CODE                                                         000054
00121     23*      30    IF(I.LE.1000000.OR.I.GT.2000000) GO TO 40                                000060
00123     24*            VAR=XDOT(I-1000000)                                                      000075
```

```
00124      25*            RETURN                                                          000101
00124      26*      C   TEST FOR PARAMETER CODE                                           000101
00125      27*      40    IF(I.LE.4000000.OR.I.GT.5000000) GO TO 50                       000105
00127      28*            VAR=P(I-4000000)                                                000122
00130      29*            RETURN                                                          000126
00130      30*      C  TEST FOR IC CODE                                                   000126
00131      31*      50    IF(I.LE.2000000.OR.I.GT.3000000) GO TO 60                       000132
00133      32*            VAR=NIC(I-2000000)                                              000147
00134      33*            RETURN                                                          000153
00134      34*      C   CODE NOT IDENTIFIED.  SET VAR TO LARGE NUMBER.                    000153
00135      35*      60    VAR=1.E36                                                       000157
00136      36*            RETURN                                                          000160
00137      37*            END - VAROUT  ************************                          000205
```

```
SUBROUTINE XFR         ENTRY POINT 000025

STORAGE USED  CODE(1) 000036; DATA(0) 000014; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000010 100L      0000 I 000000 I          0000   000002 INJP$


00100       1*      CXFR                                                                                           000010
00101       2*            SUBROUTINE XFR(X,Y,N)                                                                    000010
00103       3*            DIMENSION X(N),Y(N)                                                                      000010
00104       4*            DO 100 I=1,N                                                                             000010
00107       5*      100   Y(I)=X(I)                                                                                000010
00111       6*            RETURN                                                                                   000012
00112       7*            END $  XFR  **************                                                               000035
```

```
SUBROUTINE CUBIC     ENTRY POINT 000173

STORAGE USED  CODE(1) 000205; DATA(0) 000036; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   CBRT
 0004   SQRT
 0005   ACOS
 0006   COS
 0007   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000040 10L        0001   000074 20L        0000 R 000005 AAA       0000 R 000002 AB         0000 R 000003 ABB
 0000 R 000006 BBB        0000   000026 INJP$      0000 R 000007 STCR      0000 R 000004 STERM      0000 R 000011 TE
 0000 R 000020 TER        0000 R 000001 TERM       0000 R 000010 THETA     0000 R 000012 THETA3     0000 R 000013 X1
 0000 R 000014 X2         0000 R 000015 X3


00100      1*      CUBIC                                                                                      000000
00101      2*            SUBROUTINE CUBIC(AA,BB,ANS)                                                          000000
00103      3*            TER=AA**3/27.                                                            00000800    000000
00104      4*            TERM=BB**2/4.+TER                                                        00000900    000004
00105      5*            IF(ABS(TERM).GT..0001)GO TO 10                                           00001000    000011
00105      6*      C     *****************************************************************00001100    000011
00105      7*      C     THREE REAL ROOTS, TWO EQUAL                                              00001200    000011
00105      8*      C     *****************************************************************00001300    000011
00107      9*            AB=2.*CBRT(-BB/2.)                                                       00001400    000015
00110     10*            ABB=-AB/2.                                                               00001500    000025
00110     11*      C     *****************************************************************00001600    000025
00110     12*      C     SELECT POSITIVE ROOT                                                     00001700    000025
00110     13*      C     *****************************************************************00001800    000025
00111     14*            ANS=AMAX1(AB,ABB)                                                                    000027
00112     15*            RETURN                                                                   00002000    000034
00113     16*         10 IF(TERM.LT.0.)GO TO 20                                                   00002100    000040
00113     17*      C     *****************************************************************00002200    000040
00113     18*      C     ONE REAL ROOT, TWO CONJUGATE IMAGINARY ROOTS                             00002300    000040
00113     19*      C     *****************************************************************00002400    000040
00115     20*            STERM=SQRT(TERM)                                                         00002500    000044
00116     21*            AAA=CBRT(-BB/2.+STERM)                                                   00002600    000051
00117     22*            BBB=CBRT(-BB/2.-STERM)                                                   00002700    000057
00117     23*      C     *****************************************************************00002800    000057
00117     24*      C     SELECT REAL ROOT                                                         00002900    000057
00117     25*      C     *****************************************************************00003000    000057
00120     26*            ANS=AAA+BBB                                                                          000064
00121     27*            RETURN                                                                   00003200    000070
00121     28*      C     *****************************************************************00003300    000072
```

```
00121      29*     C      THREE REAL, UNEQUAL ROOTS                                     00003400     000070
00121      30*     C      ***************************************************************00003500     000072
00122      31*         20 STER=SQRT(-TER)                                               00003600     000074
00123      32*            THETA=ACOS(-BB/2./STER)                                       00003700     000101
00124      33*            TE=2.*SQRT(-AA/3.)                                            00003800     000111
00125      34*            THETA3=THETA/3.                                               00003900     000121
00126      35*            X1=TE*COS(THETA3)                                                          000124
00127      36*            X2=TE*COS(THETA3+2.09439)                                                  000131
00130      37*            X3=TE*COS(THETA3+4.18879)                                                  000141
00130      38*     C      ***************************************************************00004300     000141
00130      39*     C      SELECT SMALLEST POSITIVE ROOT                                 00004400     000141
00130      40*     C      ***************************************************************00004500     000141
00131      41*            ANS=AMAX1(X1,X2,X3)                                           00004600     000151
00132      42*            RETURN                                                        00005000     000163
00133      43*            END                                                           00005100     000204
```

```
SUBROUTINE IMPLIC     ENTRY POINT 000245

STORAGE USED  CODE(1) 000254; DATA(0) 000092; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003    CIMPL   000002
 0004    CORDER  000002
 0005    CWORK   000311
 0006    CV      000001
 0007    CNAMEV  000002
 0010    CTIME   000001

EXTERNAL REFERENCES (BLOCK, NAME)

 0011    NWDU$
 0012    NI02$
 0013    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000050 10L        0000     000005 100F       0001     000046 121G       0001     000122 141G       0001     000154 160G
 0001     000134 20L        0000     000006 200F       0001     000127 30L        0001     000231 40L        0001     000220 50L
 0005     000000 A          0000 I 000003 I          0000 I 000004 ICN        0003 I 000001 ICNT       0000 I 000001 ILINES
 0003 I 000000 IMPL       0000     000030 INJP$      0000 I 000000 ITERS      0000 I 000002 ITNO       0007 D 000000 NAMEV
 0004 I 000001 NOV        0004     000000 NOX        0010 R 000000 TIME       0006 R 000000 V          0005 R 000310 VOLD


00100       1*      CIMPLIC                                                                          000002
00101       2*            SUBROUTINE IMPL(CYCLES,OLINES)                                             000002
00103       3*            COMMON/CIMPL/IMPL,ICNT /CORDER/ NOX,NOV /CWORK/ A(200),VOLD                000002
00104       4*            COMMON /CV/ V /CNAMEV/ NAMEV /CTIME/ TIME                                  000002
00105       5*            DIMENSION V(1),VOLD(1)                                                     000002
00106       6*            DOUBLE PRECISION NAMEV(1)                                                  000002
00107       7*            IF(CYCLES.LE.0.) GO TO 40                                                  000002
00111       8*            IF(IMPL.GT.2)GO TO 10                                                      000005
00113       9*            ITERS=CYCLES                                                               000010
00114      10*            ITERS= MAX0(1,MIN0(ITERS,20))                                              000017
00115      11*            ILINES= ABS(OLINES)                                                        000030
00116      12*            ITNO= 0                                                                    000040
00117      13*            IMPL=1                                                                     000041
00120      14*            DO 5 I=1,NOV                                                               000046
00123      15*          5 VOLD(I) = 0.0                                                              000046
00125      16*         10 IF(IMPL.GT.1) GO TO 20                                                     000050
00127      17*            ITNO= ITNO+1                                                               000053
00131      18*            IF(ITNO.GE.ITERS) IMPL=2                                                   000056
00132      19*            IF(IMPL.EQ.2 .AND. ICNT.GE.ILINES)IMPL=3                                   000064
00134      20*            IF(IMPL.NE.2) RETURN                                                       000103
```

```
00136         IF(OLINES.LT.O.)RETURN                                      000111
00140         DO 30 I=1,NOV                                               000122
00143         IF(VOLD(I).EQ.0.123456) GO TO 30                            000122
00145         VOLD(I)= V(I)                                               000124
00146      30 CONTINUE                                                    000130
00150         RETURN                                                      000130
00151      20 ITNO=0                                                      000134
00152         IF(IMPL.GT.2) GO TO 40                                      000134
00154         IF(OLINES.LT.O.) GO TO 40                                   000140
00156         ICK=0                                                       000146
00157         DO 50 I=1,NOV                                               000154
00162         IF( ABS(V(I)).LT.1.0E-6) GO TO 50                           000154
00164         IF(VOLD(I).EQ. 0.123456)GO TO 50                            000157
00166         IF( ABS(VOLD(I)-V(I)) .LT. 0.05*ABS(V(I))) GO TO 50         000162
00173         IF(ICK.EQ.0) WRITE (6,100)                                  000172
00173     100 FORMAT(1H0)                                                 000201
00174         WRITE(6,200) NAMEV(I),VOLD(I),V(I)                          000201
00201     200 FORMAT(1H ,10X,A6,28H  NONCONVERGENCE. OLD VALUE=,F12.3,    000211
00201        1  13H   NEW VALUE=,F12.3)                                   000211
00202         ICK=1                                                       000211
00203         IF(TIME.EQ.0.)VOLD(I)= 0.123456                             000213
00205      50 CONTINUE                                                    000222
00207         IF(ICK.EQ.1) ICNT=ICNT+1                                    000222
00211      40 IMPL=4                                                      000231
00212         RETURN                                                      000232
00213         END                                                         000253
```

```
FUNCTION TBLU1        ENTRY POINT 000311

STORAGE USED  CODE(1) 000333; DATA(0) 000034; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000065 10L        0001     000136 100L       0001     000116 20L        0001     000164 30L        0001     000222 40L
 0001     000022 5L         0001     000226 50L        0001     000242 60L        0001     000244 70L        0000 R 000003 H
 0000 I 000005 I            0000 I 000006 IGC          0000 I 000007 II           0000     000011 INJP$      0000 I 000001 NA
 0000 R 000000 TBLU1        0000 R 000004 XI           0000 R 000002 X0


00100       1*      CTBLU1                                                                                  000007
00101       2*            FUNCTION TBLU1(X,XT,FT,NDX,NX)                                                    000007
00101       3*      C                                                                                       000007
00101       4*      C     PURPOSE     ONE DIMENSION LINEAR INTERPOLATION                                    000007
00101       5*      C                                                                                       000007
00101       6*      C     CALL SEQUENCE                                                                     000007
00101       7*      C                                                                                       000007
00101       8*      C             X  - VALUE OF INDEPENDENT VARIABLE                                        000007
00101       9*      C             XT - ARRAY OF LENGTH ABS(NX) CONTAINING X VALUES                          000007
00101      10*      C             FT - ARRAY OF TABLE VALUES CORRESPONDING TO XT                            000007
00101      11*      C             NDX- INDICATOR FOR STEP SPACING                                           000007
00101      12*      C                  IF NDX.EQ.0 THEN XT CONTAINS EQUAL SPACED DATA                       000007
00101      13*      C                  IF NDX.NE.0 THEN XT CONTAINS UNEQUAL SPACED DATA                     000007
00101      14*      C             NX - ABS(NX) IS THE ARRAY LENGTH                                          000007
00101      15*      C                  IF NX.LT.0 THEN TRUNCATE OUTSIDE TABLE RANGE                         000007
00101      16*      C                  IF NX.GE.0 THEN EXTRAPOLATE OUTSIDE TABLE RANGE                      000007
00101      17*      C                                                                                       000007
00101      18*      C     WRITTEN BY A.W.WARREN                            VERSION 1, APRIL 1977            000007
00101      19*      C                                                                                       000007
00103      20*            DIMENSION XT(1),FT(1)                                                             000007
00104      21*            NA=IABS(NX)                                                                       000007
00105      22*            IF(NA.GT.1)GO TO 5                                                                000011
00107      23*            TBLU1=FT(1)                                                                       000014
00110      24*            RETURN                                                                            000016
00111      25*          5 IF(NDX.NE.0) GO TO 100                                                            000022
00111      26*      C                                                                                       000022
00111      27*      C                                EQUI-SPACED TABLE INTERPOLATION                        000022
00111      28*      C                                                                                       000022
00113      29*            X0= XT(1)                                                                         000023
00114      30*            H= XT(2)-XT(1)                                                                    000025
00115      31*            XI= (X-X0)/H +1.                                                                  000027
00116      32*            I=XI                                                                              000034
```

```
00117                 IF(I.GT.0) GO TO 10                                                  000043
00121                 TBLU= FT(1)                                                          000045
00122                 IF(NX.GE.0)TBLU= FT(1) + (XI-1.)*(FT(2)-FT(1))                       000047
00124                 RETURN                                                               000061
00125              10 IF(I.LT.NA) GO TO 20                                                 000065
00127                 TBLU=FT(NA)                                                          000070
00130                 IF(NX.GE.0) TBLU= FT(NA) + (XI-NA)*(FT(NA)-FT(NA-1))                 000075
00132                 RETURN                                                               000112
00133              20 TBLU= FT(I) + (XI-I)*(FT(I+1)-FT(I))                                 000116
00134                 RETURN                                                               000132
00134       C                                                                              000132
00134       C                                   UNEQUAL SPACED TABLE INTERPOLATION         000132
00134       C                                                                              000132
00135        100  IF(X.GT.XT(1)) GO TO 30                                                  000136
00137                 TBLU=FT(1)                                                           000141
00140                 IF(NX.GE.0) TBLU= FT(1) + (X-XT(1))*(FT(2)-FT(1))/(XT(2)-XT(1))      000143
00142                 RETURN                                                               000160
00141              30 IF(X.LT.XT(NA)) GO TO 40                                             000164
00145                 TBLU= FT(NA)                                                         000171
00146                 IF(NX.GE.0) TBLU=FT(NA)+(X-XT(NA))*(FT(NA)-FT(NA-1))/(XT(NA)         000177
00146               1      - XT(NA-1))                                                     000177
00150                 RETURN                                                               000216
00151              40 I=1                                                                  000222
00152                 IGE= NA                                                              000223
00153              50 II=(IGE+I)/2                                                         000226
00154                 IF(X.LT.XT(II)) GO TO 60                                             000231
00156                 I= II                                                                000237
00157                 GO TO 70                                                             000240
00160              60 IGE= II                                                              000242
00161              70 IF(I+1.LT.IGE) GO TO 50                                              000244
00163                 TBLU= FT(I) + (FT(I+1)-FT(I))*(X - XT(I))/(XT(I+1)-XT(I))            000250
00164                 RETURN                                                               000272
00165                 END                                                                  000332
```

```
FUNCTION TBLU2       ENTRY POINT 000353


STORAGE USED  CODE(1) 000456; DATA(0) 000030; BLANK COMMON(2) 000000


EXTERNAL REFERENCES (BLOCK, NAME)

 0003   TBLU1
 0004   NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000034 10L       0001    000127 100L      0001    000053 20L       0001    000215 200L      0001    000235 240L
 0001    000241 250L      0001    000255 260L      0001    000257 270L      0001    000264 300L      0000 R 000003 E
 0000 R 000004 FF1        0000 I 000006 I          0000 I 000007 IGE        0000 I 000010 II         0000    000013 INJP$
 0000 I 000011 JJ         0000 I 000001 NA         0000 I 000002 NB         0000 I 000005 NB1        0003 R 000000 TBLU1
 0000 R 000000 TBLU2



00100      1*      CTBLU2                                                                                     000007
00101      2*           FUNCTION TBLU2(X,Y,XT,YT,FT,IX,IY,NX,NY,MX,MY)                                         000007
00101      3*      C                                                                                          000007
00101      4*      C   PURPOSE    TWO DIMENSION LINEAR INTERPOLATION                                              000007
00101      5*      C                                                                                          000007
00101      6*      C   METHOD     BINARY SEARCH TO FIND NEAREST GRID POINTS.                                      000007
00101      7*      C              TBLU1 IS USED TO REDUCE THE INTERPOLATION DIMENSION.                            000007
00101      8*      C                                                                                          000007
00101      9*      C   CALL SEQUENCE                                                                          000007
00101     10*      C                                                                                          000007
00101     11*      C              X,Y  - POINT AT WHICH INTERPOLATION IS DESIRED                                  000007
00101     12*      C              XT,YT- ARRAYS CONTAINING INDEPENDENT VARIABLE GRID POINTS                       000007
00101     13*      C              FT   - TWO DIEMSNION ARRAY OF VALUES SUCH THAT FT(I,J)                          000007
00101     14*      C                     CORRESPONDS TO XT(I),YT(J).                                              000007
00101     15*      C              IX,IY- INDICATORS FOR GRID SPACING                                              000007
00101     16*      C                        IF IX=0  THEN XT CONTAINS EQUAL SPACED VALUES                         000007
00101     17*      C                        IF IX.NE.0 THEN XT CONTAINS UNEQUAL SPACED VALUES                     000007
00101     18*      C              NX,NY- ABS(NX),ABS(NY) ARE THE ARRAY DIMENSIONS FOR XT,YT                       000007
00101     19*      C                        IF NX.LT.0 THEN TRUNCATE OUTSIDE XT RANGE                             000007
00101     20*      C                        IF NX.GT.0 THEN EXTRAPOLATE OUTSIDE XT RANGE                          000007
00101     21*      C                        LIKEWISE FOR NY AND YT VALUES.                                        000007
00101     22*      C              MX,MY- DUMMY ARGUMENTS.SET EQUAL TO ABS(NX), ABS(NY).                           000007
00101     23*      C                                                                                          000007
00101     24*      C   WRITTEN BY A.W. WARREN                          VERSION 1, JUNE 1977                       000007
00101     25*      C                                                                                          000007
00103     26*           DIMENSION XT(1),YT(1),FT(1)                                                           000007
00104     27*           NA = IABS(NX)                                                                         000007
00105     28*           MX = NA                                                                               000011
00106     29*           NB = IABS(NY)                                                                         000012
00107     30*           MY = NB                                                                               000014
```

```
00110          IF(NA.GT.1)GO TO 10                                             000015
00112          TBLU2 =  TBLU1(Y,YT,FT,IY,NY)                                   000020
00113          RETURN                                                          000030
00114       10 IF(NB.GT.1)GO TO 20                                             000034
00116          TBLU2 =  TBLU1(X,XT,FT,IX,NX)                                   000037
00117          RETURN                                                          000047
00117     C                               Y OUTSIDE YT TABLE RANGE             000047
00117     C                                                                    000047
00120       20 IF( Y.GT. YT(1))GO TO 100                                       000053
00122          E = (Y-YT(1))/(YT(2)- YT(1))                                    000056
00123          FF1 = TBLU1(X,XT,FT(1),IX,NX)                                   000064
00124          TBLU2 =FF1                                                      000076
00125          IF(NY.GT.0)TBLU2 =FF1+ E*( TBLU1(X,XT,FT(NA+1),IX,NX) -FF1)     000077
00127          RETURN                                                          000123
00127     C                                                                    000123
00130      100 IF( Y.LT. YT(NB))GO TO 200                                      000127
00132          E = (YT(NB)-Y)/(YT(NB)-YT(NB-1))                                000134
00133          NB1 = NA*(NB-1)+1                                               000142
00134          FF1 = TBLU1(X,XT,FT(NB1),IX,NX)                                 000147
00135          TBLU2 = FF1                                                     000163
00136          IF(NY.GT.0)TBLU2 = FF1+ E*(TBLU1(X,XT,FT(NB1-NA),IX,NX) -FF1)   000164
00140          RETURN                                                          000211
00143     C                                                                    000211
00140     C                               YT GRID SEARCH AND INTERPOLATION     000211
00143     C                                                                    000211
00141      200 IF(IY.NE.0)GO TO 240                                            000215
00143          I = (Y - YT(1))/(YT(2)-YT(1)) + 1.                              000216
00144          GO TO 300                                                       000233
00145      240 I=1                                                             000235
00146          IGE = NB                                                        000236
00147      250 IT = (IGE+I)/2                                                  000241
00150          IF(Y.LT. YT(IT))GO TO 260                                       000244
00152          I= IT                                                           000252
00153          GO TO 270                                                       000253
00154      260 IGE = IT                                                        000255
00155      270 IF(I+1 .LT. IGE)GO TO 250                                       000257
00155     C                                                                    000257
00157      300 E = (Y-YT(I))/(YT(I+1)-YT(I))                                   000264
00160          I1= NA*(I-1)+1                                                  000275
00161          FF1 = TBLU1(X,XT,FT(I1),IX,NX)                                  000302
00162          TBLU2 = FF1 + E*(TBLU1(X,XT,FT(I1+NA),IX,NX) -FF1)              000316
00163          RETURN                                                          000336
00164          END                                                             000455
```

