RADC-TR-72-2, Volume II
Final Technical Report
January 1972

RASTER PLOTTER SOFTWARE
Volume II - Software Documentation

United Aircraft Corporation

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) United Aircraft Corporation Hamilton Standard Division Windsor Locks, Connecticut 06096	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A

3 REPORT TITLE

RASTER PLOTTER SOFTWARE
Volume II - Software Documentation

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report

5 AUTHOR(S) (First name, middle initial, last name)

Rodney T. Boorse
David L. Sharp
Gerald L. Sperber

6 REPORT DATE	7a. TOTAL NO OF PAGES	7b NO OF REFS
January 1972	338	0

8a CONTRACT OR GRANT NO	9a. ORIGINATOR'S REPORT NUMBER(S)
F30602-71-C-0107 Job Order No. 60350000	HSER 5924 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-72-2 Volume II (of two)

10 DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY
None	Rome Air Development Center (IRAG) Griffiss Air Force Base, New York 13440

13 ABSTRACT

A color coding scheme for hierarchically representing the cartographic symbology associated with JOG series 1501-A charts has been developed and implemented at the RADC Experimental Cartographic Facility (ECF). The scheme employs a three level hierarchy, with up to ten colors available for each level of the hierarchy. The first level of color coding is accomplished on a registered, transparent overlay to the base manuscript. The annotative procedure is straightforward and permits rapid color tagging with very low error raster.

Based on this color coding scheme, computer programs which accomplish the necessary recognition and interpretation of these color codes were developed and demonstrated. Color tagged raster data recorded by the RADC Automatic Color Scanning Device comprised the input to these programs, while the output comprised lineal data fully annotated in accordance with the prescribed code structure. In this process, separate scannings are required for the base manuscript and the coded overlay.

In addition, computer programs which symbolize pre-formatted raster data for output on the RADC Graphic Plotter were developed and demonstrated. These programs accept edited and symbolized raster line center data, which has been sorted by the assigned final printed graphic color, and generate the aperture and density level commands required to drive the Graphic Plotter.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

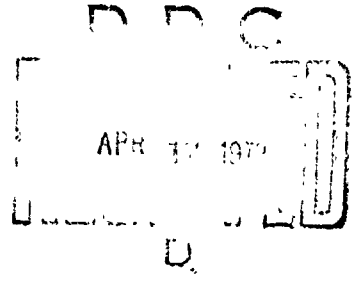
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Automated Cartography Computer Programming Faster Data Processing Color Coding						

RASTER PLOTTER SOFTWARE
Volume II - Software Documentation

Rodney T. Boorse
David L. Sharp
Gerald L. Sperber

United Aircraft Corporation



FOREWORD

This report describes the work performed by the Hamilton Standard Division of United Aircraft Corporation, Windsor Locks, Connecticut, in accordance with the terms and requirements of contract F30602-71-C-0107, Job Order Number 60350000, for Rome Air Development Center, Griffiss Air Force Base, New York. William G. McLellan (IRAG) was the RADC Project Engineer administering the program. Secondary report number is HSER 5924.

In addition to the authors, Hamilton Standard personnel contributing to program implementation and the preparation of this report were Messrs. J.R. Byrd, R.G. Hubbard, H.R. Lee, and R.I. Skolnick.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved.

Approved: WILLIAM G. MCLELLAN
Project Engineer

Approved: FRANZ H. DETTMER
Colonel, USAF
Chief, Intel & Recon Division

FOR THE COMMANDER:

FRED I. DIAMOND
Acting Chief, Plans Office

APPENDIX I

FEATURE IDENTIFICATION SOFTWARE SYSTEM OPERATING INSTRUCTIONS

APPENDIX I

TABLE OF CONTENTS

APPENDIX I

INDEX OF DIAGRAMS

c. Select DECtape-to-Disk Program Loading Options

Place the following PDP-9 DISK CONTROL UNIT panel switches in the "OFF" (down) or "ON" (up) positions at the completion of the DSKSAV loading operation. The operator will be signalled at the completion of this operation by the following system generated administrative message.

"DSKSAV"

Disk control panel settings:

- Control switches 0-29 - "OFF"
- WRITE PROTECT switch - "OFF"

d. Transfer PDP-9 Operating System Object Programs from DECtape-to-Memory-to-Disk File Storage.

(1) System maintenance program, DSKSAV loads object coded executive system programs from DECtape to active core memory. The resulting core image program is then transferred onto disk. To initiate the execution of this function, perform the following sequence of console operations:

- Set console ACCUMULATOR switches to 000000_8
- Depress CONTINUE

(2) Upon the successful completion of system program transfer from DECtape 0 to disk storage, the following administrative message is typed on-line via the console keyboard/page printer:

"DSKSAV"

e. Reset Disk Control Panel Options to Initiate the Loading and Building of an FISS Disk Resident Program File

(1) Place the following PDP-9 Disk Control Unit panel switches in the "OFF" (down) or "ON" (up) positions:

- Control switches 0-29 - "ON"
- WRITE PROTECT switch - "ON"

f. Transfer FISS Object Program from DECtape-to-Memory-to-Disk File Storage

(1) Initiate transfer of FISS Object Program from DECtape 2 to disk file storage as follows:

- Set console ACCUMULATOR switches to 000002_8
- Depress CONTINUE

(2) The following message will be typed on-line when the program load and transfer operation previously described has been completed:

"DSKSAV"

3. DISK-TO-CORE PDP-9 MONITOR LOADING INSTRUCTIONS

a. Load the Disk File Resident MONITOR System Into Active Core Memory

(1) Place system utility program, DISK, in console paper tape reader

(2) Set console ADDRESS switches to 77637_8

(3) Depress the following console function keys in the order specified below:

- STOP
- I/O RESET
- READIN

(4) Upon the successful completion of the above program loading function, the following system generated administrative message is typed on-line:

"MONITOR V4E"

(5) The above message signals the termination of the operating and applications program system disk loading sequence, and indicates that the core resident MONITOR program is available for use.

SECTION II

FEATURE IDENTIFICATION SOFTWARE SYSTEM PROGRAM

1. GENERAL

The following paragraphs describe the sequence of instructions to load the FISS program into core memory from the PDP-9 disk program catalog, and to initialize the program once control is relinquished by the resident MONITOR. Both the program loading and initialization processes are carried out on-line by way of the PDP-9 console teleprinter and keyboard device. To accomplish the loading process the user interacts with the resident PDP-9 MONITOR system as detailed in Section II.2 of this appendix. Program initialization is accomplished by interacting with the connection control program, CNTRL, and is described in Section II.3.

The following I/O message notation convention is employed in this appendix: MONITOR system and FISS program generated output messages are underscored, as shown in the example below. Keyed input messages, entered by the user, are not underscored.

Example:

System Output Message: MONITOR V4E

Operator Input Message: LOAD

2. FISS PROGRAM LOADING INSTRUCTIONS

a. MONITOR V4E

System generated message to signal the user that the MONITOR system is core memory resident and is available for use.

b. G 2

Enter the above input message to initiate the loading of the Feature Identification Software System Program from Disk Area 2 to primary memory.

c. ↑S

Enter this message to start execution of program.

Preceding page blank

3. PROGRAM INITIALIZATION AND CONTROL INSTRUCTIONS

This section provides a detailed description of the procedural steps needed to initialize the Feature Identification Software System. The program initialization process is implemented by means of a query-response control mechanism. The on-line teletype serves as the primary communications link between man and machine, and is employed to generate administrative messages instructing the user of the specific data and control parameters needed for normal execution; to generate diagnostic messages signaling the type of keying or procedural errors committed by the user during the initialization process; and to serve as the input medium for user response messages.

All user input messages must be terminated by a carriage return, unless otherwise indicated.

a. MOUNT ACSD MANUSCRIPT TAPE ON DRIVE 3. ↑ P

b. ↑ P

Enter CONTROL P upon completion of above task.

c. MOUNT ACSD OVERLAY TAPE ON DRIVE 5. ↑ P

d. ↑ P

Enter CONTROL P upon completion of above task.

e. MOUNT SCRATCH TAPE ON DRIVE 6. ↑ P

f. ↑ P

Enter CONTROL P upon completion of above task.

g. ENTER RESOLUTION AS 2,4, 5 or 6.

h. N ↑ P

Enter ACSD resolution increment value N (decimal integer = 2, 4, 5 or 6) at which input data was generated.

NOTE: Upward or downward scaling of the output data file will occur if a resolution increment other than the original ACSD machine setting is entered.

i. ENTER MIN AREA SETTING AS XX ELEMENTS.

j. Enter any odd 2-digit number between Ø1 and 31 to indicate original ACSD MIN AREA machine setting, e. g. Ø3 ↑ P.

k. ENTER LINEAL NEIGHBORHOOD RANGE AS XX UNITS.

l. Enter a two-digit number greater than zero (e. g. Ø5 ↑ P) to indicate the maximum number of resolution elements to be used to relate lineal data points along the scan axis. Parameter is not requested if a MIN AREA of 1 was indicated.

m. ENTER AREAL NEIGHBORHOOD RANGE AS XX UNITS.

n. Enter a two-digit number greater than zero (e. g. Ø2 ↑ P) to indicate the maximum number of resolution elements to be used to relate areal data points along the scan axis.

o. ENTER MINIMAL FEATURE LENGTH AS XX RASTER UNITS.

p. Enter a two-digit number (XX) between Ø1 and 31 to indicate the minimum number of raster data points in a feature data set for that data set to be retained, e. g. 15 ↑ P.

q. ENTER GAP PARAMETER AS X RASTER UNITS.

r. Enter a one-digit octal number greater than or equal to zero (e. g. 1 ↑ P) to indicate the maximum number of raster scan lines to span to establish neighborhood connection.

s. ENTER BASE COLOR CODE AS XX.

t. Enter a two-digit decimal number between ØØ and 15 to indicate the color code of the desirable base manuscript feature. The color codes are the ACSD machine settings used for that specific base manuscript and its associated overlay.

4. ERROR CONDITIONS AND PROGRAM STOPS

If an abnormal condition occurs during the execution of the program, one of the following messages will be printed on-line via the console teleprinter.

a. STOP ØØØØØ1

ACSD Tape Error - Magnetic tape error encountered during ACSD base manuscript tape generation.

b. STOP ØØØØØ2

PDP9GE Overflow - Data point encountered that exceeds the limits of CDP system. Rerun program. If error persists, dump disk and examine data set point values for possible error.

c. STOP ØØØØØ3

ACSD Tape Error - Magnetic tape error encountered during ACSD overlay tape generation.

d. STOP ØØØØØ4

Line Buffer Overflow-ACSD scan line record encountered on base manuscript tape that is larger than the fixed sized raster record input buffer.

e. STOP ØØØØØ5

Core Memory Overflow - More than 128 feature data sets simultaneously active in memory. Reload program and increase the range parameters in input messages (k) and (m).

f. STOP ØØØØØ6

Disk Overflow - Available disk storage exhausted. Reload connection program increase or decrease neighborhood range parameters (see input messages k and m).

g. STOP ØØØØØ7

Line Buffer Overflow - ACSD scan line record encountered on overlay tape that is larger than the fixed sized raster record input buffer.

5. PROCESSING STATISTICAL DATA OUTPUTS

Immediately following the successful completion of the line connection process, timing and data statistics are printer on-line via the console teleprinter. These include the number of points input and output, convergences and divergences of features, linear/areal transitions encountered, feature data sets output, and so on. Figure I-1 contains a typical example of the statistical data generated.

```
MONITOR V4E

$G 2
↑S
MOUNT ACSD MANUSCRIPT TAPE ON DRIVE 3.↑P
↑P
MOUNT ACSD OVERLAY TAPE ON DRIVE 5.↑P
↑P
MOUNT SCRATCH TAPE ON DRIVE 6.↑P
↑P
ENTER RESOLUTION AS 2,4,5,OR 6. >

4↑P
ENTER MIN AREA SETTING AS XX ELEMENTS. >

Ø1↑P
ENTER AREAL NEIGHBORHOOD RANGE AS XX UNITS. >

Ø1↑P
ENTER MINIMAL FEATURE LENGTH AS XX RASTER UNITS. >

1Ø↑P
ENTER GAP PARAMETER AS X RASTER UNITS. >

1↑P
ENTER BASE COLOR CODE AS XX. >

Ø9↑P


                RASTER-TO-LOCUS CONVERSION STATISTICS


NUMBER OF POINTS INPUT =      2835.
NUMBER OF POINTS OUTPUT =      2787.
NUMBER OF CONVERGES =        30
NUMBER OF DIVERGES =        31
NUMBER OF LINEAR/AREAL TRANSITIONS =        17
NUMBER OF FEATURE DATA SETS =       137
NUMBER OF FEATURES UNDER MINIMAL LENGTH =        51
NUMBER OF DISK SEGMENTS USED =       288
APPROXIMATE LENGTH OF PRODUCT =        11.148
RUN TIME =         1 MIN.    12 SEC.


MONITOR V4E

$
```

FIGURE I-1 . FISS PROGRAM RUN

APPENDIX II

FEATURE IDENTIFICATION SOFTWARE SYSTEM PROGRAM DESCRIPTION

APPENDIX II

TABLE OF CONTENTS

SECTION II (CONT)

SECTION III
FISS FLOW DIAGRAMS

INDEX OF DIAGRAMS

SECTION I

1. GENERAL

The system software elements described in this appendix consists of the new subprograms required and the CAST Engineering Change "A" programs modified to implement the Feature Identification Software System (FISS). The four new programs are detailed in Section II. Of the programs resulting from the CAST Engineering Change "A" effort, FILBUF required extensive changes and is detailed in Section II.1, while four other programs required only minor modifications which are described in Section II.2. All other CAST programs remain as originally documented. Section II.3 identifies the PDP-9 Object-Time System programs and Science Library macro subroutines required for various Fortran IV I/O and mathematical functions. Section III contains the flow diagrams of the FISS programs described in Section II.1 and II.2.

2. SYSTEM DESCRIPTION

The FISS Program is a self-contained, stand-alone program designed to be executed on a PDP-9 processing system having the following configuration:

Quantity	Equipment Item Description
1	DEC PDP-9 Control Processing Unit (CPU)
8	4096 18-bit Word Core Memory Modules
1	10 CPS, Console Keyboard/Teleprinter
3	7-Track, 45IPS, 556/800 BPI IBM Compatible Magnetic Tape Units W/Controller
4	DECtape Units W/Controller
1	DEC PDP-9 Paper Tape Reader
1	DEC PDP-9 Paper Tape Punch
1	1 Million 18-Bit Word Disk File W/Controller
1	ANELEX, 1000 LPM Line Printer

a. Object Program Deck Arrangement

The FISS Program is assembled and compiled as a "relocatable" program. With the exception of the control program (CNTRL), which by definition must be the first program in the object deck, the remaining object programs may be arranged in any order.

b. System Diagrams

Figure II-1 shows the control logic flow for the FISS program. PDP-9 OTS

and Science Library subprograms and routines are not shown since they are loaded and enter the procedureal flow only at the time of execution. As shown, the program as designed is divided into three functional phases of execution: Initialization and Phase I and II.

(1) Initialization

Package consists of the worker programs shown in addition to PDP-9 OTS input/output handlers and drivers loaded at time of execution. These provide a software controlled user-system interface as is required either to enter needed control information and data, or to output system generated, administrative messages via the console keyboard/teleprinter devices.

(2) Phase I

Software is devoted to the enactment of the point-to-point correlation functions as performed by the software modules indicated. Such functions are performed in a serial batch mode of operation. Overlay color information is correlated to base manuscript positional data to establish symbolic color-coded feature header information. Lineal formatted output data records are retained in disk file storage until a base color data connection pass is completed. Such data is then made available to the Phase II processor for final output formatting.

(3) Phase II

Consists of those translatory and record data formatting functions as required to transform internally formatted data records to the required standard ACS Manuscript (MMS) Tape File Format. This processor, like the previous Phase I processor, is designed to enact these functions in a serial batch processing mode.

Figure II-2 shows the overall functional block diagram of the FISS program.

c. Logic Diagram Symbology

The syllabus of logic diagram symbology, as shown in Figure II-3, defines the program flow diagram notation conventions employed in this appendix.

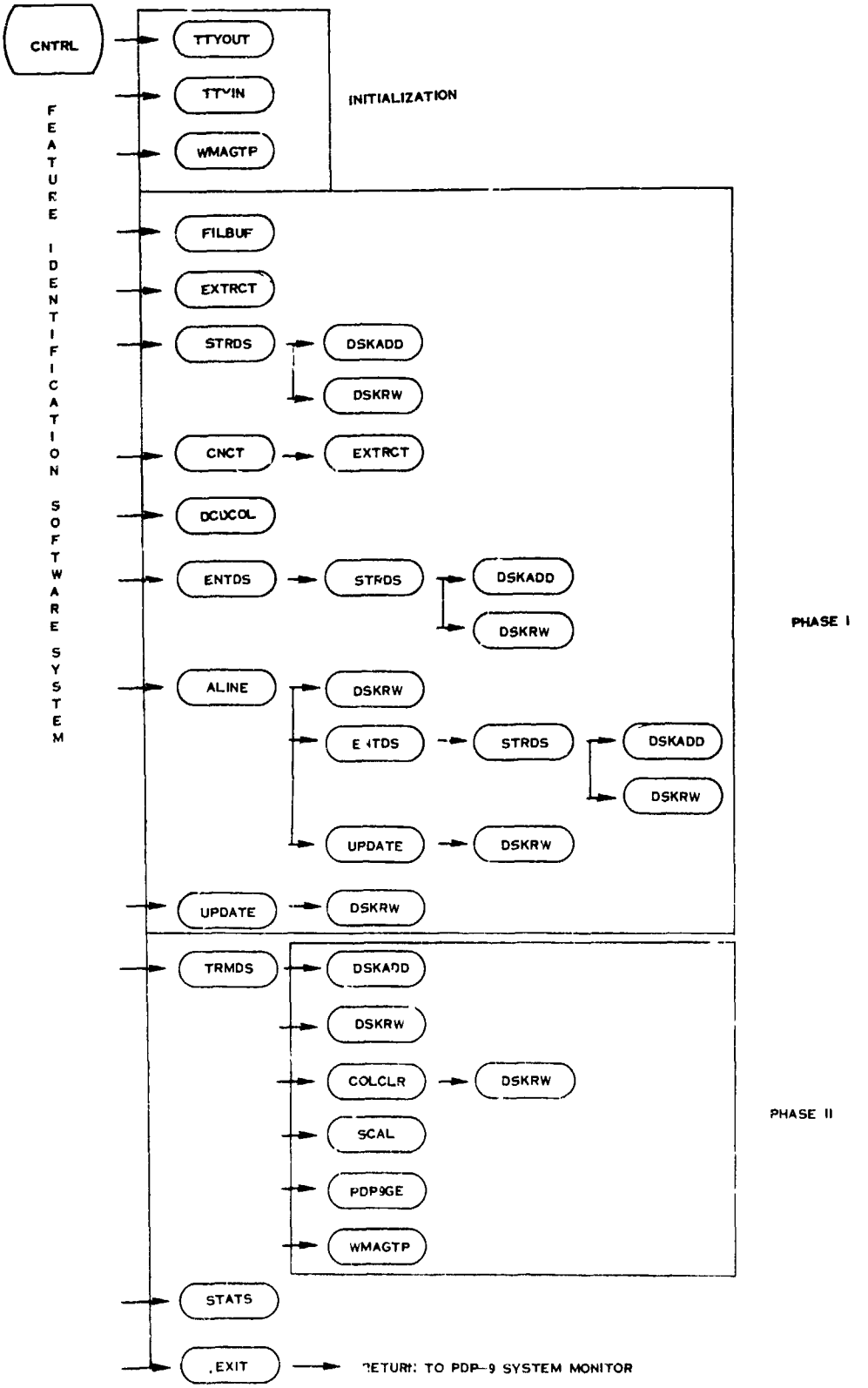


FIGURE II-1. SYSTEM LOGIC FLOW DIAGRAM

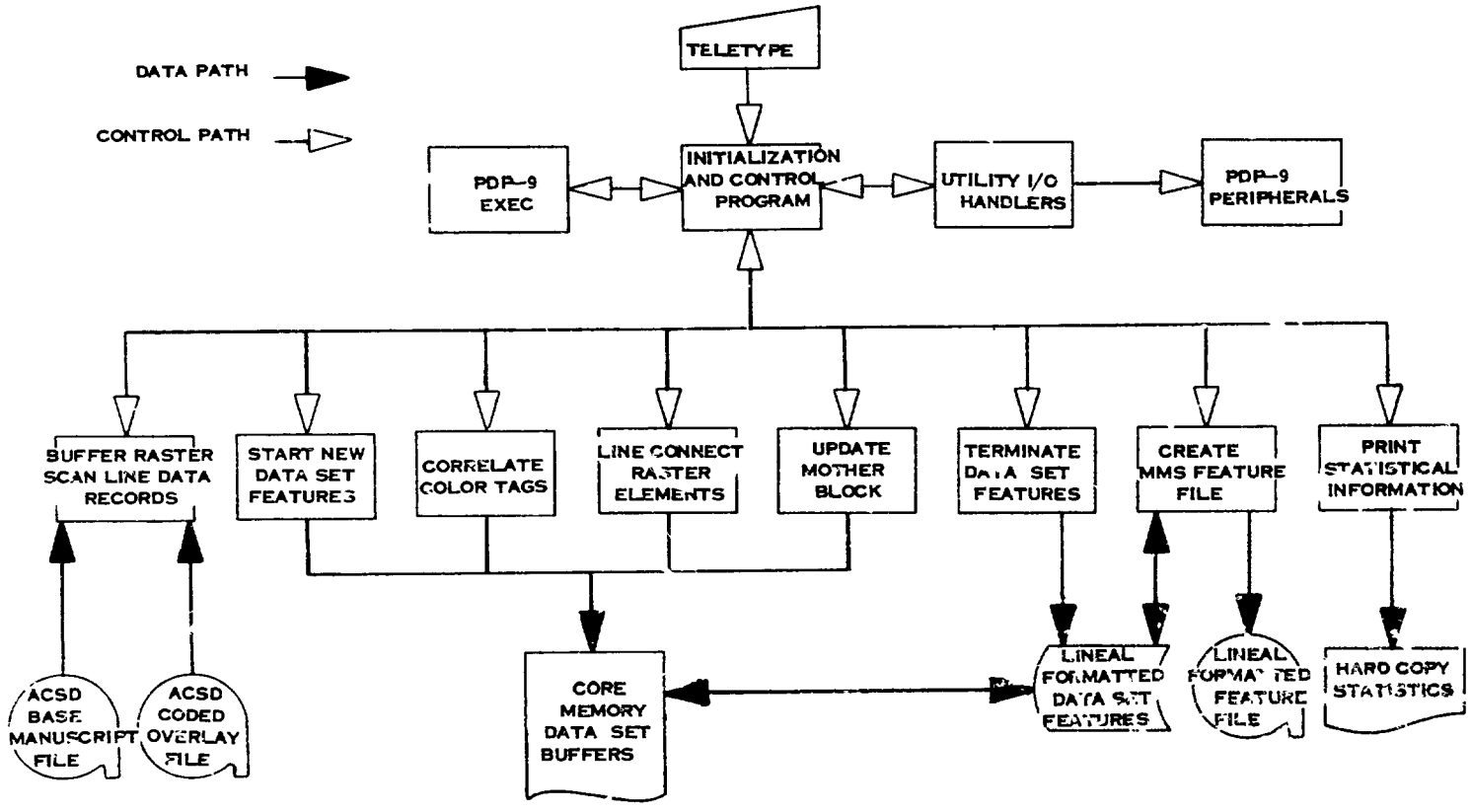


FIGURE II - 2. FEATURE IDENTIFICATION SOFTWARE SYSTEM BLOCK DIAGRAM

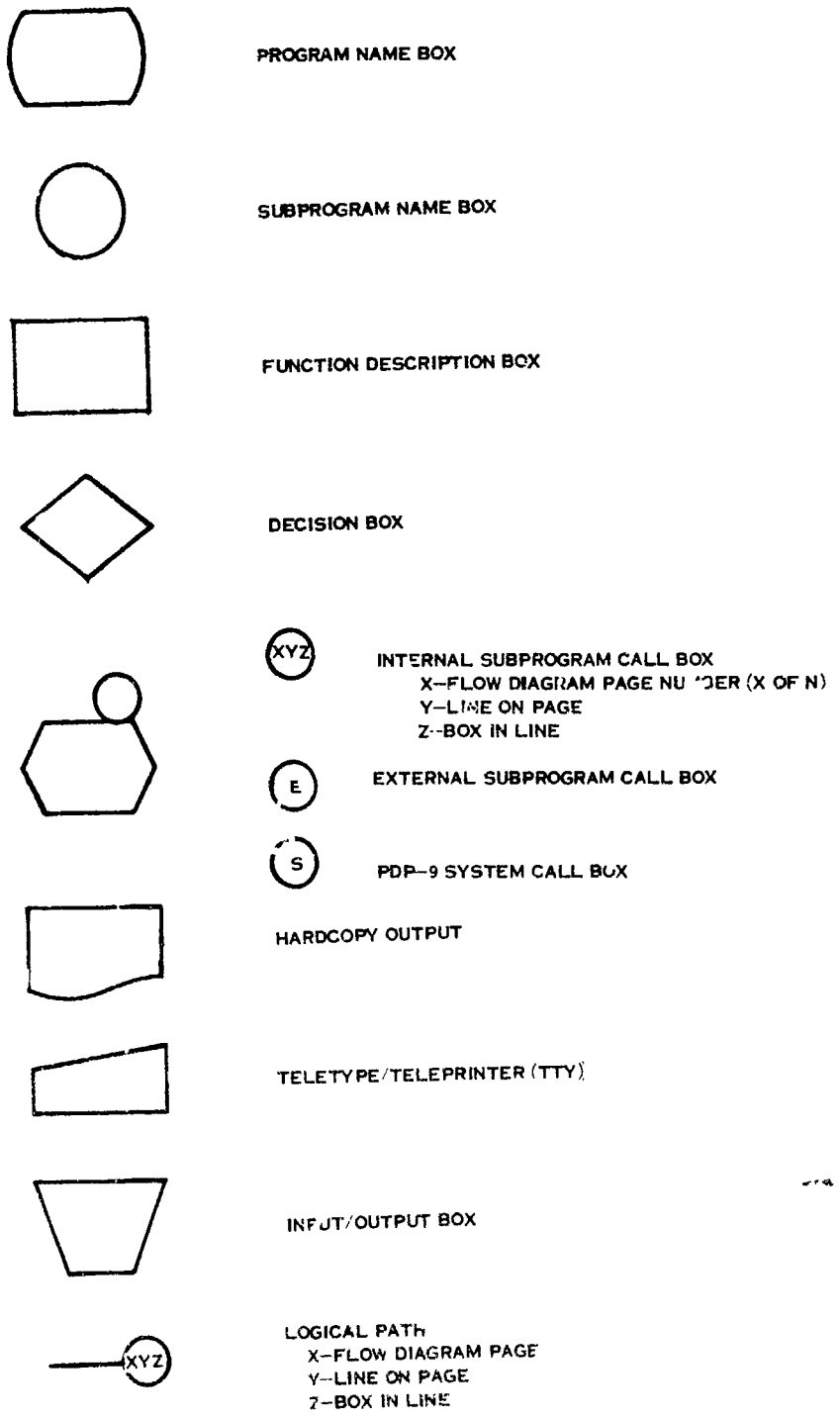


FIGURE II-3. LOGIC DIAGRAM SYMBOLOGY

SECTION II

DESCRIPTION OF SUBPROGRAMS

1. NEW FISS PROGRAMS

a. UPDATE (Update Mother Data Set)

(1) Abstract

UPDATE updates the mother data set block identifying a feature with data set linkage and color identification information.

(2) Program Description

Upon unconditional data set termination UPDATE retrieves the specified mother data set block from disk, if not already core-resident, updates correlation pointer no. 2 with data set linkage information, overwrites a data set header no longer used with the core-resident color identification word describing the data set, and rewrites the mother data set block to disk. In the case of core resident mother data set blocks, the initial disk read is bypassed. The update information is used by COLCLR for feature header formulation.

(3) Inputs

Calling Sequence -

```
L          LAC      DSCAD
L + 1      JMS*     UPDATE
L + 2      - -            Next sequential instruction
```

where:

DSCAD = core address of data set block.

(4) Outputs

Disk -

Updated mother data set block.

(5) Referenced Subprograms

DSKRW

b. DCDCOL (Decode Color)

(1) Abstract

DCDCOL updates the specified core-resident data set block color word with color identification information.

(2) Program Description

DCDCOL uses the contents of the ACC (color code) to compute a table address of the corresponding bit map representation for that color code. The data set requiring a color identification update is identified via the data set address passed in the calling sequence. For each call to DCDCOL the current color word for the specified data set is logically ORed with the bit map representation of the color code identified in the calling sequence.

(3) Inputs

Calling Sequence -

```
L                    LAC          COLOR
L + 1                JMS*         DCDCOL
L + 2                SKP                 Return location
L + 3      ARG       Ø
L + 4                                    Next sequential
                                         instruction
```

where:

ARG = core address of data set block.

(4) Outputs

Memory -

Updated data set color word.

(5) Referenced Subprograms

None

c. COLCLR (Collect Color)

(1) Abstract

COLCLR is the Raster Plotter Color Code Implementation routine that collects all the color codes from all the data sets associated with a feature in

order to formulate the proper header for that feature.

(2) Program Description

COLCLR is called by TRMDS whenever the correlation between the data sets of a feature and the overlay color codes has been completed. COLCLR chains through all the data sets associated with a given feature, extracting the required color code from each data set as it goes along. The data sets are read from disk using DSKRW. Required information from the data block headers is extracted by the internal routine HEADER. Another internal routine, GETHDR, takes the extracted color code information, separates out the proper color code and formulates the appropriate word for the feature header identification.

(3) Inputs

Calling Sequence -

```
L                JMS*        COLCLR
L + 1            JMP         .+3
L + 2            .DSA        ARG1
L + 3      ARG2  Ø
L + 4            .........                 Next sequential in-
                                           struction
```

where:

ARG1 = mother block of an end of the feature for which the color information is to be collected.

ARG2 = the header word returned to the calling program.

(4) Outputs

Memory -

Color code word that uniquely identifies the type of feature interested in.

(5) Referenced Subprograms

DSKRW

d. FILBUF (Fill Line Buffer)

(1) Abstract

FILBUF buffers raw ACSD records from magnetic tape to an internal buffer and transfers logical scan line records and its associated color information from the tape buffer to a specified scan line memory buffer and its associated color identif-

ication buffer. The raster scan value (y - coordinate) is returned to the calling program.

(2) Program Description

FILBUF is called whenever a line buffer is to be filled. Each call fills the specified line buffer with coordinate data of the next consecutive scan line. The coordinate data is followed by a trailer word indicating the end of coordinate data. Associated with each data element in the line buffer is an ACSD color code in the associated color identification buffer identifying the color code representing that particular data element.

FILBUF recognizes three types of raster elements appearing in the raw ACSD records. Line center and areal data are directly transferred from the tape to the specified line buffer. Min Area data is conditionally transferred to the specified line buffer. For a Min Areas of one, Min Area data is interpreted as noise and is not passed to the line buffer as valid data. For Min Area settings other than one, the Min Area data points are converted to a begin area point, and an end area point is generated using the begin area plus the Min Area setting minus one. If the line buffer capacity (1024 words) is reached, line buffer overflow has occurred, otherwise the y-coordinate associated with the x-coordinate data is returned to the calling program. An end-of-data condition is indicated to the calling program by a return to the location following the call arguments address.

If line buffer overflow occurs, a terminal condition is reached, and the program is aborted causing the following message to be output on the teleprinter: "STOP ØØØØØ4"(STOP 000007). If a tape error was encountered during ACSD tape generation, an error condition is recognized and the program aborted causing the following message to be output on the teleprinter: "STOP ØØØØØ1"(STOP ØØØØØ3).

(3) Inputs

(a) Calling Sequence -

L	JMS*	FILBUF	
L + 1	JMP	.+3	Return location
L + 2		ARG	
L + 3	JMP	EOD	End-of-data return location
L + 4	- -		Next sequential instruction

where:

ARG = address of buffer arguments

where buffer arguments:

ARG1 = address of line buffer address
ARG2 = y-coordinate of line in buffer

(b) Magnetic Tape -

Raw ACSD multi-color file data records (1024 24-bit words).

(4) Outputs

(a) Memory -

Scan line buffers of raster data elements and associated color identification buffers.

(b) Teleprinter -

The following message is output when line buffer overlow (base manuscript tape) occurs: "STOP" ØØØØØ4".

The following message is output when line buffer overflow (overlay tape) occurs 'STOP ØØØØØ7".

The following message is output when ACSD tape error (base manuscript tape) is encountered: "STOP ØØØØØ1".

The following message is output when an ACSD tape error (overlay tape) is encountered: "STOP ØØØØØ3".

(5) Referenced Subprograms

None

2. MODIFIED CAST PROGRAMS

a. CNTRL (Feature Identification Software System Control Program)

(1) Abstract

CNTRL is the supervisory program element of the Feature Identification Software System. Following system initialization the program buffers raster scan records from raw ACSD tape files by calling FILBUF, locates and extracts data elements and associated color code information from line buffers by calling EXTRCT,

and correlates the data elements to active feature data sets by calling the line connection algorithm. Feature data set initialization and construction are performed by STRDS and ENTDS respectively. MMS tape generation is initiated by TRMDS upon successful completion of the feature data set generation. Statistical information compiled during program execution is output via the teleprinter and control is returned to the system MONITOR.

(2) Program Description

During system initialization CNTRL receives via the on-line teletype the specific input parameters which are to be used to process the selected ACSD tapes. Feature line segments are represented as data set blocks (mother block and its associated daughter blocks). Each raster scan element (lineal or areal) is mapped into each active data set in search of a neighborhood correlation. If a raster element meets the neighborhood criteria of an active data set, it is entered into that active data set, otherwise the raster element is used to create a new data set. As raster elements in the scan line buffer are exhausted, a new scan line record is introduced. As a data set block becomes full, the block is catalogued and transferred from active memory to disk for retrieval upon final termination of the line connection process.

As feature data sets are created from positional data represented on the base manuscript, symbolic color data represented on the color overlay is correlated to the feature data sets. Color overlay data is compiled for each feature data set. Upon unconditional feature data set termination, the color information is included in the mother data set block of the feature data set.

A data set is not immediately terminated if no raster element can be associated to that data set for a particular scan line. This data set remains active for N scan lines as requested during system initialization by the gap parameter. If N scan lines are processed without a coordinate update of an active data set, the data set is unconditionally terminated.

Conditional data set termination occurs when a convergent data condition is recognized by CNTRL, i. e., when multiple data sets are related to a single raster element. When convergence occurs, the converging data sets are conditionally terminated and the object raster element is used to create a new data set which is marked as a convergent data set and is linked to the above related data sets. The convergent data set is eventually used to define the intersection point of the convergent data. Dependent upon the value of the minimal feature length parameter entered via the teletype during system initialization the convergent data set completely defines the intersection point (minima) or is of valid length to be a distinct line segment feature of its own. Once the intersection point is established, it is entered as the last point of its associated conditionally terminated data sets which are then unconditionally terminated. If two data sets converge into a single data set which is under minimal feature length, the two data sets remain correlated to each other. If either more than

two data sets converge into a single data set or the convergent data set is of minimal feature length, the convergent condition correlations are erased.

Divergent data conditions (maxima), i. e., when multiple raster elements are related to a single active data set, also require the resolution of an intersection point. In this case the single active data set is the divergent data set and is used to resolve the intersection point of the divergent data. Again, if the data set is under minimal feature length, the data is not a valid feature and is used only to resolve the intersection point which is used to create the new data sets defined by the divergent raster elements. If the divergent data set is under minimal feature length and only two data sets diverge from it, the two data sets are correlated to each other. If either the divergent data set is of minimal feature length or more than two data sets diverge from the divergent data set, the divergent condition correlations are not created.

All data sets which are unconditionally terminated, uncorrelated to any other data sets, and are under minimal feature length are considered to be noise and are not transferred to disk as a valid line segment feature. All other data sets constitute valid line segment features and are transferred to disk.

When line connection is completed CNTRL calls TRMDS to create a lineal file tape in MMS format. Color information compiled during feature data set generation is used to create the feature header record. Statistical information compiled during execution of the line connection function is output via the teleprinter upon successful program completion.

(3) Inputs

(a) Calling Sequence -

None - stand alone program

(b) Keyboard Entries -- Operator Input Response

"↑P"
"X↑P"
"XX↑P"

(c) Magnetic Tape -

1 raw ACSD base manuscript file.

1 raw ACSD color overlay file.

(4) Outputs

(a) Teleprinter -

Program generated query and administrative messages:

"MOUNT ACSD MANUSCRIPT TAPE ON DRIVE 3. ↑P"

"MOUNT ACSD OVERLAY TAPE ON DRIVE 5. ↑P"

"MOUNT SCRATCH TAPE ON DRIVE 6. ↑P"

"ENTER RESOLUTION AS 2, 4, 5, or 6.> "

"ENTER MIN AREA SETTING AS XX ELEMENTS.> "

"ENTER LINEAL NEIGHBORHOOD RANGE AS XX UNITS.> "

"ENTER AREAL NEIGHBORHOOD RANGE AS XX UNITS.> "

"ENTER MINIMAL FEATURE LENGTH AS XX RASTER UNITS.>"

"ENTER GAP PARAMETER AS X RASTER UNITS.> "

"ENTER BASE COLOR CODE AS XX.> "

Program generated statistical information.

(b) Magnetic Tape -

1 lineal formatted feature file tape.

(5) Reference Subprograms

CNCT
FILBUF
STRDS
ENTDS
TRMDS
EXTRCT
ALINE
WMAGTP
STATS
DSKRW
TTYOUT
TTYIN
DCDCOL
UPDATE

b. TRMDS (Terminate Data Set)

(1) Abstract

TRMDS creates the lineal formatted MMS data file. Once line connection is completed, the subprogram is called by CNTRL to transfer all the line segment features residing on disk to magnetic tape in MMS format. Multiple line segment features (mother data set block and 0 to N daughter data set blocks) may constitute an actual feature due to the nature of data set generation at points of maximum and minima. As each data set block is retrieved, TRMDS unpacks the coordinates, scales the coordinates by calling SCAL, converts the coordinates to MMS format and combines any multiply segmented features into a single line feature into a single line feature. TRMDS calls COLCLR to collect all the color information associated with a single line segment feature. Following the header information the coordinate data are transferred to magnetic tape by calling WMAGTP. TRMDS only processes the data existing between the first and last data set block allocated.

(2) Program Description

TRMDS uses DSKADD and DSKRW to search the disk in search of mother blocks. Once a mother block and its associated daughter blocks are located, their headers are checked for any correlation, i. e., another mother-daughter linkage which defines a continuation of the current line segment feature. If a correlation exists the correlated line segment feature becomes the current line segment feature and the above process is repeated until all the line segment features defining a single feature are known and the endpoint of the line feature is reached, or in the case of a closed feature, the correlation returns the chaining of line segment features to the original data set. If no correlation exists the subprogram immediately creates the MMS records describing the line segment feature.

All mother blocks used in creation of a single line feature are marked as used so that they will not again be processed as the disk is searched sequentially. Since all data set features are built with y-coordinates increasing in value, TRMDS processes every other line segment feature defining a single line in an inverse direction, i. e., at points of maxima and mimima the function describing the locus path changes in direction. Color information compiled for each feature data set and collected by COLCLR is used to create the feature header record. All coordinate data is unpacked, scaled, converted to GE floating format and written to magnetic tape. TRMDS also accumulates statistical information which is transferred to STATS to be output via the teleprinter.

(3) Inputs

Calling Sequence -

L	JMS*	TRMDS	
L + 1	SKP		Return location
L + 2		ARG	
L + 3	- -		Next sequential instruction

where:

ARG = address of data set argument.

where data set argument.

ARG1 = core address of data set block.

(4) Outputs

Magnetic Tape -

Lineal formatted feature data sets via subprogram WMAGTP.

(5) Referenced Subprograms

DSKRW
DSKADD
WMAGTP
PDP9GE
SCAL
COLCLR

c. EXTRCT (Data Point Extraction)

(1) Abstract

EXTRCT extracts raster elements (lineal and areal) from a scan line buffer.

(2) Program Description

EXTRCT extracts the next raster element in the specified line buffer and returns the raster element (one word if lineal; two words if areal) to the calling programs. EXTRCT also returns the color code associated with the line buffer entry or entries. Upon exit from the subprogram the value of the accumulator indicates the type of raster element returned. An end-of-line-buffer condition is indicated to

the calling program by a return to the location following the call arguments address.

(3) Inputs

Calling Sequence -

L	JMS*	EXTRCT	
L + 1	JMP	.+3	Return location
L + 2		ARG	
L + 3	JMP	EOD	End-of-data-return
L + 4	- -		Next sequential instruction

where:

ARG = address of buffer arguments.

where buffer arguments:

ARG1 = current line buffer address.

ARG2 = first data point (line center or begin area).

ARG3 = second data point-end area (areal data only).

ARG4 = entry or entries color code.

(4) Outputs

Memory -

Raster elements and element type (via ACC) to the calling program.

(5) Referenced Subprograms

None

d. ALINE (Intersection Resolution)

(1) Abstract

ALINE is called to resolve the intersection point of convergent and divergent data. The minimal feature length is used to determine whether the incoming

data set (divergence) or the outgoing data set (convergence) is of valid length to be a distinct feature of its own or is used only to resolve the intersection.

(2) Program Description

ALINE is called whenever a divergent or convergent data transition occurs. In either case a common intersection point must be defined. In the convergent case resolution of the intersection point is delayed for N scan lines until the outgoing data set is of valid length or is terminated. When either occurs the convergent intersection point becomes the last data entry in all incoming data sets which are then terminated and the first data entry in the outgoing data set, otherwise the outgoing data set is eliminated from the data base. For the divergent case resolution of the intersection point is immediate since the incoming data set contains the required data for resolution. If the incoming data set is of valid length, the divergent intersection point becomes the first data entry in all outgoing data sets and the last data entry in the incoming data set, otherwise the incoming data set is eliminated from the data base. The mother data set block of any data set feature of legitimate length is updated with color and linkage information via UPDATE.

(3) Inputs

Calling Sequence -

L	JMS*	ALINE	
L + 1	SKP		Return location
L + 2		ARG	
L + 3	- -		Next sequential instruction

where:

ARG = address of data set arguments.

where data set arguments:

ARG1 = core address of data set block.

NOTE: Contents of ACC indicate convergent or divergent data.

(4) Outputs

Memory -

Data set blocks with resolved intersection points.

(5) Referenced Subprograms

DSKRW
ENTDS
UPDATE

3. OBJECT-TIME SYSTEM AND SCIENCE LIBRARY ROUTINES

a. Object-Time System Routines

The following is a list of PDP-9 FORTRAN IV Object-Time System macro-subroutines that are loaded at execution time. The compiled FORTRAN programs which contain I/O statements include output calls in the form of globals to various system I/O handler routines. When the compiled object program is loaded, the compiler defined global symbols are utilized to search the FORTRAN library for the needed I/O handler routines. These are loaded into memory and the proper program control linkage established by the Link Loader. Detailed description for these programs are provided in the following DEC PDP-9 manual "Advanced Software System Programmer's Reference Manual"; order number DEC-9A-KFZA-D.

Program Name	Function	Ref. Manual
FIOPS	FIOPS provide the necessary call to IOPS (Input-Output Processing System) required by all FORTRAN I/O statements.	P. 11-10
OSTER(.ER)	To announce an error on the teletype.	P. 11-18
TIME	To provide the ability to record elapsed time in minutes and seconds.	P. 11-21
BCDIO	To process formatted READ/WRITE statements in FORTRAN IV programs and sub-programs.	P. 11-13
.SS	To calculate the array element address.	P. 11-12
STOP(.ST)	To process the STOP statement and return control to the monitor.	P. 11-15
SPMSG(.SP)	To print the octal number coded with STOP.	P. 11-17

b. Science Library Routines

The following PDP-9 Science I ibrary mathematical routines are utilized by the FISS programs. These programs are described in the DEC PDP-9 manual referenced in Section II, Subsection 3 .a above.

Routine Name	External Calls	Ref. Manual
ABS	.DA, REAL	P. III-5
FLOAT	.DA, REAL	P. III-5
SIGN	.DA, REAL	P. III-5
.DA(General Get Argument)	None	P. III-8
INTEGE(Integer Arithmetic)	REAL	P. III-8
REAL (Real Arithmetic)	None	P. III-8
.CB (Short Get Argument)	None	P. III-9

SECTION III

FISS FLOW DIAGRAMS

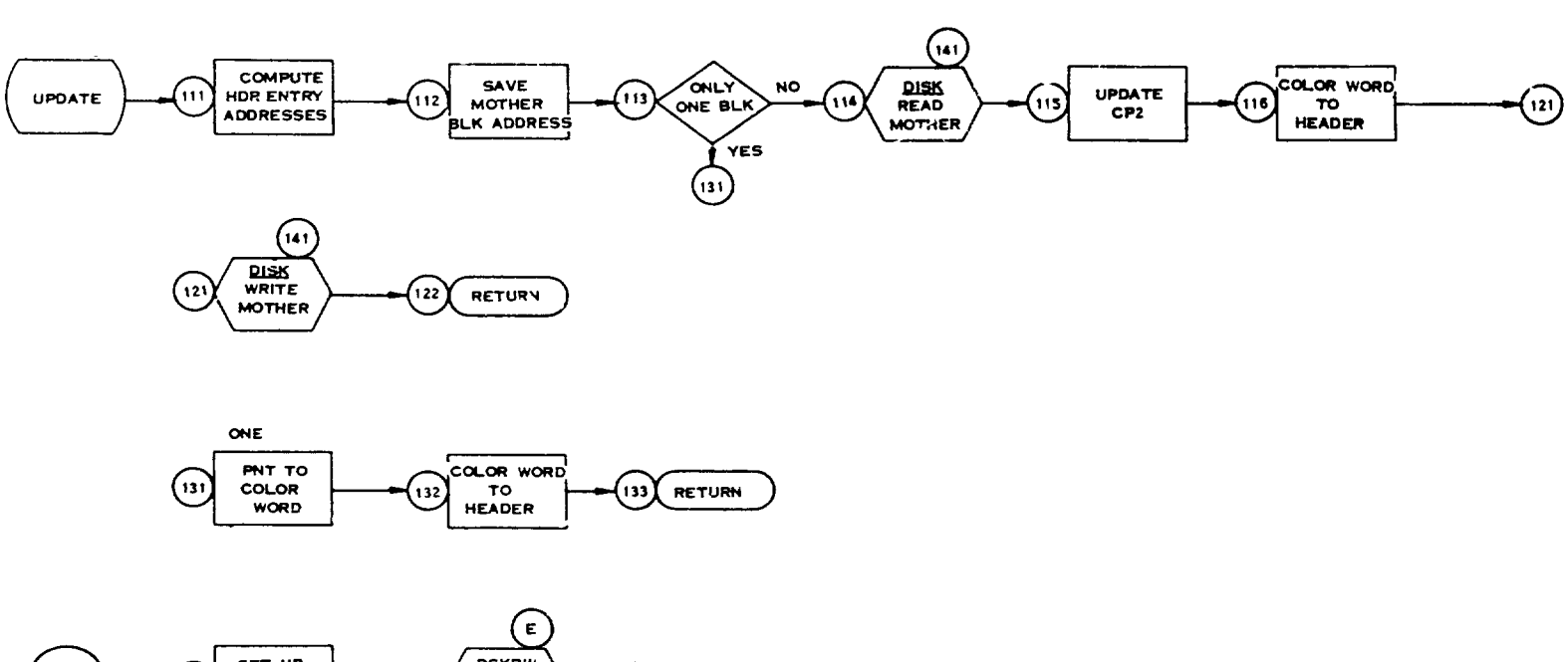


FIGURE II - 4. FLOW DIAGRAM OF UPDATE

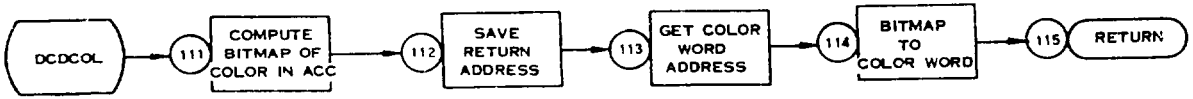


FIGURE II - 5. FLOW DIAGRAM OF DCDCOL

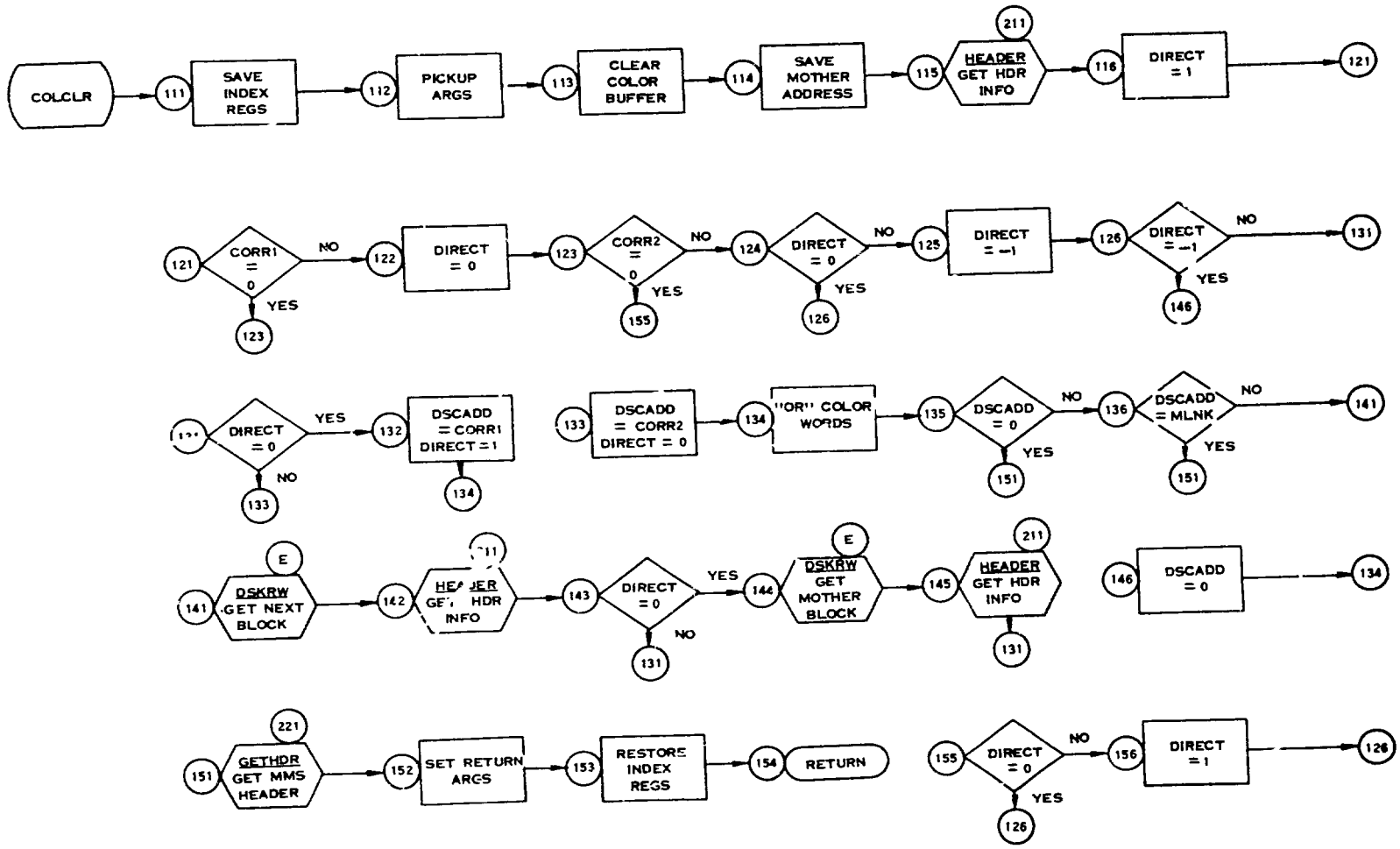


FIGURE II - 6. FLOW DIAGRAM OF COLCLR

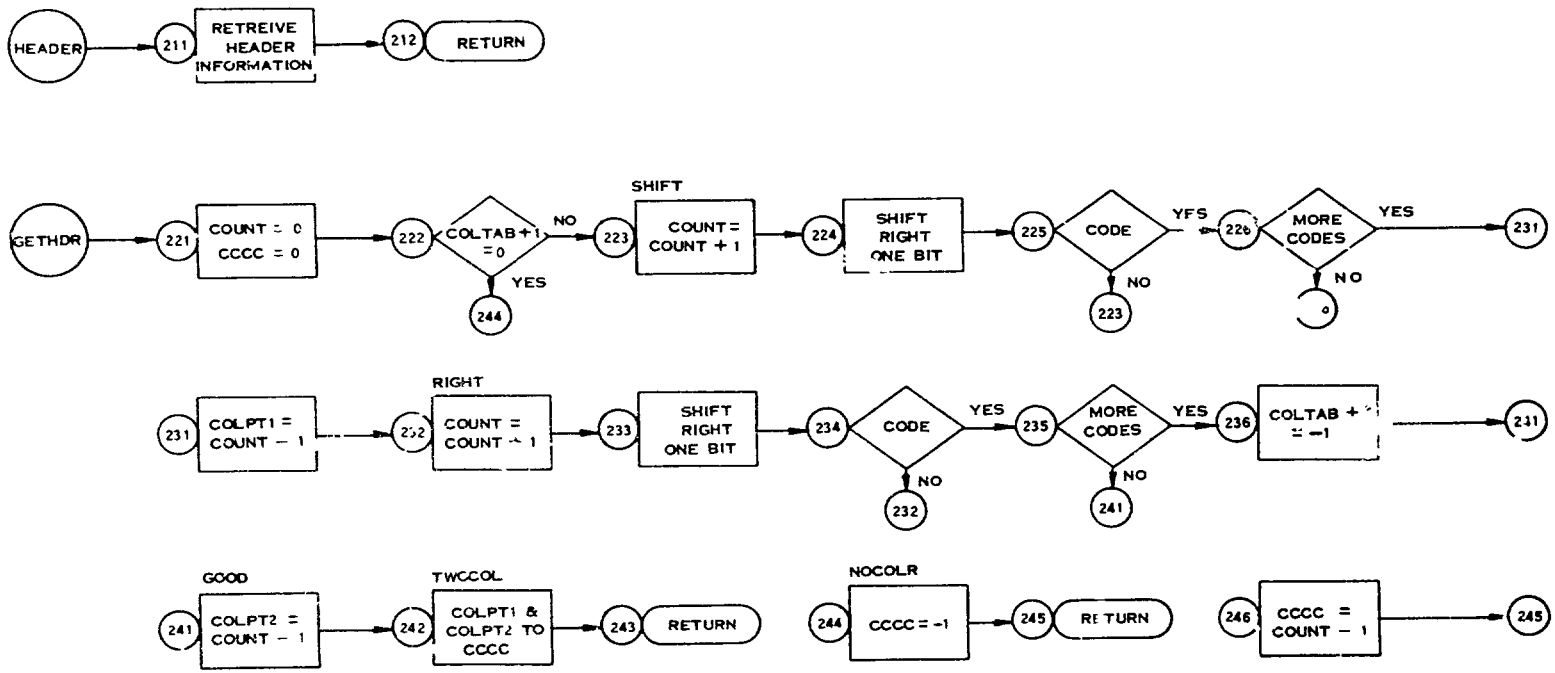


FIGURE II - 6. FLOW DIAGRAM OF COLCLR (CONCLUDED)

