National Aeronautics and Space Administration

80-10227

NASA CR-160659 JSC-13821 Revision A

Lyndon B. Johnson Space Center
Houston Texas 77058

MAY 13 1980

EARTH OBSERVATIONS DIVISION

SPACE AND LIFE SCIENCES DIRECTORATE

EARTH OBSERVATIONS DIVISION VERSION OF THE LABORATORY FOR APPLICATIONS OF REMOTE SENSING SYSTEM (EOD-LARSYS) USER GUIDE FOR THE IBM 370/148

VOLUME III — AS-BUILT DOCUMENTATION (SECTIONS 13 THROUGH 23)

Job Order 76-662

(E80-10227) EARTH OBSERVATIONS DIVISION VERSION OF THE LABORATORY FOR APPLICATIONS OF REMOTE SENSING SYSTEM (EOD-LARSYS) USER GUIDE FOR THE IBM 370/148. VOLUME 3: AS-BUILT (Lockheed Engineering and

N80-27776

Unclas
G3/43 00227

Prepared By

Lockheed Engineering and Management Services Company, Inc.
Houston, Texas

Contract NAS 9-15800

April 1980

LEMSCO-12565
Revision A

CONTENTS

TABLE

FIGURES

13. DATA-TR PROCESSOR SUBPROGRAMS

The DATA-TR processor accepts images from the MSS DATAPE and performs a linear transformation on user-defined fields using the statistical, histogram, or user-input method. Optionally, data may be rescaled. The transformed and/or rescaled data are output on the TRFORM file, logical unit 14, in either the Universal or LARSYS III format. The DATA-TR processor utilizes 8 programs that are exclusive to the processor and 33 utility subprograms. Figure 13-1 is a linkage diagram of the DATA-TR processor.

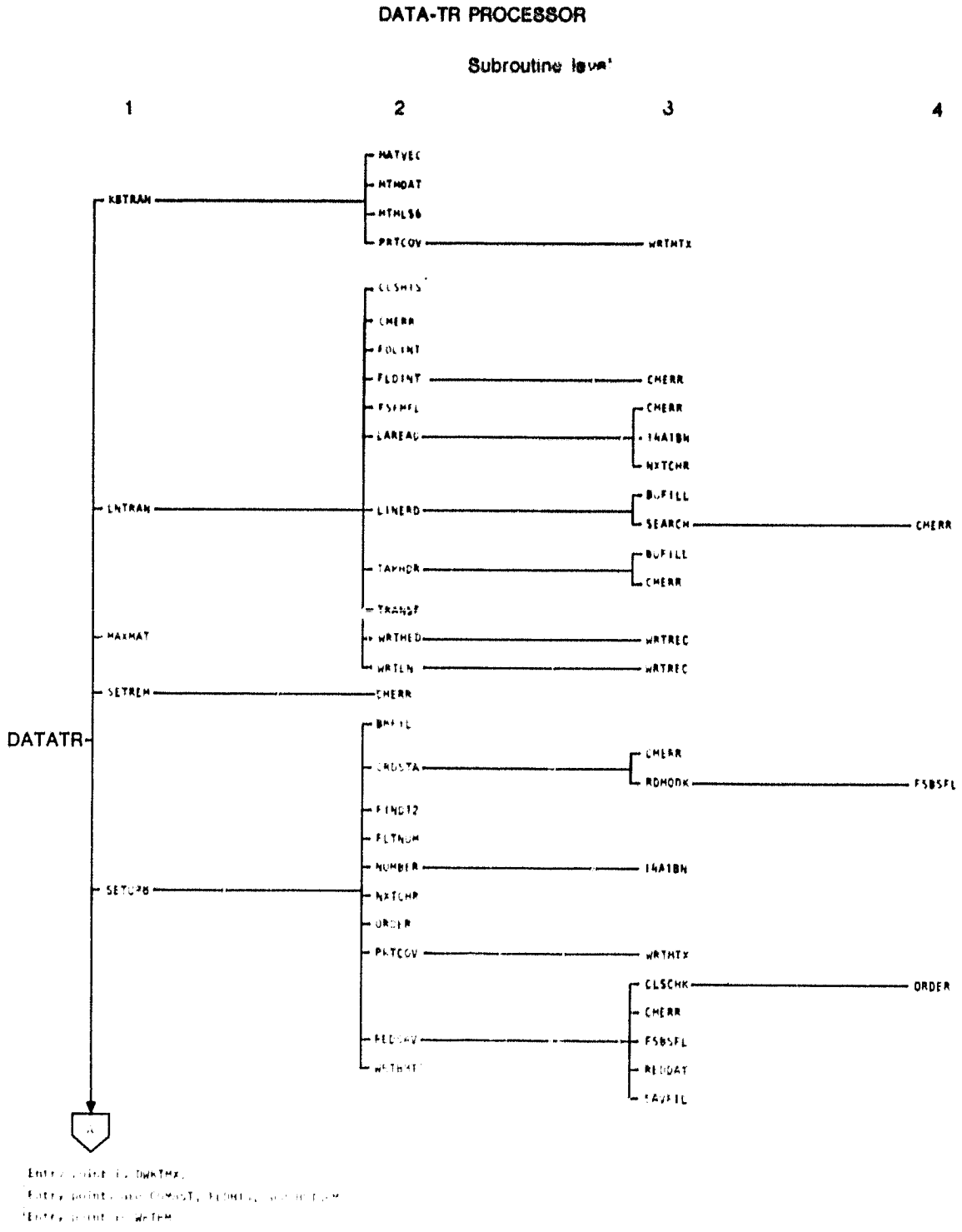


Figure 13-1.- Linkage diagram for the DATA-TR processor.

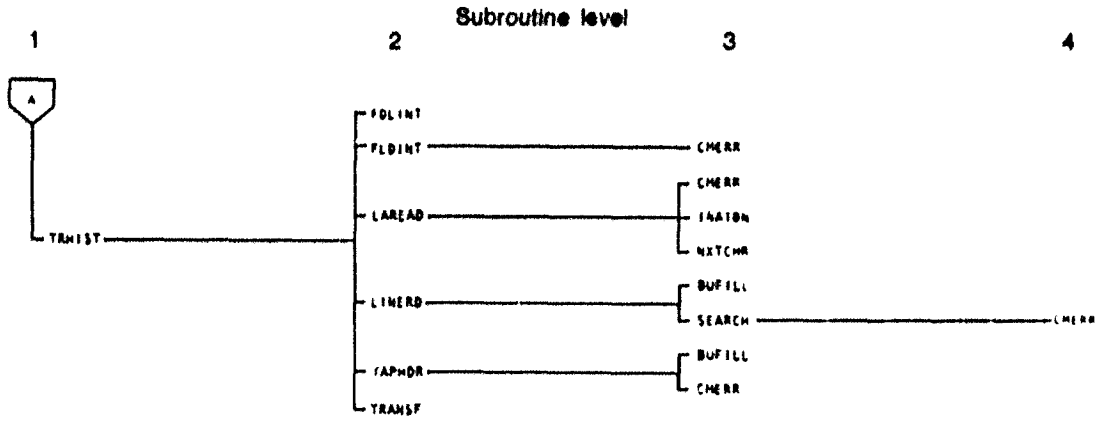


Figure 13-1.- Concluded.

13.1 DATATR

The DATATR subprogram is the driver routine for the DATA-TR processor.

13.1.1 LINKAGES

The DATATR subprogram calls the KBTRAN, LNTRAN, MAXMAT, SETREM, SETUP8, and TRHIST subprograms. It is called by MONTOR.

13.1.2 INTERFACES

The DATATR subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and TRBLCK and through the calling arguments.

13.1.3 INPUTS

Calling sequence: CALL DATATR(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In/out	Maximum usable storage in ARRAY; TOP = 10 600.

13.1.4 OUTPUTS

Not applicable.

13.1.5 STORAGE REQUIREMENTS

This subprogram requires 10 788 bytes of storage.

13.1.6 DESCRIPTION

The DATATR routine controls the data transformation processing. To determine which options are to be exercised during processing, subprogram SETUP8 is called to read the processor control cards. If scaling parameters are input by control cards, subprogram SETREM is called to initialize the arrays CON and MIN with the scaling parameters obtained from the input scaling parameter pairs. The processing continues with a call to LNTRAN, where the data are transformed. If rescaling is desired, the method is determined in SETUP8 using the SCAFLG parameter. If SCAFLG = 1, processing continues with a call to TRHIST for the histogram method; if SCAFLG = 2, KBTRAN is called for the statistical method; and, if SCAFLG = 3, LNTRAN is called for the user-input method.

If rescaling is not specifically requested by means of the RESCALE control card, no rescaling of transformed data values will be performed.

13.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 13.9.

13.1.8 LISTING

The subprogram listing is provided in volume IV, section 13.

13.2 KBTRAN

The KBTRAN subprogram rescales transformed data using the statistical method.

13.2.1 LINKAGES

The KBTRAN subprogram calls the MATVEC, MTMDAT, MTMLS6, and PRTCOV subprograms. It is called by the DATATR driver routine.

13.2.2 INTERFACES

The KBTRAN subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and TRBLCK and through the calling arguments.

13.2.3 INPUTS

Calling sequence: CALL KBTRAN(BMAT,LCOMB,ARRAY,LAM,MAX,MIN,EPS, TRANSF)

Parameter	Dimension	In/out	Definition
BMAT	480	In	Array containing the B-matrix.
LCOMB	1	In	Number of linear combinations or components in the field.
ARRAY	1	In	Working storage (see section 13.1.3).
LAM	1	In	A user-input integer, which is multiplied by the standard deviations of the input subclass statistics to derive an approximate range for rescaling the transformed data.
MAX	16	Out	Array containing maximum scaling parameter for each subclass.

Parameter	Dimension	In/out	Definition
MIN	16	Out	Array containing minimum scaling parameter for each subclass.
EPS	16	Out	Array containing scale factors for transformed data points calculated as $255/(MAX_i - MIN_i)$; i = 1,•••, LCOMB.
TRANSF	1	In	Determines if the transformed statistics will be printed.

13.2.4 OUTPUTS

This subprogram outputs the transformed covariance matrix via utility subprogram PRTCOV.

13.2.5 STORAGE REQUIREMENTS

This subprogram requires 11 648 bytes of storage.

13.2.6 DESCRIPTION

The KBTRAN subprogram computes the transformed means and covariance matrix for each subclass, prints the transformed covariance matrix if TRANSF $\neq$ 0, and calculates and stores the minimum and maximum values for each subclass.

13.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 13.9.

13.2.8 LISTING

The subprogram listing is provided in volume IV, section 13.

13.3 LNTRAN

The LNTRAN subprogram initiates the data transformation, rescales and histograms transformed data, and supervises the distribution of the transformed data.

13.3.1 LINKAGES

The LNTRAN subprogram calls the CLSHIS, CMERR, FDLINT, FLDINT, FSFMFL, LAREAD, LINERD, TAPHDR, TRANSF, WRTHED, and WRTLN subprograms. It is called by the DATATR driver routine.

13.3.2 INTERFACES

The LNTRAN subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and TRBLCK and through the calling arguments.

13.3.3 INPUTS

Calling sequence: CALL LNTRAN(IDATA,MAX,MIN,CON,BMAT,LCOMB, BMTRIG,SCAFLG,PEROUT,FILHIS,TOP,LAR,FLDNAM,NC,VERTCS,RESCAL, BIAS,NF,NPUN)

Parameter	Dimension	In/out	Definition
IDATA	TOP	In	Storage array for unpacked data.
MAX	16	In/out	Array containing maximum scaling parameter for each component of the transformation.
MIN	16	In/out	Array containing minimum scaling parameter for each component of the transformation.
CON	16	In/out	Array containing histogram scaling factor for each component of the transformation.
BMAT	480	In	Array containing the B-matrix.

Parameter	Dimension	In/out	Definition
LCOMB	1	In	Number of linear combinations or components in the field.
BMTRIG	1	-	Not used in this version of LNTRAN.
SCAFLG	1	In	Scaling flag indicating rescaling method. If = 1, histogram; = 2, statistical; = 3, user input.
PEROUT	1	In	An integer which specifies the percentage of points to be deleted from the upper and lower ends of the transformed data set in computing an approximate range for rescaling.
FILHIS	LCOMB,101	Out	Array containing histogrammed data.
TOP	1	In	Maximum usable storage in IDATA.
LAR	1	-	Not used in this version of LNTRAN.
FLDNAM	1	In	Name of field being processed.
NC	1	In	Number of points in the nonrectangular field.
VERTCS	2,''	In/out	Array containing field vertices.
RESCAL	1	In	Scaling flag. If = 0, no rescaling is applied.
BIAS	16	In	Array containing bias vector to be applied in transforming the data.
NF	1	In	Number of files to process. If = 1, output file TRFORM is rewound; if $\neq$ 1, the FSFMFL subprogram is called to position the routine to read another file.
NPUN	1	In	Punch card flag. If $\leq$ 0, no cards are input.

The control cards relevant to this routine are given in section 13 (table 13-1) of volume II of this user guide.

13.3.4 OUTPUTS

This subprogram outputs the transformed data set to the TRFORM file. This assignment must be made to tape if the transformed data set is to be saved by the user. The output transformed data set file will be in one of two formats, as specified on the FORMAT control card. These data are output also on the printer, along with a plot of the histogram (frequency distribution) of the transformed and rescaled data.

13.3.5 STORAGE REQUIREMENTS

This subprogram requires 44 662 bytes of storage.

13.3.6 DESCRIPTION

The functions provided by LNTRAN are to initiate the transformation of the data by a call to TRANSF to rescale the transformed data, to histogram the transformed data, to apply PEROUT to the distribution of the transformed data, and to output that data to the TRFORM file. Depending on the flag RESCAL, the transformed data may be either rescaled to the 0 to 255 range or output to the file unscaled as it is received from the TRANSF subprogram. If rescaling is not performed (RESCAL = 0), the transformed values are checked for being within the range 0 to 255. Any value outside the range is set to the range minimum (0) or maximum (255).

If the transformed data are to be rescaled (RESCAL > 0), rescaling is performed in LNTRAN using the following equation for each component i of the transformed data vector:

$$Y_i = CON_i (XT_i - MIN_i)$$

where

MIN_i = minimum value

XT_i = transformed data point

$CON_i = 255/(MAX_i - MIN_i)$

Y_i = rescaled transformed data point

If the OPTION PUNCH control card has been input, LNTRAN will output to the system punch file the card images containing the scaling parameters used to rescale the transformed data. The punched cards will be in control card format (OPTION SCAFAG=), and each card will contain two pairs of scaling parameters (CON,MIN). Each pair is associated with one component of the transformed data.

13.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 13.9.

13.3.8 LISTING

The subprogram listing is provided in volume IV, section 13.

13.4 MAXMAT

The MAXMAT subprogram computes an approximate transformed maximum and minimum for each component of the transformation.

13.4.1 LINKAGES

The MAXMAT subprogram does not call any other subprogram. It is called by the DATATR driver routine.

13.4.2 INTERFACES

The MAXMAT subprogram interfaces with other routines through common block TRBLCK and through the calling arguments.

13.4.3 INPUTS

Calling sequence: CALL MAXMAT(MAX,MIN,CON,BMAT,LCOMB,MAXPT)

Parameter	Dimension	In/out	Definition
MAX	16	Out	Array containing maximum scaling parameter for each component.
MIN	16	Out	Array containing minimum scaling parameter for each component.
CON	16	Out	Array containing histogram scaling factor for each component.
BMAT	480	In	Array containing the B-matrix.
LCOMB	1	In	Number of linear combinations in the field.
MAXPT	30	In	Array containing maximum data value for each channel.

13.4.4 OUTPUTS

The results are returned for use by the calling routine.

13.4.5 STORAGE REQUIREMENTS

This subprogram requires 710 bytes of storage.

13.4.6 DESCRIPTION

Using input or default maximum data values for each channel, MAXMAT computes the transformed value range (MAX and MIN) and the histogram scaling factor (CON).

13.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 13.9.

13.4.8 LISTING

The subprogram listing is provided in volume IV, section 13.

13.5 SETREM

The SETREM subprogram unpacks input scaling parameters, checks to assure one-to-one correspondence between the scaling factor and the additive scaling bias, and stores the values for later retrieval.

13.5.1 LINKAGES

The SETREM subprogram calls the CMERR subprogram. It is called by the DATATR driver routine.

13.5.2 INTERFACES

The SETREM subprogram interfaces with other routines through the calling arguments.

13.5.3 INPUTS

Calling sequence: CALL SETREM(CONMIN,CON,MIN,ADDNUM,LCOMB)

Parameter	Dimension	In/out	Definition
CONMIN	2,16	In	Array containing scale parameter pairs as read from OPTION SCAFAC control card in SETUP8.
CON	16	Out	Array containing multiplicative scaling factor for each linear combination.
MIN	16	Out	Array containing transformed data minimum for each linear combination.
ADDNUM	1	In	Total values input in CONMIN.
LCOMB	1	In	Number of linear combinations or components in the data transformation.

13.5.4 OUTPUTS

The results are returned for use by the calling routine.

13.5.5 STORAGE REQUIREMENTS

This subprogram requires 842 bytes of storage.

13.5.6 DESCRIPTION

SETREM receives the input scaling parameters from DATATR in array CONMIN. The input scale parameters consist of two values – the scaling factor CON and the additive scaling bias MIN. These values are unpacked from CONMIN and stored, respectively, in the CON and MIN arrays. SETREM checks to see that there is one-to-one correspondence between input scaling parameter pairs and the components of the transformation. If the test for input pair versus transformation component fails because of too many or too few input scaling parameter pairs, an error message is printed and DATATR is terminated via CMERR.

13.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 13.9.

13.5.8 LISTING

The subprogram listing is provided in volume IV, section 13.

13.6 SETUP8

The SETUP8 subprogram reads and analyzes all input processor control cards and sets default values for data transformation.

13.6.1 LINKAGES

The SETUP8 subprogram calls the BMFIL, CRDSTA, FIND12, FLTNUM, NUMBER, NXTCHR, ORDER, PRTCOV, REDSAV, and WRTBMT subprograms. It is called by the DATATR driver routine.

13.6.2 INTERFACES

The SETUP8 subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and TRBLCK and through the calling arguments.

13.6.3 INPUTS

Input to the SETUP8 subprogram consists of the BMFIL and SAVTAP files output by the SELECT and STAT processors, respectively.

Calling sequence: CALL SETUP8(BMAT,LCOMB,BMTRIG,PEROUT,MAXPT, ARRAY,LAM,SCAFLG,TOP,TRANSF,RESCAL,BIAS,ADDNUM,CONMIN,NPUN)

Parameter	Dimension	In/out	Definition
BMAT	480	In	Array containing B-matrix.
LCOMB	1	In	Number of linear combinations or components in the field.
BMTRIG	1	Out	If = 1, B-matrix file is input.
PEROUT	1	Out	An integer which specifies the percentage of points to be deleted from the upper and lower ends of the transformed data set in computing an approximate range for rescaling.

Parameter	Dimension	In/out	Definition
MAXPT	30	Out	Array containing maximum data value for each channel.
ARRAY	1	In	Working storage (see section 13.1.3).
LAM	1	Out	A user-input integer which is multiplied by the standard deviations of the input subclass statistics to derive an approximate range for rescaling the transformed data.
SCAFLG	1	Out	Scaling flag indicating rescaling method: = 1, histogram; = 2, statistical; = 3, user input.
TOP	1	In	Maximum usable storage in ARRAY.
TRANSF	1	Out	Determines if the transformed statistics will be printed.
RESCAL	1	Out	Scaling flag; if = 0, no rescaling is applied.
BIAS	16	Out	Array containing bias vector to be applied in transforming the data.
ADDNUM	1	Out	Total values input in CONMIN.
CONMIN	32	In	Array containing scaling parameter pairs as read from OPTION SCAFAC control card.
NPUN	1	Out	Punch card flag; if ≤ 0, no cards are input.

The control cards relevant to this routine are given in section 13 (table 13-1) of volume II of this user guide.

13.6.4 OUTPUTS

This subprogram outputs a line printer summary of the control card input. Input parameters or processing flags as a result of the OPTION PUNCH, OPTION SCAFAC, BIAS, or RESCALE control cards are returned to DATATR by subroutine argument.

13.6.5 STORAGE REQUIREMENTS

This subprogram requires 6446 bytes of storage.

13.6.6 DESCRIPTION

The SETUP8 subprogram begins by initializing flags and default values, the transformation bias vector BIAS, the maximum expected data value for each channel MAXPT, the distribution cutoff point PEROUT, and the standard deviation multiple LAM. It sets up the reread buffer and begins reading control card images. The B-matrix is read in from tape or card images; and the FEATURE, FORMAT, HED1 and HED2, COMENT, DATE, MAXPT, PEROUT, SUBCLASS, LAM, and OPTION cards are read using utility functions NXTCHR, NUMBER, and FIND12. If the rescaling option is exercised, the scaling parameter pairs CON and MIN are placed in CONMIN. If rescaling will be performed using the statistical method, statistics are read from the SAVTAP file, reduced to the specified channel set (FETVC2), and stored in ARRAY. Rescaling is not performed if the RESCALE control card is not input. If no BIAS control card is provided, the additive transformation bias vector is set = 0.

Subprogram PRTCOV is called to print the transformed covariance matrix and WRTBMT to print the input B-matrix.

13.6.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 13.9.

13.6.8 LISTING

The subprogram listing is provided in volume IV, section 13.

13.7 TRANSF

The TRANSF subprogram performs a linear data transformation.

13.7.1 LINKAGES

This routine does not call any other subprogram. It is called by LNTRAN and TRHIST.

13.7.2 INTERFACES

The TRANSF subprogram interfaces with other routines through common block TRBLCK and through the calling arguments.

13.7.3 INPUTS

Calling sequence: CALL TRANSF(XT,BMAT,IDATA,TOP,IL,K,LCOMB, NSAMP,BIAS)

Parameter	Dimension	In/out	Definition
XT	16	Out	Array containing transformed data vector: IDATA × BMAT + BIAS.
BMAT	480	In	Array containing B-matrix.
IDATA	TOP	In	Input data vector to be transformed.
TOP	1	In	Size of IDATA array; see section 13.1.3.
IL	1	In	Component of the transformed data vector.
K	1	In	Additive bias element.
LCOMB	1	In	Number of linear combinations or components in the field.
NSAMP	1	In	Number of samples.
BIAS	16	In	Additive bias vector.

13.7.4 OUTPUTS

The results are returned for use by the calling routine.

13.7.5 STORAGE REQUIREMENTS

The subprogram requires 672 bytes of storage.

13.7.6 DESCRIPTION

Subprogram TRANSF performs the following linear transformations: $\vec{Z} = A\vec{X}$; or, optionally, $\vec{Z} = A\vec{X} + \vec{b}$, where

$\vec{Z}$ = transformed data vector

A = transformation matrix; either the B-matrix or a user-supplied transformation matrix

$\vec{X}$ = input data vector

$\vec{b}$ = an additive bias vector

The B-matrix is a dimension reduction transformation generated by the SELECT processor. It may be input to the DATA-TR processor from a file created by the SELECT processor. If a user-supplied transformation matrix is used, it must be input in the same format as the B-matrix. The format of the input transformation matrix is described in section 3.1.4.2 of volume II of this user guide.

For the transformation, $\vec{Z} = A\vec{X} + \vec{b}$, the bias vector $\vec{b}$ is an option to the user. The option is exercised, and the bias vector is input via the BIAS control card. TRANSF performs the data transformation, $\vec{Z} = A\vec{X}$, in the absence of the BIAS control card.

13.7.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 13.9.

13.7.8 LISTING

The subprogram listing is provided in volume IV, section 13.

13.8 TRHIST

Using a histogram of the transformed data, the TRHIST subprogram computes scaling parameters for the transformed data.

13.8.1 LINKAGES

This routine calls the FDLINT, FLDINT, LAREAD, LINERD, TAPHDR, and TRANSF subprograms. It is called by the DATATR driver routine.

13.8.2 INTERFACES

The TRHIST subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and TRBLCK and through the calling arguments.

13.8.3 INPUTS

Calling sequence: CALL TRHIST(IDATA,AMAX,AMIN,ACON,BMAT,LCOMB, PEROUT,FILHIS,TOP,LAR,FLDNAM,NC,VERTCS,MAX,MIN,CON,BIAS)

Parameter	Dimension	In/out	Definition
IDATA	TOP	In	Array containing data to be transformed.
AMAX	16	In	Array containing maximum scaling parameter for each subclass.
AMIN	16	In	Array containing minimum scaling parameter for each subclass.
ACON	16	In	Array containing histogram scaling factor for each component of the transformation.
BMAT	480	In	Array containing B-matrix.
LCOMB	1	In	Number of linear combinations or components in the field.

Parameter	Dimension	In/out	Definition
PEROUT	1	In	An integer which specifies the percentage of points to be deleted from the upper and lower ends of the transformed data set in computing an approximate range for rescaling.
FILHIS	LCOMB,101	Out	Array containing histogrammed data.
TOP	1	In	Maximum usable storage in IDATA.
LAR	1	Out	Number of coordinates (vertices) in the field being processed.
FLDNAM	1	In	Name of field being processed.
NC	1	In	Number of points in the nonrectangular field.
VERTCS	2,11	In	Array containing field vertices.
MAX	16	Out	Array containing maximum scaling parameter for each subclass.
MIN	16	Out	Array containing minimum scaling parameter for each subclass.
CON	16	Out	Array containing histogram scaling factor for each component of the transformation.
BIAS	16	In	Array containing bias vector to be applied in transforming the data.

13.8.4 OUTPUTS

The results are returned for use by the calling routine.

13.8.5 STORAGE REQUIREMENTS

This subprogram requires 3298 bytes of storage.

13.8.6 DESCRIPTION

The TRHIST subprogram uses a histogram of the transformed data to derive the scaling parameters MAX, MIN, and CON. A histogram of a segment of the transformed image is performed to find the maximum value, MAX_i, and minimum value, MIN_i, for each component i of the transformed data. The scale factor, CON_i, is computed as $255/(MAX_i - MIN_i)$. In deriving an approximate range for the transformed data, the user may specify a percentage of points to be excluded from the upper and lower ends of the histogram by using the PEROUT control card. If not so specified, 2.5 percent of the points on the ends are excluded when determining the MAX_i and MIN_i values of the central 95 percent of the transformed data distribution.

Optionally, the user may specify the maximum expected data value for each channel n of the input data vector $\bar{x}$. Otherwise, the maximum data value for each channel is set equal to 255.