```
SUBROUTINE UNIF      ENTRY POINT 000055

STORAGE USED  CODE(1) 000067; DATA(0) 000017; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CIMPL  000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0004   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0003   000001 ICNT      0003 I 000000 IMPL      0000   000013 INJP$     0003 I 000002 ITEST     0000 D 000000 X
 0000 D 000002 Y


00100      1*     CUNIF                                                                          000000
00101      2*           SUBROUTINE UNIF(U,IX)                                                    000000
00103      3*           COMMON /CIMPL/ IMPL,ICNT,ITEST                                           000000
00104      4*           DOUBLE PRECISION X,Y                                                     000000
00105      5*           DATA Y/253967.D0/                                                        000000
00107      6*           IF(IMPL.EQ.0 .AND. ITEST.EQ.1) IX=431469                                 000000
00111      7*           IF (IX.EQ.1) IX = 431469                                                 000013
00113      8*           X= DMOD( IX*Y,16777216.D0)                                               000020
00114      9*           U= X/16777215.                                                           000035
00115     10*           IX=X                                                                     000045
00116     11*           RETURN                                                                   000046
00117     12*           END                                                                      000066
```

4.0 PERMANENT FILE MAINTENANCE PROGRAM DESCRIPTION

4.1 INTRODUCTION

The Permanent File Maintenance program (FILOAD) is used to load and modify standard component input-output descriptions which are kept on the permanent file, M18. This program is used only when it is necessary to modify the input, output, or table list of an existing standard component or when a new standard component is to be added to the system.

4.2 PROGRAM STRUCTURE

Figure 4.2-1 contains a macro flow diagram of the Permanent File Maintenance program. Statement numbers in the main (FILOAD) program are given for each of the program's five principle tasks. The sequence of performing these tasks depends on the program commands. As each command is read it is printed on the lineprinter to provide a record of progress through the set of commands.

4.2.1 Command Interpretation

The command interpretation process for the FILOAD program is shown on Figure 4.2-2. Each phrase is tested against the five possible command phrases: LIST STANDARD COMPONENTS, PURGE, NEW FILE, DUMP FILE, and SYMBOL. If one of these phases is identified, branching occurs from statement 300 to a location that performs these tasks.

The LIST STANDARD COMPONENTS command sets a flag, (LIST=1), which causes the input, output, and table lists of any new or modified components to be printed upon the completion of processing all input commands. The PURGE command causes the name of the purged component to be removed from the list of standard component names, CMPNTS. This results in the removal of all name lists associated with that component from the M18 file, when the degas process is performed at the end of the run. The SYMBOL command causes the symbol number following a standard component name to be added to characters 9 and 10 of that name via the PUTCOD routine.

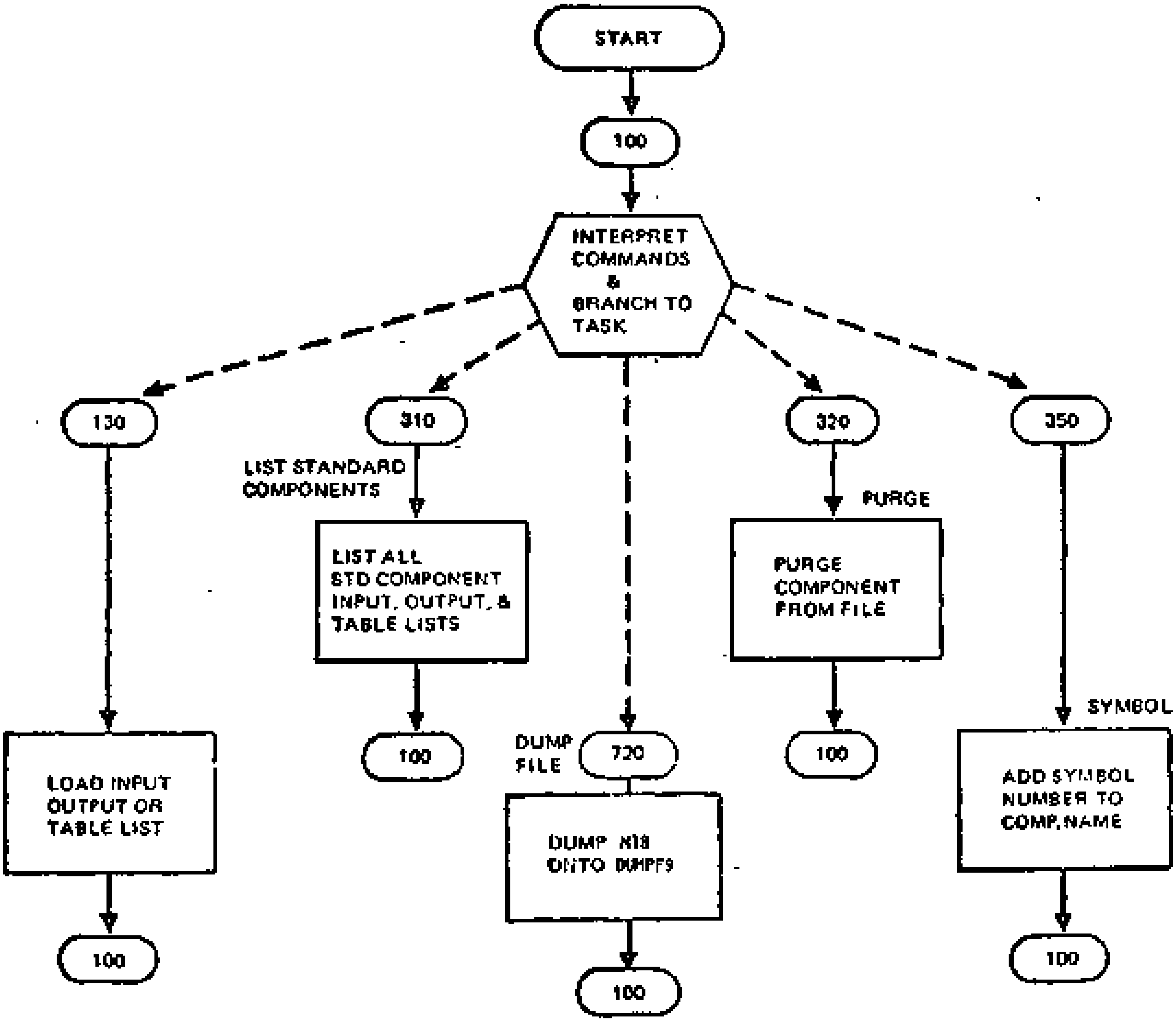

START
100
INTERPRET COMMANDS & BRANCH TO TASK
130
310
320
350
LIST STANDARD COMPONENTS
LIST ALL STD COMPONENT INPUT, OUTPUT, & TABLE LISTS
PURGE
PURGE COMPONENT FROM FILE
LOAD INPUT OUTPUT OR TABLE LIST
100
DUMP FILE
720
DUMP X18 ONTO DUMPF9
100
SYMBOL
ADD SYMBOL NUMBER TO COMP.NAME
100
100
100
100

FIGURE 4.2-1. PERMANENT FILE MAINTENANCE PROGRAM - MACRO FLOW DIAGRAM

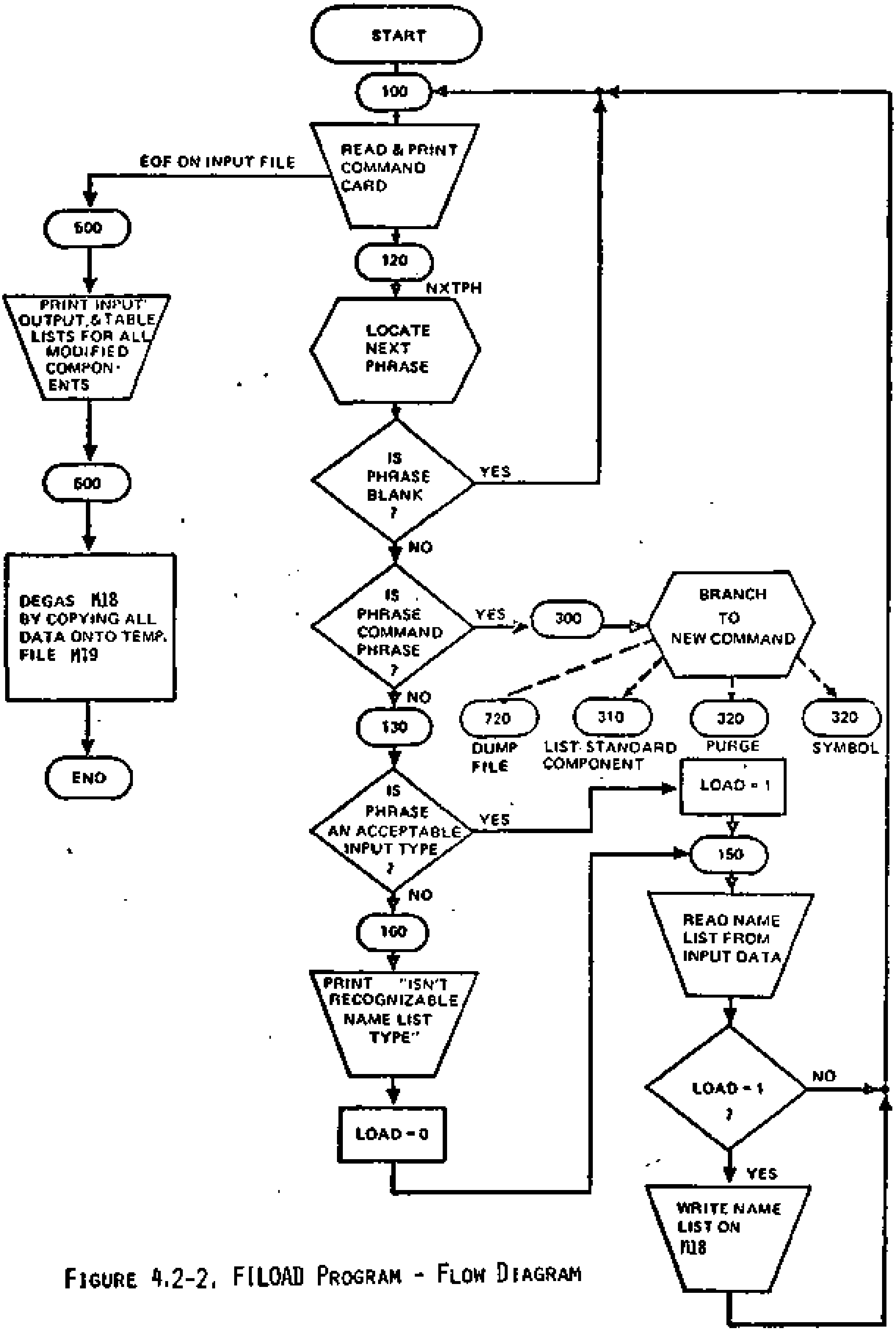

START
100
READ & PRINT COMMAND CARD
EOF ON INPUT FILE
500
PRINT INPUT OUTPUT, & TABLE LISTS FOR ALL MODIFIED COMPONENTS
600
DEGAS M18 BY COPYING ALL DATA ONTO TEMP. FILE M19
END
120
NXTPH
LOCATE NEXT PHRASE
IS PHRASE BLANK ?
YES
NO
IS PHRASE COMMAND PHRASE ?
YES
300
BRANCH TO NEW COMMAND
720
DUMP FILE
310
LIST STANDARD COMPONENT
320
PURGE
320
SYMBOL
NO
130
IS PHRASE AN ACCEPTABLE INPUT TYPE ?
YES
LOAD = 1
150
NO
160
PRINT "ISN'T RECOGNIZABLE NAME LIST TYPE"
LOAD = 0
READ NAME LIST FROM INPUT DATA
LOAD = 1 ?
NO
YES
WRITE NAME LIST ON M18

Figure 4.2-2. FILOAD Program - Flow Diagram

4.2.2 Name List Loading

If a phrase is not a command phrase, characters 3 through 6 are compared to the three acceptable input name list types: INPT, OUTP,and TABS. If one of these three types is not recognized, a warning message is printed and a flag (LOAD=0) is set to prevent data from being loaded onto the M18 file. If a recognizable name list type occurs, the component name is obtained from characters 1 and 2 of the phrase. This component name is compared to existing component names. If it is an existing component name the specified name list for that component is modified. If the component name does not match an existing component name, the new component name is added to the list of library components and a notice is printed that a new component has been added. Default input, output, and table name lists of zero length are then added to the M18 file to assure that all three lists exist for all components. This is necessary to prevent READMS errors in the Model Generation program for components that might otherwise not have table name lists. The name list contained in the input data is then read and loaded onto the M18 file.

The name list data is not in a free field format. The number of names must match that given in the phrase following the input list name, and the format of the name data must match that given in Section 7 of Volume II. Errors in formating name list data can cause erroneous lists to be loaded. These will lead to errors in connections to the affected component.

4.2.3 M18 File Degas Procedure

The WRITMS routine leaves previous versions of stored items on the permanent file as "dead space" whenever the new version is of a different length than the original. In order to remove this dead space, the FILOAD program creates a new copy of the M18 file on local file M19 upon the completion of each run. M19 is loaded by copying the input, output, and table name list for each component listed in the list CMPNTS, from M18.

It is during this copy that the name lists for any purged components are deleted. Upon the successful completion of the run, M19 is copied onto M18.

4.2.4 Permanent Files

The random access permanent file M18 is referred to in the FILOAD program as unit 18. This file contains an input, output, and table name list for each standard component and a list of all standard component names.

4.2.5 Warning Messages

Table 4.2 lists the three warning messages that can be generated by the FILOAD program. These messages are preceded by:
WARNING. If either messages 1 or 2 are printed, the name list associated with these warnings will not be loaded. Other correct name lists for that or other components will be loaded.

4.3 FILOAD PROGRAM SOURCE LISTINGS

Compilation listings of the source code for the Fiload program follows. Some of the subroutines are also used in the other programs. The names of the FILOAD routines, listed in alphabetical order, are:

BCDDUB
COMDAT
CSORT
DAND
DCMPL
DOR
DUMPPF
FILOAD
GETCOD
GETT
ISCAN
KOMSTR
LCMPH
NCODE
NUMERC
NXTPH
PUTCOD
PUTT
READMS
SHIFT
STRMOV
WRITMS

TABLE 4.2

PERMANENT FILE MAINTENANCE PROGRAM WARNING MESSAGES

1. CAN'T IDENTIFY xx AS A STANDARD COMPONENT

   The phase xx following the command PURGE or SYMBOL is not an existing standard component name. Check spelling of xx.

2. IN xxxxxxxxxx zzzz ISN'T A RECOGNIZED NAME LIST TYPE. NAME LIST WILL NOT BE LOADED.

   Characters 3 through 6, zzzz, in the phrase xxxxxxxxxx should be one of the name list types: INPT, OUTP, or TABS. Check spelling of xxxxxxxxxx.

3. xxxxxxxxxx ISN'T A VALID NUMBER OF NAMES FOR NAME LIST. NAME LIST WILL NOT BE LOADED.

   A numeric phrase giving the number of names in the following name list must follow the component name-list type phrase.

```
SUBROUTINE BCODUB    ENTRY POINT 000125

STORAGE USED  CODE(1) 000134; DATA(0) 000033; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   GETT
 0004   PUTT
 0005   NNCOD$
 0006   NDCOD$
 0007   NI02$
 0010   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000    000015 115F      0001    000004 117G      0000    000016 125F      0001    000021 127G      0001    000076 30L
 0000 D 000010 BLANKS     0000    000023 INJP$     0000 I 000014 K         0000 D 000006 PERIOD     0000 D 000004 T
 0000 D 000000 TEMP       0000 D 000002 TPHRS      0000 D 000012 TT


00101      1*          SUBROUTINE BCODUB(VALUE,PHRS)                                        000000
00103      2*          DOUBLE PRECISION VALUE, PHRS                                         000000
00104      3*          DOUBLE PRECISION TEMP                                                000000
00104      4*     C  PURPOSE  CONVERT ALPHA NUMERIC INFORMATION INTO O/P FORMAT            000000
00104      5*     C  CALL SEQUENCE    VALUE - DOUBLE PRECISION NUMERIC VALUE ON RETURN     000000
00104      6*     C                   PHRS - LEFT ADJUSTED ALPHA CHARACTERS ON INPUT       000000
00104      7*     C                                                                         000000
00104      8*     C    IF LS-CHARACTER OF PHRS IS NOT '.', THEN INSERT PERIOD              000000
00105      9*          DOUBLE PRECISION TPHRS, T / '                                        000000
00107     10*          DOUBLE PRECISION PERIOD / '.                                         000000
00111     11*          DOUBLE PRECISION BLANKS / '                                          000000
00113     12*          TPHRS = PHRS                                                         000000
00114     13*          DOUBLE PRECISION TT                                                  000004
00114     14*     C    CHECK FOR PERIOD                                                     000004
00116     15*          DO 10 K = 1, 12                                                      000004
00121     16*          CALL GETT(TPHRS, K, TT)                                              000004
00122     17*          IF (TT .EQ. PERIOD) GO TO 30                                         000011
00124     18*       10 CONTINUE                                                             000021
00126     19*          DO 20 K = 12, 1, -1                                                  000021
00131     20*          CALL GETT(TPHRS, K, T)                                               000021
00132     21*          IF (T .NE. BLANKS .AND. T .NE. PERIOD)                               000026
00132     22*         1     CALL PUTT(TPHRS, K+1, PERIOD)                                   000026
00134     23*          IF (T .NE. BLANKS .AND. T .NE. PERIOD) GO TO 30                      000052
00136     24*       20 CONTINUE                                                             000070
00140     25*          VALUE = 0                                                            000070
00141     26*          RETURN                                                               000072
00142     27*       30 CONTINUE                                                             000076
00142     28*     C                                                                         000076
```

```
00142     29*     C     NEXT, RIGHT JUSTIFY ALPHA REPRESENTATION IN                                 000076
00142     30*     C     PHRS USING R12 EDIT CODE                                                    000076
00143     31*           ENCODE(12, 115, TEMP) IPHRS                                                 000076
00146     32*       115 FORMAT(R12)                                                                 000104
00146     33*     C                                                                                 000104
00146     34*     C     NOW, WE ARE READY FOR DECODE                                                000104
00147     35*           DECODE(12, 125, TEMP) VALUE                                                 000104
00152     36*       125 FORMAT(G12.6)                                                               000113
00153     37*           RETURN                                                                      000113
00154     38*           END                                                                         000133
```