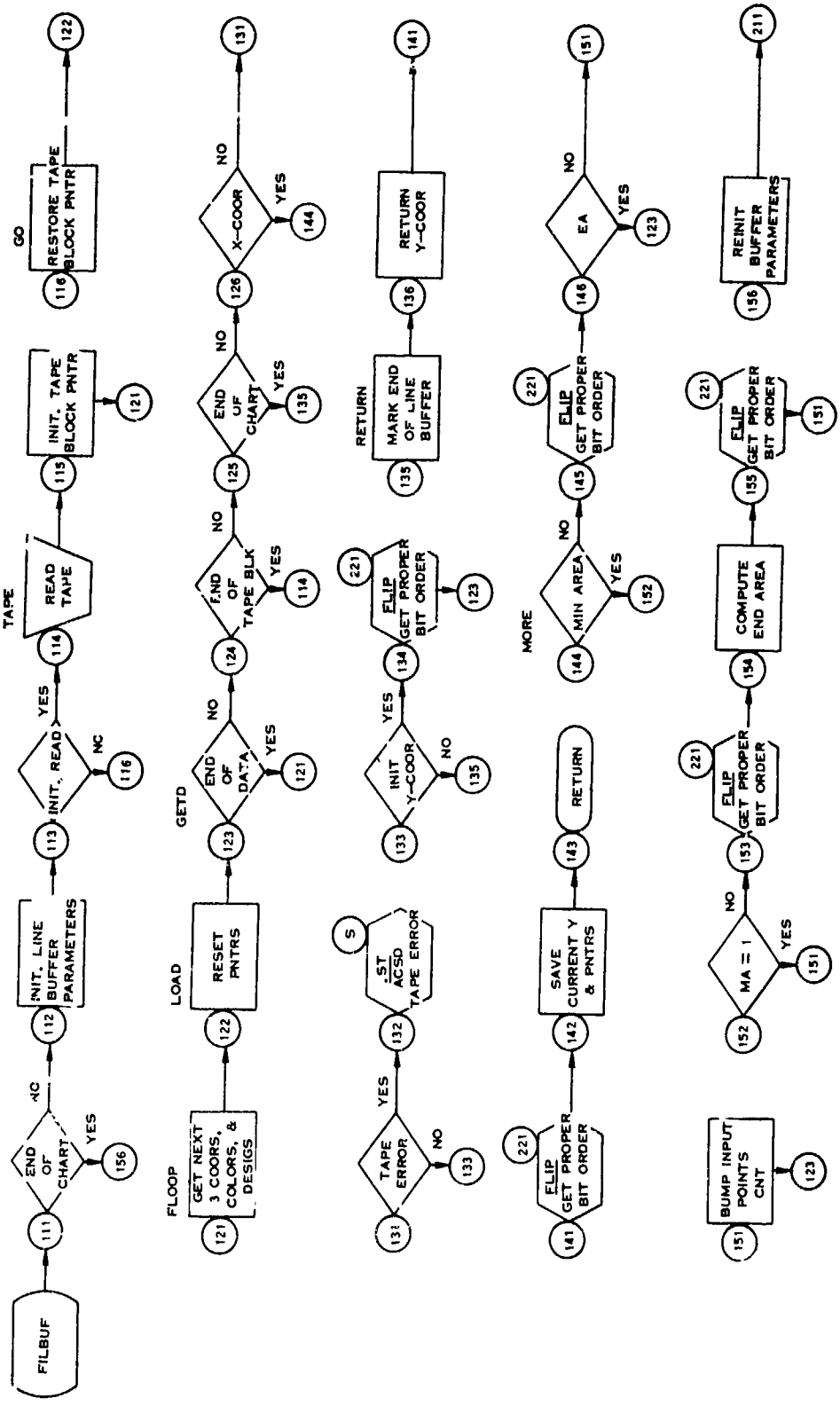


FIGURE II - 7. FLOW DIAGRAM OF FILBUF

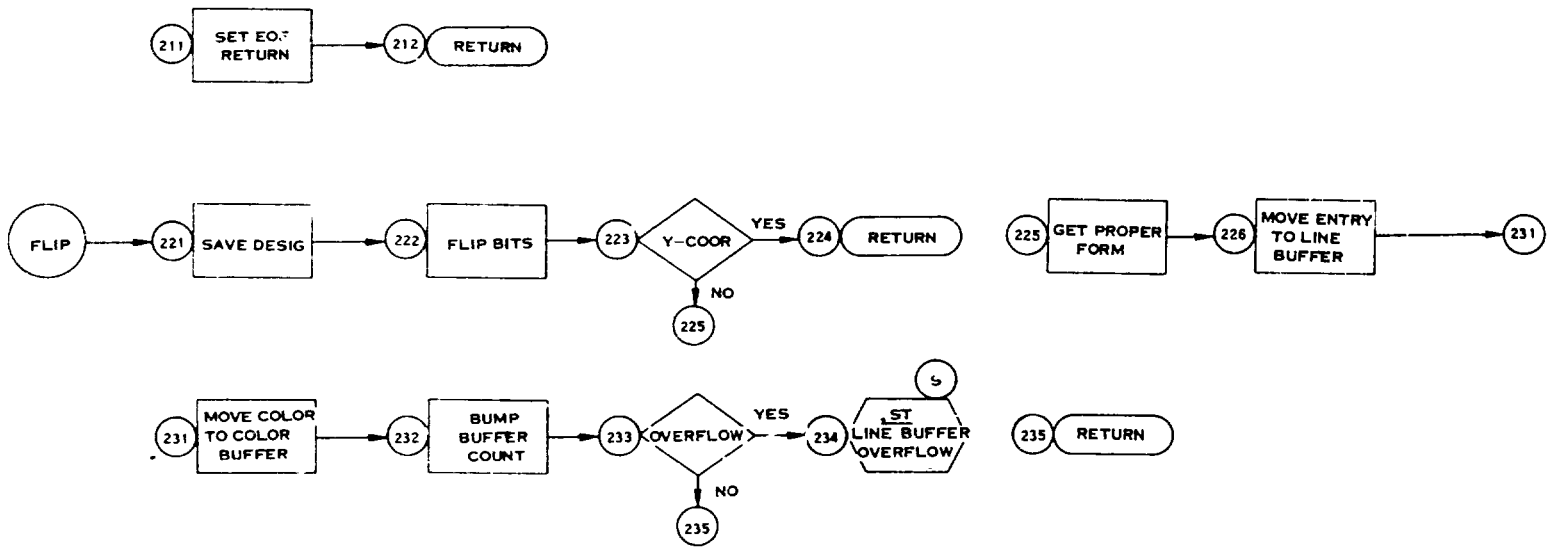


FIGURE II - 7. FLOW DIAGRAM OF FILBUF (CONDLUDED)

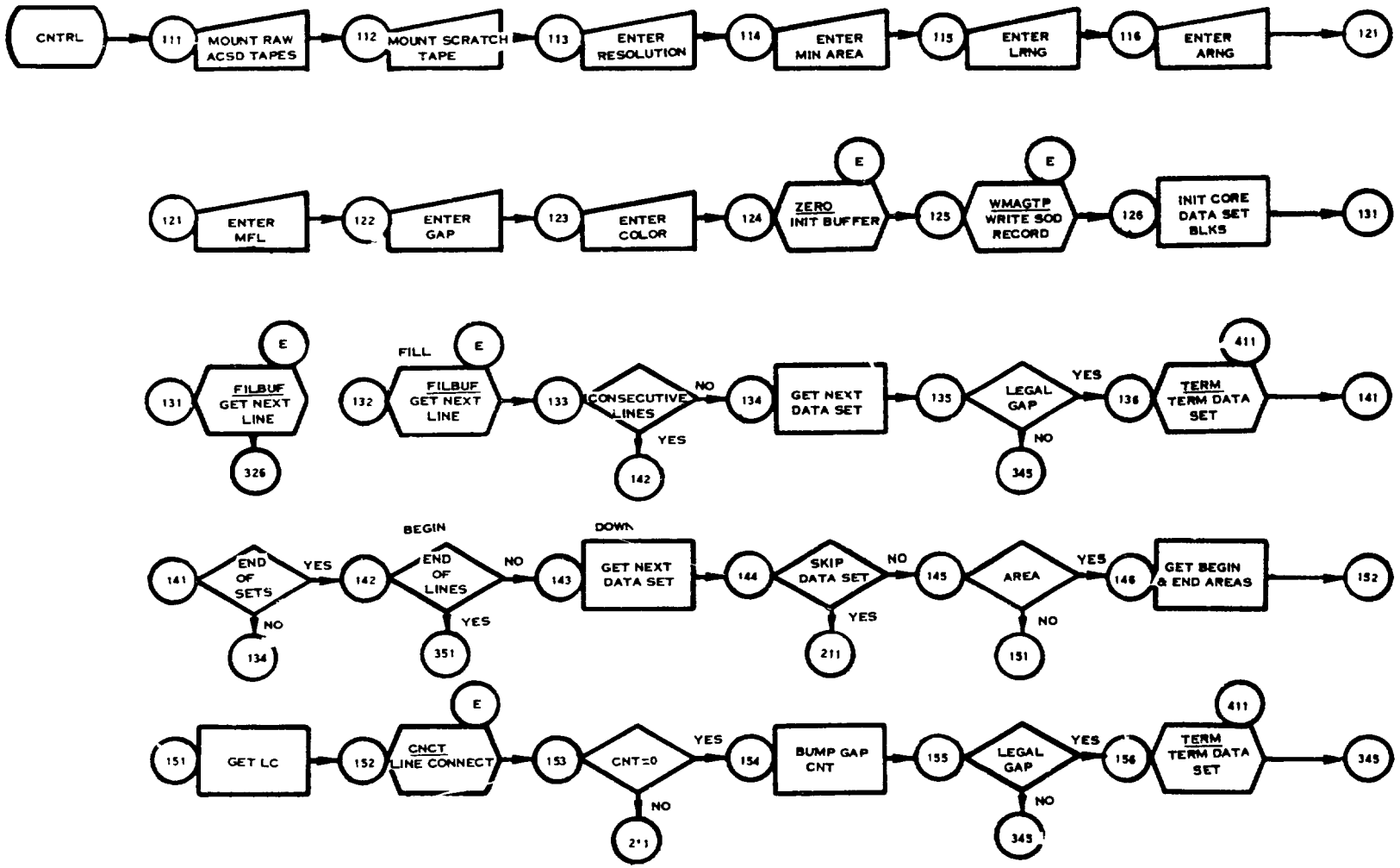


FIGURE II - 8. FLOW DIAGRAM OF CNTRL

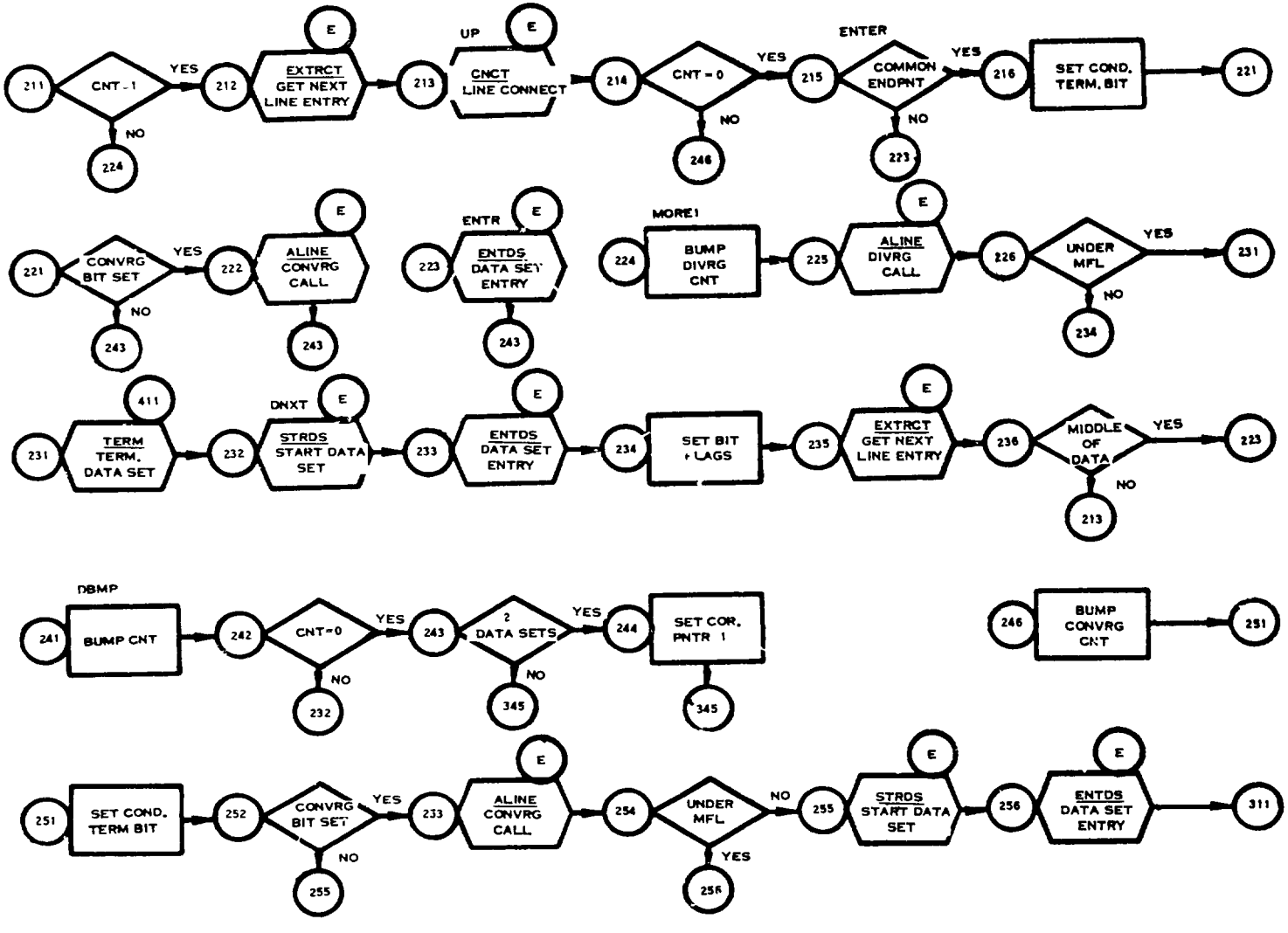


FIGURE II - 8. FLOW DIAGRAM OF CNTRL (CONTINUED)

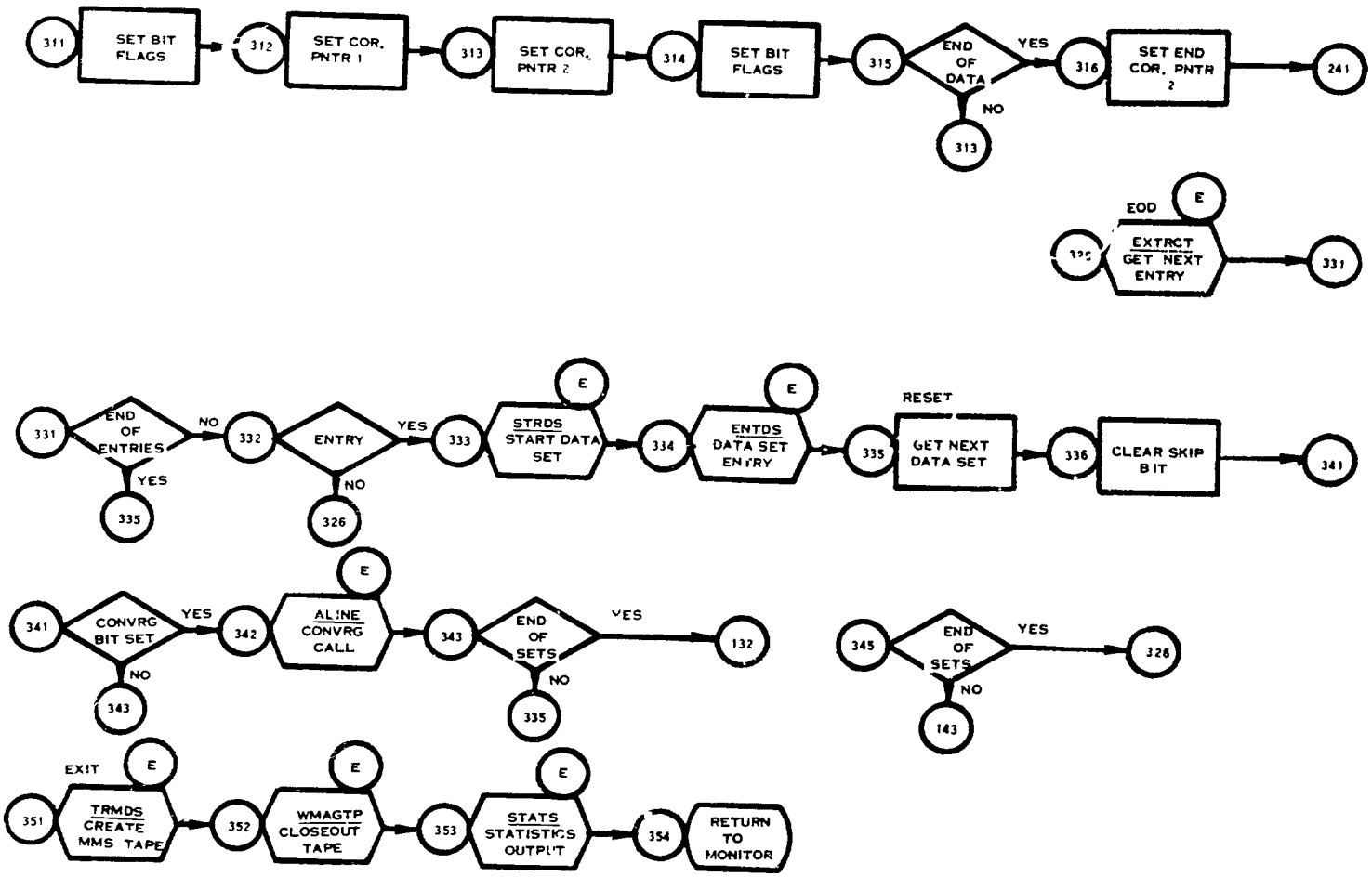


FIGURE II - 8. FLOW DIAGRAM OF CNTRL (CONTINUED)

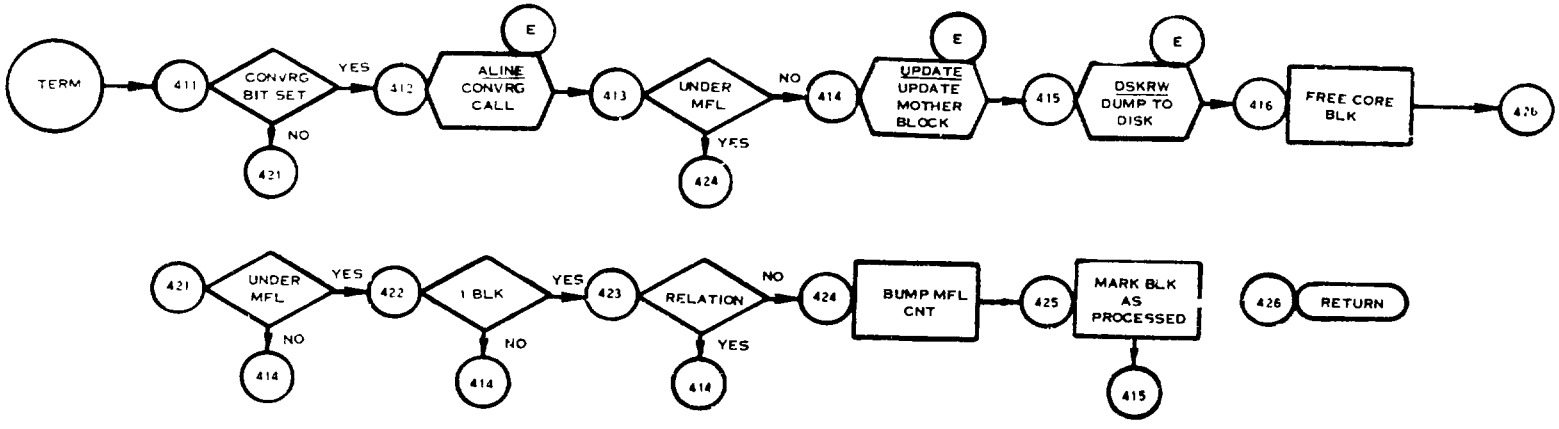


FIGURE II - 8 FLOW DIAGRAM OF CNTRL (CONCLUDED)

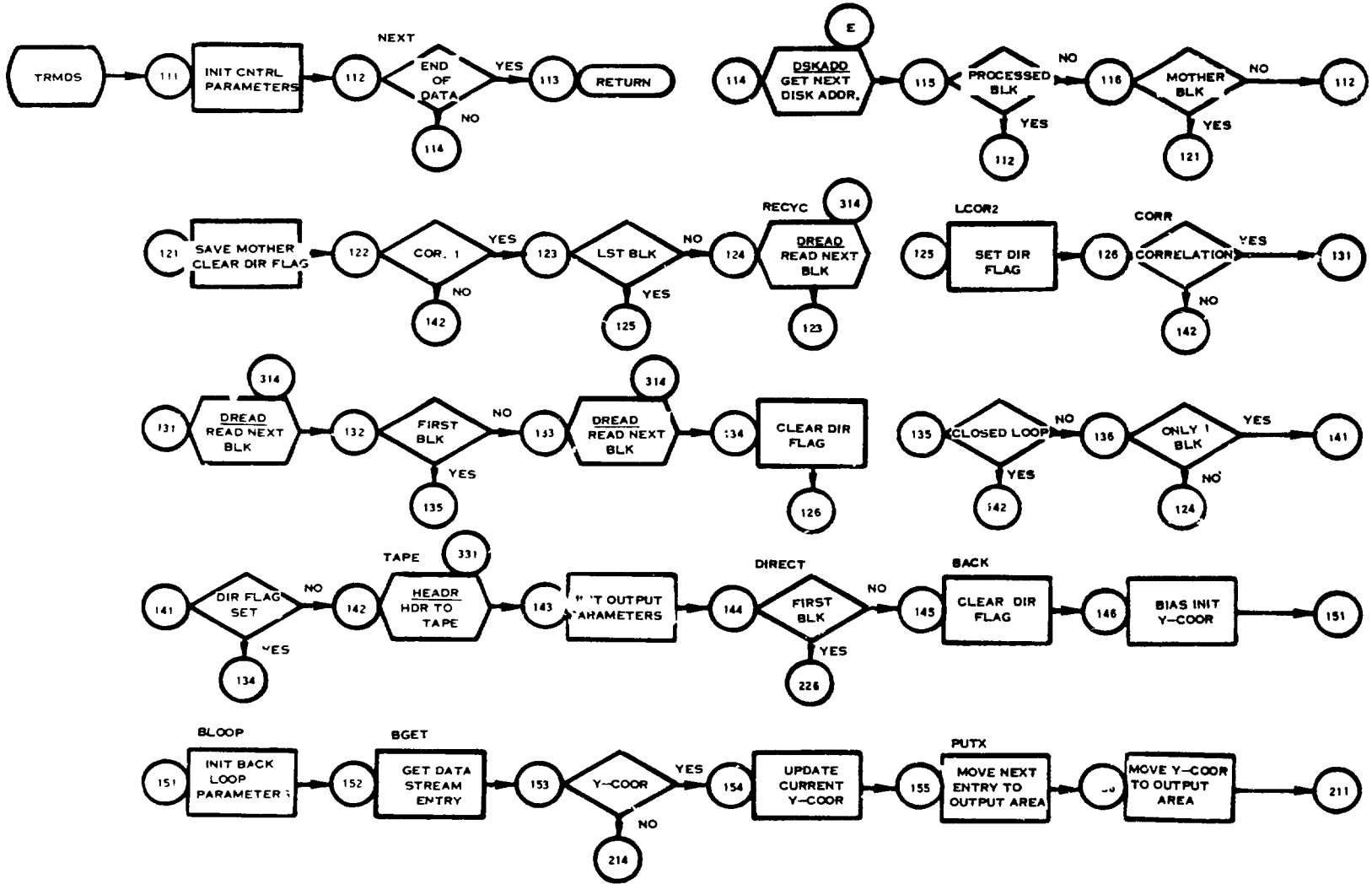


FIGURE II - 9. FLOW DIAGRAM OF TRMDS

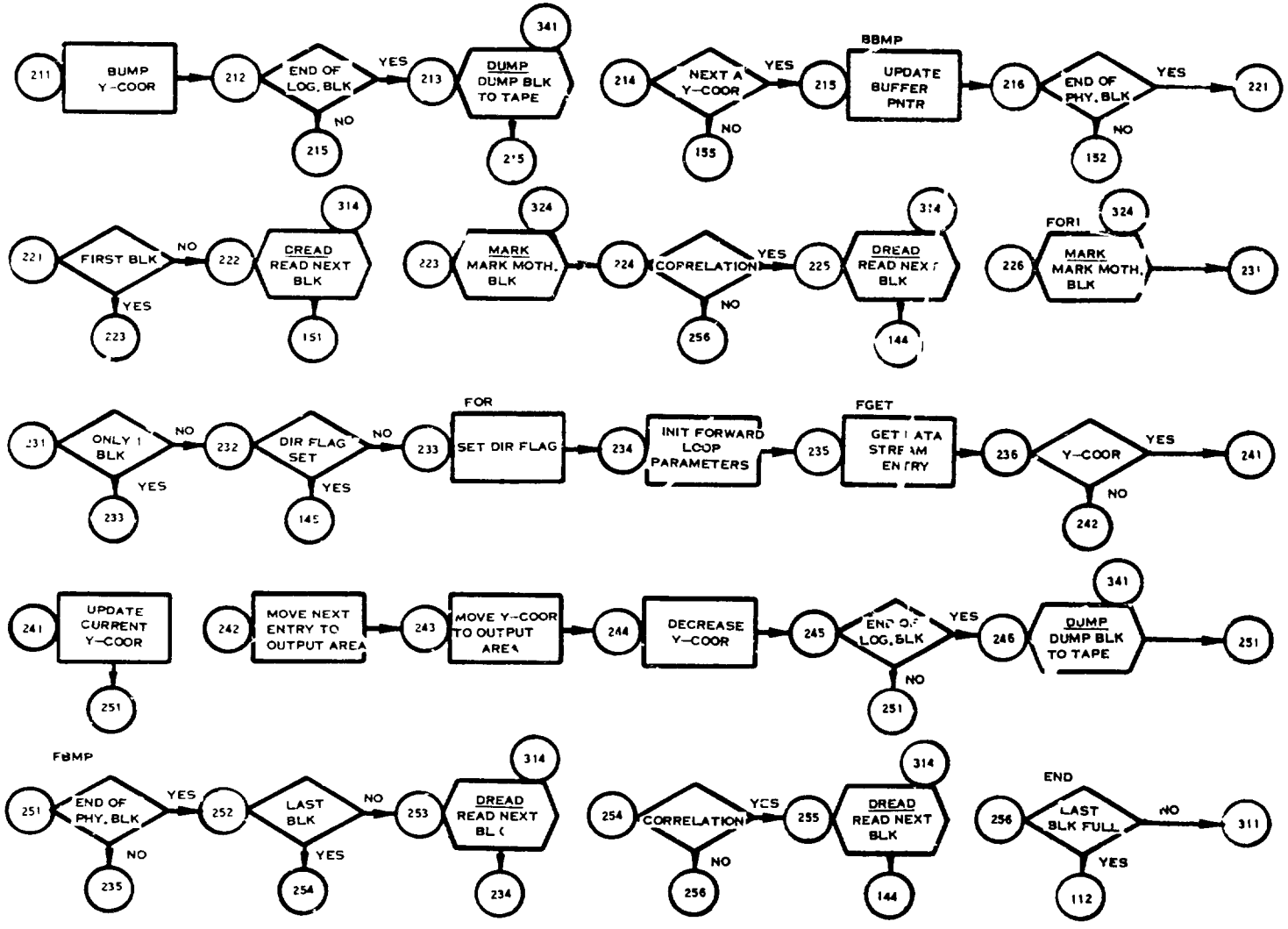


FIGURE II - 9. FLOW DIAGRAM OF TRMDS (CONTINUED)

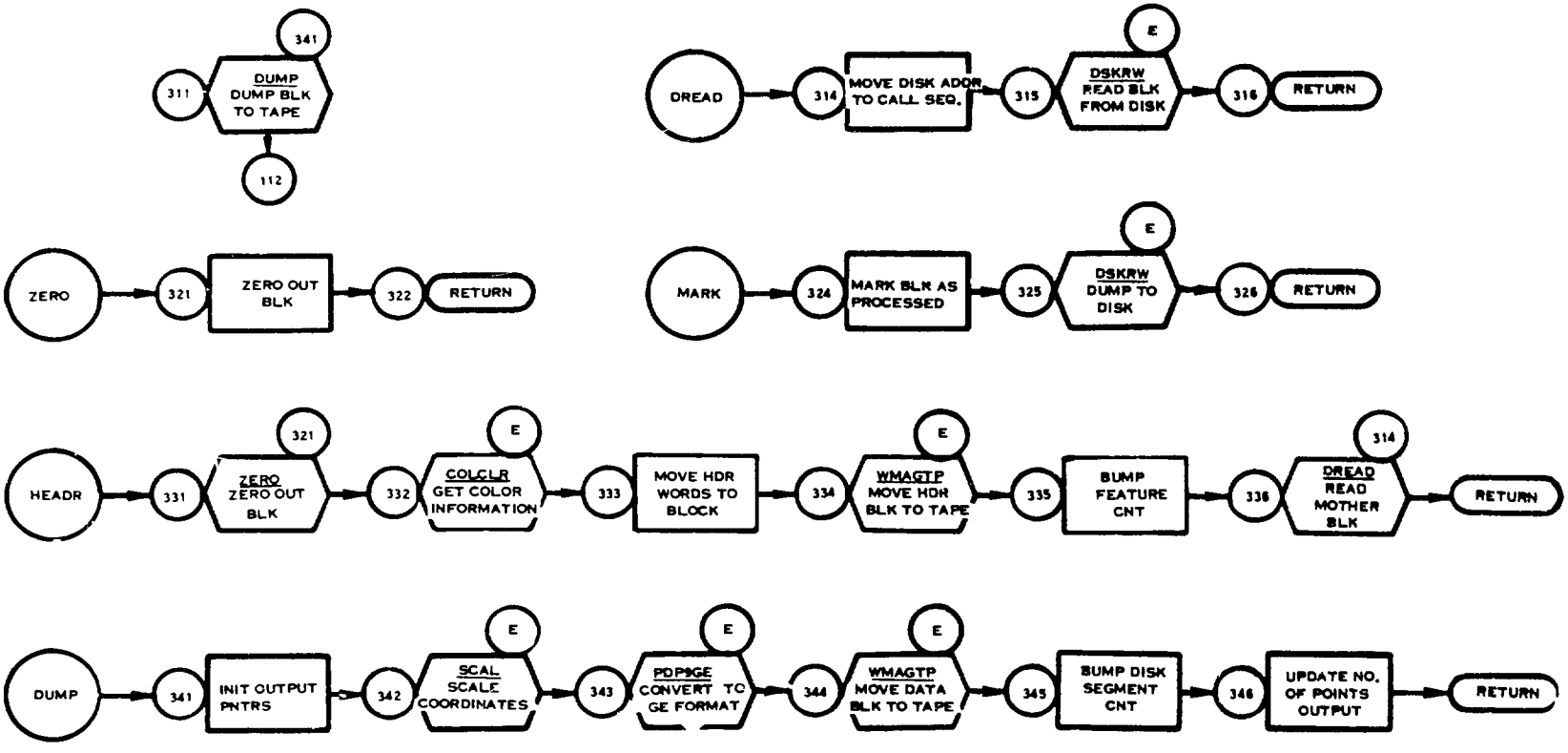


FIGURE II - 9. FLOW DIAGRAM OF TRMDS (CONCLUDED)

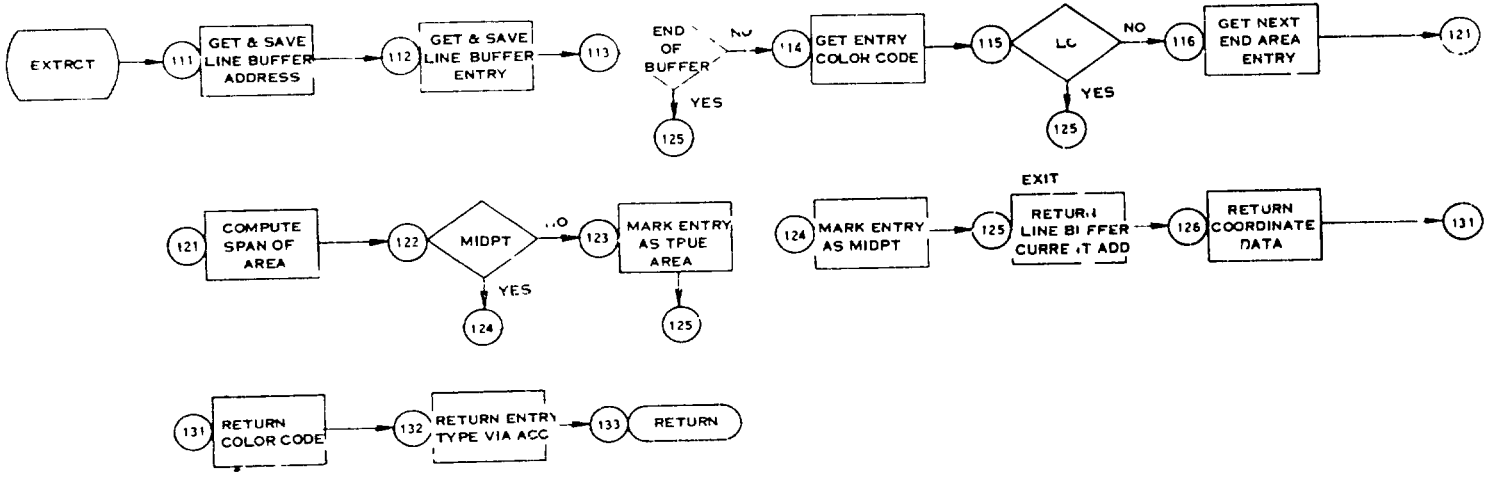


FIGURE 11 - 10. FLOW DIAGRAM OF EXTRCT

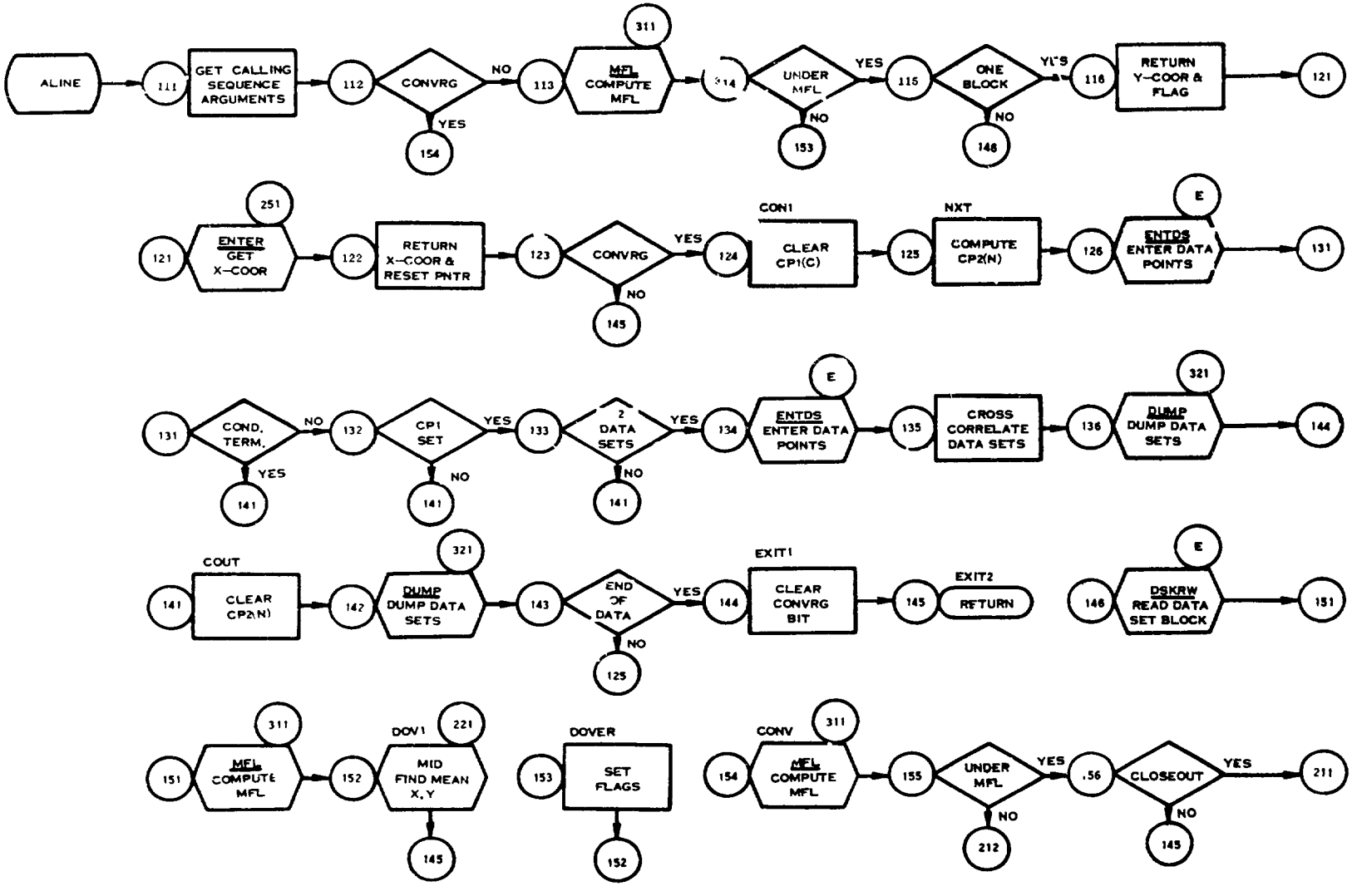


FIGURE II - 11. FLOW DIAGRAM OF ALINE

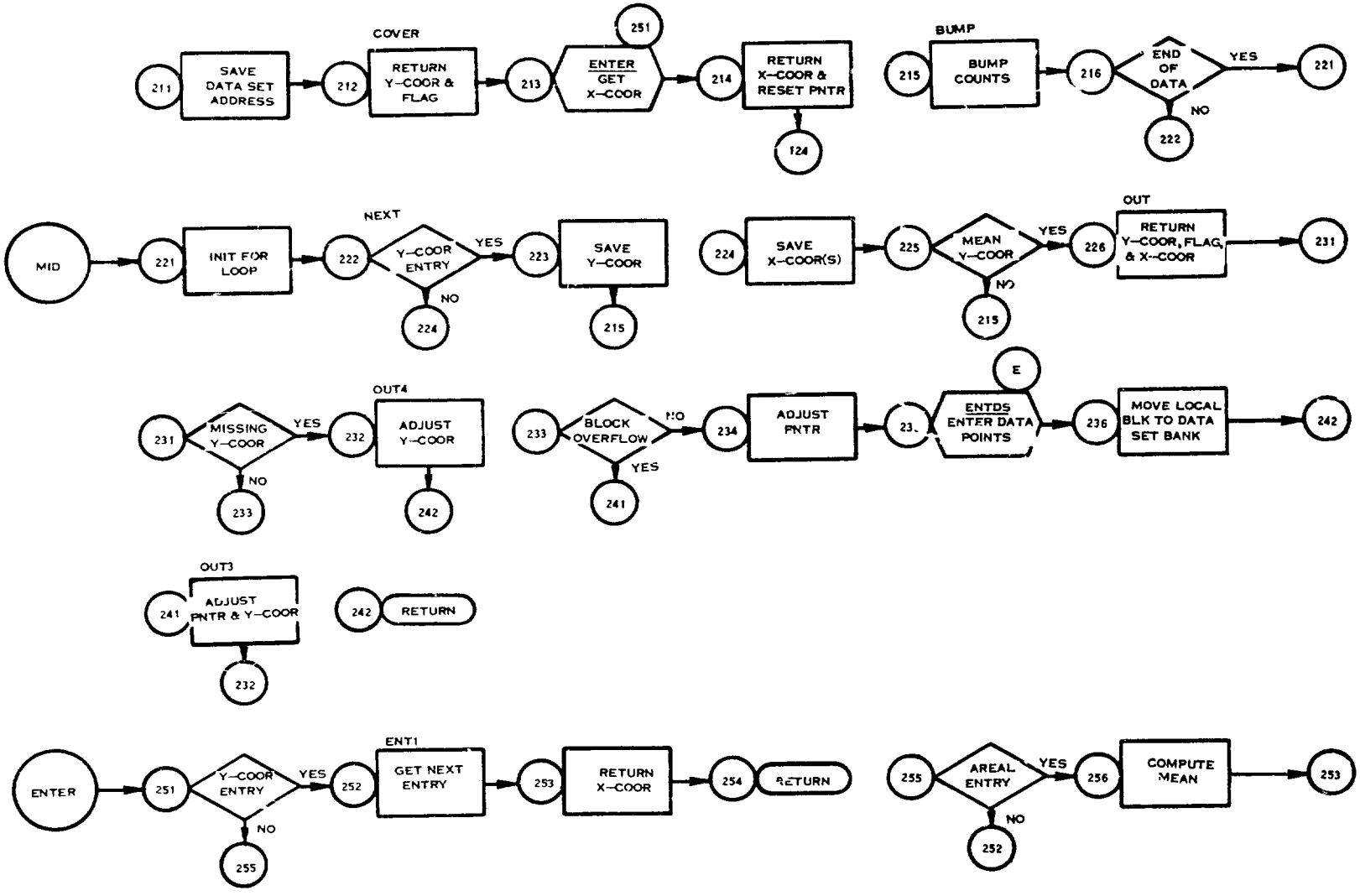


FIGURE II - 11. FLOW DIAGRAM OF ALINE (CONTINUED)

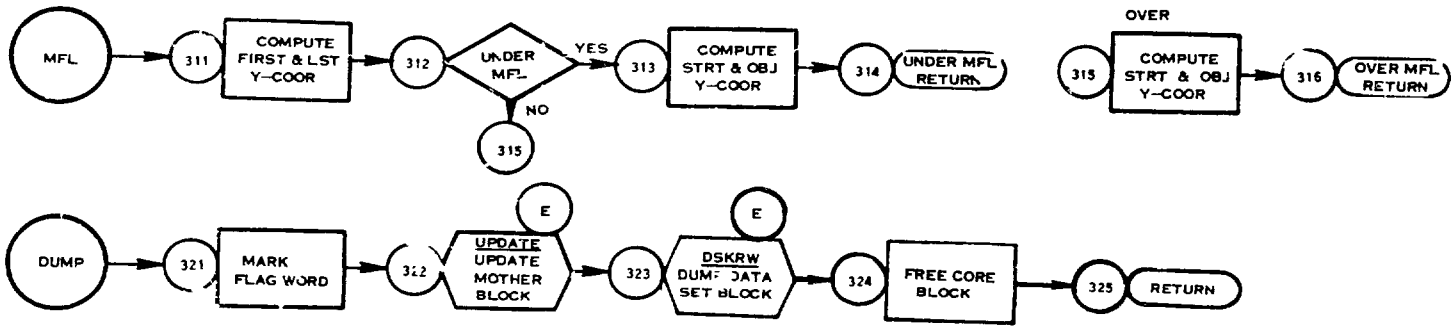


FIGURE II - 11. FLOW DIAGRAM OF ALINE (CONCLUDED)

APPENDIX III

RASTER PLOTTER SOFTWARE SYSTEM OPERATING INSTRUCTIONS

APPENDIX III

TABLE OF CONTENTS

APPENDIX III
INDEX OF DIAGRAMS

SECTION I

DISK PROGRAM CATALOGING PROCEDURE

1. GENERAL

The following paragraphs outline in detail the procedural steps needed to build a catalog of disk-resident PDP-9 operating system and RPSS applications programs. The RPSS is designed as a stand alone program and consists of one load module of twenty-one (21) subprograms. These, in addition to the PDP-9 Operating System and Science Library reside on DECtape as object programs. In order to build an executable load module of these programs on disk storage, they must be initially loaded into core memory, and the resulting core image transferred to disk file storage. The following instructions describe this procedure.

2. DECTAPE-TO-DISK LOADING INSTRUCTIONS

a. Object Program Tape Mounting Instructions

(1) Mount object program DECtapes as follows:

. DECtape #572 as DT 0 (PDP-9 Operating System)

. DECtape #568 as DT 2 (RPSS Application System)

b. Load System Maintenance Routine, DSKSAV

(1) Place program, DSKSAV, in console paper tape reader

(2) Set console ADDRESS switches to 77720 (octal).

(3) Depress the following console function keys in the order indicated below:

. STOP	-	Stop machine
. I/O RESET	-	Clear pending I/O and interrupt system status indicators, flags, and registers
. READIN	-	Initiate hard-wired "bootstrap" paper tape loading sequence.

c. Select DECtape-to-Disk Program Loading Options

Place the following PDP-9 DISK CONTROL UNIT panel switches in the "OFF"

(down) or "ON" (up) positions at the completion of the DSKSAV loading operation. The operator will be signalled at the completion of this operation by the following system generated administrative message.

"DSKSAV"

Disk control panel settings:

- Control switches 0-29 - "OFF"
- WRITE PROTECT switch - "OFF"

d. Transfer PDP-9 Operating System Object Programs from DECtape-to-Memory-to-Disk File Storage

(1) System maintenance program, DSKSAV, loads object coded executive system programs from DECtape to active core memory. The resulting core image program is then transferred onto disk. To initiate the execution of this function, perform the following sequence of console operations:

- Set console ACCUMULATOR switches to 000000_8
- Depress CONTINUE

(2) Upon the successful completion of system program transfer from DECtape 0 to disk storage, the following administrative message is typed on-line via the console keyboard/page printer:

"DSKSAV"

e. Reset Disk Control Panel Options to Initiate the Loading and Building of an RPSS Disk Resident Program File

(1) Place the following PDP-9 Disk Control Unit panel switches in the "OFF" (down) or "ON" (up) positions:

- Control switches 0-29 - "ON"
- WRITE PROTECT switch - "ON"

f. Transfer RPSS Object Programs from DECtape-to-Memory-to-Disk File Storage

(1) Initiate transfer of RPSS object program from DECtape 2 to disk file storage as follows:

. Set console ACCUMULATOR switches to 0000002_8

. Depress CONTINUE

(2) The following message will be typed on-line when the program load and transfer operation previously described has been completed:

"DSKSAV"

3. DISK-TO-CORE PDP-9 MONITOR LOADING INSTRUCTIONS

a. Load the Disk File Resident MONITOR System Into Active Core Memory

(1) Place system utility program, DISK, in console paper tape reader

(2) Set console ADDRESS switches to 77637_8

(3) Depress the following console function keys in the order specified below:

. STOP

. I/O RESET

. READIN

(4) Upon the successful completion of the above program loading function, the following system generated administrative message is typed on-line:

"MONITOR V4E"

(5) The above message signals the termination of the operating and applications program system disk loading sequence, and indicates that the core resident MONITOR program is available for use.

SECTION II

RASTER PLOTTER SOFTWARE SYSTEM (RPSS) OPERATING INSTRUCTIONS

1. GENERAL

The paragraphs which follow provide the stepwise instructions and procedures for 1) loading and 2) initializing the RPSS applications program. As previously described in SECTION I, the RPSS object program is initially transferred from DEC-tape to disk file memory storage media prior to its execution by the PDP-9 hardware and executive software system. Both the program loading from disk and initialization at time of execution are performed via the on-line console teletype (ASR33). The user interacts with the memory resident PDP-9 MONITOR system to initiate and implement program link loading functions, to enter required constants, settable control parameters, and data. The on-line procedures required to enact these two functions are described separately to avoid confusion.

In the following paragraph the following notational convention is employed to differentiate between typed message outputs generated by either the MONITOR or RPSS system, and response messages entered by the user during the enactment of program loading and intialization procedures. As shown in the example below all program generated output messages are underscored, while user inputs are not.