If the user-defined field is smaller than 2000 pixels, all pixels are used in the histogram. Otherwise, the following formula is used to determine the line increment and sample increment needed to obtain 2000 points for the histogram:

$$\alpha = \left(\frac{MN}{2000}\right)^{1/2}$$

where

M = number of samples

N = number of lines

α = increment (integer)

The input additive transformation bias vector is passed to TRHIST by subroutine argument BIAS and is used in TRANSF, which is called by TRHIST, to provide the transformation $\vec{AX} + \vec{b}$.

The function performed by TRHIST is invoked by the input RESCALE control card when neither of the other two options for rescaling (statistical and user-input) is specified.

13.8.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 13.9.

13.8.8 LISTING

The subprogram listing is provided in volume IV, section 13.

13.9 SUBPROGRAM FLOW CHARTS

No flow charts are available for the DATA-TR processor.

14. TRSTAT PROCESSOR SUBPROGRAMS

The TRSTAT processor reads the SAVTAP or a card image file produced by the STAT or ISOCLS processor and generates and files a set of transformed statistics (the A-matrix). The processor calls 6 subprograms that are exclusive to the processor and 17 utility subprograms. Figure 14-1 is a linkage diagram of the TRSTAT processor.

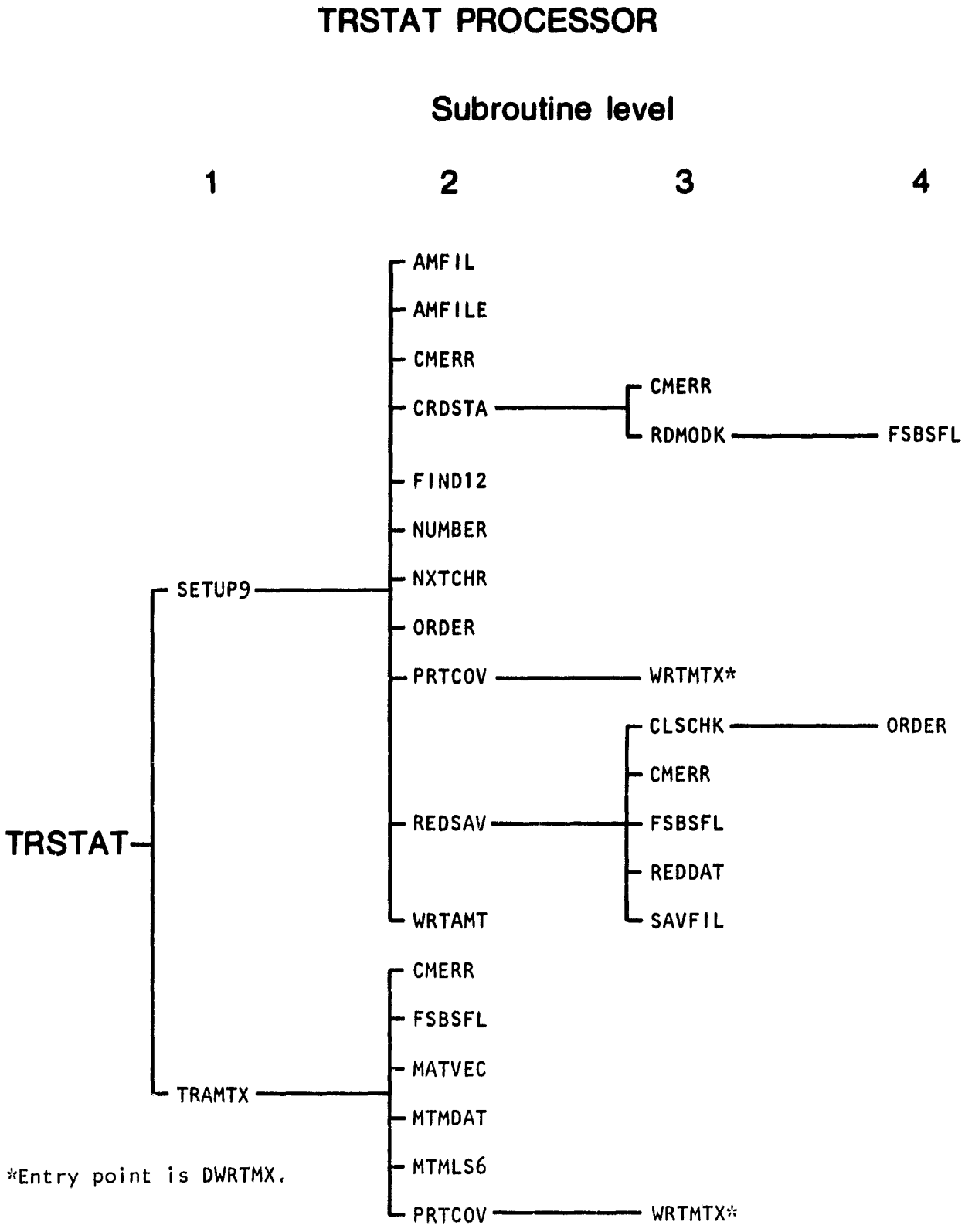


Figure 14-1.– Linkage diagram for the TRSTAT processor.

14.1 TRSTAT

The TRSTAT subprogram is the driver routine for the TRSTAT processor.

14.1.1 LINKAGES

The TRSTAT subprogram calls the SETUP9 and TRAMTX subprograms. It is called by MONTOR.

14.1.2 INTERFACES

The TRSTAT subprogram interfaces with other routines through common blocks GLOBAL and INFORM and through the calling arguments.

14.1.3 INPUTS

Calling sequence: CALL TRSTAT(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.

14.1.4 OUTPUTS

Not applicable.

14.1.5 STORAGE REQUIREMENTS

This subprogram requires 7760 bytes of storage.

14.1.6 DESCRIPTION

The TRSTAT subprogram calls SETUP9 to read and analyze control card images and the A-matrix and TRAMTX to read tape input, perform the statistical transformation, and print results.

14.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 14.7.

14.1.8 LISTING

The subprogram listing is provided in volume IV, section 14.

14.2 AMFIL

The AMFIL subprogram uses a formatted read to retrieve the A-matrix and B-vector card image files.

14.2.1 LINKAGES

The AMFIL subprogram does not call any other subprogram. It is called by SETUP9.

14.2.2 INTERFACES

The AMFIL subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

14.2.3 INPUTS

Calling sequence: CALL AMFIL(ROW,COLUMN,AMAT,VEC,B)

Parameter	Dimension	In/out	Definition
ROW	1	In	Number of the row being read.
COLUMN	1	In	Number of the column being read.
AMAT	1	In	Array containing a k-by-n matrix; $k \leq 15$ and n = number of linear combinations (≤ 30).
VEC	1	In	Array containing channel numbers to be used in processing.
B	30	In	Array containing a k-by-1 transformed mean vector.

14.2.4 OUTPUTS

The results are returned for use by the calling routine.

14.2.5 STORAGE REQUIREMENTS

This subprogram requires 664 bytes of storage.

14.2.6 DESCRIPTION

Not required.

14.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 14.7.

14.2.8 LISTING

The subprogram listing is provided in volume IV, section 14.

14.3 AMFILE

The AMFILE subprogram uses an unformatted read to retrieve the A-matrix and B-vector from tape or disk file.

14.3.1 LINKAGES

The AMFILE subprogram does not call any other subprogram. It is called by SETUP9.

14.3.2 INTERFACES

The AMFILE subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

14.3.3 INPUTS

Calling sequence: CALL AMFILE(ROW,NOCHAN,CHNVEC,AMAT,BVEC)

Parameter	Dimension	In/out	Definition
ROW	1	In	Number of rows to be processed.
NOCHAN	1	In	Number of channels to be processed.
CHNVEC	1	In	Array containing channels used in computation.
AMAT	1	In	Array containing a k-by-n matrix used in the linear transformation; $k \leq 15$ and n = number of linear combinations (≤ 30).
BVEC	1	In	Array containing a k-by-1 additive bias vector.

14.3.4 OUTPUTS

The results are returned for use by the calling routine.

14.3.5 STORAGE REQUIREMENTS

This subprogram requires 596 bytes of storage.

14.3.6 DESCRIPTION

Not required.

14.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 14.7.

14.3.8 LISTING

The subprogram listing is provided in volume IV, section 14.

14.4 SETUP9

The SETUP9 subprogram reads and analyzes control card images and the input SAVTAP file for the TRSTAT processor.

14.4.1 LINKAGES

This routine calls the AMFIL, AMFILE, CMERR, CRDSTA, FIND12, NUMBER, NXTCHR, ORDER, PRTCOV, REDSAV, and WRTAMT subprograms. It is called by the TRSTAT driver routine.

14.4.2 INTERFACES

The SETUP9 subprogram interfaces with other routines through common blocks GLOBAL and INFORM and through the calling arguments.

14.4.3 INPUTS

Input to the SETUP9 subprogram consists of the SAVTAP file output by the STAT processor.

Calling sequence: CALL SETUP9(ARRAY,TOP,AMAT,ROW,IP,TRAN,B)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In	Working storage (see section 14.1.3).
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
AMAT	1800	In	Array containing the k-by-n transformation matrix; k $\leq$ 15 and n = number of linear combinations ($\leq$30).
ROW	1	In	Number of rows in the A-matrix.
IP	1	Out	If = 1, training field definitions are output on cards; if $\neq$ 0, statistics will be output on cards (TRAMTX).

Parameter	Dimension	In/out	Definition
TRAN	1	Out	If = 1, the transformed covariance matrix will be printed (TRAMTX).
B	1	In	Array containing the additive bias vector.

The control cards relevant to this routine are given in section 14 (table 14-1) of volume II of this user guide.

14.4.4 OUTPUTS

This subprogram outputs card images and supervisory information and (optionally) the covariance matrix on the line printer.

14.4.5 STORAGE REQUIREMENTS

This subprogram requires 18 416 bytes of storage.

14.4.6 DESCRIPTION

The SETUP9 subprogram reads each input control card image, sets supervisory parameters, prints each card image, and generates a message if an error occurs. It reads the input SAVTAP file and compares the channel numbers to those input via the A-matrix. If the channels on the SAVTAP do not equal those on the A-matrix, an error message is generated and program execution terminates via subprogram CMERR. If ORIG = 1, the transformed covariance matrix is printed via subprogram PRTCOV.

14.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 14.7.

14.4.8 LISTING

The subprogram listing is provided in volume IV, section 14.

14.5 TRAMTX

The TRAMTX subprogram performs the statistical transformation and outputs the results on the SAVTAP file or (optionally) on cards.

14.5.1 LINKAGES

The TRAMTX subprogram calls the CMERR, FSBSFL, MATVEC, MTMDAT, MTMLS6, and PRTCOV subprograms. It is called by the TRSTAT driver routine.

14.5.2 INTERFACES

The TRAMTX subprogram interfaces with other routines through common blocks GLOBAL and INFORM and through the calling arguments.

14.5.3 INPUTS

Input to the TRAMTX subprogram consists of the SAVTAP file output by the STAT processor.

Calling sequence: CALL TRAMTX(ARRAY,TOP,AMAT,ROW,IP,TRAN,B)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In	Working storage (see section 14.1.3).
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
AMAT	ROW,NOFET2	In	Array containing the A-matrix.
ROW	1	In	Number of rows in AMAT.
IP	1	In	If = 1, training field definitions are output on cards; if ≠ 0, statistics are output on cards.
TRAN	1	In	If = 1, the transformed covariance matrix is printed.

Parameter	Dimension	In/out	Definition
B	30	In	Array containing the additive bias vector.

14.5.4 OUTPUTS

This subprogram outputs results on cards or file.

14.5.5 STORAGE REQUIREMENTS

This subprogram requires 18 416 bytes of storage.

14.5.6 DESCRIPTION

The TRAMTX subprogram writes statistics (if already transformed) to a file. If necessary, it performs a linear transformation by multiplying the A-matrix by the mean vector (using subprogram MATVEC), adding the bias vector to get transformed means, and then computing the transformed covariance matrix (A-matrix × covariance matrix × the transpose of the A-matrix). Subprograms MTMLS6 and MTMDAT are used to perform these operations, and the results are stored in ARRAY. The results are printed using subprogram PRTCOV and (optionally) are punched on cards.

14.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 14.7.

14.5.8 LISTING

The subprogram listing is provided in volume IV, section 14.

14.6 WRTAMT

The WRTAMT subprogram outputs the transformed statistics on the printer.

14.6.1 LINKAGES

The WRTAMT subprogram does not call any other subprogram. It is called by SETUP9.

14.6.2 INTERFACES

The WRTAMT subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

14.6.3 INPUTS

Calling sequence: CALL WRTAMT(AMAT,ROW,COLUMN,FETVC2,B)

Parameter	Dimension	In/out	Definition
AMAT	ROW,COLUMN	In	Array containing the A-matrix.
ROW	1	In	Number of rows in AMAT.
COLUMN	1	In	Number of columns in AMAT.
FETVC2	1	In	Array containing features (channels) processed.
B	1	In	Array containing the additive bias vector.

14.6.4 OUTPUTS

This subprogram outputs results on the printer.

14.6.5 STORAGE REQUIREMENTS

This subprogram requires 1132 bytes of storage.

14.6.6 DESCRIPTION

The WRTAMT subprogram prints the A-matrix, the numbers of linear combinations and channels, and the B-vector.

14.6.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 14.7.

14.6.8 LISTING

The subprogram listing is provided in volume IV, section 14.

14.7 SUBPROGRAM FLOW CHARTS

No flow charts are provided for the TRSTAT processor.

15. NDHIST PROCESSOR SUBPROGRAMS

The NDHIST processor computes an n-dimensional histogram of areas of the MSS DATAPE for which scatter plots have been requested by the user. The histogrammed pixels are output to the NHSTUN file, which is an interface to the SCTRPL processor (section 16). The NDHIST processor utilizes 13 processor subprograms and 16 utility subprograms. Figure 15-1 is a linkage diagram of the NDHIST processor.

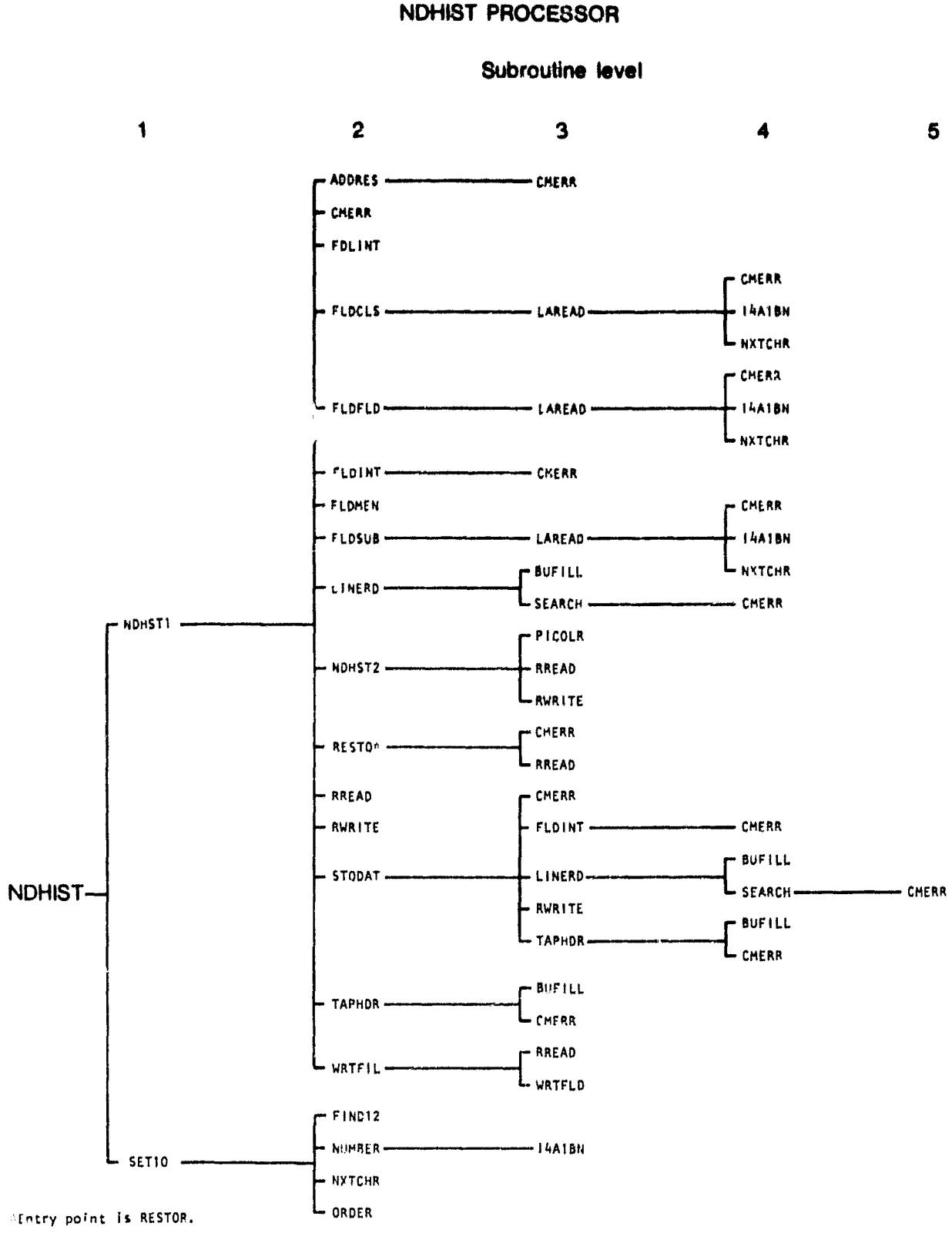


Figure 15-1.— Linkage diagram for the NDHIST processor.

15.1 NDHIST

The NDHIST subprogram is the driver routine for the NDHIST processor.

15.1.1 LINKAGES

The NDHIST subprogram calls the SET10 and NDHST1 subprograms. It is called by MONTOR.

15.1.2 INTERFACES

The NDHIST subprogram interfaces with other routines through the calling arguments.

15.1.3 INPUTS

Calling sequence: CALL NDHIST(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.

15.1.4 OUTPUTS

Not applicable.

15.1.5 STORAGE REQUIREMENTS

This subprogram requires 48 416 bytes of storage.

15.1.6 DESCRIPTION

As the driver routine for the NDHIST processor, NDHIST allocates storage for the array HIST. Dimensioned by the parameter LIMIT

(=12 000), the HIST array is used for storing the histogrammed vectors. NDHIST calls SET10 to read control card images and NDHST1 to coordinate the histogramming process.

15.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.1.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.2 ADDRES

The ADDRES subprogram divides the array ARRAY into three parts, allocates storage space for the various types of data to be processed, and computes disk addresses.

15.2.1 LINKAGES

The ADDRES subprogram calls the CMERR subprogram. It is called by the NDHST1 subprogram.

15.2.2 INTERFACES

The ADDRES subprogram interfaces with other routines through common block NDIM and through the calling arguments.

15.2.3 INPUTS

Calling sequence: CALL ADDRES(TOP,NSAMP,NOFEAT,BEGIN,BEGIN1)

Parameter	Dimension	In/out	Definition
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
NSAMP	1	In	Maximum number of samples per scan line in the user-defined test or training field.
NOFEAT	1	In	Number of channels to extract from the MSS DATAPE.
BEGIN	1	In	Beginning disk address for storing the frequency counters.
BEGIN1	1	In	Beginning disk address for storing the classified and/or clustered MAPUNT data.

15.2.4 OUTPUTS

This subprogram places the storage addresses for ARRAY and the disk addresses in the common block NDIM.

15.2.5 STORAGE REQUIREMENTS

This subprogram requires 772 bytes of storage.

15.2.6 DESCRIPTION

Subprogram ADDRES divides ARRAY into three partitions and reserves the 10 600 words for storage of the following data: the first 3600 words for training and/or test field information; the next 3000 words for field, subclass, or cluster numbers or color codes; and the last 4000 words for imagery data.

If the partitioned working storage is not large enough to contain all required information, the second partition is dumped onto high-speed disk. The disk also stores the frequency count and classified or clustered MAPUNT file. The disk addresses also are computed by ADDRES.

15.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.2.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.3 FLDCLS

The FLDCLS subprogram supervises the reading of the field definition data set when data are to be grouped on a class basis.

15.3.1 LINKAGES

The FLDCLS subprogram calls the LAREAD subprogram. It is called by NDHST1.

15.3.2 INTERFACES

The FLDCLS subprogram interfaces with other routines through common blocks INFORM and NDIM and through the calling arguments.

15.3.3 INPUTS

Calling sequence: CALL FLDCLS(FIELDS,STAMNT,*,*,*,IPT,VERTEX)

Parameter	Dimension	In/out	Definition
(1) FIELDS	4,1	Out	Array for storing field data: FIELDS(1,I) = field name for field I. FIELDS(2,I) = class number for field I. FIELDS(3,I) = subclass number for field I. FIELDS(4,I) = number of vertices + 1 for field I.
(2) STAMNT	1	In	Parameter for computed GO TO statement.
(3) *	1	In	Exit route if a field definition card is encountered.

Parameter	Dimension	In/out	Definition
(4) *	1	In	Exit route if a CLASSNAME card is encountered.
(5) *	1	In	Exit route if a $END card is encountered.
(6) IPT	1		Pointer for FIELDS array.
(7) VERTEX	1	Out	Array containing the field vertices.

The field definition card images input to this routine are described in section 3.2.3 of volume II of this user guide.

15.3.4 OUTPUTS

This subprogram stores class and subclass names in common block INFORM.

15.3.5 STORAGE REQUIREMENTS

This subprogram requires 1050 bytes of storage.

15.3.6 DESCRIPTION

The subprogram FLDCLS initiates the reading of the field definition data set and signals the calling routine NDHST1 when all fields for a given class have been processed. Calling arguments (parameters, section 15.3.3) are exit routes for the routine.

If a CLASSNAME card image is encountered, the program exits by calling argument (4); if a field definition card image is encountered, the program exits by calling argument (3); and if a $END card is encountered, the program exits by calling argument (5).

15.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.3.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.4 FLDFLD

The FLDFLD subprogram supervises the reading of the field definition data set when data are to be grouped on a field basis.

15.4.1 LINKAGES

The FLDFLD subprogram calls the LAREAD subprogram. It is called by NDHST1.

15.4.2 INTERFACES

The FLDFLD subprogram interfaces with other routines through common blocks INFORM and NDIM and through the calling arguments.

15.4.3 INPUTS

Calling sequence: CALL FLDFLD(FIELDS,STAMNT,*,*,IPT,VERTEX)

Parameter	Dimension	In/out	Definition
(1) FIELDS	4,1	Out	Array for storing the field data. (See section 15.3.3 for a more detailed description of this parameter.)
(2) STAMNT	1	In	Dummy variable.
(3) *	1	In	Exit route if a field definition card is encountered.
(4) *	1	In	Exit route if a $END card is encountered.
(5) IPT	1	Out	Pointer for FIELDS array.
(6) VERTEX	1	Out	Array containing field vertices.

The field definition card images input to this routine are described in section 3.2.3 of volume II of this user guide.

15.4.4 OUTPUTS

This subprogram stores class and subclass names (if applicable) in common block NDIM.

15.4.5 STORAGE REQUIREMENTS

This subprogram requires 794 bytes of storage.

15.4.6 DESCRIPTION

The FLDFLD subprogram initiates the reading of the field definition data set and signals the calling routine NDHST1 when a field definition card image has been read. Calling arguments (parameters, section 15.4.3) provide exit routes for the routine.

If a field definition card image is encountered, the program exits by calling argument (3); if a $END card is encountered, the program exits by calling argument (4).

15.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.4.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.5 FLDMEN

The FLDMEN subprogram computes the mean of each requested channel for each input test and/or training field. The mean is computed on a cumulative basis (one pixel at a time).

15.5.1 LINKAGES

The FLDMEN subprogram does not call any other subprogram. It is called by NDHST1.

15.5.2 INTERFACES

The FLDMEN subprogram interfaces with other routines chrough common blocks INFORM and NDIM and through the calling arguments.

15.5.3 INPUTS

Calling sequence: CALL FLDMEN(IDATA,J,NSAMP,NOFEAT,MEANS, BGCHAN,N)

Parameter	Dimension	In/out	Definition
IDATA	NSAMP,NOFEAT	In	Array containing the MSS DATAPE data.
J	1	In	Index for pixel of interest in IDATA array.
NSAMP	1	In	Number of pixels per scan line in IDATA.
NOFEAT	1	In	Total number of channels in IDATA.
MEANS	NCLRCH,1	Out	Array containing field means.
BGCHAN	1	In	Position of the first channel in IDATA to be used in computing the field means.

Parameter	Dimension	In/out	Definition
N	1	In	Total number of points in field I at the time of entry to FLDMEN.

15.5.4 OUTPUTS

The results are returned for use by the calling routine.

15.5.5 STORAGE REQUIREMENTS

This subprogram requires 812 bytes of storage.

15.5.6 DESCRIPTION

Subprogram FLDMEN computes the field mean in the following manner:

$$\text{MEANS} = \frac{N - 1 \times \text{OLD MEAN}}{N} + \frac{\text{DATA POINT}}{N}$$

where N = total number of points in the field at the time of entry to FLDMEN.

15.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.5.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.6 FLDSUB

The FLDSUB subprogram supervises the reading of the field definition data set when data are to be grouped on a subclass basis.

15.6.1 LINKAGES

This routine calls the LAREAD subprogram. It is called by NDHST1.

15.6.2 INTERFACES

The FLDSUB subprogram interfaces with other routines through common blocks INFORM and NDIM and through the calling arguments.

15.6.3 INPUTS

Calling sequence: CALL FLDSUB(FIELDS,STAMNT,*,*,*,IPT,VERTEX)

Parameter	Dimension	In/out	Definition
(1) FIELDS	4,1	Out	Array for storing field data. (See section 15.3.3 for a more detailed description of this parameter.)
(2) STAMNT	1	In	Parameter for computed GO TO statement.
(3) *	1	In	Exit route if a field definition card is encountered.
(4) *	1	In	Exit route if a CLASSNAME card is encountered.
(5) *	1	In	Exit route if a $END card is encountered.
(6) IPT	1	In	Pointer for FIELDS array.
(7) VERTEX	1	Out	Array containing field vertices.

The field definition card images relevant to this routine are described in section 3.2.3 of volume II of this user guide.

15.6.4 OUTPUTS

This subprogram stores class and subclass names in common block NDIM.

15.6.5 STORAGE REQUIREMENTS

This subprogram requires 1086 bytes of storage.

15.6.6 DESCRIPTION

The subprogram FLDSUB initiates the reading of the field definition data set and signals the calling routine NDHST1 when all the fields for a given subclass have been processed. Calling arguments (parameters, section 15.6.3) are exit routes for the routine.