```
SUBROUTINE COMDAT     ENTRY POINT 000137

STORAGE USED  CODE(1) 000164; DATA(0) 000040; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CIO     000003


EXTERNAL REFERENCES (BLOCK, NAME)

 0004   STRMOV
 0005   READMS
 0006   NWDU$
 0007   NIO1$
 0010   NIO2$
 0011   NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000   000007 101F      0001   000064 124G      0001   000106 135G      0001   000115 200L      0000   000014 201F
 0000 D 000002 AINDEX    0000 D 000000 DBLANK    0000 D 000004 DN        0000 I 000006 I         0003 I 000002 IOIAG
 0000   000027 INJP$     0003   000000 IREAD     0003 I 000001 IWRITE



00101      1*              SUBROUTINE COMDAT(CONNAM,TYPE,N,NAMES)                                              000002
00103      2*                         IMPLICIT DOUBLE PRECISION (A - Z)                                        000002
00104      3*                         IMPLICIT INTEGER (I, J, K, L, M, N)                                      000002
00105      4*                         DOUBLE PRECISION DBLANK / ' '                                            000002
00105      5*        C  PURPOSE   OBTAIN LISTS OF INPUTS, OUTPUTS, OR TABLES REQUIRED                          000002
00105      6*        C            FOR A SPECIFIED STANDARD COMPONENT                                           000002
00105      7*        C  CALL SEQUENCE   CONNAM - STANDARD COMPONENT NAME                                       000002
00105      8*        C                  TYPE    - TYPE OF LIST REQUESTED E.G. INPT,OUTP,TABS                   000002
00105      9*        C                  N       - NUMBER OF NAMES IN LIST                                      000002
00105     10*        C                  NAMES   - NAMES OF QUANTITIES                                          000002
00105     11*        C  METHOD   LISTS ARE STORED ON A RANDOM ACCESS PERMANENT FILE AND                        000002
00105     12*        C           ACCESSED VIA THE MASS STORAGE I/O FEATURES OF FTN.                            000002
00105     13*        C         FOR EACH STANDARD COMPONENT, 3 LISTS WILL BE CREATED                            000002
00105     14*        C           WITH THE INDEX NAMES  XXINPT, XXOUTP, XXTABS  WHERE XX                        000002
00105     15*        C           REPRESENTS THE STANDARD COMPONENT NAME.  THE FIRST WORD                       000002
00105     16*        C           IN EACH LIST WILL CONTAIN THE NUMBER OF WORDS IN THE LIST                     000002
00105     17*        C           PLUS 1.                                                                       000002
00107     18*              COMMON/CIO/IREAD,IWRITE,IOIAG                                                       000002
00110     19*              DOUBLE PRECISION NAMES(1)                                                           000002
00110     20*        C --->      FORM INDEX                                                                    000002
00111     21*              AINDEX=DBLANK                                                                       000002
00112     22*              CALL STRMOV(CONNAM,1,2,AINDEX,1)                                                    000004
00113     23*              CALL STRMOV(TYPE,1,4,AINDEX,3)                                                      000013
00113     24*        C --->      READ FIRST WORD IN RECORD                                                     000013
```

```
00114           CALL READNS(IB,DN,1,AINDEX)                                        000022
00115           N = DN                                                             000030
00115     C --->      READ N WORDS                                                 000030
00116           IF(N.LT.1)N=1                                                      000036
00120           CALL READNS(IB,NAMES,N,AINDEX)                                     000044
00121           IF(N.LE.1) GO TO 200                                               000052
00121     C --->      SHIFT WORDS OVER ONE TO ELLIMINATE NO. OF WORDS STORED IN 1S  000052
00123           DO 100 I=2,N                                                       000056
00126           NAMES(I-1)=NAMES(I)                                                000064
00127     100   CONTINUE                                                           000066
00131           N=N-1                                                              000066
00132           IF(IDIAG.EQ.80)WRITE(IWRITE,101)(NAMES(I),I=1,N)                   000071
00141     101   FORMAT(' CONDAT-NAMES'/(20A4))                                     000111
00142           RETURN                                                             000111
00143     200   N=0                                                                000115
00144           IF(IDIAG.EQ.80)WRITE(IWRITE,201)                                   000115
00147     201   FORMAT(' CONDAT-N=0')                                              000125
00153           RETURN                                                             000125
00154           END @ ********************************************                 000163
```

```
SUBROUTINE CSORT      ENTRY POINT 000203


STORAGE USED  CODE(1) 000215; DATA(0) 000046; BLANK COMMON(2) 000000


EXTERNAL REFERENCES (BLOCK, NAME)

 0003    ISCAN
 0004    PUTT
 0005    NERR$$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000027 125G       0001     000112 150G       0001     000021 200L       0001     000072 220L       0001     000075 300L
 0001     000077 320L       0001     000115 330L       0001     000156 340L       0001     000161 400L       0001     000171 990L
 0000 D 000022 DAI          0000 D 000024 DAII         0000 D 000004 DBLANK       0000 D 000016 DTEMP        0000 I 000000 I
 0000 I 000001 II           0000   000033 INJP$        0003 D 000000 ISCAN        0000 D 000020 ITEMP        0000 I 000011 J
 0000 I 000022 JDAI         0000 I 000024 JDAII        0000 I 000015 K            0000 D 000020 KHR1         0000 D 000002 KHR2
 0000 I 000010 K2           0000 I 000007 LIN          0000 I 000014 M            0000 I 000013 M1           0000 I 000006 N
 0000 I 000012 N2



00101        1*              SUBROUTINE CSORT (IA,NN)                                                         000002
00103        2*                           IMPLICIT DOUBLE PRECISION (A - Z)                                    000002
00104        3*                           IMPLICIT INTEGER (J - N)                                             000002
00105        4*                           DOUBLE PRECISION IA(1)                                               000002
00106        5*                     INTEGER I, II                                                              000002
00106        6*        C******                                                                                 000002
00106        7*        C     PURPOSE                                                                           000002
00106        8*        C        CSORT SORTS THE ELEMENTS OF A SINGLE-DIMENSION DOUBLE-                         000002
00106        9*        C        PRECISION ARRAY IN ASCENDING-CHARACTER (DISPLAY CODE) ORDER,                   000002
00106       10*        C        WITH A SORT OPTION THAT PLACES BLANK CHARACTERS FIRST IN THE                   000002
00106       11*        C        ALPHAMERIC SEQUENCE.                                                           000002
00106       12*        C        THE SHELL ALGORITH IS USED.                                                    000002
00106       13*        C     USAGE                                                                             000002
00106       14*        C        DIMENSION IA(J)      WHERE J=IABS(N)                                           000002
00106       15*        C        CALL CSORT(IA,N)                                                               000002
00106       16*        C     INPUT PARAMETERS                                                                  000002
00106       17*        C           IA - INPUT ARRAY TO BE SORTED IN PLACE                                      000002
00106       18*        C            N - IABS(N) IS NUMBER OF ELEMENTS IN ARRAY IA                              000002
00106       19*        C                N.LT.0  PERFORM NORMAL SORT, SEE ABSTRACT                              000002
00106       20*        C                N.GT.0  PERFORM MODIFIED SORT, SEE ABSTRACT                            000002
00106       21*        C     OUTPUT PARAMETERS                                                                 000002
00106       22*        C           IA - THE INPUT ARRAY IS SORTED IN PLACE                                     000002
00106       23*        C     USER ERROR                                                                        000002
00106       24*        C        WHEN N.EQ.0, CONTROL IS RETURNED TO THE CALLING PROGRAM                        000002
00106       25*        C        WITHOUT SORTING.                                                               000002
00106       26*        C******                                                                                 000002
00106       27*        C                                                                                       000002
```

```
00107                 DOUBLE PRECISION KHR1, KHR2                                          000002
00110                 EQUIVALENCE (ITEMP,KHR1),(DAI,JDAI),(DAII,JDAII)                     000002
00111                                     DOUBLE PRECISION DBLANK / '                      000002
00111           C * * *  VALIDITY CHECKS                                                   000002
00113                 IF (NN.EQ.0) GO TO 990                                               000002
00115                 N = IABS(NN)                                                         000004
00116                 IF (NN.LT.0) GO TO 300                                               000006
00116           C * * *  SWITCH CHARACTERS                                                 000006
00120                 KHR1 = DBLANK                                                        000011
00121                 KHR2 = 0                                                             000013
00122                 LIM = 12*N                                                           000015
00123             200 K2 = 1                                                               000021
00124                 DO 210 J=1,LIM                                                       000022
00127                 N2 = LIM-K2+1                                                        000027
00130                 I = ISCAN (KHR1,1,1,IA,K2,N2,M1)                                     000033
00131                 IF (I.EQ.0) GO TO 220                                                000051
00133                 CALL PUT(IA,I,KHR2)                                                  000053
00134                 IF (I.GE.LIM) GO TO 220                                              000060
00136             210 K2 = I+1                                                             000064
00140             220 IF (KHR1.EQ.0) GO TO 990                                             000072
00140           C * * *  SORT THE ARRAY                                                    000072
00142             300 M = N                                                                000075
00143             320 M = M/2                                                              000077
00144                 IF (M.LE.0) GO TO 400                                                000101
00146                 K = N-M                                                              000103
00147                 DO 340 J=1,K                                                         000106
00152                 I = J                                                                000112
00153             330 II = I+M                                                             000115
00154                 DAI = IA(I)                                                          000124
00155                 DAII = IA(II)                                                        000126
00156                 ITEMP = JDAI - JDAII                                                 000130
00157                 IF (ITEMP.LE.0) GO TO 340                                            000140
00161                 DTEMP = IA(I)                                                        000143
00162                 IA(I) = IA(II)                                                       000145
00163                 IA(II) = DTEMP                                                       000147
00164                 I = I-M                                                              000150
00165                 IF (I.GT.0) GO TO 330                                                000153
00167             340 CONTINUE                                                             000157
00171                 GO TO 320                                                            000157
00171           C * * *  SWITCH CHARACTERS BACK                                            000157
00172             400 IF (NN.LT.0) GO TO 990                                               000161
00174                 KHR1 = 0                                                             000163
00175                 KHR2 = DBLANK                                                        000165
00176                 GO TO 200                                                            000167
00177             990 CONTINUE                                                             000171
00200                 RETURN                                                               000171
00201                 END $ ******************************************                     000214
```

```
FUNCTION DAND          ENTRY POINT 000023

STORAGE USED  CODE(1) 000025; DATA(0) 000015; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000 D 000000 DAND       0000   000011 INJP$      0000 I 000002 I1         0000 I 000004 I2         0000 I 000006 I3
 0000 D 000002 T1         0000 D 000004 T2         0000 D 000006 T3


00101        1*           DOUBLE PRECISION FUNCTION DAND(E1, E2)                                        000000
00103        2*           DOUBLE PRECISION E1, E2                                                       000000
00104        3*           DOUBLE PRECISION T1, T2, T3                                                   000000
00105        4*           INTEGER I1(2), I2(2), I3(2)                                                   000000
00106        5*           EQUIVALENCE (I1(1), T1), (I2(1), T2), (I3(1), T3)                             000000
00107        6*           T1 = E1                                                                       000000
00110        7*           T2 = E2                                                                       000001
00111        8*           I3(1) = AND(I1(1), I2(1))                                                     000003
00112        9*           I3(2) = AND (I1(2), I2(2))                                                    000006
00113       10*           DAND = T3                                                                     000011
00114       11*           RETURN                                                                        000013
00115       12*           END                                                                           000024
```

```
FUNCTION DCMPL        ENTRY POINT 000017

STORAGE USED  CODE(1) 000021; DATA(0) 000012; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000 D 000000 DCMPL      0000   000006 INJP$     0000 I 000002 I1         0000 I 000004 I2         0000 D 000002 T1
 0000 D 000004 T2


00101       1*          DOUBLE PRECISION FUNCTION DCMPL(E1)                                           000000
00103       2*          DOUBLE PRECISION E1                                                           000000
00104       3*          DOUBLE PRECISION T1, T2                                                       000000
00105       4*          INTEGER I1(2), I2(2)                                                          000000
00106       5*          EQUIVALENCE (I1(1), T1), (I2(1), T2)                                          000000
00107       6*          T1 = E1                                                                       000000
00110       7*          I2(1) = COMPL(I1(1))                                                          000001
00111       8*          I2(2) = COMPL(I1(2))                                                          000003
00112       9*          DCMPL = T2                                                                    000005
00113      10*          RETURN                                                                        000007
00114      11*          END a DCMPL                                                                   000020
```

```
FUNCTION DOR          ENTRY POINT 000023

STORAGE USED  CODE(1) 000025; DATA(0) 000015; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000 D 000000 DOR        0000   000011 INJP$     0000 I 000002 I1        0000 I 000004 I2        0000 I 000006 I3
 0000 D 000002 T1         0000 D 000004 T2        0000 D 000006 T3


00101      1*           DOUBLE PRECISION FUNCTION DOR(E1, E2)                                         000000
00103      2*           DOUBLE PRECISION E1, E2                                                       000000
00104      3*           DOUBLE PRECISION T1, T2, T3                                                   000000
00105      4*           INTEGER I1(2), I2(2), I3(3)                                                   000000
00106      5*           EQUIVALENCE (I1(1), T1), (I2(1), T2), (I3(1), T3)                             000000
00107      6*           T1 = E1                                                                       000000
00110      7*           T2 = E2                                                                       000001
00111      8*           I3(1) = OR(I1(1), I2(1))                                                      000003
00112      9*           I3(2) = OR (I1(2), I2(2))                                                     000006
00113     10*           DOR = T3                                                                      000011
00114     11*           RETURN                                                                        000013
00115     12*           END @ DOR                                                                     000024
```

```
SUBROUTINE DUMPPF    ENTRY POINT 000255


STORAGE USED  CODE(1) 000276; DATA(0) 000102; BLANK COMMON(2) 000000


EXTERNAL REFERENCES (BLOCK, NAME)

 0003    READMS
 0004    STPMOV
 0005    GETCOD
 0006    KOMSTR
 0007    NWDU$
 0010    NIO2$
 0011    NIO1$
 0012    NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000     000032 101F       0000     000023 11F        0000     000035 113F       0001     000031 121G       0001     000047 130G
 0001     000135 151G       0001     000153 162G       0001     000142 200L       0000     000036 201F       0001     000203 620L
 0000     000043 631F       0001     000234 640L       0000     000026 65F        0000 D 000011 DBLANK      0000 D 000000 DIM
 0000 D 000016 DMAX         0000 D 000004 DPFNAM       0000 I 000010 I          0000 I 000002 IDIM        0000     000057 INJP$
 0000 I 000022 ISYMB        0000 I 000015 J            0000 I 000021 K          0006 I 000000 KOMSTR      0000 I 000003 MAX
 0000 I 000023 MAXMI        0000 D 000006 PFNAME       0000 D 000013 PINDEX



00101        1*            SUBROUTINE DUMPPF(CMPNTS,ICPMAX,TYPES,AINPUT)                                000005
00103        2*                           IMPLICIT DOUBLE PRECISION (A - Z)                             000005
00104        3*                           IMPLICIT INTEGER (I - N)                                      000005
00105        4*            DOUBLE PRECISION DPFNAM / 'PFNAME      ' /                                   000005
00105        5*      C  VERSION 1.                       REVISED  MAY 21 1976                           000005
00105        6*      C  PURPOSE   DUMP PERMANENT FILE ONTO TAPE 9 IN INPUT FORMAT                       000005
00105        7*      C  CALL SEQUENCE   CMPNTS - COMPONENT NAME LIST                                    000005
00105        8*      C                  ICPMAX - NUMBER OF COMPONENTS                                   000005
00105        9*      C                  DCPMAX - NUMBER OF COMPONENTS (DBLE PRCSN)                      000005
00105       10*      C                  TYPES  - DATA TYPE NAMES                                        000005
00105       11*      C                  AINPUT - NAME ARRAY WORK STORAGE ARRAY                          000005
00105       12*      C  DESIGNED BY  J.D. BURROUGHS                      DEC 1975                       000005
00107       13*            DOUBLE PRECISION CMPNTS(1),TYPES(3),AINPUT(1)                                000005
00110       14*            WRITE(9,11)                                                                  000005
00112       15*      11    FORMAT('NEW FILE')                                                           000012
00112       16*      C ---       LOAD FILE NAME                                                         000012
00113       17*            CALL READMS(18,PFNAME,1,DPFNAM)                                              000012
00114       18*            WRITE(9,65)PFNAME                                                            000020
00117       19*      65    FORMAT('FILE NAME=',A10)                                                     000031
00117       20*      C --->      SCAN ALL COMPONENTS                                                    000031
00120       21*            DO 640 I=1,ICPMAX                                                            000031
00120       22*      C --->      LOAD COMPONENT NAME                                                    000031
00123       23*            DOUBLE PRECISION DBLANK                                                      000031
```

```
00125                 PINDEX=DBLANK                                                       000031
00126                 CALL STRMOV(CMPNTS(I),1,2,PINDEX,1)                                 000032
00126           C --->        SCAN THREE TYPES OF LISTS REQ"D FOR EACH COMPONENT          000032
00127                 DO 620 J=1,3                                                        000047
00132                 CALL STRMOV(TYPES(J),1,4,PINDEX,3)                                  000047
00132           C --->        READ LISTS FROM FILE 19                                     000047
00133                 CALL READMS(18,DMAX,1,PINDEX)                                       000062
00134                 MAX = DMAX                                                          000066
00135                 CALL READMS(18,AINPUT,MAX,PINDEX)                                   000074
00136                 MAXM1=MAX-1                                                         000102
00136           C --->        WRITE INPUT LIST NAME AND NUMBER OF INPUTS (OUTPUTS)        000102
00137                 WRITE(9,101)PINDEX,MAXM1                                            000105
00143           101   FORMAT(A7," = ",I4)                                                 000114
00143           C --->        TEST FOR TABLE INPUTS                                       000114
00144                 IF(J.EQ.3)GO TO 200                                                 000114
00144           C --->        INPUT AND OUTPUT LIST TYPES                                 000114
00146                 IF(MAX.GT.1)WRITE(9,111)(AINPUT(K),K=2,MAX)                         000117
00155           111   FORMAT(6A10)                                                        000140
00156                 GO TO 620                                                           000140
00156           C --->        TABLE INPUT FORMAT                                          000140
00157           200   IF(MAX.LE.1)GO TO 620                                               000142
00161                 DO 240 K=2,MAX                                                      000145
00164                 CALL GETCOD(5,AINPUT(K),IDIM)                                       000153
00165                 DIM=IDIM                                                            000161
00165           C --->        WRITE TABLE NAME AND MAX. DIMENSION                         000161
00166                 WRITE(9,201)AINPUT(K),DIM                                           000167
00172           201   FORMAT(A3,F7.0)                                                     000202
00173           240   CONTINUE                                                            000202
00175           620   CONTINUE                                                            000202
00175           C --->        TEST FOR SYMBOL NUMBER                                      000202
00177                 IF(KOMSTR(CMPNTS(I),9,2,DBLANK,1).EQ.0)GO TO 640                    000202
00177           C --->        GET SYMBOL NUMBER FROM COMPONENT NAME                       000202
00201                 CALL GETCOD(5,CMPNTS(I),ISYMB)                                      000215
00202                 WRITE(9,631)CMPNTS(I),ISYMB                                         000224
00206           631   FORMAT("SYMBOL, ",A2," = ",I5)                                      000235
00207           640   CONTINUE                                                            000235
00211                 RETURN                                                              000235
00212                 END @ ****************************************                     000275
```