Example:

System Output Message: MONITOR V4E

Operator Input Message: LOAD

2. RPSS PROGRAM LOADING INSTRUCTIONS

a. MONITOR V4E

System generated message to signal the operator that the system MONITOR is core memory resident and available for use (see Paragraph 3.0 SECTION I).

b. $

System generated mnemonic indicating that the MONITOR is ready to receive operator inputs.

c. A MTF7 7

Enter above message to assign magnetic tape drive to logical unit 7.

d. LOAD

Entry of above message will cause the MONITOR to read the disk operating system link loader routine, LOAD, into active core memory.

e. LOADER V4A

System response message typed on-line after the retrieval and loading of program, LOAD, has been completed. This is followed by the system generated mnemonic;

>

which signals that, LOAD, is memory resident and available to accept user inputs.

f. A, B, C .....T, U

Mnemonics A, B ...., U are symbolic names assigned to the main and sub-programs constituting the RPSS object program load modules. The above input message must be terminated by depressing the ALT MODE Key.

g. RPSS	74451
.	.
.	.
.	.
.CB	60053

Core memory map output in response to input message 2,g above. Map contains names and start locations of RPSS, PDP-9 Object-Time and Science Library programs and routines loaded into active memory by the link loader program, LOAD.

h. $S

Operating system generated mnemonic typed on-line to signal the completion of the disk-to-core memory program link loading procedure.

At this point the "virtual" core image of the RPSS and required system's library and object-time programs reside in memory.

```
MONITOR V4E

$A MTF7 7

$LOAD


LOADER V4A
>A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U
 RPSS    74466
 SCODE   73473
 MKENT   72414
 GNR     71606
 SCAN    71515
 SHLSRT  71271
 SRCH    71136
 INTWDL  71012
 OUTWDL  70701
 PBYTE   70555
 IMESG   70227
 NPAGE   70001
 PRNSTT  67331
 NSET    67225
 SETA    67115
 BLK     67067
 SETM    67014
 SETC    66751
 BITS    66542
 MGFRW   66346
 BSPEOF  66161
 MTF.    65001
 LPA.    64316
 DTA.    53416
 DKA.    47056
 TIME    64243
 .DA     64174
 BCDIO   61145
 AUXIO   61047
 .SS     60767
 GOTO    60741
 STOP    60726
 PAUSE   60712
 SPMSG   60617
 FIOPS   60057
 OTSER   46762
 INTEGE  46643
 REAL    45607
 .CB     60037
↑S↑Q6
```

FIGURE III-1. RPSS PROGRAM LOAD MAP

j. ↑Q6

Entry of the above response input message causes the MONITOR to transfer the current core image to AREA 6 of disk file memory.

k. MONITOR V4E

Operating System generated administrative message output to signal the completion of the previsously described core-to-disk memory function (see 2. j.).

l. $

Operating System generated mnemonic output on-line to signal the "ready" status of the MONITOR; i.e., to receive user inputs.

m. G 6

Entry of the above input message cause the load module residing in disk AREA 6 to be read into active memory.

n. ↑S

Entry of the "Control S" message causes control to be transferred from the MONITOR to the entry point of the main program, RPSS. Execution of the application program will begin at this point with the initialization function. This is described in further detail in Paragraph 3.0.

3. PROGRAM INITIALIZATION AND CONTROL INSTRUCTIONS

This section provides a detailed description of the procedural steps needed to initialize the Raster Plotter Software System. The program initialization process is implemented by means of a query-response control mechanism. The on-line teletype serves as the primary communications link between man and machine, and is employed to generate administrative messages instructing the user of the specific data and control parameters needed for normal execution; to generate diagnostic messages signaling the type of keying or procedural errors committed by the user during the initialization process; and to serve as the input medium for user response messages.

All user input messages must be terminated by a carriage return, unless otherwise indicated.

The RPSS is designed to execute in one of two selectable operational modes; Automatic or "Tape Mode" and Semi-automatic or "Manual Mode".

a. Tape Mode (Automatic) -- This mode of operation processes all the raster formatted records appearing on the input data file in a serial manner, translating

each from its raw to its final raster record format. Provisions are not included to permit manual selection of data conversion based upon either channel or density parameters, or both. Instructions for initializing this mode of operation are provided in Paragraph 4.

b. Manual Mode (semi-automatic) -- This is the alternate mode of operation and provides the user with provisions for feature class selection criteria based upon various channel and density combinations. There are thirty-two (32) density/channel combinations possible, and these may be specified and assigned to known feature types at time of execution by means of the on-line Keyboard entry device. The stepwise procedures for enacting this option are described in Paragraph 6.

4. TAPE MODE INITIALIZATION INSTRUCTIONS

a. MOUNT NEW SCRATCH ON DRIVE n ENTER D WHEN DONE

Mount a full magnetic scratch tape, and set to the logical unit specified. When this task is completed respond with the following input message:

D (CR)

Note: (CR) - carriage return

b. ENTER MONTH, DAY, YEAR AS 6 NUMERICS ... MMDDYY

Enter date in the format specified by the above output message.

EXAMPLE: For February 18, 1971, enter:

021871 (CR)

c. ENTER 5 DIGIT TAPE ID NUMBER

Five digit tape identification number used to uniquely define the current output data file.

EXAMPLE

01239 (CR)

d. MOUNT FIRST REEL OF INPUT TAPE ON DRIVE m

Mount input data file and set to logical unit specified. NOTE, no response required after the completion of this operator task. However, failure to execute this step in the initialization sequence will result in a "conditional halt' when this unit is initially selected during the exectuion phase of RPSS.

e. ENTER 6 DIGIT MINIMUM AREA IN PROPER RESOLUTION

Entry equivalent to MIN AREA setting at which the source input graphic was processed by the ACSD. For example, if the machine setting was M. A. = 3, this would be entered as:

000003 (CR)

f. DO YOUR WISH TO SUPPLY SELECTION CODES EXTERNALLY

ENTER Y OR N

System generated message stating the option modes of operation. A message response of:

N (CR) - Indicates "TAPE MODE" selection

Y (CR) - Indicates "MANUAL MODE" selection

g. Enter N (CR)

h. ARE THERE MORE INPUT TAPES FOR THIS RUN IN ADDITION TO THAT

CURRENTLY MOUNTED ON DRIVE m ENTER Y OR N

Y (CR) denotes multi-reel input file

N (CR) denotes single-reel input file

i. MOUNT NEXT INPUT ON DRIVE i

This output message is only printed if the response to the previous query message (4 h) is affirmative (i.e., Y). No operator response is required at the completion of this task.

j. ENTER D IF PROPER INPUT TAPE MOUNTED ON DRIVE m

The above message is generated irrespective of the response to message 4.h. Its purpose is to proved an additional check point for input tape assignment validation.

Respond with the following input message for acknowledgement of input file assignment:

D (CR)

k. Entry of the above input message initiates the input data file(s) format conversion process. No further messages will appear until either the input data file is exhausted or the output is filled. In the latter case the following system generated message will appear.

l. AT PAUSE 2, MOUNT NEW SCRATCH ON DRIVE n AND CONTINUE

PAUSE 000002

When the above messages appear, remove the tape from the unit specified by numeric, n, and replace it with a full reel of magnetic tape.

To continue, enter the following input message:

↑P (CNTRL P) - Causes processing to continued from the point of interruption utilizing a new output tape file.

m. FIRST Y COORDINATE IS yyyyy

Message to the operator indicating the Y address of the first scan line encountered so that the graphic plotter carriage address may be set at this value. This number is in decimal. No operator response is required at the completion of this message.

n. BAD WRITE AT IWRITE = wwwww

Message to the operator indicating the probability of erroneous data being written on the output tape. The scan line where this occurs is in the vicinty of the first Y coordinate plus the value of IWRITE. This number is in decimal. No operator response is required at the completion of this message.

o. BAD READ AT IREAD = rrrrr

Message to the operator indicating the probablity of erroneous data being read from the input data tape. The scan line where this occurs is in the vicinity of the first Y coordinate plus the value of IREAD. This number is in decimal. No operator response is required at the completion of this message.

p. TOTAL READS = rrrrr

TOTAL WRITES = wwwww + 1

Message to the operator indicating the number of blocks of data read from the input tape and written onto the output tape. These numbers are in decimal. No operator response is required at the completion of this message.

```
MONITOR V4E

$G 6
↑S
MOUNT NEW SCRATCH ON DRIVE 5   ENTER   D   WHEN DONE

D
ENTER MONTH,DAY,YEAR AS 6 NUMERICS ••• MMDDYY
072071
ENTER 5 DIGIT TAPE ID NUMBER
11111
MOUNT 1ST REEL OF INPUT TAPE ON DRIVE 6
ENTER 6 DIGIT MINIMUM AREA IN PROPER RESOLUTION.

000003

DO YOU WISH TO SUPPLY SELECTION CODES EXTERNALLY
ENTER Y OR N
N
ARE THERE MORE INPUT TAPES FOR THIS RUN IN ADDITION TO THAT
CURRENTLY MOUNTED ON DRIVE 6   ENTER Y OR N

N

ENTER D IF PROPER INPUT TAPE IS MOUNTED ON DRIVE 6

D

FIRST Y COORDINATE IS    212
TOTAL READS =   330
TOTAL WRITES =   328 + 1
DO YOU WISH TO TERMINATE RUN   (Y  OR  N)

Y

REMOVE OUTPUT TAPE ON DRIVE 5
LABEL IT  DATE     7/20/71 TAPE NUMBER   11111 REEL NUMBER   0
END OF RUN - REMOVE ALL TAPES FROM DRIVES


                    *** TERMINATION ***

STOP  000000
```

FIGURE III-2. RPS5 PROGRAM RUN: TAPE MODE

5. TAPE MODE POST-PROCESSING PRINTOUTS AND RUN TERMINATION PROCEDURES

Upon completing an input data file processing cycle, a tabulation of run and I/O statistical data will be output on the PDP-9 line printer. These include elapsed run time, the number of input records processed, and the number output for each channel. At the conclusion of this file the following query-response dialogue is initiated.

a. DO YOU WISH TO TERMINATE RUN (Y OR N)

Y (CR) - Denotes unconditional termination of program. Control is returned to MONITOR system.

N (CR) - Denotes a request for an additional processing cycle.

b. DO YOU WISH TO STACK OUTPUT FILES (Y OR N)

The above message is output at the console ASR33, if, and only if, the response to message 5 a is a N. The responses to this message are as follows:

Y (CR) - Indicates the continued use of the current output tape file for the next processing cycle.

N (CR) - Denotes that a new output tape file is desired.

c. REMOVE OUTPUT TAPE ON DRIVE N
LABEL IT DATE mm/dd/yy, TAPE NUMBER nnnnn, REEL NUMBER rr

This message appears if the operator response to the previous query (5 b) is negative, i.e., a or if response to 5 b is affirmitive i.e., a Y. Operator response to this message is not required.

d. END OF RUN - REMOVE ALL TAPES FROM DRIVES

*** TERMINATION ***

This is the final message output by the system prior to termination and transfer of control to the resident MONITOR system.

6. MANUAL MODE INITILIZATION INSTRUCTIONS

a. MOUNT NEW SCRATCH ON DRIVE n ENTER D WHEN DONE

Mount a full magnetic scratch tape, and set to the logical unit specified. When this task is completed respond with the following input message:

D(CR)

Note: (CR = carriage return

b. ENTER MONTH, DAY, YEAR AS 6 NUMERICS ... MMDDYY

Enter date in the format specified by the above output message.

EXAMPLE: For February 18, 1971, enter:

021871 (CR)

c. ENTER 5 DIGIT TAPE ID NUMBER

Five digit tape identification number used to uniquely define the current output data file.

EXAMPLE

01239 (CR)

d. MOUNT FIRST REEL OF INPUT TAPE ON DRIVE m

Mount input data file and set to logical unit specified. NOTE, no response required after the completion of this operator task. However, failure to execute this step in the initialization sequence will result in a "conditional halt" when this unit is initially selected during the execution phase of RPSS.

e. ENTER 6 DIGIT MINIMUM AREA IN PROPER RESOLUTION

Entry equivalent to MIN AREA setting at which the source input graphic was processed by the ACSD. For example, if the machine setting was M.A. = 3, this would be entered as:

000003 (CR)

f. DO YOU WISH TO SUPPLY SELECTION CODES EXTERNALLY
ENTER Y OR N

System generated message stating the option modes of operation. A message response of:

N (CR) - Indicates "TAPE MODE" selection

Y (CR) - Indicates "MANUAL MODE' selection
Enter
Y (CR)

g. ENTER DEFAULT DENSITY AS NUMBERIC FROM 0 TO 7

Enter a default density so that each output will have a density on the raster plotter. For example to have density of 3, enter
3 (CR).

h. ENTER CHANNEL NUMBER FROM 1 TO 4 OR ENTER 0 TO END SELECTION CODE INPUT EACH TIME CH NR = IS TYPED

The first of the series of channel/data query-response message sequence permitting the input of data selection keys and corresponding output channel assignment codes. This is a perparatory message informing the operator how to respond. No operator response is required at this time.

i. CH NR =

At this time enter the channel number for which the feature associated with the selection codes are to be output. Entries are as follows:

1 (CR) channel 1 designated or output
.
.
.
4 (CR) channel 4 designated as output
0 (CR) Indicates the end of the channel/data query response cycle

j. ENTER CODES (WYZNNNNNMMMMM)

Enter list of feature extraction keys or codes in the format indicated

W Y Z N N N N N M M M M M

The WYZNM fields have the following meanings:

W - Process control field where:

A = All else data extraction Key
E = Data extraction exception indicator
N = Normal extraction code entry designator
X = End of extraction code list indicator

Y - Range definition code, where:

S = Denotes a single or 1:1 input code to output channel definition mapping.

R = Denotes that a set of input codes between the range specified by the NNNNN and MMMMM field inclusive, are to be mapped to the given output channel.

Z - Output density designator field

Integer value ranging from 0-7

NNNNN - Input data extraction Key (or code)

Five digit decimal number, right adjusted with leading zeroes as needed.

MMMMM - Terminal range extraction Key

Five-digit decimal number, right adjusted with leading zerors as needed

If field Y is an "S", then enter

00001

(1) For a normal single selection code of density 7 with the selection code of 39 the following entry would be made

NS70003900001 (CR)

(2) If all remaining input data is desired to be output at one specific line weight and density, the All Else data extraction key of the process control field would be entered. This eleviates the need to enter all the selection codes. To initiate this function the following entry would be made

AS700002000o1 (CR)

This response states that for the channel indicated in response to 6i all input data unless otherwise specified will be output with a density of 7.

(3) If, in conjunction with the All Else function, a specific feature is not to be output at all, the data extraction exception indicator of the process control field would be used. An example of this function is illustrated by the following entry.

ES60000500001 (CR)

This response says that the feature whose selection code is 5_{10} is not to be output.

(4) To terminate the selection code enteries for a particular channel, the following entry would be made:

X ␣ (CR) where ␣ = space

k. OK

Response from the system to the operator when the selection code entered has been accepted. Another selection code can then be entered after the OK response has been encountered.

l. ERRONEOUS SELECTION CODE ENTRY - WILL NOT BE PUT IN TABLE ENTRY AS RECEIVED IS

W Y Z NNNNN MMMMM

AT PAUSE 1 CONTINUE OR MANUALLY TERMINATE
PAUSE ØØØØØ1

A message to the operator indicating that an error existed in the last entered selection code. The operator wishes to input selection codes then he enters.

↑P (CNTRL P)

At this time the operator will then be able to re-enter the selection code. If the operator wishes to terminate execution he should enter

↑C (CNTRL C)

and control will return to the MONITOR system.

m. SELECTION CODE TABLE FULL - LAST ENTRY

W Y Z NNNNN MMMMM

A message to the operator indicating the finish of the channel/data query response cycle due to an excess of selection codes. Processing will continue as if the channel/data query response cycle terminated normally. No operator response is required at this time.

n. SELECTION CODE SUMMARY ON PRINTER. ARE CORRECTIONS REQUIRED ENTER Y OR N

This message appears at the end of the channel/data query response cycle. That is, when the response to message 6i is zero (0). The summary on the on-line printer is in numeric order according to selection codes and shows the density and

channel on which the feature will be output. (Channels here are indicated as 0, 1, 2, 3). If there are errors, a response of

Y (CR)

will restart the channel/density query response cycle. If the table is correct, enter

N (CR).

o. ARE THERE MORE INPUT TAPES FOR THIS RUN IN ADDITION TO THAT CURRENTLY MOUNTED ON DRIVE m ENTER Y OR N

Y (CR) denotes multi-reel input file
N (CR) denotes single reel input file

p. MOUNT NEXT INPUT OR DRIVE i

This output message is printed if, and only if, the response to the previous query message (6.0) is affirmative (i.e., Y). No operator response is required at the completion of this task.

q. ENTER D IF PROPER INPUT TAPE MOUNTED ON DRIVE m

The above message is generated irrespective of the response to message 6.0. Its purpose is to provide an additional check point for input tape assignment validation.

Respond with the following input message for acknowledgement of input file assignment:

D (CR)

r. Entry of the above input message initiates the input data file(s) format conversion process. No further messages will appear until either the input data file is exhausted or the output is filled. In the latter case the following system generated message will appear.

s. AT PAUSE 2, MOUNT NEW SCRATCH ON DRIVE r AND CONTINUE
PAUSE 000002

When the above messages appear, remove the tape from the unit specified by numeric, r, and replace it with a full reel of magnetic tape. To continue, enter the following input message:

↑P (CNTRL P) - causes processing to continue from the point of interruption utilizing a new output tape file.

t. FIRST Y COORDINATE IS yyyyy

Message to the operator indicating the Y address of the first scan line encountered so that the graphic plotter carriage address may be set at this value. This number is in decimal. No operator response is required at the completion of this message.

u. BAD WRITE ATA IWRITE = wwwww

Message to the operator indicating the probability of erroneous, data being written on the output tape . The scan line where this occurs is in the vicinity of the first Y coordinate plus the value of IWRITE. This number is in decimal. No operator response is required at the completion of this message.

v. BAD READ AT IREAD = rrrrr

Message to the operator indicating the probability of erroneous data being read from the input data tape. The scan line where this occurs in in the vicinity of the first Y coordinate plus the value of IREAD. This number is in decimal. No operator response is required at the completion of this message.

w. TOTAL READS = rrrrr
TOTAL WRITES = wwwww + 1

Message to the operator indicating the number of blocks of data read from the input tape and written onto the output tape. These numbers are in decimal. No operator response is required at the completion of this message.

7. POST-PROCESSING PRINTOUTS AND RUN TERMINATION PROCEDURES

Upon completing an intput data file processing cycle, a tabulation of run and I/O statistical data will be output on the PDP-9 line printer. These include elapsed run time, the number of input records processed, and the number output for each channel. At the conclusion of this file the following query-response dialogue is initiated.

a. DO YOU WISH TO TERMINATE RUN (Y OR N)

Y (CR) - Denotes unconditional termination of program. Control is returned to MONITOR system.

N (CR) - Denotes a request for an additional processing cycle.

b. DO YOU WISH TO STACK OUTPUT FILES (Y OR N)

The above message is output at the console ASR35 if, and only if, the

```
MONITOR V4E

$G 6
*S
MOUNT NEW SCRATCH ON DRIVE 5   ENTER   D   WHEN DONE

D
ENTER MONTH,DAY,YEAR AS 6 NUMERICS ... MMDDYY
072071
ENTER 5 DIGIT TAPE ID NUMBER
99999
MOUNT 1ST REEL OF INPUT TAPE ON DRIVE 6
ENTER 6 DIGIT MINIMUM AREA IN PROPER RESOLUTION.

000003

DO YOU WISH TO SUPPLY SELECTION CODES EXTERNALLY
ENTER Y OR N
Y
ENTER DEFAULT DENSITY AS NUMERIC FROM 0 TO 7

6

ENTER CHANNEL NUMBER FROM 1 TO 4 OR ENTER 0 TO END
SELECTION CODE INPUT EACH TIME   CH NR =   IS TYPED
CH NR =

4

ENTER CODES  (WYZNNNNNMMMMM)

NS50004700001

OK

X

CH NR =

0

SELECTION CODE SUMMARY ON PRINTER. ARE CORRECTIONS REQUIRED
ENTER Y OR N.

N
```

FIGURE III-3. RPSS PROGRAM RUN:MANUAL MODE

```
ARE THERE MORE INPUT TAPES FOR THIS RUN IN ADDITION TO THAT
CURRENTLY MOUNTED ON DRIVE 6   ENTER Y OR N

N

ENTER D IF PROPER INPUT TAPE IS MOUNTED ON DRIVE 6

D

FIRST Y COORDINATE IS    212
TOTAL READS =   330
TOTAL WRITES =   328 + 1
DO YOU WISH TO TERMINATE RUN   (Y  OR  N)

Y

REMOVE OUTPUT TAPE ON DRIVE 5
LABEL IT  DATE     7/20/71 TAPE NUMBER  099999 REEL NUMBER   0
END OF RUN - REMOVE ALL TAPES FROM DRIVES


                    . * TERMINATION ***

STOP  000000
```

FIGURE III-3. RPSS PROGRAM RUN:MANUAL MODE (CONCLUDED)

response to message 7a is a N. The response to this message are as follows:

Y (CR) - Indicates the continued use of the current output tape file for the next processing cycle.

N (CR) - Denotes that a new output tape file is desired.

c. REMOVE OUTPUT TAPE ON DRIVE n
LABEL IT DATE mm/dd/yy, TAPE NUMBER nnnnn, REEL NUMBER rr

This message appears if the operator response to the previous query (7b) is negative, i.e., an or if response to 7a is affirmative i.e., a Y. Operator response to this message is not required.

d. DO YOU WISH SAME SELECTION CODES AS PREVIOUS RUN (Y OR N)

If additional processing cycle is requested (i.e., response to 7a is N), this message appears to determine if the selection codes for the new input tape are the same as the previous. If they are, then the channel/data query response cycle is avoided, in this case the response should be

Y (CR).

If there are any changes in the selection codes, the channel/data query response cycle must be restarted and in this case, the response should be

N (CR).

This latter response can also be used to change mode of operation to Tape Mode, if so desired.

e. END OF RUN - REMOVE ALL TAPES FROM DRIVES

TERMINATION

This is the final message output by the system prior to termination and transfer of control to the resident MONITOR system.

8. PROCESSING STATISTICAL DATA OUTPUTS

Immediately following the successful completion of the generation of a drive tape for the graphic plotter, timing and data statistics are presented on-line via the line printer. These include the number of scan lines input and output, number of data points processed, the number of points of various ACSD designator codes and the number of points output per channel.

APPENDIX IV

RASTER PLOTTER SOFTWARE SYSTEM PROGRAM DESCRIPTION

APPENDIX IV

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT)

APPENDIX IV
INDEX OF DIAGRAMS

SECTION I

RASTER PLOTTER SOFTWARE SYSTEM PROGRAM

1. GENERAL

The RPSS Program described in this appendix consists of fifteen FORTRAN language subprograms and four MACRO assembler language routines which are functionally unique.

Section II provides a detailed description of the nineteen unique subprograms of the RPSS Program. Subsection 1 contains the program descriptions of both the FORTRAN and MACRO subprograms. Subsection 2 identifies the existing PDP-9 Object-Time System programs and Science Library MACRO subroutines called by the RPSS Program for various FORTRAN IV input/output functions. Section III contains the flow diagrams for the required RPSS assembly language subprograms. Section IV contains the autoflow diagrams for the required RPSS FORTRAN subprograms.

2. SYSTEM DESCRIPTION

The RPSS Program is a self-contained, stand alone program designed to be executed on a PDP-9 processing system having the following configuration:

Quantity	Equipment Item Description
1	DEC PDP-9 Control Processing Unit (CPU)
8	4096 18-bit Word Core Memory Modules
1	10 CPS, Console Keyboard/Teleprinter
3	7-Track, 45IPS, 556/800 BPI IBM Compatible Magnetic Tape Units W/Controller
4	DECtape Units W/Controller
1	DEC PDP-9 Paper Tape Reader
1	DEC PDP-9 Paper Tape Punch
1	1 Million 18-Bit Word Disk File W/Controller
1	ANELEX, 1000 LPM Line Printer

a. Object Program Deck Arrangement

The RPSS Program is assembled and compiled as a "relocatable" program. With the exception of the control program (RPSS), which by definition must be the first program in the object deck, the remaining object programs may be arranged in any order.

b. System Diagrams

Figure IV-1 shows the control logic flow for the RPSS program. PDP-9 OTS and Science Library subprograms and routines are not shown since they are loaded and enter the procedureal flow only at the time of execution.

Figure IV-2 shows the overall system block diagram of the RPSS program, while Figure IV-3 and IV-4 show the Tape and Manual mode block diagrams respectively.

c. Logic Diagram Symbology

The syllabus of logic diagram symbology, as shown in Figure IV-5, defines the program flow chart notation conventions employed in this appendix.

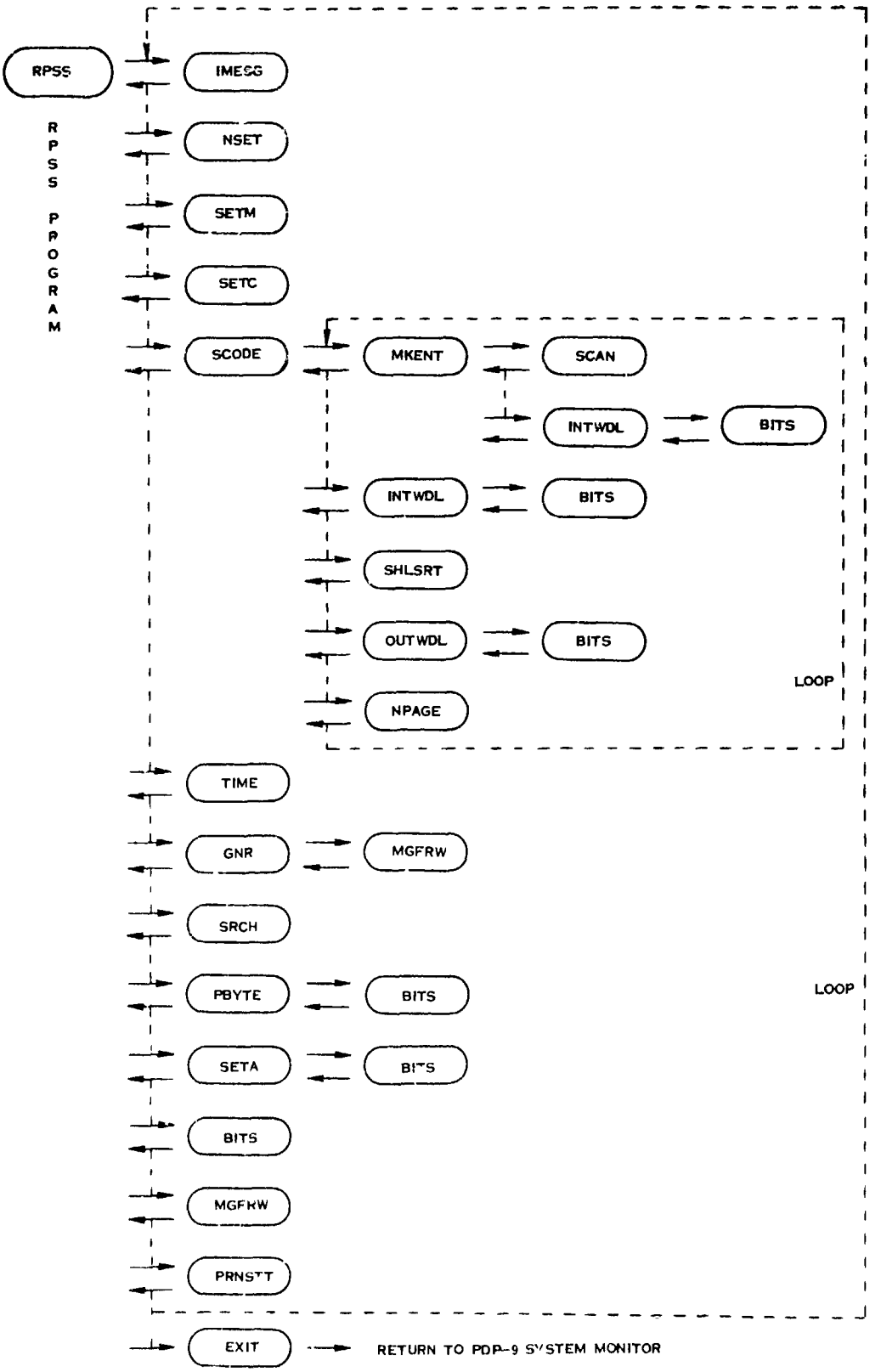


FIGURE IV-1. SYSTEM LOGIC FLOW DIAGRAM

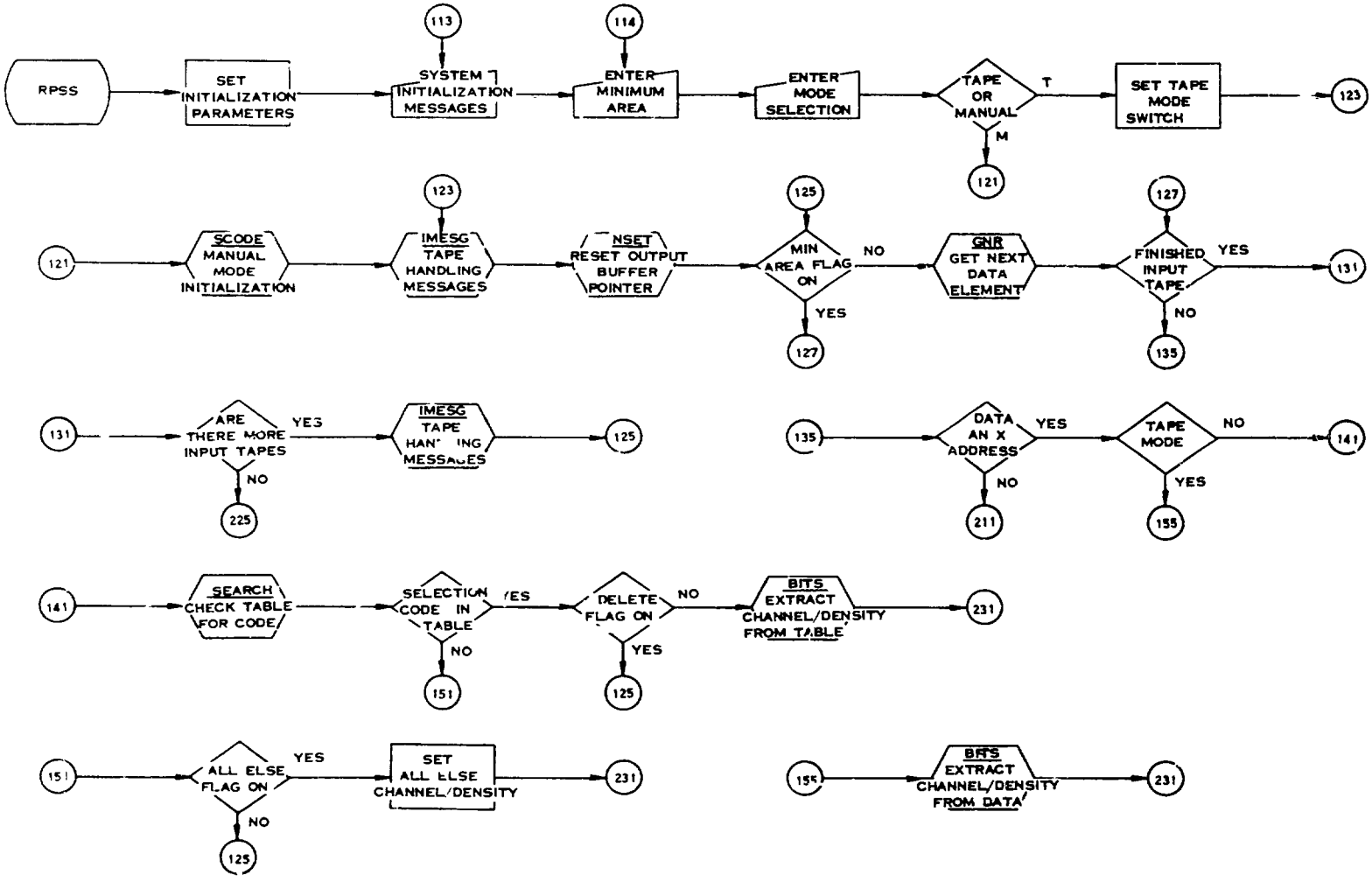


FIGURE IV - 2. RASTER PLOTTER SOFTWARE SYSTEM BLOCK DIAGRAM

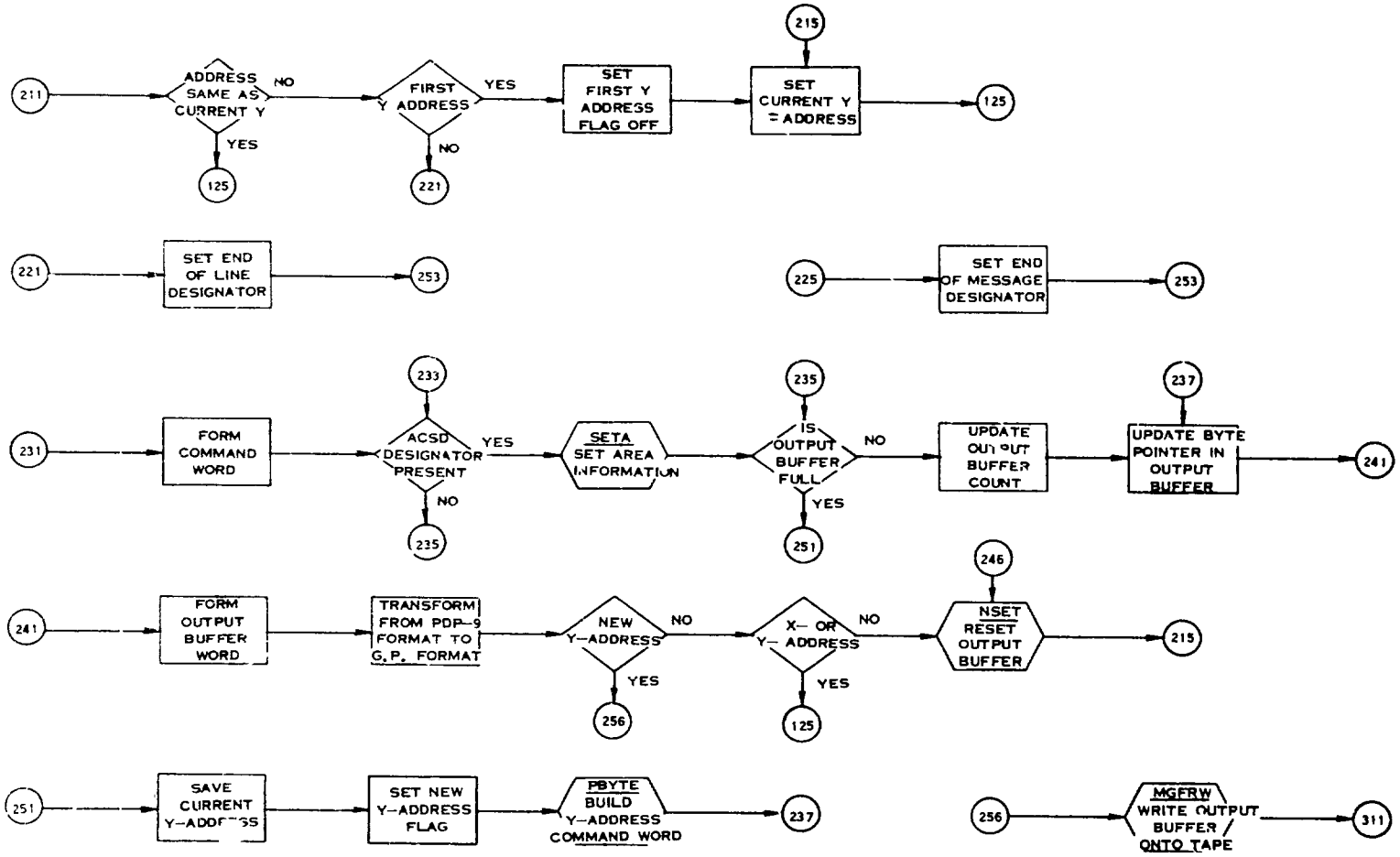


FIGURE IV - 2. RASTER PLOTTER SOFTWARE SYSTEM BLOCK DIAGRAM (CONTINUED)

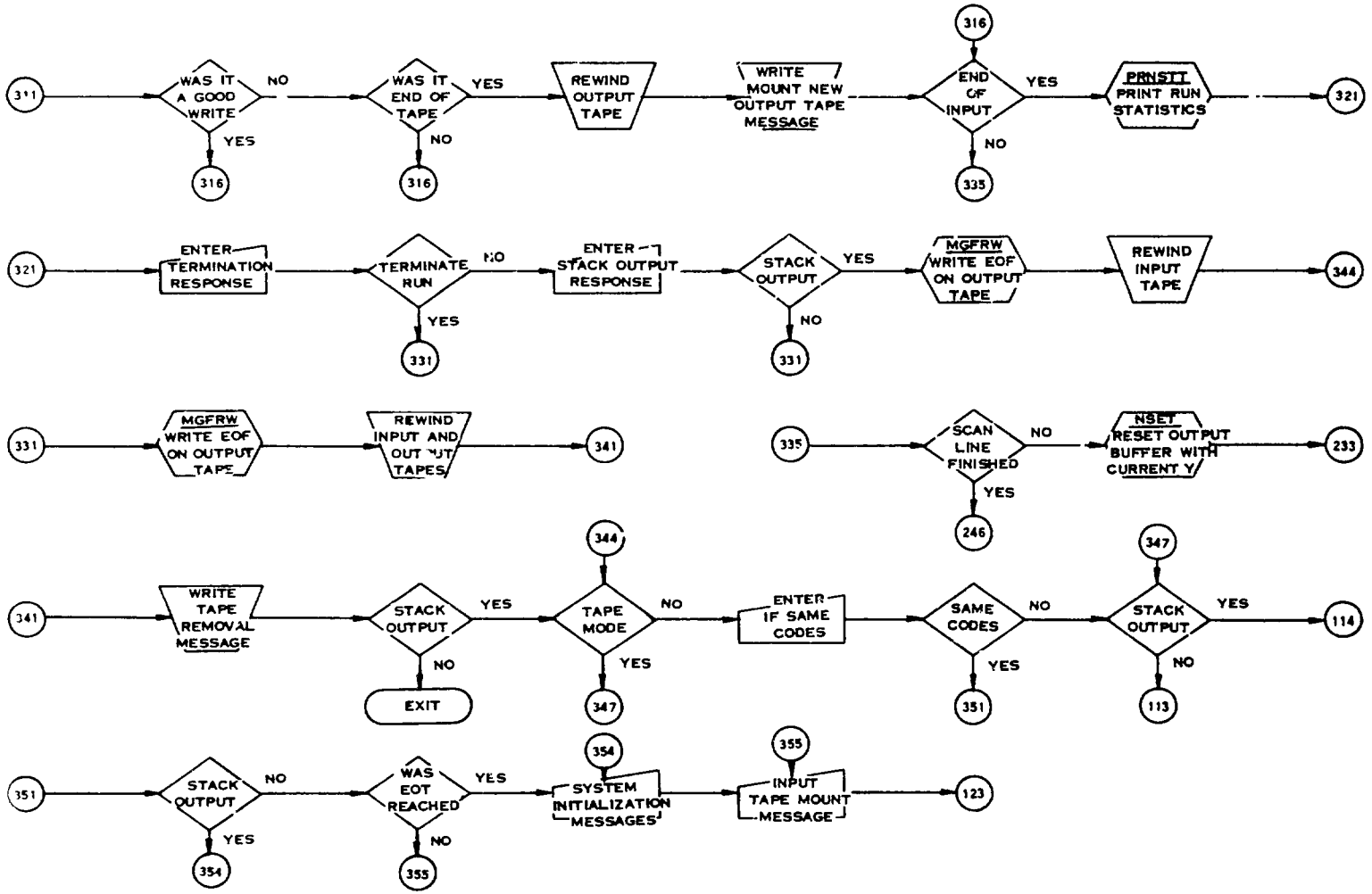


FIGURE IV - 2. RASTER PLOTTER SOFTWARE SYSTEM BLOCK DIAGRAM (CONCLUDED)

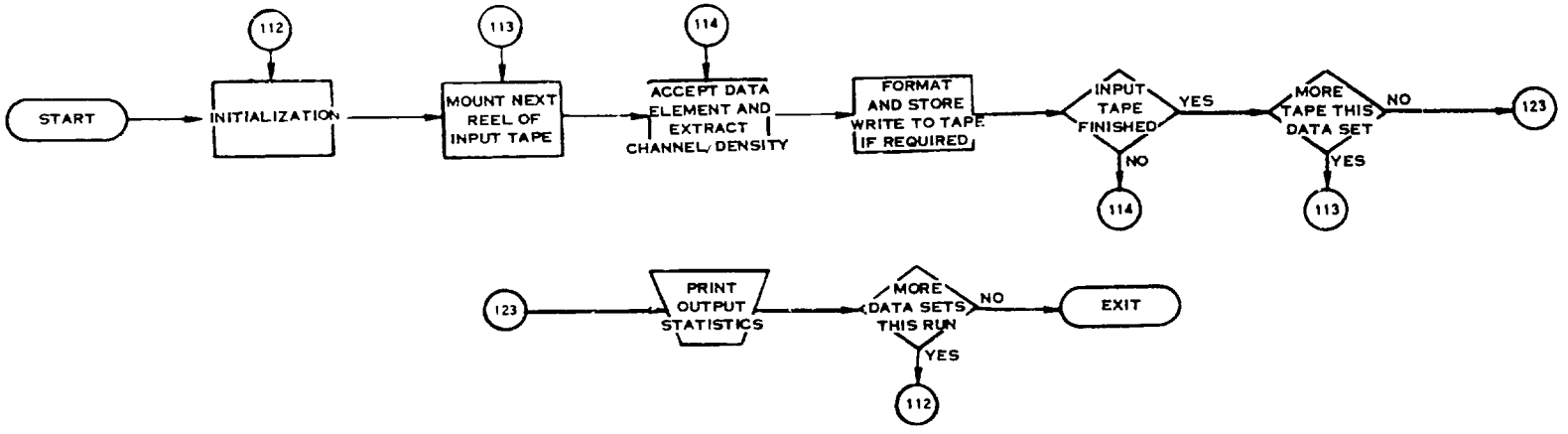


FIGURE IV - 3. RPSS TAPE MODE BLOCK DIAGRAM

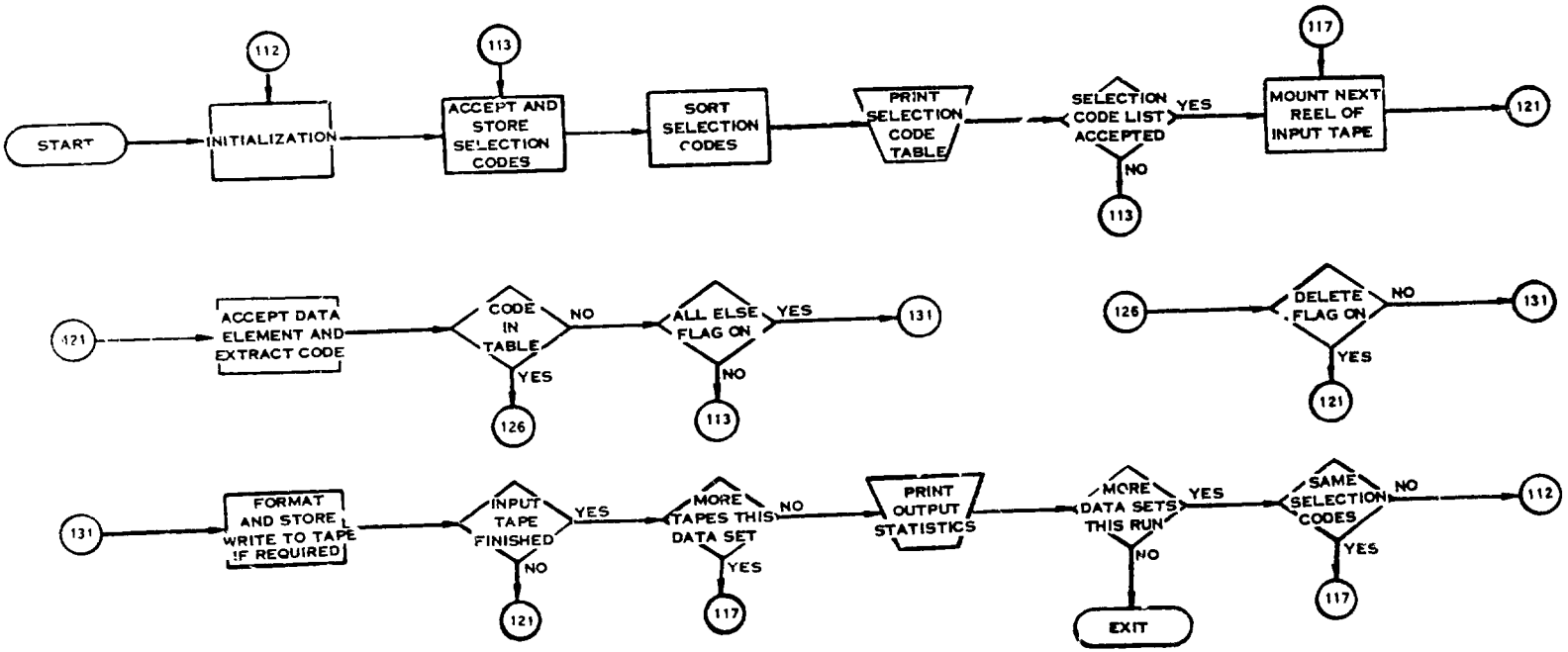


FIGURE IV - 4. RPSS MANUAL MODE BLOCK DIAGRAM

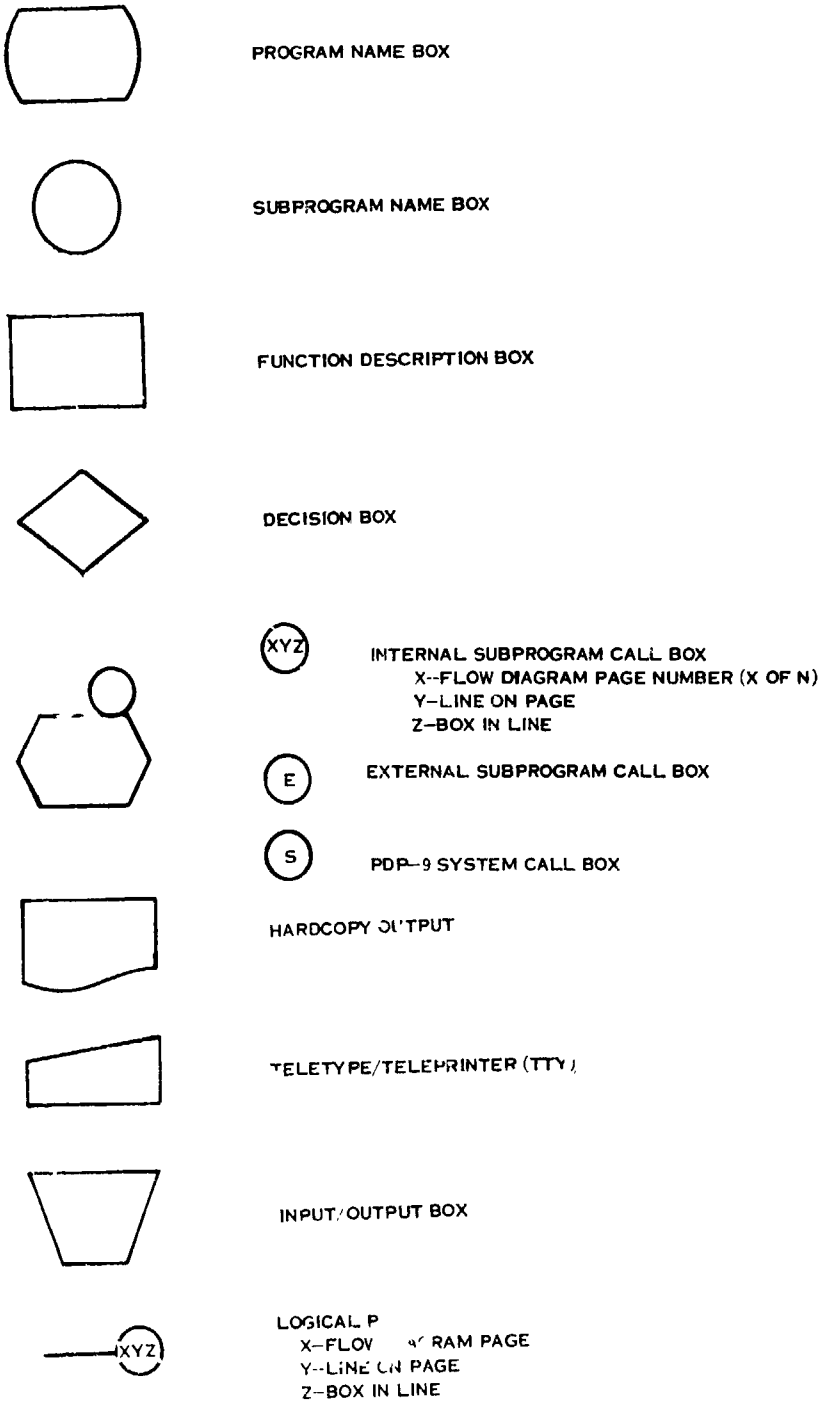


FIGURE IV-5. LOGIC DIAGRAM SYMBOLOGY

SECTION II

DESCRIPTION OF SUBPROGRAMS

1. WORKER AND UTILITY PROGRAMS

a. RPSS (Raster Plotter Software System Control Program)

(1) Abstract

RPSS is the main control program of the Raster Plotter Software System. It oversees all the supervisory functions required to handle the output format and determines whether the tape format or the selection code assignment format is to be utilized. The rogram then reads raster scan lines from the input tape by calling GNR, locates and extacts the proper selection code using the selection code table (SRCH) and reformats and places it into the output buffer area. The program then calls MGFRW to write the record on the output tape. Upon completion, statistical information is output by PRNSTT via the on-line printer and control is returned to the PDP-9 monitor system.

(2) Program Description

RPSS, during system initialization, receives via the on-line teletype all the pertinent information necessary to process the selected raster format input tape. Formatting and selection for output can proceed in one of two paths. The first path is that of data transfer and reorganization for the final output tape. In this mode, referred to as Tape Mode, it is necessary for the user to supply a minimum area for future recognition of this parameter in the data set. The input information is then read from the input tape by a call to GNR. If no EOF or EOT is encountered, the data point is typed, i. e., whether it is an X - or Y - coordinate and is then placed, after formatting, in an output buffer, located in core, for future transfer to tape. The coordinate is stored with the associated line center designator and/or area start/stop, channel number, and density in proper plotter format. For the minimum area case an additional area stop coordinate is generated. The buffer is written to the output tape when one of two conditions occur. The first when the buffer is filled and the second is when a new Y - coordinate, is encountered before the buffer is filled. Tape reading and writing is handled by calls to MGFRW. The second mode of operation, referred to as the Manual Mode, allows the user to select which features on the input tape he would like to have on the output tape. Each feature has a unique selection code for ease in recognition. SCODE handles all the necessary procedures to obtain the required information for selection code, channel number and density. The procedure for data handling is the same then as for the tape mode with the addition that each coordinate has its "selection code" compared to the selection code table (by SRCH) to ensure that it is an extracted feature. If it is not a desired feature, it is not placed in the output buffer; if it is a desired feature then all pertinent information for

proper plotter operation is placed in the buffer. Processing then continues as previously discussed. Statistical information is compiled throughout execution and is output via the on-line printer at the end of each run upon completion of successful execution.

(3) Inputs

(a) Calling Sequence -

None - stand alone program.

(b) Keyboard Entries - Operator Input Responses -

D	
mmddyy	(month, day, year)
iiiiii	(tape ID)
Y	
N	
aaaaaa	(minimum area)
CNTL P	(for continuation)

(c) Magnetic Tape -

Raster formatted data tape containing all necessary information in proper format.

