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

RATER PLOTTER SOFTWARE
Volume II - Software Documentation

United Aircraft Corporation

Approved for public release; distribution unlimited.
Best Available Copy
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.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automated Cartography</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Programming</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faster Data Processing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color Coding</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

UNCLASSIFIED
Security Classification
RASTER PLOTTER SOFTWARE

Volume II - Software Documentation

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

United Aircraft Corporation

Approved for public release; distribution unlimited.
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. DETTNER
Colonel, USAF
Chief, Intel & Recon Division

FOR THE COMMANDER:

FRED J. DIAMOND
Acting Chief, Plans Office
APPENDIX I

FEATURE IDENTIFICATION SOFTWARE SYSTEM OPERATING INSTRUCTIONS
APPENDIX I

TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Paragraph</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECTION I</td>
<td></td>
</tr>
<tr>
<td>DISK PROGRAM CATALOGING PROCEDURE</td>
<td></td>
</tr>
<tr>
<td>1. GENERAL</td>
<td>73</td>
</tr>
<tr>
<td>2. DECTAPE-TO-DISK LOADING INSTRUCTIONS</td>
<td>73</td>
</tr>
<tr>
<td>3. DISK-TO-CORE PDP-9 MONITOR LOADING INSTRUCTIONS</td>
<td>75</td>
</tr>
<tr>
<td>SECTION II</td>
<td></td>
</tr>
<tr>
<td>FEATURE IDENTIFICATION SOFTWARE SYSTEM PROGRAM</td>
<td></td>
</tr>
<tr>
<td>1. GENERAL</td>
<td>77</td>
</tr>
<tr>
<td>2. FISS PROGRAM LOADING INSTRUCTIONS</td>
<td>77</td>
</tr>
<tr>
<td>3. PROGRAM INITIALIZATION AND CONTROL INSTRUCTIONS</td>
<td>78</td>
</tr>
<tr>
<td>4. ERROR CONDITIONS AND PROGRAM STOPS</td>
<td>79</td>
</tr>
<tr>
<td>5. PROCESSING STATISTICAL DATA OUTPUTS</td>
<td>80</td>
</tr>
</tbody>
</table>
APPENDIX I

INDEX OF DIAGRAMS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-1</td>
<td>FISS Program Run</td>
<td>81</td>
</tr>
</tbody>
</table>
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 0000008
- Depress CONTINUE

(2) Upon the successful completion of system program transfer from DECtape 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
- 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 776378
(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.
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 01 and 31 to indicate original ACSD MIN AREA machine setting, e. g. 03 ↑P.
k. ENTER LINEAL NEIGHBORHOOD RANGE AS XX UNITS.

1. Enter a two-digit number greater than zero (e.g. 05 \( \uparrow \) 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. 02 \( \uparrow \) 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 01 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 \( \uparrow \) 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 \( \uparrow \) 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 00 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 000001

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

b. STOP 000002

PDP90GE 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 000003

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

d. STOP 000004

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 000005

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 000006

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

g. STOP 000007

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 printed on-line via the console teleprinter. These include the number of points input and output, convergences and divergences of features, linear/curvilinear transitions encountered, feature data sets output, and so on. Figure I-1 contains a typical example of the statistical data generated.
MONITOR V4E

SG 2
*S
MOUNT ACSD MANUSCRIPT TAPE ON DRIVE 3.*P
*M
MOUNT ACSD OVERLAY TAPE ON DRIVE 5.*P
*M
MOUNT SCRATCH TAPE ON DRIVE 6.*P
*M
ENTER RESOLUTION AS 2, 4, 5, OR 6. >

*M
ENTER MIN AREA SETTING AS XX ELEMENTS. >

*M
ENTER AREAL NEIGHBORHOOD RANGE AS XX UNITS. >

*M
ENTER MINIMAL FEATURE LENGTH AS XX RASTER UNITS. >

*M
ENTER GAP PARAMETER AS X RASTER UNITS. >

*M
ENTER BASE COLOR CODE AS XX. >

*M

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 V4F

$  

FIGURE 1-1. FISS PROGRAM RUN
APPENDIX II

FEATURE IDENTIFICATION SOFTWARE SYSTEM PROGRAM DESCRIPTION
### APPENDIX II

#### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Paragraph</th>
<th>SECTION I</th>
<th>FEATURE IDENTIFICATION SOFTWARE SYSTEM PROGRAM</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GENERAL</td>
<td></td>
<td>83</td>
</tr>
<tr>
<td>2</td>
<td>SYSTEM DESCRIPTION</td>
<td></td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>a. Object Program Deck Arrangement</td>
<td></td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>b. System Diagrams</td>
<td></td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>c. Logic Diagram Symbology</td>
<td></td>
<td>84</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Paragraph</th>
<th>SECTION II</th>
<th>DESCRIPTION OF SUBPROGRAMS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NEW FISS PROGRAMS</td>
<td></td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>a. UPDATE (Update Mother Data Set)</td>
<td></td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>b. DCDCOL (Decode Color)</td>
<td></td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>c. COLCLR (Collect Color)</td>
<td></td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>d. FILBUF (Fill Line Buffer)</td>
<td></td>
<td>91</td>
</tr>
<tr>
<td>2</td>
<td>MODIFIED CAST PROGRAMS</td>
<td></td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>a. CNTRL (Feature Identification Software System Control Program)</td>
<td></td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>b. TRMDS (Terminate Data Set)</td>
<td></td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>c. EXTRCT (Data Element Extraction)</td>
<td></td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>d. ALINE (Intersection Resolution)</td>
<td></td>
<td>99</td>
</tr>
</tbody>
</table>
SECTION II (CONT)

Paranaphragm | Page
---|---
3 | OBJECT-TIME SYSTEM AND SCIENCE LIBRARY ROUTINES 101
   | a. Object-Time System Routines .......... 101
   | b. Science Library Routines ............. 102

SECTION III
FISS FLOW DIAGRAMS

INDEX OF DIAGRAMS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>II-1</td>
<td>System Logic Flow Diagram ............... 85</td>
</tr>
<tr>
<td>II-2</td>
<td>Feature Identification Software System Block Diagram ... 86</td>
</tr>
<tr>
<td>II-3</td>
<td>Logic Diagram Symbology ................. 87</td>
</tr>
<tr>
<td>II-4</td>
<td>Flow Diagram of UPDATE .................. 104</td>
</tr>
<tr>
<td>II-5</td>
<td>Flow Diagram of DCDCOL .................. 105</td>
</tr>
<tr>
<td>II-6</td>
<td>Flow Diagram of COLCLR .................. 106</td>
</tr>
<tr>
<td>II-7</td>
<td>Flow Diagram of FILBUF .................. 108</td>
</tr>
<tr>
<td>II-8</td>
<td>Flow Diagram of CNTRL ................... 110</td>
</tr>
<tr>
<td>II-9</td>
<td>Flow Diagram of TRMDS ................... 114</td>
</tr>
<tr>
<td>II-10</td>
<td>Flow Diagram of EXTRCT .................. 117</td>
</tr>
<tr>
<td>II-11</td>
<td>Flow Diagram of ALINE ................... 118</td>
</tr>
</tbody>
</table>
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:

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

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 procedural 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 OIS 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 U-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 cor-
relation 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 -

      \[ L \quad LAC \quad DSCAD \]
      \[ L + 1 \quad JMS* \quad UPDATE \]
      \[ L + 2 \quad - - \quad \text{Next sequential instruction} \]

      where:

      \[ DSCAD \] - core address of data set block.

      (4) Outputs

      Disk -

      Updated mother data set block.

      (5) Referenced Subprograms

      DSKRW

      Preceding page blank
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 -

\[
\begin{align*}
L &: \text{ LAC} \\
L + 1 &: JMS* \\
L + 2 &: \text{SKP} \\
L + 3 &: \text{ARG} \\
L + 4 &: \text{\$} \\
\end{align*}
\]

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 contains 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 instruction

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

D;KRW
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 identifi-
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 000004" (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 000001" (STOP 000003).

(3) Inputs

(a) Calling Sequence -

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>JMS*</td>
<td>FILBUF</td>
</tr>
<tr>
<td>L + 1</td>
<td>JMP</td>
<td>.+3 Return location</td>
</tr>
<tr>
<td>L + 2</td>
<td>ARG</td>
<td></td>
</tr>
<tr>
<td>L + 3</td>
<td>JMP</td>
<td>End-of-data</td>
</tr>
<tr>
<td>L + 4</td>
<td>- -</td>
<td>return location</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Next sequential</td>
</tr>
</tbody>
</table>

instruction.

where:

ARG = address of buffer arguments

-92-
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 overflow (base manuscript tape) occurs: "STOP 000004".

The following message is output when line buffer overflow (overlay tape) occurs: "STOP 000007".

The following message is output when ACSD tape error (base manuscript tape) is encountered: "STOP 000001".

The following message is output when an ACSD tape error (overlay tape) is encountered: "STOP 000003".

(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 ENTS 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. Depending 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
"Tp"
"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
DC/DCOL
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 minima 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 -

\[
\begin{align*}
&L & \text{JMS}^* & \text{TRMDS} \\
&L + 1 & \text{SKP} & \text{Return location} \\
&L + 2 & \text{ARG} & \text{Next sequential instruction} \\
&L + 3 & \_\_\_ & \_\_\_ \\
\end{align*}
\]

where:

ARG = address of 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 -

\[
\begin{array}{ccc}
L & JMS^* & EXTRCT \\
L + 1 & JMP & .+3 \\
L + 2 & & ARG \\
L + 3 & JMP & EOD \\
L + 4 & - - & \\
\end{array}
\]

where:

\[
\text{ARG} = \text{address of buffer arguments.}
\]

where buffer arguments:

\[
\begin{align*}
\text{ARG1} &= \text{current line buffer address.} \\
\text{ARG2} &= \text{first data point (line center or begin area).} \\
\text{ARG3} &= \text{second data point-end area (areal data only).} \\
\text{ARG4} &= \text{entry or entries color code.}
\end{align*}
\]

(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 inter-
section point becomes the last data entry in all incoming data sets which are then termi-
nated 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 resolu-
tion. If the incoming data set is of valid length, the divergent intersection point be-
comes the first data entry in all outgoing data sets and the last data entry in the in-
coming 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

-100-
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.

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

The following PDP-9 Science Library 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.

<table>
<thead>
<tr>
<th>Routine Name</th>
<th>External Calls</th>
<th>Ref. Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>.DA, REAL</td>
<td>P. III-5</td>
</tr>
<tr>
<td>FLOAT</td>
<td>.DA, REAL</td>
<td>P. III-5</td>
</tr>
<tr>
<td>SIGN</td>
<td>.DA, REAL</td>
<td>P. III-5</td>
</tr>
<tr>
<td>.DA (General Get Argument)</td>
<td>None</td>
<td>P. III-8</td>
</tr>
<tr>
<td>INTEGE (Integer Arithmetic)</td>
<td>REAL</td>
<td>P. III-8</td>
</tr>
<tr>
<td>REAL (Real Arithmetic)</td>
<td>None</td>
<td>P. III-8</td>
</tr>
<tr>
<td>.CB (Short Get Argument)</td>
<td>None</td>
<td>P. III-9</td>
</tr>
</tbody>
</table>
SECTION III

FISS FLOW DIAGRAMS
FIGURE II - 5. FLOW DIAGRAM OF DCDCOL
FIGURE II-6. FLOW DIAGRAM OF COLCLR (CONCLUDED)
FIGURE II-7. FLOW DIAGRAM OF FILBUF
FIGURE II-7. FLOW DIAGRAM OF FILBUP (CONCLUDED)
FIGURE II-8. FLOW DIAGRAM OF CNTRL
FIGURE II - 8. FLOW DIAGRAM OF CNTRL (CONTINUED)
FIGURE II - 8  FLOW DIAGRAM OF CNTRL (CONCLUDED)
FIGURE II - 9. FLOW DIAGRAM OF TRMDS (CONTINUED)
FIGURE 7-10. FLOW DIAGRAM OF EXTRCT
APPENDIX III