```
MAIN PROGRAM   FILOAD

STORAGE USED  CODE(1) 001317; DATA(0) 002064; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   C10     000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0004   NXTPH
 0005   LEPPH
 0006   STPMOV
 0007   KONSTR
 0010   PUTCOD
 0011   WRITMS
 0012   NUMERC
 0013   ECPGUB
 0014   READMS
 0015   GETCOD
 0016   COMPAT
 0017   CSORT
 0020   EUMPPF
 0021   MINTRS
 0022    IDEF$
 0023    XRDU$
 0024    KIO3$
 0025    KIO2$
 0026    NKDU$
 0027    KIO1$
 0030    XLPR2$
 0031    DSORT
 0032    NSTOP$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000024 100L     0000    001233 101F     0001    000064 130L     0001    000145 136L     0000    001234 137F
 0001    000222 140L     0001    000252 146L     0001    000263 150L     0001    000304 160L     0000    001244 161F
 0001    000315 170L     0000    001266 181F     0001    000324 200L     0001    000360 208L     0001    000361 210L
 0000    001310 211F     0001    000404 212L     0001    000125 215G     0001    000430 220L     0001    000174 235G
 0001    000230 247G     0001    000456 300L     0001    000347 310G     0001    000471 315L     0001    000366 317G
 0001    000474 320L     0001    000477 330L     0001    000535 334L     0001    000557 350L     0001    000513 353G
 0001    000616 360L     0000    001312 361F     0001    000544 366G     0001    000625 400L     0001    000653 414G
 0001    000715 432G     0001    001010 453G     0001    000702 500L     0001    001117 512G     0001    001141 517G
 0000    001326 521F     0001    001214 533G     0001    001054 540L     0000    001353 541F     0001    001071 600L
 0001    001265 700L     0001    001267 720L     0001    001276 750L     0001    001312 999L     0001    001313 9992L
 0001    001314 9994L    0000 D  001217 AIN      0000 D  000343 AINPUT   0000 D  000765 CHANDS   0000 D  001366 CHEVES
 0000 D  001150 COMNAM   0000 D  000004 DPLANK   0000 D  000341 DCMPTS   0000 D  001360 DCMP1    0000 D  001211 DCPMAX
 0000 D  000020 DIM      0000 D  001226 DMAX     0000 D  000007 DPFNAM   0000 D  001160 HIKPT    0000 D  001162 HOUTP
 0000 D  001164 HTAPS    0000 I  001201 I        0000 I  000006 IBLNK    0000 I  001155 ICMMAX   0000 I  000121 ICMMOD
 0000 I  001230 ICMP1    0000 D  000321 ICON     0000 I  001156 ICPMAX   0000 I  001154 ICPMOD   0000 I  003225 ID
 0003    000002 IDIAG    0000 I  000102 IDIN     0000 I  001172 IDMIX    0003 I  000000 IFEAD    0000 I  001213 ITYPE
```

```
0003 I 001174 ITYPE     0003 I 000001 IWRITE    0000 I 001166 I18       0000 I 001167 I19       0000 I 001202 J
0007 I 000000 KONSTR    0000 I 001157 LIST      0000 I 001175 LOAD      0000 I 000003 MAX       0000 I 001213 MAXCOM
0003 I 001205 N         0003 D 000011 NAMES     0000 I 001176 NCOMP     0000 I 001214 NI        0000 I 001215 NO
0000 I 001216 NT        0000 I 001173 NTASK     0000 D 001221 OUT       0000 D 000507 OUTPUT    0000 D 001231 PFNAME
0000 D 001203 PHRS      0000 D 001171 PINDEX    0000 D 001206 SYME      0000 D 001223 TAB       0000 D 000653 TABLE
0000 D 001152 TYPE      0000 D 000677 TYPES     0000 D 001177 VALUE



00101                       IMPLICIT DOUBLE PRECISION (A - Z)                                000000
00103                       IMPLICIT INTEGER (I, J, K, L, M, N)                              000001
00104                       DOUBLE PRECISION DBLANK / '                                      000001
00106              DATA IBLNK                                                                000001
00110              DOUBLE PRECISION DPFNAM / 'PFNAME     ' /                                 000001
00112              DOUBLE PRECISION NAMES, ICOM                                              000001
00113              DOUBLE PRECISION DCMPTS / 'CMPNTS     ' /                                 000001
00113        C     PROGRAM FILOAD(INPUT=100,OUTPUT=200,TAPE5=INPUT,TAPE6=OUTPUT,             000001
00113        C    1 TAPE3,TAPE78,TAPE79,TAPE9)                                               000001
00113        C  VERSION 3.1                          REVISED  OCT 13 1976                    000001
00113        C  PURPOSE   THIS PROGRAM ADDS INPUT,OUTPUT,AND TABLE NAME LISTS                000001
00113        C               TO THE EASY PROGRAM PERMANENT FILE.                             000001
00113        C  METHOD   DATA IS READ FROM TAPE3 AND LOADED INTO THE PERMANENT FILE.         000001
00113        C               THE DATA FORMAT IS  FIRST PHRASE =RECORD NAME,                  000001
00113        C                                    SECOND PHRASE = NO. WORDS IN RECORD        000001
00113        C                  THE INPUT AND OUTPUT NAME LISTS INPUT                        000001
00113        C                  DATA IS FIXED FIELD WITH A 8A10 FORMAT.                      000001
00113        C                  THE TABLE LIST INPUT DATA IS A10,G7.0                        000001
00113        C                  FORMAT.                                                      000001
00113        C                  THE NUMERIC INPUT SPECIFIES THE MAXIMUM                      000001
00113        C                  TABLE DIMENSION.  NEGATIVE VALUES                            000001
00113        C                  INDICATE SINGLE INDEPENDENT VARIABLE TABLES.                 000001
00113        C  DESIGNED BY  J.D.BURROUGHS                       MAY 1974                    000001
00115              DIMENSION NAMES(100),CMPNTS(15),AINPUT(50),OUTPUT(50),                    000001
00115            1 TABLE(110),ICOM(8),TYPES(3),CMNDS(6),ICMMOD(15)                           000001
00116              COMMON/CIO/IREAD,IWRITE,IDIAG                                             000001
00117              EQUIVALENCE (DCMPT,CMPNTS)                                                000001
00120              DATA COMNAM / '               /                                           000001
00122              DATA TYPES(1) / 'INPT       ' /                                           000001
00124              DATA TYPES(2) / 'OUTP       ' /                                           000001
00126              DATA TYPES(3) / 'TABS       ' /                                           000001
00130              DATA CMNDS(1) / 'LIST STAND  ' /                                          000001
00132              DATA CMNDS(2) / 'PURGE       ' /                                          000001
00134              DATA CMNDS(3) / 'DUMP FILE   ' /                                          000001
00136              DATA CMNDS(4) / 'SYMBOL      ' /                                          000001
00140              DATA CMNDS(5) / 'NEW FILE    ' /                                          000001
00142              DATA CMNDS(6) / 'FILE NAME   ' /                                          000001
00144              DATA TYPE / '               ' /, ICPMOD / 0 /, ICMMAX / 6 /               000001
00150              DATA ICPMAX / -1 /                                                        000001
00152              DATA LIST / 0 /                                                           000001
00154              DATA HINPT / 'INPT         ' /                                            000001
00156              DATA HOUTP / 'OUTP         ' /                                            000001
00160              DATA HTABS / 'TABS         ' /                                            000001
00162              IREAD=5                                                                   000001
00163              IWRITE=6                                                                  000003
00163        C --->       OPEN MASS STORAGE FILE                                             000003
00164              DEFINE FILE 18(2810,302,U,I18),19(2810,302,U,I19)                         000005
```

```
00164     48*     C --->       READ COMMAND CARD                                              000005
00166     49*       100 CONTINUE                                                                000024
00167     50*            READ(3, 101, END = 500, ERR = 999) ICOM                              000024
00172     51*     101    FORMAT(8A10)                                                         000035
00173     52*     120    INDEX=1                                                              000035
00173     53*     C --->       LOCATE NEXT PHRASE                                             000035
00174     54*            CALL NXTPH(ICOM,INDEX,PINDEX)                                        000037
00175     55*            IF(PINDEX.EQ.DBLANK)GO TO 100                                        000044
00175     56*     C --->       SEARCH COMMAND LIST                                            000044
00177     57*            CALL LCMPH(PINDEX,CMMNDS,ICMMAX,1,NTASK)                             000047
00177     58*     C --->       BRANCH TO 300 IF COMMAND IS IDENTIFIED                         000047
00200     59*            IF(NTASK.NE.0)GO TO 300                                              000056
00200     60*     C --->       TEST IF COMPONENT NAME LIST HAS BEEN READ                      000056
00202     61*            IF(ICPMAX.LT.0)GO TO 400                                             000060
00202     62*     C --->       GET LIST TYPE                                                  000060
00204     63*     130    CALL STRMOV(PINDEX,3,4,TYPE,1)                                       000064
00204     64*     C --->       COMPARE TYPE TO 3 ACCEPTABLE TYPES                             000064
00205     65*            CALL LCMPH(TYPE,TYPES,3,1,ITYPE)                                     000072
00205     66*     C --->       TEST IF TYPE WAS IDENTIFIED                                    000072
00206     67*            IF(ITYPE.EQ.0)GO TO 160                                              000103
00210     68*            LOAD=1                                                               000103
00210     69*     C --->       GET COMPONENT NAME                                             000103
00211     70*            CALL STRMOV(PINDEX,1,2,COMNAM,1)                                     000105
00211     71*     C ---        BYPASS SEARCH IF COMPONENT COUNT < 1                           000105
00212     72*            IF(ICPMAX.LT.1)GO TO 136                                             000114
00212     73*     C --->       SEARCH COMPONENT NAME LIST                                     000114
00214     74*            DO 132 NCOMP=1,ICPMAX                                                000120
00217     75*            IF(NCMSTR(CMPNTS(NCOMP),1,2,COMNAM,1).EQ.0)GO TO 140                 000127
00221     76*     132    CONTINUE                                                             000145
00221     77*     C --->       NEW COMPONENT                                                  000145
00223     78*     136    ICPMAX=ICPMAX+1                                                      000145
00224     79*            NCOMP=ICPMAX                                                         000147
00224     80*     C --->       ADD DEFAULT SYMBOL NO. = 2001                                  000147
00225     81*            CALL PUTCOD(S,COMNAM,2001)                                           000150
00225     82*     C --->       ADD COMPONENT NAME TO LIST                                     000150
00226     83*            CMPNTS(ICPMAX)=COMNAM                                                000155
00227     84*            WRITE(6,137)COMNAM                                                   000161
00232     85*     137    FORMAT(3X,A4,'WILL BE ADDED AS A NEW COMPONENT')                     000167
00232     86*     C ---        LOAD NAME ARRAYS WITH DEFAULT VALUES OF D NAMES                000167
00233     87*            VALUE=COMNAM                                                         000167
00234     88*            DO 138 I=1,3                                                         000174
00234     89*     C ---        ADD TYPE NAME TO COMPONENT NAME                                000174
00237     90*            CALL STRMOV(TYPES(I),1,4,VALUE,3)                                    000176
00240     91*            NAMES(1)=I                                                           000207
00241     92*            CALL WRITMS(18,NAMES,1,VALUE)                                        000211
00242     93*     138    CONTINUE                                                             000222
00242     94*     C ---        BYPASS SEARCH IF MODIFIED COMPONENT COUNTER = 0                000222
00244     95*     140    IF(ICPMOD.EQ.0)GO TO 146                                             000222
00244     96*     C --->       TEST IF COMPONENT HAS BEEN MODIFIED BEFORE                     000222
00246     97*            DO 144 I=1,ICPMOD                                                    000223
00251     98*            J=ICPMOD(I)                                                          000231
00252     99*            IF(NCMSTR(COMNAM,1,2,CMPNTS(J),1).EQ.0)GO TO 150                     000233
00254    100*     144    CONTINUE                                                             000252
00256    101*     146    ICPMOD=ICPMOD+1                                                      000252
00256    102*     C --->       ACCUMULATE COMP. NOS. OF COMPONENTS MODIFIED                   000252
00257    103*            ICMPOD(ICPMOD)=NCOMP                                                 000255
00257    104*     C --->       GET NEXT PHRASE WHICH CONTAINS NO. OF ITEMS IN LIST            000256
```

```
00260     150   CALL NXTPH(ICOM,INDEX,PHRS)                                     000260
00260     C --->        TEXT FOR NUMERIC FIRST CHARACTER                        000260
00261           CALL NUMERC(PHRS, $180)                                         000264
00261     C --->        CONVERT HOLLORITH TO INTEGER                            000264
00262           CALL FCDBND(VALUE,PHRS)                                         000270
00263           N=VALUE                                                         000274
00264           GO TO 200                                                       000302
00265     160   WRITE(6,161)PINDEX,TYPE                                         000304
00271     161   FORMAT(/22H *** WARNING ***  IN  ,A8,2X,A10,                    000312
00271          1"ISN"T A RECOGNIZED NAME LIST TYPE.  NAME LIST WILL NOT BE LOADED"   000312
00271          1)                                                               000312
00272           LOAD=0                                                          000312
00273           GO TO 150                                                       000313
00274     180   WRITE(6,181)PHRS                                                000315
00277     181   FORMAT(/16H *** WARNING ***,A10,                                000322
00277          1"ISN"T A VALID NUMBER OF NAMES FOR NAME LIST                    000322
00277          2"NAME LIST WILL NOT BE LOADED")                                 000322
00300           GO TO 150                                                       000322
00301     200   N=N+1                                                           000324
00302           IF(N.LE.1) GO TO 220                                            000326
00304           IF(TYPE.EQ.TYPES(3))GO TO 210                                   000331
00304     C --->        READ NAMES FROM TAPE3                                   000331
00306           READ(3,101,ERR=9992)(NAMES(I),I=2,N)                            000336
00314       208 CONTINUE                                                        000360
00315           GO TO 220                                                       000360
00315     C --->        READ TABLE NAMES                                        000360
00316     210   DO 215 I=2,N                                                    000361
00321           READ(3,211,ERR=9994)NAMES(I),DIM                                000371
00325     211   FORMAT(A3,G7.0)                                                 000404
00326     212   CONTINUE                                                        000404
00327           IDIM=DIM                                                        000406
00330           CALL PUTCOD(5,NAMES(I),IDIM)                                    000414
00331     215   CONTINUE                                                        000430
00333     220   IF(N.LT.1)N=1                                                   000430
00335           NAMES(1)=N                                                      000435
00335     C --->        WRITE NAMES ON MASS STORAGE PERMANENT FILE              000435
00336           IF(LOAD.EQ.1)CALL WRITNS(IB,NAMES,N,PINDEX)                     000443
00343           GO TO 100                                                       000454
00343     C --->        COMMAND INTERPRETATION                                  000454
00341       300 CONTINUE                                                        000456
00342           GO TO(310,320,400,320,700,750),NTASK                            000456
00342     C =====================  LIST STANDARD COMPONENTS === NTASK =1         000456
00343     310   LIST=1                                                          000471
00344           GO TO 100                                                       000472
00344     C =====================  PURGE  NTASK = 2  OR  SYMBOL  ==  NTASK = 4  000472
00345     320   IF(ICPMAX.LT.0)GO TO 400                                        000474
00345     C --->        GET COMPONENT NAME                                      000474
00347     330   CALL NXTPH(ICOM,INDEX,COMNAM)                                   000477
00350           IF(COMNAM.EQ.DBLANK)GO TO 100                                   000503
00350     C --->        LOCATE NAME IN COMPONENT NAME LIST                      000503
00352           DO 336 NCOMP=1,ICPMAX                                           000513
00355           IF(KOMSTR(CMPNTS(NCOMP),1,2,COMNAM,1).EQ.0)GO TO 338            000515
00357     336   CONTINUE                                                        000532
00361           NCOMP=0                                                         000532
00362           GO TO 300                                                       000533
00363     338   IF(NTASK.NE.2)GO TO 350                                         000535
00363     C --->        MOVE COMPONENT NAMES OVER ONE TO OVERWRITE PURGED NAME  000535
```

```
00365              DO 340 I=NCOMP,ICPMAX                                      000537
00370        340   CMPNTS(I)=CMPNTS(I+1)                                      000546
00370        C --->        REDUCE NO. OF COMPONENTS                           000546
00372              ICPMAX=ICPMAX-1                                            000552
00373              GO TO 330                                                  000555
00374        350   CALL NXTPH(ICOM,INDEX,SYMB)                                000557
00375              CALL FCODUB(SYMB,SYMB)                                     000563
00376              ISYMB=SYMB                                                 000567
00377              CALL PUTCOD(5,CMPNTS(NCOMP),ISYMB)                         000575
00400              ICPMOD=ICPMOD+1                                            000606
00401              ICMPOD(ICPMOD)=NCOMP                                       000612
00402              GO TO 330                                                  000614
00403        360   WRITE(6,361)COMNAM                                         000616
00406        361   FORMAT(/13H *** WARNING ***  CAN'T IDENTIFY ,A4,           000623
00406             1' AS A STANDARD COMPONENT')                                000623
00407              GO TO 330                                                  000623
00407        C --->        GET COMPONENT NAME LIST FROM FILE 18               000625
00410        400   CALL READNS(18,DCPMAX,1,DCMPTS)                            000625
00411              ICPMAX = DCPMAX                                            000632
00412              CALL READNS(18,CMPNTS,ICPMAX,DCMPTS)                       000640
00412        C --->        SHIFT NAMES OVER 1 WORD TO ELLIMINATE NO. OF WORDS 000643
00413              DO 420 I=2,ICPMAX                                          000646
00416        420   CMPNTS(I-1)=CMPNTS(I)                                      000655
00420              ICPMAX=ICPMAX-1                                            000661
00421              IF(NTASK.LE.5)GO TO 130                                    000664
00423              GO TO(130,330,720,330,130),NTASK                           000667
00423        C --->        LIST COMPONENTS MODIFIED IF LIST=1                 000667
00424        500   MAXCOM=ICPMOD                                              000702
00425              IF(LIST.NE.1)GO TO 600                                     000703
00425        C --->        IF NO COMPS. MODIFIED, SKIP LISTING                000703
00427              IF(MAXCOM.LE.0)GO TO 600                                   000706
00427        C --->        SCAN COMPONENTS SPECIFIED                          000706
00431              DO 560 I=1,MAXCOM                                          000710
00434              J=1                                                        000717
00435              J=ICMPOD(I)                                                000721
00436              COMNAM=CMPNTS(J)                                           000725
00437        520   CALL GETCOD(5,COMNAM,ISYMB)                                000727
00440              WRITE(6,521)I,COMNAM,ISYMB                                 000734
00445        521   FORMAT(//' COMPONENT NO.',I3,'   NAME = ',A2,'   SYMBOL NO. = ',I3   000744
00445             1/' INPUTS',7X,'OUTPUTS',6X,'TABLES',7X,'DIMENSION')        000744
00445        C --->        GET INPUT,OUTPUT,AND TABLE NAMES                   000744
00446              CALL COMDAT(COMNAM,NINPT,NI,AINPUT)                        000744
00447              CALL COMDAT(COMNAM,NOUTP,NO,OUTPUT)                        000752
00450              CALL COMDAT(COMNAM,NTABS,NT,TABLE)                         000760
00451              MAX=MAX0(NI,NO,NT,1)                                       000766
00451        C --->        SCAN LONGEST LIST OF NAMES                         000766
00452              DO 550 J=1,MAX                                             001004
00452        C --->        BLANK NAMES                                        001004
00455              AIN=DBLANK                                                 001013
00456              OUT=DBLANK                                                 001015
00457              TAB=DBLANK                                                 001016
00460              ID = IBLNK                                                 001017
00461              IF(J.LE.NI)AIN=AINPUT(J)                                   001021
00463              IF(J.LE.NO)OUT=OUTPUT(J)                                   001030
00465              IF(J.GT.NT)GO TO 540                                       001037
00467              TAB=TABLE(J)                                               001043
00467        C --->        GET TABLE DIMENSION                                001043
```