(4) Outputs

(a) Teleprinter -

Program generated query and administrative messages

"MOUNT NEW SCRATCH ON DRIVE ENTER D WHEN DONE"

"ENTER MONTH, DAY, YEAR AS 6 NUMERICS . . .MMDDYY"

"ENTER 5 DIGIT, TAPE ID NUMBER"

"MOUNT 1ST REEL OF INPUT ON DRIVE _____"

"DO YOU WISH TO SUPPLY SELECTION CODES EXTERN-
ALLY ENTER Y OR N"

"ENTER 6 DIGIT MINIMUM AREA IN PROPER RESOLUTION"

"AT PAUSE 2, MOUNT NEW SCRATCH ON DRIVE_____
AND CONTINUE"

"DO YOU WISH TO TERMINATE RUN? (Y OR N)"

"REMOVE OUTPUT TAPE ON DRIVE_____LABEL IT:
DATE;_____/_____/_____, TAPE NUMBER;_____
REEL NUMBER_____"

"DO YOU WISH TO USE SAME SELECTION CODES AS
PREVIOUS RUN (Y OR N)"

"END OF RUN - REMOVE ALL TAPES FROM DRIVES"

*** TERMINATION ***

"BAD WRITE AT IWRITE = _____"

"TOTAL WRITES = _____ + 1"

(b) Magnetic Tape -

Raster formatted tape for use on the Graphic Plotter.

(5) Referenced Subprograms

SCODE
GNR
SRCH
PBYTE
IMESG
PRNSTT
NSET
SETA
SETM
SETC
MGFRW

Bit Routines: LS
LO
LEOR
LA
RS

b. SCODE (Selection Code)

(1) Abstract

SCODE is that part of the Raster Plotter Software System that handles the selection code processing before reading the input data tape. This subroutine accepts the selection codes, densities, and channel numbers that will be used as the selection criteria for the final output product. SCODE performs all the preliminary handling of the selection code information. In addition, SCODE sets the ALL ELSE channel if and when it is selected.

(2) Program Description

SCODE reads in from the TTY Keyboard the necessary information to form the Selection Code Table (ISC). The first thing read in is the default density. The table is then preset with the default density being placed in each entry of the tables. Right after this is done the ALL ELSE functions are cleared and the ALL ELSE channel is set to channel 5. The selection code is read in on a per channel basis. Then the desired channel number is read. The procedure of obtaining the selection codes is continued until either the user signifies end of selection code assignment or until the selection code table has been completely filled. Entries to the table are made by MKENT. After the last entry has been made the Selection Code Table is sorted (SHLSRT) and the entire table in sorted order is printed via the on-line printer so that the user will have a hard copy of the selection code, channel, density, and delete status. The ALL ELSE information is also output at this time. The user is then queried to determine whether or not the list is acceptable. If not acceptable the selection code assignment procedure is repeated. If accepted, control returns to the main program to continue processing.

(3) Inputs

(a) Calling Sequence - CALL SCODE

Arguments: None

(b) Teletype Keyboard -

d	(default density)
n	(channel number)
A	
R	
S	
N	
E	
X	
Y	
N	

(4) Outputs

(a) Teleprinter -

Program generated query and administrative messages.

"ENTER DEFAULT DENSITY AS NUMERIC FROM 0 TO 7"

"ENTER CHANNEL NUMBER FROM 1 TO 4 OR ENTER 0 TO END SELECTION CODE INPUT EACH TIME CH NR= IS TYPED"

"CH NR = "

"ENTER CODES (WYZNNNNNMMMMM)"

"SELECTION CODE SUMMARY ON PRINTER. ARE CORRECTIONS REQUIRED ENTER Y OR N."

(b) On-Line Printer -

Information printed in addition to that printed by subroutine NPAGE are the selection code information to appear in tabular form. The last entry at the end of the list would be

"ALL ELSE"

under the selection code heading, with the proper channel number and density appearing under the appropriate column headings. The ALL ELSE information will be printed only if ALL ELSE has been specified.

(5) Referenced Subprograms

INTWDL
MKENT
SHRSRT
NPAGE

c. MKENT (Make Table Entry)

(1) Abstract

MKENT reads the input selection code and enters it into the selection code table (lSC) for the selection code mode of operation. The status of the selection code table is checked before and after each entry. If there is an entry for a particular selection code already in the table, the previous entry will be overwritten. A return parameter indicates the status of the table after the entry has been made.

(2) Program Description

Before entering the selection code for a specific previously designated channel into the table, a series of error checks are made. These checks include

a. a test for end of input

b. erroneous entry

c. is it the ALL ELSE channel

After these checks, MKENT determines if the delete flag is either on or off, if the default density is to be implemented or if a density has been input, and whether the selection code being input is a single value or a range of values. At this time placement into the table can proceed. Before the actual entry is made the selection code table is checked for a duplicate entry (same selection code) by a call to SCAN. If SCAN indicates a duplicate, the previous entry is overwritten. If there is no duplicate entry, the pointer is advanced to the next available spot. INTWDL is then called to place the selection code in the first word and to pack the density channel number previously designated and delete flag (if on) into the second word of the table entry. After each entry, the table is checked to see if it is full. The table is large enough to hold 1000 selection codes with associated channel number, density and delete flag. If the table is full, further inputs are ignored and processing continues without the additional inputs. If a range of selection codes was input, entries are made for each selection in the entire range until the range is exhausted. The return parameter indicates the status of the table.

(3) Inputs

(a) Calling Sequence - CALL MKENT (IFLG)

Arguments: IFLG - return parameter from MKENT telling the status of the selection code table

- 3 table is full after entry made
- 2 erroneous entry, not processed
- 1 end of input encountered
  0 All Else Channel assigned
  positive number
  the location (pointer)
  number assigned the entry in selection code table.

(b) Entries - Selection Code Information -

WYZNNNNNNMMMMM

W =	A	ALL ELSE channel assignment
	E	Except = Delete Flag Set
	N	Normal Entry
	X	End of Input
Y =	S	Single selection code
	R	Range of selection codes
Z =		Density
NNNNN =		Start of selection code or signal selection code
MMMMM =		End of selection code (if Y = R, otherwise 00001)

(4) Outputs

(a) Teleprinter -

"SELECTION CODE TABLE FULL - - LAST ENTRY ------"

"ERRONEOUS SELECTION CODE ENTRY - WILL NOT BE PUT IN TABLE
ENTRY AS RECEIVED IS:

WYZNNNNNMMMMM

AT PAUSE 1, CONTINUE OR MANUALLY TERMINATE.

(5) Referenced Subprograms

SCAN
INTWDL

d. GNR (Get Next Record)

(1) Abstract

GNR gets the next record from the input tape containing the data to be processed.

(2) Program Description

GNR issues the calls to MGTRW to read the input tape and store it in a local buffer. GNR then investigates the block of data. If an EOF or EOT is encountered, control is returned to the main program noting the condition. If no EOF or EOT is encountered, GNR then determines if the block is the first block for a given raster line, or a continuation block. GNR also determines whether the data is an X - coordinate or a Y - coordinate. Subsequent calls return the next coordinate from the buffer until it is empty.

(3) Inputs

(a) Calling Sequence - CALL GNR

Arguments: NONE

(4) Outputs

(a) Memory -

Buffer area filled with the record read from the input tape.

(b) Teletype -

"FIRST Y COORDINATE IS _____"

"BAD READ AT IREAD = _____"

"TOTAL READS = _____"

(5) Referenced Subprograms

MGFRW

e. SCAN (Scan Selection Code Table)

(1) Abstract

SCAN checks the Selection Code Table (ISC) before entries are made to insure that no duplicate entries exist. If a duplicate is found, its position is returned to the calling program.

(2) Program Description

SCAN is a loop that checks each member of the selection code table (ISC) against the new selection code to be entered (IVAL). This is performed on the unsorted table as the table is being built, to avoid duplicate entries. It is performed each and every time an entry is to be made in the selection code table. If a duplicate is found, a pointer to that duplicate entry is returned to the calling program in the parameter K. If no duplicate is found, that is, if the selection code has not been previously entered in the table, a return code of -1 is passed in the parameter K.

(3) Inputs

(a) Calling Sequence - CALL SCAN (IVAL, K)

Arguments:

IVAL = Selection Code for which the table is to scanned to determine if it is a duplicate entry

K = Return argument telling status of IVAL in the selection code table where:

-1 IVAL did not previously appear in the table

$\neq$ -1 IVAL did previously appear in the table and K contains its location with the table

(4) Outputs

None

(5) Referenced Subprograms

None

f. SHLSRT (Shell Sort)

(1) Abstract

SHLSRT performs a Shell sort on the entries of the Selection Code Table (ISC). The table is ordered in ascending selection code with each two word entry of the table being moved as necessary.

(2) Program Description

SHLSRT is an algebraic comparison of the entries in the selection code table (ISC) made on the first word of the table entry. This routine is called after the last entry has been made. The table is ordered by ascending selection code with the second word of the entry being switched when necessary, to remain with its proper selection code.

(3) Inputs

(a) Calling sequence - CALL SHLSRT

Arguments: None

(4) Outputs

None

(5) Referenced Subprograms

None

g. SRCH (Search The Table)

(1) Abstract

SRCH performs a binary search of the sorted Selection Code Table to determine if a particular selection code is in the table. The return argument

tells the status of the selection code.

(2) Program Description

SRCH is called after the input data tape is read to determine if a particular selection code is in the Selection Code Table (ISC). A binary search is incorporated to optimize the search time as the table lookup has to be done for each coordinate from the input tape. If the selection code is found a return argument indicating the location in the table is returned. If the selection code is not found, i. e., not in the table, a value of -1 is returned.

(3) Inputs

(a) Calling Sequence - CALL SRCH (IARG, K)

Arguments:

IARG = The selection code the table is being searched for.

K = Return argument. If it is -1, the selection code was not found. If a positive integer then it is a pointed to the position in the Selection Code Table to where the selection code is.

(4) Outputs

None

(5) Referenced Subprograms

None

h. INTWDL (Input - Table Word Logic)

(1) Abstract

INTWDL makes the entry in the Selection Code Table (ISC). The first word of the entry is the selection code and the second word contains the density, channel number and delete flag, packed in proper format for output on the graphic plotter.

(2) Program Description

INTWDL utilizes the bit manipulation routines in order to pack the second word of the Selection Code Table (ISC) entry. The parameters to be put into the table are paired in the labelled common B1 and the location is passed in the argument K. The selection code is immediately placed into the first word as a integer. The form of the second word is as follows:

bits	0	-	5	are zero
	0	-	8	contain the density
	9			contains a 1
	10	-	11	contains the channel number
	12	-	16	are zero
	17			is the delete flag

The second byte (bits 6 - 11) of the second word contains the density and channel number in the proper format for output on the graphic plotter. Before the entry is made, the location in the selection code table is cleared to zero to avoid any extraneous information for the bit routines.

(3) Inputs

(a) Calling Sequence- CALL INTWDL (K)

Arguments:

K = Selection Code Table location for which the entry is to be made.

(4) Outputs

(a) Memory -

Selection code table entry containing selection code, density, channel number and delete flag.

(5) Referenced Subprograms

Bit Routines: LS
LO
LOR

i. OUTWDL (Output - Table Word Logic)

(1) Abstract

OUTWDL reads the entry from the selection code table and decodes the second word of the entry into the density, channel number and delete flag.

(2) Program Description

The first word of the Selection Code Table (ISC) entry contains only the selection code. The second word in packed form contains the density, channel number and delete flag (see INTWDL). The bit manipulation routines are used to unpack the second word and extract the desired information. The unpacked parameters are then passed by the labeled common H1.

(3) Inputs

(a) Calling Sequence - CALL OUTWDL (K)

Argument:

K = Selection Code Table location for which the entry is to be extracted.

(4) Outputs

(a) Memory -

The selection code, density, channel number and delete flag stored in labeled common H1.

(5) Reference Subprograms

Bit Routines:

LA
RS

j. PBYTE (Byte Placement)

(1) Abstract

PBYTE is a byte manipulation routine that shifts bytes either left or right in a PDP-9, 18-bit (3-byte) word.

(2) Program Description

PBYTE shifts bytes either left or right in the PDP-9, 18-bit, word. The byte location, IB, of the input word, IW, is shift to byte location, IOB, of the output word, IOW. The bit manipulation routines are used to perform the necessary shifts. If IB is greater than IOB, a shift left is performed; if IB is less than IOB, a shift right is performed. The amount shifted, in bytes, is determined by the difference between IB and IOB.

(3) Inputs

(a) Calling Sequence - CALL PBYTE (IW, IB, IOW, IOB)

Arguments:

IW = Input word whose byte is to be shifted

IB = Byte location in input word of byte to be shifted. 1 is most significant byte, 2 is middle byte, and 3 is least significant byte.

IOW= Output word into which the shifted byte from IW is to be placed.

IOB= Byte location output word of where shifted byte from IW is to be placed in IOW. 1 is most significant byte, 2 is middle byte, and 3 is least significant byte.

NOTE: IW and IOW may be the same word in the calling program.

(4) Outputs

(a) Memory -

A PDP-9, 18 bit, word with byte shifted.

(5) Referenced Subprograms

Bit Routines: LA
LO
LS
RS

k IMESG (Messages)

(1) Abstract

IMESG is the routine that handles the input tape messages to determine if additional processing is desired.

(2) Program Description

IMESG queries the user to determine if there are any further input tapes required for the present run. If there are, it issues the proper tape mount message and insures that the proper tape is mounted for processing. IMESG also switches the magnetic tape drives unit numbers which are used for the input so that continual processing may take place. IMESG also sets the done flag if no further inputs are required.

(3) Inputs

(a) Calling Sequence - CALL IMESG

Arguments: None

(b) Keyboard entries -

Y
N
D

(4) Outputs

(a) Teletype -

"ARE THERE MORE INPUT TAPES FOR THIS RUN IN ADDITION TO THAT CURRENTLY MOUNTED ON DRIVE _____ ENTER Y OR N."

"MOUNT NEXT INPUT TAPE ON DRIVE_____"

"ENTER D IF PROPER INPUT TAPE IS MOUNTED ON DRIVE_____"

(b) Memory -

Done flag, located in common J1, is either set or not set depending on response to the first query.

(5) Referenced Subprogram

None

l. NPAGE (New Page)

(1) Abstract

NPAGE writes the heading information for the Selection Code Table -- Input Summary on the line printer. Information printed is the month, day, year, page number, output tape number and file number, revision number. Also printed is the heading for the table for the selection code, channel number, density and delete flag. The title is also printed at the top of the page.

(2) Program Description

NPAGE is a series of write statements to the on-line printer. The same information appears on the top of each page that is needed for the Selection Code Table - Input Summary.

(3) Inputs

(a) Calling Sequence - CALL NPAGE

Arguments: None

(4) Outputs

(a) On-line printer -

Program generates heading information on the on-line printer
"RPS SELECTION CODE TABLE - INPUT SUMMARY --/--/-- PAGE --"
"TAPE NUMBER ----"
"FILE NUMBER --"
"REVISION --"
"SELECTION CODE CHANNEL DENSITY DELETE FLAG"

(5) Referenced Subprograms

None

m. PRNSTT (Print Run Statistics)

(1) Abstract

PRNSTT prints the statistics for a given run after execution of that run has been completed.

(2) Program Description

PRNSTT prints the statistics for a given run on the on-line printed that have been accumulated for the run. These include: data; tape number; file number; elapsed time for the run in minutes and seconds; the number of scan lines input; total number of coordinates; number of designator points processed; channel totals processed for each channel.

(3) Inputs

(a) Calling **Sequence - CALL PRNSTT**

Arguments: None

(4) Outputs

(a) On-line Printer -

"RPS RUN STATISTICS OF mm/dd/yy"

"TAPE NUMBER ---------"

"FILE NUMBER ----------"

"RUN STATISTICS"

"ELAPSED TIME = -------MINUTES AND --------SECONDS"

"---------SCAN LINES INPUT CONTAINING-----X COOR-DINATES"

"X DESIGNATOR OF POINTS PROCESSED WERE"

CODE	COUNT
. . .	. . .
. . .	. . .
. . .	. . .

"CHANNEL TOTALS: CHANNELØ =--------. CHANNEL 1 = ----.

"CHANNEL 2 = --------. CHANNEL 3 = ---.-------".

(5) Referenced Subprograms

None

n. NSET (Set Buffer)

(1) Abstract

NSET resets the pointers to the output buffer. It also clears the buffer and presets the buffer to the high ignore value.

(2) Program Description

NSET resets the next entry counter to the beginning of the output buffer IOBUF. The LSTBYT and LSTWRD parameters are initialized to the original starting state. At the same time the output buffer is preset to the high order ignore value. This is contained in four consecutive PDP-9 words which gives us three consecutive Graphic Plotter words (24 bits each). These words have a high order ignore value, less than the EOL, such that when encountered, the plotter will go over the maximum scan position and thus not plot. This value is preset as a line centere coordinate to insure a turning off of any areal data.

(3) Inputs

(a) Calling Sequence - CALL NSET

Arguments: None

(4) Outputs

(a) Memory -

A present buffer to be used as the output buffer storage for Graphic Plotter formatted data.

(5) Reference Subprograms

None

o. SETA (Set Area)

(1) Abstract

SETA is used to set the area stop coordinate when a minimum area designator is obtained from the input data.

(2) Program Description

SETA operates on the coordinate in different manners depending upon whether the coordinate is line center, area start or minimum area data. For line center data, no processing is done. The minimum flag is set to zero and control is returned to the main program. For area start, the line center indicator is removed from the coordinate control word to signify area start. The minimum flag is set to zero and again control is returned to the main program. For minimum area, the minimum area designator is masked out of the coordinate control word to simulate an area start. A line center coordinate is added to end minimum area, and the line center indicator is removed from the coordinate control word of the first word to signify area start. That is, minimum area has been converted to an area start and an area stop. Minimum area flag is set equal to one.

(3) Inputs

(a) Calling Sequence - CALL SETA

Arguments: None

(4) Outputs

(a) Memory -

An area start and area stop coordinates from minimum area settings when applicable.

(5) Referenced Subprograms

Bit Routines: LEOR
LA

p. SETM (Set Masks)

(1) Abstract

SETM is an assembler language routine that sets the masks used by the RPS system to obtain the desired byte of a given data word.

(2) Program Description

SETM sets the labelled common MASK which contains MASK1 and MASK2, Into these locations it sequentially places the octal constants 77ØØØØ, ØØ77ØØ, ØØØØ77, ØØ7777, 77ØØ77, 7777ØØ. These then are used as masks to extract desired information, a byte at a time, from the data words used by the RPS system.

(3) Inputs

(a) Calling Sequence - CALL SETM (MASK)

Arguments:

MASK = the first word (starting address) if the 6 word field for which the masks will be formed.

(4) Outputs

Memory -

Preset masks.

(5) Referenced Subprograms

None

q. SETC (Set Constants)

(1) Abstract

SETC is an assembler language routine that sets the masks necessary for the output buffer area before the tape write.

(2) Program Description

SETC places the coordinate corresponding to the high order "ignore" value for the raster plotter. This value is one less than the EOL character. The masks are placed in the array INSET, used by RPSS to preset the output buffer area. The mask is four words long of the form Ø437ØØ, ØØØ437, ØØ37ØØ, 37ØØØØ.

The masks are four, 18-bit words which form three consecutive and identical 24-bit words that is Ø437ØØØØ. This is the desired mask for the raster plotter.

(3) Inputs

(a) Calling Sequence - CALL SETC (INSET (1))

Arguments:

INSET = first word (starting address) of the 4 word field in which the mask will be formed.

(4) Outputs

Memory -

Preset mask.

(5) Referenced Subprograms

None

r. BITS (Bit Routines)

(1) Abstract

BITS is an assembler language subroutine with five entries that allow the assembly language logic and shifting functions to be Fortran Callable. The logic functions incorporated in the routine are:

Logical AND (LA)

Logical OR (LO)

Logical Exclusive OR (LEOR)

Left Shift (LS)

Right Shift (RS)

(2) Program Description

These functions are used by the RPS system routines that require the manipulation of bits on an integer word. For the logical functions, the routines perform the necessary operation on the first two words and place the result in the third word of the calling sequence. For the shift functions, the first word is shifted by the amount indicated in the second word and stored in the third. Operations are on a bit basis.

(3) Inputs

(a) Calling Sequence -

Logical AND: CALL LA (IA, IB, IC)

Arguments:

IA = Integer word to be AND'ed with IB

IB = Integer word to be AND'ed with IA

IC = Integer word containing result

i. e., IA .AND. IB →IC

Logical OR: CALL LO (IA, IB, IC)

Arguments:

IA = Integer word to be ORed with IB

IB = Integer word to be ORed with IA

IC = Integer word containing result

i. e., IA .OR. IB →IC

Logical Exclusive OR CALL LEOR (IA, IB, IC)

Arguments:

IA = Integer word to be EORed with IB

IB = Integer word to be EORed with IA

IC = Integer word containing result

i. e. IA .EOR. IB→IC

Left Shift: CALL LS (IA, IB, IC)

Arguments:

IA = Integer word to be shifted left.

IB = Integer word containing the number of bits to be shifted.

IC = Integer word containing result

i. e. Shift IA left IB bits and store in IC

Right Shift: CALL RS (IA, IB, IC)

Arguments:

IA = Integer word to be shifted right

IB = Integer word containing the number of bits to be shifted

IC = Integer word containing result

i. e. Shift IA right IB bits and store in IC

(4) Outputs

Memory word that is result of logical operation.

(5) Referenced Subprogram

None

s. MGFRW (Magnetic Tape Read/Write)

(1) Abstract

MGFRW is an input/output Fortran callable, assembly language utility program used to read and write blocks of data to and from standard magnetic tape.

(2) Program Description

Subprogram MGFRW is utilized by the RPS system to transfer data between core memory and standard magnetic tape. The tape handling functions performed are: Read Tape, Write Tape, Rewind Tape, and Write End-of-File (WEOF). A calling program specifies which of these functions is to be implemented by a numeric code in the calling sequence. Upon entry into MGFRW, this code and the remaining parameters needed to initiate the indicated tape operation are extracted from the calling sequence. The required instruction sequence is executed and the

program enters a "wait" loop until the I/O operation is completed. At completion of a read/write operation, MGFRW retrieves and interrogates the bit settings of the I/O status register. A unique code is stored in a predetermined location in the calling sequence to indicate the success or failure of the requested read/write operation. Control is then returned to the calling program. The calling program may proceed normally, or initiate an exception procedure, or terminate, whichever is applicable.

(3) Inputs

(a) Calling Sequence - Call MGFRW (ARG1, ARG2, ARG3, ARG4, ARG5) all arguments are integer words and:

ARG1 = machine tape number

ARG2 = number of words to be transferred to or from memory.

ARG3 = symbolic name of memory buffer into or from which the specified number of words are to be transferred.

ARG4 = function code (right adjusted); where:

0 = Read Tape 1 = Write Tape

2 = WEOF 3 = Rewind

ARG5 = completion status code word which is initially zero and is set by MGFRW as follows:

1 = EOT (End-of-Tape) 2 = EOF (End-of-File)

3 = No Error 4 = Parity Error

5 = No MTF Flag 6 = Word Length Error

7 = Error Flag

NOTE: If ARG2 (number of words) is not equal to the physical size of the tape record read, the word length error (7) code will be stored in ARG5 of the calling sequence. If ARG2 is less than the record size, N words will be read into the buffer designated by ARG3 and the tape advanced to the end-of-record gap. If ARG2 specifies a word count greater than or equal to the record size, only

N words (record length) will be read into memory.

(b) Memory -

Data blocks of size specified by ARG2.

(4) Outputs

(a) Status Register -

Contents of status register stored in ARG5 of calling sequence. The numeric values and their meaning are:

Status Value	Meaning
1	Physical end-of-tape
2	End-of-file detected
3	Normal Termination
4	Parity error
5	Improper termination
6	Wrong word length
7	Error flag

All status values are fixed point binary numbers.

(b) Memory -

Data blocks of size specified by ARG2.

(5) Referenced Subprograms

None

2. OBJECT-TIME SYSTEM AND SCIENCE LIBRARY ROUTINES

a. Object-Time System Routines

The following is a list of PDP-9 FORTRAN IV Object-Time System macro-subroutines that are loaded at execution time. The compiled FORTRAN coded conversion programs which contain I/O statements include output calls in the form of globals to various system I/O handler routines. When the compiled object program is loaded, the compiler defined global sysbols are utilized to search the FORTRAN library for the needed I/O handler routines. These are loaded into memory and the proper program control linkage established by the Link Loader. Detailed description for these programs are provided in the following DEC PDP-9 manual: "Advanced Software System Programmer's Reference Manual"; order number DEC-9A-KFZA-D.

Program Name	Function	Ref. Manual
AUXIO	To process the auxiliary I/O statements; BACK SPACE, REWIND, in FORTRAN IV program and subprograms.	P. 11-8
BCDIO	To process formatted READ/ WRITE statements in FORTRAN IV programs and subprograms.	P. 11-3
FIOPS	FIOPS provides the necessary call to IOPS (Input-Output Processing System) required by all FORTRAN I/O statements.	P. 11-10
GOTO (.GO)	To compute the index of a computed GO TO.	P. 11-14
OTSER (.ER)	To announce an error on the teletype.	P. 11-18
PAUSE (.PA)	To process the PAUSE statement.	P. 11-16
SPMSG (.SP)	To print the octal coded number with STOP and PAUSE.	P. 11-17
.SS	To calculate the array element address.	P. 11-12

Program Name	Function	Ref. Manual
STOP (.ST)	To process the STOP statement and return control to the monitor.	P. 11-15
TIME	To provide the ability to record elapsed time in minutes and seconds.	P. 11-21

b. Science Library Routines

The following PDP-9 Science Library routines are utilized by the RPSS programs. These programs are described in the DEC PDP-9 manual referenced in Section II, Subsection 2a above:

Routine Name	External Cells	Ref. Manual
.CB (Short Get Argument)	None	P.III-9
.DA (General Get Argument)	None	P.III-8
INTEGE (Integer Arithmetic)	REAL	P.III-8
REAL (Real Arithmetic)	None	P.III-8

SECTION III

ASSEMBLY LANGUAGE SUBPROGRAM FLOW DIAGRAMS

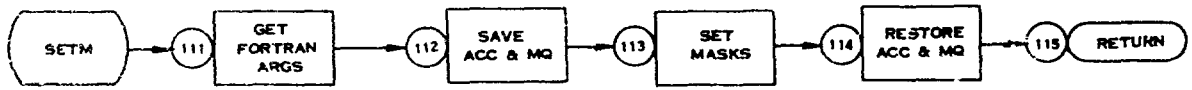


FIGURE IV - 6. FLOW DIAGRAM OF SETM

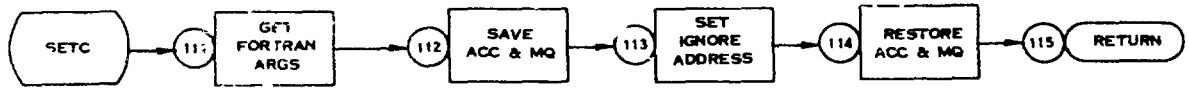


FIGURE IV - 7. FLOW DIAGRAM OF SETC

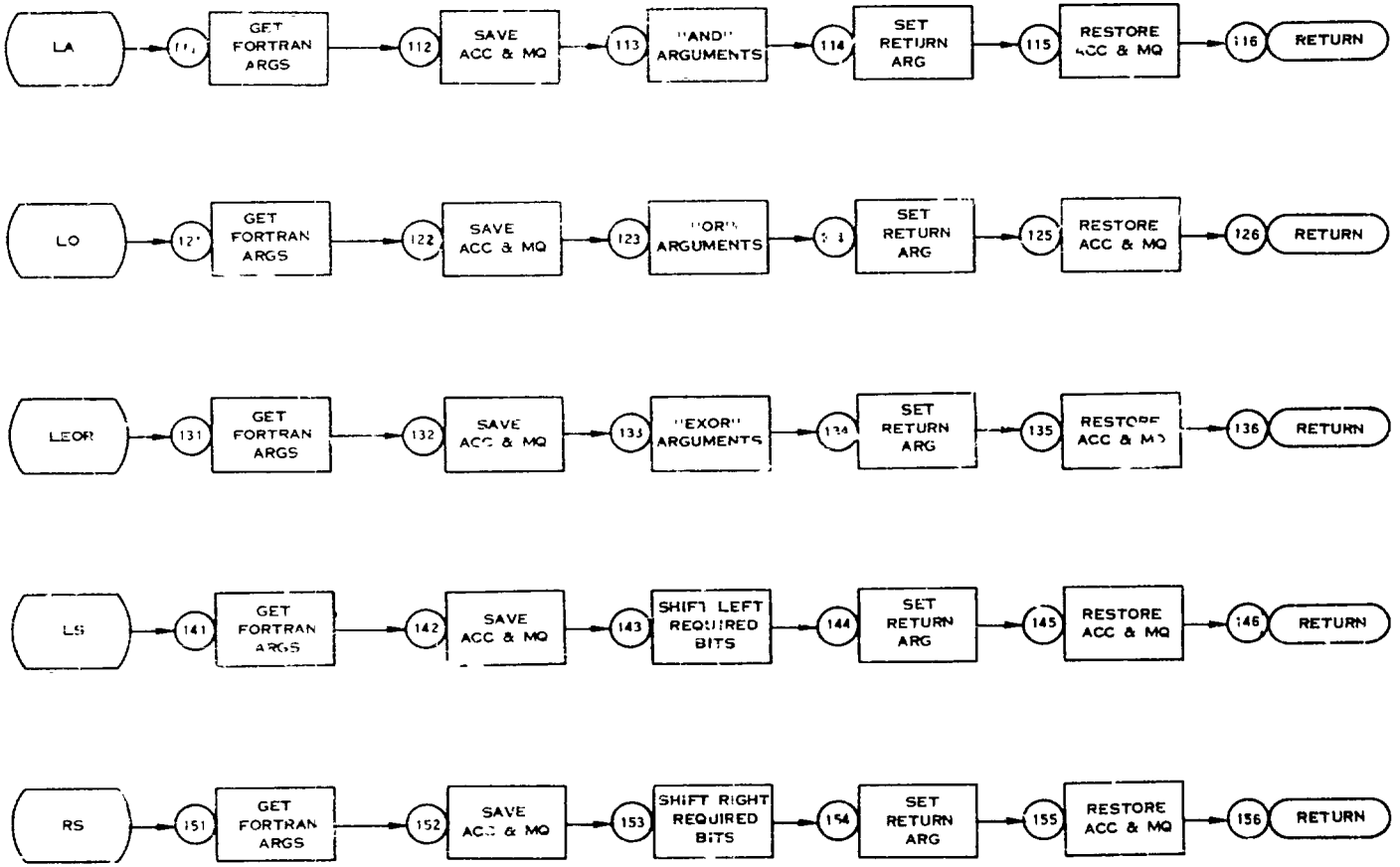


FIGURE IV - 8 FLOW DIAGRAM OF BITS

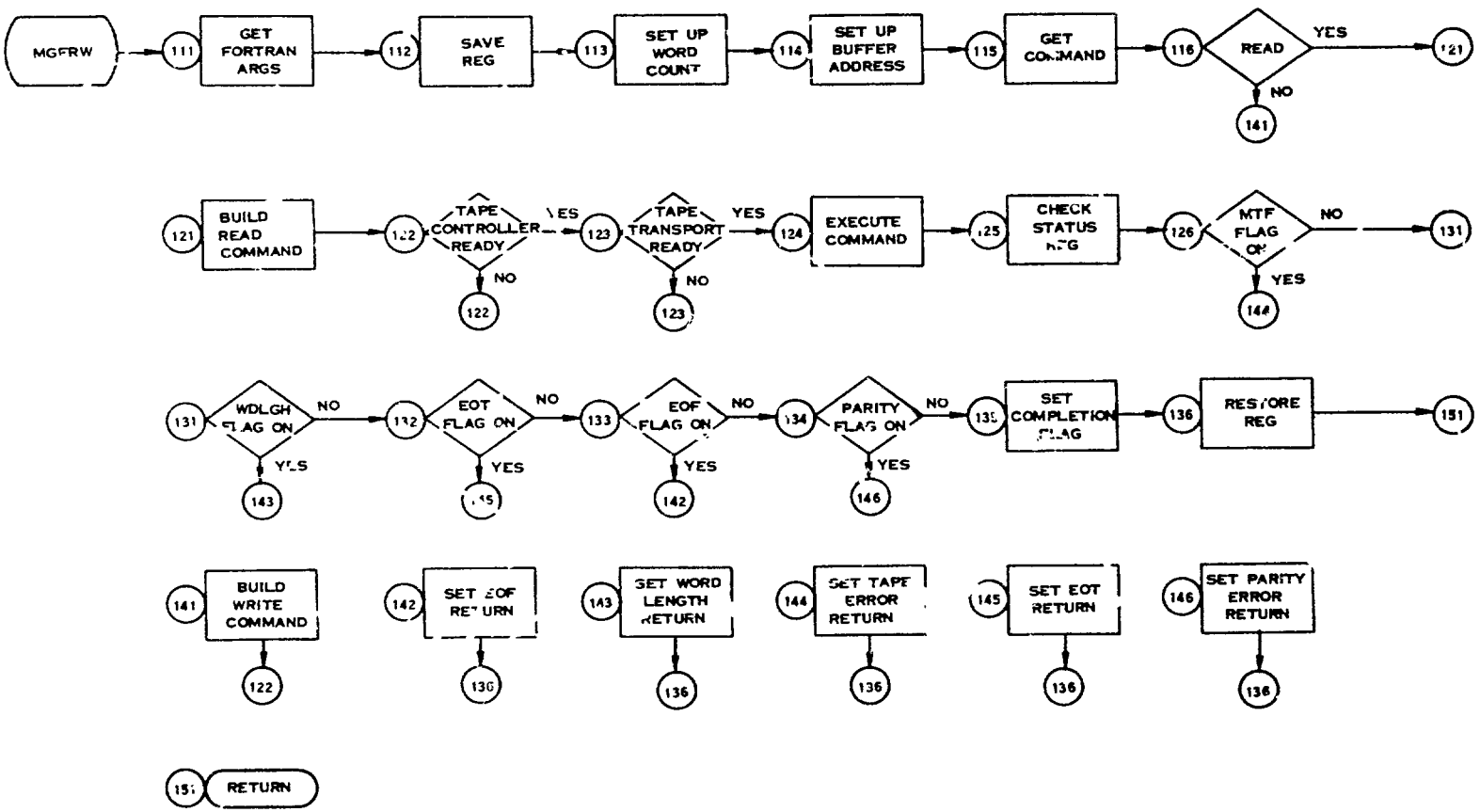


FIGURE IV - 9. FLOW DIAGRAM OF MGFRW

SECTION V

FORTRAN SUBPROGRAM AUTOFLOW DIAGRAMS

Preceding page blank

FORTRAN MODULE (LIST,NAMSQ,AUTOSEQ)