If no CLASSNAME cards are input and a SUBCLASS card is encountered, the program exits by calling argument (4); if a field definition card is encountered, the program exits by calling argument (3); and, if a $END card is encountered, the program exits by calling argument (5).

15.6.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.6.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.7 NDHST1

The NDHST1 subprogram coordinates the histogramming process.

15.7.1 LINKAGES

The NDHST1 subprogram calls the ADDRES, CMERR, FDLINT, FLDCLS, FLDFLD, FLDINT, FLDMEN, FLDSUB, LINERD, NDHST2, RESTO, RREAD, RWRITE, STODAT, TAPHDR, and WRTFIL subprograms. It is called by the NDHIST driver routine.

15.7.2 INTERFACES

The NDHST1 subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and NDIM and through the calling arguments.

15.7.3 INPUTS

The MSS DATAPE and the MAPUNT file output by the ISOCLS, LABEL, or DISPLAY processor are input to the NDHST1 subprogram.

Calling sequence: CALL NDHST1(HIST,FIELDS,MEANS,VERTEX,LIMIT, ARRAY,TOP)

Parameter	Dimension	In/out	Definition
HIST	1	In	An array passed to NDHST2, which contains the histogrammed vectors.
FIELDS	4,1	In	Array for storing field data. (See section 15.3.3 for a more detailed description of this parameter.)
MEANS	1	In	An array passed to FLDMEN that contains the means of each input field.
VERTEX	1	In	Array containing field vertices.
LIMIT	1	In	A parameter set by NDHIST that dimensions HIST; LIMIT = 12 000.

Parameter	Dimension	In/out	Definition
ARRAY	1	In	Working storage (see section 15.1.3).
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.

15.7.4 OUTPUTS

This subprogram outputs results on the NDIM file.

15.7.5 STORAGE REQUIREMENTS

This subprogram requires 4206 bytes of storage.

15.7.6 DESCRIPTION

The subprogram NDHST1 sets up the logic and structure for histogramming up to 16 channels and for writing the NDIM file.

Depending on whether the data are to be grouped on a class, subclass, or field basis, subprogram FLDCLS, FLDSUB, or FLDFLD is called to read a field definition card. The rectangular coordinates surrounding the input test or training field returned by FLDINT are used to compute the number of lines and samples to be read from the MSS DATAPE and the MAPUNT file.

Subprogram ADDRES is called to compute the needed space for storing information in ARRAY and on the high-speed disk. If a MAPUNT file is being input, subprogram STODAT reads the tape and stores the information on the high-speed disk.

The MSS DATAPE and MAPUNT file, if applicable, are read into core one scan line at a time by FLDINT and RESTO, respectively. FLDINT computes the indexes for extracting the information of interest from each scan line of data from the MSS DATAPE. Subprograms NDHST2 and FLDMEN, if applicable, are called for each

pixel of interest. NDHST2 performs the histogramming, and FLDMEN computes the means of the field. If the pixel passed to NDHST2 is found to be a unique vector, NDHST2 sets the parameter VECSWT to 1. Each time VECSWT is set to 1, subprogram RESTO (entry RESTOR) is called, if applicable, to retrieve the pixel of interest from the MAPUNT (on the disk). The cluster or subclass number is stored in ARRAY. If ARRAY cannot store all of the required information, the identifiers are dumped onto the disk and the storage in ARRAY is reused.

When all field definition cards for a class, subclass, or field are processed, subprogram WRTFIL outputs the NDIM file. If the field definition cards are not grouped on a per-field basis, when a $END card is encountered, one additional NDIM file is output.

15.7.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.7.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.8 NDHST2

The NDHST2 subprogram computes a 1- to 16-channel histogram for either one or two sets of data.

15.8.1 LINKAGES

The NDHST2 subprogram calls the PICOLR, RREAD, and RWRITE subprograms. It is called by NDHST1.

15.8.2 INTERFACES

The NDHST2 subprogram interfaces with other routines through common blocks GLOBAL and NDIM and through the calling arguments.

15.8.3 INPUTS

Calling sequence: CALL NDHST2(J,IDATA,HIST,NOFET2,VECSWT,NSAMP, ARRAY,VECCNT,OVRFLO,NOFEAT,BGCHAN)

Parameter	Dimension	In/out	Definition
J	1	In	Index for IDATA array.
IDATA	1	In	Array containing information from the MSS DATAPE.
HIST	SIZE,MAXVEC	Out	Array containing the unique data vectors.
NOFET2	1	In	Number of plotting channels.
VECSWT	1	Out	Set = 1 if a unique vector was found; otherwise, set = 0.
NSAMP	1	In	Number of samples per scan line in IDATA array.
ARRAY	1	Out	Array containing the color codes.
VECCNT	1	Out	Index for ARRAY.

Parameter	Dimension	In/out	Definition
OVRFLO	1	Out	Table overflow switch.
NOFEAT	1	In	Total number of channels in IDATA array.
BGCHAN	1	In	Beginning channel in IDATA array for extracting color code channels.

15.8.4 OUTPUTS

This subprogram stores the number of unique vectors at any given time (NOVEC) in the common block NDIM.

15.8.5 STORAGE REQUIREMENTS

This subprogram requires 2094 bytes of storage.

15.8.6 DESCRIPTION

The subprogram NDHST2 computes a 1- to 16-channel histogram. The frequency is computed as a function of either the plotting channels or both plotting and color channels. The number of plotting channels is limited to 16; color channels are limited to 4.

The plotting channels are packed in one to four computer words. If applicable, another subprogram, PICOLR, packs the color channels into one computer word.

Each new plotting vector i is compared against a table of vectors. If a matching vector j is not found, the vector i is inserted into the table of existing vectors, and the frequency for vector i is set equal to 1. If a match is found and color channels are not being considered, the frequency for vector i is incremented by 1. If a match is found and color channels are being considered,

the frequency for vector i is incremented by 1. If a match is found and color channels are being considered, the color channels associated with the matching vector j are compared against the color channels associated with vector i. If the color channels for vectors i and j match, the frequency for the matching vector j is incremented by 1.

If the table containing the histogrammed vectors becomes full, the program will not insert a new entry but will continue to histogram the existing vectors.

15.8.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.8.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.9 PICOLR

The PICOLR subprogram extracts from the unpacked data the radiance values to be used as color codes on the PLOTAP. These values are packed into a computer color word (COLWRD) and are returned to the calling routine NDHST2.

15.9.1 LINKAGES

The PICOLR subprogram does not call any other subprogram. It is called by NDHST2.

15.9.2 INTERFACES

The PICOLR subprogram interfaces with other routines through common block NDIM and through the calling arguments.

15.9.3 INPUTS

Calling sequence: CALL PICOLR(IDATA,K,NOFEAT,COLWRD,NSAMP, NOFET2,BGCHAN)

Parameter	Dimension	In/out	Definition
IDATA	1	In	Array containing information from the MSS DATAPE.
K	1	In	Index for IDATA array.
NOFEAT	1	In	Total number of channels in IDATA array.
COLWRD	1	Out	Contains packed color codes.
NSAMP	1	In	Number of samples per scan line in IDATA array.
NOFET2	1	In	Number of plotting channels.
BGCHAN	1	In	Beginning color channel in IDATA array.

15.9.4 OUTPUTS

The results are returned for use by the calling routine.

15.9.5 STORAGE REQUIREMENTS

This subprogram requires 570 bytes of storage.

15.9.6 DESCRIPTION

Not required.

15.9.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.9.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.10 RESTO

The RESTO subprogram has two entry points, RESTO and RESTOR. RESTO reads a scan line of data from the MAPUNT file (previously stored on high-speed disk) into core; and RESTOR returns K, the pixel of interest, to the calling routine NDHST1.

15.10.1 LINKAGES

The RESTO subprogram calls the CMERR and RREAD subprograms. It is called by NDHST1.

15.10.2 INTERFACES

The RESTO subprogram interfaces with other routines through common blocks GLOBAL and IDWORD and through the calling arguments.

15.10.3 INPUTS

Input to the RESTO subprogram consists of the MAPUNT file output by the DISPLAY, ISOCLS, LABEL, or TESTSP processor.

Calling sequences:

a. CALL RESTO(ILINE,NSAMP,BEGIN1)

Parameter	Dimension	In/out	Definition
ILINE	1	In	Line number of the MAPUNT file to retrieve from the high-speed disk.
NSAMP	1	In	Number of samples per scan line on the MAPUNT file.
BEGIN1	1	In	Beginning disk address for storing the MAPUNT data.

b. ENTRY RESTOR(K,NUMB)

Parameter	Dimension	In/out	Definition
K	1	In	Index for IDWORD array.
NUMB	1	Out	Contains the cluster or subclass number of interest.

15.10.4 OUTPUTS

The results are returned for use by the calling routine.

15.10.5 STORAGE REQUIREMENTS

This subprogram requires 670 bytes of storage.

15.10.6 DESCRIPTION

Not required.

15.10.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.10.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.11 SET10

The SET10 subprogram reads and analyzes the control cards for the NDHIST processor and prints a summary report of the options requested by the user.

15.11.1 LINKAGES

The SET10 subprogram calls the FIND12, NUMBER, NXTCHR, and ORDER subprograms. It is called by the NDHIST driver routine.

15.11.2 INTERFACES

The SET10 subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and NDIM and through the calling arguments.

15.11.3 INPUTS

Calling sequence: CALL SET10(LIMIT)

Parameter	Dimension	In/out	Definition
LIMIT	1	In	Parameter set in NDHIST; LIMIT = 12 000.

The control cards relevant to this routine are given in section 15 (table 15-1) of volume II of this user guide.

15.11.4 OUTPUTS

This subprogram outputs a summary on the line printer. Depending on user requests, various switches and parameter values are initialized and stored in common blocks NDIM, INFORM, and GLOBAL.

15.11.5 STORAGE REQUIREMENTS

This subprogram requires 4612 bytes of storage.

15.11.6 DESCRIPTION

SET10 initializes parameters and generates an input summary. It then sets up the reread buffer, reads control cards, and generates a message in the event an error occurs. A list of user-requested options is printed.

15.11.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.11.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.12 STODAT

The STODAT subprogram reads and stores the MAPUNT file on high-speed disk.

15.12.1 LINKAGES

The STODAT subprogram calls the CMERR, FLDINT, LINERD, RWRITE, and TAPHDR subprograms. It is called by NDHST1.

15.12.2 INTERFACES

The STODAT subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

15.12.3 INPUTS

Input to the STODAT subprogram consists of the MAPUNT file output by the DISPLAY, ISOCLS, LABEL, or TESTSP processor.

Calling sequence: CALL STODAT(ILINE,NSAMP,HIST,LIMIT,BEGIN1)

Parameter	Dimension	In/out	Definition
ILINE	1	In	Number of lines on MAPUNT file.
NSAMP	1	In	Number of samples per scan line on MAPUNT file.
HIST	LIMIT	In	Temporary storage for MAPUNT data.
LIMIT	1	In	Parameter dimensioning for HIST processor; LIMIT = 12 000.
BEGIN1	1	In	Beginning disk address for storing MAPUNT data.

15.12.4 OUTPUTS

This subprogram stores, in the common block GLOBAL, the number of the next file to be processed from the MAPUNT tape.

15.12.5 STORAGE REQUIREMENTS

This subprogram requires 1234 bytes of storage.

15.12.6 DESCRIPTION

The coordinates of the user-input test and/or training fields are used to compute the numbers of lines and samples to read from the MAPUNT tape. The area clustered or classified must be the same as the area being histogrammed. Only one file from the MAPUNT tape is in core at a given time.

15.12.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 15.14.

15.12.8 LISTING

The subprogram listing is provided in volume IV, section 15.

15.13 WRTFIL

The WRTFIL subprogram writes the NDIM file on unit NHSTUN. The NDIM file is an interface to the SCTRPL processor.

15.13.1 LINKAGES

This routine calls the RREAD and WRTFLD subprograms. It is called by NDHST1.

15.13.2 INTERFACES

The WRTFIL subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and NDIM and through the calling arguments.

15.13.3 INPUTS

Calling sequence: CALL WRTFIL(HIST,MEANS,ID,COLOR,FIELDS, VERTEX,I)

Parameter	Dimension	In/out	Definition
HIST	1	In	Array containing the histogrammed vectors.
MEANS	1	In	Array containing the means of the test and/or training fields.
ID	1	In	Array containing the field, cluster, and/or subclass numbers for the corresponding vectors in HIST.
COLOR	1	In	Array containing the color codes for the corresponding vectors in HIST.
FIELDS	4,1	In	Array containing the field information.

Parameter	Dimension	In/out	Definition
VERTEX	2,1	In	Array containing the field vertices.
I	1	In/out	Number of records to be processed.

15.13.4 OUTPUTS

This subprogram outputs the NDIM file on tape or disk. The content and format of the NDIM file are given in volume II, section 15.

15.13.5 STORAGE REQUIREMENTS

This subprogram requires 2732 bytes of storage.

15.13.6 DESCRIPTION

Subprogram WRTFIL writes histogram information from NHSTUN onto disk or tape and outputs the following information on the line printer: identification information for each data file, such as class, subclass, or field name; field vertices; number of unique data vectors found; and mean statistics for each input field.

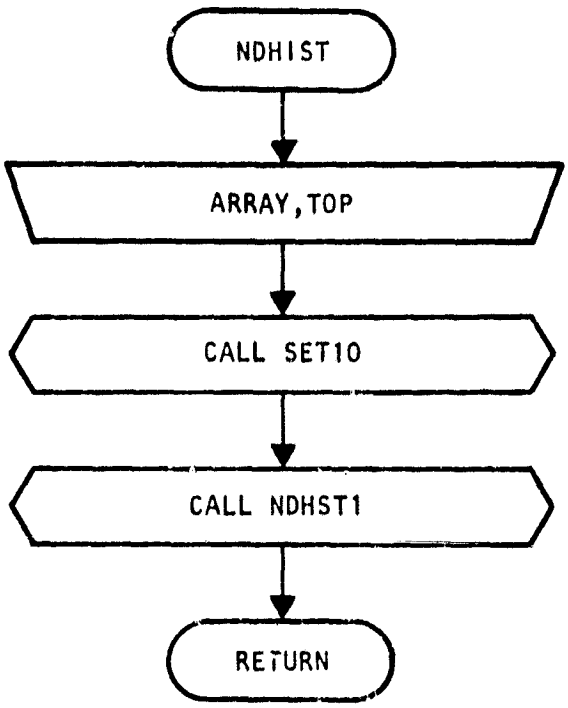
15.13.7 FLOW CHART

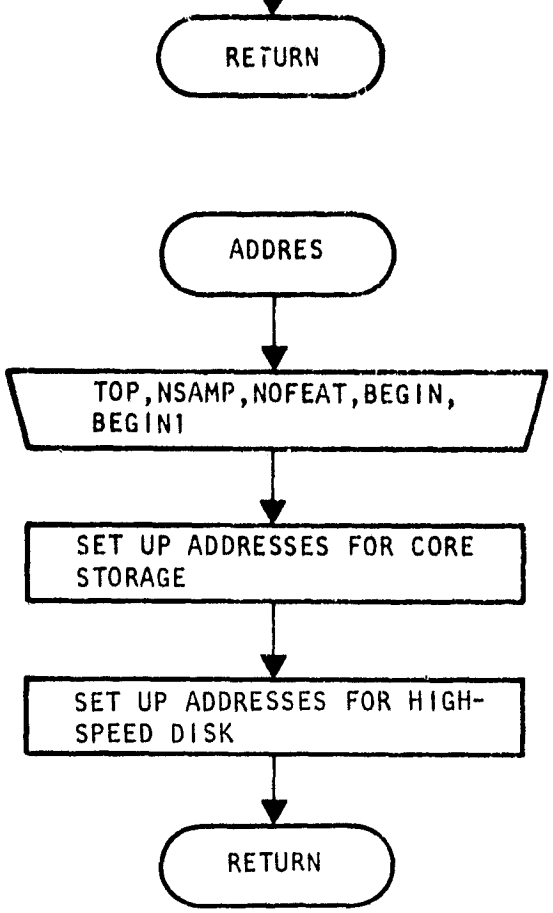
The available subprogram flow charts for this processor are provided in section 15.14.

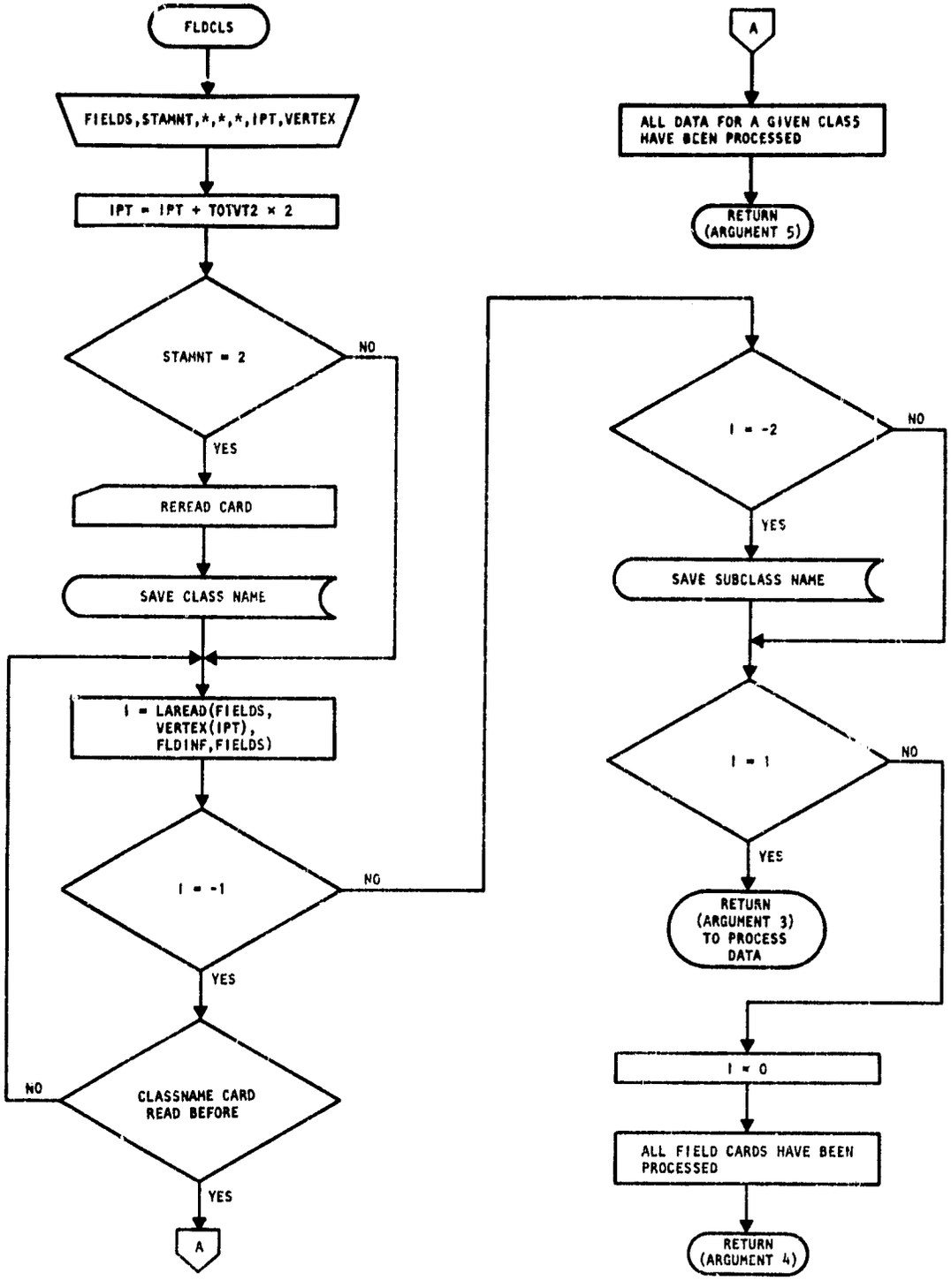
15.13.8 LISTING

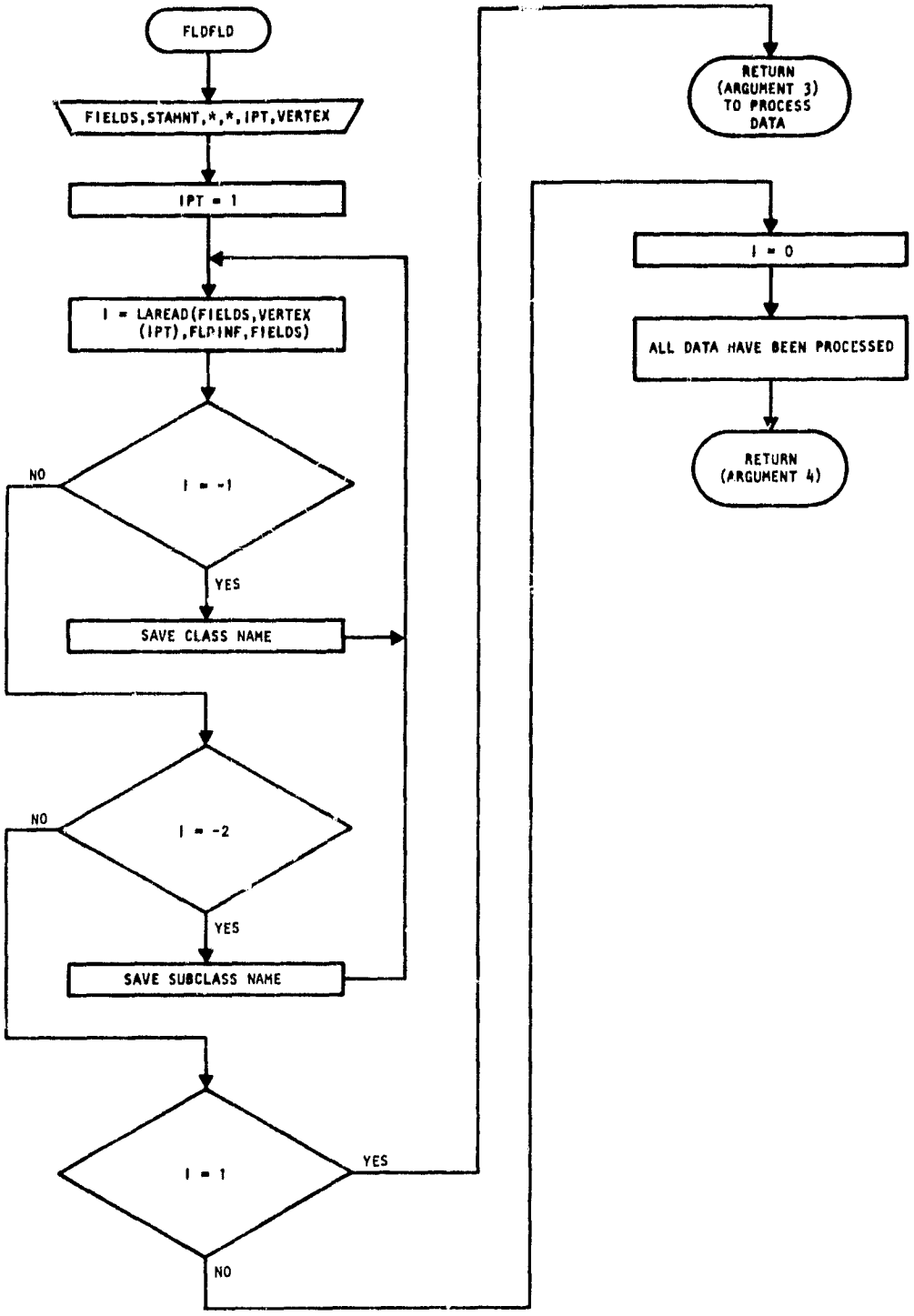
The subprogram listing is provided in volume IV, section 15.