```
00470              CALL GETCOD(S,TAB,ID)                                           001046
00471         540  WRITE(6,541)AIN,OUT,TAB,ID                                      001054
00477         541  FORMAT(3X,A10,3X,A10,3X,A8,5X,I4)                               001071
00500         550  CONTINUE                                                        001071
00502         560  CONTINUE                                                        001071
00502         C --->       DEGAS MASS STORAGE FILE                                 001071
00502         C --->       IF NO COMPONENTS EXIST, CAUSE ABEND TO PREVENT DEGASSING 001071
00504         600  CONTINUE                                                        001071
00505              AIN=-1.                                                         001071
00506              IF(ICPMAX.LE.0)I=SQRT(AIN)                                      001072
00506         C ---        SORT COMPONENTS INTO ALPHABETICAL ORDER                 001072
00510              CALL CSORT(CMPNTS,ICPMAX)                                       001105
00510         C --->       SCAN ALL COMPONENTS                                     001105
00511              DO 640 I=1,ICPMAX                                               001111
00511         C --->       LOAD COMPONENT NAME                                     001111
00514              PINDEX=BLANK                                                    001122
00515              CALL STRMOV(CMPNTS(I),1,2,PINDEX,1)                             001124
00515         C --->       SCAN THREE TYPES OF LISTS REQ'D FOR EACH COMPONENT      001124
00516              DO 640 J=1,3                                                    001141
00521              CALL STRMOV(TYPES(J),1,4,PINDEX,3)                              001143
00521         C --->       READ LISTS FROM FILE 19                                 001143
00522              CALL READMS(18,DMAX,1,PINDEX)                                   001154
00523              MAX = DMAX                                                      001162
00524              CALL READMS(18,AINPUT,MAX,PINDEX)                               001170
00524         C --->       WRITE LISTS ONTO FILE 19                                001170
00525              CALL WRITMS(19,AINPUT,MAX,PINDEX)                               001176
00526         640  CONTINUE                                                        001210
00526         C --->       SHIFT COMPONENT NAMES OVER 1 WORD                       001210
00531              J=ICPMAX                                                        001210
00532              DO 660 I=1,ICPMAX                                               001214
00535              CMPNTS(J+1)=CMPNTS(J)                                           001215
00536         660  J=J-1                                                           001217
00536         C --->       ADD NO. OF COMPONENTS + 1 AS FIRST WORD IN LIST         001217
00540              DCMP1=ICPMAX+1                                                  001223
00541              ICMP1 = DCMP1                                                   001233
00541         C --->       STORE COMPONENT NAME LIST                               001233
00542              CALL WRITMS(19,CMPNTS,ICMP1,DCMPTS)                             001240
00542         C ---        STORE PFNAME                                            001240
00543              CALL READMS(18,PFNAME,1,DPFNAM)                                 001246
00544              CALL WRITMS(19,PFNAME,1,DPFNAM)                                 001254
00545              STOP                                                            001262
00545         C =========  NEW FILE  === NTASK = 5                                 001262
00546         700  ICPMAX=0                                                        001265
00547              GO TO 100                                                       001265
00547         C ==========  DUMP FILE  === NTASK = 3                               001265
00550         720  CALL DUMPPF(CMPNTS,ICPMAX,TYPES,AINPUT)                         001267
00551              GO TO 150                                                       001274
00551         C ==========  FILE NAME  === NTASK = 6                               001274
00552         750  CALL NXTPH(ICDN,INDEX,PFNAME)                                   001276
00553              CALL WRITMS(18,PFNAME,1,DPFNAM)                                 001302
00554              GO TO 100                                                       001310
00555           999 GO TO 150                                                      001312
00556          9992 GO TO 209                                                      001313
00557          9994 GO TO 212                                                      001314
00560              END @ ****************************************                 001316
```

```
SUBROUTINE GETCOD    ENTRY POINT 000103

STORAGE USED  CODE(1) 000115; DATA(0) 000021; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003    SHIFT
 0004    DAND
 0005    DOR
 0006    DCMPL
 0007    NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000074 100L      0000 D 000006 CODE      0004 D 000000 DAND      0000 D 000000 DBLANK    0006 D 000000 DCMPL
 0005 D 000000 DOR        0000 I 000006 IA        0000   000011 INJP$     0000 I 000004 ISHIFT    0000 I 000005 IWORD
 0000 D 000002 MASK       0003 D 000000 SHIFT


00100      CGETCOD                                                                       000002
00101            SUBROUTINE GETCOD(N,IARRAY,ICODE)                                       000002
00103                          IMPLICIT DOUBLE PRECISION (A - Z)                         000002
00104            IMPLICIT INTEGER (I - N)                                                000002
00105            INTEGER ICODE                                                           000002
00106                          DOUBLE PRECISION DBLANK                                   000002
00106      C  PURPOSE   RETRIEVE A 4 DIGIT CODE.  VALUE OF CODE MUST BE BETWEEN          000002
00106      C            2047*, STORED 5 CODES/WORD, FROM AN ARRAY OF PARAMETER           000002
00106      C            CODES.  THIS ROUTINE IS USED TO REDUCE THE STORAGE REQUIRED      000002
00106      C            TO STORE THE I/O CODE LISTS FOR EACH ANALYSIS MODULE.            000002
00106      C  CALL SEQUENCE    N   LOCATION OF CODE IN ARRAY IARRAY (5 CODES/WORD).       000002
00106      C                       IARRAY   INTEGER ARRAY WHICH RECIEVES CODE NUMBER.    000002
00106      C                       ICODE    VALUE OF CODE INPUT TO ROUTINE.              000002
00110            DOUBLE PRECISION SHIFT                                                  000002
00111            DIMENSION IARRAY(1)                                                     000002
00112            DOUBLE PRECISION MASK / 00003000000000000000777 70000                   000002
00114            DOUBLE PRECISION CODE                                                   000002
00115            INTEGER IA(2)                                                           000002
00116            EQUIVALENCE (IA(1), CODE)                                               000002
00117            INTEGER ISHIFT                                                          000002
00120            INTEGER IWORD                                                           000002
00120      C  DETERMINE WHICH WORD IN ARRAY CONTAINS THE NTH CODE.                       000002
00121            IWORD=(N-1)/5+1                                                         000002
00121      C  DETERMINE THE NUMBER OF BITS TO SHIFT CODE TO RIGHT MOST 12 BITS.          000002
00122            ISHIFT = (MOD(N-1, 5) -4) * 12                                          000011
00122      C  SHIFT CODE BITS TO RIGHT HAND POSITION.                                    000011
00123            CODE=SHIFT(IARRAY(IWORD),ISHIFT)                                        000017
00123      C  MASK OUT UNWANTED BITS TO LEFT OF CODE.                                    000017
00124            CODE=DAND(MASK, CODE)                                                   000020
```

```
00125     30*          ICODE = 0                                                   000035
00126     31*          FLD(24,12,ICODE) = FLD(12,12,IA(2))                         000036
00126     32*     C  TEST SIGN BIT.                                                000036
00127     33*          IF(ICODE.LT.2048) GO TO 100                                 000045
00127     34*     C  RESTORE 1 BITS FOR NEGATIVE CODE.                             000045
00131     35*          ICODE=OR(ICODE, COMPL(MASK))                                000051
00132     36*          ICODE = -1                                                  000062
00133     37*          FLD(24,12,ICODE) = FLD(12,12,IA(2))                         000064
00134     38*     100  CONTINUE                                                    000074
00135     39*          RETURN                                                      000074
00136     40*          END ******************************************              000114
```

```
$SUBROUTINE GETT      ENTRY POINT 000055

STORAGE USED  CODE(1) 000063; DATA(0) 000015; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000037 100L       0000 D 000002 BLANKS    0000   000007 INJP$     0000 I 000001 IPOS      0000 I 000000 NWORD
 0000 D 000004 S1         0000 R 000004 S2


00101       1*             SUBROUTINE GETT(S, I, T)                                                000002
00101       2*      C                                                                              000002
00101       3*      C      GETT(S, I, T) EXTRACTS THE I'TH CHARACTER FROM THE STRING,              000002
00101       4*      C      STORED TEN CHARACTERS PER D/P  WORD, BEGINNING IN WORD                  000002
00101       5*      C      S(1), AND INSERTS IT, LEFT-JUSTIFIED, INTO WORD T                       000002
00103       6*             DOUBLE PRECISION S(1), T, S1                                            000002
00104       7*             DIMENSION S2(2)                                                         000002
00105       8*             EQUIVALENCE(S1,S2)                                                      000002
00105       9*      C                                                                              000002
00105      10*      C      DETERMINE D.P. WORD CONTAINING I'TH CHARACTER                           000002
00106      11*             NWORD=(I-1)/10 + 1                                                      000002
00107      12*             S1 = S(NWORD)                                                           000011
00107      13*      C                                                                              000011
00107      14*      C      DETERMINE THE RELATIVE POSITION OF 1ST BIT OF CHARACTER                 000011
00107      15*      C      IN 1108, DOUBLE PRECISION WORD.                                         000011
00110      16*             IPOS=MOD(I-1,10) * 6                                                    000015
00110      17*      C                                                                              000015
00110      18*      C      EXTRACT FROM S(NWORD) AND INSERT INTO T AFTER SETTING                   000015
00110      19*      C      T TO BLANKS (NOTICE - NO TYPE CONVERSION)                               000015
00111      20*             DOUBLE PRECISION BLANKS / '                                             000022
00113      21*             T = BLANKS                                                              000022
00114      22*             IF(IPOS.GT.35)GO TO 100                                                 000024
00116      23*             FLD(0,6,T)=FLD(IPOS,6,S2(1))                                            000027
00117      24*             RETURN                                                                  000033
00120      25*      100    IPOS = IPOS - 36                                                        000037
00121      26*             FLD(0,6,T) = FLD(IPOS,6,S2(2))                                          000041
00122      27*             RETURN                                                                  000045
00123      28*             END @ SUBROUTINE GETT                                                   000062
```

```
FUNCTION ISCAN       ENTRY POINT 000061

STORAGE USED  CODE(1) 000077; DATA(0) 000022; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   GETT
 0004   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000010 106G       0001   000020 112G       0001   000040 60CL       0000   000007 INJP$      0000 I 000000 ISCAN
 0000 I 000005 L          0000 I 000006 M          0000 D 000001 T1         0000 D 000003 T2


00101        1*           FUNCTION ISCAN(S1, K1, N1, S2, K2, N2, M1)                               000010
00101        2*     C                                                                             000010
00101        3*     C     EACH OF THE N1 CHARACTERS OF STRING S1, BEGIN-                          000010
00101        4*     C     NING WITH CHARACTER POSITION K1 (COUNTING                               000010
00101        5*     C     FROM LEFT TO RIGHT), IS COMPARED WITH (POSSIBLY)                        000010
00101        6*     C     EACH OF THE N2 CHARACTERS OF STRING S2, BEGIN-                          000010
00101        7*     C     NING WITH CHARACTER K2.                                                 000010
00101        8*     C                                                                             000010
00101        9*     C     IF A MATCH IS MADE, THEN M1 RETURNS WITH THE                            000010
00101       10*     C     CHARACTER POSITION IN S1 FOR WHICH A CORRES-                            000010
00101       11*     C     PONDING CHARACTER WAS FOUND IN S2.  M2, THE VALUE OF                    000010
00101       12*     C     THE FUNCTION ISCAN, IS RETURNED CONTAINING THE                          000010
00101       13*     C     POSITION IN STRING S2 OF THE MATCHED CHARACTER.                         000010
00101       14*     C                                                                             000010
00101       15*     C     IF NO MATCH IS MADE, THEN BOTH M1 AND M2 ARE                            000010
00101       16*     C     SET TO ZERO                                                             000010
00101       17*     C                                                                             000010
00101       18*     C     EX.  GIVEN STRING S1 CONTAINING 60 CHARACTERS (10 WORDS).               000010
00101       19*     C          TO FIND, THE BEGINNING OF A REAL OR INTEGER                        000010
00101       20*     C          CONSTANT EMBEDDED IN STRING S1                                     000010
00101       21*     C                                                                             000010
00101       22*     C     LET S2 BE THE STRING '+-.0123456789'                                    000010
00101       23*     C                                                                             000010
00101       24*     C     THEN, WRITE                                                             000010
00101       25*     C                                                                             000010
00101       26*     C          M2 = ISCAN(S1, 1, 60, S2 1, 13, M1)                                000010
00101       27*     C                                                                             000010
00103       28*           DOUBLE PRECISION S1(1), S2(1)                                           000010
00104       29*           DOUBLE PRECISION T1, T2                                                 000010
00104       30*     C                                                                             000010
00          31*           DO 800 L = K1, N1                                                       000010
00          32*           CALL GETT(S1, L, T1)                                                    000010
00111       33*           DO 600 M = K2, N2                                                       00
```

```
00114      34*            CALL GETT(S2, M, T2)                                                        000020
00114      35*      C     REPLACE "KOMPAR"                                                            000020
00114      36*      C     CALL KOMPAR(T1, T2, I)                                                      000020
00114      37*      C     IF (I .EQ. 0) M1 = L                                                        000020
00114      38*      C     IF (I .EQ. 0) ISCAN = M                                                     000020
00114      39*      C     IF (I .EQ. 2) RETURN                                                        000020
00115      40*            IF (T1.NE.T2)GO TO 600                                                      000025
00117      41*            M1 = L                                                                      000030
00120      42*            ISCAN = M                                                                   000032
00121      43*            RETURN                                                                      000034
00122      44*        600 CONTINUE                                                                    000043
00124      45*        800 CONTINUE                                                                    000043
00124      46*      C                                                                                 000043
00126      47*            M1 = 0                                                                      000043
00127      48*            ISCAN = 0                                                                   000044
00132      49*            RETURN                                                                      000045
00131      50*            END $ SUBROUTINE ISCAN                                                      000076
```

```
FUNCTION KOMSTR        ENTRY POINT 000051

STORAGE USED  CODE(1) 000066; DATA(0) 000014; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003    STRMOV
 0004    NLPR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000033 200L       0001    000037 300L       0000    000010 INJP$      0000 I 000001 I1         0000 I 000003 I2
 0000 I 000000 KOMSTR    0000 D 000001 T1          0000 D 000003 T2


00101      1*            FUNCTION KOMSTR(S1, K1, N, S2, K2)                                   000000
00101      2*     C                                                                           000000
00101      3*     C      EACH OF THE N1 CHARACTERS OF STRING S1, BEGIN-                       000000
00101      4*     C      NING WITH CHARACTER POSITION K1 (COUNTING                            000000
00101      5*     C      FROM LEFT TO RIGHT), IS COMPARED WITH (POSSIBLY)                     000000
00101      6*     C      EACH OF THE N2 CHARACTERS OF STRING S2, BEGIN-                       000000
00101      7*     C      NING WITH CHARACTER K2.                                              000000
00101      8*     C                                                                           000000
00101      9*     C      SET                                                                  000000
00101     10*     C           I = KOMSTR(S1, K1, N, S2, K2)                                   000000
00101     11*     C                                                                           000000
00101     12*     C             IF S1 = S2, THEN I = 0                                        000000
00101     13*     C             IF S1 '<' S2, THEN I = -1                                     000000
00101     14*     C             IF S1 '>' S2, THEN I = 1                                      000000
00101     15*     C                                                                           000000
00101     16*     C                                                                           000000
00103     17*            DOUBLE PRECISION S1, S2                                              000000
00104     18*            DOUBLE PRECISION T1,T2                                               000000
00105     19*            EQUIVALENCE(T1,I1),(T2,I2)                                           000000
00105     20*     C                                                                           000000
00106     21*            T1 = 0.0D0                                                           000003
00107     22*            T2 = 0.0D0                                                           000001
00110     23*            CALL STRMOV(S1,K1,N,T1,1)                                            000003
00111     24*            CALL STRMOV(S2,K2,N,T2,1)                                            000012
00112     25*            IF (I1 - I2)100,200,300                                              000021
00115     26*     100    KOMSTR = -1                                                          000025
00116     27*            RETURN                                                               000027
00117     28*     200    KOMSTR = 0                                                           000033
00120     29*            RETURN                                                               000033
00121     30*     300    KOMSTR = 1                                                           000037
00:       31*            RETURN                                                               000040
001       32*            END # FUNCTION KOMSTR                                                000065
```

```
SUBROUTINE LCMPH     ENTRY POINT 000076

STORAGE USED  CODE(1) 000133; DATA(0) 000012; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000041 100L      0001    000052 300L      0000 D 000000 DBLANK    0000    000003 INJP$     0000 I 000002 LOCS


00101      1*          SUBROUTINE LCMPH(IPHRS,ICOML,ICLMAX,ICLMIN,LOK)                          000002
00103      2*                      IMPLICIT DOUBLE PRECISION (A - Z)                                 000002
00104      3*                      IMPLICIT INTEGER (J - N)                                          000002
00105      4*                      DOUBLE PRECISION DBLANK                                           000002
00107      5*                  INTEGER ICLMAX, ICLMIN                                                000002
00107      6*     C  PURPOSE   LOCATE PHRASE IN STRING OF COMMAND PHRASES                           000002
00107      7*     C  CALL SEQUENCE   IPHRS  - PHRASE TO BE IDENTIFIED                               000002
00107      8*     C                  ICOML  - LIST OF COMMAND PHRASES                               000002
00107      9*     C                  ICLMAX - MAX. NO. OF COMMAND PHRASES TO SEARCH                 000002
00107     10*     C                  ICLMIN - MIN. NO. OF COMMAND PHRASES TO SEARCH                 000002
00107     11*     C                  LOK    - LOCATION OF IPHRS IN ICOML                            000002
00107     12*     C                    (LOK = 0  IF PHRASE NOT FOUND)                               000002
00107     13*     C      NOTE  *LOC* IS A UNIVAC FUNCTION, HENCE *LOK*.                             000002
00107     14*     C  DESIGNED BY   J.D. BURROUGHS                  OCT 1973                         000002
00110     15*          DIMENSION ICOML(ICLMAX)                                                       000002
00111     16*          IF(ICLMIN.LT.1)ICLMIN=1                                                       000002
00113     17*          IF(ICLMAX.LT.ICLMIN)ICLMAX=ICLMIN                                             000010
00113     18*     C ==========  ASSURE THAT SEARCH STARTS BETWEEN ICLMIN AND ICLMAX                  000010
00115     19*          IF(LOK.LT.ICLMIN.OR.LOK.GT.ICLMAX)LOK=ICLMIN                                  000016
00115     20*     C ========  SAVE STARTING POINT OF SEARCH                                          000016
00117     21*          LOCS=LOK                                                                      000036
00120     22*     100  IF(IPHRS.NE.ICOML(LOK)) GO TO 300                                             000041
00122     23*          RETURN                                                                        000046
00123     24*     300  LOK=LOK+1                                                                     000052
00123     25*     C ========  RETURN TO START IF LAST COMMAND PHRASE IS REACHED                      000052
00124     26*          IF(LOK.GT.ICLMAX) LOK=ICLMIN                                                  000054
00124     27*     C ==========  STOP SEARCH WHEN STARTING POINT IS REACHED                           000054
00126     28*          IF(LOK.NE.LOCS) GO TO 100                                                     000062
00130     29*          LOK=0                                                                         000065
00131     30*          RETURN                                                                        000066
00132     31*          END 2 ******************************************                              000132
```