RASTER PLOTTER SOFTWARE SYSTEM OPERATING INSTRUCTIONS
## APPENDIX III
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Paragraph</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SECTION I</strong>&lt;br&gt;DISK PROGRAM CATALOGING PROCEDURE</td>
<td></td>
</tr>
<tr>
<td>1. GENERAL</td>
<td>121</td>
</tr>
<tr>
<td>2. DECTAPE-TO-DISK LOADING INSTRUCTIONS</td>
<td>121</td>
</tr>
<tr>
<td>3. DISK-TO-CORE PDP-9 MONITOR LOADING INSTRUCTIONS</td>
<td>123</td>
</tr>
<tr>
<td><strong>SECTION II</strong>&lt;br&gt;RASTER PLOTTER SOFTWARE SYSTEM PROGRAM</td>
<td></td>
</tr>
<tr>
<td>1. GENERAL</td>
<td>125</td>
</tr>
<tr>
<td>2. RPSS PROGRAM LOADING INSTRUCTIONS</td>
<td>125</td>
</tr>
<tr>
<td>3. PROGRAM INITIALIZATION AND CONTROL INSTRUCTIONS</td>
<td>128</td>
</tr>
<tr>
<td>4. TAPE MODE INITIALIZATION INSTRUCTIONS</td>
<td>129</td>
</tr>
<tr>
<td>5. TAPE MODE POST-PROCESSING PRINTOUTS AND RUN TERMINATION PROCEDURES</td>
<td>133</td>
</tr>
<tr>
<td>6. MANUAL MODE INITIALIZATION INSTRUCTIONS</td>
<td>133</td>
</tr>
<tr>
<td>7. POST-PROCESSING PRINTOUTS AND RUN TERMINATION PROCEDURES</td>
<td>139</td>
</tr>
<tr>
<td>8. PROCESSING STATISTICAL DATA OUTPUTS</td>
<td>142</td>
</tr>
</tbody>
</table>
## APPENDIX III
### INDEX OF DIAGRAMS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>III-1</td>
<td>RPSS Program Load Map</td>
<td>127</td>
</tr>
<tr>
<td>III-2</td>
<td>RPSS Program Run (Tape Mode)</td>
<td>132</td>
</tr>
<tr>
<td>III-3</td>
<td>RPSS Program Run (Manual Mode)</td>
<td>140</td>
</tr>
</tbody>
</table>
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"

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 0000008
- 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 0000028
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 776378

(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 initialization 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 MTFV 7

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

Preceding page blank

-125-
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. **LS**

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 VAE

SA MTF7 7

$LOAD

LOADER V4A
>ABCDEFGHIJKLMNOPQRSTUVWXYZ
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
S1Q6

FIGURE III-1. RPSS PROGRAM LOAD MAP
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 previously 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 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

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 provide 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**

\[\text{PAUSE 000002}\]

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:

\[\text{P (CNTRL P)} - \text{Causes processing to continued from the point of interruption utilizing a new output tape file.}\]

m. **FIRST Y COORDINATE IS yyyy**

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 = wwww**

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.

o. **BAD READ AT IREAD = i'rrrr**

Message to the operator indicating the probability 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**

\[\text{TOTAL WRITES = wwww + 1}\]

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 REL NUMBER 0
END OF RUN - REMOVE ALL TAPES FROM DRIVES

*** TERMINATION ***

STOP 000000

FIGURE III-2. RPSs PROGRAM RUN; TAPE MODE

-132-
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**
   
   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 affirmative i.e., a Y. Operator response to this message is not required.

d. **END OF RUN - REMOVE ALL TAPES FROM DRIVE**

***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 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 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 ICSD. For example, if the machine setting was M.A. = 3, this would be entered as:

060003 (CR)

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

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 NUMERIC 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 (WYZNNNNMMMM)**

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

W Y Z 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.
\[ R = \text{Denotes that a set of input codes between the range specified by the NNNNN and MMMMM field inclusive, are to be mapped to the given output channel.} \]

\[ Z = \text{Output density designator field} \]

- Integer value ranging from 0-7

\[ \text{NNNNN} = \text{Input data extraction Key (or code)} \]

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

\[ \text{MMMMM} = \text{Terminal range extraction Key} \]

- Five-digit decimal number, right adjusted with leading zeroes 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

\[ \text{NS70003900001 (CR)} \]

(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

\[ \text{AS700002000001 (CR)} \]

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.

\[ \text{ES600005000001 (CR)} \]

This response says that the feature whose selection code is 510 is not to be output.
(d) To terminate the selection code entries for a particular channel, the following entry would be made:

\[ X \rightarrow (CR) \text{ where } \rightarrow = \text{space} \]

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.

1. **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 \[00001\]

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.

\[ \uparrow P \quad (\text{CNTRL P}) \]

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

\[ \uparrow C \quad (\text{CNTRL C}) \]

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, i., 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 IWRI T E = wwww**

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 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.

w. **TOTAL READS = rrrrr**

**TOTAL WRITES = wwww + 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 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 ASR35 if, and only if, the
MONITOR V4E

SG 6
15
MOUNT NEW SCRATCH ON DRIVE 5 ENTER D WHEN DONE

D
ENTER MONTH, DAY, YEAR AS 6 NUMERICS ... MMDDYY
078071
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 (WYZNNNNNMNNN)

NS50004700001
OK
X
CH NR =
0

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

N

FIGURE II-3. RPSS PROGRAM RUN: MANUAL MODE
ARE THERE MORE INPUT TAPES FOR THIS RUN IN ADDITION TO THI
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)

-141-
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 _:mmn, REEL NUMBER rr

   This message appears if the operator response to the previous query (7b) is negative, i.e., a no 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 data 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>
<thead>
<tr>
<th>Paragraph</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SECTION I</strong> &lt;br&gt;RASTER PLOTTER SOFTWARE SYSTEM PROGRAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>GENERAL</td>
<td>143</td>
</tr>
<tr>
<td>2</td>
<td>SYSTEM DESCRIPTION</td>
<td>143</td>
</tr>
<tr>
<td>a.</td>
<td>Object Program Deck Arrangement</td>
<td>143</td>
</tr>
<tr>
<td>b.</td>
<td>System Diagrams</td>
<td>144</td>
</tr>
<tr>
<td>c.</td>
<td>Logic Diagram Symbology</td>
<td>144</td>
</tr>
<tr>
<td><strong>SECTION II</strong> &lt;br&gt;DESCRIPTION OF SUBPROGRAMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>WORKER AND UTILITY PROGRAMS</td>
<td>153</td>
</tr>
<tr>
<td>a.</td>
<td>RPSS (Raster Plotter Software System Control Program)</td>
<td>153</td>
</tr>
<tr>
<td>b.</td>
<td>SCODE (Selection Code)</td>
<td>156</td>
</tr>
<tr>
<td>c.</td>
<td>MKENT (Make Table Entry)</td>
<td>158</td>
</tr>
<tr>
<td>d.</td>
<td>GNR (Get Next Record)</td>
<td>160</td>
</tr>
<tr>
<td>e.</td>
<td>SCAN (Scan Selection Code Table)</td>
<td>161</td>
</tr>
<tr>
<td>f.</td>
<td>SHLSRT (Shell Sort)</td>
<td>162</td>
</tr>
<tr>
<td>g.</td>
<td>SRCH (Search The Table)</td>
<td>162</td>
</tr>
<tr>
<td>h.</td>
<td>INTWDL (Input-Table Word Logic)</td>
<td>163</td>
</tr>
<tr>
<td>i.</td>
<td>OUTWDL (Output-Table Word Logic)</td>
<td>165</td>
</tr>
<tr>
<td>j.</td>
<td>PBYTE (Byte Placement)</td>
<td>165</td>
</tr>
<tr>
<td>Paragraph</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>k. IMSEG (Messages)</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>l. NPAGE (New Page)</td>
<td>168</td>
<td></td>
</tr>
<tr>
<td>m. PRNSTT (Print Run Statistics)</td>
<td>168</td>
<td></td>
</tr>
<tr>
<td>n. NSET (Set Buffer)</td>
<td>170</td>
<td></td>
</tr>
<tr>
<td>o. SETA (Set Area)</td>
<td>171</td>
<td></td>
</tr>
<tr>
<td>p. SETM (Set Masks)</td>
<td>171</td>
<td></td>
</tr>
<tr>
<td>q. SFTC (Set Constants)</td>
<td>172</td>
<td></td>
</tr>
<tr>
<td>r. BITS (Bit Routines)</td>
<td>173</td>
<td></td>
</tr>
<tr>
<td>s. MGFRW (Magnetic Tape Read/Write)</td>
<td>175</td>
<td></td>
</tr>
<tr>
<td>2 OBJECT-TIME SYSTEM AND SCIENCE LIBRARY ROUTINES</td>
<td>178</td>
<td></td>
</tr>
<tr>
<td>a. Object-Time System Routines</td>
<td>178</td>
<td></td>
</tr>
<tr>
<td>b. Science Library Routines</td>
<td>179</td>
<td></td>
</tr>
</tbody>
</table>

SECTION III
ASSEMBLY LANGUAGE SUBPROGRAM FLOW DIAGRAMS

SECTION IV
FORTRAN SUBPROGRAM AUTOFLOW DIAGRAMS
## Appendix IV
### Index of Diagrams

<table>
<thead>
<tr>
<th>Figure</th>
<th>Diagram Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV-1</td>
<td>System Logic Flow Diagram</td>
<td>145</td>
</tr>
<tr>
<td>IV-2</td>
<td>Raster Plotter Software System Block Diagram</td>
<td>146</td>
</tr>
<tr>
<td>IV-3</td>
<td>RPSS Tape Mode Block Diagram</td>
<td>149</td>
</tr>
<tr>
<td>IV-4</td>
<td>RPSS Manual Mode Block Diagram</td>
<td>150</td>
</tr>
<tr>
<td>IV-5</td>
<td>Logic Diagram Symbology</td>
<td>151</td>
</tr>
<tr>
<td>IV-6</td>
<td>Flow Diagram of SETM</td>
<td>182</td>
</tr>
<tr>
<td>IV-7</td>
<td>Flow Diagram of SETC</td>
<td>182</td>
</tr>
<tr>
<td>IV-8</td>
<td>Flow Diagram of BITS</td>
<td>183</td>
</tr>
<tr>
<td>IV-9</td>
<td>Flow Diagram of MGFRW</td>
<td>184</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Equipment Item Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DEC PDP-9 Control Processing Unit (CPU)</td>
</tr>
<tr>
<td>8</td>
<td>1096 18-bit Word Core Memory Modules</td>
</tr>
<tr>
<td>1</td>
<td>10 CPS, Console Keyboard/Teleprinter</td>
</tr>
<tr>
<td>3</td>
<td>7-Track, 45IPS, 556/800 BPI IBM Compatible Magnetic Tape Units W/Controller</td>
</tr>
<tr>
<td>4</td>
<td>DECTape Units W/Controller</td>
</tr>
<tr>
<td>1</td>
<td>DEC PDP-9 Paper Tape Reader</td>
</tr>
<tr>
<td>1</td>
<td>DEC PDP-9 Paper Tape Punch</td>
</tr>
<tr>
<td>1</td>
<td>1 Million 18-Bit Word Disk File W/Controller</td>
</tr>
<tr>
<td>1</td>
<td>ANELEX, 1000 LPM Line Printer</td>
</tr>
</tbody>
</table>

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.

Preceding page blank
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 procedural 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 (CONTINUED)
FIGURE IV - 2. RASTER PLOTTER SOFTWARE SYSTEM BLOCK DIAGRAM (CONCLUDED)
FIGURE IV - 3. RPSS TAPE 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 program then reads raster scan lines from the input tape by calling GNR, locates and extracts 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, 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 counter 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. SCODF 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 EXTERNALLY 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 WRITE = ________"

"TOTAL WRITE = ________ + 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: \[\text{LS, LO, LEOR, LA, RS}\]

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 6 TO END SELECTION CODE INPUT EACH TIME CH NR = IS TYPED"

"CH NR = "

"ENTER CODES (WYZNNMNMMMN"

"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 (ISC) 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.

-158-
(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 -

WYZNNN
NNNNNNNM

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

(4) Outputs

(a) Teleprinter -

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

WYZNNNNMNMNM

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 MGFRW 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 -

-160-
"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

≠ -1 IVAL did previously appear in the table and K contains its location with the table
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 labeled common Bl and the location is passed in the argument K. The selection code is immediately placed into the first word as an 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, LI, OR
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
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 TAPE 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.
1. 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
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 FRNSTT

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 COORDINATES"

"X DESIGNATOR OF POINTS PROCESSED WERE"

<table>
<thead>
<tr>
<th>CODE</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

-169-
"CHANNEL TOTALS: CHANNEL = --------- 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 IOEÜF. The LSTBYT and LSTWRD parameters are initialized to the original starting state; 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 center 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 770000, 007700, 000077, 007777, 770077, 777700. These then are used as masks to extract desired information, a byte at a time, from the data words used by the RP3 system.

(3) Inputs

(3) 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.

ω: 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 RP3S to preset the output buffer area. The mask is four words long of the form 043700, 000437, 043700, 370000.

The masks are four, 18-bit words which form three consecutive and identical 24-bit words that is 04370000. 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

(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.
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

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:

<table>
<thead>
<tr>
<th>Status Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Physical end-of-tape</td>
</tr>
<tr>
<td>2</td>
<td>End-of-file detected</td>
</tr>
<tr>
<td>3</td>
<td>Normal Termination</td>
</tr>
<tr>
<td>4</td>
<td>Parity error</td>
</tr>
<tr>
<td>5</td>
<td>Improper termination</td>
</tr>
<tr>
<td>6</td>
<td>Wrong word length</td>
</tr>
<tr>
<td>7</td>
<td>Error flag</td>
</tr>
</tbody>
</table>

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 global symbols 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"; or the number DEC-9A-KFZA-D.