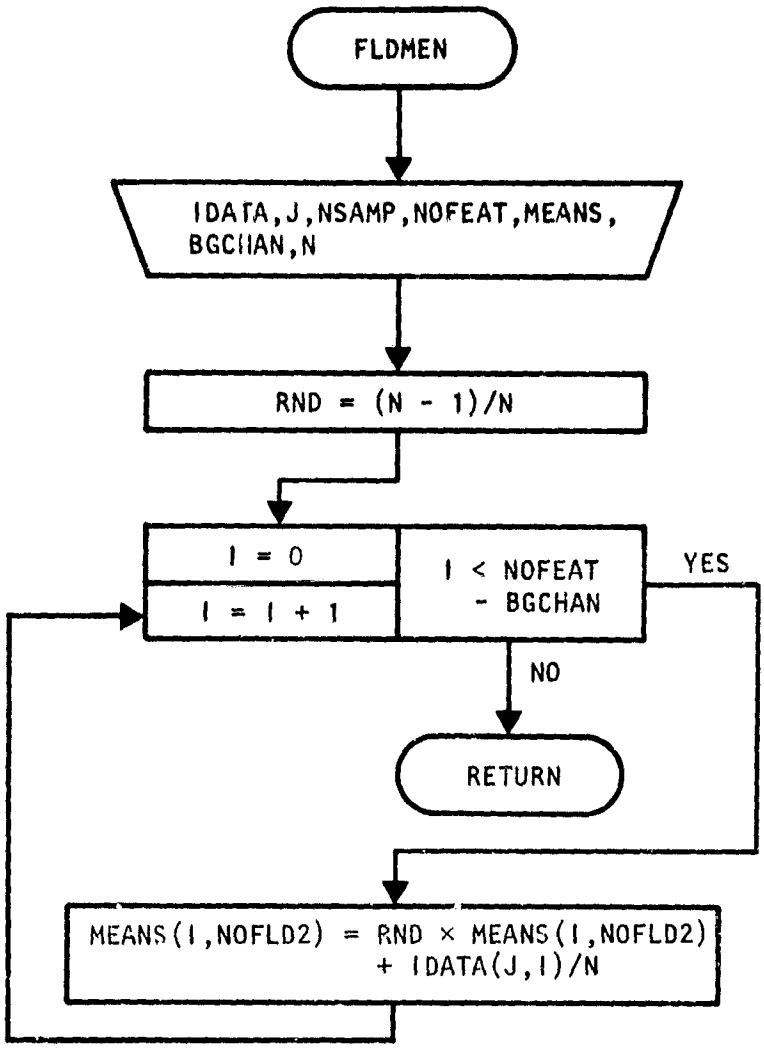
15.14 SUBPROGRAM FLOW CHARTS

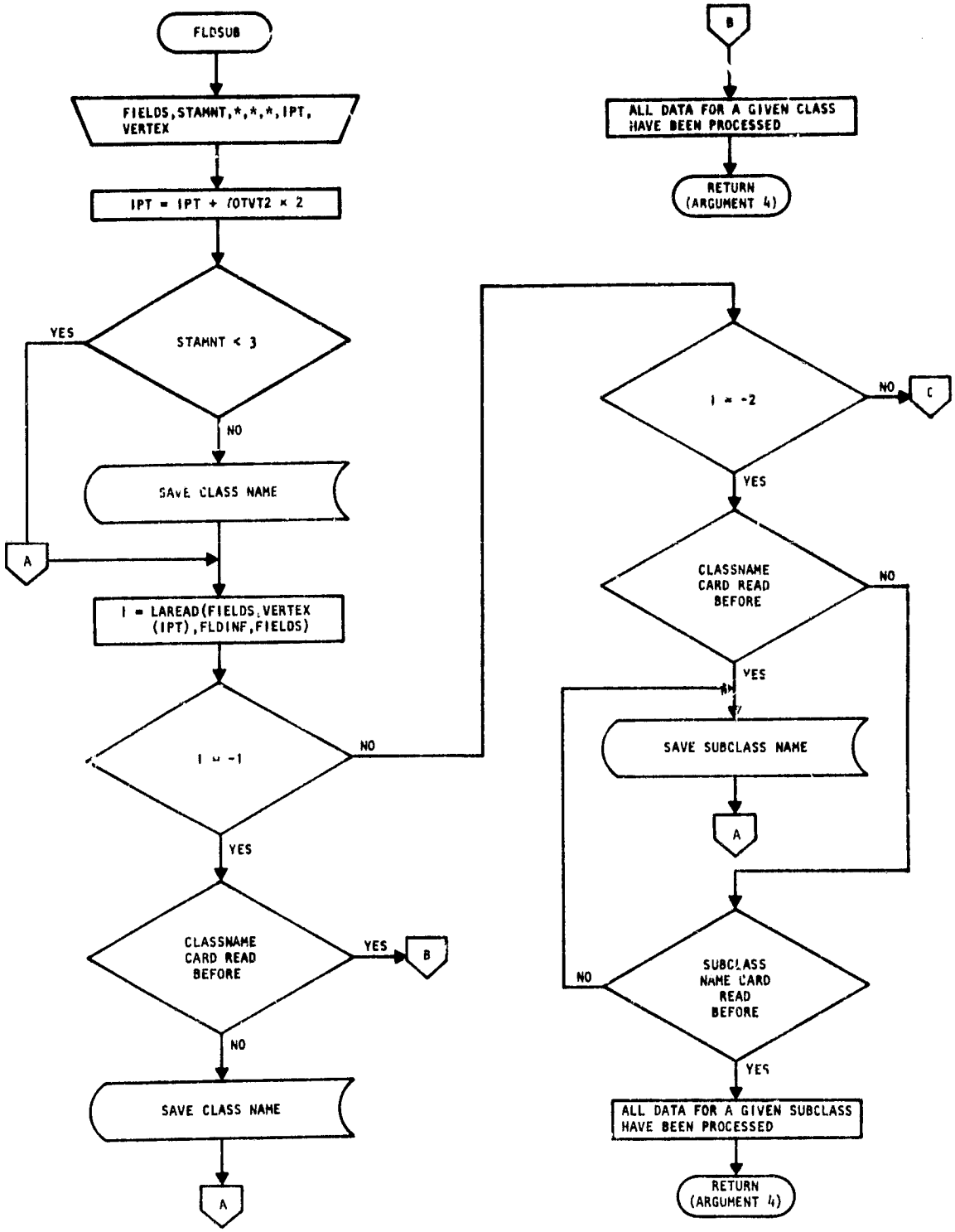


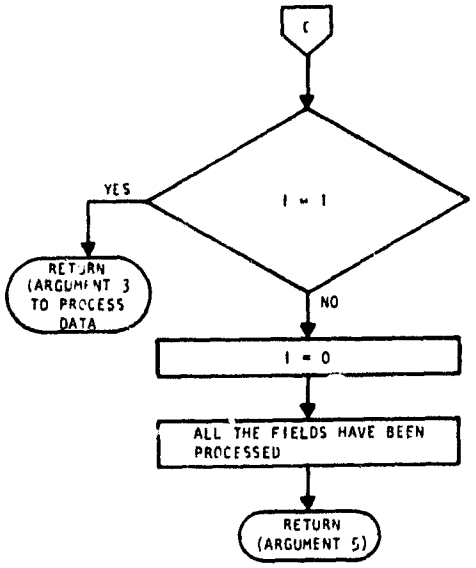


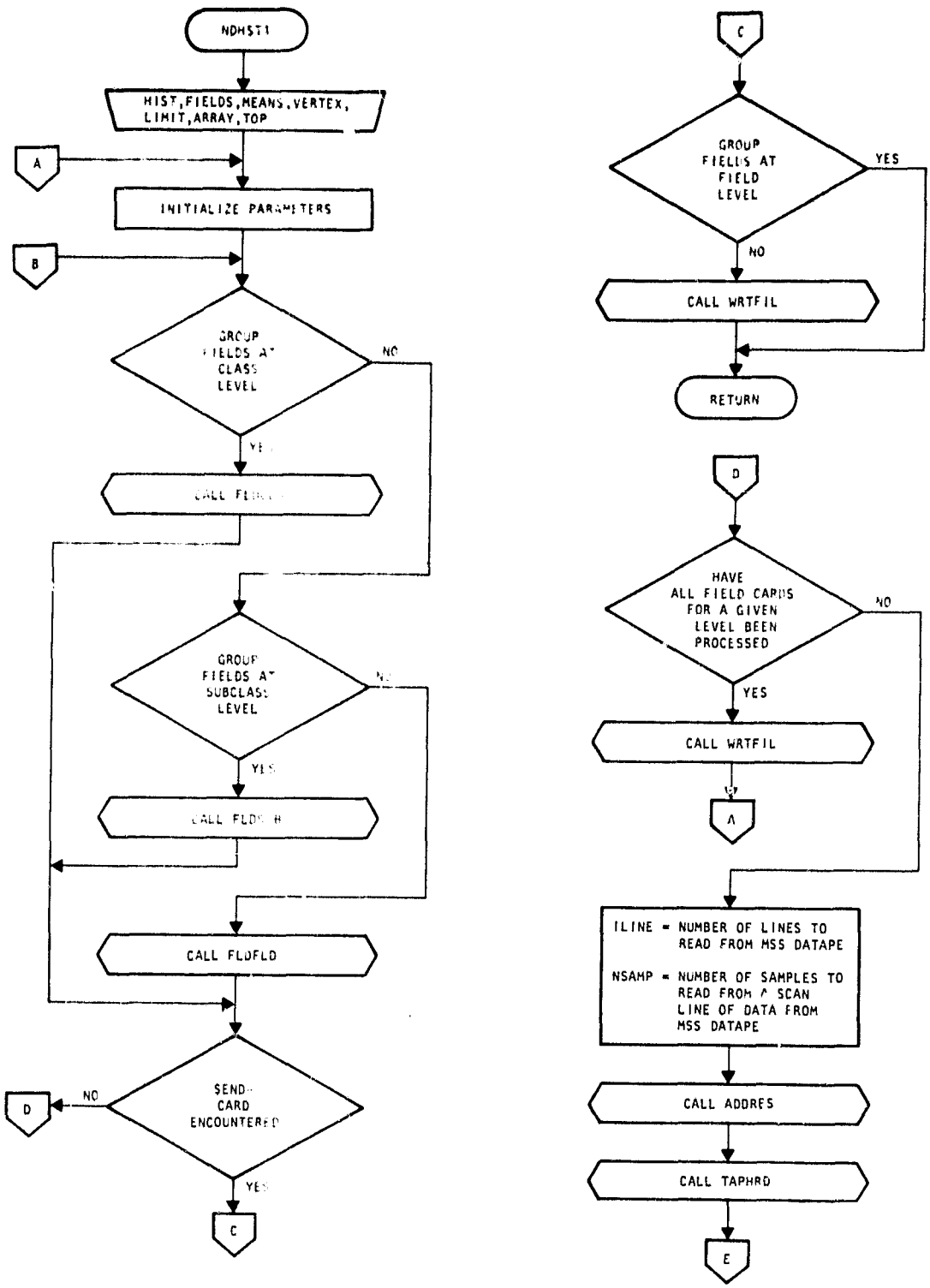

FLDCLS
FIELDS,STAMNT,*,*,*,IPT,VERTEX
IPT = IPT + TOTVT2 × 2
STAMNT = 2
NO
YES
REREAD CARD
SAVE CLASS NAME
I = LAREAD(FIELDS, VERTEX(IPT), FLDINF,FIELDS)
I = -1
NO
YES
CLASSNAME CARD READ BEFORE
NO
YES
A
A
ALL DATA FOR A GIVEN CLASS HAVE BCEN PROCESSED
RETURN (ARGUMENT 5)
I = -2
NO
YES
SAVE SUBCLASS NAME
I = 1
NO
YES
RETURN (ARGUMENT 3) TO PROCESS DATA
I = 0
ALL FIELD CARDS HAVE BEEN PROCESSED
RETURN (ARGUMENT 4)

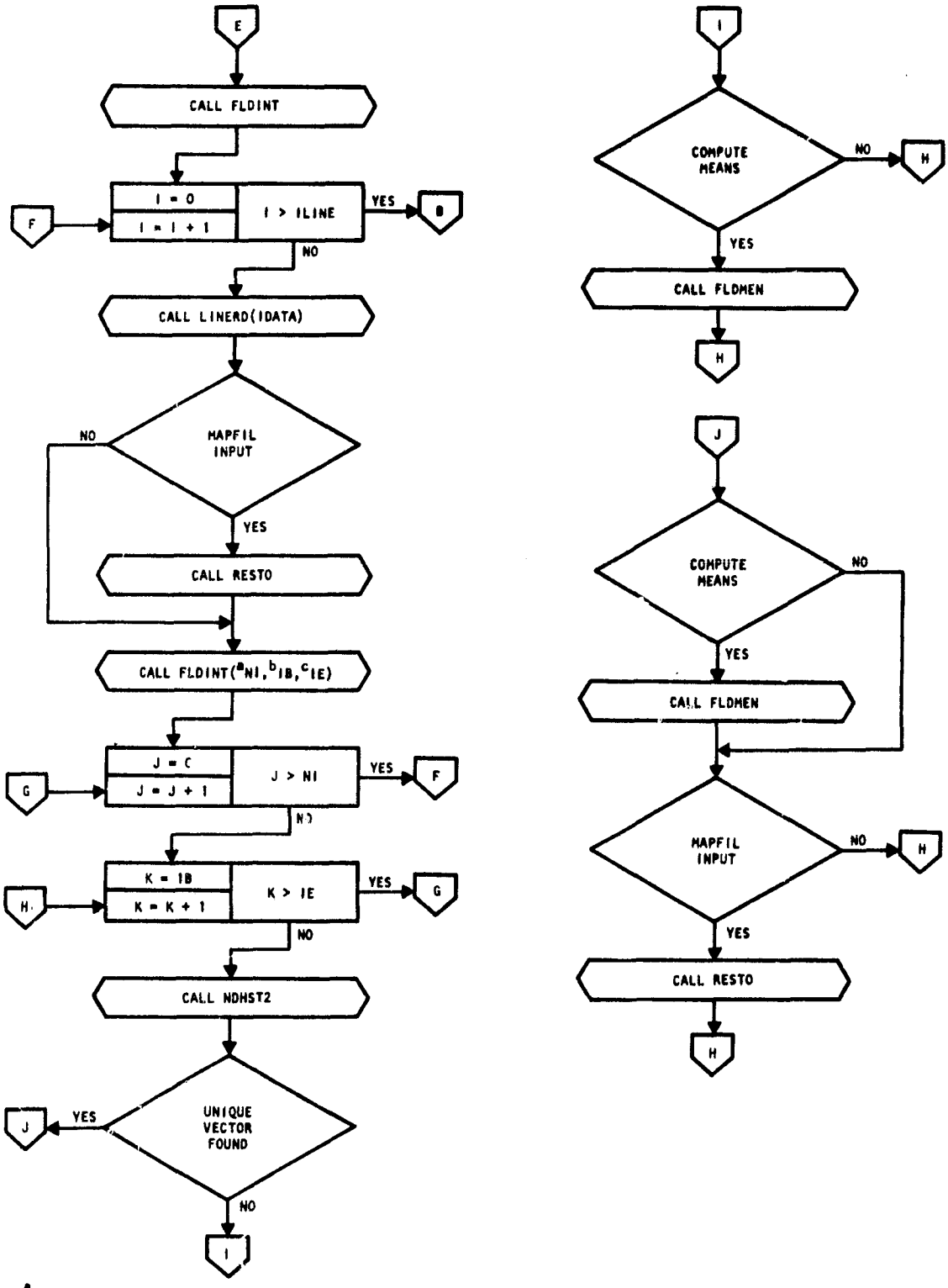

FLDFLD
FIELDS,STAHNT,*,*,IPT,VERTEX
IPT = 1
I = LAREAD(FIELDS,VERTEX (IPT),FLDINF,FIELDS)
I = -1
NO
YES
SAVE CLASS NAME
I = -2
NO
YES
SAVE SUBCLASS NAME
I = 1
YES
NO
RETURN (ARGUMENT 3) TO PROCESS DATA
I = 0
ALL DATA HAVE BEEN PROCESSED
RETURN (ARGUMENT 4)


FLDMEN
IDATA,J,NSAMP,NOFEAT,MEANS,
BGCHAN,N
RND = (N - 1)/N
I = 0
I = I + 1
I < NOFEAT
- BGCHAN
YES
NO
RETURN
MEANS(I,NOFLD2) = RND × MEANS(I,NOFLD2)
+ IDATA(J,I)/N

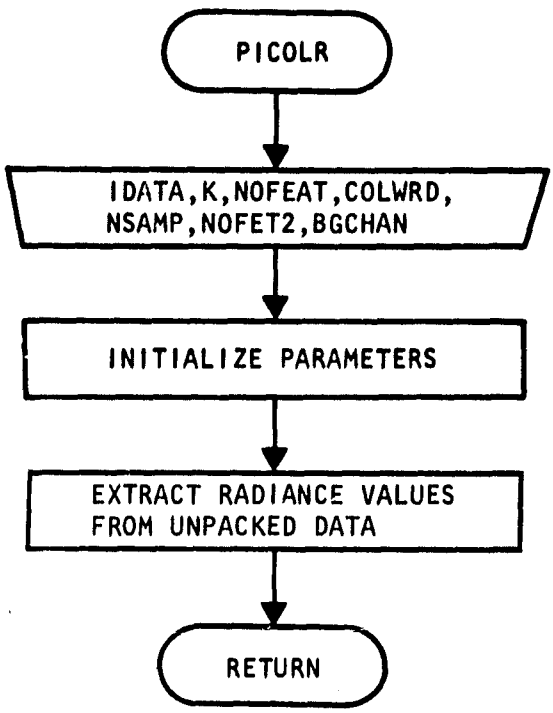

FLDSUB
FIELDS,STAMNT,*,*,*,IPT,
VERTEX
IPT = IPT + IOTVT2 * 2
STAMNT < 3
YES
NO
SAVE CLASS NAME
A
I = LAREAD(FIELDS,VERTEX
(IPT),FLDINF,FIELDS)
I = -1
NO
YES
CLASSNAME
CARD READ
BEFORE
YES
B
NO
SAVE CLASS NAME
A
B
ALL DATA FOR A GIVEN CLASS
HAVE BEEN PROCESSED
RETURN
(ARGUMENT 4)
I = -2
NO
C
YES
CLASSNAME
CARD READ
BEFORE
NO
YES
SAVE SUBCLASS NAME
A
NO
SUBCLASS
NAME CARD
READ
BEFORE
YES
ALL DATA FOR A GIVEN SUBCLASS
HAVE BEEN PROCESSED
RETURN
(ARGUMENT 4)

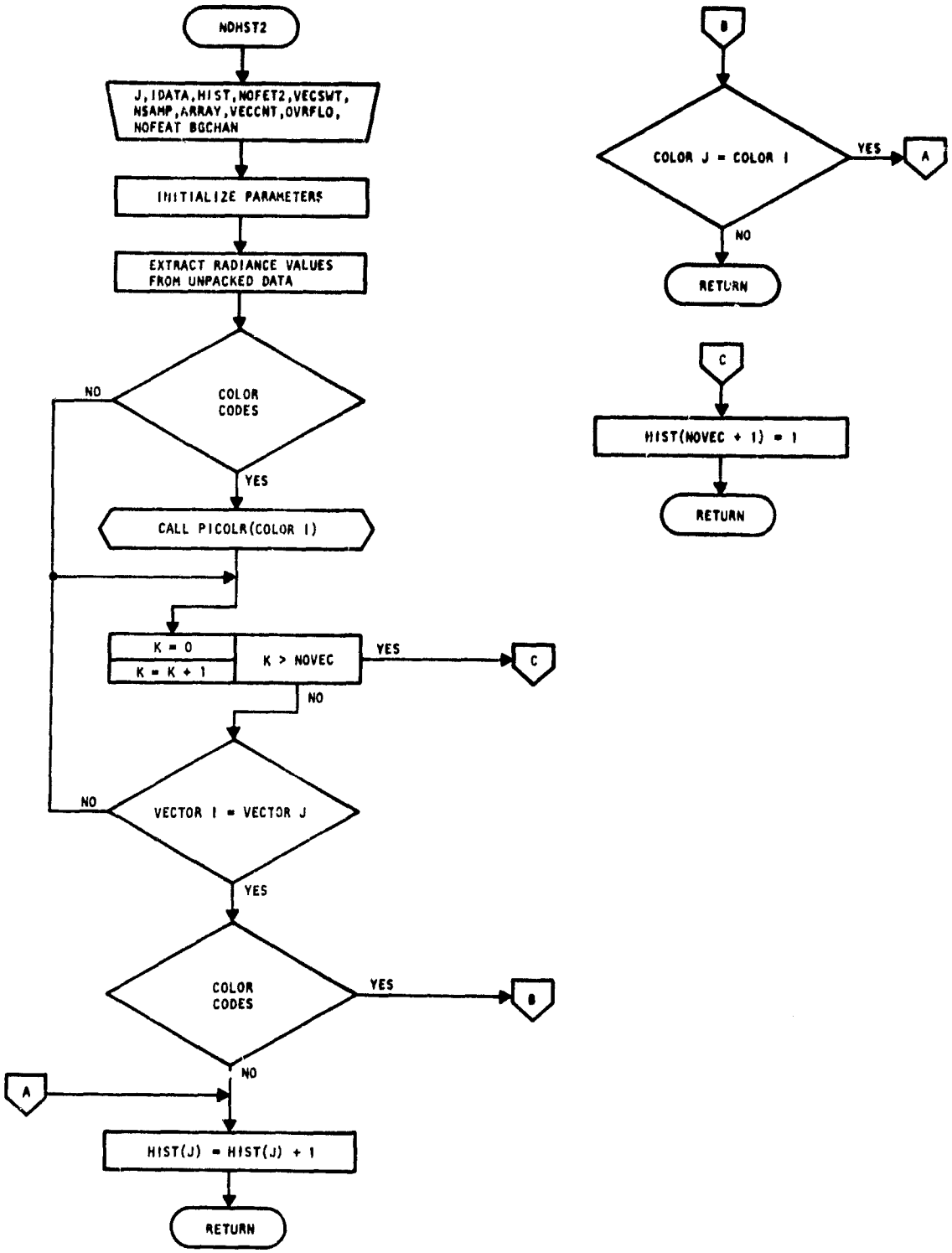

C
I = 1
YES
RETURN (ARGUMENT 3 TO PROCESS DATA
NO
I = 0
ALL THE FIELDS HAVE BEEN PROCESSED
RETURN (ARGUMENT 5)


NDHST1
HIST,FIELDS,MEANS,VERTEX,
LIMIT,ARRAY,TOP
A
INITIALIZE PARAMETERS
B
GROUP
FIELDS AT
CLASS
LEVEL
NO
YES
CALL FLDCLS
GROUP
FIELDS AT
SUBCLASS
LEVEL
NO
YES
CALL FLDSUB
CALL FLDFLD
SEND-
CARD
ENCOUNTERED
NO
D
YES
C
C
GROUP
FIELDS AT
FIELD
LEVEL
YES
NO
CALL WRTFIL
RETURN
D
HAVE
ALL FIELD CARDS
FOR A GIVEN
LEVEL BEEN
PROCESSED
NO
YES
CALL WRTFIL
A
ILINE = NUMBER OF LINES TO
READ FROM MSS DATAPE
NSAMP = NUMBER OF SAMPLES TO
READ FROM A SCAN
LINE OF DATA FROM
MSS DATAPE
CALL ADDRES
CALL TAPHRD
E

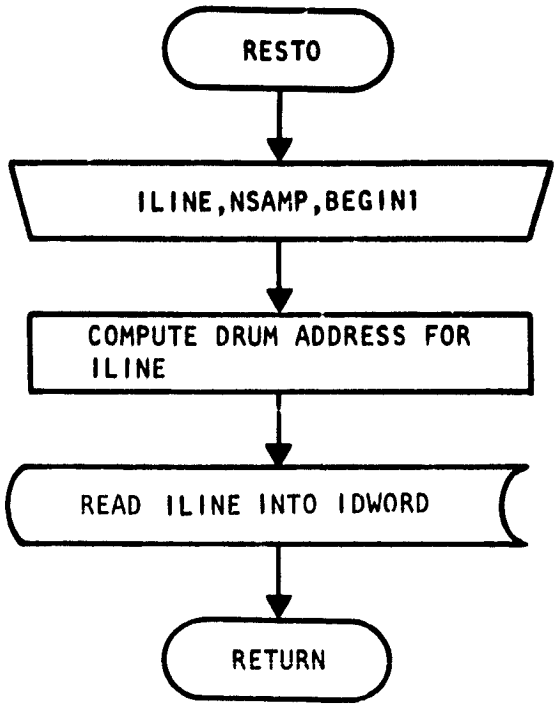


[a]NI = Number of intersections on a scan line.
[b]IB = Beginning index in IDATA for NI(J).
[c]IE = Ending index in IDATA for NI(J).

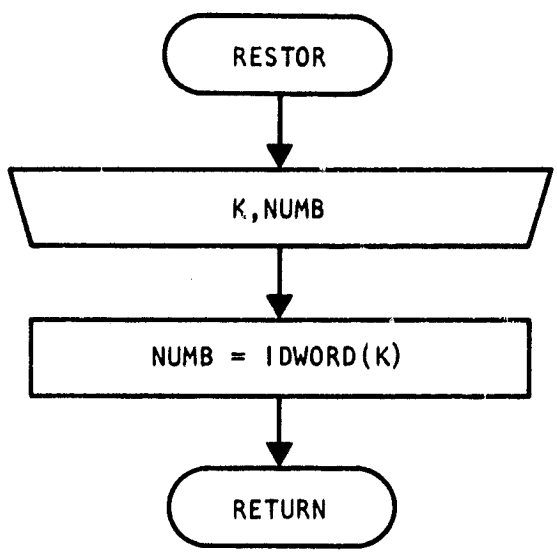

PICOLR
IDATA,K,NOFEAT,COLWRD,
NSAMP,NOFET2,BGCHAN
INITIALIZE PARAMETERS
EXTRACT RADIANCE VALUES
FROM UNPACKED DATA
RETURN

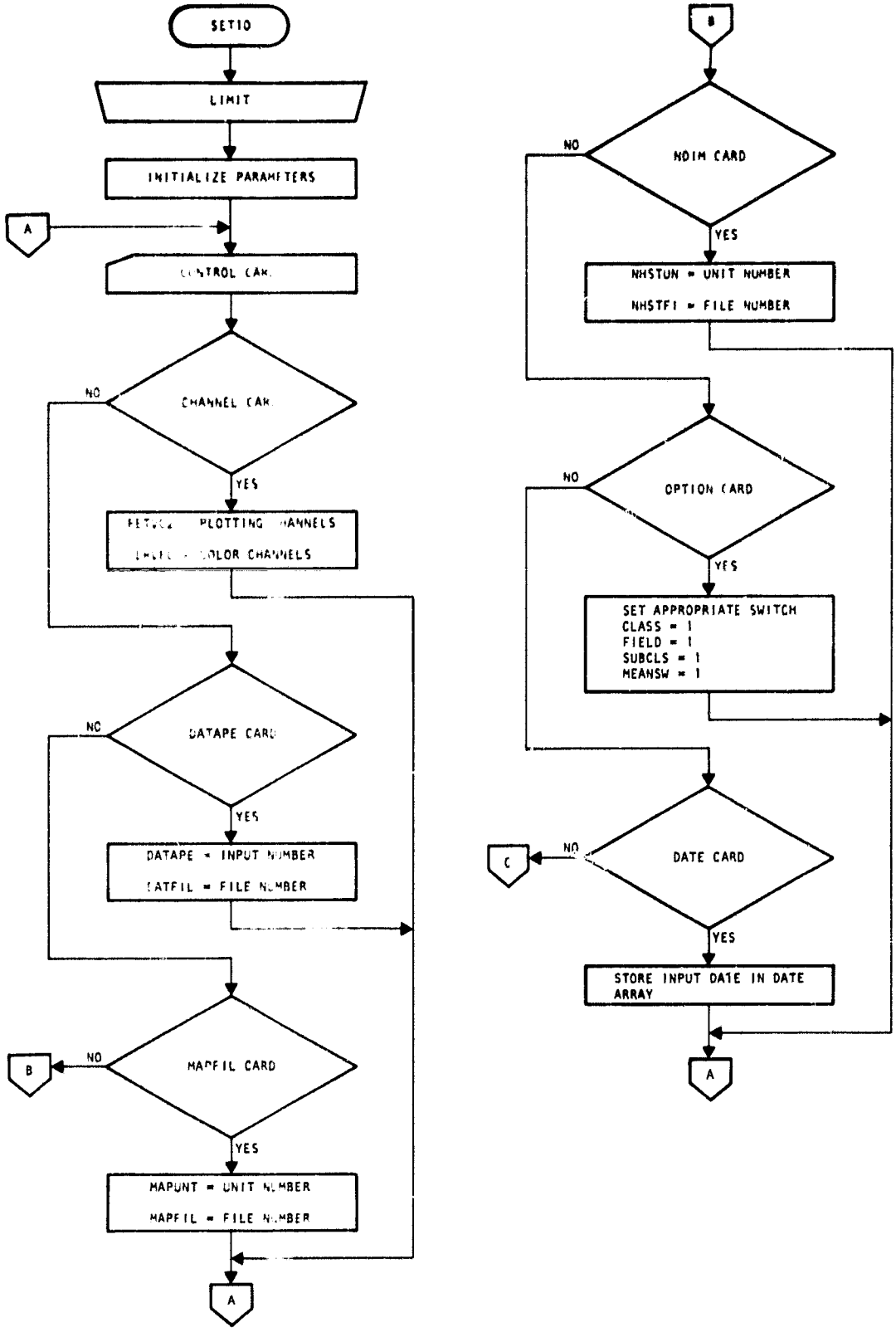

NDHST2
J, IDATA, HIST, NOFET2, VECSWT, NSAMP, ARRAY, VECCNT, OVRFLO, NOFEAT BGCHAN
INITIALIZE PARAMETERS
EXTRACT RADIANCE VALUES FROM UNPACKED DATA
COLOR CODES
NO
YES
CALL PICOLR(COLOR I)
K = 0
K = K + 1
K > NOVEC
YES
C
NO
VECTOR I = VECTOR J
NO
YES
COLOR CODES
YES
B
NO
A
HIST(J) = HIST(J) + 1
RETURN
B
COLOR J = COLOR I
YES
A
NO
RETURN
C
HIST(NOVEC + 1) = 1
RETURN