```
FUNCTION NCODE        ENTRY POINT 000131

STORAGE USED  CODE(1) 000142; DATA(0) 000035; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   KOMSTR
 0004   ISCAN
 0005   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000112 100L       0001    000117 200L       0000 D 000001 ALPHA      0000 D 000013 CMP        0000 I 000015 I
 0000    000031 INJP$      0004 I 000000 ISCAN      0000 I 000017 J          0000 I 000020 K          0003 I 000000 KOMSTR
 0000 I 000016 M1          0000 I 000000 NCODE      0000 D 000011 PFN


00101      1*            FUNCTION NCODE( CODE )                                         000000
00103      2*           DOUBLE PRECISION CODE, ALPHA(4),PFN,CMP                         000000
00104      3*           DATA ALPHA/'ABCDEFGHIJ  KLMNOPQRST                              000000
00104      4*          1 'UVWXYZ0123  456789       '/                                   000000
00106      5*           DATA PFN/'PFNAME       '/,CMP/'CMPNTS'                          000000
00106      6*       C                                                                   000000
00106      7*       C           TEST FOR CMPNTS                                         000000
00111      8*           IF(KOMSTR(CMP,1,6,CODE,1).EQ.0) GO TO 100                       000000
00111      9*       C                                                                   000000
00111     10*       C           TEST FOR PFNAME                                         000000
00113     11*           IF(KOMSTR(PFN,1,6,CODE,1).EQ.0) GO TO 200                       000010
00113     12*       C                                                                   000010
00113     13*       C           GET FIRST CHARACTER   1.LE.I.LE.26                      000010
00115     14*           I= ISCAN(CODE,1,1,ALPHA,1,26,M1)                                000021
00115     15*       C                                                                   000021
00115     16*       C           GET SECOND CHARACTER   1.LE.J.LE.36                     000021
00116     17*           J= ISCAN(CODE,2,1,ALPHA,1,36,M1)                                000033
00116     18*       C                                                                   000033
00116     19*       C           DEFAULT IS 'INPT'                                       000033
00117     20*           K=1                                                             000045
00117     21*       C                                                                   000045
00117     22*       C           TEST FOR O  'OUTP'                                      000045
00120     23*           IF(KOMSTR(CODE,3,1,ALPHA,15).EQ.0) K=2                          000047
00120     24*       C                                                                   000047
00120     25*       C           TEST FOR T  'TAPS'                                      000047
00122     26*           IF(KOMSTR(CODE,3,1,ALPHA,20).EQ.0) K=3                          000062
00122     27*       C                                                                   000062
00124     28*           NCODE= K+(I-1)*3 + (J-1)*78                                     000075
00125     29*           RETURN                                                          000106
00126     30*       100 NCODE=2809                                                      000112
00127     31*           RETURN                                                          000113
```

```
00130      32*        200 NCODE=2810                                                                                000117
00131      33*            RETURN                                                                                    000120
00132      34*            END @  UTIL                                                                               000141
```

```
SUBROUTINE NUMERC    ENTRY POINT 000034

STORAGE USED  CODE(1) 000042; DATA(0) 000014; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   ISCAN
 0004   NERR4$
 0005   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000 I 000004 I          0000    000010 INJP$     0003 I 000000 ISCAN     0000 I 000005 M1        0000 D 000003 NUM



00101      1*          SUBROUTINE NUMERC(PHRS, $)                                                         000000
00103      2*                         IMPLICIT DOUBLE PRECISION (A - Z)                                   000000
00104      3*                    IMPLICIT INTEGER (I - N)                                                 000000
00104      4*      C  PURPOSE    TO DETECT WHEN THE LEFT MOST CHARACTER IN A STRING                       00000C
00104      5*      C             IS NUMERIC                                                               000030
00104      6*      C  CALL SEQUENCE   PHRS - STRING OF CHARACTERS                                         000000
00104      7*      C             RETURNS(A) - RETURN TAKEN IF CHARACTER IS NOT NUMERIC                   00000C
00105      8*          DIMENSION NUM(2)                                                                   000000
00106      9*          DOUPLE PRECISION NUM /'1234567890  ', '.-'                                         000000
00106     10*      C --->        COMPARE FIRST CHARACTER TO NUMERICS                                      000000
00110     11*          I=ISCAN(PHRS,1,1,NUM,1,14,M1)                                                      000050
00111     12*          IF(I.LE.0) RETURN 2                                                                000011
00113     13*          RETURN                                                                             000020
00114     14*          END ******************************************                                     000041
```

```
SUBROUTINE NXTPH     ENTRY POINT 000265

STORAGE USED  CODE(1) 000313; DATA(0) 000047; BLANK COMMON(2) 000000

 COMMON BLOCKS

 0003   CIO     000003

EXTERNAL REFERENCES (BLOCK, NAME)

 0004   GETT
 0005   STPMOV
 0006   NWDU$
 0007   NID2$
 0010   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001     000021 132G      0001     000114 155G      0001     000063 20CL      0000     000024 251F      0001     000107 300L
 0001     000157 350L      0001     000166 400L      0001     000173 490L      0001     000177 500L      0001     000202 510L
 0001     000212 600L      0001     000224 700L      0000     000031 601F      0000 D 000006 COMMA      0000 D 000004 DBLANK
 0000 D 000010 EQUALS     0000 I 000020 I          0000 D 000002 IBLNK      0000 I 000017 ICMAXC     0003 I 000002 IDIAG
 0000 I 000022 INBLNK     0003    000040 INJP$     0000 I 000016 IPMAXC     0003    000000 IREAD     0000 I 000021 ISTART
 0000 I 000023 ISTOP      0003 I 000001 IWRITE     0000 D 000000 KAR        0000 D 000012 LFTPAR     0000 D 000014 RGTPAR


00101      1*           SUBROUTINE NXTPH(ICOM,INDEX,IPHRS)                                                 000000
00103      2*                       IMPLICIT DOUBLE PRECISION (A - Z)                                      000000
00104      3*                       IMPLICIT INTEGER (I - N)                                               000000
00105      4*           DOUBLE PRECISION KAR, ICOM, IBLNK, IPHRS                                           000000
00106      5*                       DOUBLE PRECISION DBLANK / '                                            000000
00110      6*           DOUBLE PRECISION COMMA / ',          ' /                                           000000
00112      7*           DOUBLE PRECISION EQUALS / '=          ' /                                          000000
00114      8*           DOUBLE PRECISION LFTPAR / '(          ' /                                          000000
00116      9*           DOUBLE PRECISION RGTPAR / ')          ' /                                          000000
00116     10*       C  PURPOSE   LOCATE NEXT PHRASE IN COMMAND STRING.                                     000000
00116     11*       C  CALL SEQUENCE   ICOM   - COMMAND STRING                                             000000
00116     12*       C                  INDEX  - INDEX TO NEXT CHARACTER TO BE EXAMINED                     000000
00116     13*       C                  IPHRS  - NEXT PHRASE (RETURNED BLANK IF NONE FOUND)                 000000
00116     14*       C  DELIMITERS    3 OR MORE BLANKS, COMMA, EQUALS, LEFT OR RIGHT PARENTHES              000000
00120     15*           COMMON/CIO/IREAD,IWRITE,IDIAG                                                      000000
00121     16*           DIMENSION ICOM(1)                                                                  000000
00122     17*           DATA IBLNK/'                                                                       000000
00124     18*           IPMAXC=16                                                                          000000
00125     19*           ICMAXC=80                                                                          000001
00126     20*           IPHRS=IBLNK                                                                        000003
00126     21*       C ---        RETURN IF AT COLUMN 80                                                    000003
00127     22*           IF(INDEX.GT.ICMAXC)RETURN                                                          000005
```

```
00127       C ---       LOCATE FIRST NON-BLANK, NON-DELIMITER CHARACTER            000005
00131       150   DO 200 I=INDEX,ICMAXC                                            000014
00134             CALL GETT(ICOM,I,KAR)                                            000021
00135             IF (KAR .EQ. COMMA .OR. KAR .EQ. EQUALS .OR.                     000026
00135            1    KAR .EQ. LFTPAR .OR. KAR .EQ. RGTPAR) GO TO 200              000026
00137             IF(KAR.NE.IBLNK) GO TO 300                                       000057
00141       200   CONTINUE                                                         000064
00143             INDEX=ICMAXC                                                     000064
00144             IF(IDIAG.GE.100.)WRITE(IWRITE,251)INDEX,IPHRS                    000066
00151       251   FORMAT(14HNXTPHR2 INDEX=,I3,1X,A10)                              000103
00151       C ---       RETURN WHEN REST OF STRING IS EMPTY                        000103
00152             RETURN                                                           000103
00152       C ---       LOCATE NEXT DELIMITER  (END OF PHRASE)                     000103
00153       300   ISTART=I                                                         000107
00154             DO 400 I=ISTART,ICMAXC                                           000110
00157             CALL GETT(ICOM,I,KAR)                                            000114
00160             IF (KAR .EQ. COMMA .OR. KAR .EQ. EQUALS .OR.                     000121
00160            1    KAR .EQ. LFTPAR .OR. KAR .EQ. RGTPAR) GO TO 490              000121
00162             IF(KAR.EQ.IBLNK) GO TO 350                                       000151
00164             INBLNK=0                                                         000154
00165             GO TO 400                                                        000155
00166       350   IF(INBLNK.GE.2) GO TO 500                                        000157
00170             INBLNK=INBLNK+1                                                  000162
00171       400   CONTINUE                                                         000167
00173             INDEX=ICMAXC                                                     000167
00174             GO TO 600                                                        000171
00175       490   ISTOP=I-1                                                        000173
00176             GO TO 510                                                        000175
00177       500   ISTOP=I-3                                                        000177
00200       510   INDEX=I                                                          000202
00200       C ---       TEST TO LIMIT PHRASE TO <= 10 CHARACTERS                   000202
00201             IF((ISTOP-ISTART+1).LE.IPMAXC) GO TO 700                         000203
00203       600   ISTOP=ISTART+IPMAXC-1                                            000212
00203       C ---       TEST TO PREVENT PHRASE FROM GOING BEYOUND COL. 80          000212
00204             IF(ISTOP.GT.ICMAXC) ISTOP=ICMAXC                                 000215
00206       700   INBLNK=ISTOP-ISTART+1                                            000224
00206       C ---       LOAD PHRASE                                                000224
00207             CALL STRMOV(ICOM,ISTART,INBLNK,IPHRS,1)                          000227
00210             IF(IDIAG.GE.100.)WRITE(IWRITE,801)INDEX,IPHRS                    000234
00215       801   FORMAT(13HNXTPHR INDEX=,I3,1X,A10)                               000253
00216             RETURN                                                           000253
00217             END ****************************************                     000312
```

```
SUBROUTINE PUTCOD    ENTRY POINT 000106


STORAGE USED  CODE(1) 000111; DATA(0) 000030; BLANK COMMON(2) 000000


EXTERNAL REFERENCES (BLOCK, NAME)

 0003   DAND
 0004   DOR
 0005   DCMPL
 0006   SHIFT
 0007   NERR3$


STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000 D 000002 B7777     0003 D 000000 DAND      0000 D 000000 DBLANK    0005 D 000000 DCMPL     0004 D 000000 DOR
 0000 D 000010 E1        0000 D 000012 E2        0000 I 000014 IA        0000 D 000014 ICOD      0000    000017 INJP$
 0000 I 000007 ISHIFT    0000 I 000006 IWORD     0000 D 000004 MASK      0006 D 000000 SHIFT



00100      1*      CPUTCOD                                                                         000002
00101      2*           SUBROUTINE PUTCOD(N,IARRAY,ICODE)                                          000002
00103      3*                      IMPLICIT DOUBLE PRECISION (A - Z)                               000002
00104      4*                      IMPLICIT INTEGER (I - N)                                        000002
00105      5*                      DOUBLE PRECISION DBLANK                                         000002
00107      6*                DOUBLE PRECISION DAND, DOR, DCMPL                                     000002
00107      7*      C  PURPOSE    PLACE A 4 DIGIT CODE.  VALUE OF CODE MUST BE BETWEEN              000002
00107      8*      C             2047*, STORED 5 CODES/WORD, FROM AN ARRAY OF PARAMETER            000002
00107      9*      C             CODES.  THIS ROUTINE IS USED TO REDUCE THE STORAGE REQUIRED       000002
00107     10*      C             TO STORE THE I/O CODE LISTS FOR EACH ANALYSIS MODULE.             000002
00107     11*      C  CALL SEQUENCE    N   LOCATION OF CODE IN ARRAY IARRAY (5 CODES/WORD).        000002
00107     12*      C                   IARRAY   INTEGER ARRAY WHICH RECIEVES CODE NUMBER.          000002
00107     13*      C                   ICOUT    VALUE OF CODE INPUT TO ROUTINE.                    000002
00110     14*           DOUBLE PRECISION SHIFT                                                     000002
00111     15*           DOUBLE PRECISION IARRAY(1)                                                 000002
00112     16*           DOUBLE PRECISION B7777 / 000000000000000077770000                          000002
00114     17*           DOUBLE PRECISION ICOD,MASK                                                 000002
00115     18*           INTEGER IA(2)                                                              000002
00116     19*           EQUIVALENCE (IA(1), ICOD)                                                  000002
00117     20*           IA(1) = 0                                                                  000002
00120     21*           FLD(12,12,IA(2)) = FLD(24,12,ICODE)                                        000003
00121     22*           MASK = B7777                                                               000013
00121     23*      C  DETERMINE WHICH WORD IN ARRAY IS TO BE MODIFIED.                             000013
00122     24*           IWORD=(N-1)/5+1                                                            000015
00122     25*      C  DETERMINE NO. OF BITS TO SHIFT CODE TO LEFT.                                 000015
00123     26*           ISHIFT = (4 - MOD(N-1, 5)) * 12                                            000024
00123     27*      C  SHIFT CODE + MASK TO PROPER BIT LOCATION IN WORD.                            000024
00124     28*           ICOD=SHIFT(ICOD,ISHIFT)                                                    000034
00125     29*           MASK=SHIFT(B7777,ISHIFT)                                                   000042
```

```
00125      30*      C  PLACE CODE BITS INTO CORRECT LOCATION IN WORD OF IARRAY.                    000042
00126      31*            E1 = DAND(IARRAY(IWORD), DCMPL(MASK))                                      000047
00127      32*            E2 = DAND(ICOD, MASK)                                                      000063
00130      33*            IARRAY(IWORD) = DOR(E1, E2)                                                000070
00131      34*            RETURN                                                                     000075
00132      35*            END $ ****************************************                             000110
```

```
SUBROUTINE PUTT      ENTRY POINT 000067

STORAGE USED  CODE(1) 000073; DATA(0) 000015; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NLPP3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000040 100L       0001   000053 200L       0000   000005 INJP$     0000 I 000001 IPOS      0000 I 000000 NWORD
 0000 D 000002 S1         0000 R 000002 S2


00101      1*             SUBROUTINE PUTT(S, I, T)                                                        000002
00101      2*      C                                                                                      000002
00101      3*      C      PUTT(S, I, T) EXTRACTS THE LEFTMOST CHARACTER                                   000002
00101      4*      C      FROM THE DOUBLE PRECISION WORD T, AND INSERTS IT INTO                           000002
00101      5*      C      THE I'TH POSITION OF DOUBLE PRECISION STRING S,                                 000002
00101      6*      C      BEGINNING WITH S(1).                                                            000002
00103      7*             DOUBLE PRECISION S(1), T, S1                                                    000002
00104      8*             DIMENSION S2(2)                                                                 000002
00105      9*             EQUIVALENCE(S1,S2)                                                              000002
00105     10*      C                                                                                      000002
00105     11*      C      DETERMINE WORD CONTAINING I'TH CHARACTER                                        000002
00106     12*             NWORD=(I-1)/10 + 1                                                              000002
00107     13*             S1 = S(NWORD)                                                                   000011
00107     14*      C                                                                                      000011
00107     15*      C      DETERMINE RELATIVE POSITION OF CHARACTER IN WORD                                000014
00110     16*             IPOS=MOD(I-1,10) * 6                                                            000015
00111     17*             IF(IPOS.GT.35)GO TO 100                                                         000022
00111     18*      C                                                                                      000022
00111     19*      C      EXTRACT LEFTMOST CHARACTER FROM D/P WORD T AND                                  000022
00111     20*      C      INSERT IT INTO STRING S                                                         000022
00113     21*             FLD(IPOS,6,S2(1))=FLD(0,6,T)                                                    000025
00114     22*             GO TO 200                                                                       000036
00115     23*      100    IPOS = IPOS - 36                                                                000040
00116     24*             FLD(IPOS,6,S2(2)) = FLD(0,6,T)                                                  000043
00117     25*      200    S(NWORD) = S1                                                                   000053
00120     26*             RETURN                                                                          000057
00121     27*             END @ SUBROUTINE PUTT                                                           000072
```

```
SUBROUTINE READMS    ENTRY POINT 000070

STORAGE USED  CODE(1) 000107; DATA(0) 000511; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

 0003   NCODE
 0004   NRDA$
 0005   NIO1$
 0006   NIO2$
 0007   NWDU$
 0010   NERR3$

STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000    000460 1001F      0001    000031 115G      0001    000041 122G      0001    000020 90L       0001    000047 999L
 0000 D 000471 DCODE      0000 I 000457 I         0000 I 000471 ICODE      0000    000476 INJP$     0000 I 000456 LPAGE
 0003 I 000000 NCODE      0000 D 000000 PAGE