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Function</th>
<th>Ref. Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUXIO</td>
<td>To process the auxiliary I/O statements; BACK SPACE, Rewind, in FORTRAN IV program and subprograms.</td>
<td>P. 11-8</td>
</tr>
<tr>
<td>BCDIO</td>
<td>To process formatted READ/WRITE statements in FORTRAN IV programs and subprograms.</td>
<td>P. 11-3</td>
</tr>
<tr>
<td>FIOPS</td>
<td>FIOPS provides the necessary call to IOPS (Input-Output Processing System) required by all FORTRAN I/O statements.</td>
<td>P. 11-10</td>
</tr>
<tr>
<td>GOTO (.GO)</td>
<td>To compute the index of a computed GO TO.</td>
<td>P. 11-14</td>
</tr>
<tr>
<td>OTSER (.ER)</td>
<td>To announce an error on the teletype.</td>
<td>P. 11-18</td>
</tr>
<tr>
<td>PAUSE (.PA)</td>
<td>To process the PAUSE statement.</td>
<td>P. 11-16</td>
</tr>
<tr>
<td>SPMSG (.SP)</td>
<td>To print the octal coded number with STOP and PAUSE.</td>
<td>P. 11-17</td>
</tr>
<tr>
<td>SS</td>
<td>To calculate the array element address.</td>
<td>P. 11-12</td>
</tr>
<tr>
<td>Program Name</td>
<td>Function</td>
<td>Ref. Manual</td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>STOP (.ST)</td>
<td>To process the STCP statement and return control to the monitor.</td>
<td>P. 11-15</td>
</tr>
<tr>
<td>TIME</td>
<td>To provide the ability to record elapsed time in minutes and seconds.</td>
<td>P. 11-21</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Routine Name</th>
<th>External Cells</th>
<th>Ref. Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>.CB (Short Get Argument)</td>
<td>None</td>
<td>P. III-9</td>
</tr>
<tr>
<td>.DA (General Get Argument)</td>
<td>None</td>
<td>P. III-8</td>
</tr>
<tr>
<td>INTEGE (Integer Arithmetic)</td>
<td>REAL</td>
<td>P. III-8</td>
</tr>
<tr>
<td>REAL (Real Arithmetic)</td>
<td>None</td>
<td>P. III-8</td>
</tr>
</tbody>
</table>
Figure IV - 6. Flow Diagram of SETM

Figure IV - 7. Flow Diagram of SETC
FIGURE IV - b  FLOW DIAGRAM OF BITS
SECTION V

FORTRAN SUBPROGRAM AUTOFLOW DIAGRAMS
### INPUT LISTING

<table>
<thead>
<tr>
<th>CARD NO</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DIMENSION TSC(1000,21),A(9)</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION MASK1(3),MASK2(3),MASK3(6)</td>
</tr>
<tr>
<td>3</td>
<td>DIMENSION ITBUF(750)</td>
</tr>
<tr>
<td>4</td>
<td>DIMENSION INSFT(41),IOBUF(1141)</td>
</tr>
<tr>
<td>5</td>
<td>DIMENSION ICNT(25),ICN(4)</td>
</tr>
<tr>
<td>6</td>
<td>COMMON /AI/,ISC,ITM,TIC,ITC,INSC,MAXSC</td>
</tr>
<tr>
<td>7</td>
<td>COMMON /AI/,INSC,INCH,INPH,INPL</td>
</tr>
<tr>
<td>8</td>
<td>COMMON /GI/,IEF,IEFCH,IEFREN</td>
</tr>
<tr>
<td>9</td>
<td>COMMON /DI/,IDE*,ISC,ISCE,INNEN</td>
</tr>
<tr>
<td>10</td>
<td>COMMON /EI/,TN,TM,TE,TX,TT,TY,TT,IX,RY</td>
</tr>
<tr>
<td>11</td>
<td>COMMON /FI/,TXY,ITY,ITI,INT,INPTER</td>
</tr>
<tr>
<td>12</td>
<td>COMMON /GI/,IDRI,IMDR2,IDP3,IDEC</td>
</tr>
<tr>
<td>13</td>
<td>COMMON /HI/,INSC,ICCH,INPH,INPL</td>
</tr>
<tr>
<td>14</td>
<td>COMMON /JI/,IMTF,IMTV,ITCAP,ITAGE,ITFILE,ITPV</td>
</tr>
<tr>
<td>15</td>
<td>COMMON /JI/,IMGT1,IMGT2,IMPNE</td>
</tr>
<tr>
<td>16</td>
<td>COMMON /MASK/,MASK1,MASK2</td>
</tr>
<tr>
<td>17</td>
<td>COMMON /GTMP/,ITCP,ITCN,ITCN2,IX,IXT,STAT</td>
</tr>
<tr>
<td>18</td>
<td>COMMON /ITBF/,ITBUF,NSY,NXS</td>
</tr>
<tr>
<td>19</td>
<td>COMMON /AILC/,IMINF,IMWHR</td>
</tr>
<tr>
<td>20</td>
<td>COMMON /IOFT/,IVOUT,IVOUT2,IVSTAT</td>
</tr>
<tr>
<td>21</td>
<td>COMMON /K1/,ICNT,ICH</td>
</tr>
<tr>
<td>22</td>
<td>COMMON /L1/,IMIN,ISCI,INF</td>
</tr>
<tr>
<td>23</td>
<td>COMMON /M1/,INSET,LISTRD,LISTTY,NENT,ITHUF</td>
</tr>
<tr>
<td>24</td>
<td>EQUVALENCE (MASK1(1),MASKS(1))</td>
</tr>
<tr>
<td>25</td>
<td>EQUVALENCE (MASK1(1),MASK1(1)),(MASK2(1),MASK2(1))</td>
</tr>
<tr>
<td>26</td>
<td>EQUVALENCE (MASK2(1),MASK2(1)),(MASK2(1),MASK2(1))</td>
</tr>
</tbody>
</table>
27     C
28     C  INITIALIZATION
29     C
30      ISWCH = 12
31      IWRITE = 0
32      ITRY = 8
33      IOT = 4
34      INT = 2
35      IMP = 1
36      IMPRO = 1
37      IPERM = 6
38      IORD = 6
39      IORD1 = 7
40      IODEC = 3
41      ITLIF = 1
42      N0 = 0
43      IV = 5
44      IFT = 7
45      ISTAT = 0
46      MVS = 0
47      NXS = 0
48      IMY = 0
49      ISPEC = 0
50      ITMIN = 0
51      ITSPEC = 0
52      ITOFF = 0
53      ITMAX = 1000
54      ITCWW = 1
<table>
<thead>
<tr>
<th>CARD NO</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>54</td>
<td>MENS = 1</td>
</tr>
<tr>
<td>55</td>
<td>MASS = 00000</td>
</tr>
<tr>
<td>57</td>
<td>ITY = 4</td>
</tr>
<tr>
<td>58</td>
<td>NUMS = 746</td>
</tr>
<tr>
<td>59</td>
<td>INGT1 = 100</td>
</tr>
<tr>
<td>60</td>
<td>INGT2 = 100</td>
</tr>
<tr>
<td>61</td>
<td>REFV = 0</td>
</tr>
<tr>
<td>67</td>
<td>CALL SET(1)</td>
</tr>
<tr>
<td>68</td>
<td>CALL SET(1)</td>
</tr>
<tr>
<td>69</td>
<td>SET UP THE AIPARY CONSTANTS</td>
</tr>
<tr>
<td>70</td>
<td>CALL PRYCLNKST(1),1,1PUNW,2</td>
</tr>
<tr>
<td>71</td>
<td>LPI = MSAL</td>
</tr>
<tr>
<td>72</td>
<td>CALL SRTICOL,1,1EFL</td>
</tr>
<tr>
<td>73</td>
<td>CALL LGT,1,1,1PUNW</td>
</tr>
<tr>
<td>74</td>
<td>AUP = 1</td>
</tr>
<tr>
<td>75</td>
<td>OUTPUT SCRATCH MOUNT MESSAGE</td>
</tr>
<tr>
<td>76</td>
<td>I WRITE (ITY,1000) 1001</td>
</tr>
<tr>
<td>77</td>
<td>1000 FORMAT 1,2,3,[MOUNT NEW SCRATCH ON DRIVE 11,14M ENTER 0 WHEN,</td>
</tr>
<tr>
<td>78</td>
<td>1541 NONX)</td>
</tr>
<tr>
<td>79</td>
<td>READ (ITY,7,2000) ALL1</td>
</tr>
<tr>
<td>80</td>
<td>2000 FORMAT (ALL)</td>
</tr>
<tr>
<td>81</td>
<td>C</td>
</tr>
<tr>
<td>82</td>
<td>C CHECK</td>
</tr>
<tr>
<td>83</td>
<td>C</td>
</tr>
</tbody>
</table>
IF (A(1) .NE. 0) GO TO 1

C DATM MESSAGE AND SAVE IT

WRITE (LTY,1001)
1001 FORMAT (40H ENTER MONTH, DAY, YEAR AS 6 NUMERICS ... YYYY)
READ (LTY,2001) INMONTH, IDAY, IYEAR

2001 FORMAT (312)

C TAP TO MESSAGE AND SAVE IT

WRITE (LTY,1102)
1102 FORMAT (40H ENTER 6 DIGIT TAPE IN NUMBER)
REAL (LTY,2002) INT

2002 FORMAT (15)

C

C INPUT 48BRT MESSAGE

WRITE (LTY,1003) 1002
1003 FORMAT (30H INSERT 1ST REEL OF INPUT TAPE ON DRIVE 11)
WRITE (LTY,1101)

1101 FORMAT (40H ENTER 6 DIGIT MINTUM AREA IN PROPER RESOLUTION)
READ (LTY,2100) IYMA6
2100 FORMAT (16)

C

C PRINT OF 48BRT MESSAGE 21x TAPE + P = MANUAL

C

DO 700 J = 1,75
07/22/71

INPUT LISTING

AUTOFLOW CHART SFT - EPSS

CARD NO **** CONTENTS ****

111 TO 0 (CNT(J)) = 0
112 00 TOI J = 1.4
113 TOI (CH(J)) = 0
114 50 WRITE (TTY, 1100)
115 1100 FORMAT (504 D) YAU WISH TO SUPPLY SELETION CODES EXTERNALLY / 174
116 1 ENTER Y OR N
117 READ (TTY, 2000) AIJ
118 IF (AIJ .EQ. TV) GO TO 21
119 IF (AIJ .LE. TV) GO TO 50
120 MODS = 1
121 GO TO 200
122 21 CALL SCODE
123 MNOF = 2
124 200 CONTINUE
125 C
126 C START PROCESSING OF DATA
127 C
128 CALL IMSG
129 CALL HSET
130 ICR1 = -1
131 IFT = 0
132 IMINF = 0
133 CALL LSW(IASEDFN, 3, IWRK2)
134 CALL _L(IWRK2, IAECH, IWRK2)
135 CALL LEOR(IWRK2, 4, IWRK2)
136 CALL LS(IWRK2, 4, IAECH)
137 INUT1 = 0
138 INUT2 = 0
139      TATST = 0
140      C
141      CALL TIME (IMIN, ISEC, ITIME)
142      C
143      300 IF (IIMIN .NE. 1) CALL G0
144      IF (IITYPE .LT. 2) GO TO 301
145      IF (IIOINF .EQ. 1) GO TO 300
146      CALL IMSG
147      GO TO 300
148      C
149      C  NOT FOR D.E.FAT - CHECK IF Y OF Y
150      C  IF IFT = 0 THEN FIRST Y
151      C
152      301 IF (IITYPE .EQ. 1) GO TO 302
153      IF (IIOINF .EQ. TCOPY) GO TO 303
154      IF (IFT .NE. 0) GO TO 300
155      IFT = 1
156      GO TO 300
157      C
158      C  PROCESS THF Y
159      C
160      302 IWRK1 = 0
161      IWRK2 = 0
162      C
163      C  CHECK MODF: I=TAPE 1*SF TABLE
164      C
165      IF (MODF .EQ. 1) GO TO 300
166      C,
<table>
<thead>
<tr>
<th>CARD NO</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>167</td>
<td>SC MODE - SET THE SELECTION CODE</td>
</tr>
<tr>
<td>168</td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>CALL LA(INW1,INW2,INW3)</td>
</tr>
<tr>
<td>170</td>
<td>CALL KS(INW1,INW2)</td>
</tr>
<tr>
<td>171</td>
<td></td>
</tr>
<tr>
<td>172</td>
<td>IS IT IN SC TABLE?</td>
</tr>
<tr>
<td>173</td>
<td></td>
</tr>
<tr>
<td>174</td>
<td>CALL SRCH(INW1,INDEX)</td>
</tr>
<tr>
<td>175</td>
<td>IF (INDEX) 304,303,303</td>
</tr>
<tr>
<td>176</td>
<td></td>
</tr>
<tr>
<td>177</td>
<td>NOT IN TABLE, CHECK ALL FLSF</td>
</tr>
<tr>
<td>178</td>
<td></td>
</tr>
<tr>
<td>179</td>
<td>J00 IF (AFF + NF = 1) GO TO 300</td>
</tr>
<tr>
<td>190</td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>ALL FLSF ON = SET IT UP</td>
</tr>
<tr>
<td>192</td>
<td></td>
</tr>
<tr>
<td>193</td>
<td>IFUF1 = IACCOM</td>
</tr>
<tr>
<td>194</td>
<td></td>
</tr>
<tr>
<td>195</td>
<td>%04 INIT2 = TCORD0</td>
</tr>
<tr>
<td>196</td>
<td>CALL LA(INW1,INW2,NST)</td>
</tr>
<tr>
<td>197</td>
<td>ICNT(INST) = ICNT(INST) + 1</td>
</tr>
<tr>
<td>198</td>
<td>CALL LA(INR1,INR2,INKCH)</td>
</tr>
<tr>
<td>199</td>
<td>CALL KS(INKCH,6,INWCH)</td>
</tr>
<tr>
<td>200</td>
<td>IC(MWCH+1) = ICH(INKCH+1) + 1</td>
</tr>
<tr>
<td>201</td>
<td>GO TO 305</td>
</tr>
<tr>
<td>202</td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>IN TABLE - CHECK DFLEET FLAG</td>
</tr>
<tr>
<td>204</td>
<td></td>
</tr>
<tr>
<td>205</td>
<td>J03 INW4 = ISC(INDEX,2)</td>
</tr>
</tbody>
</table>
CALL LAT1(WK4,MASK13,WK23)