CARD NO	**** CONTENTS ****
1	DIMENSION ISC(1000,2),A(9)
2	DIMENSION MASK1(3),MASK2(3),MASKS(6)
3	DIMENSION ITBUF(740)
4	DIMENSION INSET(4),IOBUF(1364)
5	DIMENSION ICNT(25),ICH(4)
6	COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
7	COMMON /B1/ INSC,INCH,INOEN,INDEL
9	COMMON /C1/ IAEF,IAFCH,IAFOEN
9	COMMON /D1/ A,IDEN,ISCS,ISCE,IODEN
10	COMMON /E1/ TN,TA,TE,TX,TY,TD,TO,TT,TS,TR,TM
11	COMMON /F1/ ITTY,IPT,IDT,INP,IPRTR
12	COMMON /G1/ IOR1,IOR2,IOR3,IOEC
13	COMMON /H1/ IOSC,IOCH,IOOEN,IODEL
14	COMMON /I1/ IMONTH,IDAY,IYEAR,IPAGE,IFILE,IPEV
15	COMMON /J1/ IMGT1,IMGT2,IODNE
16	COMMON /MASK/ MASK1,MASK2
17	COMMON /GNR1/ ITYPE,ICORD,ICON1,ICON2,NX,IX,IFT,ISTAT
18	COMMON /TBUF/ ITBUF,NYS,NXS
19	COMMON /AFLC/ IMINF,IMNAR
20	COMMON /IOUT/ IOUT1,IOUT2,IATST
21	COMMON /K1/ ICNT,ICH
22	COMMON /L1/ IMIN,ISEC,IOEF
23	COMMON /M1/ INSET,LSTWRD,LSTRYT,NENT,IOBUF
24	EQUIVALENCE (MASK1(1),MASKS(1))
25	EQUIVALENCE (MASK11,MASK1(1)),(MASK12,MASK1(2)),(MASK13,MASK1(3))
26	EQUIVALENCE (MASK21,MASK2(1)),(MASK22,MASK2(2)),(MASK23,MASK2(3))

```
C
C     INITIALIZATION
C
      ISWCH = 12
      IWRITE = 0
      ITTY = 8
      IPT = 4
      IDT = 3
      IAD = 2
      IPRTR = 1
      IDR1 = 5
      IDR2 = 6
      IDR3 = 7
      IDEC = 3
      IFILE = 1
      NX = 0
      IY = 0
      IET = 0
      ISTAT = 0
      NYS = 0
      NXS = 0
      IMIN = 0
      ISEC = 0
      ITMIN = 0
      ITSEC = 0
      IOFF = 0
      ITMAX = 1000
      ITCUR = 1
```

```
CARD NO  ****                    CONTENTS                    ****
      55        MINSC = 1
      56        MAXSC = 99999
      57        ITTYR = 4
      58        NRWEDS = 756
      59        IMGT1 = IOR2
      60        IMGT2 = IOR3
      61        IREV = 0
      62        CALL SETM(MASKS(1))
      63        CALL SETC(INSET(1))
      64      C
      65      C     SET UP THE BINARY CONSTANTS
      66      C
      67        CALL PBYTE(INSET(1),1,IDUMC,2)
      68        IEOL = MASK11
      69        CALL RS(IEOL,1,IEOL)
      70        CALL LS(7,14,IEOM)
      71        IOMC = 1
      72      C
      73      C     OUTPUT SCRATCH MOUNT MESSAGE
      74      C
      75          1 WRITE (ITTY,1000) IOR1
      76       1000 FORMAT (,28H MOUNT NEW SCRATCH ON DRIVE ,I1,19H   ENTER   D   WHEN,
      77           15H DONE)
      78        READ (ITTY3,2000) A(1)
      79       2000 FORMAT (A1)
      80      C
      81      C     CHECK
      82      C
```

```
      IF (A(1) .NE. TO) GO TO 1
C
C     DATE MESSAGE   AND  SAVE IT
C
      WRITE (ITTY,1001)
 1001 FORMAT (46H ENTER MONTH,DAY,YEAR AS 6 NUMERICS ... MMDDYY)
      READ (ITTY,2001) IMONTH,IDAY,IYEAR
 2001 FORMAT (3I2)
C
C     TAPE ID MESSAGE  AND  SAVE IT
C
  500 WRITE (ITTY,1002)
 1002 FORMAT (29H ENTER 5 DIGIT TAPE ID NUMBER)
      READ (ITTY,2002) IDT
 2002 FORMAT (I5)
C
C     INPUT COUNT MESSAGE
C
      WRITE (ITTY,1003) IDT2
 1003 FORMAT (39H MOUNT 1ST REEL OF INPUT TAPE ON DRIVE ,I1)
      WRITE (ITTY,1101)
 1101 FORMAT (49H ENTER 4 DIGIT MINIMUM AREA IN PROPER RESOLUTION.)
      READ (ITTY,2100) IMNAR
 2100 FORMAT (I6)
C
C     MODE OF OPERATION MESSAGE  :1= TAPE ,  2 = MANUAL
C
      DO 700 J = 1,25
```

```
CARD NO      ****                          CONTENTS                          ****
     111        700 ICNT(J) = 0
     112            DO 701 J = 1,4
     113        701 ICH(J) = 0
     114         50 WRITE (ITTY,1100)
     115       1100 FORMAT (50H DO YOU WISH TO SUPPLY SELECTION CODES EXTERNALLY / 13H
     116          1 ENTER Y OR N)
     117            READ (ITTY,2000) A(1)
     118            IF (A(1) .EQ. TY) GO TO 21
     119            IF (A(1) .NE. TN) GO TO 50
     120            MODE = 1
     121            GO TO 200
     122         21 CALL SCODE
     123            MODE = 2
     124        200 CONTINUE
     125          C
     126          C     START PROCESSING OF DATA
     127          C
     128            CALL IMESG
     129            CALL NSET
     130            ICURY = -1
     131            IFT = 0
     132            IMINF = 0
     133            CALL LS(IAEDEN,3,IWRK2)
     134            CALL LOR(IWRK2,IAECH,IWRK2)
     135            CALL LEOR(IWRK2,4,IWRK2)
     136            CALL LS(IWRK2,6,IAECOM)
     137            IOUT1 = 0
     138            IOUT2 = 0
```

```
      IATST = 0
C
      CALL TIME (IMIN,ISEC,IOFF)
C
  300 IF (IMINF .NE. 1) CALL GNX
      IF (ITYPE.GT. 2) GO TO 301
      IF (IDONF .EQ. 1) GO TO 390
      CALL IMESG
      GO TO 300
C
C     NOT EOF OR EOT - CHECK IF Y OR Y
C     IF IFT = 0 THEN FIRST Y
C
  301 IF (ITYPE .EQ. 3) GO TO 302
      IF (ICORD .EQ. ICUPY) GO TO 305
      IF (IFT .NE. 0) GO TO 380
      IFT = 1
      GO TO 370
C
C     PROCESS THE Y
C
  302 IWRK1 = 0
      IWRK2 = 0
C
C     CHECK MODE: 1=TAPE ; 2=SC TABLE
C
      IF (MODE .EQ. 1) GO TO 360
C
```

```
CARD NO  ****                    CONTENTS                    ****
   167         C     SC MODE - GET THE SELECTION CODE
   168         C
   169               CALL LA(ICOM2,MASK23,IWRK1)
   170               CALL RS(IWRK1,6,IWRK1)
   171         C
   172         C     IS IT IN SC TABLE
   173         C
   174               CALL SRCH(IWRK1,INDEX)
   175               IF (INDEX) 306,303,303
   176         C
   177         C     NOT IN TABLE,CHECK ALL ELSE
   178         C
   179           306 IF (IAFF .NE. 1) GO TO 300
   180         C
   181         C     ALL ELSE ON - SET IT UP
   182         C
   183               IOUT1 = IAECOM
   184           304 IOUT2 = ICORD
   185               CALL LA(ICOM2,MASK13,IATST)
   186               ICNT(IATST) = ICNT(IATST) + 1
   187               CALL LA(IOUT1,192,IWKCH)
   188               CALL RS(IWKCH,6,IWKCH)
   189               ICH(IWKCH+1) = ICH(IWKCH+1) + 1
   190               GO TO 305
   191         C
   192         C     IN TABLE - CHECK DELETE FLAG
   193         C
   194           303 IWRK4 = ISC(INDEX,2)
```

```
195               CALL LAND(IWRK4,MASK13,IWRK3)
196               IF (IWRK3 .EQ. 1) GO TO 390
197         C
198         C     DELETE OFF  -  SET IT UP
199         C
200               CALL LAND(IWRK4,MASK12,IOUT1)
201               GO TO 304
202         C
203         C     TAPE MODE
204         C
205           360 CALL LAND(ICON2,MASK12,IOUT1)
206               GO TO 304
207         C
208         C     MAIN STORAGE LOOP
209         C
210           305 IF ((IATST .NE. 0) CALL SETA
211               ISWCH = 313
212               IF (NENT .EQ. 567) GO TO 307
213               NENT = NENT + 1
214         C
215         C     PUT IN FIRST WORD
216         C
217           313 LSTBYT = LSTBYT + 1
218               IF (LSTBYT .NE. 4) GO TO 308
219               LSTWRD = LSTWRD + 1
220               LSTBYT = 1
221           308 CALL PBYTE(IOUT1,2,IOBUF(LSTWRD),LSTBYT)
222         C
```

```
CARD NO    ****                    CONTENTS                    ****
   223     C     NOW 3 BYTES OF SECOND
   224     C
   225           (LSTBYT .NE. 3) GO TO 309
   226           LSTWRD = LSTWRD + 1
   227           IOBUF(LSTWRD) = IOUT2
   228           GO TO 310
   229     C
   230     C     NOT EVEN - DO IT BY BYTES
   231     C
   232       309 DO 311 KK = 1,3
   233           LSTBYT = LSTBYT + 1
   234           IF (LSTBYT .NE. 4) GO TO 312
   235           LSTWRD = LSTWRD + 1
   236           LSTBYT = 1
   237       312 CALL PBYTE(IOUT2,KK,IOBUF(LSTWRD),LSTBYT)
   238       311 CONTINUE
   239     C
   240       310 CONT. OF
   241           IF (ISWCH - 314) 313,314,317
   242     C
   243       313 IF ((ITYPE .EQ. 3) .OR. (ITYPE .EQ. 4)) GO TO 300
   244       320 CALL NSET
   245       370 ICURY = ICORD
   246           GO TO 300
   247     C
   248     C     BUFFER FULL
   249     C
   250       307 IOUT1S = IOUT1
```

```
      IOUT2S = IOUT2
      IOUT2 = IONE
  318 IOUT1 = IOUMC
      ISWCH = 316
      CALL PBYTE(IOUMC,2,IOBUF(1),1)
      DO 319 KK = 2,3
  319 CALL PBYTE(IOUBY,KK-1,IOBUF(1),KK)
      CALL PBYTE(IOUBY,3,IOBUF(2),1)
      GO TO 315
C
  380 IOUT2 = IEOL
      GO TO 318
  314 IOSTAT = 0
      CALL MGFRW(IOR1,NPWRDS,IOBUF(1),1,IOSTAT)
      IWRITE = IWRITE + 1
      IF (IOSTAT .GT. 4) WRITE (8,2222) IWRITE
 2222 FORMAT (23H BAD WRITE AT IWRITE = ,I5)
      IF (IOSTAT .EQ. 3  .OR.  IOSTAT .EQ. 4) GO TO 3
      IF (IOSTAT .NE. 1) GO TO 316
C
C     END OF TAPE
C
      ISWCH = 317
      GO TO 309
  317 CONTINUE
      IOFF = 1
      IIMIN = IIMIN + IMIN
      IISEC = IISEC + ISEC
```

```
CARD NO    ****                    CONTENTS                    ****
     279          IF (IISEC .LT. 60) GO TO 600
     280          IIMIN = IIMIN + 1
     281          IISEC = IISEC - 60
     282    600 CONTINUE
     283          IOFF = 0
     284          REWIND IDR1
     285          WRITE (ITTY,1050) IDR1
     286   1050 FORMAT (40H AT PAUSE 2, MOUNT NEW SCRATCH ON DRIVE ,I1,14H  AND CO
     287         INTINUE)
     288    C
     289          PAUSE 2
     290          CALL TIME (IMIN,ISEC,IOFF)
     291    C
     292     316 IF (ITYPE .LE. 2) GO TO 391
     293          IF (ITYPE .EQ. 4) GO TO 320
     294          CALL NSET
     295          IOUT1 = IOUTLS
     296          IOUT2 = IOUT2S
     297          GO TO 305
     298    C
     299     390 IOUT2 = IEOM
     300          WRITE (8,5000) IWRITE
     301    5000 FORMAT (16H TOTAL WRITES = ,I5,4H + 1)
     302          IWRITE = 0
     303          GO TO 318
     304    C
     305     391 GO TO 400
     306    C
```

```
C     OUTPUT TAPE ERROR
C
  350 STOP 12
C
C     PRINT OUTPUT STATISTICS
C
  400 IOFF = 1
      IMIN = IMIN + IIMIN
      ISEC = ISEC + IISEC
      IF (ISEC .LT. 60) GO TO 7000
      IMIN = IMIN + 1
      ISEC = ISEC - 60
 7000 CONTINUE
      IIMIN = 0
      IISEC = 0
      CALL PRNSTT
      IOFF = 0
      IFLG2 = 0
C
C     RUN TERMINATION MESSAGE
C
      WRITE (ITTY,1200)
 1200 FORMAT (42H DO YOU WISH TO TERMINATE RUN   (Y  OR  N))
      READ (ITTY,2000) A(1)
      IF (A(1) .EQ. TY) GO TO 401
C
C     CONTINUE RUN - MULTIFILE MESSAGE
C
```

```
CARD NO    ****                    CONTENTS                    ****
     335          WRITE (ITTY,1201)
     336     1201 FORMAT (47H DO YOU WISH TO STACK OUTPUT FILES   (Y  OR  N))
     337          READ (ITTY,2000) A(1)
     338          IF (A(1) .EQ. TY) GO TO 402
     339     C
     340     C    NEW OUTPUT TAPE REQUIRED
     341     C
     342          IFLG2 = 1
     343      401 REWIND IMGT2
     344          CALL MGERW(IDR1,NRWRDS,IDRUF(1),2,IOSTAT)
     345          REWIND IDR1
     346     C
     347          IMGT1 = IDR2
     348          IMGT2 = IDR3
     349     C
     350     C    TAPE REMOVAL MESSAGE
     351     C
     352          WRITE (ITTY,1202) IDR1,IMONTH,IDAY,IYEAR,IDT,IREEL
     353     1202 FORMAT (28H REMOVE OUTPUT TAPE ON DRIVE,1X,I1/ 15H LABEL IT  DATE,
     354         14X,I2,1H/,I2,1H/,I2,14H TAPE NUMBER  ,I6,14H REEL NUMBER  ,I2)
     355     C
     356     C    CHECK IF STACKED OUTPUT REQUIRED
     357     C
     358          IF (IFLG2 .NE. 1) GO TO 999
     359          GO TO 403
     360     C
     361     C    STACK OUTPUT
     362     C
```

```
      402 CALL MGFRW(IOR1,NRWRDS,IOPUF(1),2,IOSTAT)
          REWIND IMGT2
C
C     CHECK MODE
C
      403 CONTINUE
          IF (MODE .EQ. 1) GO TO 404
C
C     SELECTION CODE MODE - - SAME SC MESSAGE
C
          WRITE (ITTY,1203)
 1203 FORMAT (61H DO YOU WISH TO USE THE SAME SELECTION CODES AS PREVIOU
     1S RUN / 11H (Y  OR  N))
          READ (ITTY,2000) A(1)
          IF (A(1) .NE. IY) GO TO 404
C
C     CHECK FOR STACKED OUTPUT
C
          IF (IFLG2 .EQ. 1) GO TO 405
C
C     CHECK FOR EOT - - STACK OUTPUT
C
          IF (IOSTAT .NE. 1) GO TO 406
C
C     OUTPUT TAPE FINISHED - CANNOT STACK - NEED NEW OUTPUT TAPE
C
      405 WRITE (ITTY,1000) IO81
          READ (ITTY,2000) A(1)
```

```
CARD NO  ****                          CONTENTS                          ****
     391        IF(ALL .NE. TO) GO TO 405
     392        WRITE (ITTY,1001)
     393        READ (ITTY,2001) IMONTH,IDAY,IYEAR
     394        WRITE (ITTY,1002)
     395        READ (ITTY,2002) IDT
     396      406 WRITE (ITTY,1003)IDR2
     397        GO TO 200
     398      404 CONTINUE
     399      C
     400      C     COMPLETELY NEW INPUT FORMAT - CHECK FOR STACKED OUTPUT
     401      C
     402        IF (IFLG2 .EQ. 1) GO TO 1
     403        GO TO 500
     404      C
     405      C     TERMINATION PROCEDURE - - TERMINATION MESSAGE
     406      C
     407        999 WRITE (ITTY,1204)
     408       1204 FORMAT (42H END OF RUN - REMOVE ALL TAPES FROM DRIVES///20X,20H **
     409           1* TERMINATION ***)
     410      C
     411      C
     412        STOP
     413        END
     414        SUBROUTINE SCODE                                            10
     415      C                                                             20
     416      C *** PART OF THE RPS SYSTEM THAT HANDLES THE SELECTION CODE PROCESSING  30
     417      C *** OF INPUT DATA.  THIS SUBROUTINE ACCEPTS SELECTION CODES, DENSITIES  40
     418      C *** AND CHANNEL NUMBERS THAT WILL BE THE SELECTION CRITERIA FOR THE    50
```

```
C *** FINAL OUTPUT PRODUCT ONCE PROCESSING IN PHASE II IS COMPLETED.
C *** THIS IS AN OPTION IN ADDITION TO THE MEANS OF PROCESSING THE DATA
C *** DIRECTLY FROM THE INPUT TAPE
C
      DIMENSION ISC(1000,2),A(9)
      COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
      COMMON /B1/ INSC,INCH,INDEN,INDEL
      COMMON /C1/ IAFF,IAECH,IAFDEN
      COMMON /D1/ A,IDEN,ISCS,ISCF,IDDEN
      COMMON /E1/ TN,TA,TE,TX,TY,TD,TP,TT,TS,TR,TM
      COMMON /E1/ TTTY,IPT,IDT,INP,IPRTR
      COMMON /G1/ IDR1,IDR2,IDR3,IDEC
      COMMON /H1/ IDSC,IDCH,IDDEN,IDDEL
      COMMON /I1/ IMONTH,IDAY,IYEAR,IPAGE,IFILE,IREV
C
      WRITE (ITTY,1004)
 1004 FORMAT (45H ENTER DEFAULT DENSITY AS NUMERIC FROM 0 T) 7)
C
C *** READ DENSITY
C
      READ (ITTY,2003) IDDEN
 2003 FORMAT (I1)
C
C *** PRESET SC TABLE
C
      INSC = 0
      INCH = 0
      INDEL = 0
```

```
CARD NO  ****                          CONTENTS                          ****
     447        ICH = 0
     448        DO 2 J = 1,ITMAX
     449      2 CALL INTWDL(J)
     450  C
     451  C *** CLEAR ALL ELSE FUNCTION
     452  C
     453        IREV = 0
     454        IAEF = 0
     455        IAECH = 5
     456        IAEDEN = IDDEN
     457  C                                                                 500
     458      9 WRITE (ITTY,1009)                                           510
     459   1009 FORMAT (51H ENTER CHANNEL NUMBER FROM 1 TO 4 OR ENTER 0 TO END/ 52
     460       1H SELECTION CODE INPUT EACH TIME   CH NR =   IS TYPED)
     461     11 WRITE (ITTY,1011)
     462   1011 FORMAT (8H CH NR =)
     463        READ (ITTY,2003) INCH
     464        IF (INCH .EQ. 0) GO TO 100
     465        INCH = INCH - 1
     466        IF (INCH .GE. 4) GO TO 9
     467  C
     468  C *** AUTOMATIC/MANUAL MESSAGE
     469  C
     470        INP = ITTY
     471        WRITE (ITTY,1012)
     472   1012 FORMAT (12H ENTER CODES,2X,15H(WYZNNNNNNMMMM))
     473  C
     474  C *** PUT THE ENTRIES IN THE TABLE
```

```
      C
            NR = 0
            MINS1 = -1
            MINS2 = -2
            MINS3 = -3
            MINS4 = -4
         10 CALL MKENT(NR)
            IF (NR .GE. 0) GO TO 10
            IF (NR .EQ. MINS1) GO TO 11
            IF (NR .EQ. MINS2) GO TO 10
            IF (NR .EQ. MINS3) GO TO 100
            IF (NR .EQ. MINS4) PAUSE 10
            GO TO  10
      C
      C *** INPUT COMPLETE -   SORT THE TABLE
      C
        100 CALL SHLSRT
      C
      C *** PRINT THE TABLE
      C
            IOELT = 0
            CALL NPAGE
            LCNT = 0
            ICNT = ITCUR - 1
            DO 15 I = 1,ICNT
      C
      C *** OBTAIN PARAMETERS FROM SC TABLE
      C
```

```
CARD NO  ****                    CONTENTS                    ****
     503          CALL OUTMDL(I)
     504          WRITE (IPRTR,1018) IOSC,IOCH,IODEN,IODEL
     505    1018 FORMAT (6X,I5,15X I2,11X,I1,12X,I1)
     506          LCNT = LCNT + 1
     507          IF (LCNT .LT. 40) GO TO 15
     508          LCNT = 0
     509          IPAGE = IPAGE + 1
     510          CALL NPAGE
     511       15 CONTINUE
     512       C
     513       C *** CHECK ALL ELSE AND OUTPUT IF ON
     514       C
     515          IF (IAEF .NE. 1) GO TO 14
     516          WRITE (IPRTR,1018) IAECH,IAEDEN
     517    1019 FORMAT (5X,8HALL ELSE,13X,I2,11X,I1)
     518       C
     519       C *** SUMMARY MESSAGE
     520       C
     521       14 WRITE (ITTY,1020)
     522    1020 FORMAT (61H SELECTION CODE SUMMARY ON PRINTER. ARE CORRECTIONS REQ
     523         1UIRED / 14H ENTER Y OR N.)
     524       C
     525       C *** READ ANSWER
     526       C
     527          READ (ITTY,2000) A(1)
     528          IF (A(1) .NE. TY) GO TO 16
     529       C
     530       C *** CORRECTIONS REQUIRED
```

```
C
      IREV = IREV + 1
      GO TO 9
   16 CONTINUE
      IF (A(1) .NE. TN) GO TO 14
C
C *** NO CORRECTIONS - - - RETURN
C
      RETURN
      END
      SUBROUTINE MKENT(IFLG)
C
C *** READS THE INPUT SELECTION CODE AND ENTERS IT INTO THE
C *** SELECTION CODE TABLE
C
      DIMENSION ISC(1000,2), A(9)
      COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
      COMMON /B1/ INSC,INCH,INDEN,INDEL
      COMMON /C1/ IAEF,IAECH,IAEDEN
      COMMON /D1/ A,IDEN,ISCS,ISCF,IDDEN
      COMMON /E1/ TN,TA,TE,TX,TY,TD,TP,TT,TS,TR,TM
      COMMON /F1/ ITTY,IPT IDT,INP,IPRTR
      COMMON /G1/ IDR1,IDR2,IDR3,IDEC
      COMMON /H1/ IOSC,IOCH,IODEN,IODEL
C
      IFLG = 0
C
C *** READ THE ENTRY
```

```
CARD NO  ****                          CONTENTS                          ****
  559    C
  560          READ (INP,2004) A(1),A(2),IDEN,ISCS,ISCE
  561     2004 FORMAT (A1,A1,I1,2I5)
  562    C
  563    C *** TEST FOR END
  564    C
  565          IF (A(1) .NE. TX) GO TO 1
  566          IFLG = -1
  567          RETURN
  568    C
  569    C *** NOT END - - - CHECK THE ENTRY
  570    C *** ERROR CHECK CODE
  571    C
  572        1 IF ((A(1) .NE. TA) .AND. (A(1) .NE. TF) .AND. (A(1) .NE. TX) .AND.
  573         1 (A(1) .NE. TN)) GO TO 4
  574          IF ((A(2) .NE. TS) .AND. (A(2) .NE. TR)) GO TO 4
  575          IF ((IDEN .LT. 0) .OR. (IDEN .GT. 8)) GO TO 4
  576          IF ((ISCS .LT. MINSC) .OR. (ISCS .GT. MAXSC)) GO TO 4
  577          IF (((A(2) .EQ. TR) .AND. ((ISCE.LT.MINSC) .OR. (ISCE.GT.MAXSC)))
  578         1 GO TO 4
  579    C
  580    C *** ENTRY OK --- NOW DETERMINE THE TYPE
  581    C
  582    C
  583          IF (A(1) .NE. TA) GO TO 2
  584    C
  585    C *** ENTRY IS ALL ELSE
  586    C
```

```
587               IAEF = 1
588               IAECH = INCH
589               IF (IDEN .NE. 0) IAEDEN = IDEN
590               RETURN
591         C
592         C *** NOT ALL ELSE - - - MUST BE AN AND OR DELETE
593         C
594             2 INDEL = 0
595               IF (A(1) .EQ. TE) INDEL = 1
596         C
597         C *** SET UP FOR INTWDL - PRESET DEFAULT DENSITY AND CHANGE IF NECESSARY
598         C
599               INDEN = IDDEN
600               IF (IDEN .GT. 0) INDEN = IDEN
601         C
602         C *** SET UP DO LOOP
603         C
604               JJ = ISCS
605               KK = ISCE
606               IF (A(2) .NE. TR) KK = ISCS
607         C
608         C *** DO LOOP FORMS THE TABLE
609         C
610               DO 6 I = JJ,KK
611               INSC = I
612               CALL SCAN(I,K)
613         C
614         C *** CHECK FOR PREVIOUS ENTRY
```

```
CARD NO    ****                          CONTENTS                          ****
     615   C
     616         IF (K .GT. 0) GO TO 7
     617   C
     618   C *** NO PREVIOUS ENTRY
     619   C
     620         K = ITCUR
     621         ITCUR = ITCUR + 1
     622   C
     623   C *** IF PREVIOUS ENTRY, OVERWRITE, IF NOT ADD AT END OF TABLE
     624   C
     625       7 CALL INTMDL(K)
     626   C
     627   C *** CHECK IF TABLE FULL
     628   C
     629         IF (ITCUR .GT. ITMAX) GO TO 8
     630   C
     631   C *** TABLE  NOT FULL
     632   C
     633       6 CONTINUE
     634         IF (INP .EQ. ITTY) WRITE (ITTY,1012)
     635    1012 FORMAT (3H OK)
     636         IFLG = K
     637         RETURN
     638   C
     639   C *** TABLE FULL - STOP ENTERING  -  WRITE LAST ENTRY MESSAGE
     640   C
     641       8 WRITE (ITTY,1010) A(1),A(2),IDEN,ISCS,ISCE
     642    1010 FORMAT (39H SELECTION CODE TABLE FULL - LAST ENTRY,
```

```
643             1/1X,A1,1X,A1,1X,I1,1X,I5,1X,I4)
644              IFLG = -3
645              RETURN
646       C
647       C *** ERROR FOUND - PRINT ERROR MESSAGE
648       C
649           4 WRITE (ITTY,1008) A(1),A(2),IOFN,ISCS,ISCE
650        1008 FORMAT (58H ERRONEOUS SELECTION CODE ENTRY - WILL NOT BE PUT IN TA
651             1BLE/ 22H ENTRY AS RECEIVED IS /1X,A1,1X,A1,1X,I1,1X,I5,1X,I4/43H A
652             2T PAUSE 1, CONTINUE OR MANUALLY TERMINATE)
653       C
654              PAUSE 1
655       C
656              IFLG = -2
657              RETURN
658              END
659              SUBROUTINE GNR
660       C
661       C *** GET THE NEXT RECORD FROM INPUT TAPE
662       C
663              DIMENSION ITBUF(740)
664              COMMON /GNP1/ ITYPE,ICOPD,ICON1,ICON2,NX,IX,IFT,ISTAT
665              COMMON /TBUF/ ITBUF,NYS,NXS
666              COMMON/J1/INGT1,INGT2,IOONE
667       C
668              IF (IFT .NE. 2) GO TO 1
669              IREAD = 0
670              NRWRDS = 640
```

```
CARD NO  ****                    CONTENTS                    ****
     671          ITAPE = IMGT2
     672          CALL RFOF(ITAPE)
     673        3 CONTINUE
     674          CALL MGFRW(IMGT2,NRWRDS,ITBUF(1),0,ISTAT)
     675          IREAD = IREAD + 1
     676          IF (ISTAT .EQ. 1  .OR.  ITBUF(1) .EQ. 61440) GO TO 6
     677          IF (ISTAT .GT. 4) GO TO 2
     678      C
     679          ITYPE = ISTAT
     680        7 IX = 14
     681          ICORD = ITBUF(10)
     682          ICON1 = ITBUF(11)
     683          IF (IFT .EQ. 0) WRITE (9,1010) ITBUF(10)
     684       1010 FORMAT (23H FIRST Y COORDINATE IS ,I6)
     685          ICON2 = ITBUF(12)
     686          NX = ITBUF(6)
     687          NYS = NYS + 1
     688          NXS = NXS + NX
     689      C
     690      C     TYPE IS Y
     691      C
     692          IF (ITYPE .EQ. 3) ITYPE = 4
     693          IF (ITYPE .EQ. 4) RETURN
     694      C
     695        1 IF (NX .EQ. 0) GO TO 3
     696          IF (IX .GT. 640) GO TO 4
     697        5 ICORD = ITBUF(IX)
     698          ICON1 = ITBUF(IX+1)
```

```
699               ICON2 = ITBUF(IX+2)
700               IX = IX + 4
701               ITYPE = 3
702         C
703         C     TYPE IS X
704         C
705               NX = NX - 1
706               RETURN
707         C
708             4 CONTINUE
709               CALL MGERW(ITAGT2,NRWRDS,ITBUF(1),0,ISTAT
710               IREAD = IREAD  1
711               IF (ISTAT .GT. 4) WRITE (5,1020) IREAD
712               IF (ISTA. .EQ. 1  .OR.  ITBUF(1) .EQ. 61440) GO TO 4
713               IX = 6
714               GO TO 5
715         C
716             2 ITYPE = 5
717               ITEST = -32650
718               IF (ISTAT .EQ. 4) ITYPE = 3
719               IF (ITBUF(7) .EQ. ITEST) ITYPE = 4
720               IF (ITYPE .EQ. 5) WRITE (5,1020) IREAD
721               IF (ITBUF(7) .EQ. ITEST) GO TO 7
722          1020 FORMAT (21H BAD READ AT IREAD = ,I5)
723               RETURN
724         C
725         C     EOF OR EOT ENCOUNTERED
726         C
```

```
CARD NO  ****                          CONTENTS                          ****
   727        6 ITYPE = ISTAT
   728          IF (ITBUF(1) .EQ. 61440) ITYPE = 2
   729          WRITE (8,1030) IREAD
   730       1030 FORMAT (15H TOTAL READS = ,I5)
   731          RETURN
   732          END
   733          SUBROUTINE SCAN(IVAL,K)
   734     C
   735     C *** SCANS SELECTION CODE TABLE FOR DUPLICATE ENTRIES
   736     C
   737          DIMENSION ISC(1000,2)
   738          COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
   739     C
   740          JJ = ITCUR - 1
   741          DO 1 I =1,JJ
   742          IF (ISC(I,1) .NE. IVAL) GO TO 1
   743     C
   744     C *** DUPLICATE FOUND
   745     C
   746          K = I
   747          RETURN
   748        1 CONTINUE
   749     C
   750     C *** NO DUPLICATE - I.E. NOT IN TABLE
   751     C
   752          K = -1
   753          RETURN
   754          END
```

```
      SUBROUTINE SHLSRT
C
C *** SORT THE SELECTION CODE TABLE IN INCREASING ORDER BY FIRST WORD
C
      DIMENSION ISC(1000,2),IB(1,2)
      COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
C
      LN = ITCUR - 1
    1 LN = LN/2
      IF (LN .EQ. 0) GO TO 2
C
C *** CONTINUE SORT
C
      J = 1
      K = ITCUR - 1 - LN
    3 I = J
    4 IM = I + LN
      IF (ISC(I,1) .LE.ISC(IM,1)) GO TO 5
C
C *** PERFORM EXCHANGE
C
      DO 6 JJ = 1,2
      IB(1,JJ) =ISC(I,JJ)
      ISC(I,JJ) = ISC(IM,JJ)
    6 ISC(IM,JJ) = IB(1,JJ)
      I = I - LN
      IF (I .GE. 1) GO TO 4
C
```

```
CARD NO      ****                    CONTENTS                    ****
   783       C *** IN CORRECT ORDER
   784       C
   785           5 J = J + 1
   786       C
   787       C *** MORE TO SORT
   788       C
   789             IF (J - K) 3,3,1
   790       C
   791       C *** SORT FINISHED
   792       C
   793           2 CONTINUE
   794             RETURN
   795             END
   796             SUBROUTINE SRCH(IARG,K)
   797       C
   798       C *** PERFORMS A BINARY SEARCH OF THE SELECTION CODE TABLE
   799       C
   800             DIMENSION ISC(1000,2)
   801             COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
   802       C
   803             IFRST = 1
   804             ILAST = ITCUR - 1
   805           1 IF ((ILAST - IFRST) .LT. 0) GO TO 4
   806             LN = (ILAST - IFRST)/2 + IFRST
   807             IF (IARG .EQ. ISC(LN,1)) GO TO 2
   808             IF (IARG .GT. ISC(LN,1)) GO TO 3
   809       C
   810       C *** VALUE IN FIRST PART OF TABLE SECTION BEING SEARCHED
```

```
C
      ILAST = LN - 1
      GO TO 1
    3 IFRST = LN + 1
C
C *** VALUE IN SECOND PART OF TABLE SECTION
C
      GO TO 1
C
C *** VALUE FOUND IN TABLE
C
    2 K = LN
      RETURN
C
C *** VALUE NOT IN TABLE
C
    4 K = -1
      RETURN
      END
      SUBROUTINE INTMDL(K)
C
C *** SETS UP THE SELECTION CODE TABLE ENTRIES
C ***       FIRST WORD IS THE SELECTION CODE
C ***       SECOND WORD  BYTE 1 IS ZERO
C ***                    BYTE 2 IS DENSITY AND CHANNEL NUMBER
C ***                    BYTE 3 IS DELETE FLAG
C
      DIMENSION ISC(1000,2)
```

```
CARD NO  ****                    CONTENTS                    ****
    839        COMMON /A1/ ISC,ITMAX,ITCUP,MINSC,MAXSC
    840        COMMON /B1/ INSC,INCH,INDEN,INDEL
    841  C
    842  C *** CLEAR TABLE ENTRY
    843  C
    844        ISC(K,1) = 0
    845        ISC(K,2) = 0
    846  C
    847  C *** SET FIRST WORD
    848  C
    849        ISC(K,1) = INSC
    850  C
    851  C *** FORM SECOND BYTE OF SECOND WORD
    852  C
    853        ITEMP = 0
    854        CALL LS(INDEN,3,ITEMP)
    855        CALL LO(ITEMP,INCH,ITEMP)
    856        CALL LFOR(ITEMP,4,ITEMP)
    857        CALL LS(ITEMP,6,ITEMP)
    858        ITEMP = ITEMP + INDEL
    859  C
    860  C *** SET SECOND WORD
    861  C
    862        ISC(K,2) = ITEMP
    863        RETURN
    864        END
    865        SUBROUTINE OUTWOL(K)
    866  C
```

```
C *** DECODE THE SELECTION CODE TABLE ENTRIES
C
      DIMENSION ISC(1000,2)
      COMMON /AL/ ISC,ITMAX,ITCUR,MINSC,MAXSC
      COMMON /HI/ IOSC,IOCH,IODEN,IODEL
C
C *** OBTAIN SELECTION CODE
C
      IOSC = ISC(K,1)
C
C *** STRIP SECOND WORD
C
      ITEMP = ISC(K,2)
      CALL LA(ITEMP,1,IODEL)
      CALL RS(ITEMP,4,ITEMP)
      CALL LA(ITEMP,56,IODEN)
      IOCH = ITEMP - IODEN
      CALL RS(IODEN,3,IODEN)
      IOCH = IOCH - 4
      RETURN
      END
      SUBROUTINE PBYTE(IW,IB,IOW,ICR)
C
C *** SHIFTS BYTES FROM A GIVEN LOCATION IN ONE WORD TO A GIVEN
C *** LOCATION IN ANOTHER WORD
C ***          IW = INPUT WORD
C ***          IB = BYTE LOCATION OF INPUT WORD TO BE SHIFTED -  1,2 OR 3
C ***         IOW = OUTPUT WORD
```

```
CARD NO    ****                    CONTENTS                    ****
     895   C ***        IOB = BYTE LOCATION TO BE SHIFTED INTO IN OUTPUT WORD-1,2,3
     896         DIMENSION MASK1(3),MASK2(3),MASKS(6)
     897         COMMON /MASK/ MASK1,MASK2
     898         EQUIVALENCE (MASKS(1),MASK1(1))
     899   C
     900         CALL LA(IW,MASK1(IB),IWRK1)
     901         CALL LA(IOW,MASK2(IOB),IWRK2)
     902         IX = 6*(IB - IOB)
     903         IF (IX) 1,2,3
     904       1 IX = -IX
     905         CALL RS(IWRK1,IX,IWRK1)
     906       2 CALL LO(IWRK1,IWRK2,IOW)
     907         RETURN
     908       3 CALL LS(IWRK1,IX,IWRK1)
     909         GO TO 2
     910         END
     911         SUBROUTINE IMESG
     912   C
     913   C *** WRITES INPUT TAPE MESSAGES
     914   C
     915         COMMON /E1/ IN,IA,IE,IX,IY,IO,IP,IT,IS,IR,IM
     916         COMMON /F1/ ITTY,IPT,IDT,INP,IPRTR
     917         COMMON /G1/ IDR1,IDR2,IDR3,IDEC
     918         COMMON /J1/ IMGT1,IMGT2,IDONE
     919   C
     920   C *** MORE TAPE MESSAGE
     921   C
     922         WRITE (ITTY,1018) IMGT1
```

```
923          '018 FORMAT (54H ARE THERE MORE INPUT TAPES FOR THIS RUN IN ADDITION T,
924              16HO THAT/ 27H CURRENTLY MOUNTED ON DRIVE,1X,I1,15H   ENTER Y OR N,
925          C
926          C *** READ ANSWER
927          C
928                READ (ITTY,2015) ANS
929           2015 FORMAT (A1)
930          C
931          C *** SET DONE FLAG IF NO
932          C
933                IDONE = 0
934                IF (ANS .NE. TY) IDONE = 1
935          C
936          C *** CHECK IF DONE
937          C
938                IF (IDONE .EQ. 1) GO TO 1
939          C
940          C *** MOUNT NEXT TAPE MESSAGE
941          C
942                WRITE (ITTY,1019) IMGT2
943           1019 FORMAT (32H MOUNT NEXT INPUT TAPE ON DRIVE ,I1)
944          C
945          C *** PROPER TAPE MOUNTED MESSAGE
946          C
947              1 WRITE (ITTY,1020) IMGT1
948           1020 FORMAT (50H ENTER 0 IF PROPER INPUT TAPE IS MOUNTED ON DRIVE ,I1)
949                READ (ITTY,2015) DON
950                IF (DON .NE. TD) GO TO 1
```

```
CARD NO      ****                    CONTENTS                    ****
     951     C
     952     C *** PROPER TAPE MOUNTED - RESET DRIVE NUMBERS AND THEN RETURN
     953     C
     954           IF (IMGT1 .EQ. IDR3) GO TO 2
     955           IMGT1 = IDR3
     956           IMGT2 = IDR2
     957           RETURN
     958         2 IMGT1 = IDR2
     959           IMGT2 = IDR3
     960           RETURN
     961           END
     962           SUBROUTINE NPAGE
     963     C
     964     C *** OUTPUTS THE HEADING INFORMATION FOR THE SELECTION CODE LIST
     965     C
     966           COMMON /F1/ ITTY,IPT,IDT,INP,IPRTR
     967           COMMON /I1/ IMONTH,IDAY,IYEAR,IPAGE,IFILE,IREV
     968     C
     969           WRITE (IPRTR,1013) IMONTH,IDAY,IYEAR,IPAGE
     970      1013 FORMAT (47H1    RPS SELECTION CODE TABLE - INPUT SUMMARY  ,I2,1H/,
     971          1I2,1H/,I2,10H     PAGE ,I2)
     972           WRITE (IPRTR,1014) IDT
     973      1014 FORMAT (14HO TAPE NUMBER ,I5)
     974           WRITE (IPRTR,1015) IFILE
     975      1015 FORMAT (14H  FILE NUMBER ,I2)
     976           WRITE (IPRTR,1016) IREV
     977      1016 FORMAT (14H  REVISION    ,I2)
     978           WRITE (IPRTR,1017)
```

```
 1017 FORMAT (53HO SELECTION CODE      CHANNEL      DENSITY      OFLET,
     16HE FLAG)
      RETURN
      END
      SUBROUTINE PRNSTY
C
C *** PRINT RUN STATISTICS
C
      DIMENSION ICNT(25),ICH(4),ITBUF(740)
      COMMON /K1/ ICNT,ICH
      COMMON /TBUF/ ITBUF,NYS,NXS
      COMMON /L1/ MIN,ISEC,IOFF
      COMMON /F1/ ITTY,IPT,IDT,INP,IPRTR
      COMMON /I1/ IMONTH,IDAY,IYEAR,IPAGE,IFILE,IREV
C
      WRITE (IPRTR,1300) IMONTH,IDAY,IYEAR
 1300 FORMAT (24H1 RPS RUN STATISTICS OF ,I2,1H/,I2,1H/,I2)
      WRITE (IPRTR,1301) IDT
 1301 FORMAT (14HO TAPE NUMBER ,I5)
      WRITE (IPRTR,1302) IFILE
 1302 FORMAT (14H  FILE NUMBER ,I2)
C
C     STATISTICS
C
      WRITE (IPRTR,1303) (MIN,ISEC,NYS,NXS,(K,ICNT(K),K=1,25)
 1303 FORMAT(/15H RUN STATISTICS//16H ELAPSED TIME = ,I6,1X,12HMINUTES A
     1ND ,I6,1X,7HSECONDS////I6,1X,27HSCAN LINES INPUT CONTAINING,1X,I6,
     214H X COORDINATES////38H X DESIGNATOR OF POINTS PROCESSED WERE////2
```

```
CARD NO  ****                    CONTENTS                    ****
1007          35X,4HCODE,12X,5HCOUNT,25(/25X,I4,12X,I6))
1008           WRITE (IPRTR,1304) (ICH(K),K=1,4)
1009         1304 FORMAT (30H0CHANNEL TOTALS:  CHANNEL 0 = ,I6,1X,13H CHANNEL 1 = ,I
1010          16,1X,13H CHANNEL 2 = ,I6,1X,13H CHANNEL 3 = ,I6)
1011           RETURN
1012           END
1013           SUBROUTINE NSET
1014         C
1015         C *** RESETS OUTPUT BUFFER POINTERS AND CLEARS THE BUFFER TO THE HIGH
1016         C *** IGNORE VALUE
1017           DIMENSION INSET(4),IOBUF(1364)
1018           COMMON /M1/ INSET,LSTWRD,LSTBYT,NENT,IOBUF
1019         C
1020           NENT = 1
1021           LSTBYT = 1
1022           LSTWRD = 2
1023           DO 1 J = 1,756,4
1024           DO 2 K = 1,4
1025           KK = J + K - 1
1026         2 IOBUF(KK) = INSET(K)
1027         1 CONTINUE
1028           RETURN
1029           END
1030           SUBROUTINE SETA
1031         C
1032         C ***
1033         C
1034           DIMENSION MASK1(3),MASK2(3)
```

```
      COMMON /GNR1/ ITYPE,ICORD,ICON1,ICON2,NX,IX,IFT,ISTAT
      COMMON /AFLC/ IMINF,IMNAR
      COMMON /MASK/ MASK1,MASK2
      COMMON /IOUT/ IOUT1,IOUT2,IATST
      EQUIVALENCE (MASK13,MASK1(3))
C
      IMINF = 0
      IF (IATST .GT. 3) GO TO 2
      GO TO (1,2,3),IATST
    2 CONTINUE
      RETURN
    1 CALL LEDR(256,IOUT1,IOUT1)
      RETURN
    3 CALL LA(MASK13,ICON2,IWRK)
      CALL LEDR(IWRK,ICON2,ICON2)
      ICORD = ICORD + IMNAR
      IMINF = 1
      GO TO 1
      END
      BLOCK DATA
      COMMON /E1/ TN,TA,TE,TX,TY,TD,TP,TT,TS,TR,TM
      DATA TN,TA,TE/5HN    ,5HA    ,5HE    /
      DATA TX,TY,TD,TP/5HX    ,5HY    ,5HD    ,5HP
      DATA TT,TS,TR,TM/5HT    ,5HS    ,5HR    ,5HM
      END
      SUBROUTINE SETM (I)
C
C     THIS IS AN ASSEMBLY LANGUAGE ROUTINE
```

```
CARD NO  ****  CONTENTS                                  ****
1063     C
1064           RETURN
1065           END
1066           SUBROUTINE SETC (I)
1067     C
1068     C     THIS IS AN ASSEMBLY LANGUAGE ROUTINE
1069     C
1070           RETURN
1071           END
10 2           SUBROUTINE LD (I,J,K)
1073     C
1074     C     THIS IS AN ASSEMBLY LANGUAGE ROUTINE
1075     C
1076           RETURN
1077           END
1078           SUBROUTINE LEOR (I,J,K)
1079     C
1080     C     THIS IS AN ASSEMBLY LANGUAGE ROUTINE
1081     C
1082           RETURN
1083           END
1084           SUBROUTINE LS (I,J,K)
1085     C
1086     C     THIS IS AN ASSEMBLY LANGUAGE ROUTINE
1087     C
1088           RETURN
1089           END
1090           SUBROUTINE RS (I,J,K)
```

```
      C
      C     THIS IS AN ASSEMBLY LANGUAGE ROUTINE
      C
            RETURN
            END
            SUBROUTINE MGERW (I,J,K,L,M)
      C
      C     THIS IS AN ASSEMBLY LANGUAGE ROUTINE
      C
            RETURN
            END
            SUBROUTINE REOF (I)
      C
      C     THIS IS AN ASSEMBLY LANGUAGE ROUTINE
      C
            RETURN
            END
            SUBROUTINE TIME(I,J,K)
      C     THIS IS A SYSTEM SUBROUTINE
            RETURN
            END
            SUBROUTINE LA(I,J,K)
      C     THIS IS AN ASSEMBLY LANGUAGE SUBROUTINE
            RETURN
            END
```

PG.BX	NAME	PG.BX	NAME	PG.BX	NAME	PG.BX	NAME	PG.BX	NAME
19.07	GNP	26.01	SRCH	20.13	2	6.07	310	9.17	401
34.01	IMFSG	54.02	TIMF	32.07	2	6.06	311	10.01	402
28.01	INTWDL	32.05	1	34.20	2	6.05	312	10.03	403
55.02	LA	26.02	1	42.04	2	7.01	313	10.08	404
49.02	LFOR	42.06	1	26.09	2	7.05	314	10.13	405
48.02	LO	2.14	1	3.21	200	6.17	315	10.24	406
50.02	LS	22.06	1	3.19	21	8.12	316	19.16	5
52.02	MGFRW	19.14	1	24.12	3	8.31	317	24.06	5
16.01	MKENT	40.07	1	19.10	3	6.21	318	3.13	50
36.01	NPAGE	24.09	1	42.08	3	6.24	319	2.22	500
40.01	NSET	34.12	1	26.11	3	7.02	320	17.18	6
30.01	OUTWDL	16.14	1	32.09	3	8.17	350	19.01	6
32.01	PBYTE	13.19	10	4.08	300	5.14	360	24.02	6
38.01	PRNSTT	14.01	100	4.12	301	7.03	370	8.05	600
53.02	RECS	13.10	11	4.18	302	7.04	380	20.03	7
51.02	RS	14.18	14	5.09	303	4.01	390	17.16	7
27.01	SCAN	14.13	15	5.03	304	8.16	391	3.08	700
13.01	SCODE	14.23	16	6.12	305	24.13	4	9.04	7000
42.01	SETA	24.16	2	5.01	306	19.19	4	3.11	701
47.02	SETC	13.06	2	6.20	307	16.19	4	17.23	8
46.01	SETM	17.01	2	6.09	308	26.07	4	12.09	9
24.08	SHLSRT	40.05	2	6.01	309	9.01	400	9.24	999

CARD TO PAGE/BOX NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

FORTRAN MODULE RPSS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - PROCEDURES

(000237)	6.05	312	(000234)	6.03			
(000238)	6.06	311					
(000240)	6.07	310	(000228)	6.11			
(000221)	6.09	308	(000218)	6.19			
(000210)	6.12	305	(000190)	5.09	(000297)	8.15	
(000211)	6.14		(000210)	6.12			
(000217)	6.17	315	(000259)	6.27			
(000250)	6.20	307	(000212)	6.15			
(000253)	6.21	318	(000303)	4.04	(000267)	7.04	
(000257)	6.24	319					
(000257)	6.24		(000257)	6.25			
(000243)	7.01	313	(000241)	6.08			
(000244)	7.02	320	(000293)	8.13			
(000265)	7.03	370	(000156)	4.15			
(000261)	7.04	380	(000154)	4.14			
(000263)	7.05	314	(000241)	6.08			
(000268)	7.11		(000266)	7.08			
(000275)	8.01	317	(000241)	6.08			
(000282)	8.05	600	(000279)	8.03			
(000292)	8.12	316	(000269)	7.11	(000269)	7.1.	
(000305)	8.16	391	(000292)	8.12			
(000309)	8.17	350					
(000313)	9.01	400	(000305)	8.16			
(000319)	9.04	7000	(000316)	9.02			
(000343)	9.17	401	(000331)	9.11			
(000407)	9.24	999	(000358)	9.23			
(000363)	10.01	402	(000338)	9.15			
(000368)	10.03	03	(000359)	9.23			

CARD ID PAGE/BOX NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

(000398) 10.09 404 (000369) 10.04 (000377) 10.10
(000389) 10.13 405 (000381) 10.11 (000391) 10.17
(000396) 10.24 406 (000385) 10.12

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE SCOPE

(000416) 13.01 SCOPE (000122) 3.19-X
(000449) 13.06 2
(000449) 13.06 (000449) 13.07
(000458) 13.09 9 (000466) 13.15 (000533) 14.22
(000461) 13.10 11 (000483) 13.21
(000481) 13.19 10 (000482) 13.20 (000484) 13.22 (000487) 13.24
(000487) 13.26 (000486) 13.24
(000491) 14.01 100 (000464) 13.13 (000495) 13.23
(000501) 14.06 (000511) 14.13
(000511) 14.13 15 (000507) 14.10
(000521) 14.18 14 (000515) 14.14 (000535) 14.24
(000534) 14.23 16 (000529) 14.21

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE MKENT(IFLG)

(000543) 16.01 MKENT (000481) 13.19-X
(000590) 16.13 (000589) 16.11
(000572) 16.14 1 (000565) 16.04

(000649) 16.19 4 (000577) 16.08 (000572) 16.15 (000574) 16.15 (000575) 16.17 (000576) 16.18

(000594) 17.01 2 (000583) 16.09

(000597) 17.04 (000595) 17.02

(000602) 17.07 (000600) 17.05

(000608) 17.10 (000606) 17.08

(000611) 17.11 (000633) 17.18

(000625) 17.16 7 (000616) 17.13

(000633) 17.13 6

(000636) 17.21 (000634) 17.19

(000641) 17.23 8 (000629) 17.17

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE GNR

(000727) 19.01 6 (000674) 19.13 (000712) 19.25

(000729) 19.04 (000728) 19.02

(000661) 19.07 GNR (000143) 4.00-X

(000673) 19.10 3 (000695) 19.14

(000695) 19.14 1 (000648) 19.07

(000695) 19.14 (000693) 20.11

(000697) 19.16 5 (000714) 19.26

(000708) 19.19 4 (000696) 19.15

(000712) 19.25 (000711) 19.22

(000680) 20.03 7 (000721) 20.21

(000685) 20.07 (000683) 20.04

(000693) 20.11 (000692) 20.09

(000716) 20.13 2 (000677) 20.01

07/22/71 TABLE OF CONTENTS AND REFERENCES AUTOFLOW CHART SET -

CARD ID PAGE/BOX NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

(000799) 26.01 SRCH (000174) 4.22-X

(000805) 26.02 1 (000813) 26.06 (000818) 26.12

(000827) 26.07 4 (000805) 26.02

(000822) 26.09 2 (000807) 26.04

(000814) 26.11 3 (000808) 26.05

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE INTWDL(K)

(000832) 28.01 INTWDL (000449) 13.06-X (000621) 17.16-X

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE OUTWDL(K)

(000847) 30.01 OUTWDL (000503) 14.04-X

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE PRYTE(IW,IR,IOW,IOR)

(000890) 32.01 PRYTE (000067) 2.09-X (000237) 6.05-X (000221) 5.09-X (000255) 4.22-X (000257) 4.24-X
(000258) 6.26-X

(000904) 32.05 1

(000906) 32.07 2 (000903) 32.04 (000909) 32.10

(000908) 32.09 3 (000903) 32.04

CARD ID PAGE/BOX NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE IMESG

(000913) 34.01 IMESG (000128) 3.22-X (000146) 4.16-X
(000936) 34.08 (000934) 34.06
(000947) 34.12 1 (000938) 34.09 (000950) 34.16
(000958) 34.20 2 (000954) 34.17

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE NPAGE

(000944) 36.01 NPAGE (000494) 14.03-X (000510) 14.12-X

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE PRNSTT

(000985) 38.01 PRNSTT (000322) 9.06-X

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE NSET

(001015) 40.01 NSET (000129) 3.23-X (000244) 7.02-X (000294) 8.14-X
(001024) 40.03 (001027) 40.07
(001025) 40.04 (001026) 40.06

(001026) 40.05 2

(001027) 40.07 1

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE SETA

(001032) 42.01 SETA (000210) 6.13-X

(001044) 42.04 2 (001042) 42.02 (001043) 42.03

(001046) 42.06 1 (001043) 42.03 (001052) 42.10

(001048) 42.08 3 (001043) 42.03

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - BLOCK DATA

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE SETM(I)

(001062) 46.01 SETM (000062) 2.07-X

CHART TITLE - SUBROUTINE SETC(I)

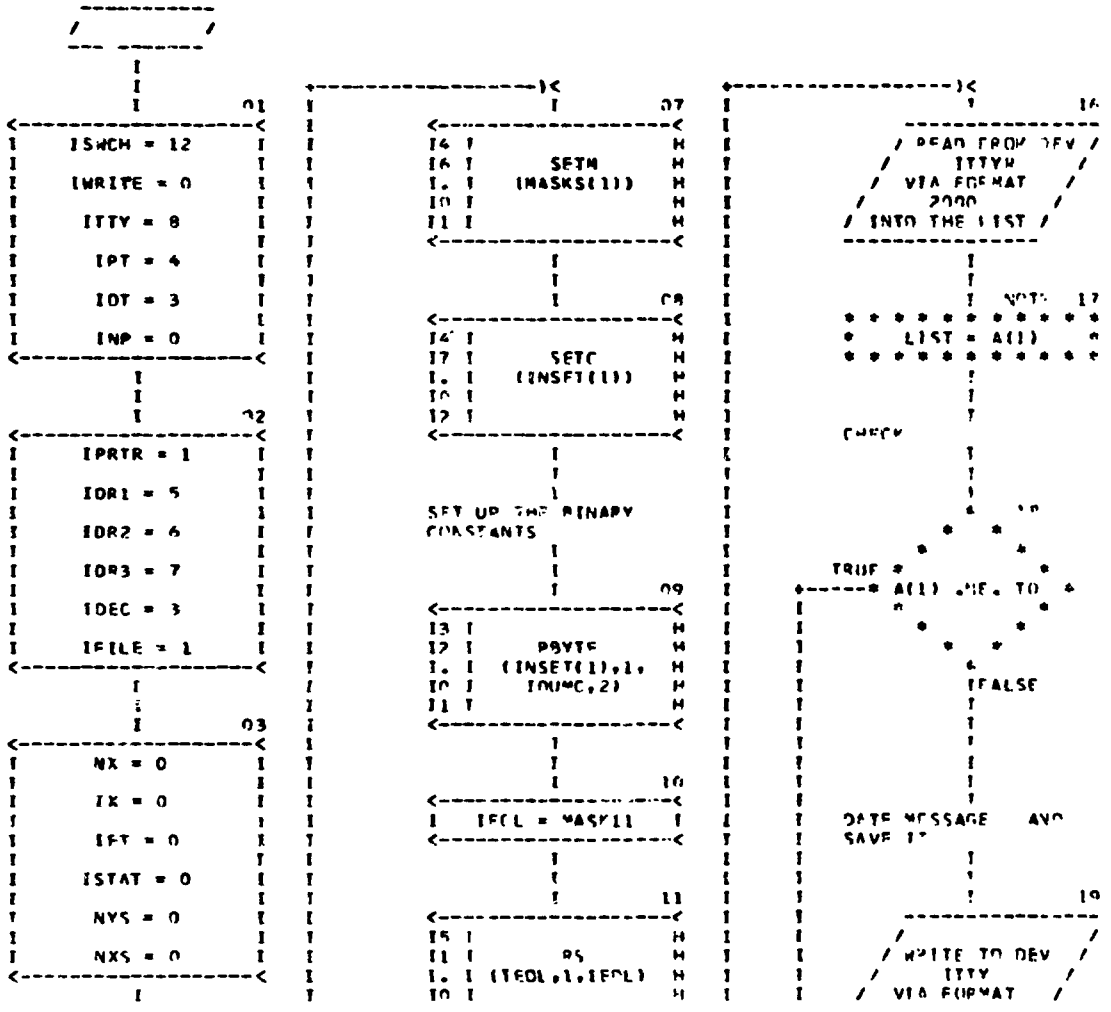
(001068) 47.02 SETC (000063) 2.08-X

CHART TITLE - SUBROUTINE LO(I,J,K)

CHART TITLE - INTRODUCTORY COMMENTS

INITIALIZATION

CHART TITLE - PROCEDURES



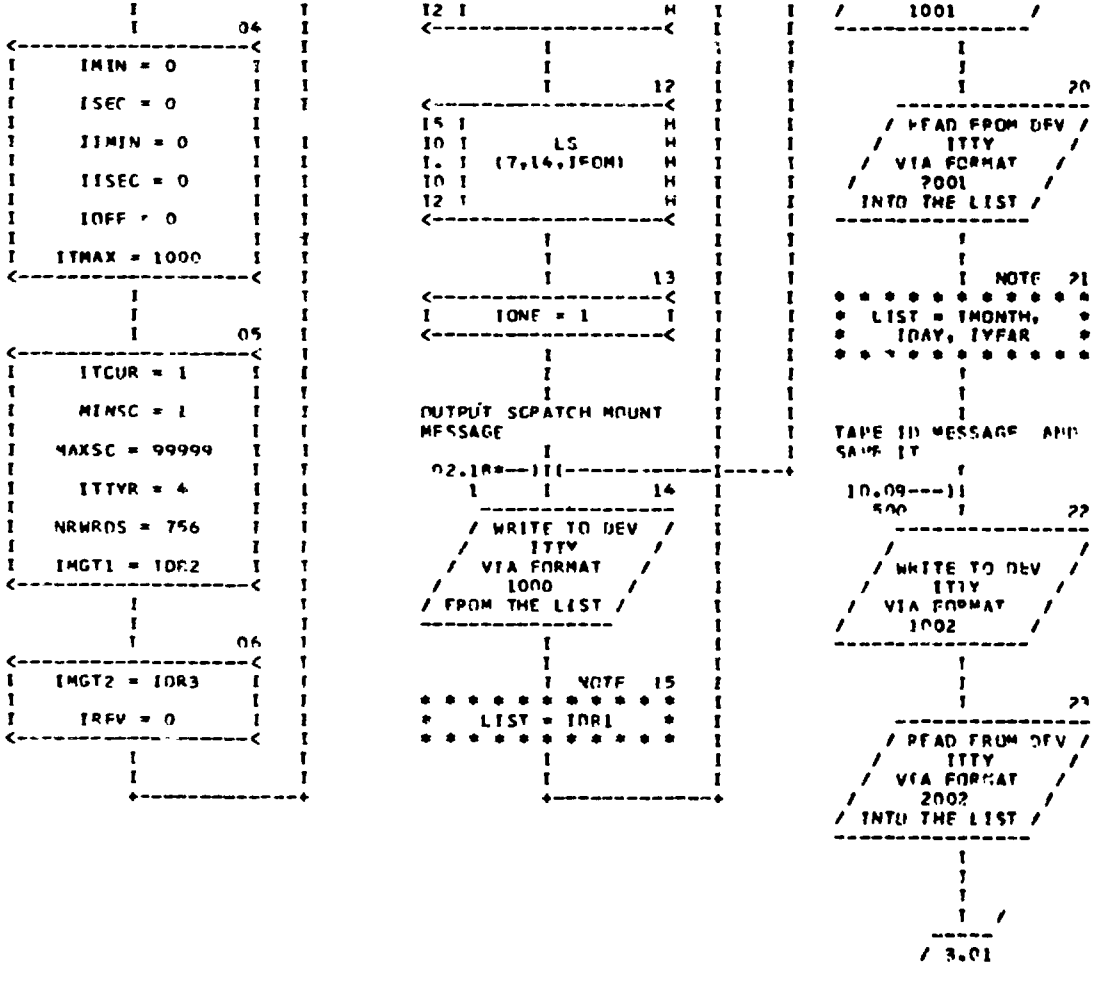
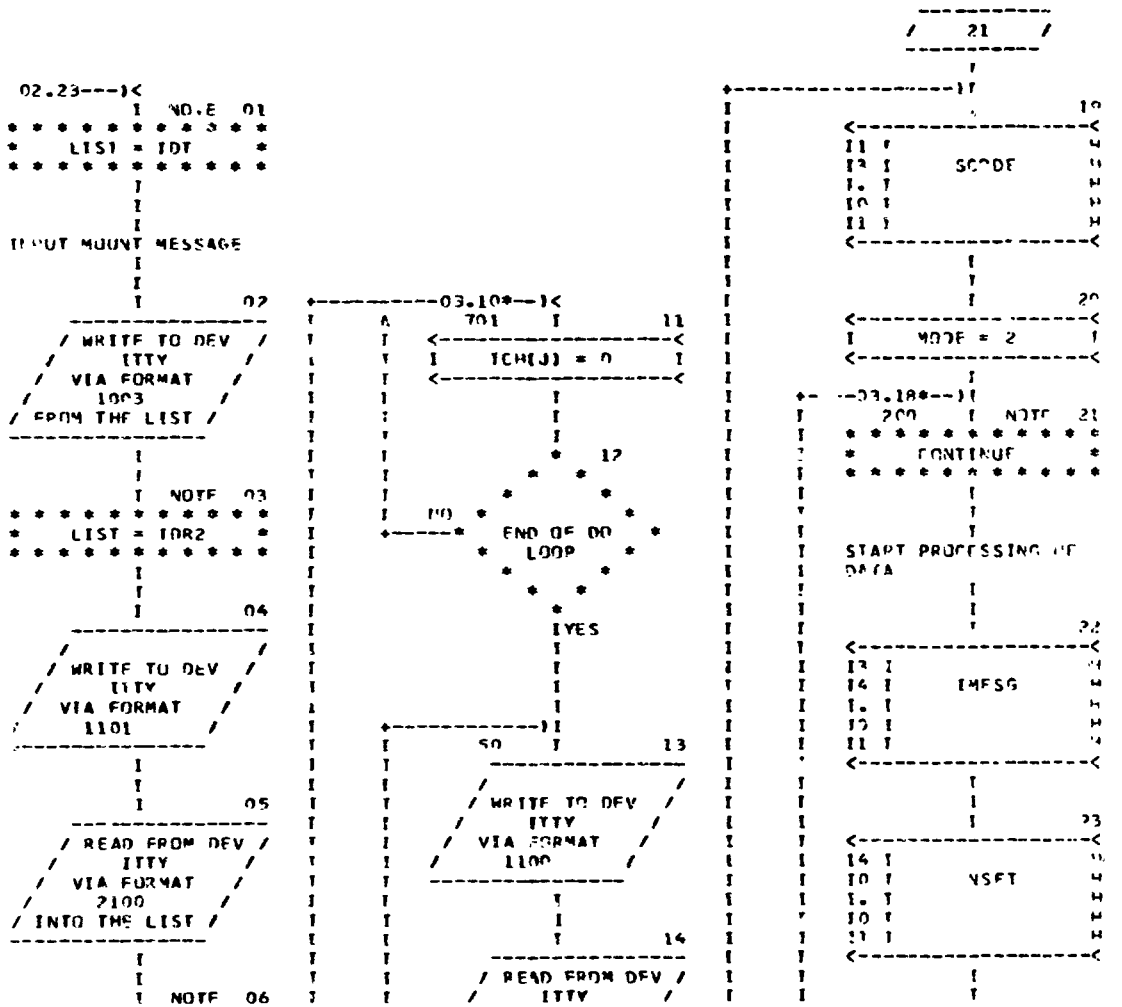

04
IMIN = 0
ISEC = 0
IIMIN = 0
IISEC = 0
IOFF = 0
ITMAX = 1000
05
ITCUR = 1
MINSC = 1
MAXSC = 99999
ITTYR = 4
NRWRDS = 756
IMGT1 = IDR2
06
IMGT2 = IDR3
IREV = 0
12
LS
(7,14,IFOM)
13
IONE = 1
OUTPUT SCRATCH MOUNT MESSAGE
14
WRITE TO DEV ITTY VIA FORMAT 1000 FROM THE LIST
NOTE 15
LIST = IDR1
1001
READ FROM DEV ITTY VIA FORMAT ?001 INTO THE LIST
NOTE 21
LIST = IMONTH, IDAY, IYEAR
TAPE ID MESSAGE AND SAVE IT
22
WRITE TO DEV ITTY VIA FORMAT 1002
23
READ FROM DEV ITTY VIA FORMAT 2002 INTO THE LIST
3.01