00100       1*      CREADMS                                                                                000002
00101       2*            SUBROUTINE READMS(NUNIT,ARRAY,NWORDS,CODE)                                       000002
00101       3*      C  CALL SEQUENCE    NUNIT   =  TAPE NO.                                                000002
00101       4*      C                   ARRAY =    ARRAY TO BE LOADED                                      000002
00101       5*      C                   NWORDS  =  NO. OF WORDS IN ARRAY                                   000002
00101       6*      C                   CODE    =  CODE (SEARCH KEY)                                       000002
00103       7*            DOUBLE PRECISION ARRAY(1),CODE,PAGE(151),DCODE                                   000002
00104       8*            EQUIVALENCE (DCODE,ICODE)                                                        000002
00104       9*      C     SINCE FORMAL ARGUMENT MAY NOT BE USED IN EQUIVALENCE                             000002
00104      10*      C     STATEMENT, ASSIGN THE VALUE TO A LOCAL VARIABLE.                                 000002
00105      11*            DCODE = CODE                                                                     000002
00106      12*            LPAGE=64                                                                         000004
00107      13*            IF(NUNIT.EQ.7) GO TO 90                                                          000006
00111      14*            LPAGE= 151                                                                       000011
00112      15*            ICODE= NCODE(CODE)                                                               000013
00112      16*      C           TRANSFER DATA FROM BUFFER                                                  000013
00112      17*      C                                                                                      000013
00113      18*         90 READ(NUNIT'ICODE,ERR=999) (PAGE(I),I=1,LPAGE)                                    000020
00121      19*            DO 100 I=1,NWORDS                                                                000041
00124      20*        100 ARRAY(I) = PAGE(I)                                                               000041
00126      21*            RETURN                                                                           000043
00127      22*      999   WRITE(6,1001)NUNIT                                                               000047
00132      23*      1001  FORMAT(/' ERROR OCCURRED DURING READMS ON UNIT ',I3)                             000054
00133      24*            RETURN                                                                           000054
00134      25*            END $  SUBROUTINE READMS  ***********************                                000106
```

TABLE B.1

UNIFORMITY OF MAGNESIUM DOPING IN LANTHANUM CHROMITE FILMS FORMED BY EVD

Sample	Section	Approx. Mg^{2+}/La^{3+} mole ratio
1	1 (center)	0.025
	2 Downstream end	0.02
2	1 Upstream end	0.01
	2	0.002
	3	0.002
	4	0.0025
	5	0.0025
	6	0.0025
	7	0.0025
	8	0.003

numbered consecutively, starting with the side where the reacting gases enter the EVD reactor. Values given are for mole per cent of the chromium (or, possibly, lanthanum) replaced in the perovskite structure.

It is seen that the magnesium is relatively evenly distributed, except for sample 2, section 1, which is near the upstream entrance of the gases to the EVD reactor. Incomplete mixing of the reacting metal chloride vapors may be the problem at this point.

It is anticipated that a suitable composition for the interconnection will be $LaCr_{.75-x}Al_{.25}Mg_xO_3$ where x will lie optimally somewhere between 0.002 and 0.02, values within the capability of the present EVD process as shown by the two interconnection layer preparations of Table B.1. Although the larger value of x (0.02) provides the highest electrical conductivity, the lower value minimizes unwanted oxygen transport through the interconnection region. The adaptation of the EVD process to conveniently add magnesium in the proper concentration solves a major problem encountered previously with this process.

10.2.1 Photoelectron Spectroscopy of Lanthanum Chromites

Small deviations from stoichiometry and valence state changes of the chromium ion, due to addition or other ions, are important in determining the degree and type of conduction processes in lanthanum chromite. X-ray photoelectron spectroscopy (XPS or ESCA) can give information on these properties and on subtle crystal field and conduction band phenomena in oxide conductors.

The detailed nature of the conduction processes and composition variations with time of operation in the lanthanum chromite interconnection may significantly affect the operating characteristics and useful operating life of the fuel cell stack. For example, vacancies at normal metal ion sites of the perovskite crystal structure of lanthanum chromite could enhance unwanted metal ion migration through the interconnection and cause gradual formation of interface layers. This could result from imperfections in the initial preparation, from vaporization losses during operation, and from gradual impurity doping from other parts of the cell or the gaseous surroundings over prolonged periods of fuel cell operation. A combination of ESCA with sensitive chemical analysis and x-ray diffraction can provide information on such small but important defects in the interconnection material.

We are using ESCA to examine a series of lanthanum chromites with small differences in the lanthanum-to-chromium ratio and with different degrees of doping in either the A or B crystallographic positions of the ABO_3 perovskite structure. For example, comparing $LaCrO_3$ with two similarly prepared non-stoichiometric samples, $La_{.97}CrO_3$ and $LaCr_{.97}O_3$ shows splittings of the binding energy values derived for the 3 p levels of chromium. This suggests possible vacancies at normal lanthanum and chromium ion sites, causing a shift to a higher effective valence for part of the chromium ions. Second phase lanthanum and chromium-containing compounds are also a possible explanation. Work of Anderson* on the temperature of phase transitions in lanthanum chromites is consistent with the vacancy explanation, although the vacancy-containing lanthanum chromite compositions might perhaps be metastable.

Substitution of Mg^{2+} or Sr^{2+} for Cr^{3+} or La^{3+} on the B and A sites of the ABO_3 perovskite structure also produces a splitting of the 3 p binding energy level. This might be attributed to the additional presence of a "Cr^{4+}" ion, causing a one electron volt shift of part of the p electrons to a higher energy level than occurs with only Cr^{3+} present.

Since the ESCA technique is sensitive to small changes in stoichiometry and doping, it should be helpful in detecting such small differences in the interconnection which are present, initially, or after prolonged operation of the fuel cell stack.

*Reported at Workshop in High-Temperature Solid Oxide Fuel Cells, held at Brookhaven National Laboratory, May 5 and 6, 1977.

10.3 Appendix C - Analytical Techniques for Examination of Thin Oxide Layers

Chemical and micrographic analysis of the thin interconnection layers reouires sperial techniques that can examine both the chemical and physical uniformity of thin layers and microsections and give information on changes in valence states or chemical binding of the constituent ions, as altered by the alkaline earth or aluminum additions to lanthanum chromite.

Techniques we have used most extensively are (1) SEM (Scanning Electron Microscope) with EDAX (Energy Dispersive Analyzer for X-rays) attachment, (2) DIMA (Direct Imaging Mass Spectrometer), (3) ESCA (X-ray photoelectron spectroscopy), and (4) Monochromatic X-ray diffraction.

10.3.1 Scanning Electron Microscopy (SEM)

In this instrument, electrons from a hot filament are accelerated down the center of an evacuated column, focused by electron lenses, and used to scan, in form of a fine beam, across the specimen. This produces a line image of the scanning coil current on a cathode ray tube (CRT) surface in a manner similar to that of a television screen. Secondary electrons produced at the specimen strike a collector. The resulting current is amplified and used to modulate the brightness of the CRT. Since the collection times for the secondary electrons are small, compared to the scanning rate, there is a nearly one-to-one correspondence between the quantity of secondary electrons emitted at any point of the specimen surface and the brightness of the CRT. Thus, an image of the surface is built up progressively on the CRT screen.

The magnification is determined by the ratio of sizes of the scanning raster on the CRT screen compared to the specimen surface. The method is relatively non-destructive has high resolution for surface topography and an excellent depth of focus. The latter is very useful in examining fracture edges which often are not planar surfaces. A particularly useful addition is the EDAX attachment, which allows a two-to-three micrometer diameter spot on the surface to be located by examination of the CRT image of the surface. The X-rays emitted from this spot down to a depth of one-to-two micrometers are collected by the energy dispersive X-ray analyzer using a Li-Si detector. Qualitative concentration analysis can be attained for elements above sodium on the periodic chart. The small electron beam size facilitates the investigation of possible inhomogeneities or multi-layer structures which might be present on the fracture edge of a thin interconnection layer. Figure C.1 (bottom), shows a photograph of this combined equipment. A new Tracer-Northern quantitative EDAX unit was recently purchased which will also be available in the near future.

10.3.2 Direct Imaging Mass Analyzer (DIMA)

This instrument generates analytical data by directing a primary ion beam of oxygen, nitrogen, or argon onto the surface of a sample to be analyzed. The primary beam sputters the sample in a layer-by-layer fashion, ionizing a small fraction of the sputtered atoms in the process. Both positive and negative ions are produced and either may be selected for analysis. Secondary ions are accelerated into a magnetic field, where they are separated according to their mass-to-charge ratio. The outstanding feature of the instrument is that it is perfectly stigmatic, i.e., capable of maintaining a one-to-one correspondence between a point of origin on the sample and its final location in the detection system. This instrument can produce direct ion images of high spacial resolution (1μ) of any selected ion from the sample. In the detection system the ion image is converted to an equivalent electron image which may be observed visually from a fluorescent screen and recorded photographically or electronically.

The CAMECA SMI-300 instrument is shown in Figure C.2. The DIMA suffers no fundamental limitation in detecting all elements in the periodic chart, and because a mass spectrometer is used as the identifying instrument, isotopic analysis is possible. There is no overlap of signal peaks from different elements, such as occurs sometimes with the EDAX technique.

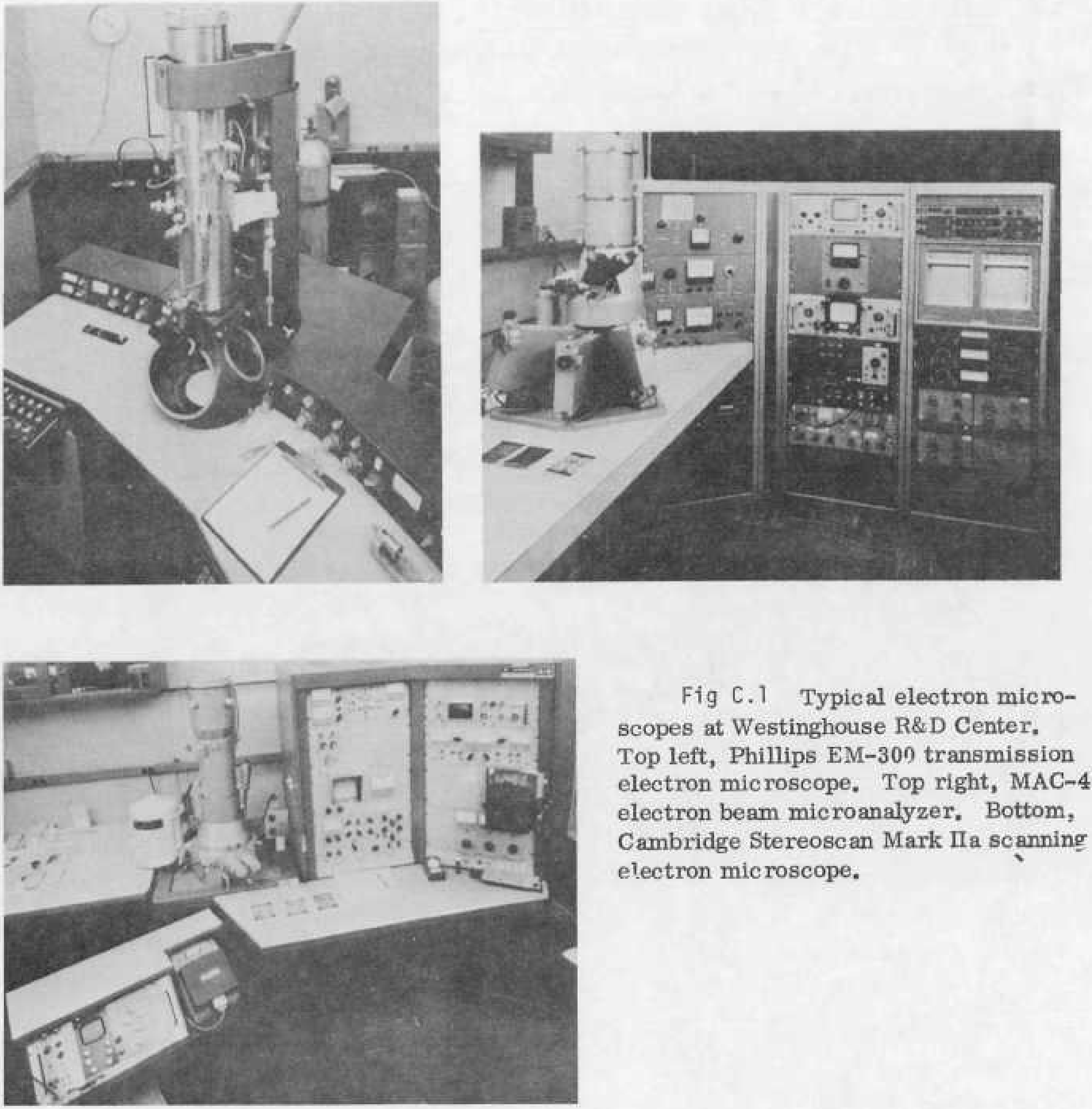


Fig C.1 Typical electron microscopes at Westinghouse R&D Center. Top left, Phillips EM-300 transmission electron microscope. Top right, MAC-400 electron beam microanalyzer. Bottom, Cambridge Stereoscan Mark IIa scanning electron microscope.

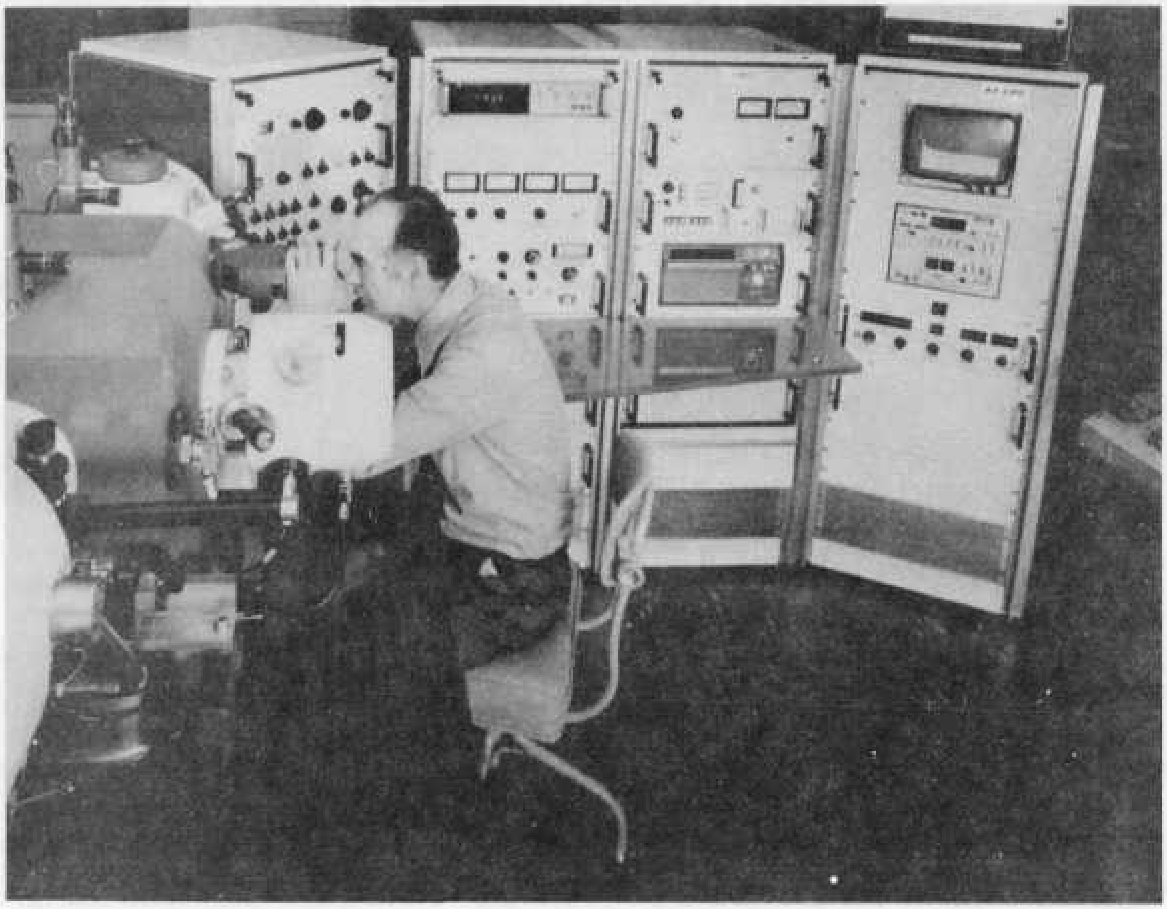


Fig C.2 CAMECA direct imaging mass analyzer.

10.3.3 X-Ray Photoelectron Spectroscopy (ESCA or XPS)

X-ray photoelectron spectroscopy is a technique for high resolution measurement of electron energies that are ejected from a sample irradiated with X-rays. Such electron energies are characteristic of the atom and its chemical environment. Thus, energy associated with the various ESCA peaks provides both elemental identification, as well as molecular structure and valence state data (chemical shifts). Compositional data from the 5 to 20 Å region next to the surface are typical of this method. Figure C.3 shows the ESCA equipment.