IF (WK23 .LT. 1) GO TO 370

C

C DELETE OFF - SET IT UP

C

CALL LAT1(MASK, MASK, (OUTL)

GO TO 104

C

TAPE MODE

C

360 CALL LAT1C472,MASK1Z,(OUTL)

GO TO 104

C

MAIN STORAGE LOOP

C

104 IF (LIST1 .LT. 0) CALL SFTU

150 I5WCH = 313

170 IF (NEXT .LT. 57) GO TO 107

190 NEXT = NEXT + 1

C

PUT IN FIRST WORD

C

214

215 LSTRTY = LSTRTY + 1

216 IF (LSTRTY .LT. 4) GO TO 360

217 LSTRWP = LSTRWP + 1

220 LSTRY = 1

304 CALL PRTF(OUTL,?,INBUF,LSTWP1,LSTRTY)
<table>
<thead>
<tr>
<th>CARD NO</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>221</td>
<td>C NOW 3 RYTES OF SECOND</td>
</tr>
<tr>
<td>224</td>
<td>C</td>
</tr>
<tr>
<td>225</td>
<td>(LSTMOV ,NE. 3) GO TO 300</td>
</tr>
<tr>
<td>226</td>
<td>LSTWRK = LSTWRK + 1</td>
</tr>
<tr>
<td>227</td>
<td>I0BUF(LSTWRK) = I0UT2</td>
</tr>
<tr>
<td>228</td>
<td>GO TO 310</td>
</tr>
<tr>
<td>229</td>
<td>C</td>
</tr>
<tr>
<td>230</td>
<td>C NOT EVEN - ON IT 4V RYTES</td>
</tr>
<tr>
<td>231</td>
<td>C</td>
</tr>
<tr>
<td>232</td>
<td>309 DO 311 KK = 1,3</td>
</tr>
<tr>
<td>233</td>
<td>LSTMOV = LSTMOV + 1</td>
</tr>
<tr>
<td>234</td>
<td>IF (LSTMOV ,NE. 4) GO TO 312</td>
</tr>
<tr>
<td>235</td>
<td>LSTWRK = LSTWRK + 1</td>
</tr>
<tr>
<td>236</td>
<td>LSTMOV = 1</td>
</tr>
<tr>
<td>237</td>
<td>312 CALL PDYC(l0UT2, ,K, I0BUF(LSTWRK), LSTMOV), LSTMOV</td>
</tr>
<tr>
<td>238</td>
<td>311 CONTINUE</td>
</tr>
<tr>
<td>239</td>
<td>C</td>
</tr>
<tr>
<td>240</td>
<td>310 CON' , UF</td>
</tr>
<tr>
<td>241</td>
<td>IF (LSWHK - 114) 313, 314, 317</td>
</tr>
<tr>
<td>242</td>
<td>C</td>
</tr>
<tr>
<td>243</td>
<td>313 IF ((ITYPE ,EQ. 7) ,NR. (ITYPE ,EQ. 4)) GO TO 300</td>
</tr>
<tr>
<td>244</td>
<td>320 CALL NSFT</td>
</tr>
<tr>
<td>245</td>
<td>370 ICURV = ICORD</td>
</tr>
<tr>
<td>246</td>
<td>GO TO 300</td>
</tr>
<tr>
<td>247</td>
<td>C</td>
</tr>
<tr>
<td>248</td>
<td>C BUFFER FULL</td>
</tr>
<tr>
<td>249</td>
<td>C</td>
</tr>
<tr>
<td>250</td>
<td>307 I0UT15 = I0UT1</td>
</tr>
</tbody>
</table>
251  INOUT2 = INOUT2
252  INOUT2 = IUNE
253  INUM1 = INUMC
254  ISWCH = 314
255  CALL PRTE(IUNMC,2,IAUF111,1)
256  GO TO 243
257  CALL PRTE(IURV,IK-1,IAUF111,1)
258  CALL PRTE(IURV,1,IAUF111,1)
259  GO TO 317
260  C
261  310 INOUT2 = IENL
262  GO TO 310
263  314 ISTAT = 0
264  CALL VIRMAR(INR,NKROD,IUAUF111,1,ISTAT)
265  WRITE = WRITE + 1
266  IF ISTAT .GT. 4 WRITE (6,222) WRITE
267  222 FORMAT (23H MADE WRITE AT WRITE = ,15)
268  IF ISTAT .EQ. 1 OR ISTAT .EQ. 4) GO TO 269
269  IF ISTAT .NE. 1) GO TO 318
270  C
271  C END OF IAP1
272  C
273  ISWCH = 317
274  GO TO 304
275  317 CONTINUE
276  IOFF = 1
277  IIMIN = IIMIN + IMIN
278  IIFSC = IIFSC + IIFSC
07/22/71  INPUT LISTING

CARD NO  **** CONTENTS  ****
270  IF (I1SFC >LT. 60) GO TO 400
280  IIMIN = IIMIN + 1
290  I1SEC = I1SEC - 60
291  600 CONTINUE
292  I0FF = 0
293  REWIND I0R1
294  WRITE (ITTY,17490) I0R1
295  1050 FORMAT (40H AT PAUSE 2, MOUNT NEW SCRATCH ON DRIVE +11,14M AND CD
296  I1NTIME)
297  C
298  PAUSE 2
299  CALL TIME (IIMIN,I1SEC,I0FF)
300  C
301  I16 IF (ITYPE .LT. 2) GO TO 301
302  IF (ITYPE .LT. 4) GO TO 120
303  CALL NSFT
304  I1OUT1 = I1OUT1
305  I1OUT2 = I1OUT2
306  GO TO 305
307  C
308  390 I1OUT2 = IEUM
309  WRITE (8,SNOO00) IWRITE
310  5000 FORMAT (16H TOTAL WRITES = ,15,4M + 1)
311  IWRITE = 0
312  GO TO 318
313  C
314  391 GO TO 400
315  C
C OUTPUT TAPE ERRNR
C
350 STOP 12
C
C PRINT OUTPUT STATISTICS
C
400 I0FF = 1
IMIN = IMIN + IIMIN
ISEC = ISEC + IISFC
IF ISEC .LT. 60 GO TO 7000
IMIN = IMIN + 1
ISEC = ISEC - 60
7000 CONTINUE
IMIN = 0
IISFC = 0
CALL PRNSTT
I0FF = 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) ALL
IF (ALL .EQ. 'Y') GO TO 401
C
C CONTINUE RUN - MULTIFILE MESSAGE
C
<table>
<thead>
<tr>
<th>CARD NO</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>335</td>
<td>WRITE (ITY,1201)</td>
</tr>
<tr>
<td>336</td>
<td>1201 FORMAT (A47M,D30M) A11 (YOU WISH TO STACK OUTPUT FILES (Y OR N))</td>
</tr>
<tr>
<td>337</td>
<td>READ (ITY,2000) A11</td>
</tr>
<tr>
<td>338</td>
<td>IF (A11,.EQ.,'Y') GO TO 402</td>
</tr>
<tr>
<td>339</td>
<td>C</td>
</tr>
<tr>
<td>340</td>
<td>C NEW OUTPUT TAPE REQUIRED</td>
</tr>
<tr>
<td>341</td>
<td>C</td>
</tr>
<tr>
<td>342</td>
<td>IFLG2 = 1</td>
</tr>
<tr>
<td>343</td>
<td>401 REWIND IMG2</td>
</tr>
<tr>
<td>344</td>
<td>CALL NSFRM(1DR1,NRWRDS,NRUF1,2,INSTA)</td>
</tr>
<tr>
<td>345</td>
<td>REWIND 1DR1</td>
</tr>
<tr>
<td>346</td>
<td>C</td>
</tr>
<tr>
<td>347</td>
<td>IMG1 = IOR2</td>
</tr>
<tr>
<td>348</td>
<td>IMG2 = IOR4</td>
</tr>
<tr>
<td>349</td>
<td>C</td>
</tr>
<tr>
<td>350</td>
<td>C TAPE REMOVAL MESSAGE</td>
</tr>
<tr>
<td>351</td>
<td>C</td>
</tr>
<tr>
<td>352</td>
<td>WRITE (ITY,1202) TDRL,(MONTH,1DAY,1YEAR,1OT,TRFEL</td>
</tr>
<tr>
<td>353</td>
<td>1202 FORMAT (28M REMOVE OUTPUT TAPE IF DRIVE,1X,11/19H LABEL IT DATF,</td>
</tr>
<tr>
<td></td>
<td>14X,12,1H/,12,1H/,12,1H,1APF NUMBER ,1A,1A4 RFFL NUMAFL ,1P)</td>
</tr>
<tr>
<td>354</td>
<td>C</td>
</tr>
<tr>
<td>355</td>
<td>C CHECK IF STACKED OUTPUT REQUIRED</td>
</tr>
<tr>
<td>356</td>
<td>C</td>
</tr>
<tr>
<td>357</td>
<td>IF (IFLG2 .NE.,1) GO TO 999</td>
</tr>
<tr>
<td>358</td>
<td>GO TO 403</td>
</tr>
<tr>
<td>359</td>
<td>C</td>
</tr>
<tr>
<td>360</td>
<td>C STACK OUTPUT</td>
</tr>
<tr>
<td>361</td>
<td>C</td>
</tr>
<tr>
<td>362</td>
<td>C</td>
</tr>
</tbody>
</table>
402 CALL MGFRM(IDR1,NUMDNS,10DUN1),2,1,ISTAT)
404 REWIND IMG2
405 C
406 C CHECK MESSAGE
407 C
408 403 CONTINUE
409 IF (MINE .EQ. 1) GO TO 404
410 C
411 C SELECTION CODE MESSAGE -- SAME SC MESSAGE
412 C
413 WRITE (ITTY,1201)
414 1201 FORMAT (6IH DO YOU WISH TO USE THE SAME SELECTION CODE AS PREVIOUS)
415 IS RUN / 114 (Y OR N))
416 READ (ITTY,2000) A(1)
417 IF (A(1) .EQ. 1) GO TO 404
418 C
419 C CHECK FOR STACKED OUTPUT
420 C
421 IF (11FLG2 .NE. 1) GO TO 405
422 C
423 C CHECK FOR FAT -- STACK OUTPUT
424 C
425 IF (11STAT .NE. 1) GO TO 406
426 C
427 C OUTPUT TARP FINISHED -- CANNOT STACK - NEED NEW OUTPUT TARP
428 C
429 405 WRITE (ITTY,1000) 1001
430 1000 READ (ITTY,2000) A(1)
07/27/71 INPUT LISTING

CARO NO 0000 CONTENTS 0000

201 IF(ALL .NE. 0) GO TO 405
202 WRITE (ITYY,1001)
203 READ (ITYY,2001) I MONTH, IDAY, I YEAR
204 WRITE (ITYY,1001)
205 READ (ITYY,2002) INT
206 400 WRITE (ITYY,1001) INTZ
207 GO TO 200
208 404 CONTINUE
209 C
210 C COMPLETELY NEW INPUT FORMAT - CHECK FOR STACKED OUTPUT
211 C
212 IF (IFLGZ .EQ. 1) GO TO 1
213 GO TO 900
214 C
215 C TERMINATION PROCEDURE - TERMINATION MESSAGE
216 C
217 994 WRITE (ITYY,1204)
218 1204 FORMAT (42X END OF RUN - REMOVE ALL TAKES FROM P01/115/12777) UN
219 1E TERMINATION ****)
220 C
221 C
222 STOP
223 C
224 SUBROUTINE SCONE
225 C
226 C *** PART OF THE RMS SYSTEM THAT HANDLES THE SELECTION CODE PROCESSING
227 C
228 C *** OF INPUT DATA. THIS SUBROUTINE ACCEPTS SELECTION CODES, CHANNELS
229 C *** AND CHANNEL NUMBERS THAT WILL BE THE SELECTION CRITERIA FOR THE
230
C *** FINAL OUTPUT PRODUCT (NCF PROCESSING IN PHASE III 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(100, 2), AI(9)
COMMON /AI/ ISC, ITMAX, ITMAX, MINSC, MAXSC
COMMON /BI/ INSC, INCH, INDEX, INDEN
COMMON /CI/ TAPE, TAPCH, TAPDEN
COMMON /DI/ A, INDEX, ISC, ISCF, IDEN
COMMON /FI/ F1TA, F1TE, FY, FT, TP, TTS
COMMON /FI/ F1TY, IPT, IPT, IPTR
COMMON /GI/ IDEN2, IDEN2, IDEN2, IDEN2
COMMON /HI/ HSC, INCH, INCH, INCH, INDEN
COMMON /II/ MONTH, TDAY, IYEAR, IPAGE, IFILF, IFIV