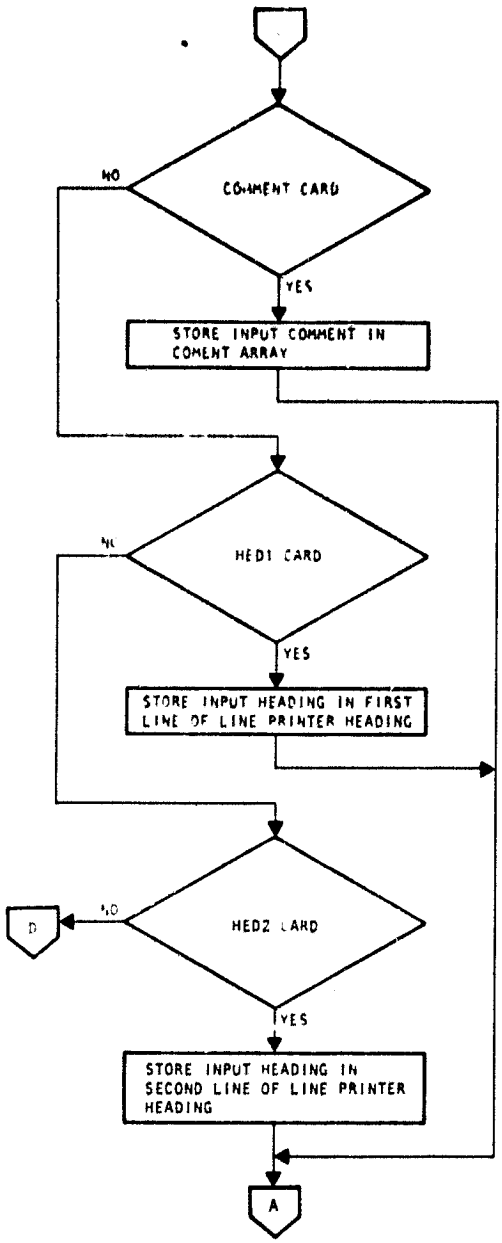
RESTO entry 1:

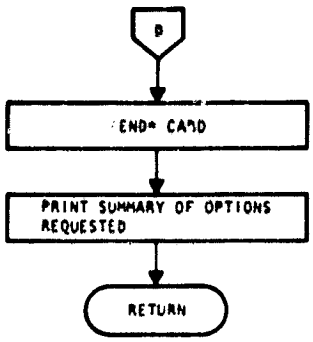


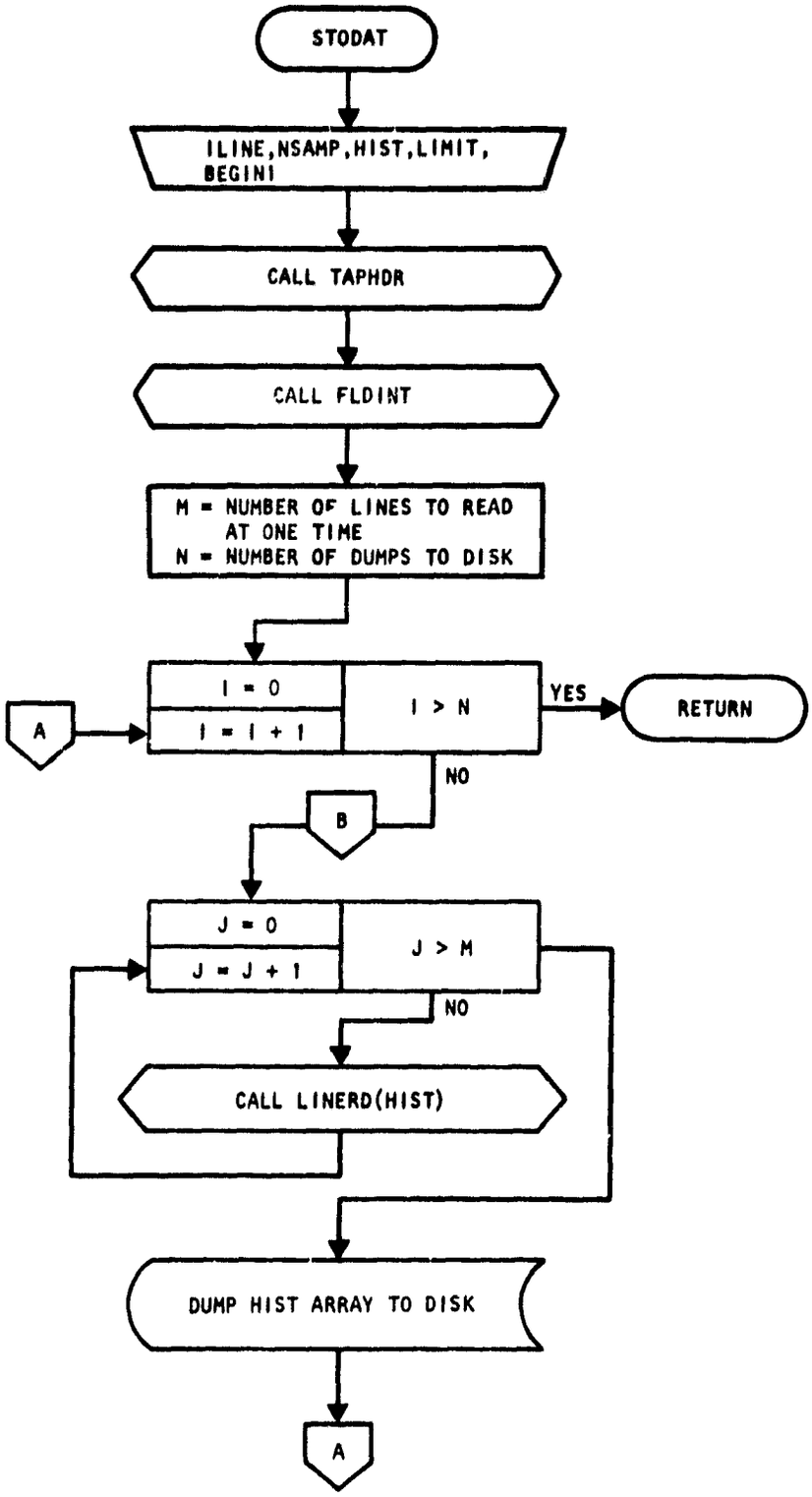
RESTO entry 2:

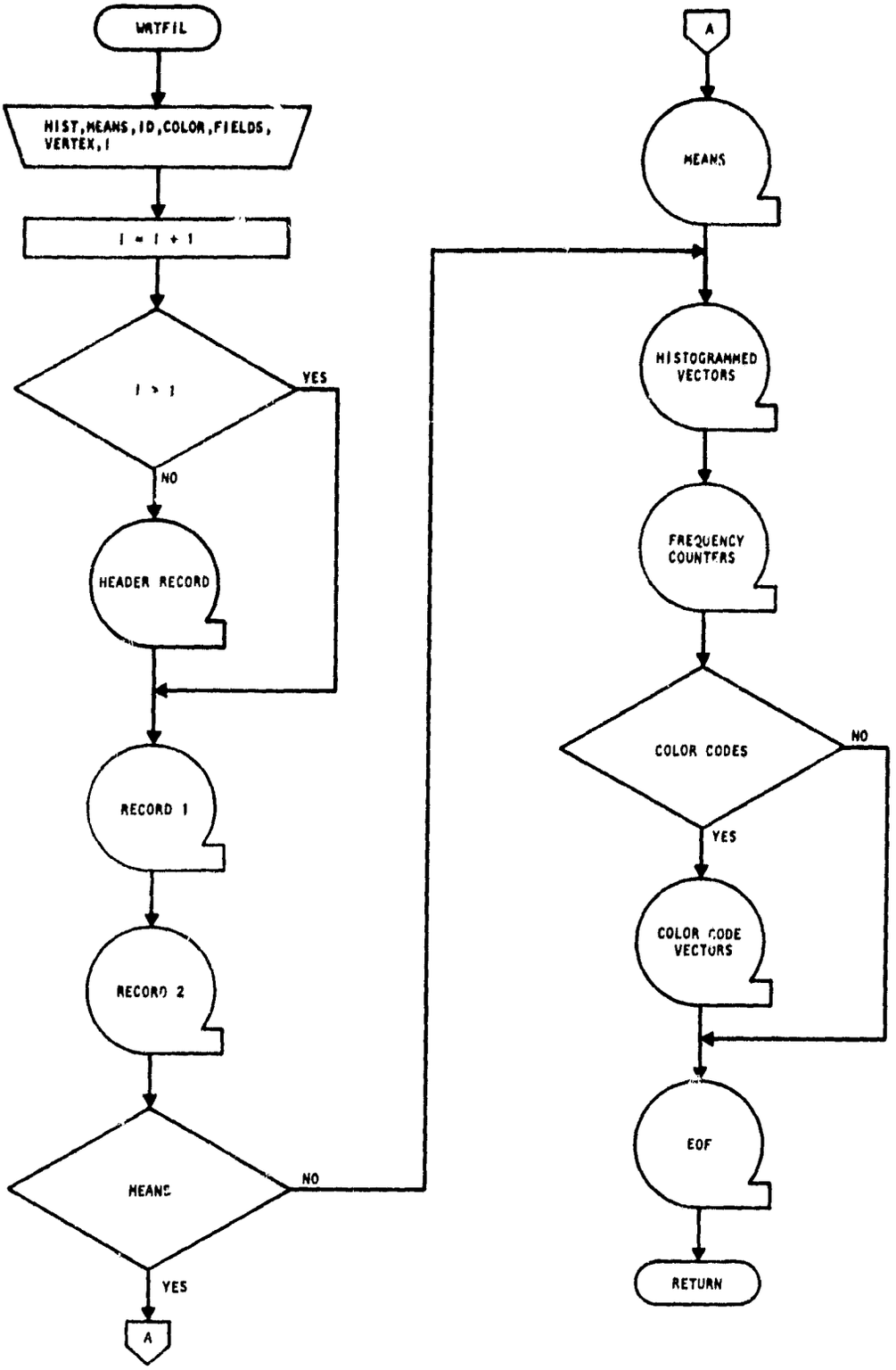



SETIO
LIMIT
INITIALIZE PARAMETERS
A
CONTROL CARD
CHANNEL CARD
NO
YES
SET PLOTTING CHANNELS
COLOR CHANNELS
DATAPE CARD
NO
YES
DATAPE = INPUT NUMBER
DATFIL = FILE NUMBER
MAPFIL CARD
B
NO
YES
MAPUNT = UNIT NUMBER
MAPFIL = FILE NUMBER
A
B
NDIM CARD
NO
YES
NHSTUN = UNIT NUMBER
NHSTFI = FILE NUMBER
OPTION CARD
NO
YES
SET APPROPRIATE SWITCH
CLASS = 1
FIELD = 1
SUBCLS = 1
MEANSW = 1
DATE CARD
C
NO
YES
STORE INPUT DATE IN DATE ARRAY
A


COMMENT CARD
NO
YES
STORE INPUT COMMENT IN COMENT ARRAY
HED1 CARD
NO
YES
STORE INPUT HEADING IN FIRST LINE OF LINE PRINTER HEADING
HED2 CARD
NO
YES
STORE INPUT HEADING IN SECOND LINE OF LINE PRINTER HEADING
D
A


D
END* CARD
PRINT SUMMARY OF OPTIONS REQUESTED
RETURN


STODAT
ILINE,NSAMP,HIST,LIMIT,
BEGINI
CALL TAPHDR
CALL FLDINT
M = NUMBER OF LINES TO READ
AT ONE TIME
N = NUMBER OF DUMPS TO DISK
A
I = 0
I = I + 1
I > N
YES
RETURN
NO
B
J = 0
J = J + 1
J > M
NO
CALL LINERD(HIST)
DUMP HIST ARRAY TO DISK
A


WRTFIL
HIST,MEANS,ID,COLOR,FIELDS,
VERTEX,I
I = I + 1
I > 1
YES
NO
HEADER RECORD
RECORD 1
RECORD 2
MEANS
NO
YES
A
A
MEANS
HISTOGRAMMED
VECTORS
FREQUENCY
COUNTERS
COLOR CODES
NO
YES
COLOR CODE
VECTORS
EOF
RETURN

16. SCTRPL PROCESSOR SUBPROGRAMS

The SCTRPL processor reads the NHSTUN file written by the NDHIST processor, determines the line and sample bin levels for each unique data vector, outputs a spectral plot in Universal format, and creates and stores a scatter plot for each file stored on NHSTUN. In addition to 16 subprograms that are exclusive to the processor, SCTRPL invokes 20 utility subprograms (documented in section 19). Figure 16-1 is a linkage diagram for the SCTRPL processor.

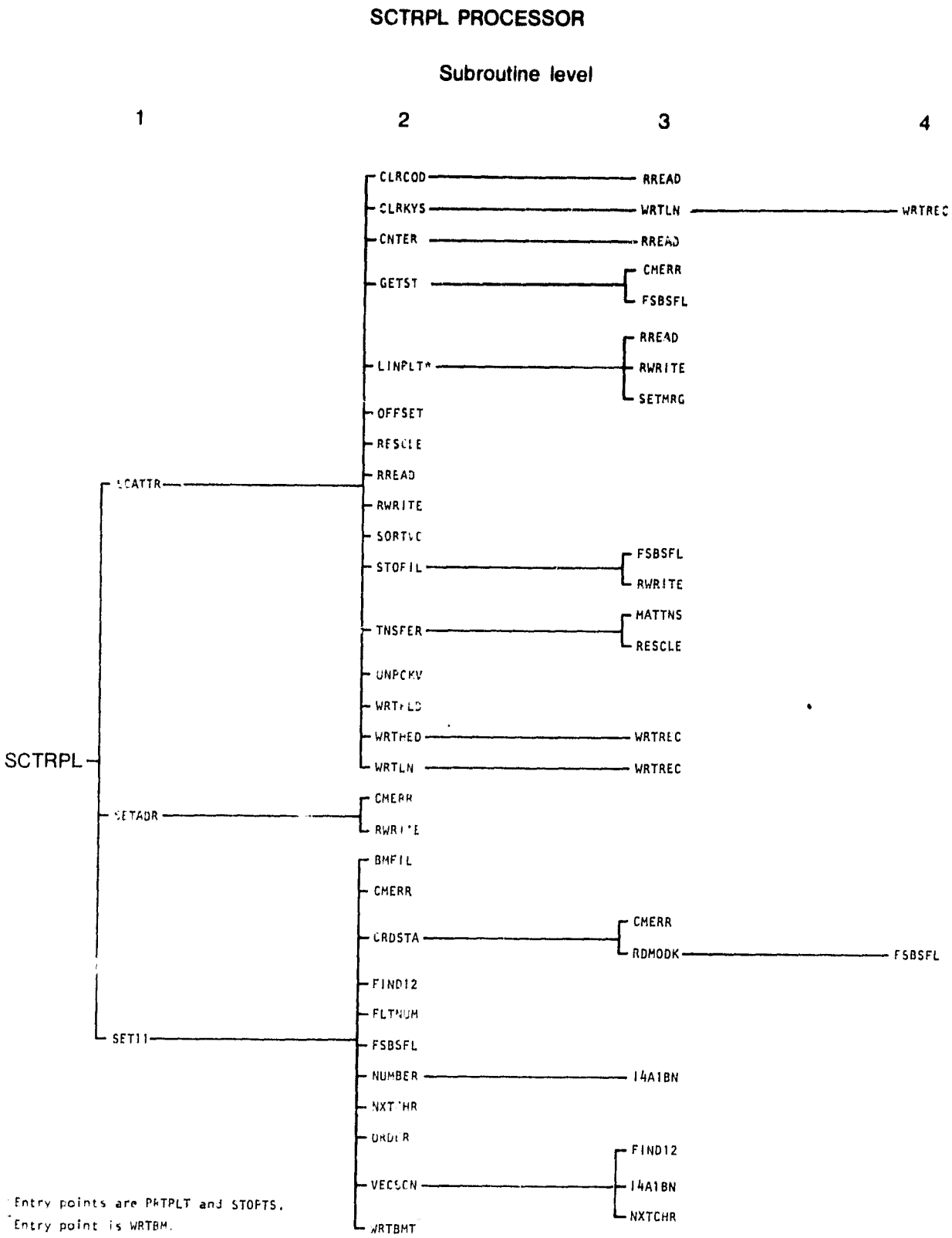


Figure 16-1.— Linkage diagram for the SCTRPL processor.

16.1 SCTRPL

The SCTRPL subprogram is the driver routine for the SCTRPL processor.

16.1.1 LINKAGES

This routine calls the SETADR, SET11, and SCATTR subprograms. It is called by CONTOR.

16.1.2 INTERFACES

The SCTRPL subprogram interfaces with other routines through common blocks INFORM and SCTTER and through the calling arguments.

16.1.3 INPUTS

Calling sequence: CALL SCTRPL(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.

16.1.4 OUTPUTS

Not applicable.

16.1.5 STORAGE REQUIREMENTS

This subprogram requires 48 720 bytes of storage.

16.1.6 DESCRIPTION

Storage is allocated to a large array BUFF in the SCTRPL routine. Dimensioned by the parameter LIMIT (= 12 000), BUFF is used as a

work array by several routines. Other subroutines perform tasks in executing SCTRPL: SET11 is called to read control card images; SETADR is called to read the NHSTUN tape and compute the base addresses for ARRAY; and SCATTR is called to coordinate the routines for outputting the scatter plot to tape.

16.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.1.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.2 CLRCOD

The CLRCOD subprogram retrieves the values to be used for the color codes to be output on the plot tape (PLOTAP).

16.2.1 LINKAGES

The CLRCOD subprogram calls the RREAD subprogram. It is called by SCATTR.

16.2.2 INTERFACES

The CLRCOD subprogram interfaces with other routines through common blocks INFORM and SCTTER and through the calling arguments.

16.2.3 INPUTS

Calling sequence: CALL CLRCOD(IB,MEANS,IDATA,IPOSTN,II)

Parameter	Dimension	In/out	Definition
IB	1	In	Index for computing the disk address for retrieving the packed color code of interest.
MEANS	1	In	Array containing the means of the test or training fields.
IDATA	1	Out	Output array containing the color code of interest.
IPOSTN	1	In	Index used to compute the position in IDATA for storing the color code of interest.
II	1	In	Index used in computing the disk address for retrieving the cluster, subclass, or field identification number of interest.

16.2.4 OUTPUTS

The results are returned for use by the calling routine.

16.2.5 STORAGE REQUIREMENTS

This subprogram requires 912 bytes of storage.

16.2.6 DESCRIPTION

The values to be used for the color codes may be extracted directly from the high-speed disk, one word at a time, or from an array in core.

16.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.2.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.3 CLRKYS

After the color-coded spectral plot has been output to tape, the subprogram CLRKYS adds the color keys to the spectral plot image.

16.3.1 LINKAGES

The CLRKYS subprogram calls the WRTLN subprogram. It is called by SCATTR.

16.3.2 INTERFACES

The CLRKYS subprogram interfaces with other routines through the calling arguments.

16.3.3 INPUTS

Calling sequence: CALL CLRKYS(XSIZ,IDATA,NOSUB2,CH,MEANS,NC)

Parameter	Dimension	In/out	Definition
XSIZ	1	In	Number of samples per scan line to output on tape.
IDATA	XSIZ,CH	Out	Array for storing a scan line of color-coded keys.
NOSUB2	1	In	Number of color keys to output.
CH	1	In	Number of channels to output on tape (NC + 1).
MEANS	NC,NOSUB2	In	Array containing the color codes.
NC	1	In	Number of channels in MEANS array.

16.3.4 OUTPUTS

This subprogram outputs the color keys on the SCTRUN tape in the Universal or LARSYS III format.

16.3.5 STORAGE REQUIREMENTS

This subprogram requires 1220 bytes of storage.

16.3.6 DESCRIPTION

Each color key is displayed as a 10-by-10 image with a field of zeros outlining each square. The number of keys per scan line is a function of the total number of samples per scan line output to tape.

16.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.3.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.4 CNTER

The CNTER subprogram retrieves the frequency count from the disk for the histogrammed vector of interest.

16.4.1 LINKAGES

The CNTER subprogram calls the RREAD subprogram. It is called by SCATTR.

16.4.2 INTERFACES

The CNTER subprogram interfaces with other routines through common block SCTTER and through the calling arguments.

16.4.3 INPUTS

Calling sequence: CALL CNTER(IB,IDATA,IPOSTN,II,COUNTR)

Parameter	Dimension	In/out	Definition
IB	1	In	Index for computing the disk address for retrieving the frequency of the histogrammed vector of interest.
IDATA	1	Out	Output array containing the frequency counter for the vector of interest.
IPOSTN	1	In	Index for storing the frequency in the IDATA array.
II	1	In	Index for computing the disk address for retrieving the frequency of the vector of interest.
COUNTR	1	In/out	Contains the frequency counter.

16.4.4 OUTPUTS

The results are returned for use by the calling routine.

16.4.5 STORAGE REQUIREMENTS

This subprogram requires 550 bytes of storage.

16.4.6 DESCRIPTION

The CNTER subprogram computes disk addresses, retrieves the frequency count for the histogrammed vector from the disk, and stores it in the IDATA array as the last channel on the output data set. If the frequency is greater than 255, it is reset to 255.

16.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.4.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.5 LINPLT

The LINPLT subprogram constructs the pixel frequency plot on disk and prints it on the line printer.

16.5.1 LINKAGES

The LINPLT subprogram calls the RREAD, RWRITE, and SETMRG subprograms. It is called by SCATTR.

16.5.2 INTERFACES

The LINPLT subprogram interfaces with other routines through common blocks GLOBAL and SCTTER and through the calling arguments.

16.5.3 INPUTS

Calling sequences:

a. CALL LINPLT

b. ENTRY STOPTS(COUNTR,LINE,SAMPLE)

Parameter	Dimension	In/out	Definition
COUNTR	1	In	Frequency counter for the pixel of interest.
LINE	1	In	Second coordinate of the pixel of interest.
SAMPLE	1	In	First coordinate of the pixel of interest.

c. ENTRY PRTPLT(PNTR,PNTRS)

Parameter	Dimension	In/out	Definition
PNTR	1	In	Array containing the integer frequency counters used in constructing the plot image.

Parameter	Dimension	In/out	Definition
PNTRS	101	In	Array containing the floating-point frequency counters used in constructing the plot image.

16.5.4 OUTPUTS

This subprogram stores the pixel frequency plot on disk and outputs it on the line printer.

16.5.5 STORAGE REQUIREMENTS

This subprogram requires 5702 bytes of storage.

16.5.6 DESCRIPTION

There are three entry points into the routine: LINPLT, STOPTS, and PRTPLT. LINPLT computes the scales for the x- and y-axes. The maximum resolution that may be plotted is 100. If needed, the data may be rescaled.

Using the coordinate of the pixel, STOPTS computes an address for storing the frequency or log of the frequency of occurrence on the high-speed disk for each histogrammed vector on the SCTRUN file. The entire plot is stored on the disk.

PRTPLT reads the plot of the frequency into core, assigns each frequency counter a symbol that represents the range of the frequency, and prints the symbolic image on the line printer.

16.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.5.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.6 MATTNS

The MATTNS subprogram multiplies a matrix A by a vector B to obtain a matrix C. A vector D is added to the one-dimensional vector C.

16.6.1 LINKAGES

The MATTNS subprogram does not call any other subprogram. It is called by subprogram TNSFER.

16.6.2 INTERFACES

The MATTNS subprogram interfaces with other routines through the calling arguments.

16.6.3 INPUTS

Calling sequence: CALL MATTNS(A,B,C,D,L,M)

Parameter	Dimension	In/out	Definition
A	L,M	In	An L-by-M matrix.
B	M	In	An M-by-1 vector.
C	L	Out	An L-by-1 vector.
D	L	In	An L-by-1 vector.
L	1	In	Number of rows in A, C, and D.
M	1	In	Number of columns in A and rows in B.

16.6.4 OUTPUTS

The results are returned for use by the calling routine.

16.6.5 STORAGE REQUIREMENTS

This subprogram requires 656 bytes of storage.

16.6.6 DESCRIPTION

Not required.

16.6.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.6.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.7 OFFSET

The OFFSET subprogram computes the value of each sample location on a scan line (x-axis) and computes the value of each line number (y-axis) output to the PLOTAP.

16.7.1 LINKAGES

The OFFSET subprogram does not call any other subprogram. It is called by the SCATTR subprogram.

16.7.2 INTERFACES

The OFFSET subprogram interfaces with other routines through common blocks GLOBAL and SCTTER and through the calling arguments.

16.7.3 INPUTS

Calling sequence: CALL OFFSET(YSCALE,XSCALE)

Parameter	Dimension	In/out	Definition
YSCALE	YSIZ	Out	Array containing the line numbers to be output on the PLOTAP.
XSCALE	XSIZ	Out	Array containing the sample values to be output on the PLOTAP.

16.7.4 OUTPUTS

The results are returned for use by the calling routine.

16.7.5 STORAGE REQUIREMENTS

This subprogram required 858 bytes of storage.

16.7.6 DESCRIPTION

The upper and lower limits of the sample and line values of each sample location are obtained by user input, input defaults, or by the minimum and maximum values of the transformed data vectors.