CHART TITLE - PROCEDURES



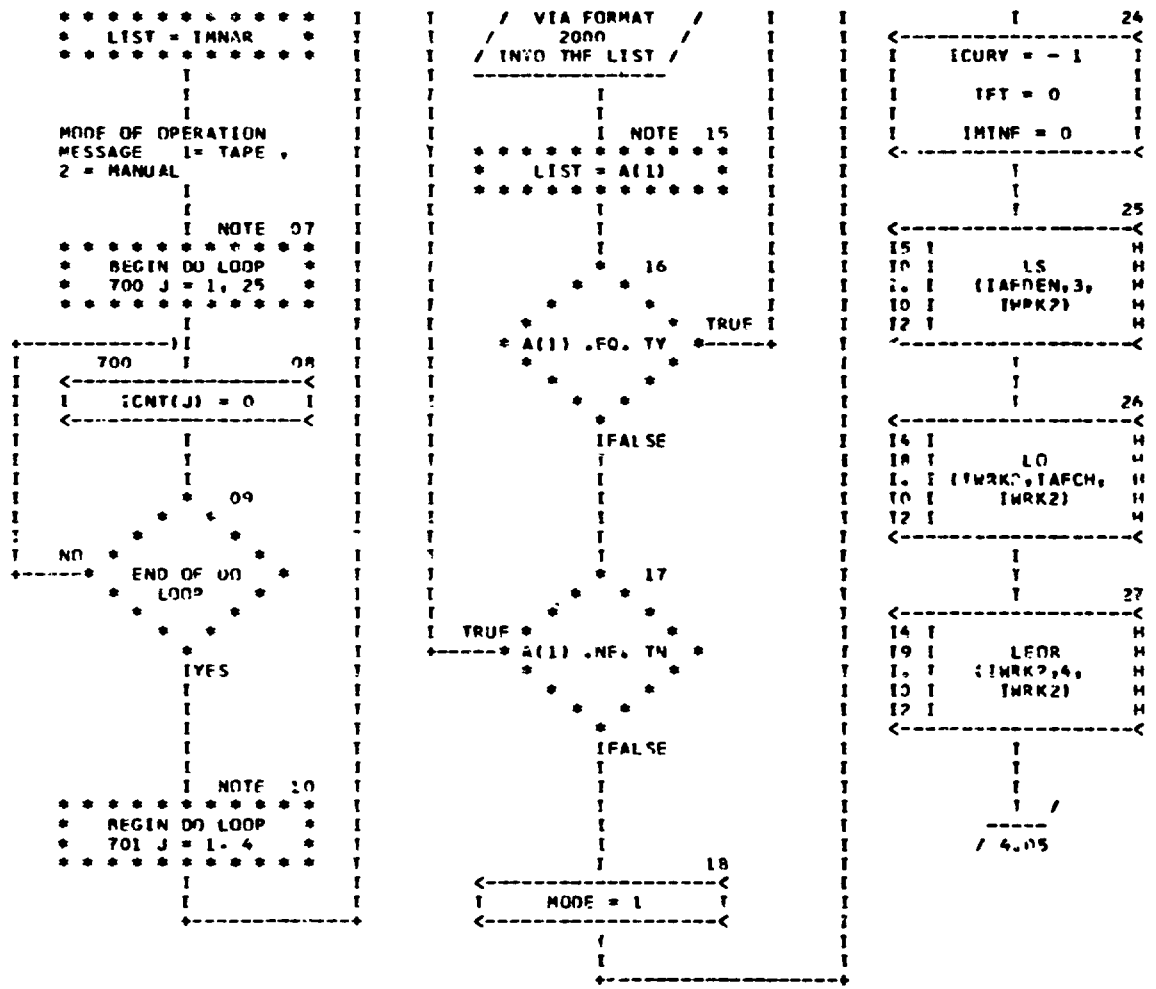
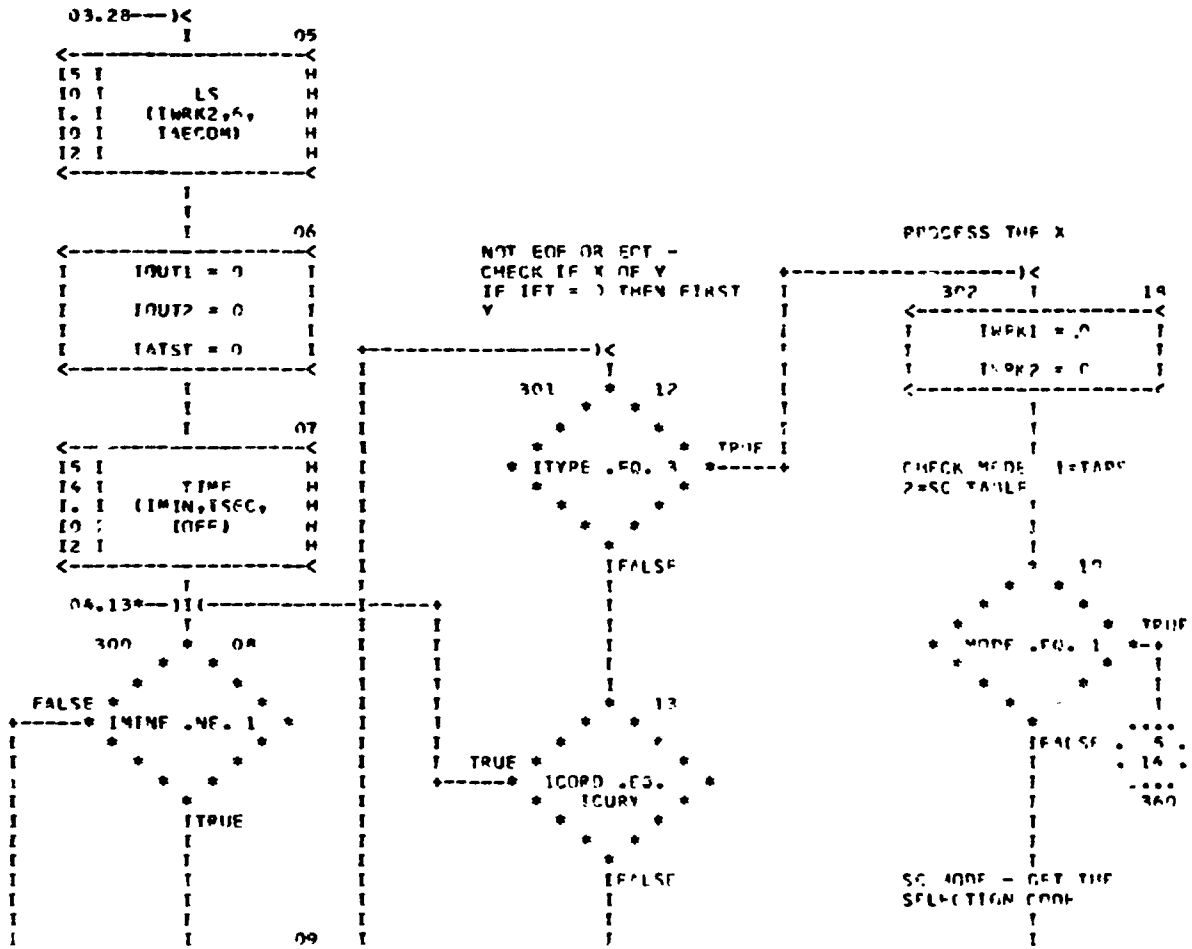

LIST = IMNAR
MODE OF OPERATION
MESSAGE 1= TAPE ,
2 = MANUAL
NOTE 07
BEGIN DO LOOP
700 J = 1, 25
700
08
ICNT(J) = 0
09
NO
END OF DO LOOP
YES
NOTE 10
BEGIN DO LOOP
701 J = 1, 4
/ VIA FORMAT /
/ 2000 /
/ INTO THE LIST /
NOTE 15
LIST = A(1)
16
A(1) .EQ. TY
TRUE
FALSE
17
A(1) .NE. TN
TRUE
FALSE
18
MODE = 1
24
ICURV = - 1
IFT = 0
IMINF = 0
25
LS
(IAFDEN,3,
IWRK2)
26
LO
(IWRK2,IAFCH,
IWRK2)
27
LEOR
(IWRK2,4,
IWRK2)
/ 4.05

CHART TITLE - PROCEDURES



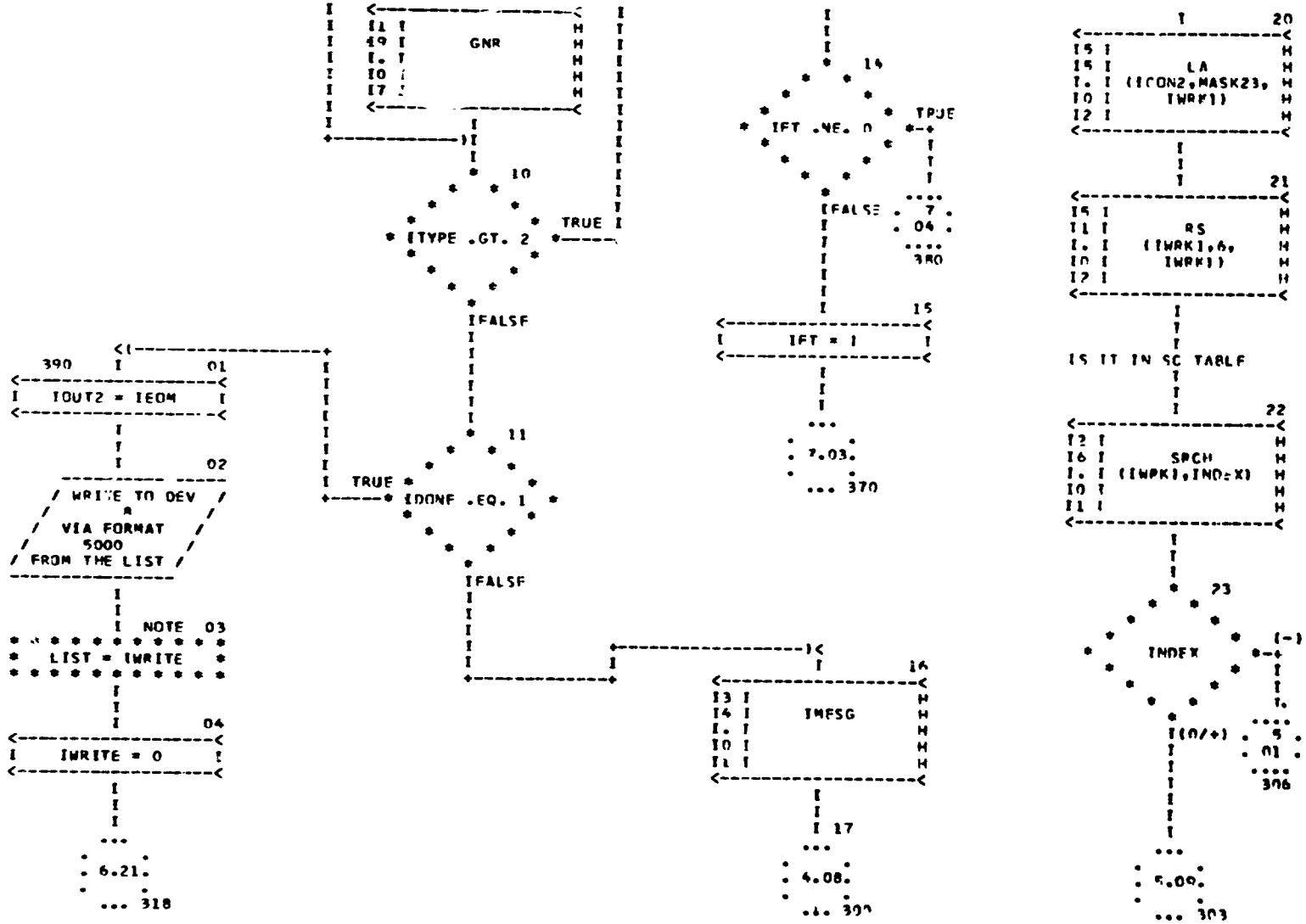
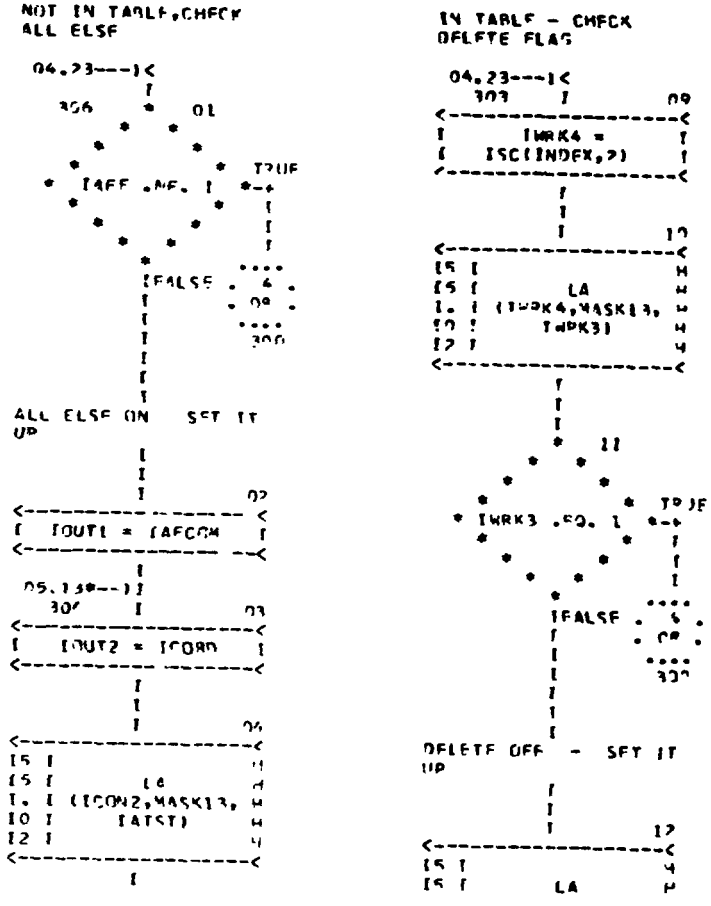

GNR
10
TYPE .GT. 2
TRUE
FALSE
11
IDONE .EQ. 1
TRUE
FALSE
14
IFT .NE. 0
TRUE
FALSE
7.04.
380
15
IFT = 1
7.03.
370
390
01
IOUT2 = IEOM
02
WRITE TO DEV
VIA FORMAT
5000
FROM THE LIST
NOTE 03
LIST = IWRITE
04
IWRITE = 0
6.21.
318
16
IMESG
17
4.08.
309
1
20
LA
(ICON2,MASK23,
IWRK1)
21
RS
(IWRK),6,
IWRK1)
IS IT IN SC TABLE
22
SRCH
(IWRK),IND=X)
23
INDEX
(-)
(0/+)
5.
01.
306
5.00.
303

CHART TITLE - PROCEDURES



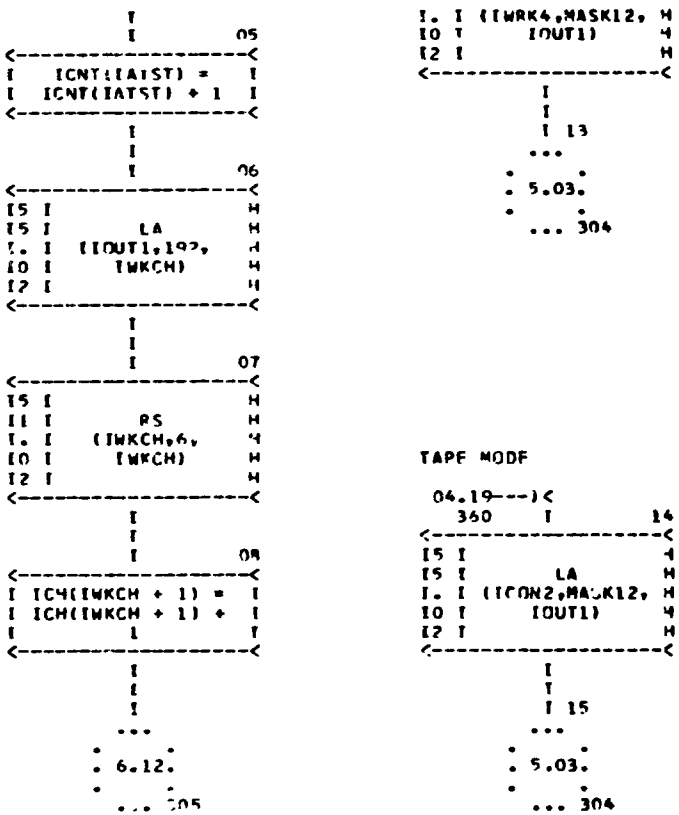
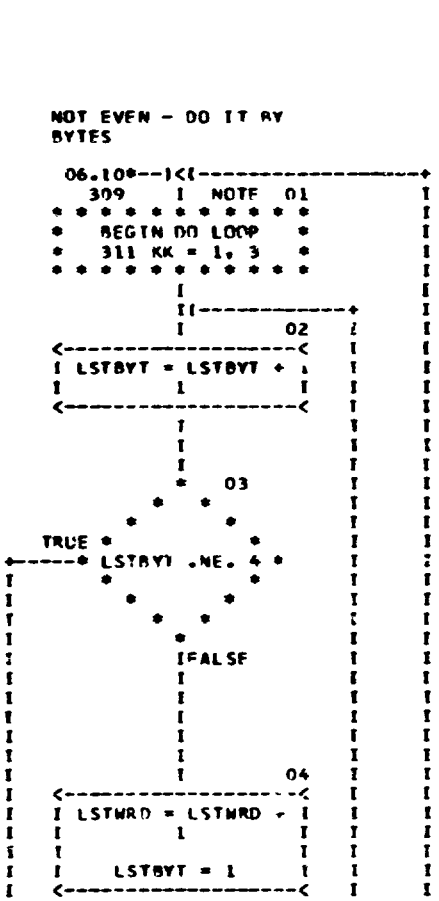
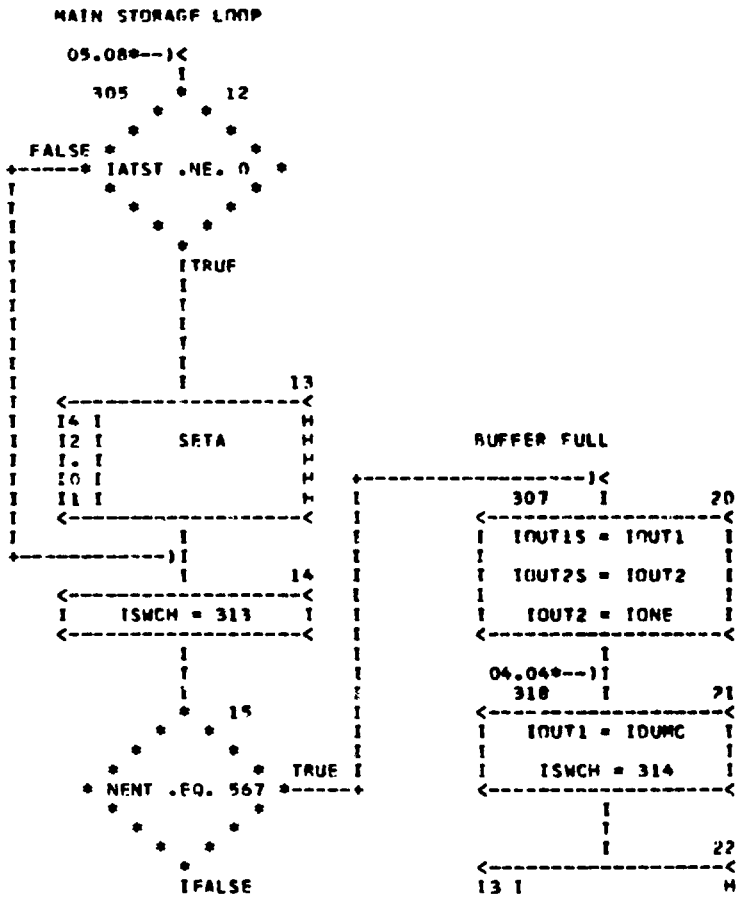

05
ICNT(IATST) =
ICNT(IATST) + 1
06
LA
(IOUT1,192,
IWKCH)
07
RS
(IWKCH,6,
IWKCH)
08
ICH(IWKCH + 1) =
ICH(IWKCH + 1) +
1
6.12.
205
WRK4,MASK12,
OUT1)
13
5.03.
304
TAPE MODE
04.19---)
360
14
LA
CON2,MASK12,
OUT1)
15
5.03.
304

CHART TITLE - PROCEDURES





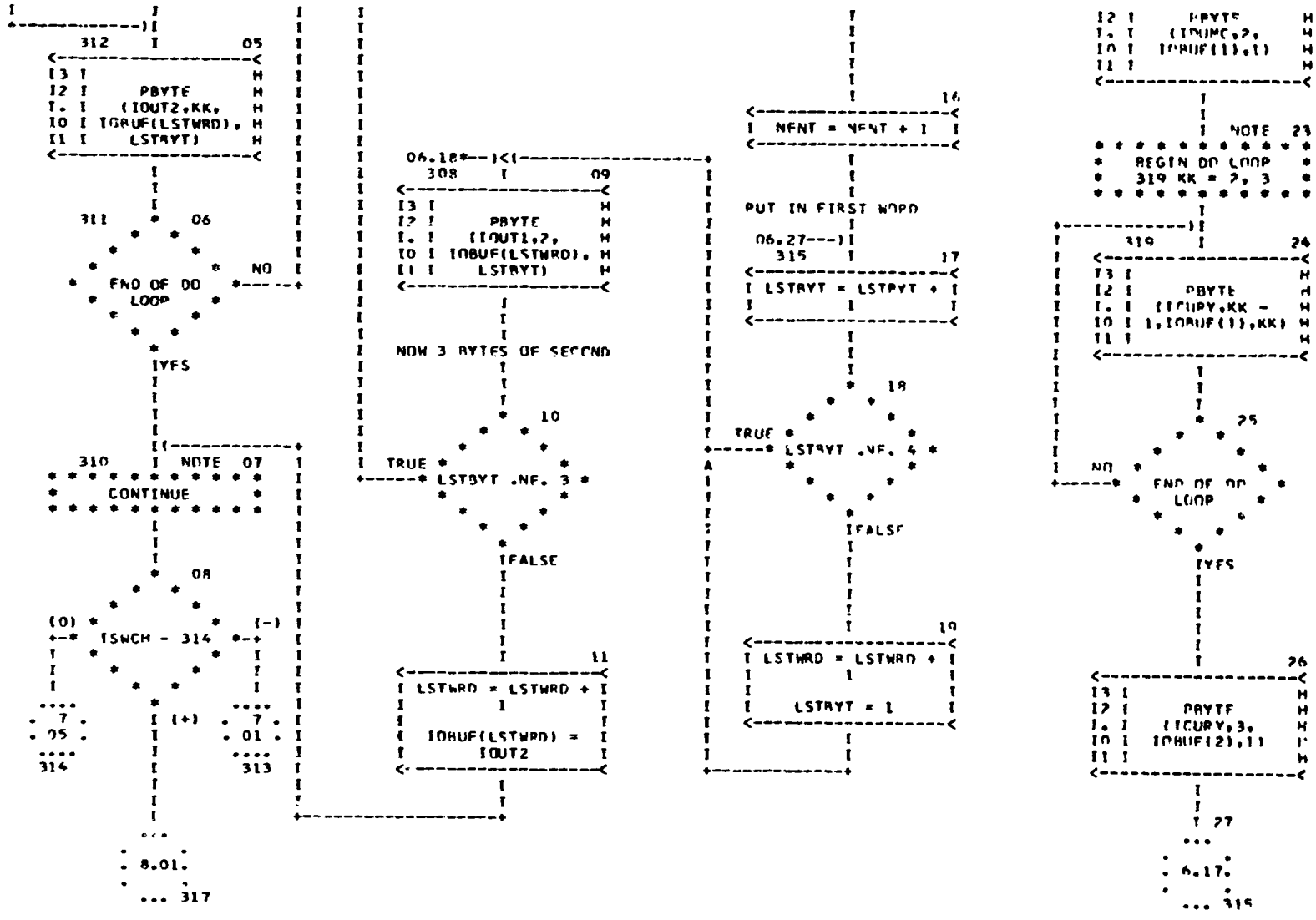
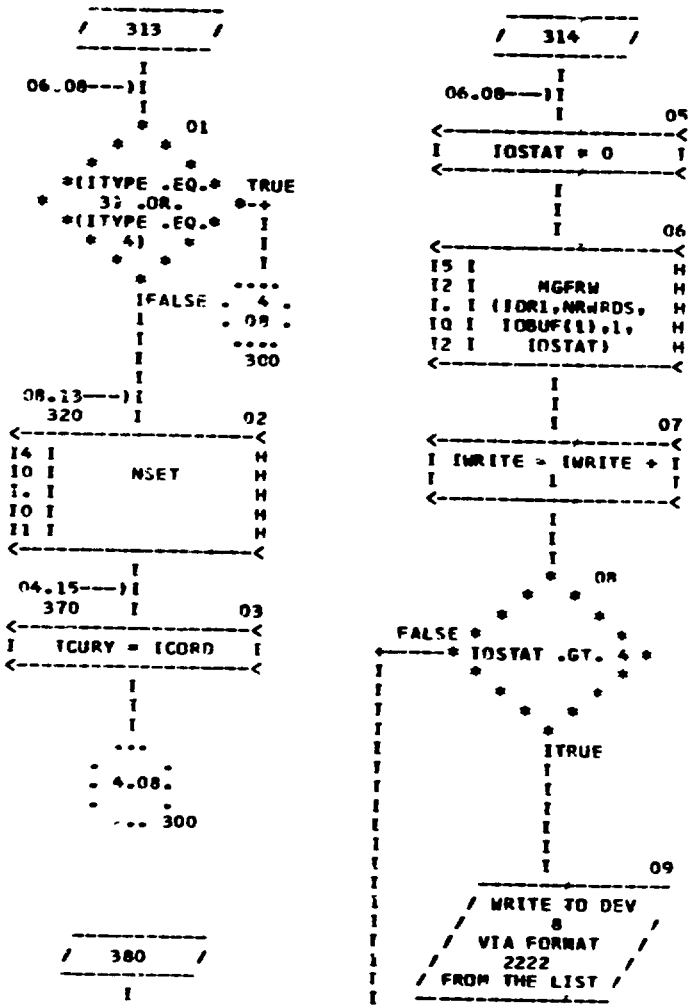

312
05
PBYTE
(IOUT2,KK,
IOBUF(LSTWRD),
LSTBYT)
311
06
NO
END OF DO
LOOP
YES
310
NOTE 07
CONTINUE
08
(0)
ISWCH - 314
(-)
(+)
7
05
314
7
01
313
8.01.
317
06.18
308
09
PBYTE
(IOUT1,2,
IOBUF(LSTWRD),
LSTBYT)
NOW 3 BYTES OF SECOND
10
TRUE
LSTBYT .NE. 3
FALSE
11
LSTWRD = LSTWRD + 1
IOBUF(LSTWRD) =
IOUT2
16
NENT = NENT + 1
PUT IN FIRST WORD
06.27
315
17
LSTBYT = LSTBYT + 1
18
TRUE
LSTBYT .NE. 4
FALSE
19
LSTWRD = LSTWRD + 1
LSTBYT = 1
PBYTE
(IOUMC,2,
IOBUF(1),1)
NOTE 23
BEGIN DO LOOP
319 KK = 2, 3
319
24
PBYTE
(IOUPY,KK -
1,IOBUF(1),KK)
25
NO
END OF DO
LOOP
YES
26
PBYTE
(IOUPY,3,
IOBUF(2),1)
27
6.17.
315

CHART TITLE - PROCEDURES



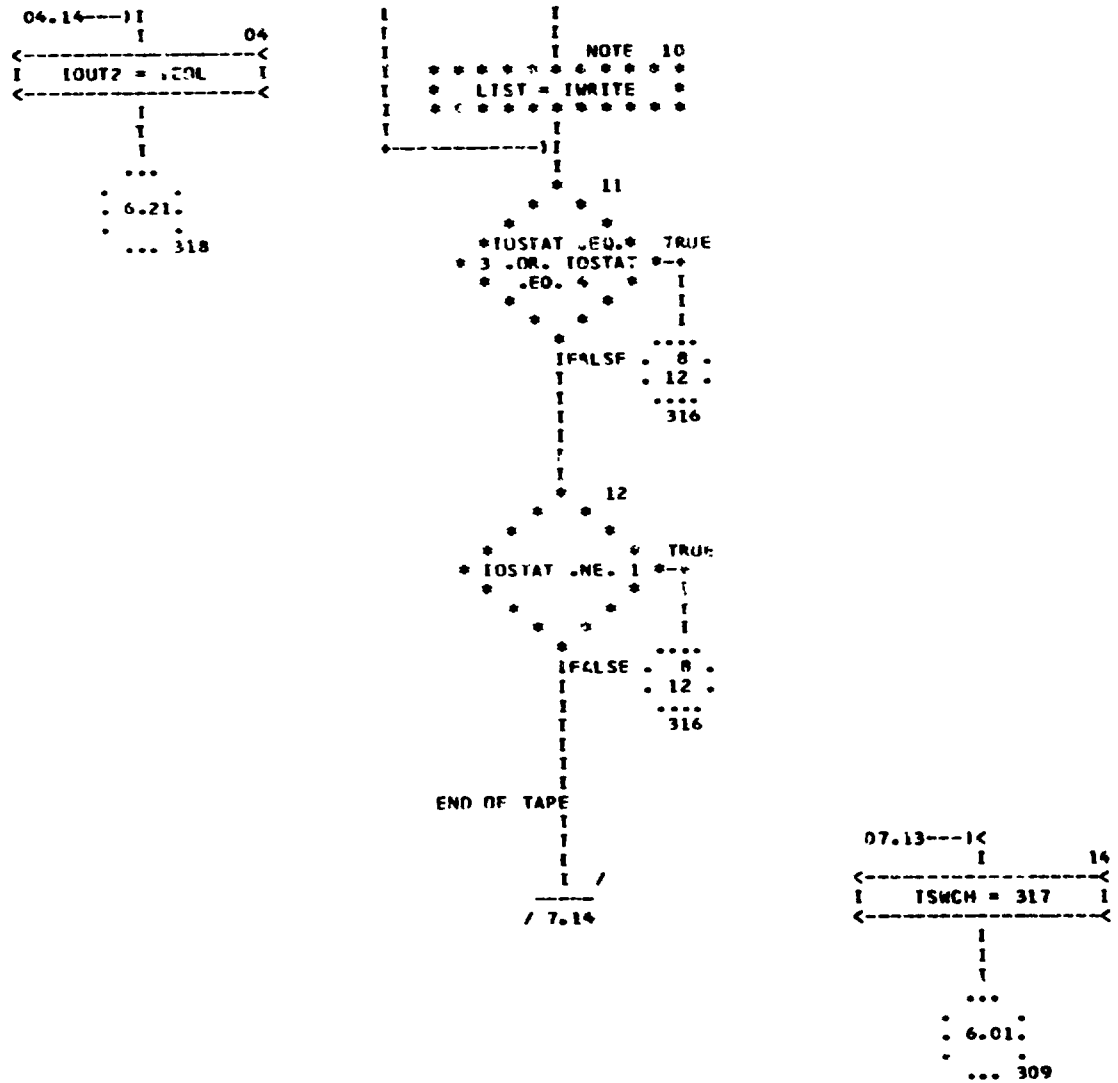
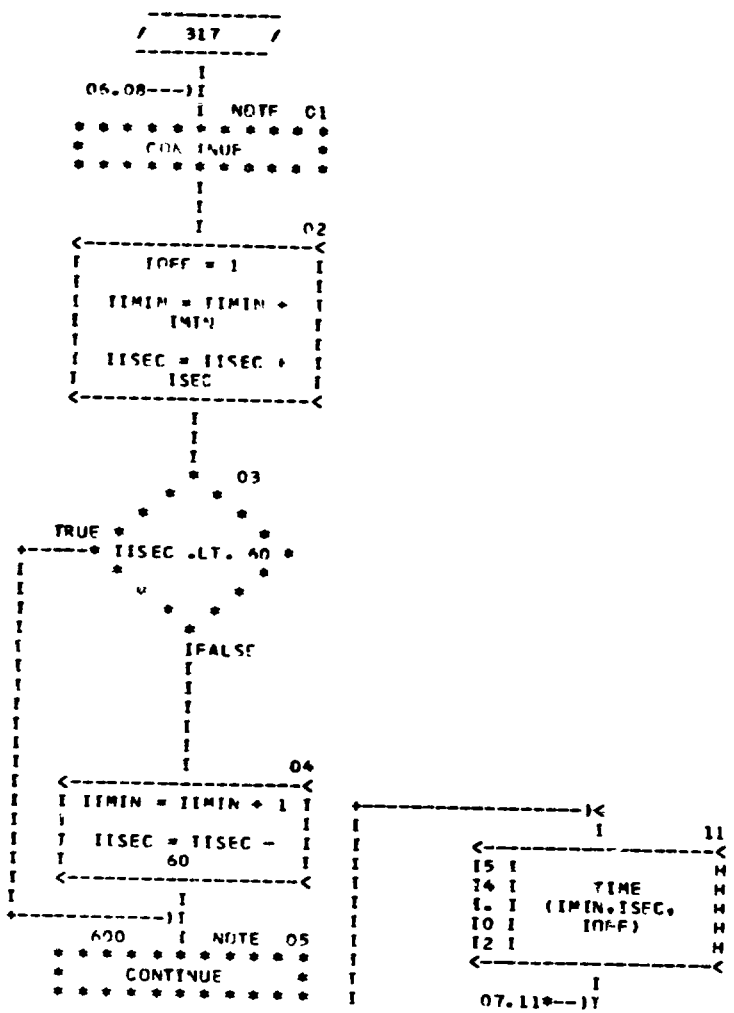
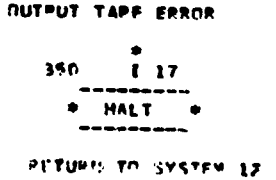

04.14
04
IOUT2 = ICOL
6.21.
... 318
NOTE 10
LIST = IWRITE
11
IOSTAT .EQ. 3 .OR. IOSTAT .EQ. 4
TRUE
IFALSE . 8 . . 12 .
316
12
IOSTAT .NE. 1
TRUE
IFALSE . 8 . . 12 .
316
END OF TAPE
/ 7.14
07.13
14
ISWCH = 317
6.01.
... 309

CHART TITLE - PROCEDURES



OUTPUT TAPE ERROR



RETURN TO SYSTEM 17

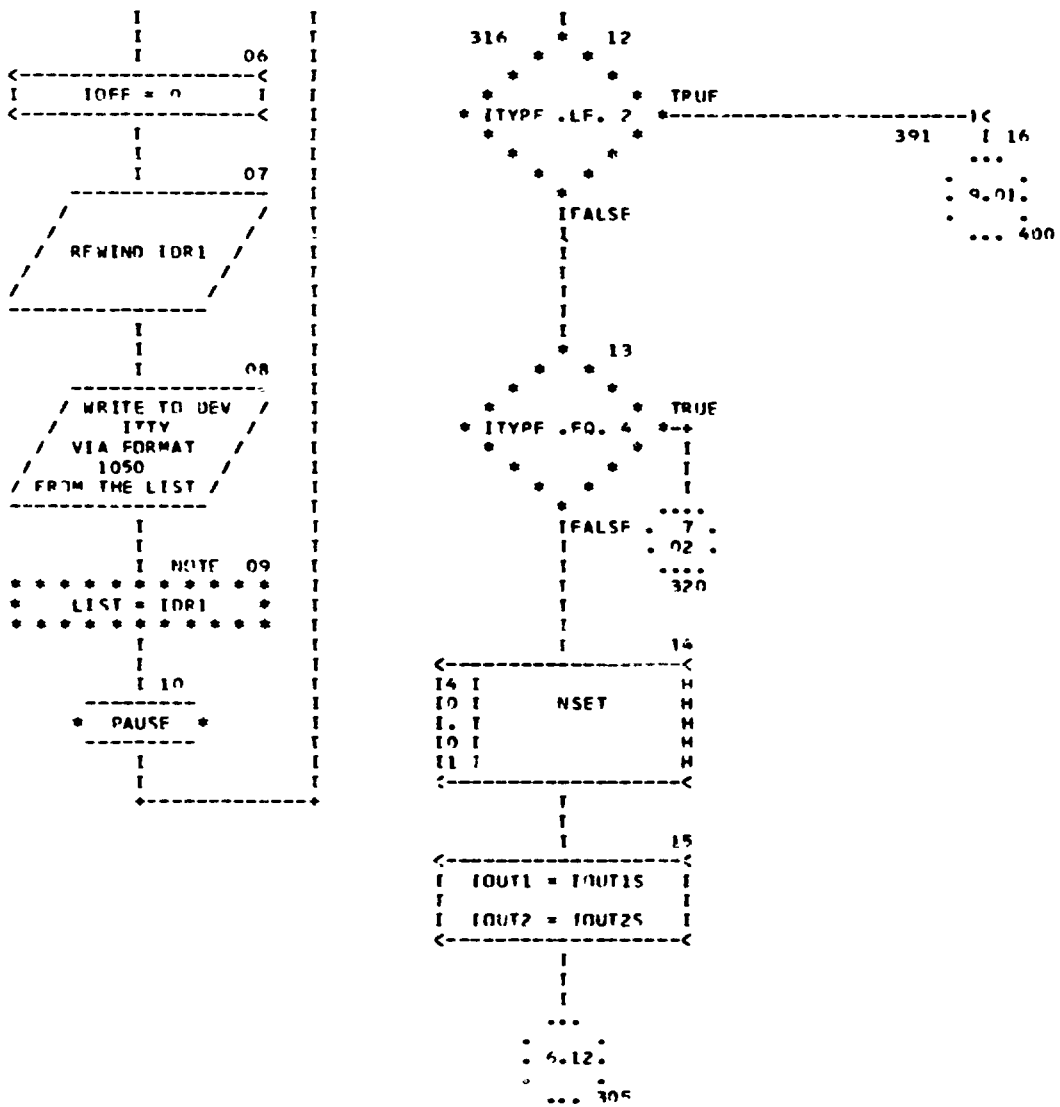
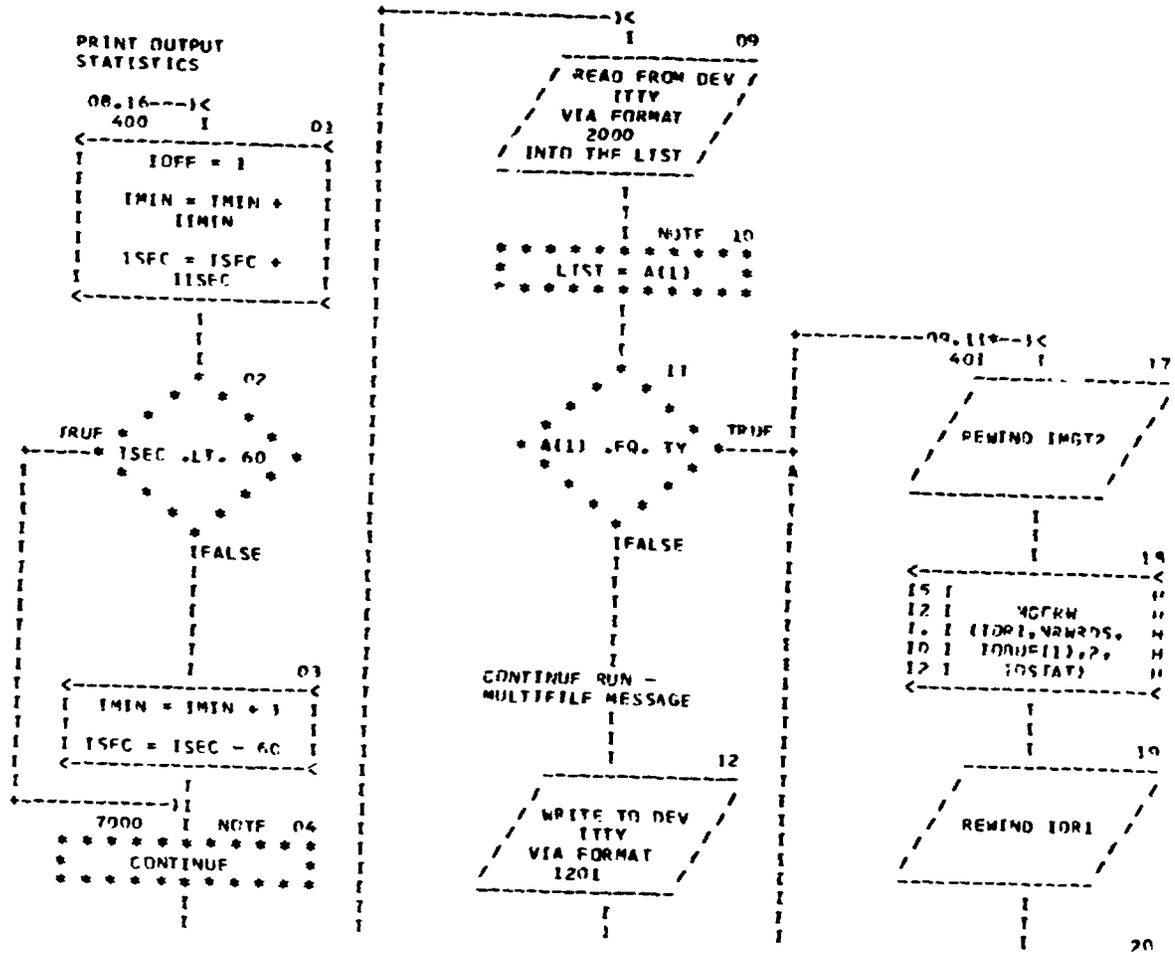

06
IOFF = 0
07
REWIND IDR1
08
WRITE TO DEV ITTY VIA FORMAT 1050 FROM THE LIST
09
NOTE
LIST = IDR1
10
PAUSE
316
12
TRUE
ITYPE .LE. 2
391
16
IFALSE
9.01.
400
13
TRUE
ITYPE .EQ. 4
IFALSE
7.
02.
320
14
NSET
15
IOUT1 = IOUT1S
IOUT2 = IOUT2S
6.12.
305

CHART TITLE - PROCEDURES



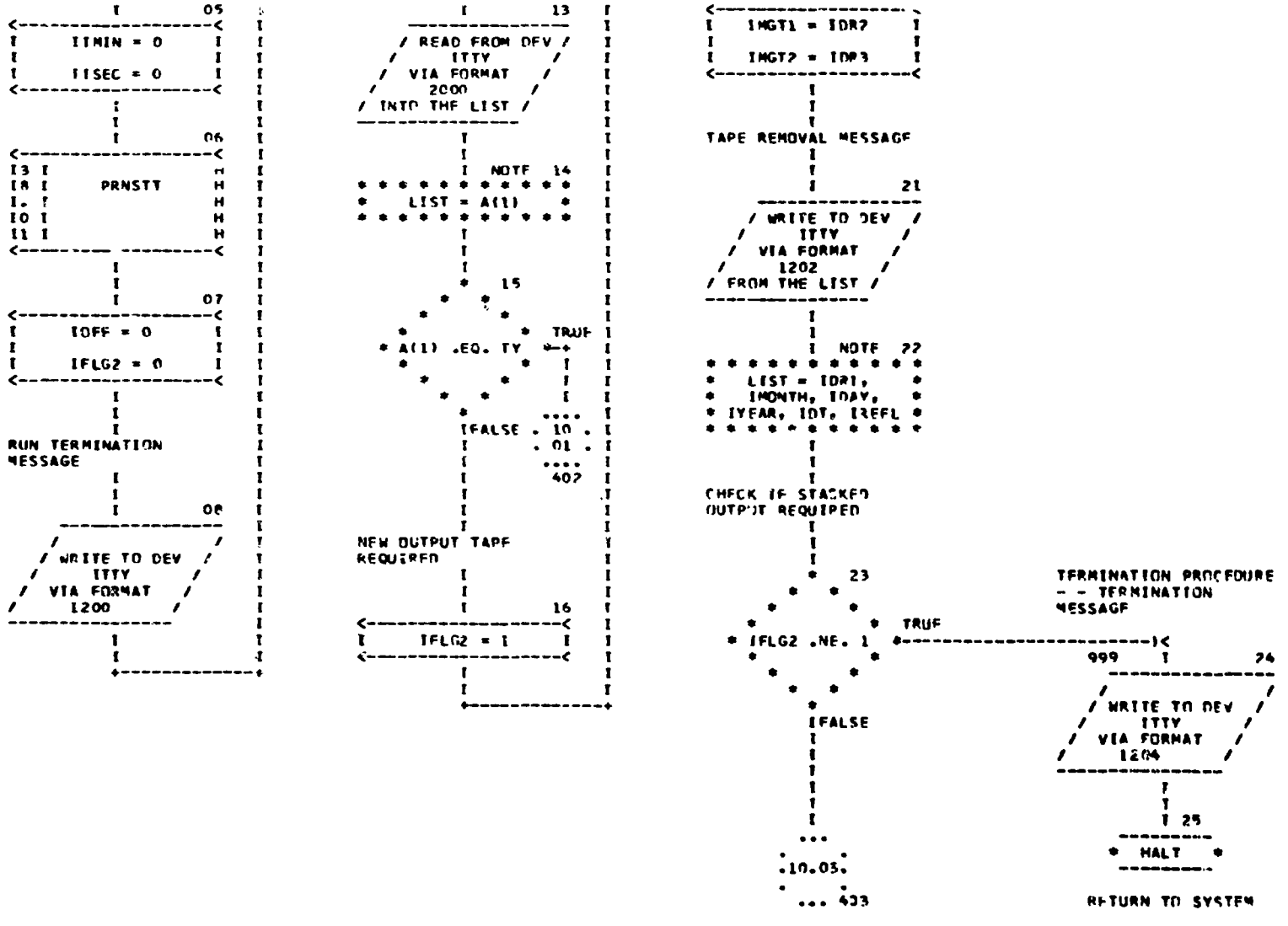
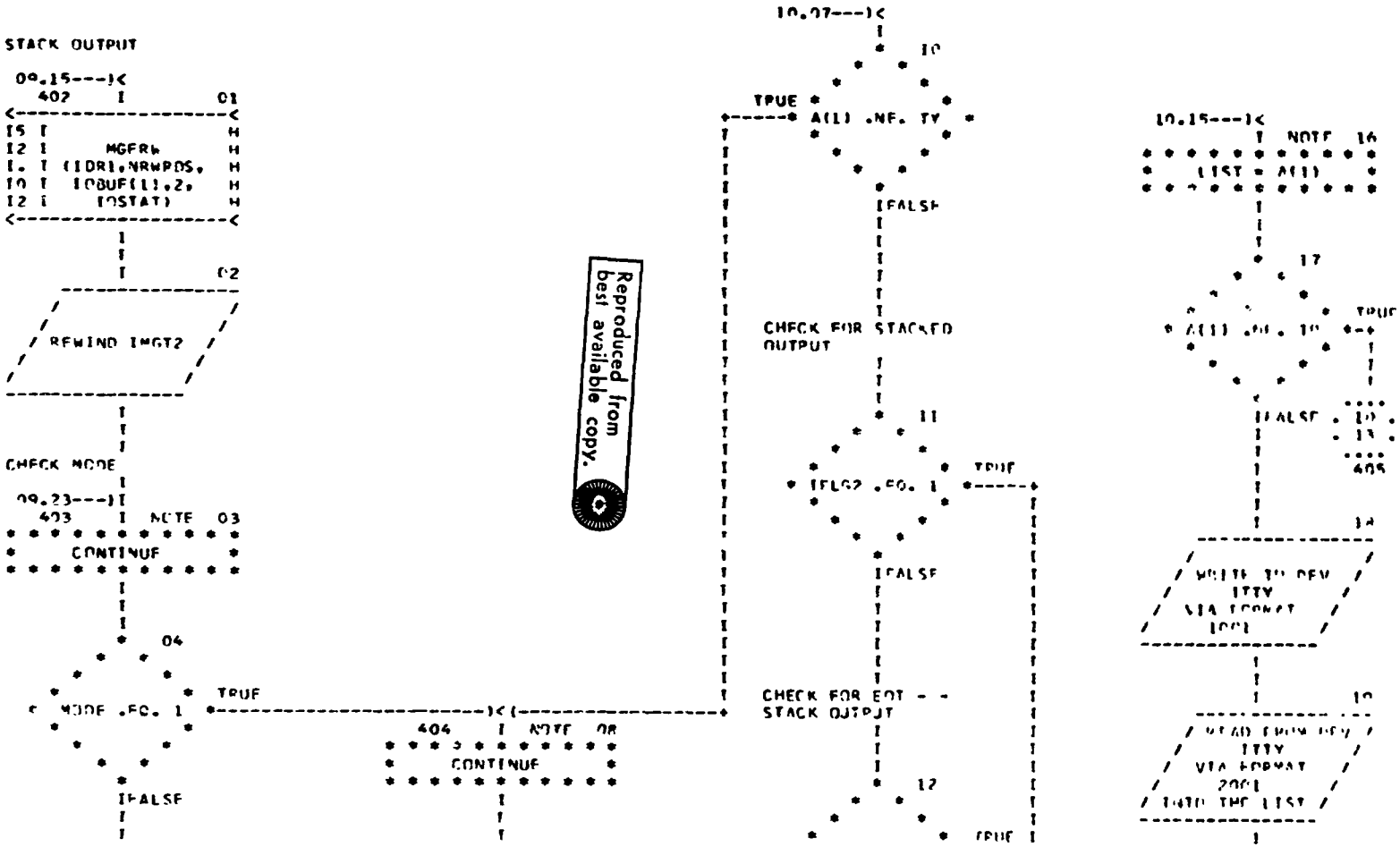

05
ITMIN = 0
ITSEC = 0
06
PRNSTT
07
IOFF = 0
IFLG2 = 0
RUN TERMINATION
MESSAGE
08
WRITE TO DEV
ITTY
VIA FORMAT
1200
13
READ FROM DEV
ITTY
VIA FORMAT
2000
INTO THE LIST
NOTE 14
LIST = A(1)
15
A(1) .EQ. TY
TRUE
IFALSE
10
01
402
NEW OUTPUT TAPE
REQUIRED
16
IFLG2 = 1
IMGT1 = IDR2
IMGT2 = IDR3
TAPE REMOVAL MESSAGE
21
WRITE TO DEV
ITTY
VIA FORMAT
1202
FROM THE LIST
NOTE 22
LIST = IDR1,
IMONTH, IDAY,
IYEAR, IDT, IREFL
CHECK IF STACKED
OUTPUT REQUIRED
23
IFLG2 .NE. 1
TRUE
IFALSE
10.03.
433
TERMINATION PROCEDURE
- - TERMINATION
MESSAGE
999
24
WRITE TO DEV
ITTY
VIA FORMAT
1204
25
HALT
RETURN TO SYSTEM

CHART TITLE - PROCEDURES



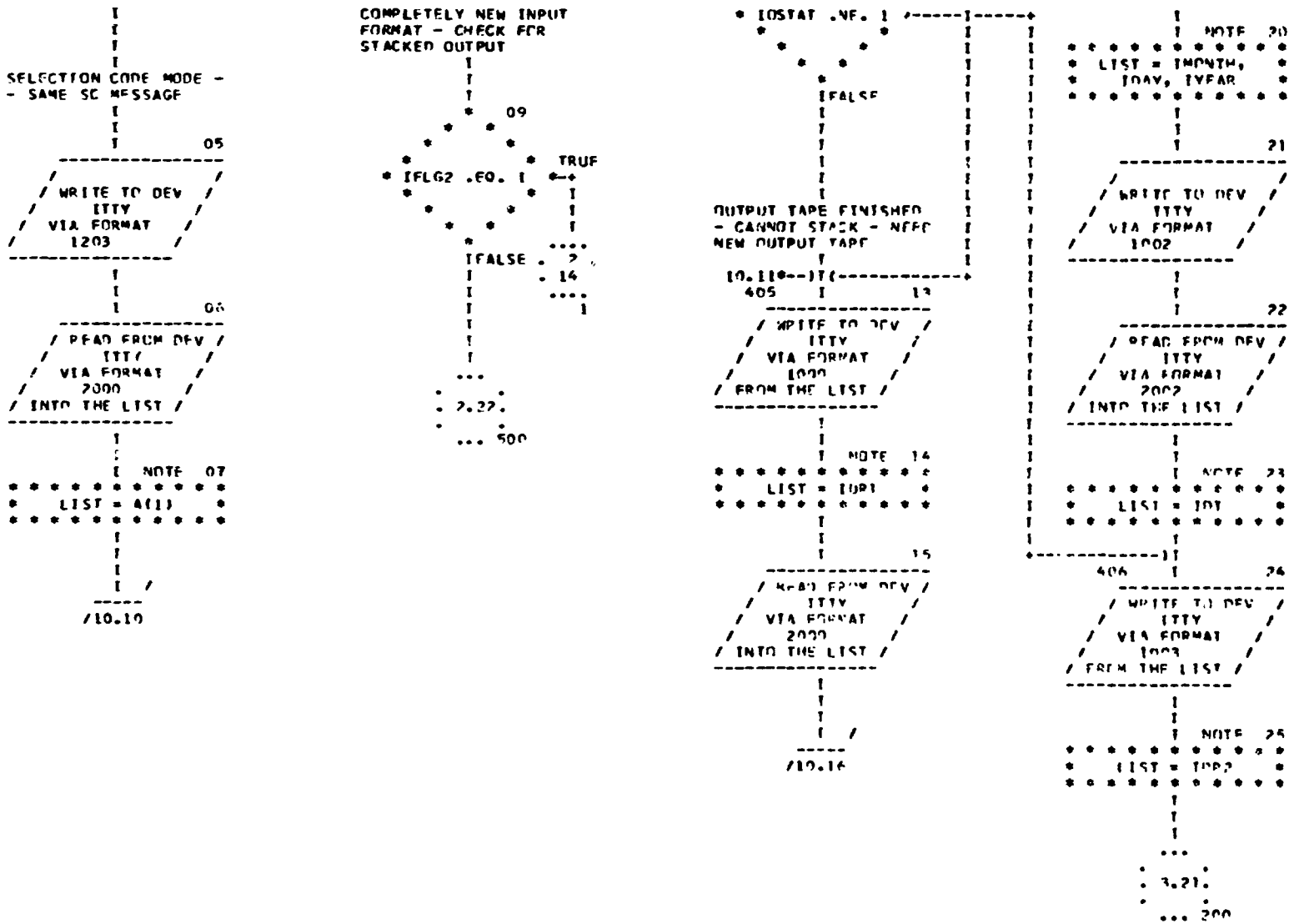

SELECTION CODE MODE -
- SAME SC MESSAGE
05
WRITE TO DEV
ITTY
VIA FORMAT
1203
06
READ FROM DEV
ITTY
VIA FORMAT
2000
INTO THE LIST
NOTE 07
LIST = A(I)
/10.10
COMPLETELY NEW INPUT
FORMAT - CHECK FOR
STACKED OUTPUT
09
IFLG2 .EQ. 1
TRUE
IFALSE
2,22,
... 500
IOSTAT .NE. 1
IFALSE
OUTPUT TAPE FINISHED
- CANNOT STACK - NEED
NEW OUTPUT TAPE
10.11
405
13
WRITE TO DEV
ITTY
VIA FORMAT
1000
FROM THE LIST
NOTE 14
LIST = IOB1
15
READ FROM DEV
ITTY
VIA FORMAT
2000
INTO THE LIST
/10.16
NOTE 20
LIST = IMONTH,
IDAY, IYEAR
21
WRITE TO DEV
ITTY
VIA FORMAT
1002
22
READ FROM DEV
ITTY
VIA FORMAT
2002
INTO THE LIST
NOTE 23
LIST = IOT
406
24
WRITE TO DEV
ITTY
VIA FORMAT
1003
FROM THE LIST
NOTE 25
LIST = IOB2
3,21,
... 200