C

WRITE (ITTY, IOMA)
1004 FORMAT (45H ENTER DEFAULT DENSITY AS NUMERIC FROM 0 TO 1)
C
C *** READ DENSITY
C
READ (ITTY, 2009) IDEN
2003 FORMAT (11)
C
C *** PRESET SC TABLE
C
INSC = 0
INCH = 0
INDEN = 0
07/22/71 INPUT LISTING

07/22/71 AUTOFLOW CHART SET - RIP5

CARD NO  **** CONTENTS  ****

447  ICH = 0
448  DO J = 1, I_MAX
449  C CALL INTW0L(J)
450  C
451  C *** CLEAR ALL ELSE FUNCTION
452  C
453  IREV = 0
454  IAFF = 0
455  IAECN = 9
456  TAEN4 = INDEN
457  C

458  9 WRITE (1TITY, 1009)
459  1009 FORMAT (51H ENTER CHANNEL NUMBER FROM 1 TO 4 OR ENTER 0 TO END) 12
460  IN SELECTION CODE INPUT EACH TIME  CH NR = 15 TYPED)
461  11 WRITE (1TITY, 1011)
462  1011 FORMAT (15H CH NR =)
463  IF (INCH .EQ. 0) GO TO 100
464  INCH = INCH - 1
465  IF (INCH .LE. 4) GO TO 9
466  C
467  C *** AUTOMATIC/MANUAL MESSAGE
468  C
469  INP = ITTY
470  WRITE (1TITY, 1012)
471  1012 FORMAT (12H ENTER CODES, 2X, 15H(NNWNWWWWWWWW)) 1
472  C
473  C *** PUT THE ENTRIES IN THE TABLE
C

MR = 0
MIN1 = -1
MIN2 = -2
MIN3 = -3
MIN4 = -4
10 CALL MKENT(MR)
    IF (MR .GE. 0) GO TO 10
    IF (MR .EQ. MIN1) GO TO 11
    IF (MR .EQ. MIN2) GO TO 10
    IF (MR .EQ. MIN3) GO TO 100
    IF (MR .EQ. MIN4) PAUSE 10
    GO TO 10
C
C *** INPUT COMPLETE - SORT THE TABLE
C
100 CALL SHLSRT
C
C *** PRINT THE TABLE
C
IDELT = 0
CALL MPAGE
LCNT = 0
ICNT = ITCUR - 1
DO 15 I = 1, ICNT
C
C *** OBTAIN PARAMETERS FROM SC TABLE
C
CALL OUTWD(1)
WRITE (IPRT1,1018) IDSC, INCH, INLEN, IIONEL
1018 FORMAT (6X,15,15X, 12,11X,11,12X,11)
LCNT = LCNT + 1
IF (LCNT .LT. 40) GO TO 15
LCNT = 0
IPAGE = IPAGE + 1
CALL NPAGE
15 CONTINUE
C
C *** CHECK ALL ELSE AND OUTPUT IF ON
C
IF (IAEF .NE. 1) GO TO 14
WRITE (IPRT1,1019) IAECH, IAEDEH
1019 FORMAT (5R, RHALL ELSE=13X,17,11X,11)
C
C *** SUMMARY MESSAGE
C
14 WRITE (ITTY,1020)
1020 FORMAT (6M SELECTION CODE SUMMARY ON PRINTER. ARE CORRECTIONS RFD
LUIRED / 14M ENTR Y OR N.)
C
C *** READ ANSWER
C
READ (ITTY,2000) A(1)
IF (A(1) .NE. TV) GO TO 14
C
C *** CORRECTIONS REQUIRED
531 C
532 IREV = IREV + 1
533 GO TO 9
534 16 CONTINUE
535 IF (A(I) .NE. TM) GO TO 14
536 C
537 C *** NU CORRECTIONS --- RETURN
538 C
539 RETURN
540 END
541 SUBROUTINE KEINT(IFLG)
542 C
543 C *** READS THE INPUT SELECTION CODE AND PUTF55 11 INTO THE
544 C *** SELECTION CODE TABLE
545 C
546 TINESION TSCI(PO20,21), A(91)
547 COMMON /A1/ ISC, IAEC, IAEC, MAXC, MAXC
548 COMMON /A2/ INCR, INCR, INCR, INCR
549 COMMON /C1/ IAEF, IAEH, IAEI
550 COMMON /D1/ IDEN1, IDC, IDC, IDEN
551 COMMON /E1/ IN, TA, TE, TX, TY, TO, TP, TT, TS, TR, TM
552 COMMON /F1/ TTY, IPT, IDT, INP, PRTR
553 COMMON /G1/ IDR1, IDR2, IDR3, IDRC
554 COMMON /H1/ ISC, IOCH, IOCH, IOCH
555 C
556 IFLG = 0
557 C
558 C *** READ THE ENTRY
07/22/71 INPUT LISTING

CARD NO CONTENTS

559 C
560 RFA, (1NP, 2004) A(1), A(2), IPEK, ISSC, I9CF
561 2004 FORMAT (A1, A1, I1, I2S)
562 C
563 C *** TEST FOR END
564 C
565 IF (A(1), NE, TK) 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, TK), NE)
573 1 (A(1), NE, TF)) GO TO 4
574 IF ((A(2), NE, TS), AND, (A(2), NE, T9)) GO TO 4
575 IF ((IDFN, LT, 7), OR, (IDFN, GT, 8)) GO TO 4
576 IF ((ISCS, LT, MINSC), OR, (ISCS, LT, MAXSC)) GO TO 4
577 IF ((A(2), EQ, TR), AND, (ISCY, LT, INSC), OR, (ISCY, GT, INSC))
578 1 GO TO 4
579 C
580 C *** ENTRY OK --- NOW DETERMINE T' PF
581 C
582 C
583 IF (A(1), NE, TA) GO TO 2
584 C
585 C *** ENTRY IS ALL ELSE
586 C
IAEF = 1
IAECH = INCH
IF (IDEN .NE. 0) IAEDEN = IDEN
RETURN
C
C *** NOT ALL ELSE --- MUST BE AN AND OR DELFTF
C
2 INDEL = 0
IF (A(1) .EQ. TE) INDEL = 1
C
C *** SFT UP FOR INTWOL - PRESET DEFAULT DENSITY AND CHANG IF NECESSARY
C
INDEN = IDDEN
IF IDDEN .GT. 0D INDEN = 1D5
C
C *** SFT UP DO LONP
C
JJ = ISCS
KK = ISCE
IF (A(2) .NE. TR) KL = ISCS
C
C *** DO LONP FURMS THE TABLE
C
DO 10 I = JJ, KK
10 INSC = I
CALL SCAN(I,K)
C
C *** CHECK FOR PREVIOUS ENTRY
INPUT LISTING

07/22/71

CARD NO CONTENTS

415 C

416 IF (K .GT. 0) GO TO 7

417 C

418 C *** NO PREVIOUS ENTRY

419 C

420 K = ITCUR

421 ITCUR = ITCUR + 1

422 C

423 C *** IF PREVIOUS ENTRY, OVFRWRITE, IF NOT ADD AT END OF TABLE

424 C

425 7 CALL INTWDL(K)

426 C

427 C *** CHECK IF TABLE FULL

428 C

429 IF (ITCUR .GT. ITMAX) GO TO 8

430 C

431 C *** TABLE NOT FULL

432 C

433 6 CONTINUE

434 IF (INP .EQ. ITTY) WRITE (ITY,1012)

435 1012 FORMAT (1H40)

436 IFLG = K

437 RETURN

438 C

439 C *** TABLE FULL—STOP ENTERING—WRITE LAST ENTRY MESSAGE

440 C

441 IF WRITE (ITY,1020) A(I),A(I),IDEN,ISCS,ISCE

442 1020 FORMAT (39H SELECTION CONT TABLE FULL—LAST ENTRY)

443 C

444 .
2/1X,A1,IX,A1,IX,11,IX,15,IX,15)

IFLG = -3
RETURN

C

C *** ERROR FOUND - PRINT ERROR MESSAGE

C

WRITE (1111,1000) A11, A17, IFN, SC, ISC, ISCE

1000 FORMAT (5AH ERRORIOUS SELECTION CODE ENTRY - WILL NOT BE PRINTED)

IVL=27H ENTRY AS RECEIVED IS /1X,A1,IV,A1,IX,11,IX,15,IX,15/1234A

? PAUSE 1, CONTINUE OR MANUALLY TERMINATE

C

PAUSE 1

C

IFLG = -2
RETURN
END

C

SUBROUTINE CNR

C

C *** GET THE NEXT RECORD FROM INPUT TAPF

C

DIMENSION ITI,T4U(T4U)


COMMON /CNOP/ T4P,T4P,T4P,T4P,NX,IX,E1T,T1T,S1T

COMMON /CNOP/ T4P,T4P,T4P,T4P,NX,IX,E1T,T1T,S1T

C

IF (IX<NC-7) GO TO 1

IEREAD = 0

NEWPOS = NEWPOS
07/27/71  INPUT LISTING

CARO NO  CONTENTS

471     ITAPE = IMG12
472     CALL PFOF(ITAPE)
473     3 CONTINUE
474     CALL NCFWR(IMG12, NWRDS, ITBUF(11), D, ISTAT)
475     IREAD = IREAD + 1
476     IF (ISTAT .EQ. 1 .OR. ITBUF(11) .EQ. 61440) GO TO 4
477     IF (ISTAT .GT. 1) GO TO 2
478
479     C
480     ITYPE = ISTAT
481     7 IX = 14
482     ICOND = ITBUF(10)
483     ICON1 = ITBUF(11)
484     IF (IFT .EQ. 7) WRITE (9,1010) ITBUF(10)
485     1010 FORMAT (23H FIRST Y COORDINATE IS ,16)
486     ICON2 = ITBUF(12)
487     NX = ITBUF(6)
488     NYS = NYS + 1
489     NXS = NXS + NX
490
491     C
492     TYPE IS Y
493
494     C
495     IF (ITYPE .EQ. 3) IYPE = 4
496     IF (ITYPE .EQ. 4) RFTUP
497
498     C
499     1 IF (NX .EQ. 0) GO TO 3
500     IF (IX .GT. 640) GO TO 4
501     9 ICOND = ITBUF(IX)
502     ICON1 = ITBUF(IX+1)
ICOM2 = ITRUF(IY+2)
IX = IX + 4
ETYPE = 3
C
VTYPF IS X
C
NX = NX - 1
RETURN
C
4 CONTINUE
CALL MGFRU(TMSC,T,NEWAG,ITRUF(1),S,ISTAT
IRFD = IRFAD 1
IF (ISTAT .GT. 4) WRITE (4,1020) IRFD
IF (ISTAT .EQ. 1) .NR. ITRUF(1),EQ. 11440) GO TO 4
IX = 6
GO TO 5
C
2 I TYPE = 4
ITEST = -17459
IF (ISTAT .GT. 4) I TYPE = 9
IF (ITRUF(7) .EQ. ITFST) I TYPE = 4
IF (I TYPE .EQ. 4) WRITE (A,1020) (READ
IF (ITRUF(7) .EQ. ITEST) GO TO 7
LO70 FORMAT (25H BAD READ AT IREAD = .15)
RETURN
C
C END OR EOT ENCOUNTERED
C
INPUT LISTING

CONTENTS

CARD NO 777

1 TYPE = ISTAT

724 IF ITAUFIL.EQ.414401 ITYPE = 2

729 WRITE (6,1030) IREAD

1030 FORMAT (1SH TOTAL READS = ,15).

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 /AI/ ISC,ITMAX,ITCUR,MINS,MAXS

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 "I

747 K = -1

748 1 CONTINUE

749 C

750 C *** NO DUPLICATE - I.F. 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),IN(1,7)
COMMON /AI/ ISC,ITMAX,ITCUR,MNINC,MNASC