Scales are computed in one of the following ways.

a. If data have not been transformed (BMKEY = 0), user input or default values for the upper and lower ranges will be used.

b. If data have been transformed and rescaled (RESCALE = 1), the minimum and maximum ranges of the transformed data are used.

c. If data have been transformed but not rescaled (RESCALE = 0), user input or default values for the upper and lower ranges will be used.

Sample values and line numbers are computed and stored, respectively, in XSCALE and YSCALE.

16.7.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.7.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.8 RESCLE

The RESCLE subprogram rescales the transformed data to the range specified by the user.

16.8.1 LINKAGES

This routine does not call any other subprogram. It is called by the SCATTR and TNSFER subprograms.

16.8.2 INTERFACES

The RESCLE subprogram interfaces with other routines through common block SCTTER and through the calling arguments.

16.8.3 INPUTS

Calling sequence: CALL RESCLE(DATA,SWTCH,NVECT)

Parameter	Dimension	In/out	Definition
DATA	2,NVECT	In/out	As an input array or word, contains the transformed vector; as an output array or word, contains the rescaled vector.
SWTCH	1	In	Key indicating when to return to calling routine; if SWTCH = 1, rescale one vector; if SWTCH = 0, rescale NVECT vectors.
NVECT	1	In	Number of vectors to rescale.

16.8.4 OUTPUTS

The results are returned for use by the calling routine.

16.8.5 STORAGE REQUIREMENTS

This subprogram requires 700 bytes of storage.

16.8.6 DESCRIPTION

The routine may either rescale an entire array of transformed data before returning to the calling routine SCATTR or may rescale only one vector before returning to the calling routine TNSFER.

16.8.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.8.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.9 SCATTR

The subprogram SCATTR sets up the logic and structure for creating the color-coded spectral plots.

16.9.1 LINKAGES

The SCATTR subprogram calls the CLRCOD, CLRKYS, CNTER, GETST, LINPLT, OFFSET, RESCLE, RREAD, RWRITE, SORTVC, STOFIL, TNSFER, UNPCKV, WRTFLD, WRTHED, and WRTLN subprograms. It is called by the SCTRPL driver routine.

16.9.2 INTERFACES

The SCATTR subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and SCTTER and through the calling arguments.

16.9.3 INPUTS

Input to the SCATTR subprogram consists of the NHSTUN, SAVTAP, and BMFILE files output by the NDHIST, STAT, and SELECT processors, respectively.

Calling sequence: CALL SCATTR(FIELDS,VERTEX,TNSDAT,MEANS,PLOT, PNTR,IDATA,TOP,LIMIT,BUFF)

Parameter	Dimension	In/out	Definition
FIELDS	4,1	In	Array containing the histogrammed field information.
VERTEX	2,1	In	Array containing histogrammed field vertices.
TNSDAT	2,1	In	Array containing processed vectors to be used in creating the spectral plot.

Parameter	Dimension	In/out	Definition
MEANS	1	In	Array containing means of the fields.
PLOT	1	In	Array containing packed histogrammed vectors.
PNTR	1	In	Array containing pointers for retrieving the vectors in the TNSDAT array.
IDATA	XSIZ,NC	In	Array containing data to be output to tape.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
LIMIT	1	In	Maximum usable storage in the BUFF array; LIMIT = 12 000.
BUFF	1	In	Array containing the line-printer pixel-frequency plot image.

16.9.4 OUTPUTS

This subprogram outputs the scatter plot data on tape and prints scatter plot tape parameters and the scatter plot symbolic image on the line printer (subprogram LINPLT).

16.9.5 STORAGE REQUIREMENTS

This subprogram requires 8710 bytes of storage.

16.9.6 DESCRIPTION

The second record of the NHSTUN file is read and this information is passed to WRTFLD, which prints a summary of the histogrammed fields on the NHSTUN file.

Subroutine GETST retrieves a subset of the means from the SAVTAP file, if applicable. The means will be the color codes output on the PLOTAP file.

Subprogram STOFIL reads in the remainder of the NHSTUN file. The information is stored on the disk and retrieved as needed.

The histogrammed vectors are read into core by blocks. If the vectors are to be transformed, the subprogram TNSFER is called; if the vectors are not to be transformed, subprogram UNPCKV is called. These routines pass the unpacked data back into the TNSDAT array.

The two-component vectors in TNSDAT are sorted by subprogram SORTVC in descending order according to the second component. If applicable and if required at this point, subprogram RESCLE rescales the transformed vectors in the TNSDAT array.

The OFFSET subprogram computes the sample values for each pixel location (x-axis) and the line number (y-axis) to be output on the PLOTAP file. A summary of the information being output to the PLOTAP file is printed on the line printer.

Subroutine WRTHED outputs the header record to tape, and the LINPLT subprogram computes the scales for the line-printer pixel-frequency plot, if applicable. The parameter YSIZ determines the number of lines, and the parameter XSIZ determines the number of samples to output to tape.

All the vectors that belong to one scan line are collected. For each vector of interest, the sample value (first element of the vector) determines the location within the IDATA array for storing the color code. Subprogram CLRCOD retrieves the appropriate color code; subprogram CNTER retrieves the frequency for

the vector of interest; and, if applicable, the LINPLT subprogram (entry STOPTS) computes the position of the vector on the line printer plot. Subroutine WRTLN outputs the data set to tape.

After creation of the spectral plot, subprogram CLRKYS adds the color keys to the image. If applicable, the LINPLT subprogram (entry PRTPLT) outputs the pixel-frequency plot to the line printer.

16.9.7 FLOW CHART

The available subprogram flow charts for the processor are provided in section 16.17.

16.9.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.10 SETADR

The SETADR subprogram computes addresses for ARRAY in the blank common block.

16.10.1 LINKAGES

The SETADR subprogram calls the CMERR and RWRITE subprograms. It is called by the SCTRPL driver routine.

16.10.2 INTERFACES

The SETADR subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and SCTTER and through the calling arguments.

16.10.3 INPUTS

Input to the SETADR subprogram consists of the NHSTUN file output by the NDHIST processor and the SAVTAP file output by the STAT or ISOCLS processor.

Calling sequence: CALL SETADR(*,*,TOP,BUFF,LIMIT)

Parameter	Dimension	In/out	Definition
*	1	In	Continue to process file.
*	1	In	All files on NHSTUN have been processed.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
BUFF	1	In	Temporary storage used to zero out a portion of the disk.
LIMIT	1	In	Maximum usable storage in BUFF array; LIMIT = 12 000.

16.10.4 OUTPUTS

Computed addresses and parameters read from the NHSTUN file are stored in common blocks SCTTER and INFORM.

16.10.5 STORAGE REQUIREMENTS

This subprogram requires 1662 bytes of storage.

16.10.6 DESCRIPTION

The SETADR subprogram computes addresses with respect to ARRAY. Storage is allocated to the various routines by the variable dimensioning technique.

The field information and color codes (input by the user or from the SAVTAP file) are stored in the top portion of ARRAY. The unpacked, transformed, or rescaled vectors from the NHSTUN file are stored in the remainder of ARRAY. The addresses required for storing the NHSTUN file on high-speed disk are computed and stored in common block SCTTER. The first data record from the NHSTUN file is read, and the information is used to compute these addresses.

If applicable, 10 201 words on the disk are initialized to zero.

16.10.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.10.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.11 SET11

The SET11 subprogram reads and analyzes the control cards for the SCTRPL processor.

16.11.1 LINKAGES

The SET11 subprogram calls the BMFIL, CMERR, CRDSTA, FIND12, FLTNUM, FSBSFL, NUMBER, NXTCHR, ORDER, VECSCN, and WRTBMT subprograms. It is called by the SCTRPL driver routine.

16.11.2 INTERFACES

The SET11 subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and SCTTER and through the calling arguments.

16.11.3 INPUTS

Input to the SET11 subprogram consists of the NHSTUN file output by the NDHIST processor.

Calling sequence: CALL SET11(MEANS,MENS,BUFF)

Parameter	Dimension	In/out	Definition
MEANS	1	In	Array containing the integer color codes input by the user.
MENS	1	Out	Array containing the floating-point color codes.
BUFF	1	In	Array used as temporary storage for module STAT file.

The control cards relevant to this routine are given in section 16 (table 16-1) of volume II of this user guide.

16.11.4 OUTPUTS

This subprogram outputs a summary report of the user-requested options on the line printer. Depending on user requests, various switches and parameters are initialized and stored in common blocks INFORM, GLOBAL, and SCTTER.

16.11.5 STORAGE REQUIREMENTS

This subprogram requires 8512 bytes of storage.

16.11.6 DESCRIPTION

The SET11 subprogram sets default values, initializes parameters, and prints an input summary. It then sets up the reread buffer and reads input control card images, generating error messages if invalid cards are read. It calls subprogram BMFIL to read in the B-matrix from card images or file and compares input channels with the number of plotting channels. If the two channel sets are not equal, it generates an error message. Color codes are read in from tape or file, and a list of user-selected options is printed.

16.11.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.11.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.12 SORTVC

The SORTVC subprogram sorts the elements of a floating-point array in descending order.

16.12.1 LINKAGES

This routine does not call any other subprogram. It is called by SCATTR.

16.12.2 INTERFACES

The SORTVC subprogram interfaces with other routines through the calling arguments.

16.12.3 INPUTS

Calling sequence: CALL SORTVC(HIST,PNTR,ICOL,NOVEC,IBG,IEN,II)

Parameter	Dimension	In/out	Definition
HIST	ICOL,NOVEC	In	Array containing vectors to be sorted.
PNTR	1	Out	Array containing pointers for ordering the array HIST.
ICOL	1	In	Number of the row in HIST to order.
NOVEC	1	In	Number of elements to sort.
IBG	1	Out	First location in the HIST array to begin retrieving the vectors.
IEN	1	Out	Ending location in the HIST array to stop retrieving the vectors.
II	1	In	Index for IBG and IEN arrays.

16.12.4 OUTPUTS

The results are returned for use by the calling routine.

16.12.5 STORAGE REQUIREMENTS

This subprogram requires 1116 bytes of storage.

16.12.6 DESCRIPTION

The sorting is performed on the ICOL row of the floating-point array. The pointers for ordering the input array HIST are passed back to the calling routine SCATTR in the array PNTR.

16.12.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.12.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.13 STOFIL

The STOFIL subprogram reads the remainder of the NHSTUN file and stores the information on high-speed disk for later retrieval.

16.13.1 LINKAGES

The STOFIL subprogram calls the FSBSFL and RWRITE subprograms. It is called by SCATTR.

16.13.2 INTERFACES

The STOFIL subprogram interfaces with other routines through common blocks GLOBAL and SCTTER and through the calling arguments.

16.13.3 INPUTS

Input to the STOFIL subprogram consists of the NHSTUN file output by the NDHIST processor.

Calling sequence: CALL STOFIL(LIMIT,MEANS,BUFF)

Parameter	Dimension	In/out	Definition
LIMIT	1	In	Dimensions the array BUFF by setting the parameter at 12 000.
MEANS	1	Out	Array containing the field means.
BUFF	1	In	Array for temporary storage.

16.13.4 OUTPUTS

This subprogram stores the results on high-speed disk.

16.13.5 STORAGE REQUIREMENTS

This subprogram requires 1434 bytes of storage.

16.13.6 DESCRIPTION

The remainder of the NHSTUN file consists of the field means (if applicable); the data vectors; the field, subclass, or cluster numbers; the frequency counters; and the color codes (if applicable). STOFIL reads these data and, after each read, stores the information on high-speed disk for later retrieval. STOFIL then calls FSBSFL to position the NHSTUN past the EOF of the file just read.

16.13.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.13.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.14 TNSFER

The TNSFER subprogram coordinates the reduction of each 1- to 16-element data vector from the NHSTUN file to 2 components.

16.14.1 LINKAGES

This routine calls the MATTNS and RESCLE subprograms. It is called by SCATTR.

16.14.2 INTERFACES

The TNSFER subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and SCTTER and through the calling arguments.

16.14.3 INPUTS

Calling sequence: CALL TNSFER(PLOT,TNSDAT,NVECT,IAAA)

Parameter	Dimension	In/out	Definition
PLOT	SIZE,NVECT	In	Array containing the packed data vectors; SIZE = $\frac{\text{NOFET2}}{4}$.
TNSDAT	2,NVECT	In	Array containing the transformed and/or rescaled data vectors.
NVECT	1	In	Number of vectors in PLOT array.
IAAA	1	In	Key indicating that two or more blocks of data vectors have been transformed.

16.14.4 OUTPUTS

This subprogram stores the minimum and maximum values for each component in common block SCTTER.

16.14.5 STORAGE REQUIREMENTS

This subprogram requires 1286 bytes of storage.

16.14.6 DESCRIPTION

The subroutine TNSFER unpacks the data vector, and the MATTNS subprogram performs the transformation. If applicable, RESCLE rescales the two-component vector to a user-specified range.

The minimum and maximum values of each component are determined and saved.

16.14.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.14.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.15 UNPCKV

The subprogram UNPCKV unpacks a two-channel vector from the NHSTUN file and stores the unpacked data in the TNSDAT array.

16.15.1 LINKAGES

The UNPCKV subprogram does not call any other subprogram. It is called by SCATTR.

16.15.2 INTERFACES

The UNPCKV subprogram interfaces with other routines through common block SCTTER and through the calling arguments.

16.15.3 INPUTS

Calling sequence: CALL UNPCKV(PLOT,TNSDAT,NVECT)

Parameter	Dimension	In/out	Definition
PLOT	1	In	Array containing the packed data.
TNSDAT	2,NVECT	Out	Array containing the unpacked data.
NVECT	1	In	Number of vectors in the PLOT array.

16.15.4 OUTPUTS

This subprogram stores the results in common block SCTTER.

16.15.5 STORAGE REQUIREMENTS

This subprogram requires 568 bytes of storage.

16.15.6 DESCRIPTION

The UNPCKV subroutine accomplishes the unpacking by shifting bits and using the Fortran AND function.

16.15.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 16.17.

16.15.8 LISTING

The subprogram listing is provided in volume IV, section 16.

16.16 VECSCN

The VECSCN function scans the COLOR control card, changes alphanumeric characters to integer mode, and returns one character at a time to the calling routine.

16.16.1 LINKAGES

The VECSCN function calls the FIND12, I4A1BN, and NXTCHR subprograms. It is called by SET11.

16.16.2 INTERFACES

The VECSCN subprogram interfaces with other routines through the calling arguments.

16.16.3 INPUTS

Calling sequence: VECSCN(VECTR,NVCELT,CARD,COL)

Parameter	Dimension	In/out	Definition
VECTR	1	Out	Array containing the user-input color codes.
NVCELT	1	Out	Number of color channels.
CARD	1	In	Array containing the alphanumeric data.
COL	1	In	Pointer for the function NXTCHR.

The COLOR control card is described in section 16 (table 16-1) of volume II of this user guide.

16.16.4 OUTPUTS

This subprogram returns one character at a time to the calling routine and stores converted integer numbers in the array VECTR.

16.16.5 STORAGE REQUIREMENTS

This subprogram requires 1224 bytes of storage.

16.16.6 DESCRIPTION

The VECSCN function uses the function NXTCHR to return one character at a time to the calling routine and the subprogram I4A1BN to change the characters from alphanumeric to integer mode.

Elements are input on card images by groups. A multiplicative factor k may be input to repeat a group of elements. To determine the number of groups of elements being input, the total number of elements or implied number of elements on the card (TOTNUM) is divided by the number of elements (NVCELT) within the parentheses. The number of elements within the parentheses represents the number of color channels, and the number of groups of elements is the number of colors to output on the PLOTAP file.

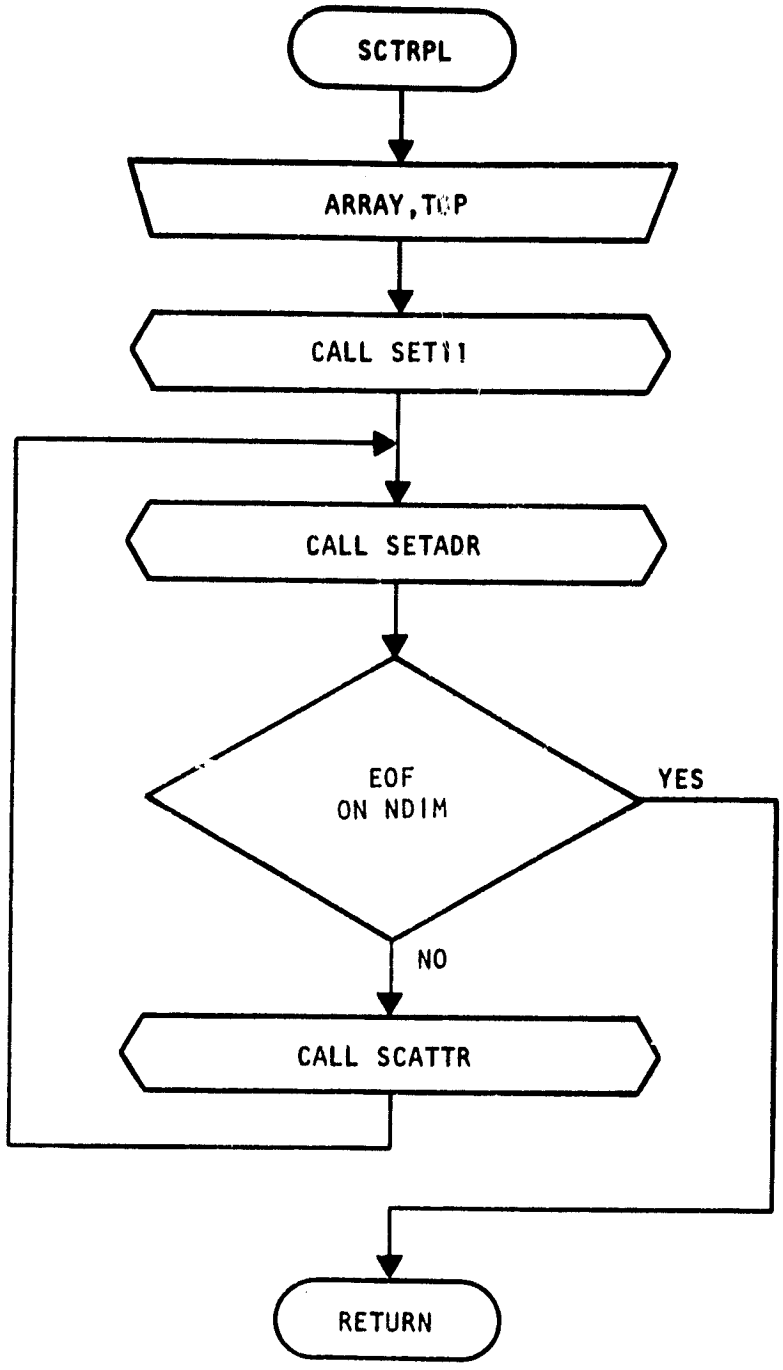
16.16.7 FLOW CHART

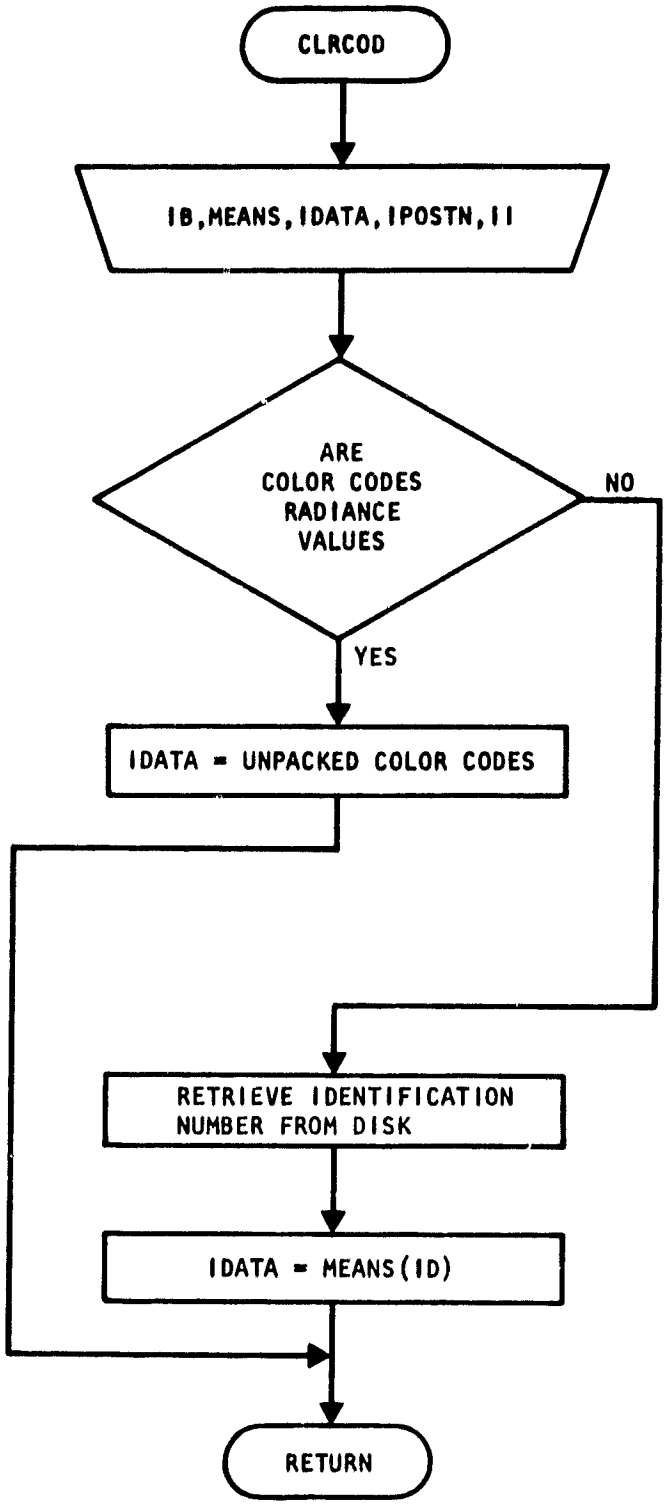
The available subprogram flow charts for this processor are provided in section 16.17.

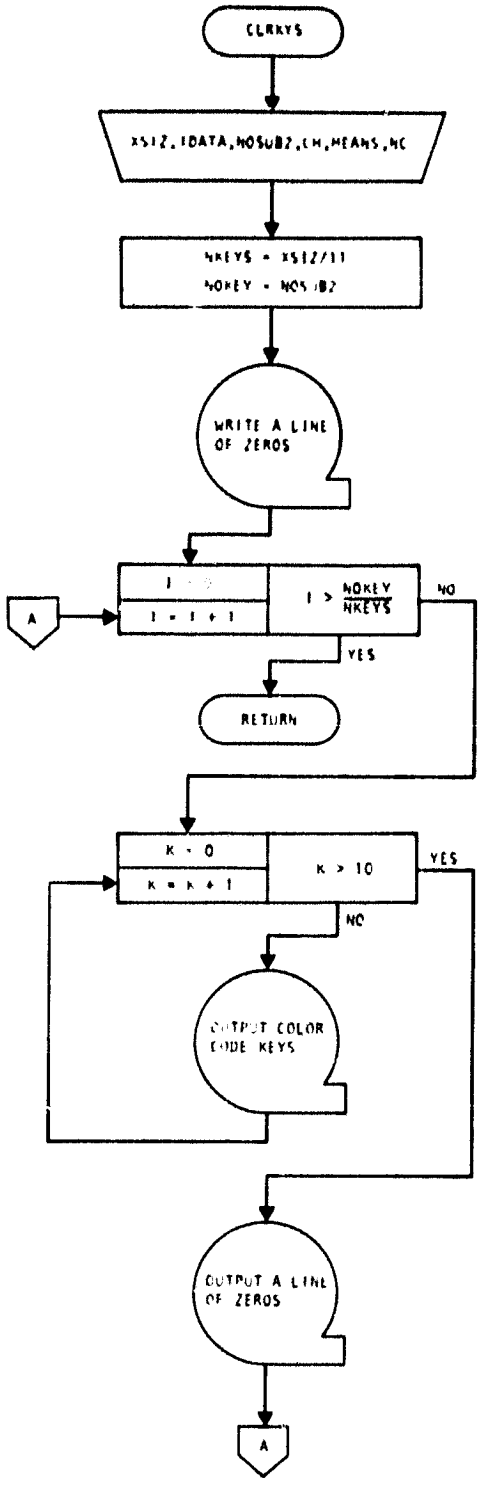
16.16.8 LISTING

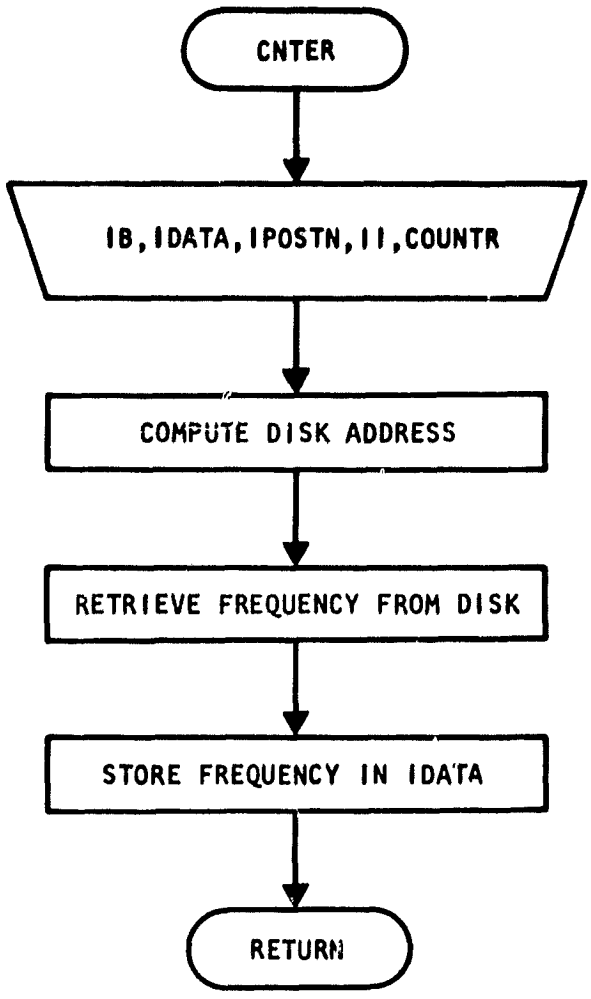
The subprogram listing is provided in volume IV, section 16.