CHART TITLE - NON-PROCEDURAL STATEMENTS

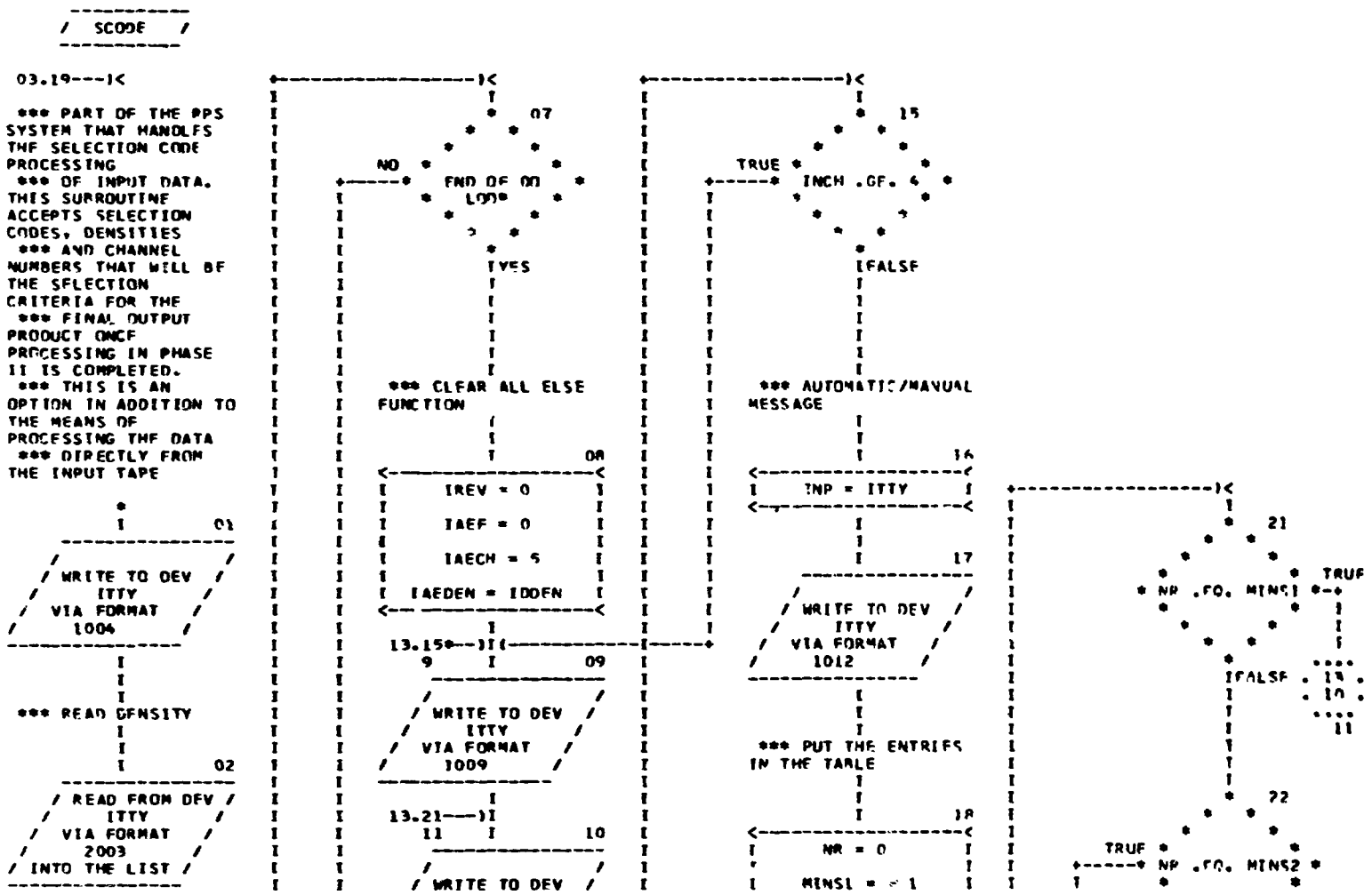
```
      DIMENSION ISC(1000,2),A(9)
      DIMENSION MASK1(3),MASK2(3),MASKS(6)
      DIMENSION ITB  (740)
      DIMENSION INSET(6),IOBUF(1364)
      DIMENSION ICNT(25),ICH(4)
      COMMON /A1/ ISC,ITMAX,ITCHR,MAXSC,MAXSC
      COMMON /B1/ INSC,INCH,INDEN,INDEL
      COMMON /C1/ IAFF,IAFCH,IAFDEN
      COMMON /D1/ A,I  N,ISCS,ISCF,IDDEN
      COMMON /E1/ TN,TA,TF,TX,TY,TD,TP,TT,TS,TR,TM
      COMMON /F1/ ITTY,IPT,IDT,INP,IPRTP
      COMMON /G1/ IOBL,IOB2,IOB3,IOEC
      COMMON /H1/ IOSC,IOCH,IODEN,IODEL
      COMMON /I1/ IMONTH,IDAY,IYEAR,IPAGE,IFILE,IREV
      COMMON /J1/ IMGT1,IMGT2,IDONE
      COMMON /MASK/ MASK1,MASK2
      COMMON /GNR1/ ITYPE,ICORD,ICON1,ICON2,NX,IX,IFT,ISTAT
      COMMON /TRUE/ ITRUE,NYS,NXS
      COMMON /AFLC/ IMINF,IMMAP
      COMMON /IOUT/ IOUT1,IOUT2,IATST
      COMMON /K1/ ICNT,ICH
      COMMON /L1/ IMIN,ISEC,IOFF
      COMMON /M1/ INSET,LSTWRD,LSTBYT,NENT,IOBUF
      EQUIVALENCE (MASK1(1),MASKS(1))
      EQUIVALENCE (MASK11,MASK1(1)),(MASK12,MASK1(2)),(MASK13,MASK1(3))
```

```
      EQUIVALENCE (MASK21,MASK2(1)),(MASK22,MASK2(2)),(MASK23,MASK2(3))
1000  FORMAT (28H MOUNT NEW SCRATCH ON DRIVE ,I1,19H   ENTER   0   WHEN,
     5H DONE)
2000  FORMAT (A1)
1001  FORMAT (46H ENTER MONTH,DAY,YEAR AS 6 NUMERICS ... MMDDYY)
2001  FORMAT (3I2)
1002  FORMAT (29H ENTER 5 DIGIT TAPE ID NUMBER)
2002  FORMAT (I5)
1003  FORMAT (39H MOUNT 1ST REEL OF INPUT TAPE ON DRIVE ,I1)
1101  FORMAT (49H ENTER 6 DIGIT MINIMUM AREA IN PROPER RESOLUTION.
2100  FORMAT (I6)
1100  FORMAT (50H DO YOU WISH TO SUPPLY SELECTION CODES EXTERNALLY / 13H
      ENTER Y OR N)
2222  FORMAT (23H BAD WRITE AT IWRITE = ,I5)
1050  FORMAT (40H AT PAUSE 2, MOUNT NEW SCRATCH ON DRIVE ,I1,14H  /AND CO
     NTINUE)
5000  FORMAT (16H TOTAL WRITES = ,I5,4H + 1)
1200  FORMAT (42H DO YOU WISH TO TERMINATE RUN   (Y  OR  N))
1201  FORMAT (47H DO YOU WISH TO STACK OUTPUT FILES   (Y  OR  N))
1202  FORMAT (28H REMOVE OUTPUT TAPE ON DRIVE,1X,I1/ 15H LABEL IT  DATE,
     4X,I2,1H/,I2,1H/,I2,14H TAPE NUMBER  ,I6,14H REEL NUMBER  ,I2)
1203  FORMAT (61H DO YOU WISH TO USE THE SAME SELECTION CODES AS PREVIOU
```

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      S RUN / 11H (Y  OR  N))
1204  FORMAT (42H END OF RUN - REMOVE ALL TAPES FROM DRIVES///20X,20H **
      * TERMINATION ***)
```

CHART TITLE - SUBROUTINE SCODE



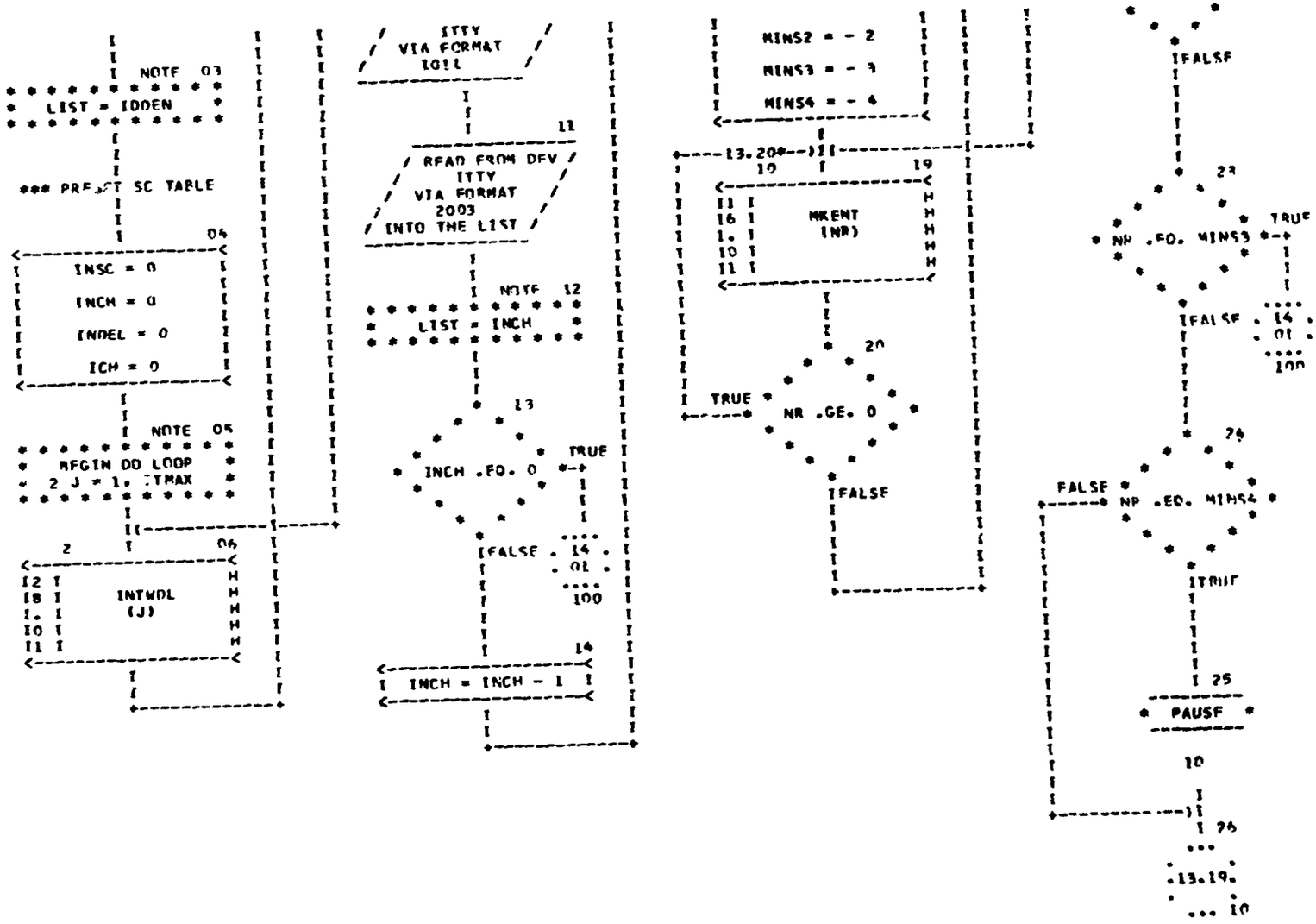
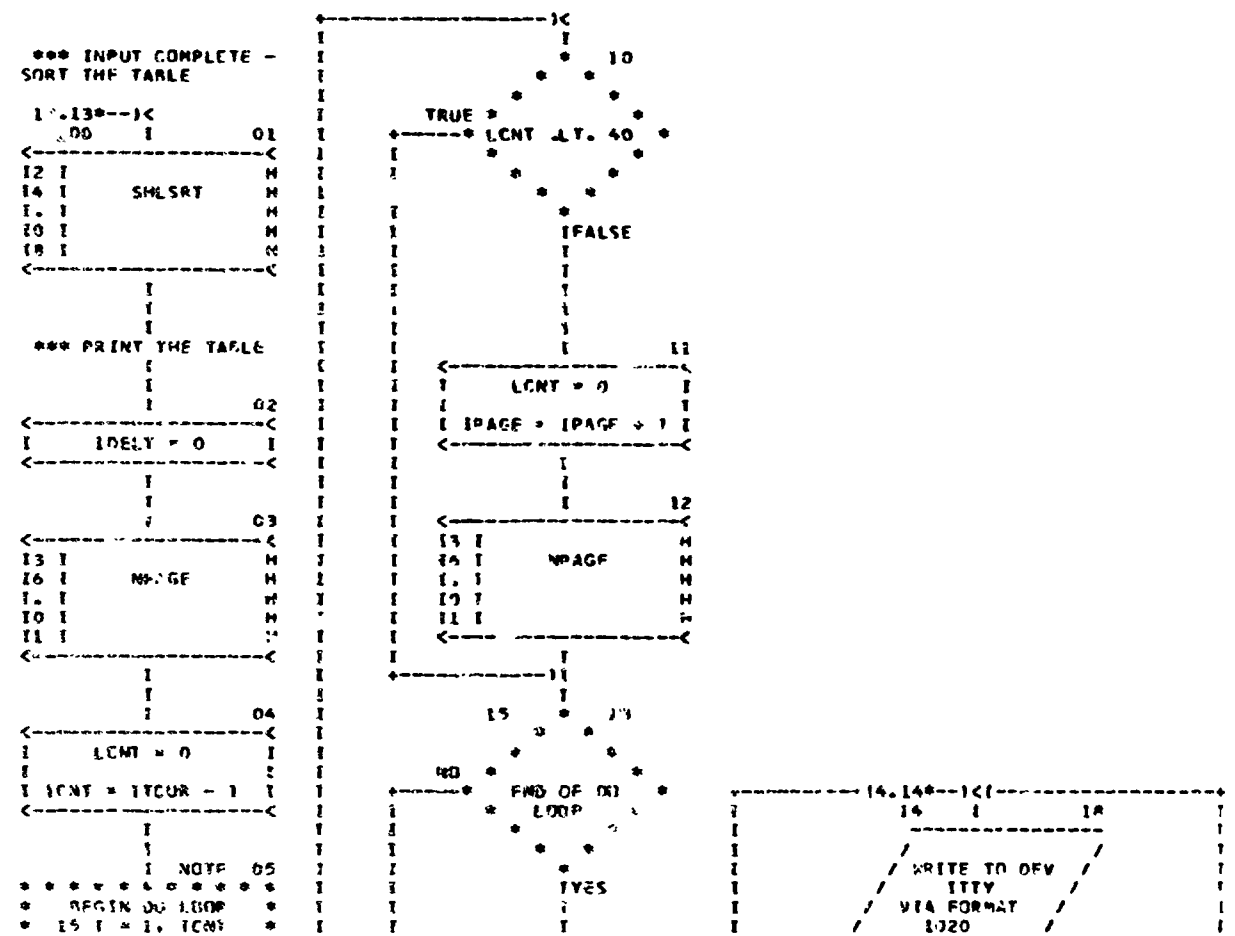

NOTE 03
LIST = IODEN
*** PRESET SC TABLE
04
INSC = 0
INCH = 0
INDEL = 0
ICH = 0
NOTE 05
BEGIN DO LOOP
J = 1, CTMAX
2
06
INTNDL
(J)
ITTY
VIA FORMAT
1011
11
READ FROM DEV
ITTY
VIA FORMAT
2003
INTO THE LIST
NOTE 12
LIST = INCH
13
INCH .EQ. 0
TRUE
IFALSE
14
INCH = INCH - 1
MINS2 = - 2
MINS3 = - 3
MINS4 = - 4
13.20
10
19
NKENT
(NR)
20
NR .GE. 0
TRUE
IFALSE
IFALSE
23
NR .EQ. MINS3
TRUE
IFALSE
14
01
100
24
NR .EQ. MINS4
FALSE
ITRUE
25
PAUSE
10
24
13.19.
10

CHART TITLE - SUBROUTINE SCODE



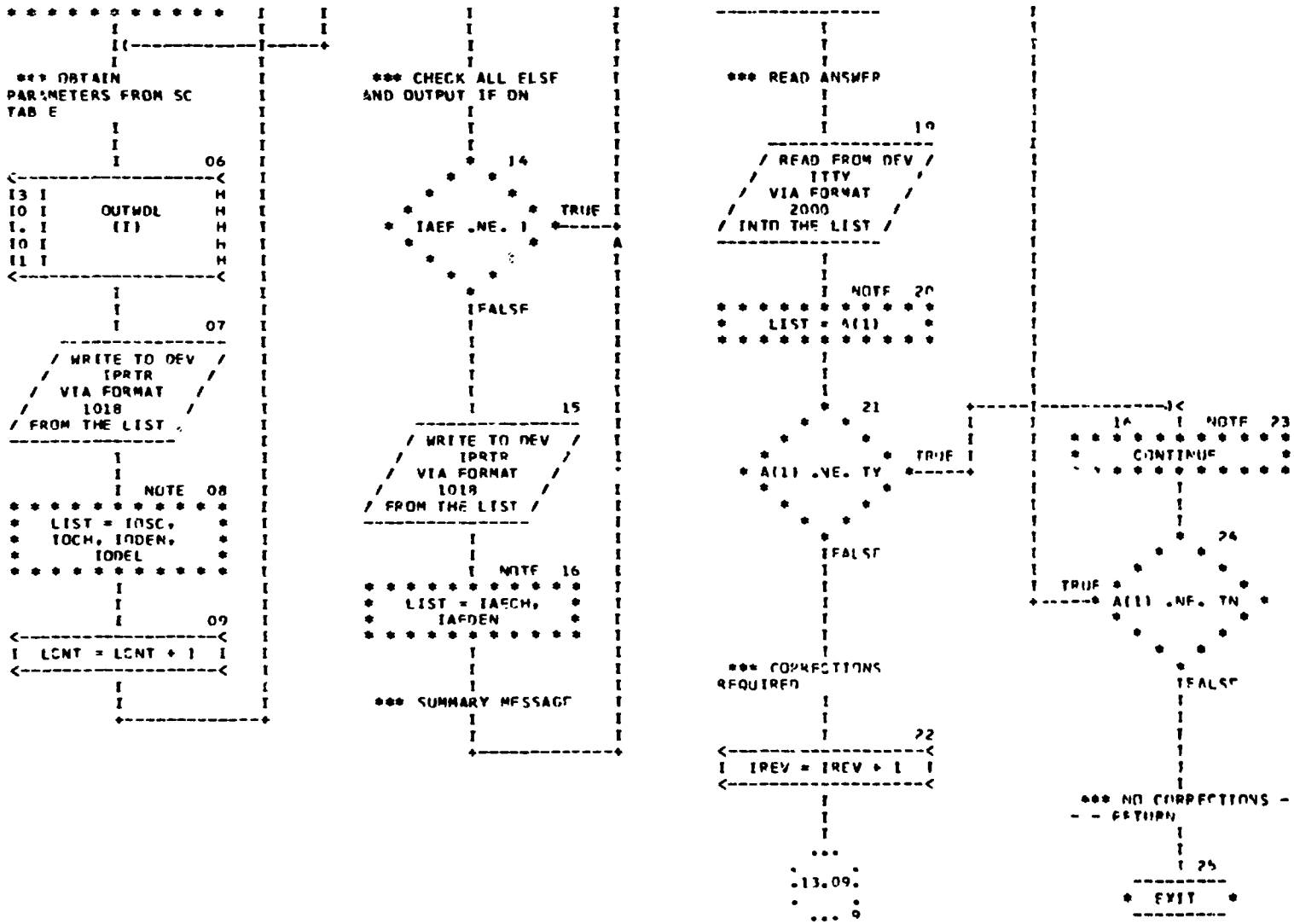
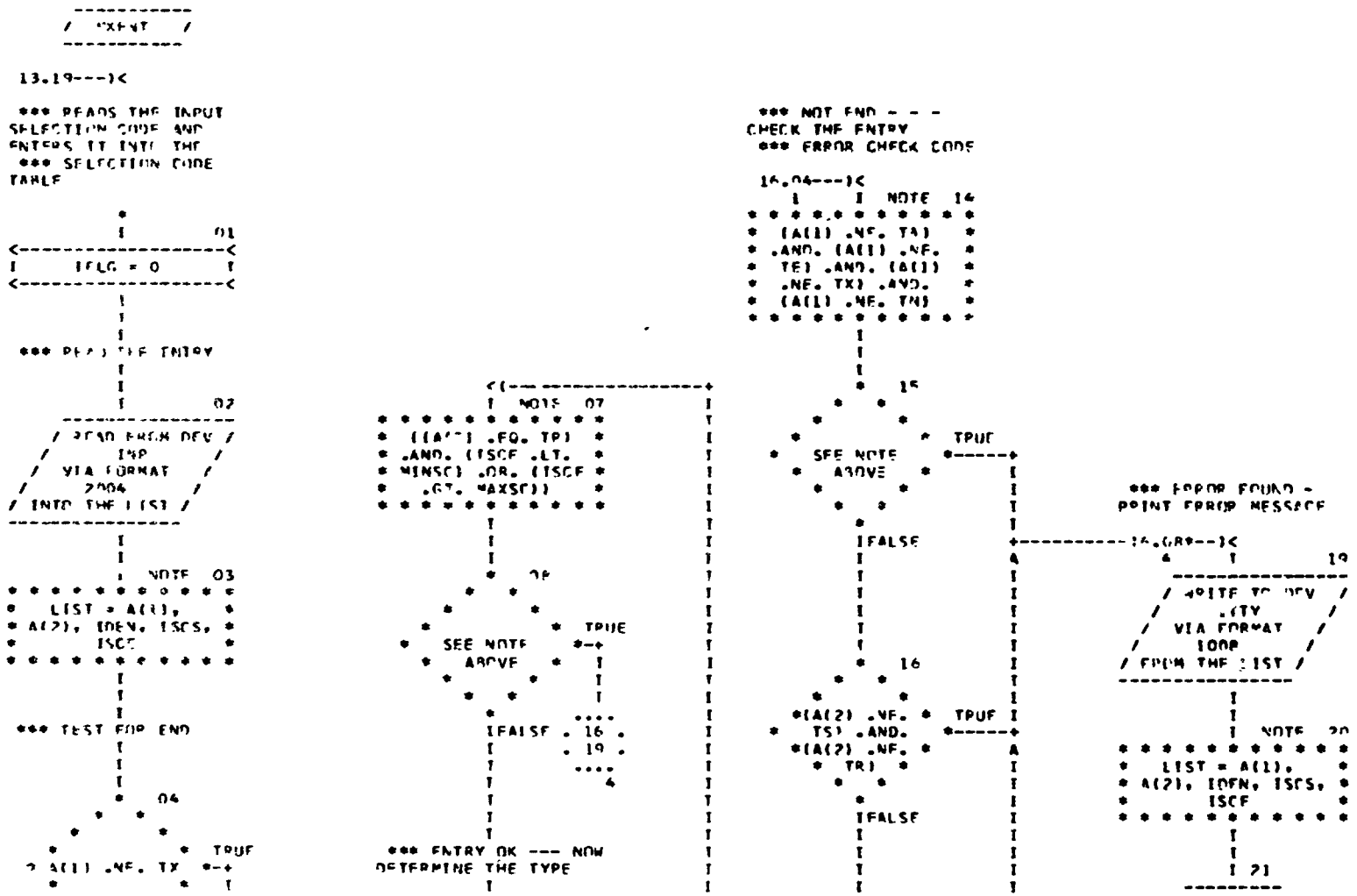
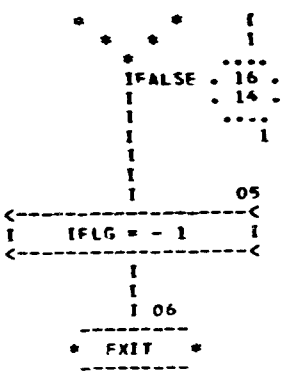

*** OBTAIN PARAMETERS FROM SC TAB E
06
I3 I
IO I
I. I
IO I
I1 I
OUTWDL (I)
07
/ WRITE TO DEV /
IPRTR
/ VIA FORMAT /
1018
/ FROM THE LIST /
NOTE 08
LIST = IOSC, IOCH, IODEN, IODEL
09
I CONT = CONT + 1 I
*** CHECK ALL ELSE AND OUTPUT IF ON
14
IAEF .NE. 1
TRUE
FALSE
15
/ WRITE TO DEV /
IPRTR
/ VIA FORMAT /
1018
/ FROM THE LIST /
NOTE 16
LIST = IAECH, IAEDEN
*** SUMMARY MESSAGE
*** READ ANSWER
19
/ READ FROM DEV /
ITTY
/ VIA FORMAT /
2000
/ INTO THE LIST /
NOTE 20
LIST = A(I)
21
A(I) .NE. TY
TRUE
FALSE
*** CORRECTIONS REQUIRED
22
I IREV = IREV + 1 I
.13.09.
... 9
1A
NOTE 23
CONTINUE
24
A(I) .NE. TN
TRUE
FALSE
*** NO CORRECTIONS - - RETURN
25
EXIT

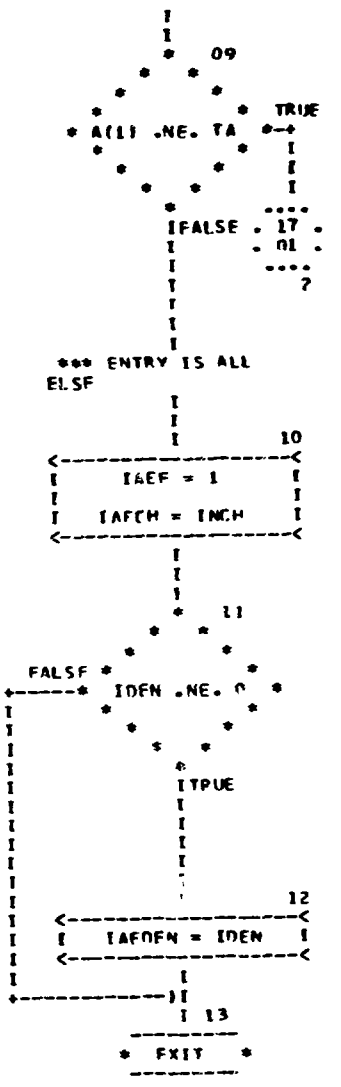
CHART TITLE - NON-PROCEDURAL STATEMENTS

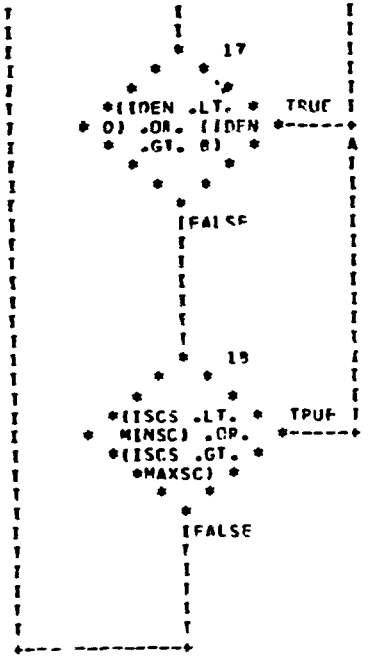
```
      DIMENSION ISC(1000,2),A(9)
      COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
      COMMON /B1/ INSC,INCH,INDEN,INDEL
      COMMON /C1/ IAEF,IAFCH,IAFDEN
      COMMON /D1/ A,IDEN,ISCS,ISCE,TDDEN
      COMMON /E1/ TN,TA,TF,TX,TY,TD,TP,TT,TS,TR,TM
      COMMON /F1/ ITTY,IPT,IOT,IMP,IPRTR
      COMMON /G1/ IOR1,IOR2,IOR3,IDEC
      COMMON /H1/ IOSC,IOCH,IODEN,IODEL
      COMMON /I1/ IMONTH,IDAY,IYEAR,IPAGE,IFILE,IREV
 1004 FORMAT (45H ENTER DEFAULT DENSITY AS NUMERIC FROM 0 TO 7)
 2003 FORMAT (I1)
 1009 FORMAT (51H ENTER CHANNEL NUMBER FROM 1 TO 4 OR ENTER 0 TO END/ 52
     H SELECTION CODE INPUT EACH TIME   CH NR.=   IS TYPED)
 1011 FORMAT (8H CH NR =)
 1012 FORMAT (12H ENTER CODES,2X,15H(WYZNNNNNMMMM))
 1018 FORMAT (6X,I5,15X I2,11X,I1,12X,I1)
 1019 FORMAT (5X,8HALL ELSE,13X,I2,11X,I1)
 1020 FORMAT (61H SELECTION CODE SUMMARY ON PRINTER. ARE COPIES REQ
     UIRED / 14H ENTER Y OR N.)
```

CHART TITLE - SUBROUTINE MKENT(IFLG)




IFALSE . 16 . . 14 . 1
05
IFLG = - 1
06
EXIT


09
A(I) .NE. TA
TRUE
IFALSE . 17 . . 01 . ?
*** ENTRY IS ALL ELSE
10
IAEF = 1
IAFCH = INCH
11
FALSE
IDEN .NE. 0
ITRUE
12
IAFDEN = IDEN
13
EXIT


17
*(IDEN .LT. 0) .OR. (IDEN .GT. 8)
TRUE
IFALSE
18
*(ISCS .LT. MINSC) .OR. *(ISCS .GT. MAXSC)
TRUE
IFALSE

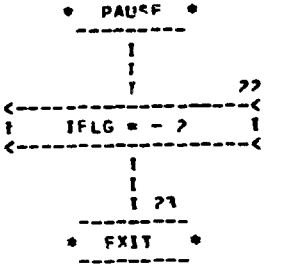
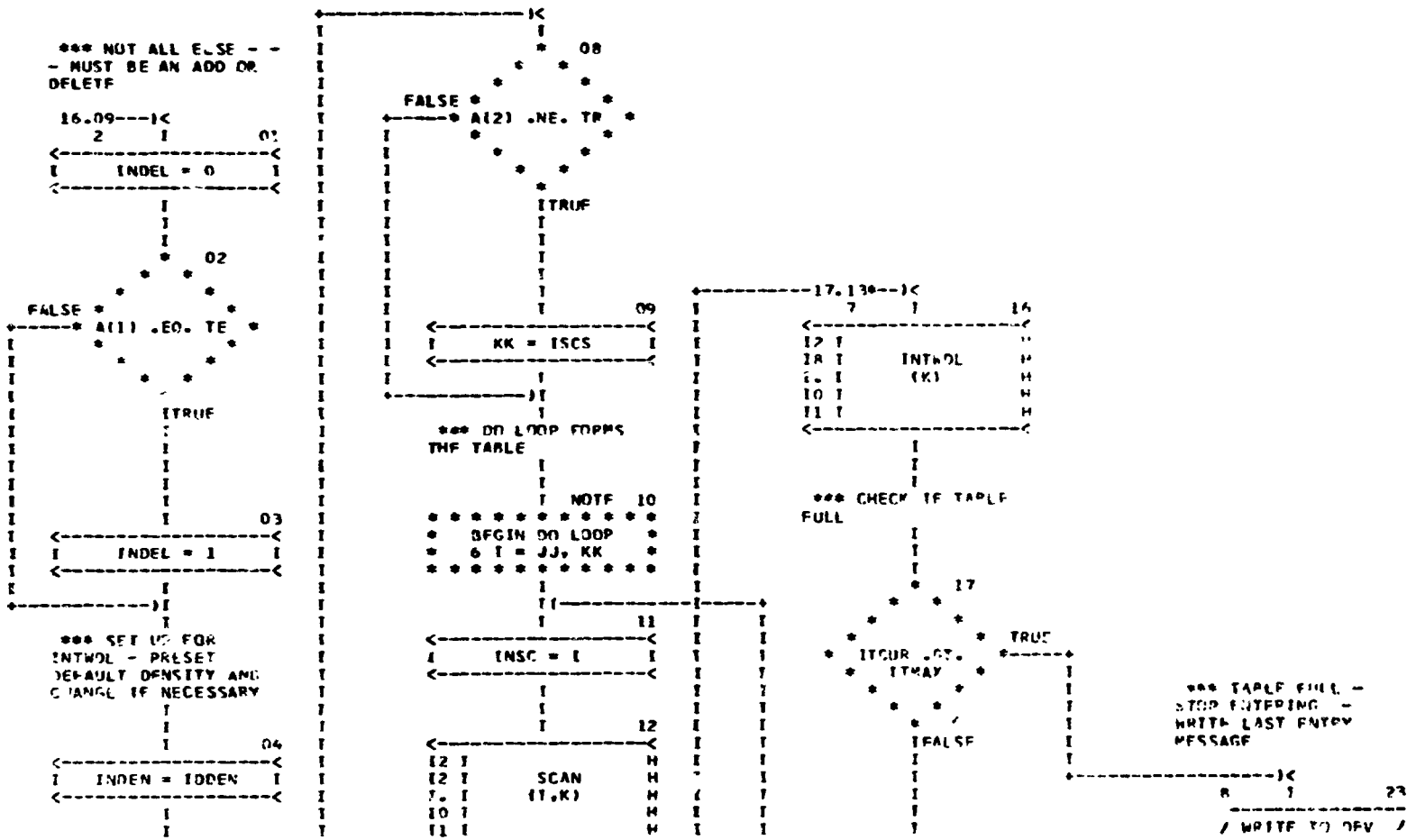
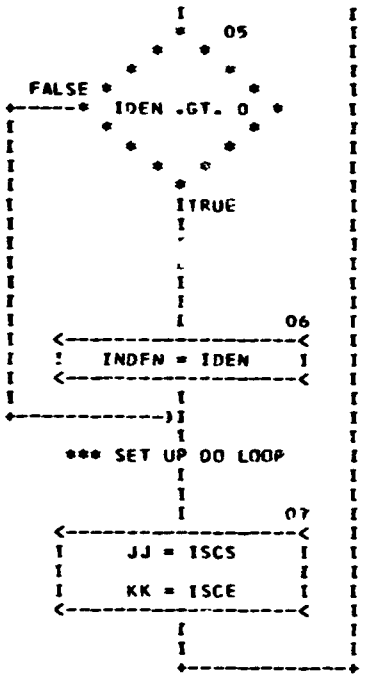
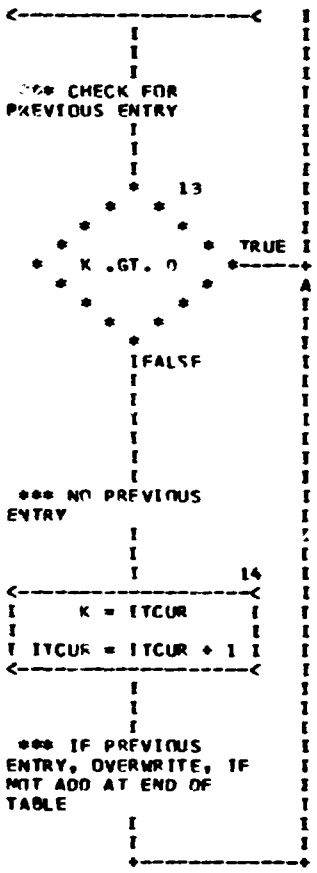
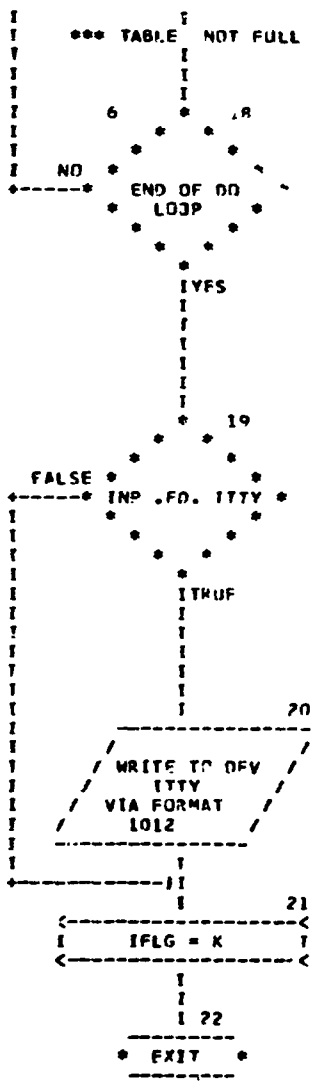

PAUSE
22
IFLG = - 2
23
EXIT

CHART TITLE - SUBROUTINE MKENT(IFLG)




05
FALSE
IDEN .GT. 0
TRUE
06
INDEN = IDEN
*** SET UP DO LOOP
07
JJ = ISCS
KK = ISCE


*** CHECK FOR PREVIOUS ENTRY
13
K .GT. 0
TRUE
FALSE
*** NO PREVIOUS ENTRY
14
K = ITCUR
ITCUR = ITCUR + 1
*** IF PREVIOUS ENTRY, OVERWRITE, IF NOT ADD AT END OF TABLE


*** TABLE NOT FULL
6
NO
END OF DO LOOP
YES
19
FALSE
INP .EQ. ITTY
TRUE
20
WRITE TO DEV ITTY VIA FORMAT 1012
21
IFLG = K
22
EXIT

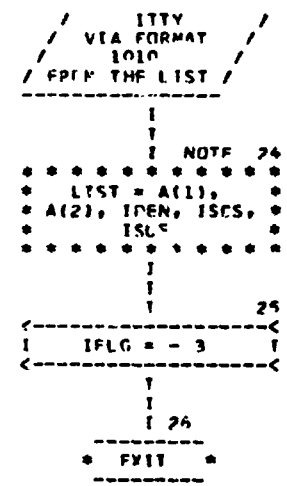
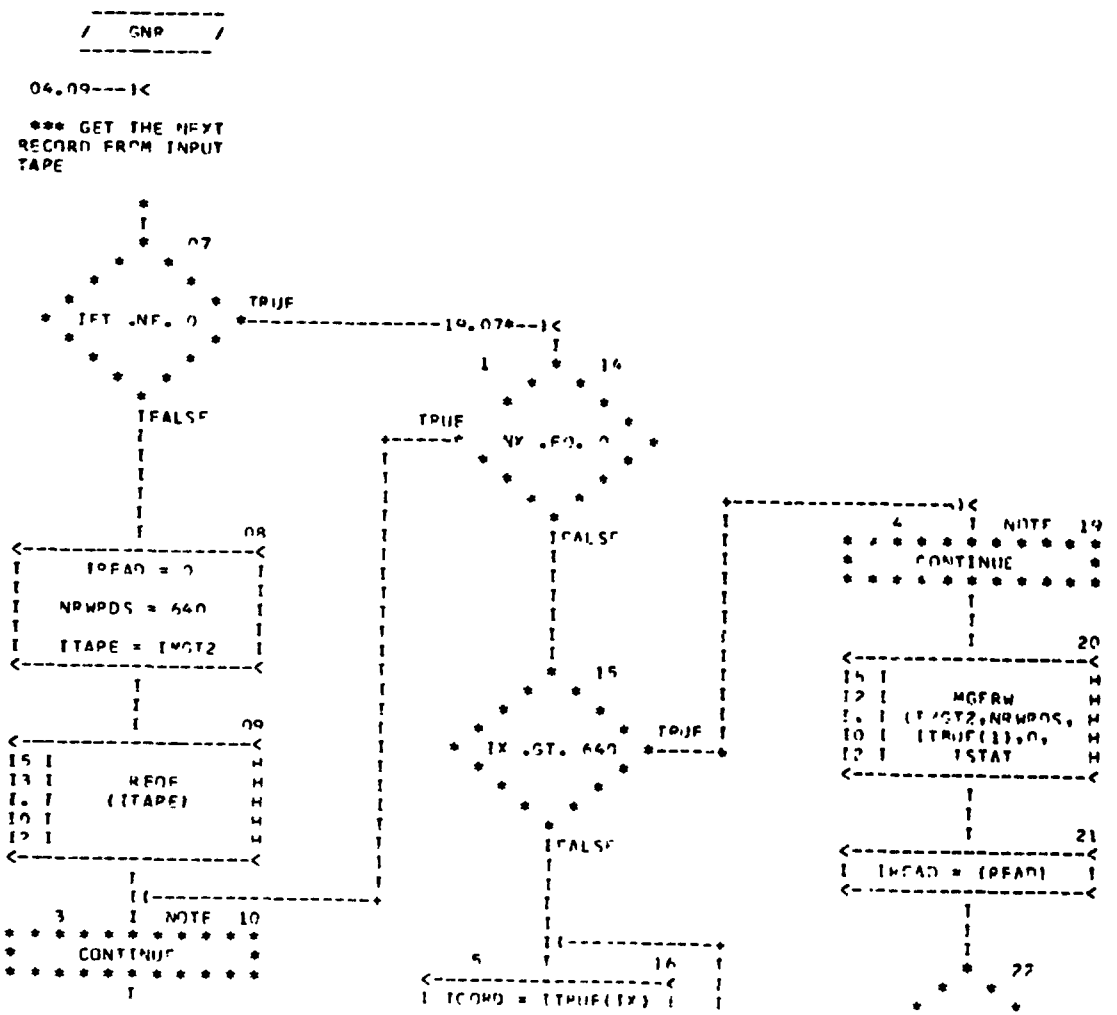

ITTY VIA FORMAT 1010 PRINT THE LIST
NOTE 24
LIST = A(1), A(2), IDEN, ISCS, ISCE
25
IFLG = - 3
26
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      DIMENSION ISC(1000,2), A(9)
      COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
      COMMON /B1/ INSC,INCH,INDEN,INDEL
      COMMON /C1/ IAEF,IAECH,IAEDEN
      COMMON /D1/ A,IDEN,ISCS,ISCE,IDDEN
      COMMON /E1/ TN,TA,TE,TX,TY,TD,TP,TT,TS,TR,TM
      COMMON /F1/ ITTY,IPT,IDT,INP,IPRTR
      COMMON /G1/ IDR1,IDR2,IDR3,IDEC
      COMMON /H1/ IDSC,IDCH,IDDEN,IDDEL
 2004 FORMAT (A1,A1,I1,2I5)
 1012 FORMAT (3H OK)
 1010 FORMAT (39H SELECTION CODE TABLE FULL - LAST ENTRY,
     /1X,A1,1X,A1,1X,I1,1X,I5,1X,I5)
 1008 FORMAT (58H ERRONEOUS SELECTION CODE ENTRY - WILL NOT BE PUT IN TA
     8LE/ 22H ENTRY AS RECEIVED IS /1X,A1,1X,A1,1X,I1,1X,I5,1X,I5/43H A
     T PAUSE 1, CONTINUE OR MANNUALLY TERMINATE)
```

CHART TITLE - SUBROUTINE GNR



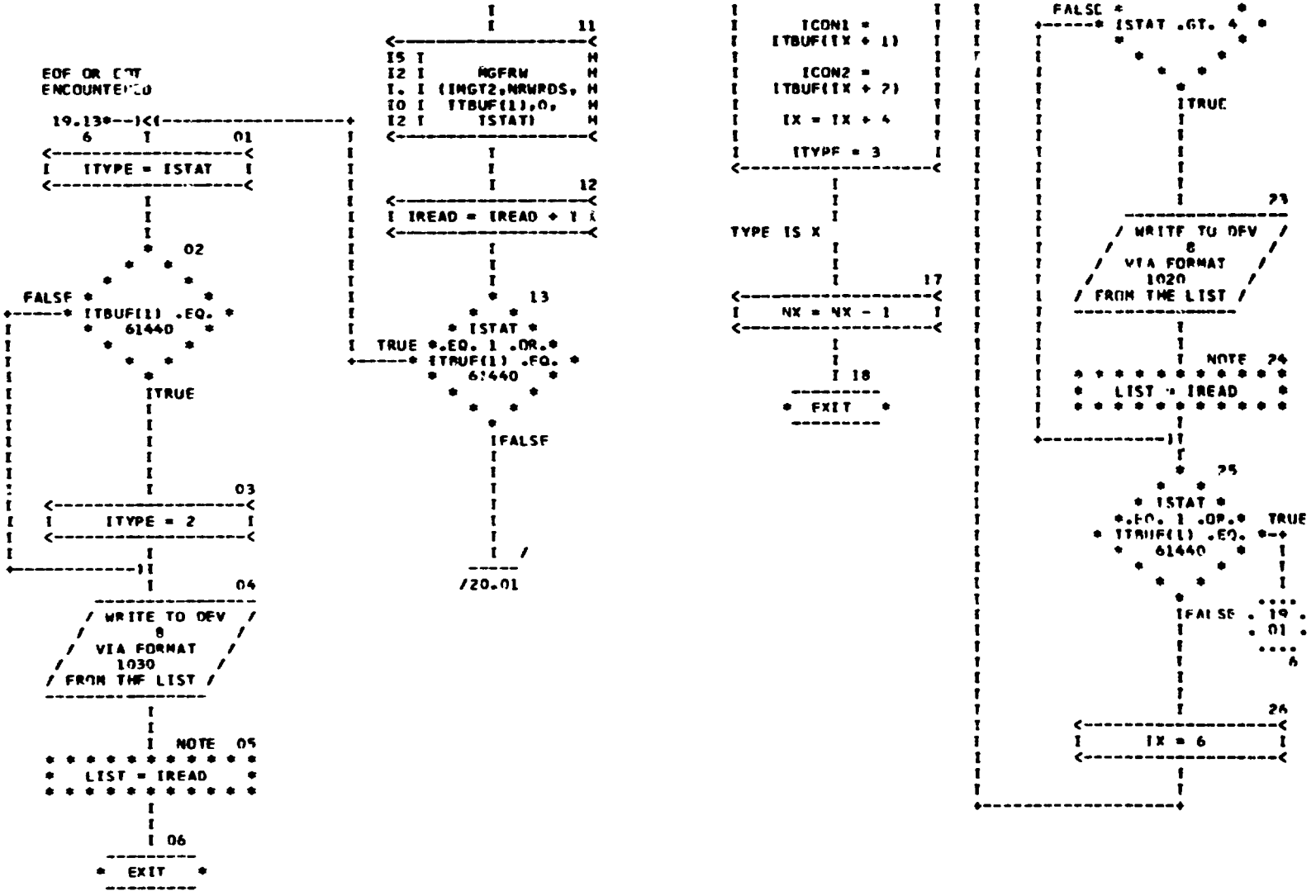
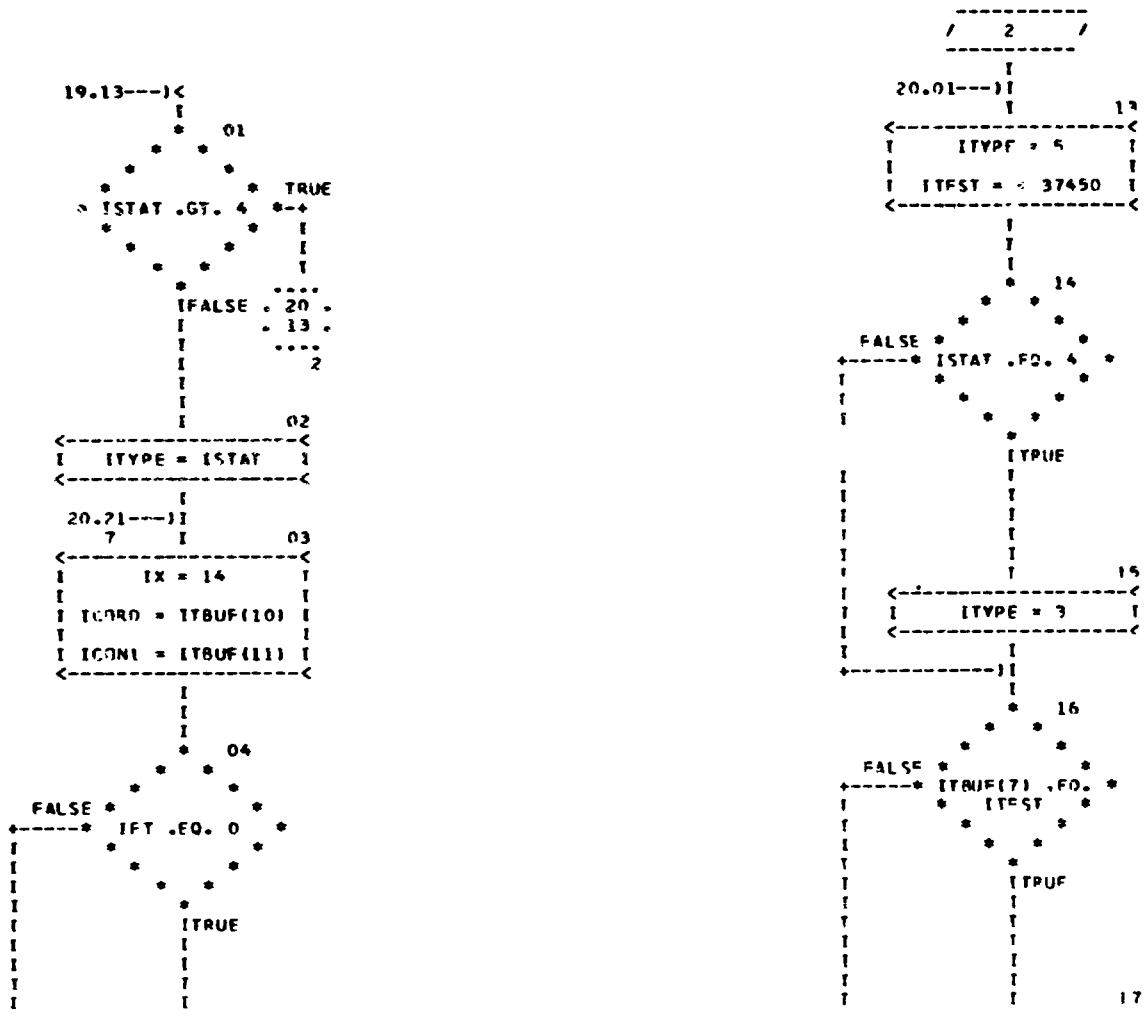

EOF OR EOT
ENCOUNTERED
19.13
6
01
ITYPE = ISTAT
02
FALSE
ITBUF(1) .EQ.
61440
ITRUE
03
ITYPE = 2
04
WRITE TO DEV
8
VIA FORMAT
1030
FROM THE LIST
NOTE 05
LIST = IREAD
06
EXIT
11
MGFRW
(IMGT2,NRWRDS,
ITBUF(1),0,
ISTAT)
12
IREAD = IREAD + 1
13
ISTAT
TRUE
.EQ. 1 .OR.
ITBUF(1) .EQ.
61440
IFALSE
/20.01
ICON1 =
ITBUF(IX + 1)
ICON2 =
ITBUF(IX + 2)
IX = IX + 4
ITYPE = 3
TYPE IS X
17
NX = NX - 1
18
EXIT
FALSE
ISTAT .GT. 4
23
WRITE TO DEV
8
VIA FORMAT
1020
FROM THE LIST
NOTE 24
LIST = IREAD
25
ISTAT
.EQ. 1 .OR.
ITBUF(1) .EQ.
61440
TRUE
IFALSE
19
01
6
26
IX = 6