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 IN = I + LN
IF (ISC(I,1) .LT. ISC(IN,1)) GO TO 5
C
C *** PERFORM EXCHANGE
C
NO 6 JJ = 1,2
F(I,JJ) = ISC(I,JJ)
ISC(I,JJ) = ISC(IN,JJ)
6 ISC(IN,JJ) = FR(I,JJ)
7 LN = I - LN
8 IF (I .GE. 1) GO TO 4
C
<table>
<thead>
<tr>
<th>CARD NO</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>783</td>
<td>C *** IN CORRECT ORDER</td>
</tr>
<tr>
<td>784</td>
<td>C</td>
</tr>
<tr>
<td>785</td>
<td>5 J = J + 1</td>
</tr>
<tr>
<td>786</td>
<td>C</td>
</tr>
<tr>
<td>787</td>
<td>C *** MORE TO SORT</td>
</tr>
<tr>
<td>788</td>
<td>C</td>
</tr>
<tr>
<td>789</td>
<td>IF (J - K) 3, 3, 1</td>
</tr>
<tr>
<td>790</td>
<td>C</td>
</tr>
<tr>
<td>791</td>
<td>C *** SORT FINISHED</td>
</tr>
<tr>
<td>792</td>
<td>C</td>
</tr>
<tr>
<td>793</td>
<td>2 CONTINUE</td>
</tr>
<tr>
<td>794</td>
<td>RETURN</td>
</tr>
<tr>
<td>795</td>
<td>END</td>
</tr>
<tr>
<td>796</td>
<td>SUBROUTINE SRCM(IARG, K)</td>
</tr>
<tr>
<td>797</td>
<td>C</td>
</tr>
<tr>
<td>798</td>
<td>C *** PERFORMS A BINARY SEARCH OF THE SELECTION CODE TABLE</td>
</tr>
<tr>
<td>799</td>
<td>C</td>
</tr>
<tr>
<td>800</td>
<td>DIMENSION ISG(1000, 2)</td>
</tr>
<tr>
<td>801</td>
<td>COMMON /AL/ ISG, ITMAX, ITCUR, MINSC, MAXSC</td>
</tr>
<tr>
<td>802</td>
<td>C</td>
</tr>
<tr>
<td>803</td>
<td>IFIRST = 1</td>
</tr>
<tr>
<td>804</td>
<td>ILAST = ITCUR + 1</td>
</tr>
<tr>
<td>805</td>
<td>1 IF (ILAST - IFIRST) .LT. 0) GO TO 4</td>
</tr>
<tr>
<td>806</td>
<td>LN = (ILAST - IFIRST)/2 + IFIRST</td>
</tr>
<tr>
<td>807</td>
<td>IF (IARG .EQ. ISGIN, LN) GO TO 7</td>
</tr>
<tr>
<td>808</td>
<td>IF (IARG .GT. ISGIN, LN) GO TO 3</td>
</tr>
<tr>
<td>809</td>
<td>C</td>
</tr>
<tr>
<td>810</td>
<td>C *** VALUE IN FIRST PART OF TABLE SECTION BEING SEARCHED</td>
</tr>
</tbody>
</table>
C
   ILAST = LN - 1
   GO TO 1
   3 IFIRST = 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 INTWDL(K)
   C
   C *** SETS UP THE SELECTION CODE TABLE ENTRIES
   C
   C *** FIRST WORD IS THE SELECTION CODE
   C
   C *** SECOND WORD BYTE 1 IS ZFST
   C
   C *** BYTE 2 IS DENSITY AND CHANNEL NUMBER
   C
   C *** BYTE 3 IS DELFTE FLAG
   C
   DIMENSION ISCG1000,21
07/27/71 INPUT LISTING

AUTOFLOW CHART SET - RNS

CARO NO

**** CONTENTS ****

R39 COMMON /A1/ ISc, ITMAX, ITCP, PINS, MAXSC
R40 COMMON /A1/ ISc, INCH, INPEN, INDEL
R41 C
R43 C *** CLEAR TABLE ENTRY
R44 C
R45 ISC(K,1) = 0
R46 ISC(K,2) = 0
R47 C
R48 C *** SET FIRST WORD
R49 C
R50 ISC(K,1) = ISc
R51 C
R52 C *** FORM SECOND BYTE OF SECOND WORD
R53 C
R54 ITemp = 0
R55 CALL LS(INDF, K, ITemp)
R56 CALL LN(ITEMP, INCH, ITEMp)
R57 CALL LFOR(ITEMP, 4, ITemp)
R58 CALL LS(ITEMP, 6, ITEMp)
R59 C
R60 C *** SET SECOND WORD
R61 C
R63 ISC(K,2) = ITemp
R64 RETURN
R65 END
R66 C
R67 C SUBROUTINE MYSQLO(K)
C *** DECODE THE SELECTION CODE TABLE ENTRIES
C
C

DIMENSION ISC (100, 2)
COMMON /ALL/ ESC, ETMAX, ETCUR, MINSC, MAXSC
COMMON /ALL/ TOSC, I0CH, I0NEN, I0NFL

C

C *** DATAIN SELECTION CODE
C
C
TOSC = ISC(K,1)

C *** STRIP SECOND WORD
C

ITEMP = ISC(K,2)
CALL LA (ITEMP, I, I0NFL)
C
ITEMP = ISC(ITEMP, A, I0NFL)
CALL LA (ITEMP, A, I0NEN)

I0CH = ITEMP - I0NEN
CALL RS (I0NEN, 4, I0NEN)
I0CH = I0CH - 4
RETURN

C

C *** SHIFTS BYTES FROM A GIVEN LOCATION IN ONE WORD TO A GIVEN
C *** LOCATION IN ANOTHER WORD
C
C *** IW = INPUT WORD
C
C *** IB = BYTE LOCATION OF INPUT WORD TO BE SHIFTED - I+7 OR 3
C
C *** IOW = OUTPUT WORD
07/22/71   INPUT LISTING

AUTOFLOW CHART SET - RPSS

CARD NO   ****   CONTENTS   ****

895   C ***  1DB = BYTE LOCATION TO BE SHIFTED INTO OUTPUT WORD 1, 2, 3
896   DIMENSION MASK1(3), MASK2(3), MASK3(6)
897   COMMON /MASK/ MASK1, MASK2
898   EQUIVALENCE (MASK1(1), MASK1(1))
899   C
900   CALL LA1(IW, MASK1(18), IWRK1)
901   CALL LA1(IW, MASK2(10B), IWRK2)
902   IX = 6*(IB - 1DB)
903   IF (IX) 1, 2, 3
904   1 IX = -IX
905   CALL RS(IWRK1, IX, IWRK1)
906   2 CALL LOI(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 *** WRITFS INPUT TAPE MESSAGES
914   C
915   COMMON /EI/ IN, TA, TE, TX, TY, TD, TP, TS, TQ, TN
916   COMMON /IF1/ ITTY, IPT, INT, INP, IPATA
917   COMMON /GI/ IDF1, IDF2, IDF3, IDDEC
918   COMMON /JI/ IMGT1, IMGT2, IDONE
919   C
920   C *** MORE TAPE MESSAGE
921   C
922   WRITE (ITTY, IDF1) IMGT1
018 FORMAT (54H ARE THERE MORE INPUT TAPES FOR THIS RUN IN ADDITION TO
AND THATS 27H CURRENTLY MOUNTED ON DRIVE 1x, 11, 15H ENTER Y OR N.
C
C *** READ ANSWER
C
READ (TTY,2015) ANS
2015 FORMAT (A1)
C
C *** SET DONE FLAG IF NO
C
IDONE = 0
IF (ANS .NE. TY) IDONE = 1
C
C *** CHECK IF DONE
C
IF (IDONE .EQ. 1) GO TO 1
C
C *** MOUNT NEXT TAPE MESSAGE
C
WRITE (TTY,1019) INGT2
1019 FORMAT (32H MOUNT NEXT INPUT TAPE ON DRIVE .11)
C
C *** PROPER TAPE MOUNTED MESSAGE
C
1 WRITE (TTY,1020) INGT1
1020 FORMAT (50H ENTER D IF PROPER INPUT TAPE IS MOUNTED ON DRIVE .11)
C
READ (TTY,2015) DNN
C
IF (DNN .NE. TD) GO TO 1
C *** PROPER TAPE MOUNTED - RESET DRIVE NUMBERS AND THEN RETURN
C
C IF (INIGT1 .EQ. IDR3) GO TO 2
INIGT1 = IDR3
RETURN
2 INIGT2 = IDR3
RETURN
END
SUBROUTINE SPACE
C C *** OUTPUTS THE HEADING INFORMATION FOR THE SELECTION CODE LIST
C COMMON (FL, IT, IPF, ID, IBM.-IPRTR
COMMON (FL, IF, IFMT, ID, IYEAR, IPAGE, FILE, FREV
C WRITE (IPRT-1016) imonth, iday, iyear, ipage
120,140,120,140 PAGE, 121
WRITE (IPRT-1014) IDY
WRITE (IPRT-1015) IFILE
103,103,103,103 FILE" 153
103,103,103,103 REPV
WRITE (IPRT-1016) REV
103,103,103,103 REPV
WRITE (IPRT-1017) REV
103,103,103,103 REPV
-222-
979 1017 FORMAT (93HO SELECTION CODE, CHANNEL, DENSITY, DFLET),
980 16HE FLAG)
981 RETURN
982 END
983 SUBROUTINE PRNSTY
984 C
985 C *** PRINT RUN STATISTICS
986 C
987 DIMENSION ICNT(25), ICH(4), ITBUF(740)
988 COMMON /KL/, ICNT, ICH
989 COMMON /TBUF/ ITBUF, NYS, NXS
990 COMMON /LIF/ EMIN, ISEG, IOFF
991 COMMON /F1/ LTYT, IPR, INT, IPRTR
992 COMMON /L1/ IMONTH, IDAY, IYAR, IPAGE, IFILE, IRFV
993 C
994 WRITE (IPRT, 1300) IMONTH, IDAY, IYEAR
995 1300 FORMAT (24HI RPS RUN STATISTICS OF ,12,1H/,12,1H/,12)
996 WRITE (IPRT, 1301) IDT
997 1301 FORMAT (14HO TAPE NUMBER ,15)
998 WRITE (IPRT, 1302) ITLF
999 1302 FORMAT (14H FILE NUMBER ,12)
1000 C
1001 C STATISTICS
1002 C
1003 WRITE (IPRT, 1303) (MIN, ISEG, NYS, NXS, (K, ICNT(K), K=1, 25)
1004 1303 FORMAT (/15H RUN STATISTICS/, /16H ELAPSED TIME = ,2F12.2, /16H MINUTES A
1005 1ND ,16,1X,7SECONDS, 16,1X,7MSECONDS, 16,1X,7MSCAN LINES INPUT CONTAINING, 16,1X,
1006 214H X COORDINATES, 16,1X,8DESIGNATOR OF POINTS PROCESSED WFP///2
07/22/71 INPUT LISTING

CONTENTS

1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034

39X,AMODE,12X,MCOUNT,25I,25X,14,12X,(4)
WRITE (IPRINT,1304) (ICHECK,K=1,4)
1304 FORMAT (I3MOCH,CHANNEL TOTALS: CHANNEL O = ),A,IX,13H CHANNEL 1 = ,I
16,IX,13H CHANNEL 2 = ,IA,IX,13H CHANNEL 3 = ,16)
RETURN
END
SUBROUTINE NSET
C
C *** AFSET output buffer pointers and clears the buffer to the high
C *** IGNORE VALUE
DINHPNION INSFT(16),INBUF(1364)
COMMON /M1/ INSFT,LSTWAD,LSTBYT,NEXT,INBUF
C
NEXT = 1
LSTBYT = 1
LSTWAD = 2
DO 1 J = 1,756,6
DO 2 K = 1,4
KK = J * K - 1
2 INBUF(KK) = INSET(K)
1 CONTINUE
RETURN
END
SUBROUTINE SETA
C
C ***
C
DIMENSION MASK(3),MASK2(3)
COMMON /GR1/ ITYPE, ICORD, ICON1, ICON2, NX, IX, IFT, ISTAT
COMMON /APLCS/ IMINF, IMMAR
COMMON /MASK/ MASK1, MASK2
COMMON /IOU/ IOUT1, IOUT2, IATST
EQUIVALENCE (MASK1, MASK2)