10.3.4 Monochromatic X-Ray Powder Diffraction

This is a much older technique than the others but is very useful for obtaining crystal structure data on small quantities of a second phase. The use of monochromatic radiation reduces background signal interference allowing minor phases to be detected more readily. Qualitatively, it is used for identification of a single crystalline structure or detection of additional crystal forms in the sample.

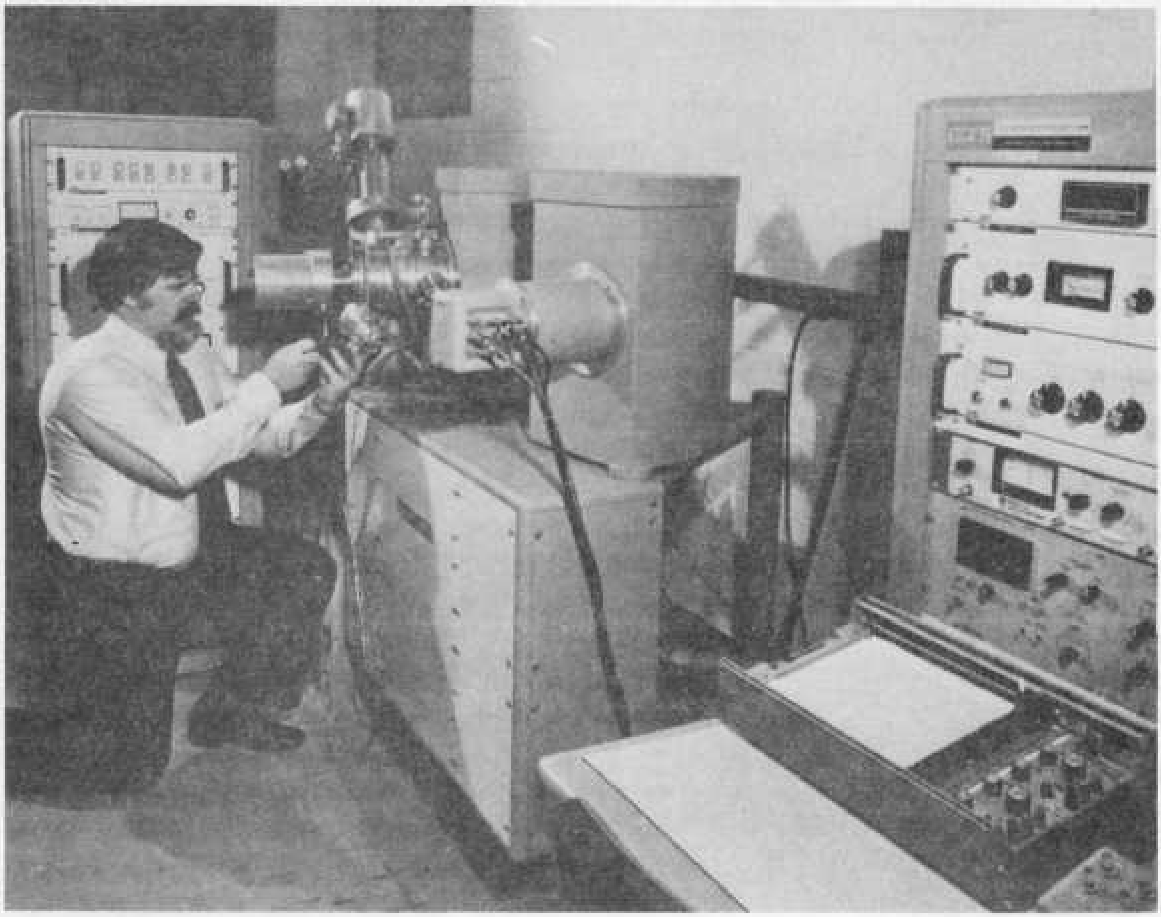


Figure C.3. GCA/McPherson Photoelectron Spectrometer, ESCA 36

10.4 Appendix D - Electrical Measurement Apparatus For Component Testing

10.4.1 *Apparatus Requirements:*

- temperature up to 1100°C
- flowing gas atmosphere (N_2-O_2, N_2-H_2-H_2O, etc.),
- up to eight electrical leads for specimen current, potentials, and temperature,
- current source, 0 to 10A dc, constant current or constant voltage operation,
- rapid voltage measurement with 1 microvolt resolution,
- separate switching for current inputs and voltage outputs to provide flexibility for a variety of combination specimens.

Specifically, the apparatus employs a Hoskins tube furnace with a 2 inch O.D. by 13 inch long high purity alumina tube (closed at one end) for the specimen chamber. A removable flange at the opposite end of the tube contains four feed-through seals for 0.190 inch diameter alumina tubes. Two of these tubes provide an inlet and outlet for the flowing gas; the other two carry eight platinum leads into the furnace hot zone where the specimen is supported by its various electrode leads. Two alumina disks, midway along the tubes, act as radiation baffles for the hot zone.

The specimen geometry is very similar to that of the actual fuel cell. In most cases it consists of a short length of 1/2 inch diameter porous ceramic tubing on which one or more fuel cell component films have been deposited, plus auxiliary surface electrodes for current input and voltage measurements.

DC current for the combination specimen testing is provided by a Kepco JMK6-10M power supply. This is a precision unit which can deliver either stabilized output voltage or current, over the range of 0-6V, 0-10A; moreover, the unit is externally controllable so that it can be used as a potentiostat. Operation in the constant current mode is especially valuable for short-term resistance stability tests of components such as the interconnection. Potentiostatic operation may be useful for future electrode polarization measurements.

For voltage measurement, a Keithley 160 Digital Multimeter will be used. This is a 3-1/2 digit unit with ranges as low as 1 millivolt (resolution of 1 microvolt). It is essential to have a voltage resolution of this magnitude for the following reason: typical thin film fuel cell components have specific resistances of 40 milliohms/cm^2 or less at 1000°C.

This means that some of the potentials to be measured will be less than 40 millivolts, assuming a maximum test current density of 1000 mA/cm^2. Recently developed interconnection material may be between one-to-two orders of magnitude lower in resistance than previous materials; consequently, potential measurement capability in the sub-millivolt range is required.

The final major item in the apparatus is a dual switch unit, designed for flexibility in making measurements on combination specimens. Important features of this unit are (1) independent selection of up to six current inputs and six voltage outputs on one combination specimen, (2) flexibility in changing inputs and outputs to different types of combination specimens, and (3) low thermal noise L + N switches.

10.5 Appendix E - Crossed-Electrode (CE) Method For Measuring Thin Film Specimen Resistance

Due to the inherently-low resistance of thin film specimens, it is essential to: (a) enhance the specimen resistance by employing a very small specimen area, and (b) eliminate or minimize the resistance of electrode leads and their contacts with the electrodes. The Crossed-Electrode (CE) method for measuring thin film specimens, discussed in a previous report,* meets these requirements well. It is shown schematically in Figures E.1 and E.2. The active thin film specimen area is only where the electrodes cross, which satisfies requirement (a). Requirement (b) is satisfied by employing a four-terminal resistance measurement, with the extensions of the current electrodes serving as potential electrodes. The ultimate limitation in the CE method begins to appear when the specimen resistance is low enough to be comparable with the lateral resistance of the electrodes in direct contact with the specimen. At this point, a lateral potential drop occurs across the electrode which causes the apparent resistance of the specimen (as measured by the four-terminal method) to appear lower than the true specimen resistance. An analysis of this effect has been carried out which enables one to correct for it, or better still, to choose an electrode width and thickness for which the effect will be small. It was found** that

$$R_a = \frac{\sqrt{R_e R_t}}{\sinh\sqrt{R_e/R_t}}$$

or approximately

$$R_a = R_t - \frac{1}{6} R_e \text{ for } R_t > R_e$$

*Fifth Quarter Report, Thin Film Fuel Battery/Fuel Cell Power Generating System, DOE Contract EY-76-C-03-1197, July 18, 1977.

**A distributed parameter treatment of the problem was used which was similar to that discussed in the first annual report.

Dwg. 6437A26

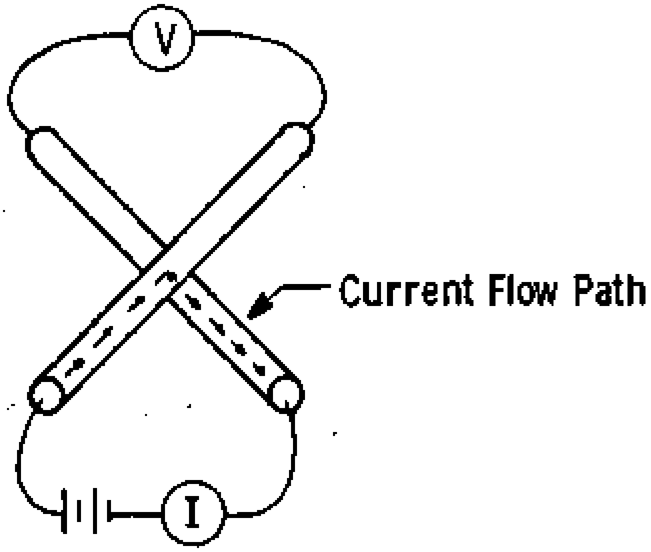


Fig. E.1 — Schematic of crossed-rod method for measuring electrical contact resistance

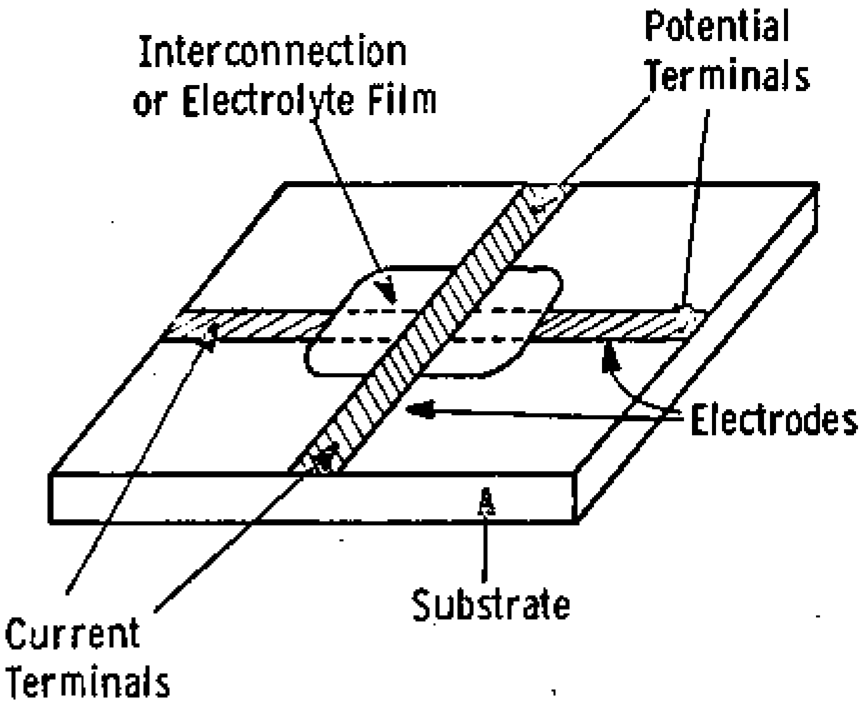


Fig. E.2 — Schematic of crossed-electrode geometry for combination testing

where

R_a = apparent specimen resistance

R_t = true specimen resistance

R_e = lateral (sheet) resistance of both electrodes

= 2 $\rho e/\delta e$

with

ρ_e = resistivity of electrode material

δ_e = thickness of electrode

A convenient form of the results for the purpose of correcting experimental data appears in the graph of Figure E.3.

Examination of the approximate equation above shows that there are two ways in which the relative size of the effect can be reduced: (1) reduce the lateral electrode resistance by using thicker electrodes and (2) increase the specimen resistance by decreasing its area; i.e., by decreasing the electrode width.

Curve 693468-A

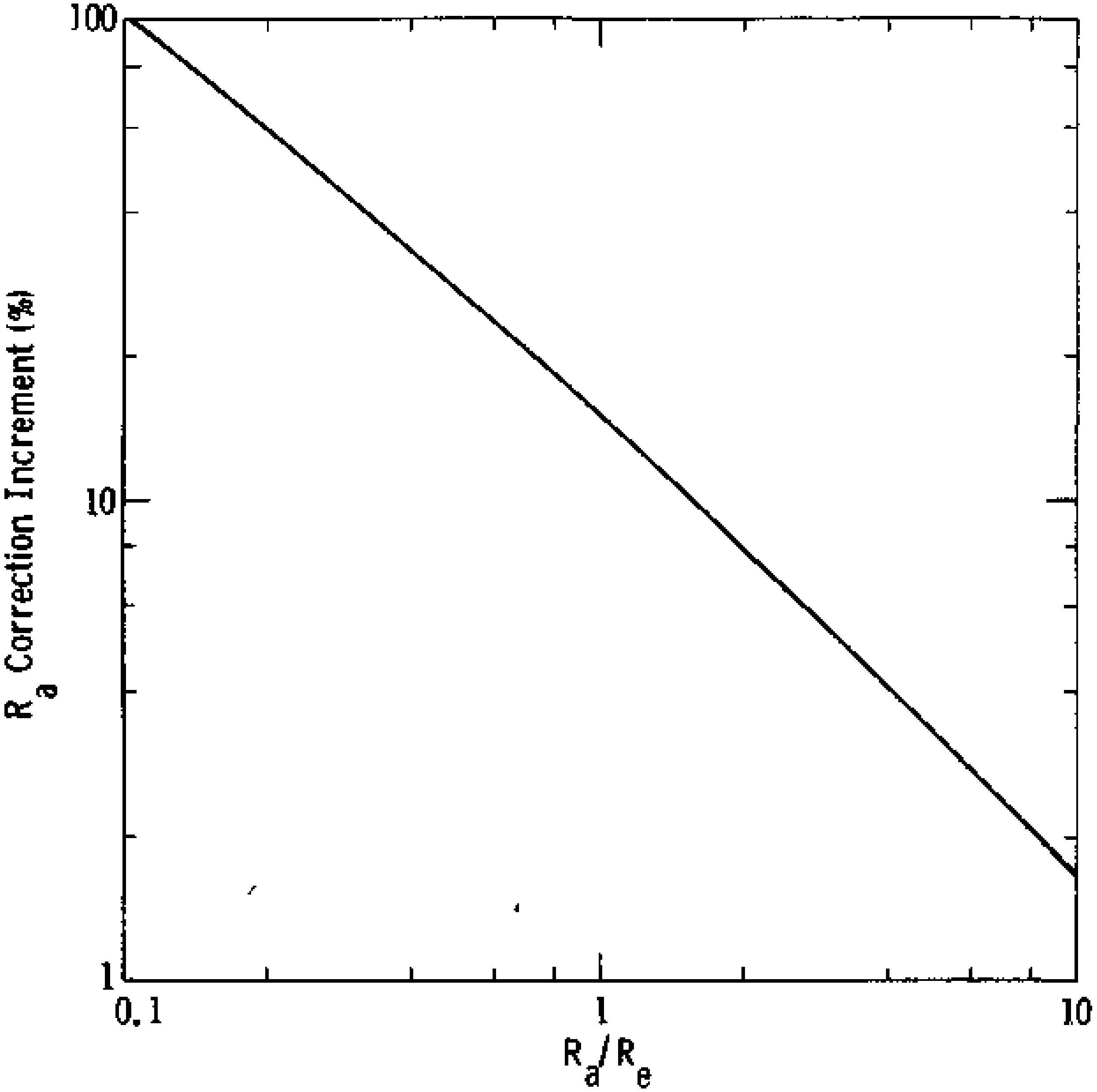


Fig. E.3 - Correction for the apparent specimen resistance, R_a, when it becomes comparable to the lateral electrode resistance, R_e, in a crossed-electrode measurement.

10.6 Appendix F - Resistance Measurements on an Interconnection-Film-Fuel Electrode Combination Specimen

Two types of measurement will be considered here: a two-contact method and a three-contact method. Figure F.1 shows an enlarged longitudinal cross-section through the contact region of the specimen, while Figure F.2 shows the corresponding distributed resistance analog.

10.6.1 Two-Contact Method

In this method leads #1 and 3 carry current, and leads #2 and 4 measure specimen potential drop (lead #5 is not used). It is seen that the resistance being measured is the sum of two interconnection film regions (one directly under each contact) plus the fuel electrode resistance in the gap between them. From transmission line theory*, the resistance R_1, corresponding to the region under one contact is

$$R_1 = \frac{\sqrt{r_f r_i}}{\tanh\left(\sqrt{\frac{r_f}{r_i}}\,\ell\right)} \cong \frac{r_i}{\ell} + \frac{1}{3} r_f \ell \quad \text{for} \quad \sqrt{\frac{r_f}{r_i}}\,\ell < 1$$

where

$r_i = \frac{\rho_i \delta_i}{C}$ = interconnection film resistance per unit length of tube

$r_f = \frac{\rho_f}{C\delta_f}$ = fuel electrode film resistance per unit length of tube

ρ_i, ρ_f = resistivities of interconnection and fuel electrode

δ_i, δ_f = thicknesses of interconnection and fuel electrode

ℓ = measurement contact width

*The resistance R_1 corresponds to the input impedance of an open-circuit transmission line of length ℓ.

Dwg. 6406A86

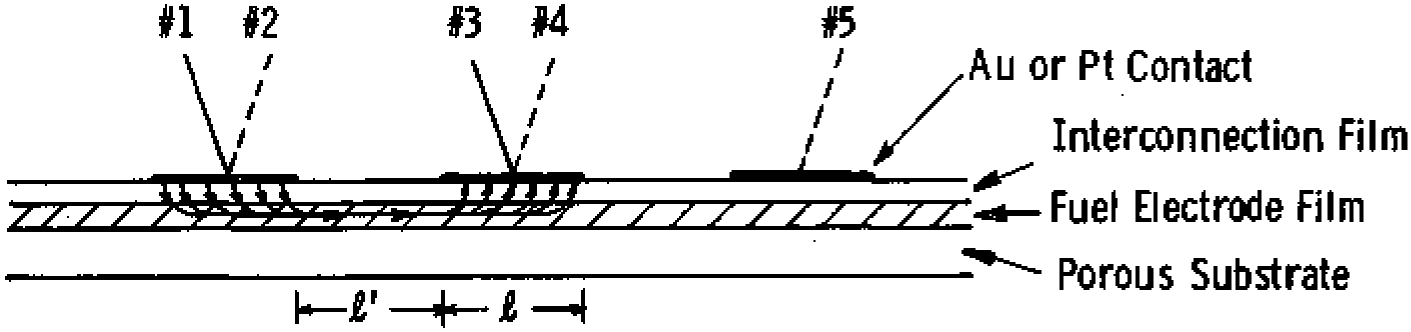


Fig. F.1 — Enlarged cross-section showing various specimen layers and measurement contacts. (Potential lead ---, current lead —, current flow lines →→)

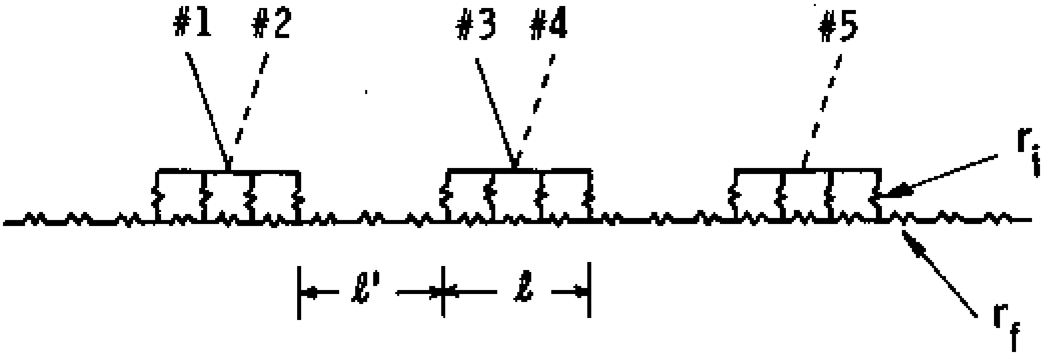


Fig. F.2 — Distributed resistance analog of the specimen films and contacts

ℓ' = measurement contact spacing

C - tube circumference

The total resistance R_2 measured externally through both contacts is then

$$R_2 = 2R_1 + r_f{}^{\ell'} = 2\,\frac{r_i}{\ell} + r_f\left(\frac{2}{3}\,\ell + \ell'\right)$$

Important conclusions regarding the two-contact method are (1) for good measurements of the interconnection film resistance, both the contact width ℓ and contact separation ℓ' must be kept small and (2) interconnection film polarity-sensitive effects cannot be measured because one effectively has two specimens in series, back to back.

10.6.2 Three-Contact Method

For this method, leads #1 and 3 carry current and leads #4 and 5 measure specimen potential drop (lead #2 is not used). It can be shown that the resistance being measured here corresponds to the interconnection film region directly under the middle contact. From transmission line theory*, the apparent resistance R_3, corresponding to the region under the middle contact, is for this case

$$R_3 = \frac{\sqrt{r_f r_i}}{\sinh \sqrt{\frac{r_f}{r_i}}\,\ell} \cong \frac{r_i}{\ell} - \frac{1}{6}\,r_f\,\ell \quad \text{for} \quad \sqrt{\frac{r_f}{r_i}}\,\ell \leqq 1$$

Important conclusions regarding the three-contact method are (1) for good measurements of the interconnection film resistance, only the contact width ℓ need be kept small, (2) interconnection film polarity-sensitive effects can be measured (one effectively has a single specimen), and (3) the error in measurement due to fuel-electrode resistance is much less (~ a factor of 5) than for the two-contact method.

*The resistance R_3 corresponds to the transfer impedance between the two ends of an open-circuit transmission line of length ℓ.

Numerical Example Comparing the Two Methods

Assume the following specimen parameters:

$\rho_i = 2\ \Omega\text{cm}$ (comparable to newest interconnection materials)

$\rho_f = 2 \times 10^{-4}\ \Omega\text{cm}$

$\delta_i = \delta_f = 20 \times 10^{-4}$ cm

$\ell = \ell' = 0.2$ cm

$C = 4$ cm

Then $r_i \equiv \dfrac{\rho_i \delta_i}{C} = 0.001\ \Omega\text{cm}$

$$r_f \equiv \frac{\rho_f}{C\delta_f} \approx 0.025\ \Omega/\text{cm}$$

Since $\sqrt{\dfrac{r_f}{r_i}}\,\ell = 1$ we may use the approximate formulas.

For the two-contact method:

$$R_2 = 2\frac{r_i}{\ell} + r_f\left(\frac{2}{3}\ell + \ell'\right)$$

$$= 2 \times \frac{0.001\ \Omega\text{cm}}{0.2\ \text{cm}} + 0.025\ \Omega/\text{cm}\ \left(\frac{5}{3} \times 0.2\ \text{cm}\right)$$

$$R_2 = 10.0\ \text{m}\Omega + 8.3\ \text{m}\Omega = 18.3\ \text{m}\Omega$$

The first term is interconnection material resistance, the second term is fuel electrode resistance. Hence, the error in measuring the interconnection material is +83% for this case.

For the three-contact method:

$$R_3 = \frac{r_i}{\ell} - \frac{1}{6} r_f \ell$$

$$= \frac{0.001\ \Omega\text{cm}}{0.2\ \text{cm}} - \frac{1}{6} \times 0.025\ \Omega/\text{cm} \times 0.2\ \text{cm}$$

$$R_3 = 5.00\ \text{m}\Omega - 0.83\ \text{m}\Omega = 4.17\ \text{m}\Omega$$

The error in measurement of the interconnection material is -15% for this case.

11.0 Permanent Record Book Entries

Figuring books, containing information described in this report are:

206811	pp.	111-115
207720	pp.	75-114
207349	pp.	147-156
207906	pp.	1-41, 57-58, 60-67, 68-76
207773	pp.	57-69
207599	pp.	96-100
208043	pp.	1-110
207728	pp.	65-89, 91-132, 134-156
207905	pp.	6-22, 26-38, 40, 41, 44, 46-61, 64-68, 71-81, 83-130
208023	pp.	1-14, 25-34
207840	pp.	53-63