16.17 SUBPROGRAM FLOW CHARTS

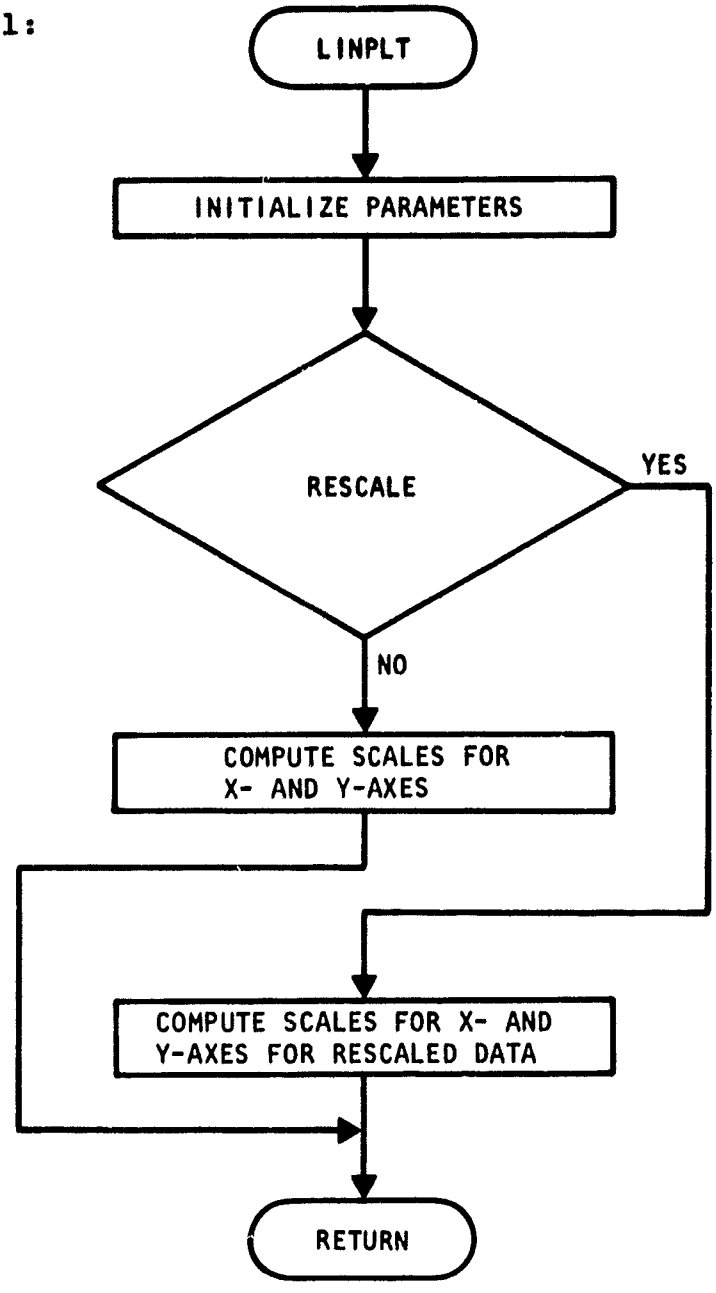



CLRCOD
IB,MEANS,IDATA,IPOSTN,II
ARE
COLOR CODES
RADIANCE
VALUES
NO
YES
IDATA = UNPACKED COLOR CODES
RETRIEVE IDENTIFICATION
NUMBER FROM DISK
IDATA = MEANS(ID)
RETURN

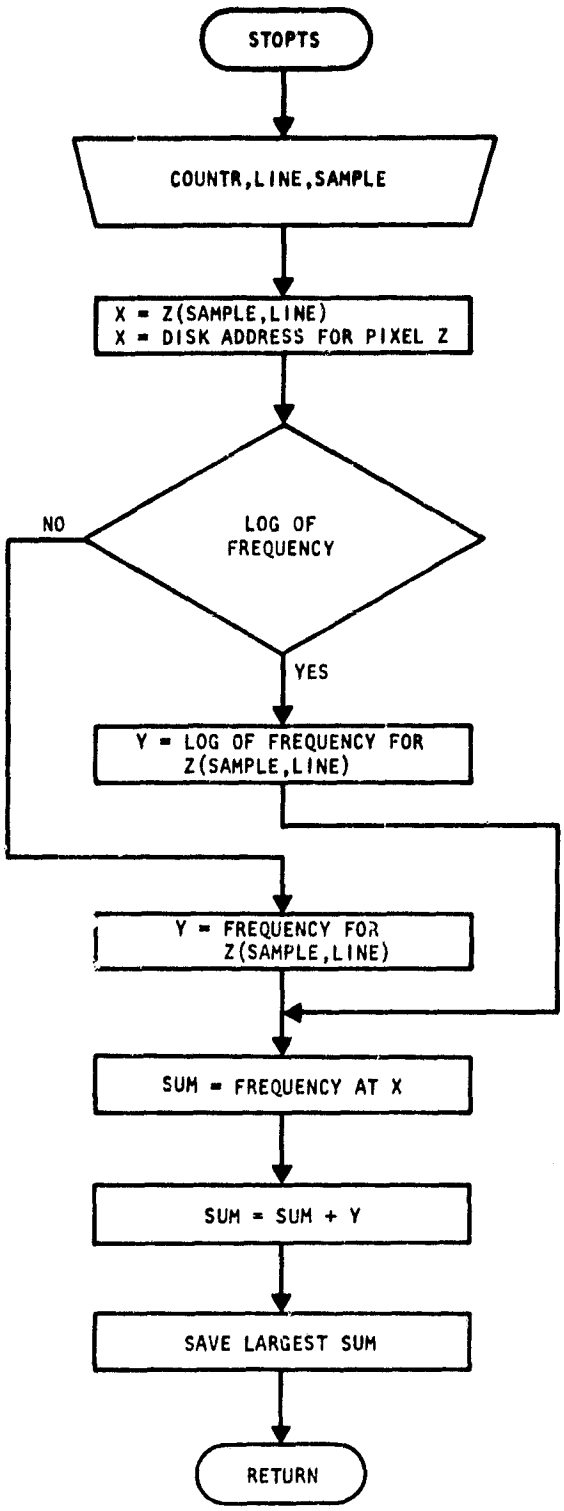

CLRKYS
XSIZ, IDATA, NOSUBZ, CH, MEANS, NC
NKEYS = XSIZ/11
NOKEY = NOSUBZ
WRITE A LINE OF ZEROS
I = 0
I = I + 1
I > NOKEY/NKEYS
NO
YES
A
RETURN
K = 0
K = K + 1
K > 10
YES
NO
OUTPUT COLOR CODE KEYS
OUTPUT A LINE OF ZEROS
A


CNTER
IB, IDATA, IPOSTN, II, COUNTR
COMPUTE DISK ADDRESS
RETRIEVE FREQUENCY FROM DISK
STORE FREQUENCY IN IDATA
RETURN

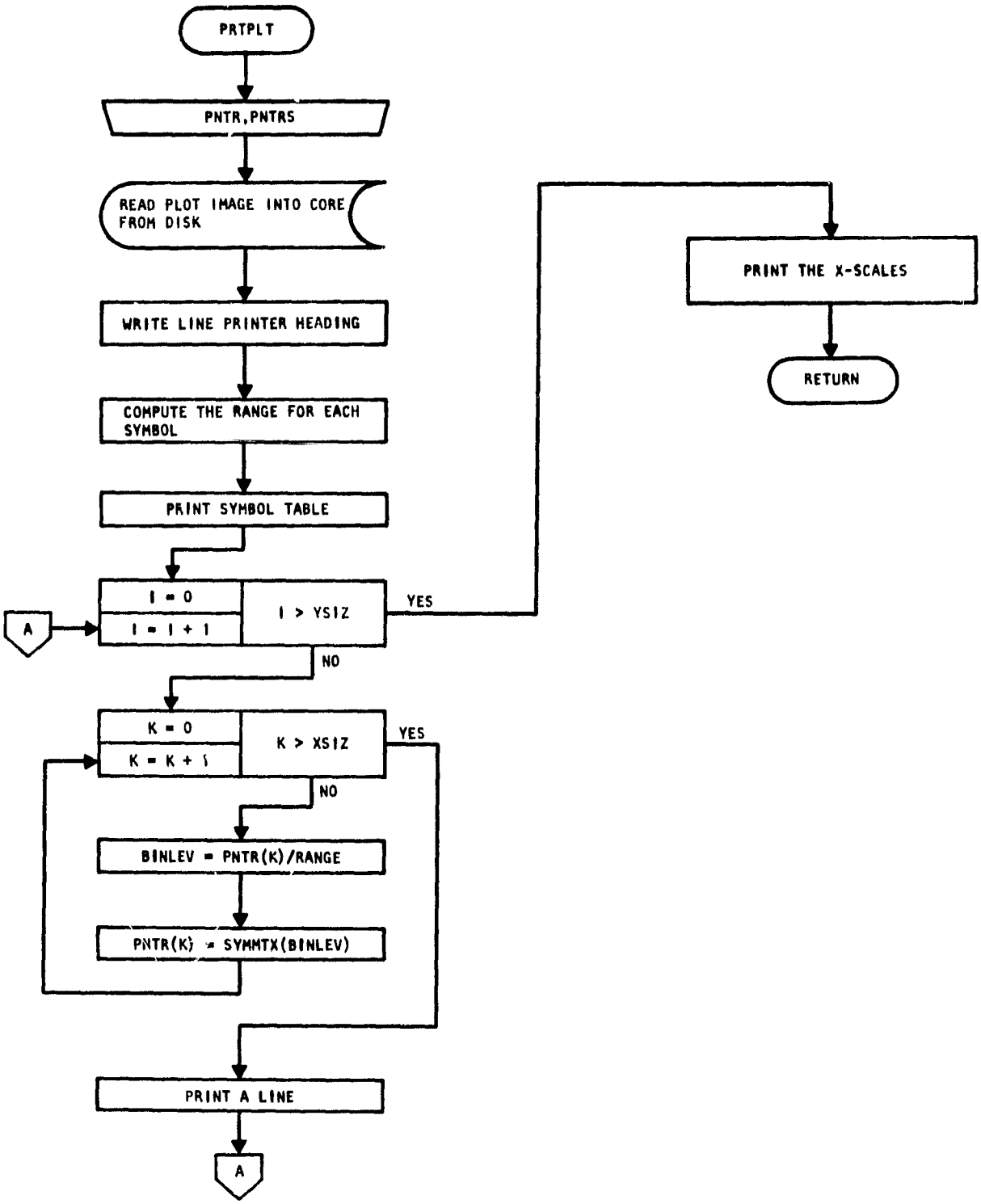
LINPLT entry 1:

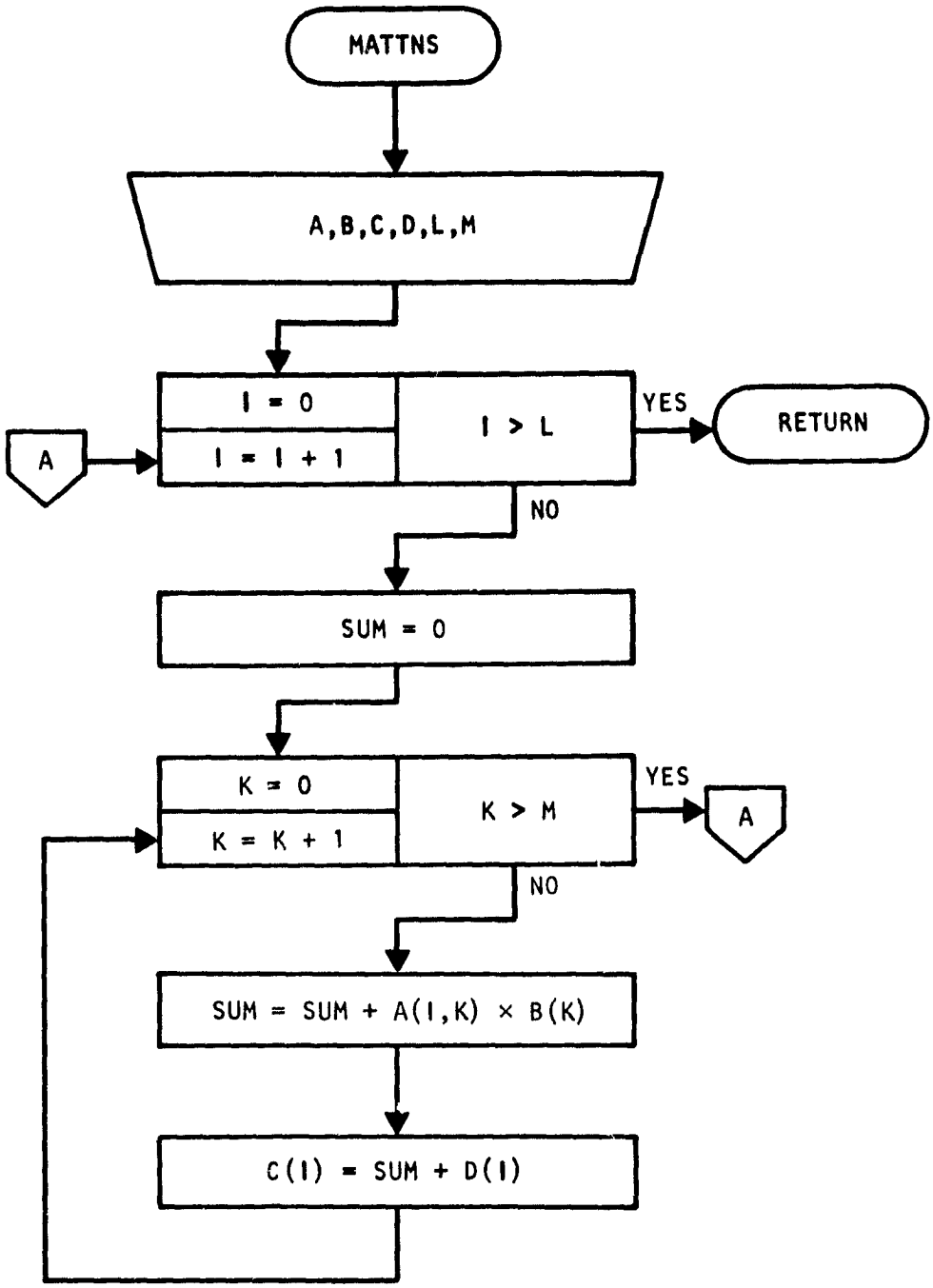


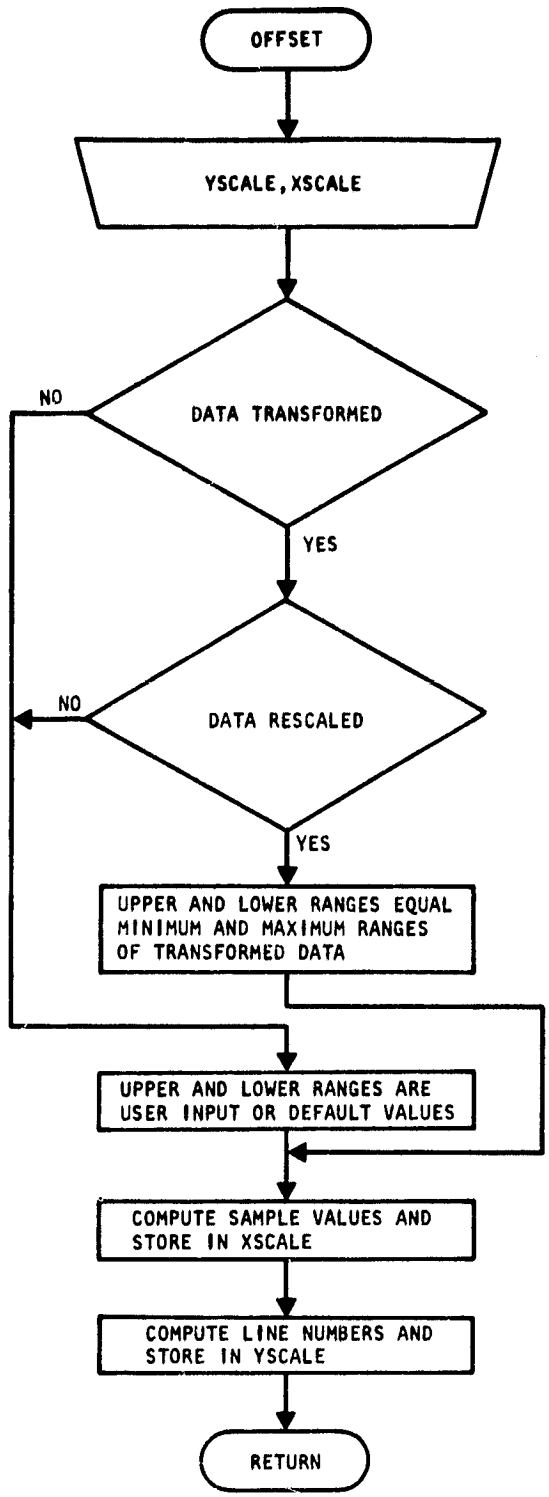
LINPLT entry 2:

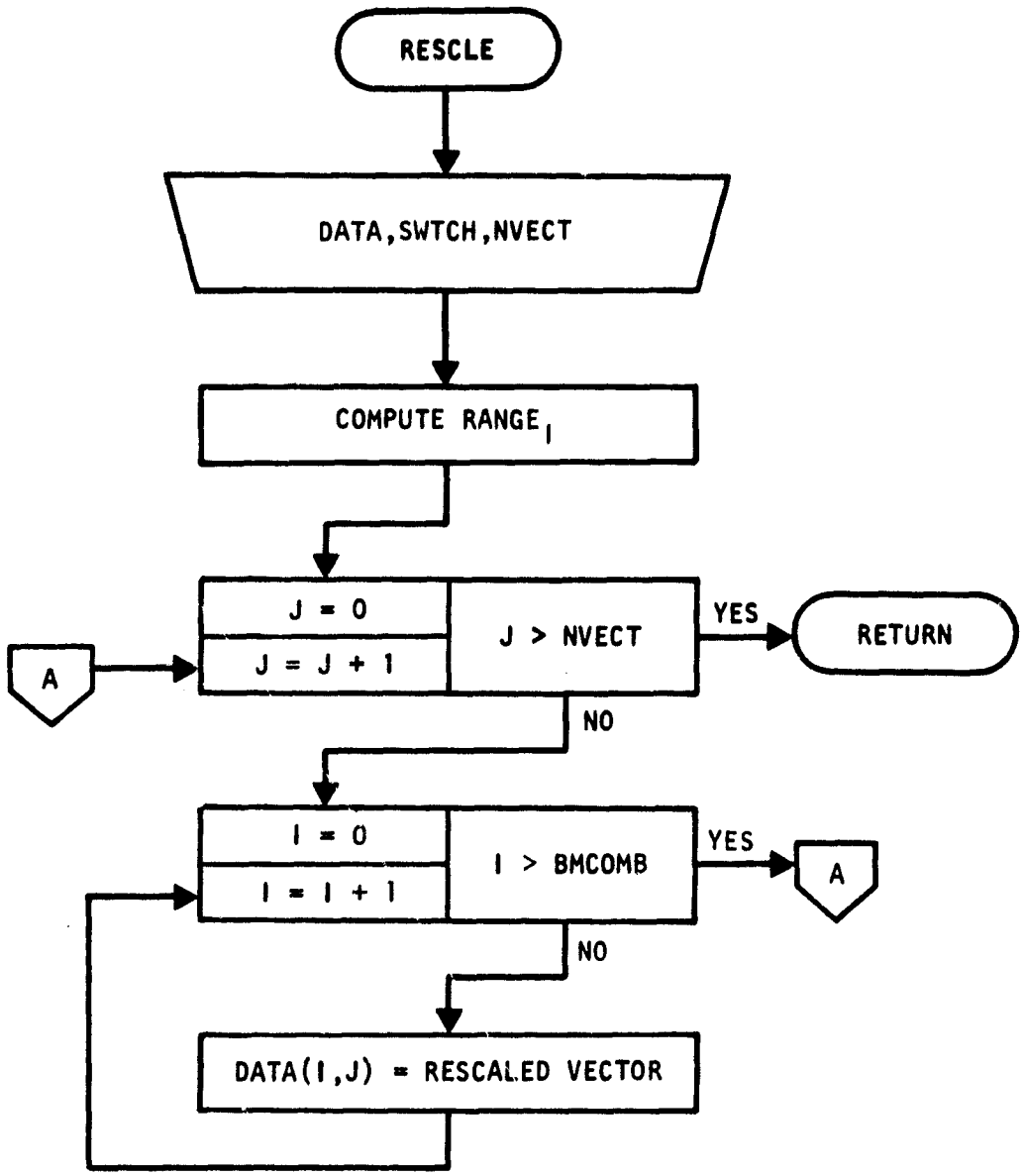


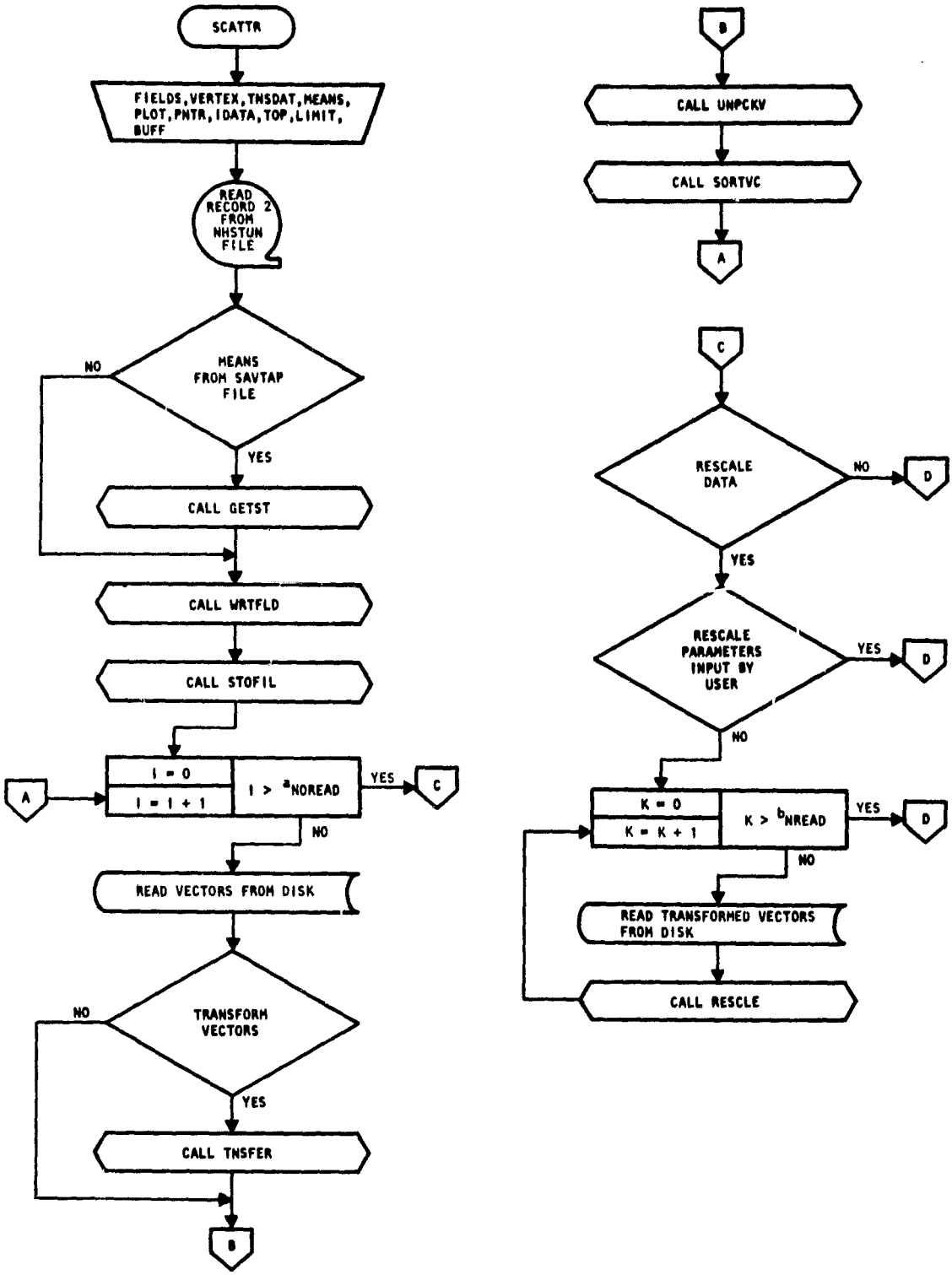
LINPLT entry 3:




MATTNS
A,B,C,D,L,M
I = 0
I = I + 1
I > L
YES
RETURN
A
NO
SUM = 0
K = 0
K = K + 1
K > M
YES
A
NO
SUM = SUM + A(I,K) × B(K)
C(I) = SUM + D(I)