C
IMINF = 0
IF (IATST .GT. 3) GO TO 2
GO TO (1, 2, 3), IATST
2 CONTINUE
RETURN
1 CALL LENO(256, IOUT1, IOUT2)
RETURN
3 CALL LAM (MASK1, ICON2, IMNK)
CALL LEDR (IMNK, ICON2, ICON1)
ICORD = ICORD + IMMAR
IMINF = 1
GO TO 1
END
BLOCK DATA
COMMON /EI/ TN, TA, TE, TX, TY, TD, TP, TT, TS, TR, TM
DATA TN, TA, TE/SHH, SHA, SHE /
DATA TX, TY, TD, TP/SHX, SHY, SHD, SHP
DATA TT, TS, TR, TM/SHT, SHS, SHR, SHH
END
SUBROUTINE SETHM (I)
C
C THIS IS AN ASSEMBLY LANGUAGE ROUTINE
<table>
<thead>
<tr>
<th>Card No</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/22/71</td>
<td>INPUT LISTING</td>
</tr>
<tr>
<td>1043</td>
<td>C</td>
</tr>
<tr>
<td>1044</td>
<td>RETURN</td>
</tr>
<tr>
<td>1045</td>
<td>END</td>
</tr>
<tr>
<td>1046</td>
<td>SUBROUTINE SETC (I)</td>
</tr>
<tr>
<td>1047</td>
<td>G</td>
</tr>
<tr>
<td>1048</td>
<td>C</td>
</tr>
<tr>
<td>1049</td>
<td>THIS IS AN ASSEMBLY LANGUAGE ROUTINE</td>
</tr>
<tr>
<td>1050</td>
<td>C</td>
</tr>
<tr>
<td>1051</td>
<td>RETURN</td>
</tr>
<tr>
<td>1052</td>
<td>END</td>
</tr>
<tr>
<td>1053</td>
<td>SUBROUTINE LEOR (I,J,K)</td>
</tr>
<tr>
<td>1054</td>
<td>G</td>
</tr>
<tr>
<td>1055</td>
<td>C</td>
</tr>
<tr>
<td>1056</td>
<td>THIS IS AN ASSEMBLY LANGUAGE ROUTINE</td>
</tr>
<tr>
<td>1057</td>
<td>C</td>
</tr>
<tr>
<td>1058</td>
<td>RETURN</td>
</tr>
<tr>
<td>1059</td>
<td>END</td>
</tr>
<tr>
<td>1060</td>
<td>SUBROUTINE LEOR (I,J,K)</td>
</tr>
<tr>
<td>1061</td>
<td>C</td>
</tr>
<tr>
<td>1062</td>
<td>THIS IS AN ASSEMBLY LANGUAGE ROUTINE</td>
</tr>
<tr>
<td>1063</td>
<td>C</td>
</tr>
<tr>
<td>1064</td>
<td>RETURN</td>
</tr>
<tr>
<td>1065</td>
<td>END</td>
</tr>
<tr>
<td>1066</td>
<td>SUBROUTINE LEOR (I,J,K)</td>
</tr>
</tbody>
</table>
C THIS IS AN ASSEMBLY LANGUAGE ROUTINE
C
1092 RETURN
1093 END
1094 SUBROUTINE MGFW (I,J,K,L,M)
1095 C
1096 C THIS IS AN ASSEMBLY LANGUAGE ROUTINE
1097 C
1098 RETURN
1099 END
1100 SUBROUTINE RPOF (I)
1101 C
1102 C THIS IS AN ASSEMBLY LANGUAGE ROUTINE
1103 C
1104 RETURN
1105 END
1106 SUBROUTINE TMEII(I,J,K)
1107 C THIS IS A SYSTEM SUBROUTINE
1108 RETURN
1109 END
1110 SUBROUTINE LAI(J,K)
1111 C THIS IS AN ASSEMBLY LANGUAGE SUBROUTINE
1112 C RETURN
1113 END
<table>
<thead>
<tr>
<th>PG.BX</th>
<th>NAME</th>
<th>BPG.BX</th>
<th>NAME</th>
<th>AUTOFLOW CHART SET</th>
<th>RPSS</th>
<th>PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.07</td>
<td>CHPG</td>
<td>26.01</td>
<td>SRCH</td>
<td>90.15</td>
<td>2</td>
<td>6.07</td>
</tr>
<tr>
<td>34.01</td>
<td>IMFSCG</td>
<td>94.02</td>
<td>TMEF</td>
<td>32.07</td>
<td>2</td>
<td>6.06</td>
</tr>
<tr>
<td>28.01</td>
<td>ENMOL</td>
<td>32.05</td>
<td></td>
<td>34.20</td>
<td>2</td>
<td>6.05</td>
</tr>
<tr>
<td>59.02</td>
<td>LA</td>
<td>26.02</td>
<td></td>
<td>42.06</td>
<td>2</td>
<td>7.01</td>
</tr>
<tr>
<td>48.02</td>
<td>LFUR</td>
<td>26.09</td>
<td></td>
<td>7.14</td>
<td>1</td>
<td>3.21</td>
</tr>
<tr>
<td>48.02</td>
<td>LN</td>
<td></td>
<td>22.06</td>
<td>5.16</td>
<td>1</td>
<td>3.19</td>
</tr>
<tr>
<td>50.02</td>
<td>LS</td>
<td></td>
<td>19.14</td>
<td>24.12</td>
<td>3</td>
<td>6.31</td>
</tr>
<tr>
<td>52.02</td>
<td>MFRING</td>
<td></td>
<td>40.07</td>
<td>19.10</td>
<td>3</td>
<td>6.21</td>
</tr>
<tr>
<td>36.01</td>
<td>MPAGEF</td>
<td>24.09</td>
<td></td>
<td>42.08</td>
<td>1</td>
<td>6.24</td>
</tr>
<tr>
<td>40.01</td>
<td>MSET</td>
<td>34.12</td>
<td></td>
<td>24.11</td>
<td>1</td>
<td>7.07</td>
</tr>
<tr>
<td>50.02</td>
<td>MNEMT</td>
<td>16.14</td>
<td></td>
<td>32.09</td>
<td>3</td>
<td>8.17</td>
</tr>
<tr>
<td>32.01</td>
<td>MBYTE</td>
<td>13.19</td>
<td>10</td>
<td>4.08</td>
<td>10</td>
<td>5.14</td>
</tr>
<tr>
<td>59.01</td>
<td>MNTY</td>
<td>14.01</td>
<td>100</td>
<td>4.12</td>
<td>301</td>
<td>7.33</td>
</tr>
<tr>
<td>51.02</td>
<td>MCVF</td>
<td>13.10</td>
<td>11</td>
<td>4.18</td>
<td>302</td>
<td>7.04</td>
</tr>
<tr>
<td>31.02</td>
<td>RS</td>
<td>14.10</td>
<td>14</td>
<td>5.09</td>
<td>303</td>
<td>4.31</td>
</tr>
<tr>
<td>27.01</td>
<td>SCAN</td>
<td>14.13</td>
<td>15</td>
<td>5.05</td>
<td>304</td>
<td>8.16</td>
</tr>
<tr>
<td>13.01</td>
<td>SLIMV</td>
<td>14.00</td>
<td>16</td>
<td>4.12</td>
<td>305</td>
<td>24.17</td>
</tr>
<tr>
<td>42.01</td>
<td>SFTF</td>
<td>26.14</td>
<td>2</td>
<td>9.01</td>
<td>306</td>
<td>19.19</td>
</tr>
<tr>
<td>47.02</td>
<td>SFTC</td>
<td>13.06</td>
<td>2</td>
<td>6.20</td>
<td>307</td>
<td>14.19</td>
</tr>
<tr>
<td>46.01</td>
<td>SFTW</td>
<td>17.01</td>
<td>2</td>
<td>6.09</td>
<td>308</td>
<td>28.67</td>
</tr>
<tr>
<td>24.08</td>
<td>SHLSRT</td>
<td>40.05</td>
<td>2</td>
<td>6.01</td>
<td>309</td>
<td>9.01</td>
</tr>
<tr>
<td>Chart Title - Procedures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000074) 2.14 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000094) 7.22 500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000111) 3.08 700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000111) 3.10 700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000113) 3.11 700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000119) 3.13 50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000127) 3.19 21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000127) 3.21 200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000299) 4.01 390</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000143) 4.08 360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000144) 4.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000152) 4.12 301</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000169) 4.18 302</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000179) 5.01 306</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000145) 5.03 304</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000194) 5.09 303</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000205) 5.14 360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000232) 6.01 309</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(000233) 6.07</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

References (Source Sequence No. and Page/Box)
### Chart Title - Non-Procedural Statements

| 000594 | 17.01 7 | 000593 | 16.09 | 000595 | 17.02 |       |       |       |       |       |       |
| 000597 | 17.04 4 | 0006606 | 17.08 |       |       |       |       |       |       |       |       |
| 000502 | 17.07 7 | 0006600 | 17.05 |       |       |       |       |       |       |       |       |
| 000608 | 17.10 4 | 0006066 | 17.08 |       |       |       |       |       |       |       |       |
| 000611 | 17.11 7 | 000433 | 17.18 |       |       |       |       |       |       |       |       |
| 000624 | 17.16 7 | 000616 | 17.13 |       |       |       |       |       |       |       |       |
| 000633 | 17.13 6 |       |       |       |       |       |       |       |       |       |       |
| 000634 | 17.21 1 | 000634 | 17.19 |       |       |       |       |       |       |       |       |
| 000641 | 17.23 8 | 000629 | 17.17 |       |       |       |       |       |       |       |       |

### Chart Title - Subroutine GNR

<p>| 000727 | 19.01 6 | 000676 | 19.13 | 000712 | 19.25 |
| 000729 | 19.04 4 | 000728 | 19.02 |       |       |       |
| 000651 | 19.07 GRR | 000716 | 4.0 X |       |       |       |
| 000671 | 19.10 3 | 000694 | 19.14 |       |       |       |
| 000695 | 19.14 1 | 000648 | 19.07 |       |       |       |
| 000645 | 19.14 1 | 000643 | 20.11 |       |       |       |
| 000697 | 19.16 5 | 000714 | 19.26 |       |       |       |
| 000701 | 19.19 4 | 000696 | 19.15 |       |       |       |
| 000717 | 19.25 2 | 000711 | 19.22 |       |       |       |
| 000680 | 20.03 7 | 000721 | 20.21 |       |       |       |
| 000685 | 20.07 2 | 000693 | 20.04 |       |       |       |
| 000693 | 20.11 2 | 000692 | 20.04 |       |       |       |
| 000716 | 20.13 2 | 000677 | 20.01 |       |       |       |</p>
<table>
<thead>
<tr>
<th>CARD NO</th>
<th>PAGE/BOX</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>(000719)</td>
<td>20.16</td>
<td>(000718)</td>
</tr>
<tr>
<td>(000720)</td>
<td>20.18</td>
<td>(000719)</td>
</tr>
<tr>
<td>(000721)</td>
<td>20.21</td>
<td>(000720)</td>
</tr>
</tbody>
</table>

**CHART TITLE - NON-PROCEDURAL STATEMENTS**

**CHART TITLE - SUBROUTINE SCAN(IVAL,K)**

<table>
<thead>
<tr>
<th>CARD NO</th>
<th>PAGE/BOX</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>(000735)</td>
<td>22.01</td>
<td>SCAN</td>
</tr>
<tr>
<td>(000742)</td>
<td>22.03</td>
<td>(000748)</td>
</tr>
<tr>
<td>(000748)</td>
<td>22.06 1</td>
<td>(000742)</td>
</tr>
</tbody>
</table>

**CHART TITLE - NON-PROCEDURAL STATEMENTS**

**CHART TITLE - SUBROUTINE SHLSRT**

<table>
<thead>
<tr>
<th>CARD NO</th>
<th>PAGE/BOX</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>(000777)</td>
<td>24.01</td>
<td>(000779)</td>
</tr>
<tr>
<td>(000779)</td>
<td>24.02 6</td>
<td>(000785)</td>
</tr>
<tr>
<td>(000785)</td>
<td>24.06 9</td>
<td>(000772)</td>
</tr>
<tr>
<td>(000772)</td>
<td>24.14</td>
<td>(000791)</td>
</tr>
<tr>
<td>(000791)</td>
<td>14.01 7</td>
<td>(000789)</td>
</tr>
<tr>
<td>(000789)</td>
<td>24.07</td>
<td>(000789)</td>
</tr>
<tr>
<td>(000771)</td>
<td>24.13 4</td>
<td>(000781)</td>
</tr>
<tr>
<td>(000793)</td>
<td>24.16 7</td>
<td>(000764)</td>
</tr>
</tbody>
</table>

**CHART TITLE - NON-PROCEDURAL STATEMENTS**

**CHART TITLE - SUBROUTINE SINC(IARG,K)**
<table>
<thead>
<tr>
<th>Chart Title - Non-Procedural Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chart Title - Subroutine INTWOL()</td>
</tr>
<tr>
<td>(000832) 28.01 INTWOL (000449) 17.06-X (00062) 17.16-X</td>
</tr>
<tr>
<td>Chart Title - Non-Procedural Statements</td>
</tr>
<tr>
<td>Chart Title - Subroutine OUTWOL()</td>
</tr>
<tr>
<td>(000847) 30.01 OUTWOL (000933) 14.06-X</td>
</tr>
<tr>
<td>Chart Title - Non-Procedural Statements</td>
</tr>
<tr>
<td>Chart Title - Subroutine PAYTE()</td>
</tr>
<tr>
<td>(000893) 32.01 PAYTE (000547) 2.09-X (000237) 6.05-X (000221) 5.09-X (000259) 4.27-X (000297) 4.24-X</td>
</tr>
<tr>
<td>(000004) 32.05 1</td>
</tr>
<tr>
<td>(000004) 32.06 2</td>
</tr>
<tr>
<td>(000004) 32.04 3</td>
</tr>
<tr>
<td>(000004) 32.04 3</td>
</tr>
<tr>
<td>(000004) 32.04 3</td>
</tr>
<tr>
<td>Chart Title</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>Non-Procedural Statements</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Non-Procedural Statements</td>
</tr>
<tr>
<td>Non-Procedural Statements</td>
</tr>
<tr>
<td>Non-Procedural Statements</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
(001026)  40.05  2
(001027)  40.07  1