CHART TITLE - SUBROUTINE GNR



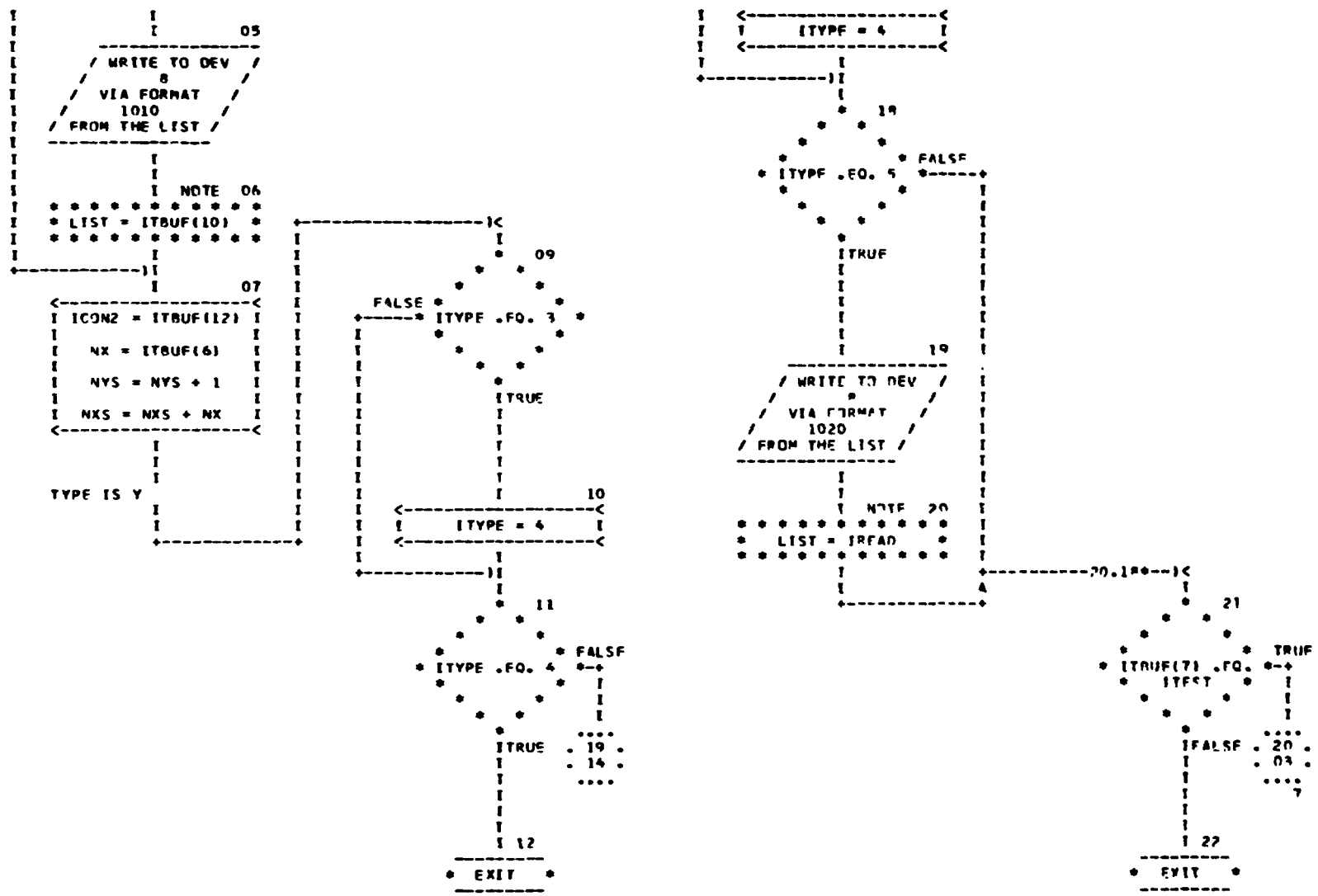
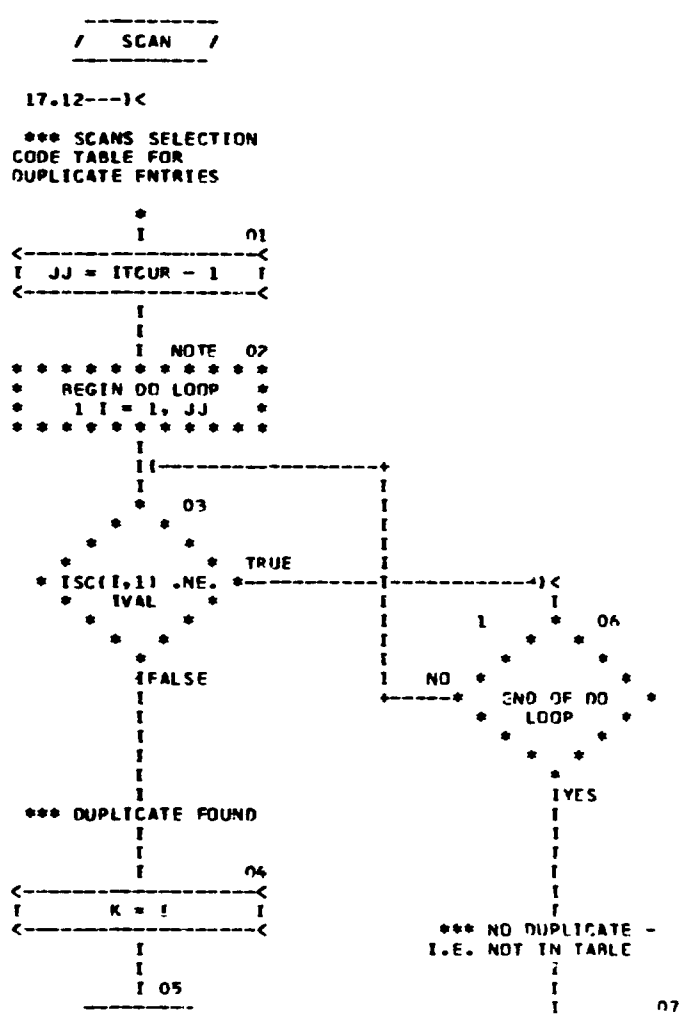

05
WRITE TO DEV 8 VIA FORMAT 1010 FROM THE LIST
NOTE 06
LIST = ITBUF(10)
07
ICON2 = ITBUF(12)
NX = ITBUF(6)
NYS = NYS + 1
NXS = NXS + NX
TYPE IS Y
09
ITYPE .EQ. 3
FALSE
TRUE
10
ITYPE = 4
11
ITYPE .EQ. 4
FALSE
TRUE
19
12
EXIT
ITYPE = 4
19
ITYPE .EQ. 5
FALSE
TRUE
19
WRITE TO DEV 1020 FROM THE LIST
NOTE 20
LIST = IREAD
20.1
21
ITRUE(7) .EQ. ITEST
TRUE
FALSE
20
03
7
22
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      DIMENSION ITBUF(740)
      COMMON /GNR1/ ITYPE,ICORD,ICON1,ICON2,NX,IX,IFT,ISTAT
      COMMON /TBUF/ ITBUF,NYS,NYS
      COMMON/J1/IMGT1,IMGT2,IDONE
1010  FORMAT (23H FIRST Y COORDINATE IS ,I6)
1020  FORMAT (21H BAD READ AT IREAD = ,I5)
1030  FORMAT (15H TOTAL READS = ,I5)
```

CHART TITLE - SUBROUTINE SCAN(IVAL,K)



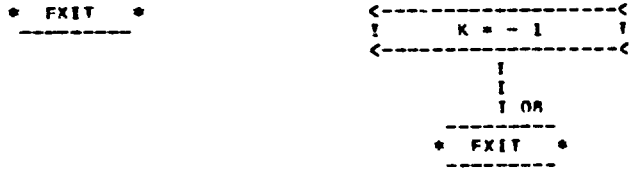
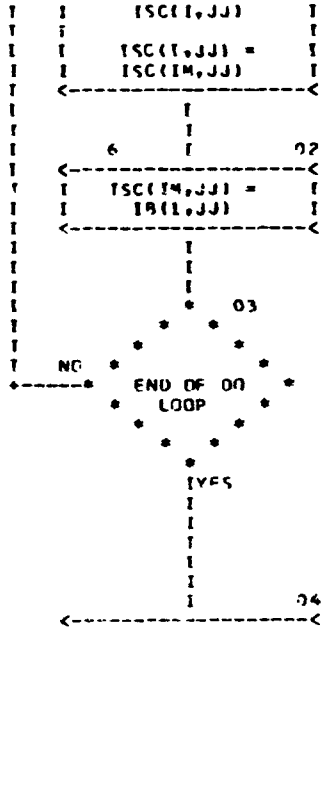
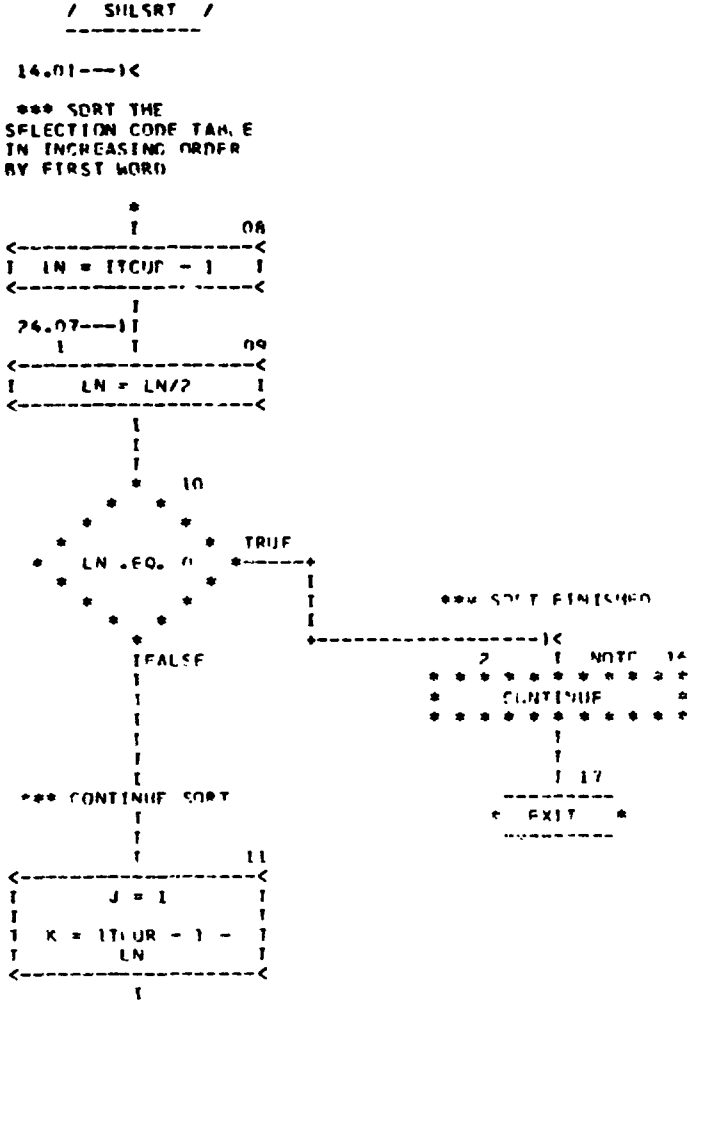

* FXIT *
K = - 1
I
I
I 08
* FXIT *

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      DIMENSION ISC(1000,2)
      COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
```

CHART TITLE - SUBROUTINE SHLSRT





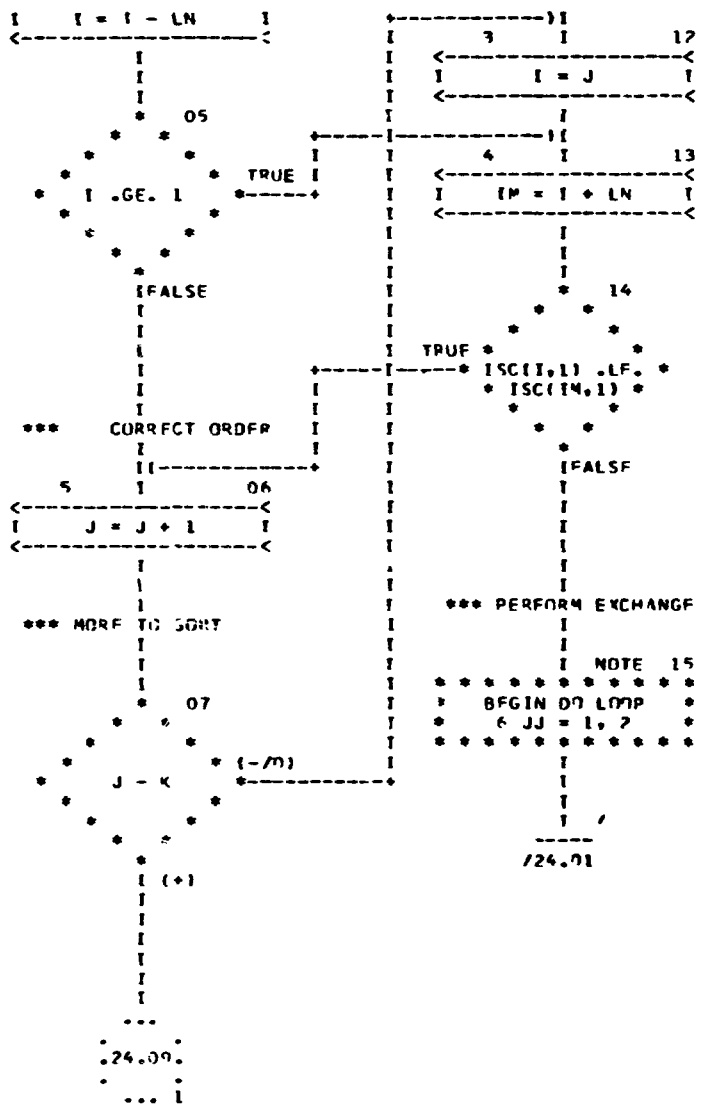
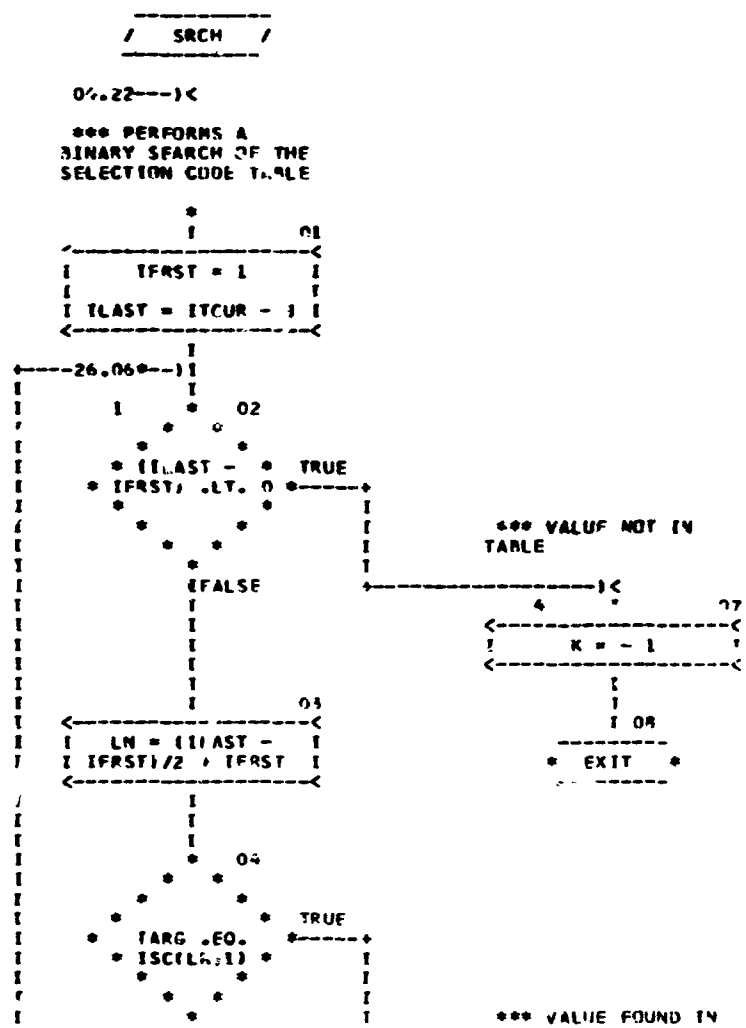

I = I - LN
05
I .GE. 1
TRUE
FALSE
*** CORRECT ORDER
06
J = J + 1
*** MORE TO SORT
07
J = K
(-/0)
(+)
.24.09.
... 1
3
12
I = J
4
13
IM = I + LN
14
ISC(I,1) .LE. ISC(IM,1)
TRUE
FALSE
*** PERFORM EXCHANGE
NOTE 15
BEGIN DO LOOP
6 JJ = 1, 2
/24.01

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      DIMENSION ISC(1000,2),IT(1,2)
      COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
```

CHART TITLE - SUBROUTINE SRCH(IARG,K)



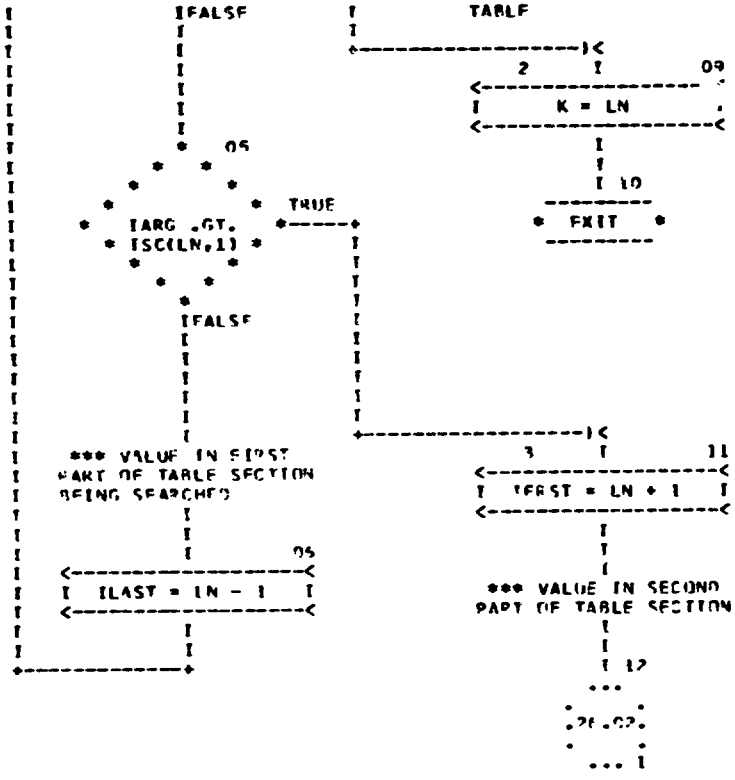
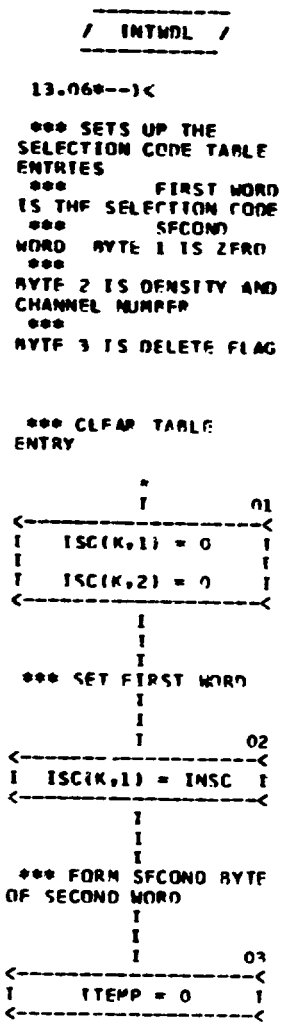

IFALSE
05
IARG .GT.
ISC(LN,1)
TRUE
IFALSE
*** VALUE IN FIRST
PART OF TABLE SECTION
BEING SEARCHED
05
ILAST = LN - 1
TABLE
2
09
K = LN
10
EXIT
3
11
IFRST = LN + 1
*** VALUE IN SECOND
PART OF TABLE SECTION
12

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      DIMENSION ISC(1000,2)
      COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
```

07/22/71 AUTOFLOW CHART SET - PASS PAGE 28

CHART TITLE - SUBROUTINE INTWDL(K)



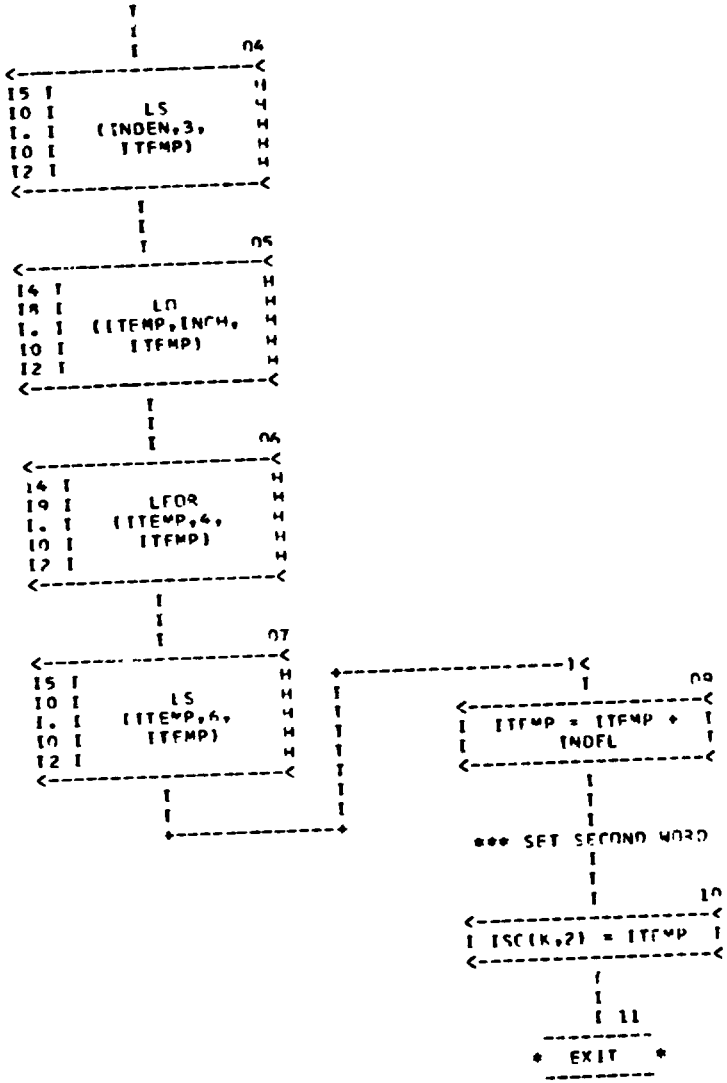
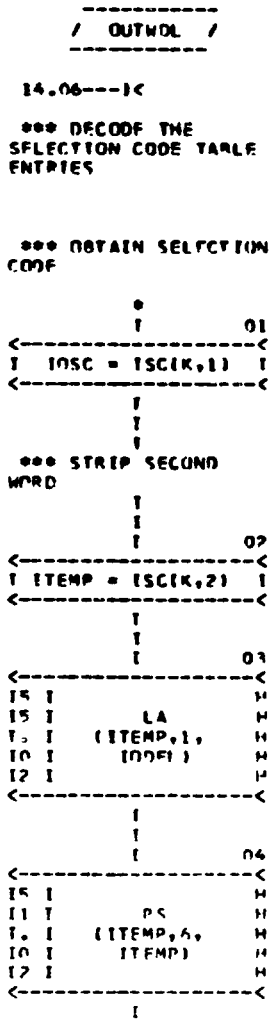

04
LS
(INDEN,3,
ITEMP)
05
LD
(ITEMP,INCH,
ITEMP)
06
LFOR
(ITEMP,4,
ITEMP)
07
LS
(ITEMP,6,
ITEMP)
09
ITEMP = ITEMP +
INDFL
*** SET SECOND WORD
10
ISC(K,2) = ITEMP
11
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      DIMENSION ISC(1000,2)
      COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
      COMMON /B1/ INSC,INCH,INDEN,INDEL
```

CHART TITLE - SUBROUTINE OUTWOL(K)



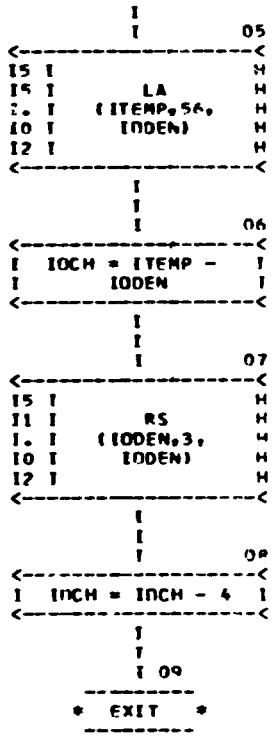
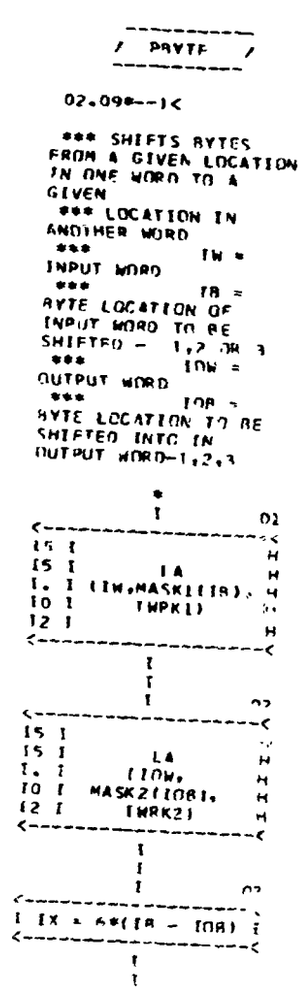

05
LA
(ITEMP,56,
IODEN)
06
IOCH = ITEMP -
IODEN
07
RS
(IODEN,3,
IODEN)
08
INCH = INCH - 4
09
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      DIMENSION ISC(1000,2)
      COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
      COMMON /H1/ IOSC,IOCH,IODEN,IODEL
```

07/22/71 AUTOFLOW CHART SET - RPSS PAGE 33

CHART TITLE - SUBROUTINE PBYTE(IW,IB,IOW,IOB)



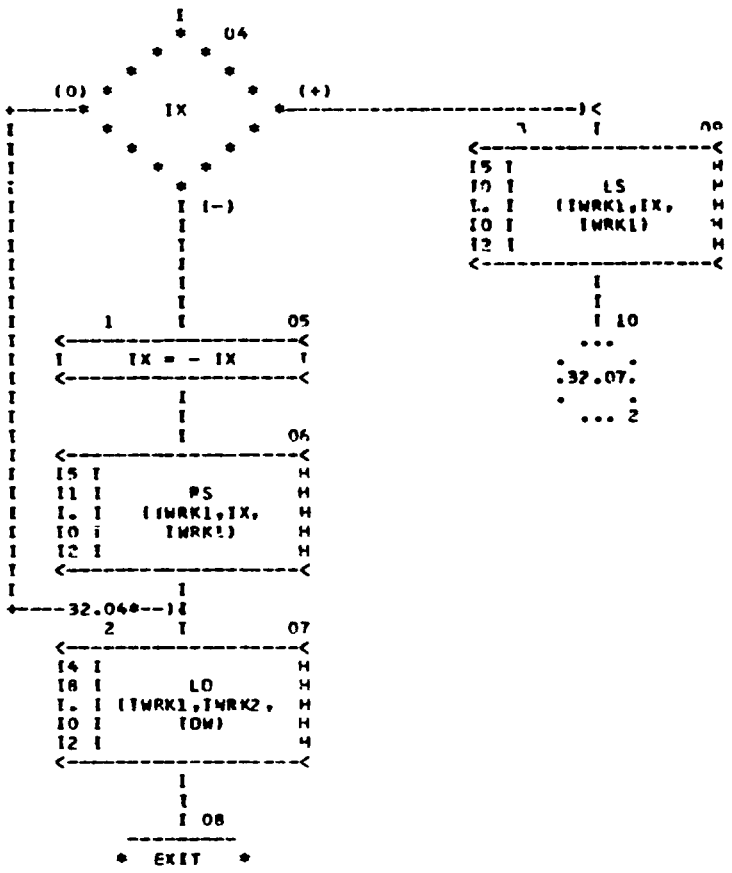
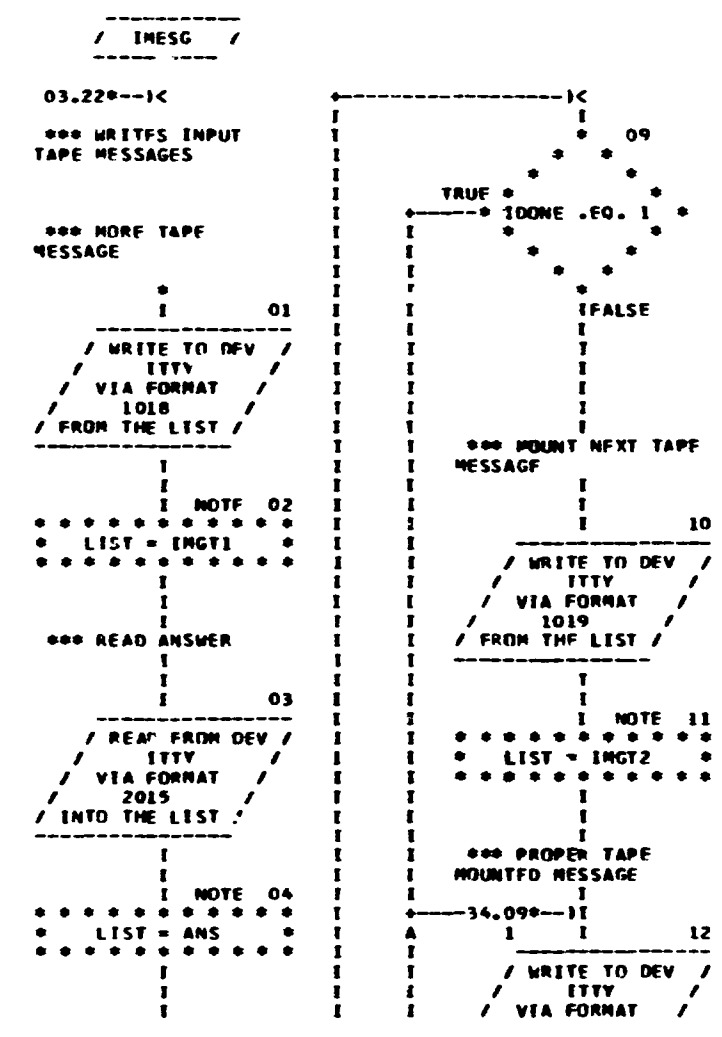

04
(0)
IX
(+)
(-)
1
05
IX = - IX
06
PS
(IWRK1,IX,
IWRK1)
32.04
2
07
LO
(IWRK1,IWRK2,
IOW)
08
EXIT
LS
(IWRK1,IX,
IWRK1)
10
32.07.
2

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
DIMENSION MASK1(3),MASK2(3),MASK5(6)
COMMON /MASK/ MASK1,MASK2
EQUIVALENCE (MASK5(1),MASK1(1))
```

CHART TITLE - SUBROUTINE IMESG



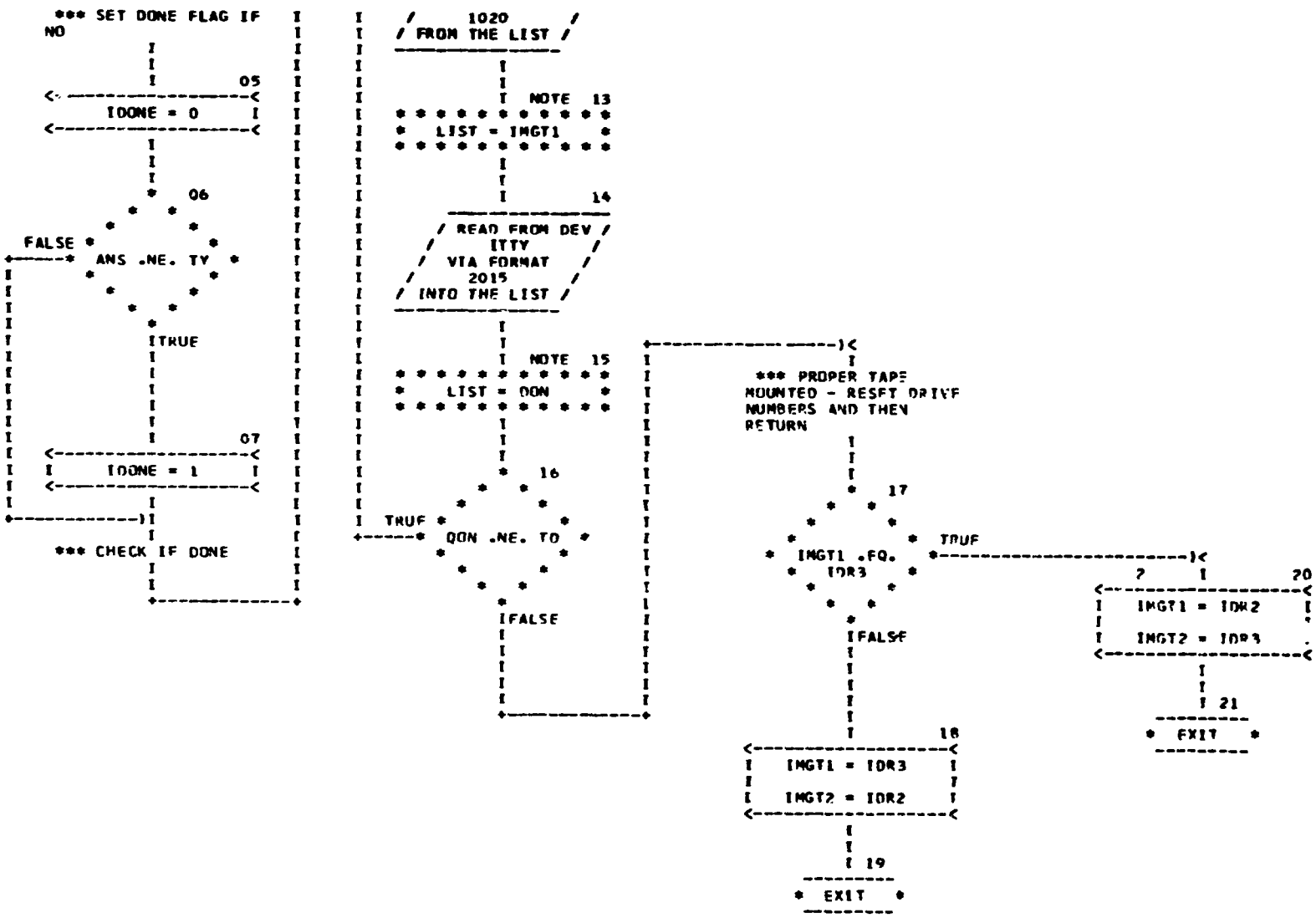
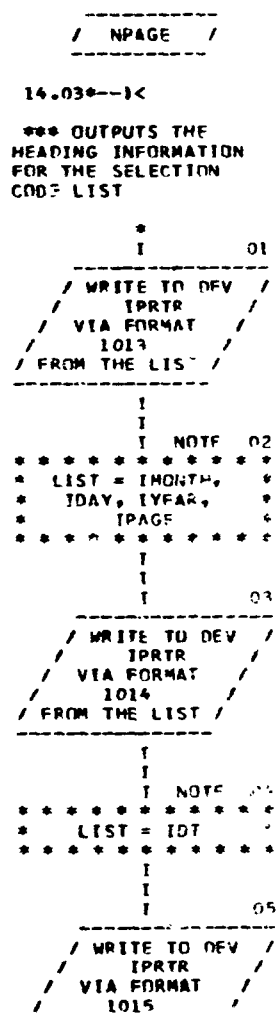

*** SET DONE FLAG IF NO
05
IDONE = 0
06
FALSE
ANS .NE. TY
TRUE
07
IDONE = 1
*** CHECK IF DONE
/ 1020 / FROM THE LIST /
NOTE 13
LIST = IMGT1
14
/ READ FROM DEV / ITTY / VIA FORMAT / 2015 / INTO THE LIST /
NOTE 15
LIST = QON
16
TRUE
QON .NE. TO
FALSE
*** PROPER TAPE MOUNTED - RESET DRIVE NUMBERS AND THEN RETURN
17
TRUE
IMGT1 .EQ. IDR3
FALSE
? 1 20
IMGT1 = IDR2
IMGT2 = IDR3
21
EXIT
18
IMGT1 = IDR3
IMGT2 = IDR2
19
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON /E1/ TN,TA,TF,TX,TY,TJ,TP,TT,TS,TR,TM
      COMMON /F1/ ITTY,IPT,IDT,INP,IPRTR
      COMMON /G1/ IDR1,IDR2,IDR3,IDEC
      COMMON /J1/ IMGT1,IMGT2,IDONE
1018  FORMAT (54H ARE THERE MORE INPUT TAPES FOR THIS RUN IN ADDITION T,
     6HO THAT/ 27H CURRENTLY MOUNTED ON DRIVE,1X,I1,15H  ENTER Y OR N,
2015  FORMAT (A1)
1019  FORMAT (32H MOUNT NEXT INPUT TAPE ON DRIVE ,I1)
1020  FORMAT (50H ENTER D IF PROPER INPUT TAPE IS MOUNTED ON DRIVE ,I1)
```

CHART TITLE - SUBROUTINE NPAGE



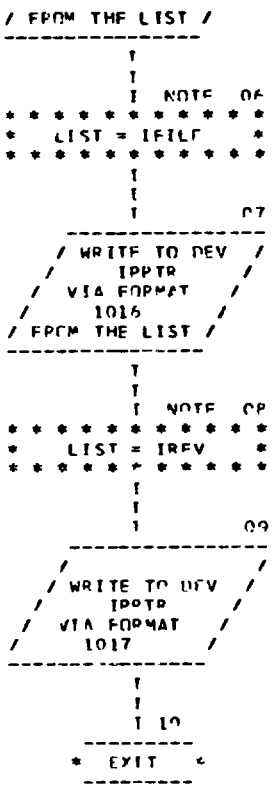

/ FROM THE LIST /
NOTE 06
LIST = IFILE
07
/ WRITE TO DEV /
IPPTR
/ VIA FORMAT /
1016
/ FROM THE LIST /
NOTE 08
LIST = IREV
09
/ WRITE TO DEV /
IPPTR
/ VIA FORMAT /
1017
10
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON /F1/ ITTY,IPT,IDT,INP,IPRTR
      COMMON /I1/ IMONTH,IDAY,IYEAR,IPAGE,IFILE,IREV
1013  FORMAT (47H1    RPS SELECTION CODE TABLE - INPUT SUMMARY  ,I2,1H/,
      I2,1H/,I2,10H     PAGE ,I2)
1014  FORMAT (14HO TAPE NUMBER ,I5)
1015  FORMAT (14H  FILE NUMBER ,I2)
1016  FORMAT (14H  REVISION    ,I2)
1017  FORMAT (53HO SELECTION CODE      CHANNEL      DENSITY      DELET,
      6HE FLAG)
```

CHART TITLE - SUBROUTINE PRNSTT

```
                 -----------
                 /  PRNSTT  /
                -----------

               09.06---)<

               *** PRINT RUN
               STATISTICS

                      *
                      I                 01
              -----------------------
              / WRITE TO DEV        /
             /     IPRTR           /
            /   VIA FORMAT        /
           /       1300          /
          / FROM THE LIST       /
         -----------------------
                      I
                      I
                      I  NOTE   02
          * * * * * * * * * * * *
          *   LIST = IMONTH,    *
          *     IDAY, IYEAR     *
          * * * * * * * * * * * *
                      I
                      I
                      I                 03
              -----------------------
              / WRITE TO DEV        /
             /     IPRTR           /
            /   VIA FORMAT        /
           /       1301          /
          / FROM THE LIST       /
         -----------------------
                      I
                      I
                      I  NOTE   04
          * * * * * * * * * * * *
          *    LIST = IDT       *
          * * * * * * * * * * * *
                      I
                      I
                      I                 05
              -----------------------
              / WRITE TO DEV        /
             /     IPRTR           /
            /   VIA FORMAT        /
           /       1302          /
          / FROM THE LIST       /
         -----------------------
                      I
```

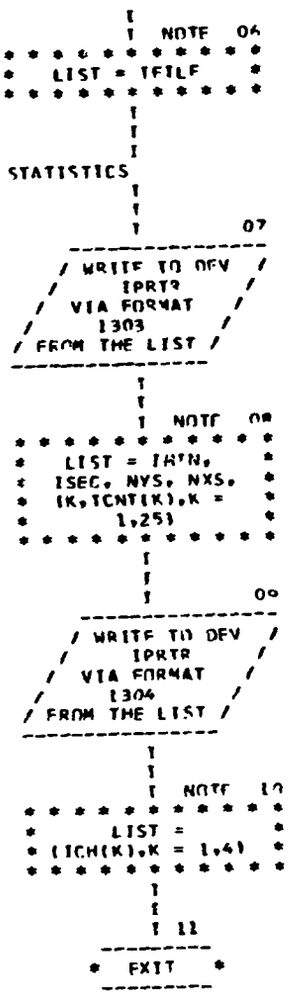
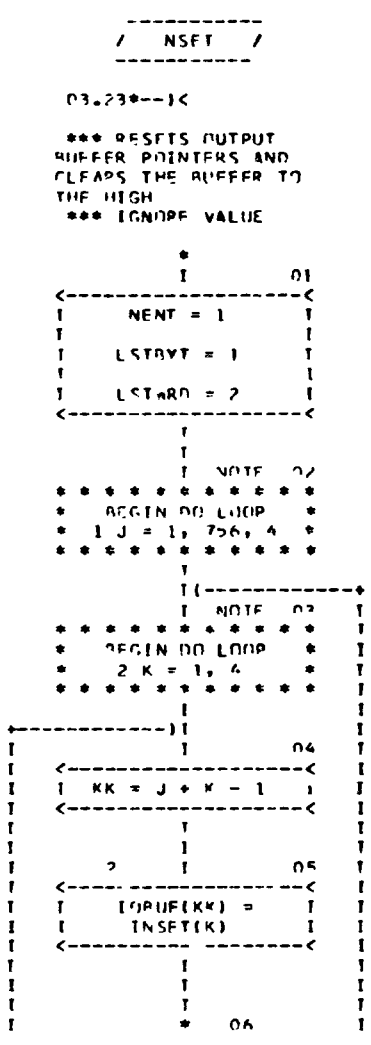

NOTE 06
LIST = TITLE
STATISTICS
07
WRITE TO DEV IPRTR VIA FORMAT 1303 FROM THE LIST
NOTE 08
LIST = IRUN, ISEC, NYS, NXS, (K,TCNT(K),K = 1,25)
09
WRITE TO DEV IPRTR VIA FORMAT 1304 FROM THE LIST
NOTE 10
LIST = (ICH(K),K = 1,4)
11
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      DIMENSION ICNT(25),ICH(4),ITBUF(740)
      COMMON /K1/ ICNT,ICH
      COMMON /TBUF/ ITBUF,NYS,NXS
      COMMON /L*/ IMIN,ISEC,IOFF
      COMMON /F1/ ITTY,IPT,IDT,INP,IPPTR
      COMMON /I1/ IMONTH,IDAY,IYEAR,IPAGE,IFILE,IPEV
1300  FORMAT (24HL RPS RUN STATISTICS OF ,I2,1H/,I2,1H/,I2)
1301  FORMAT (14HO TAPE NUMBER ,I5)
1302  FORMAT (14H  FILE NUMBER ,I2)
1303  FORMAT(//15H RUN STATISTICS//16H ELAPSED TIME = ,I6,1X,12HMINUTES A
     ND ,I6,1X,7HSECONDS////I4,1X,27HSCAN LINES INPUT CONTAINING,1X,I6,
     14H X COORDINATES////3PH X DESIGNATOR OF POINTS PROCESSED WERE////2
     5X,4HCODE,12X,5HCOUNT,25(/25X,I4,12X,I6))
1304  FORMAT (30HOCHANNEL TOTALS   CHANNEL 0 = ,I6,1X,13H CHANNEL 1 = ,I
     6,1X,13H CHANNEL 2 = ,I6,1X,13H CHANNEL 3 = ,I6)
```

CHART TITLE - SUBROUTINE NSET



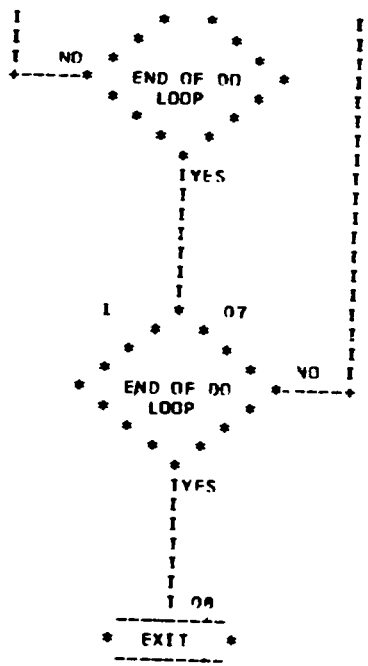
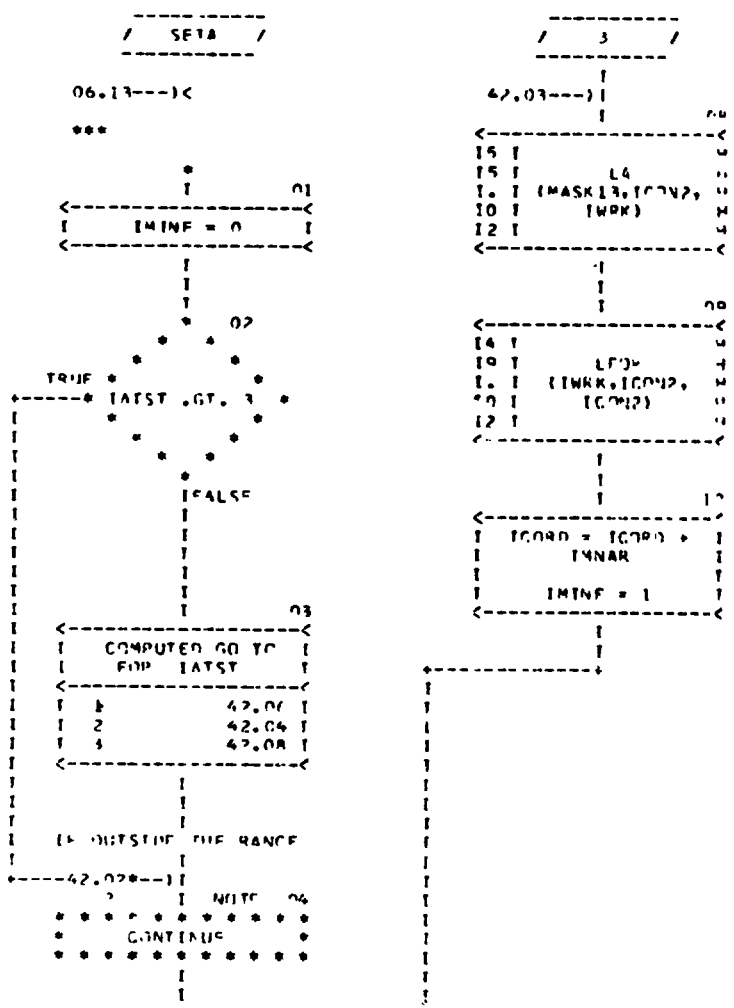

NO
END OF DO
LOOP
YES
1
07
NO
END OF DO
LOOP
YES
08
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      DIMENSION INSET(4),IOBUF(1364)
      COMMON /M1/ INSET,LSTWRD,LSTBYT,NENT,IOBUF
```

CHART TITLE - SUBROUTINE SETA



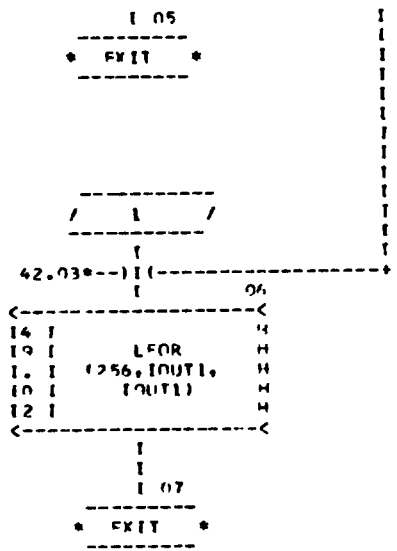

I 05
EXIT
1
42.03*--)I(--
I 06
LEOR
(256,IOUTI,
IOUTI)
I 07
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
DIMENSION MASK1(3),MASK2(3)
COMMON /GNR1/ ITYPE,ICORD,ICON1,ICON2,NX,IX,IFT,ISTAT
COMMON /AFIC/ IMINF,IMNAR
COMMON /MASK/ MASK1,MASK2
COMMON /IOUT/ IOUT1,IOUT2,IATST
EQUIVALENCE (MASK13,MASK1(3))
```

CHART TITLE - BLOCK DATA

Preceding page blank

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON /F1/ TN,TA,TF,TX,TY,TD,TP,TT,TS,TR,TM
      DATA TN,TA,TF/5HN    ,5HA    ,5HF    /
      DATA TX,TY,TD,TP/5HX    ,5HY    ,5HD    ,5HP    /
      DATA TT,TS,TR,TM/5HT    ,5HS    ,5HR    ,5HM    /
```

CHART TITLE - SUBROUTINE SETM(I)

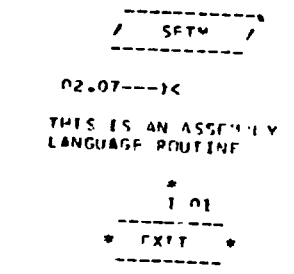


CHART TITLE - SUBROUTINE SETC(I)

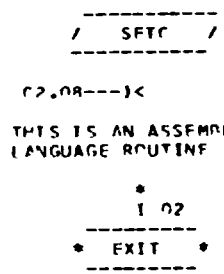


CHART TITLE - SUBROUTINE LO(I,J,K)

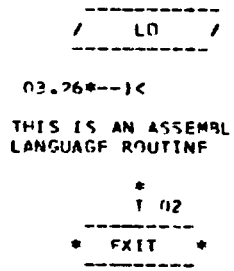


CHART TITLE - SUBROUTINE EXP(I,J,K)

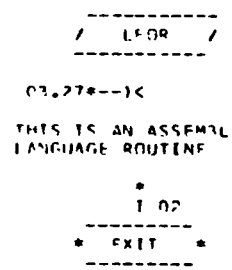


CHART TITLE - SUBROUTINE LS(I,J,K)

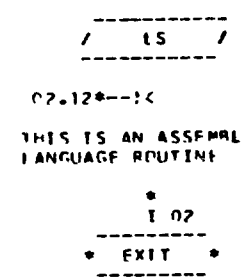


CHART TITLE - SUBROUTINE RS(I,J,K)

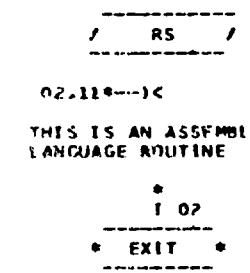


CHART TITLE - SUBROUTINE MGFRW(I,J,K,L,M)

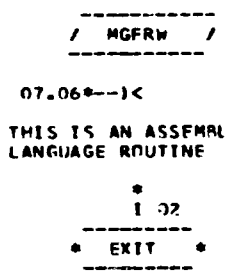


CHART TITLE - SUBROUTINE REOF(I)

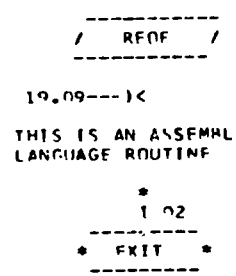


CHART TITLE - SUBROUTINE TIMF(I,J,K)

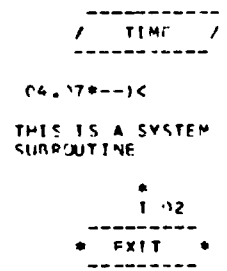


CHART TITLE - SUBROUTINE LA(I,J,K)