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - SUBROUTINE SETA
(001032)  42.01 SFPA  (000216)  6.13-X
(001044)  42.04  2  (001042)  42.02  (001043)  42.03
(001046)  42.06  1  (001049)  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 SFM(II)
(001067)  44.01 SFM  (000062)  2.07-X

CHART TITLE - SUBROUTINE SFIC(II)
(001068)  47.02 SFIC  (000063)  2.08-X

CHART TITLE - SUBROUTINE LO(I, J, K)
EQUVALENCE (MASK21, MASK211) (MASK22, MASK2) (MASK3, MASK21)

1000 FORMAT (28H MOUNT NEW SCRATCH ON DRIVE 11, 1MH ENTER D WHEN 5H DONE)

2000 FORMAT (A1)

1001 FORMAT (46H ENTER MONTH, DAY, YEAR AS 6 NUMERICS ... M999999)

2001 FORMAT (I2)

1002 FORMAT (29H ENTER 5 DIGIT TAPE ID NUMBER)

2002 FORMAT (I5)

1003 FORMAT (39H MOUNT 1ST REL OF INPUT TAPE ON DRIVE 11)

1101 FORMAT (49H ENTER 6 DIGIT MINIMUM AREA IN PROPER RF SOLUTION)

2100 FORMAT (16)

1100 FORMAT (150H DO YOU WISH TO SUPPLY SELECTION CODES EXTERNALLY / 13H ENTER Y OR N)

2222 FORMAT (23H BAD WRITE AT WRITE = .

1040 FORMAT (40H AT PAUSE 2, MOUNT NEW SCRATCH ON DRIVE 11, 1MH AND NO MINUE)

5000 FORMAT (16H TOTAL WRITES = .15, 4H + )

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 (29H REMOVE OUPT PUT TAPE ON DRIVE 11, 15H LABEL IT DATA, 4X, 12, 1H/12, 1H/12, 1H/12, 1H TAPE NUMBER 16, 14H REL NUMBER 12)

1203 FORMAT (161H DO YOU WISH TO USE THE SAME SELECTION CODES AS PREVIOUS
03.19-SC

*** PART OF THE RPS
** SYSTEM THAT HANDLES
** THE SELECTION CODE
** PROCESSING
** OF INPUT DATA.
** THIS SUBROUTINE
** ACCEPTS SELECTION
** CODES, DENSITIES
** AND CHANNEL
** NUMBERS THAT WILL BE
** THE SELECTION
** CRITERIA FOR THE
** FINAL OUTPUT
** PRODUCT OR/CE
** PROCESSING IN PHASE
** IT IS COMPLETED.
** THIS IS AN
** OPTION IN ADDITION TO
** THE MEANS OF
** PROCESSING THE DATA
** DIRECTLY FROM
** THE INPUT TAPE

- SCODE -

1.01

/ WRITE TO Devn
/ ITTY
/ VIA FORMAT
/ 1004

1.02

/ READ DENSITY
/ ITTY
/ VIA FORMAT
/ 2003
/ INTO THE LIST

1.07

NO

1.09

FND OF 07

** CLEAR ALL ELSE
** FUNCTION

1.11

1.13

1.19

1.21

TRUE

FALSF

10

19

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1
DIMENSION ISCL(N),2,A(I)
COMMON /AI/ ISCL,ITMAX,ITCUR,MINSC,MAXSC
COMMON /BI/ NINC,INCH,INDEN,INDEL
COMMON /CI/ IEF,FCH,IADEN
COMMON /DI/ A,INDEN,ISCC,ISCF,TODEN
COMMON /FI/ ITTV,ITP,INT,IMP,IPRTR
COMMON /GI/ ID,INR2,INR3,IDEK
COMMON /HI/ INSC,INCH,INDEN,INDEL
COMMON /II/ INC,ITV,ITR,TYX,TPS,TP2,TP3,TP4,TP5

1004 FORMAT (55H ENTER DEFAULT DENSITY AS NUMERIC FROM 0 TO 7)
2003 FORMAT (11)
1009 FORMAT (51H ENTER CHANNEL NUMBER FROM 1 TO 4 OR ENTER 0 THEN RUN/ #
N SELECTION CODE INPUT EACH TIME CH NR.* IS TYPED)
1011 FORMAT (58CH NR. =)
1012 FORMAT (12H ENTER VOLUME,2X,15H(MINIMUMNAME))
1018 FORMAT (6X,15X,12X,11X,12X,11X)
1019 FORMAT (5X,0HSMALL ELSE,13X,12X,11X)
1020 FORMAT (6H SELECTION CODE SUMMARY ON PRINTER. ARE CASE IMP'S KP)
UIRED / 14H ENTER Y OR N.)
DIMENSION ISC(1000,2), A(9)
COMMON /AI/ ISC, IMAX, IMC, MINC, MAXC
COMMON /BI/ ISC, INC, INDE, IDEN
COMMON /CI/ IAEF, IAEC, IAEDE
COMMON /DI/ A, IDEN, ISC, ISCE, ISRED
COMMON /E1/ TN, TA, TE, TX, TO, TP, TV, TS, TR, TN
COMMON /F1/ TTY, IPT, IDT, INP, IPTR
COMMON /G1/ IDR1, IDR2, IDM, IDFC
COMMON /H1/ ISC, IDCH, IDEN, IDFL

2004 FORMAT (A1, A1, I1, I19)
1012 FORMAT (3H 01)
1010 FORMAT (3H SELECTION CODE TABLE FULL - 1ST ENTRY, /I1X, A1, I1X, A1, I1X, I1, I1X, 19, I1X, 19)
1008 FORMAT (3H SELECTION CODE ENTRY - WILL NOT BE PUT IN TABLE/ 2ND ENTRY AS RECEIVED IS /I1X, A1, I1X, A1, I1X, I1X, I1X, 19, I1X, 19/4TH A
T PAUSE 1, CONTINUE OR MANUALLY TERMINATE)
DIMENSION TRUF(140)
COMMON /GMR1/ TYPE, ICORD, ICANI, ICANL, IX, IFI, ISTAT
COMMON /TRUF/ ITBUF, NYS, NYS
COMMON/J1/IMG1, IMG2, JNONE

1010 FORMAT (23H FIRST Y COORDINATE IS ,14)
1020 FORMAT (21H RAD READ AT IREAD = ,19)
1030 FORMAT (19H TOTAL READS = ,15)
CHART TITLE = SUBROUTINE SILSRT

/* SILSRT */

14.01 -- IC
*** SORT THE SELECTION CODE TABLE IN INCREASING ORDER BY FIRST WORD

01
  IN = IECIP - 1

26.07 -- IC
  LN = LN/2

26.07

02
  LN, EQ. IF TRUF

LN, EQ.
  TRUE

03
  NC

END OF ON
  LOOP

YES

04

/* CONTINUE SORT */

11
  J = 1

17
  EXIT

/* CONTINUE FINISH */

2
  NOT 1A

2
  CONTINUE

1
CHART TITLE - SUBROUTINE SRCH(ARG,K)

--- SRCH ---

07/22/71

AUTOFLOW CHART SET - PPSS

PAGE 76

--- SRCH ---

07/22---1X

*** PERFORMS A
BINARY SEARCH OF THE
SELECTION CODE 1..NLE

01

IF FIRST = 1

IF LAST = ITCUR - 1

---26.06---1X

1 02

(1.LAST - TRUE
(1.FIRST) > LT. 0

{FALSE

*** VALUE NOT IN
TABLE

---37---1X

4

T

K = 1

---44---1X

1 04

IF FIRST \\IHAST -
IF FIRST

---51---1X

1 04

FARG .EQ.
ISCILL(1)

*** VALUE FOUND IN
/ INTWDL /

13,066-->)<

*** SETS UP THE SELECTION CODE TABLE ENTRIES ***
*** FIRST WORD IS THE SELECTION CODE ***
*** SCOND WORD BYTE 1 IS ZFRD ***
*** ZYTF 2 IS ORPRTY AND CHANNEL NUMPF ***
*** ZYTF 3 IS DELETE FLAG ***

*** CLEAR TABLE ENTRY ***

<table>
<thead>
<tr>
<th></th>
<th>01</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISC(X,1)</td>
<td>0</td>
</tr>
<tr>
<td>ISC(X,2)</td>
<td>0</td>
</tr>
</tbody>
</table>

*** SET FIRST WORD ***

<table>
<thead>
<tr>
<th></th>
<th>02</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISC(X,1)</td>
<td>INSC T</td>
</tr>
</tbody>
</table>

*** FORM SCOND BYTE OF SECOND WORD ***

<table>
<thead>
<tr>
<th></th>
<th>03</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP</td>
<td>0</td>
</tr>
</tbody>
</table>
DIMENSION ISC(1000,7)
COMMON /A1/ ISC,ITMAX,ITCUR,MINSC,MAXSC
COMMON /B1/ INSC,INCH,INDEN,INDEL
PORTAL

14.06--1C

*** DECODE THE SELECTION CODE TABLE ENTRIES

*** OBTAIN SELECTION CODE

  01
  02
  03
  04

*** STRIP SECOND WORD

  01
  02
  03

05
  01
  02
  03

06
  01
  02
  03

07
  01
  02
  03
02.098-1C

*** SHIFTS BYTE FROM A GIVEN LOCATION IN ONE WORD TO A GIVEN LOCATION IN ANOTHER WORD ***

INPUT WORD

*** BYTE LOCATION OF INPUT WORD TO BE SHIFTED ***

1.WH

OUTPUT WORD

*** BYTE LOCATION TO BE SHIFTED ****

INPUT WORD-1,2,3
07/22/71

AUTOFLOW CHART SET - RPS5

PAGF 39

CHART TITLE - NON-PROCEDURAL STATEMENTS

COMMON /EI/ TN,TA,TF,TX,TY,\},TP,TT,TS,TP,TN
COMMON /FI/ ITTY,IPT,IOD,INP,'IPRT
COMMON /GI/ IDR1,IDR2,IDR3,IDEC
COMMON /JI/ INGT1,INGT2,IONE

1018 FORMAT (54H ARE THERE MORE INPUT TAPES FOR THIS RUN IN ADDITION TO,
6M THATS 2TH CURRENTLY MOUNTED ON DRIVE IX,II,ISH ENTER Y OR N.

2015 FORMAT (4I)

1019 FORMAT (32H MOUNT NEXT INPUT TAPE ON DRIVE IX)

1020 FORMAT (50H ENTER D IF PROPER INPUT TAP IS MOUNTED ON DRIVE IX)
AUTOFLOW CHART SET

CHART TITLE - SUBROUTINE NPAGE

---

14.09-1C

*** OUTPUTS THE MEANING INFORMATION FOR THE SELECTION CMD: LIST

I

01

WRITE TO DEV / IPRT
// VIA FORMAT / 1014 /
// FROM THE LIST /

I

02

NOTE

LIST = (MONTH, *
// IDAY, IYPA, *
// IPAGE *

I

03

WRITE TO DEV / IPRT
// VIA FORMAT / 1014 /
// FROM THE LIST /

I

04

NOTE

LIST = IOT *

I

05

WRITE TO DEV / IPRT
// VIA FORMAT / 1014 /
DIMENSION ICNT(25), ICH(4), ITBUF(140)
COMMON /K1/ ICNT, ICH
COMMON /ITBUF/ ITBUF, KYS, NXS
COMMON /L'/: ICHS, ISYS, IOFF
COMMON /F1/: ITTY, IOT, INP, IPFTR
COMMON /I1/: INOUTH, ITANG, IYEAR, IPAGE, IFILL, IPREV
1400 FORMAT (24HL, A8, RH RUN STATISTICS OF , 10, 14, 12, 14, 12)
1501 FORMAT (12AH, TARE NUMBER , 15)
1502 FORMAT (14H, FILE NUMBER , 12)
1503 FORMAT (12H, RH RUN STATISTICS//14H, ELAPSED TIME = , 1A, 1X, 12H, MINUTES A
NO , 1A, 1X, 7H, SECONDS//10H, 1X, 27H, SCAN LINES INPUT CONTAINING, 13, 1H,
14H, X, COORDINATES//13H, X, DESIGNER OF POINTS PROCESS ED WERE//
5H, 73, 13, 1H, 2X, 14, 12H, 16H)
1504 FORMAT (12H, CHANNEL TOTALS (CHANNEL 0 = , 1A, 1X, 13H, CHANNEL 1 = , 1
A, 1X, 13H, CHANNEL 2 = , 1A, 1X, 13H, CHANNEL 3 = , 14)
07/22/71

AUTOFLOW CHART SET - RPSS

PAGE 91

CHART TITLE - SUBROUTINE RS(i,j,k)

---

/ RS /
---

02.114-1C

THIS IS AN ASSEMBLY LANGUAGE ROUTINE

- I OP
---

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

---------------
/ MGFRI /
---------------
07.06--1C
THIS IS AN ASSEMBLY LANGUAGE ROUTINE

-02

- EXIT -
AUTOFLOW CHART SFT - PPS

CHART TITLE - SUBROUTINE REDF

/ REDF /
-------------
10.00---10

THIS IS AN ASSEMBLY LANGUAGE ROUTINE

* 1.02
* EXIT
*