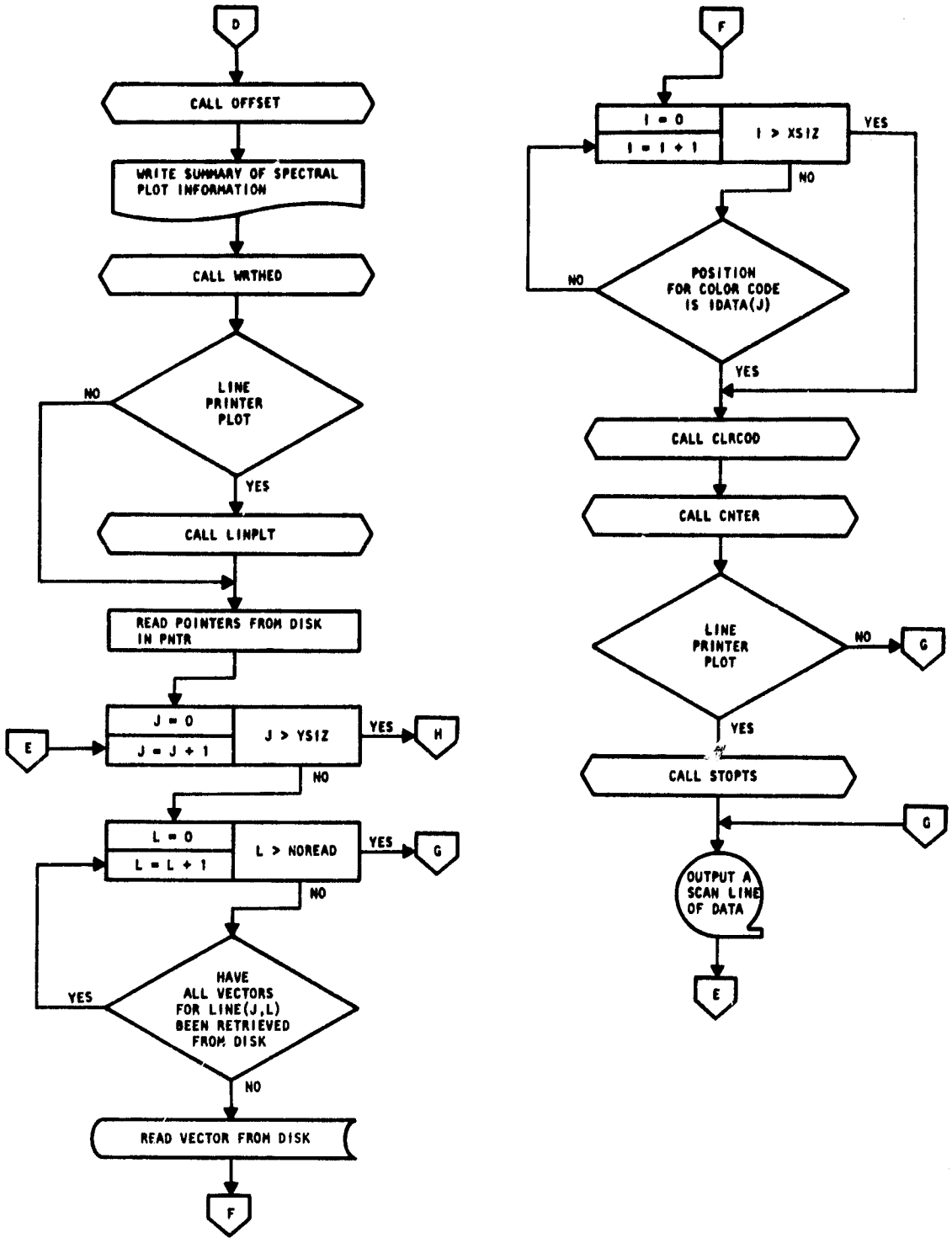

OFFSET
YSCALE, XSCALE
DATA TRANSFORMED
NO
YES
DATA RESCALED
NO
YES
UPPER AND LOWER RANGES EQUAL MINIMUM AND MAXIMUM RANGES OF TRANSFORMED DATA
UPPER AND LOWER RANGES ARE USER INPUT OR DEFAULT VALUES
COMPUTE SAMPLE VALUES AND STORE IN XSCALE
COMPUTE LINE NUMBERS AND STORE IN YSCALE
RETURN

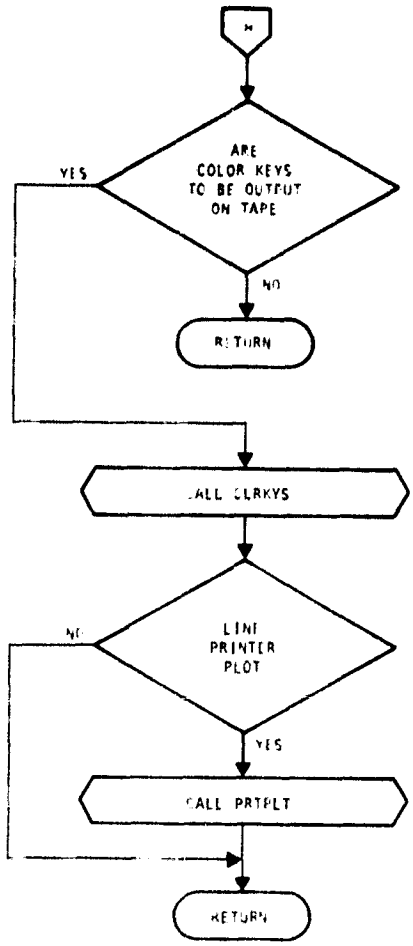

RESCLE
DATA,SWTCH,NVECT
COMPUTE RANGE$_I$
J = 0
J = J + 1
J > NVECT
YES
RETURN
A
NO
I = 0
I = I + 1
I > BMCOMB
YES
A
NO
DATA(I,J) = RESCALED VECTOR

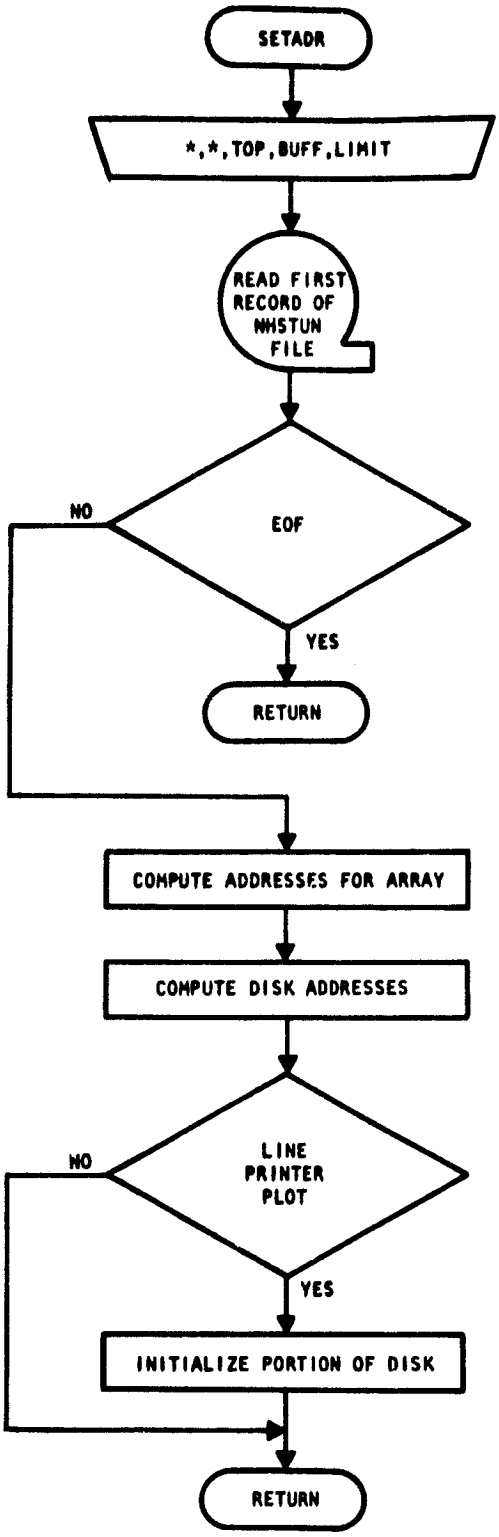


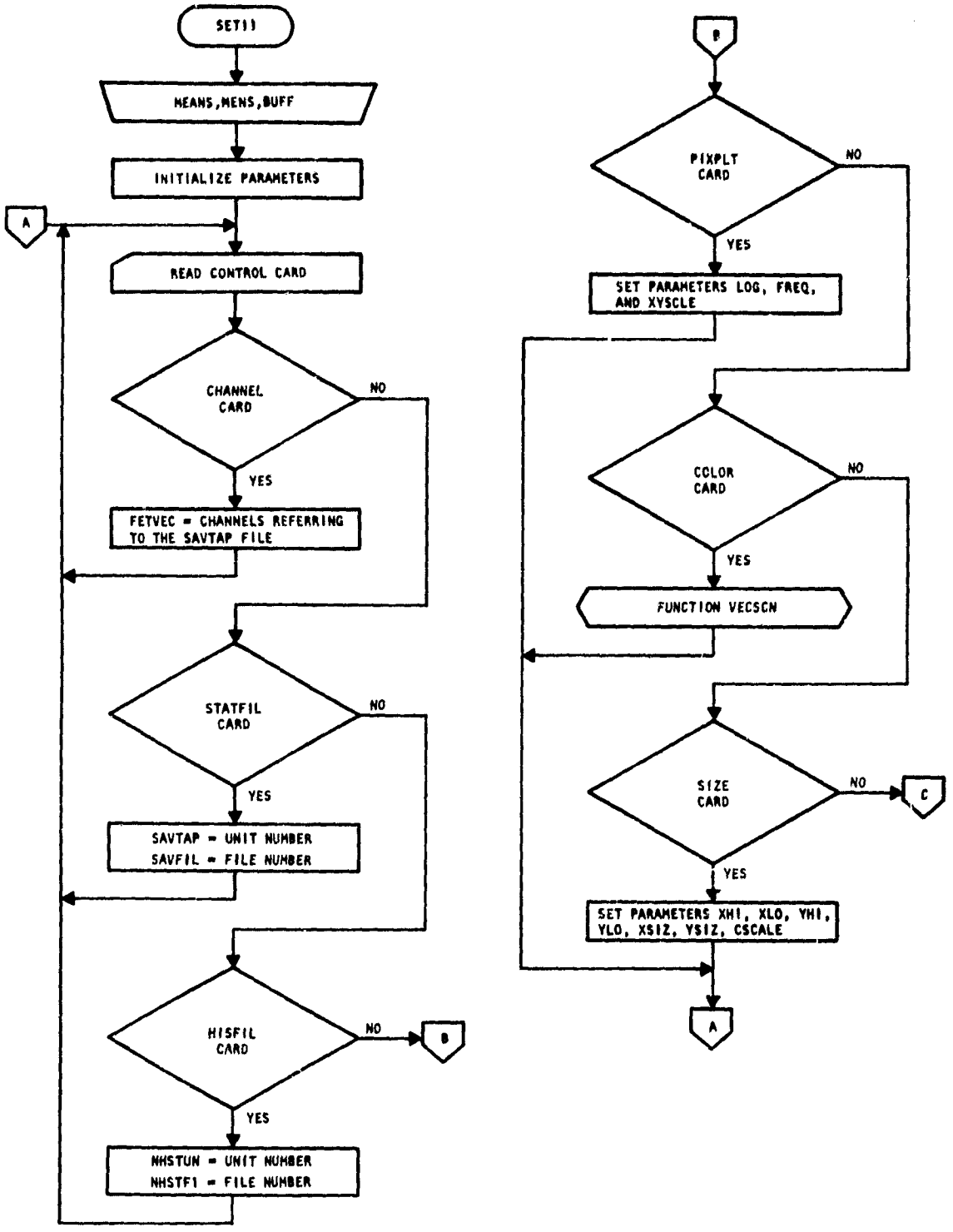
[a]NOREAD = number of reads from disk to retrieve vectors.

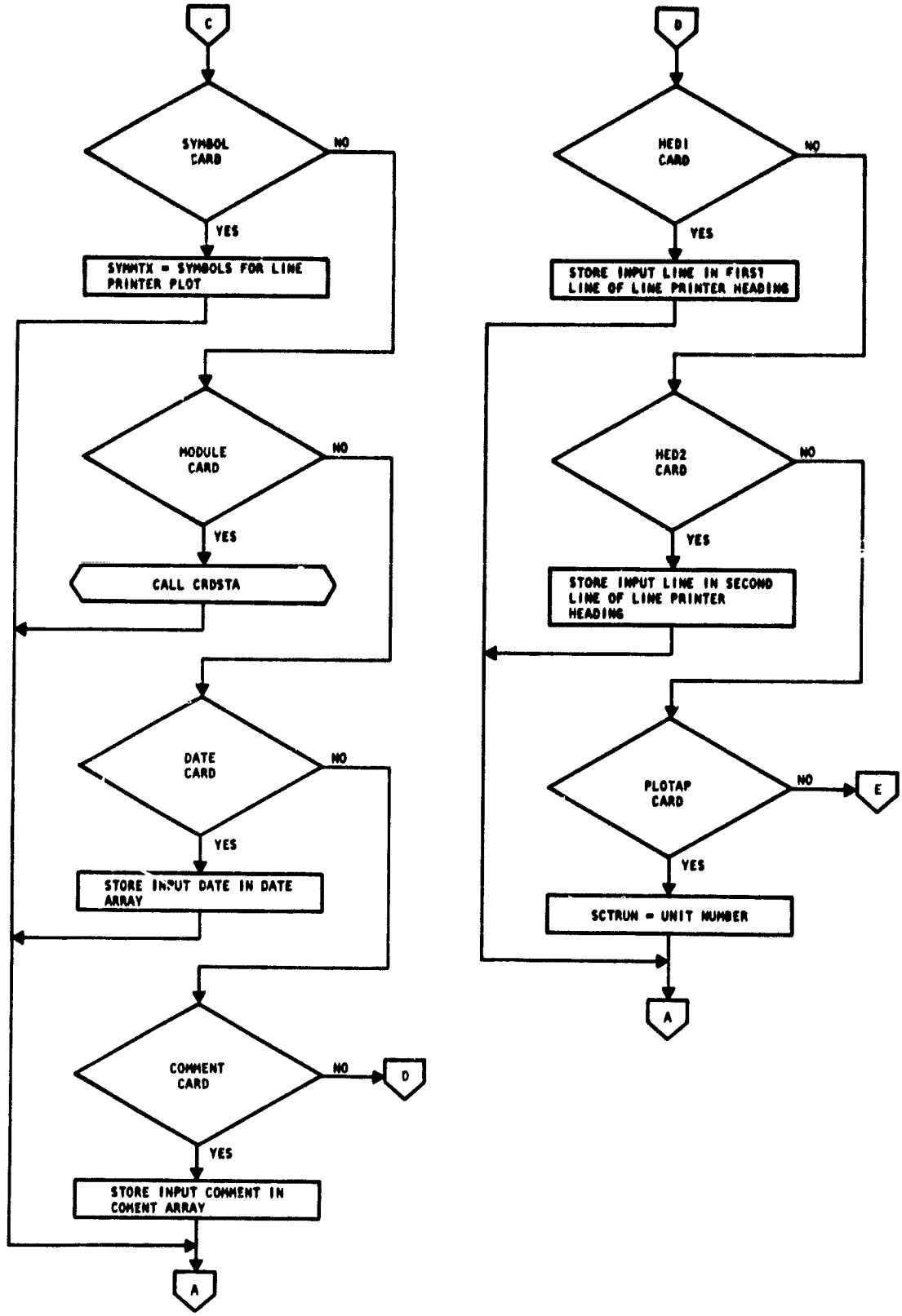
[b]NREAD = number of reads from disk to retrieve transformed vectors.

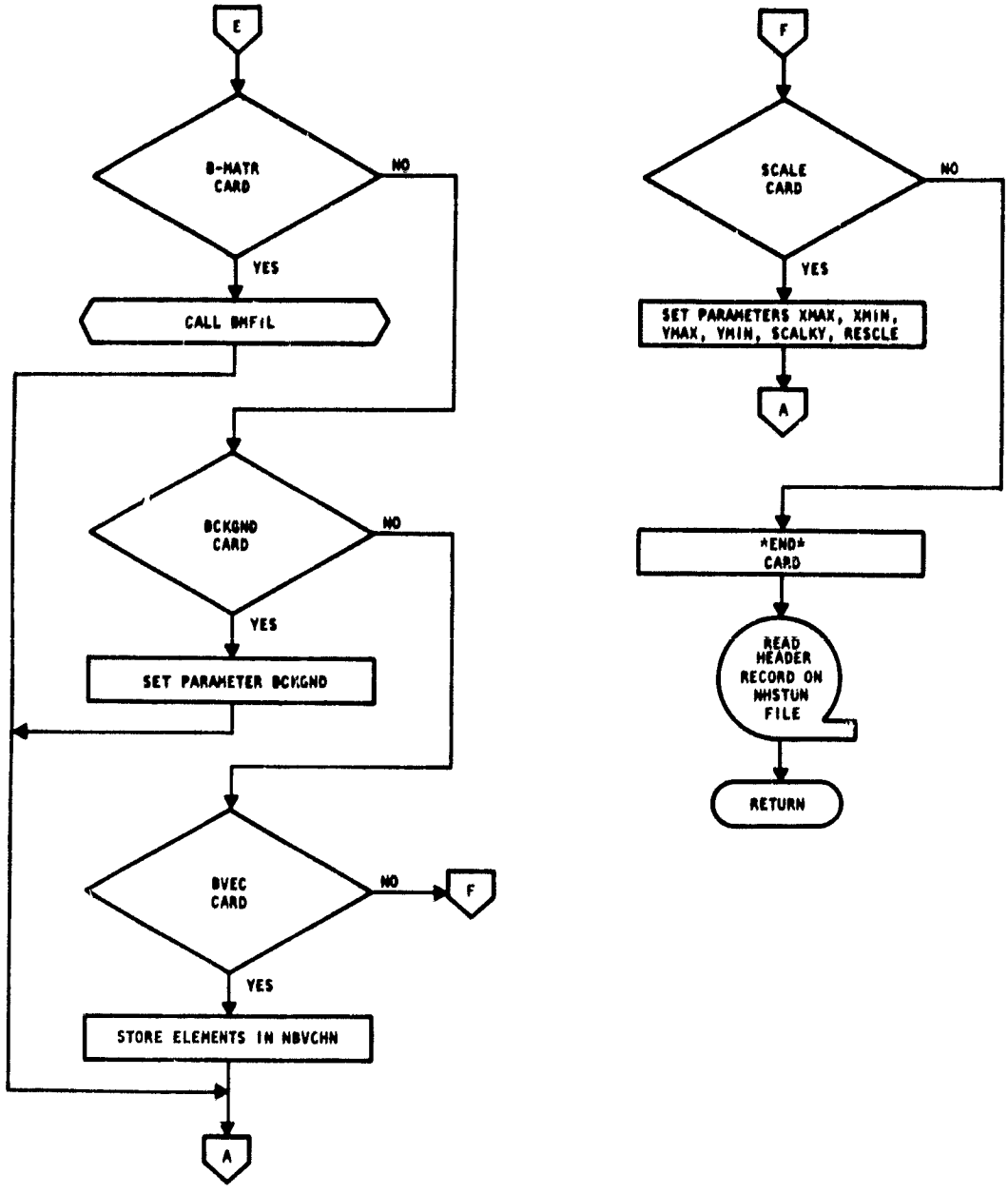

D
CALL OFFSET
WRITE SUMMARY OF SPECTRAL PLOT INFORMATION
CALL WRTHED
LINE PRINTER PLOT
NO
YES
CALL LINPLT
READ POINTERS FROM DISK IN PNTR
E
J = 0
J = J + 1
J > YSIZ
YES
H
NO
L = 0
L = L + 1
L > NOREAD
YES
G
NO
HAVE ALL VECTORS FOR LINE(J,L) BEEN RETRIEVED FROM DISK
YES
NO
READ VECTOR FROM DISK
F
F
I = 0
I = I + 1
I > XSIZ
YES
NO
POSITION FOR COLOR CODE IS IDATA(J)
NO
YES
CALL CLRCOD
CALL CNTER
LINE PRINTER PLOT
NO
G
YES
CALL STOPTS
G
OUTPUT A SCAN LINE OF DATA
E

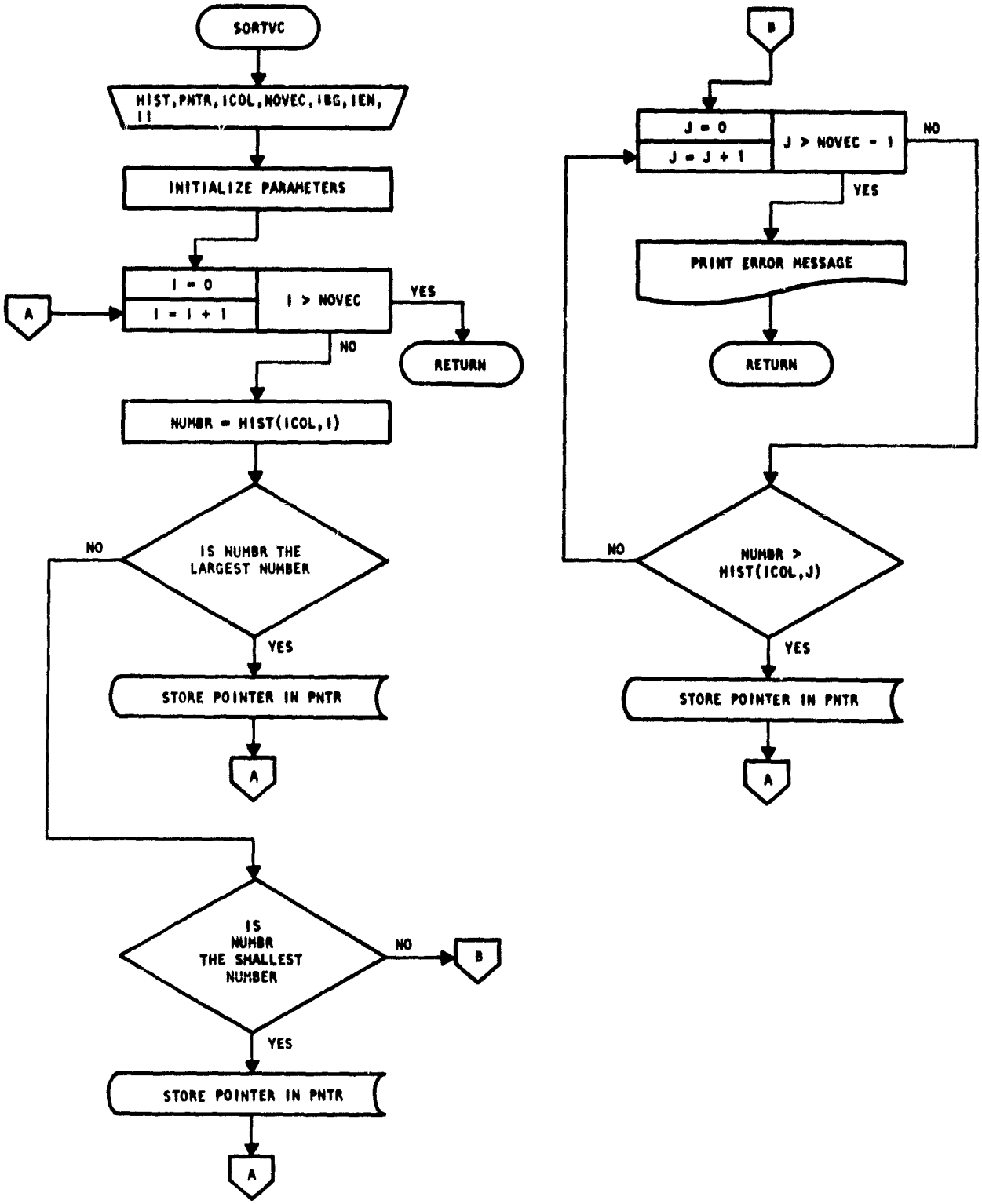

ARE
COLOR KEYS
TO BE OUTPUT
ON TAPE
YES
NO
RETURN
CALL CLRKYS
LINE
PRINTER
PLOT
NO
YES
CALL PRTPLT
RETURN

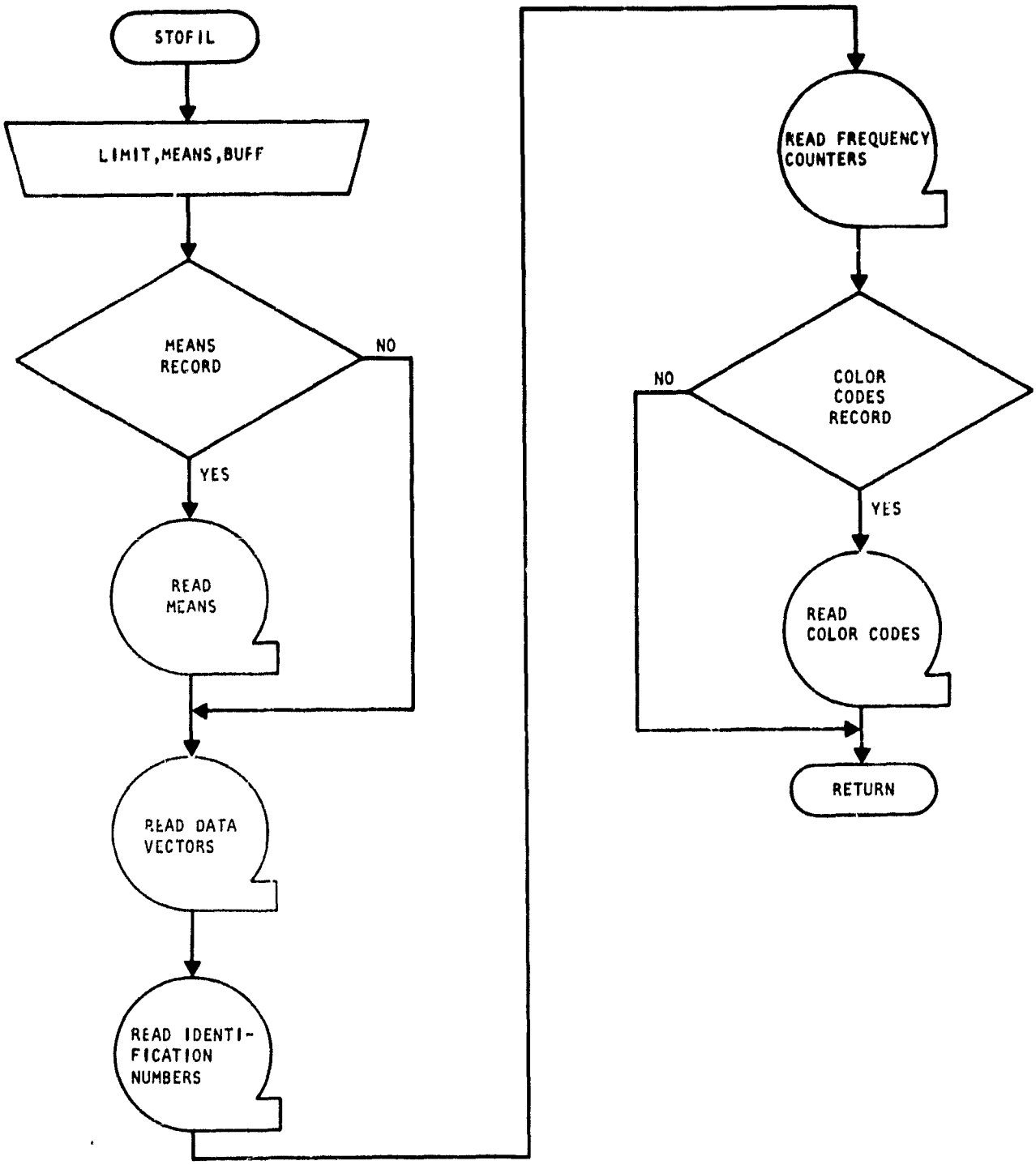

SETADR
,,TOP,BUFF,LIMIT
READ FIRST RECORD OF NHSTUN FILE
EOF
NO
YES
RETURN
COMPUTE ADDRESSES FOR ARRAY
COMPUTE DISK ADDRESSES
LINE PRINTER PLOT
NO
YES
INITIALIZE PORTION OF DISK
RETURN

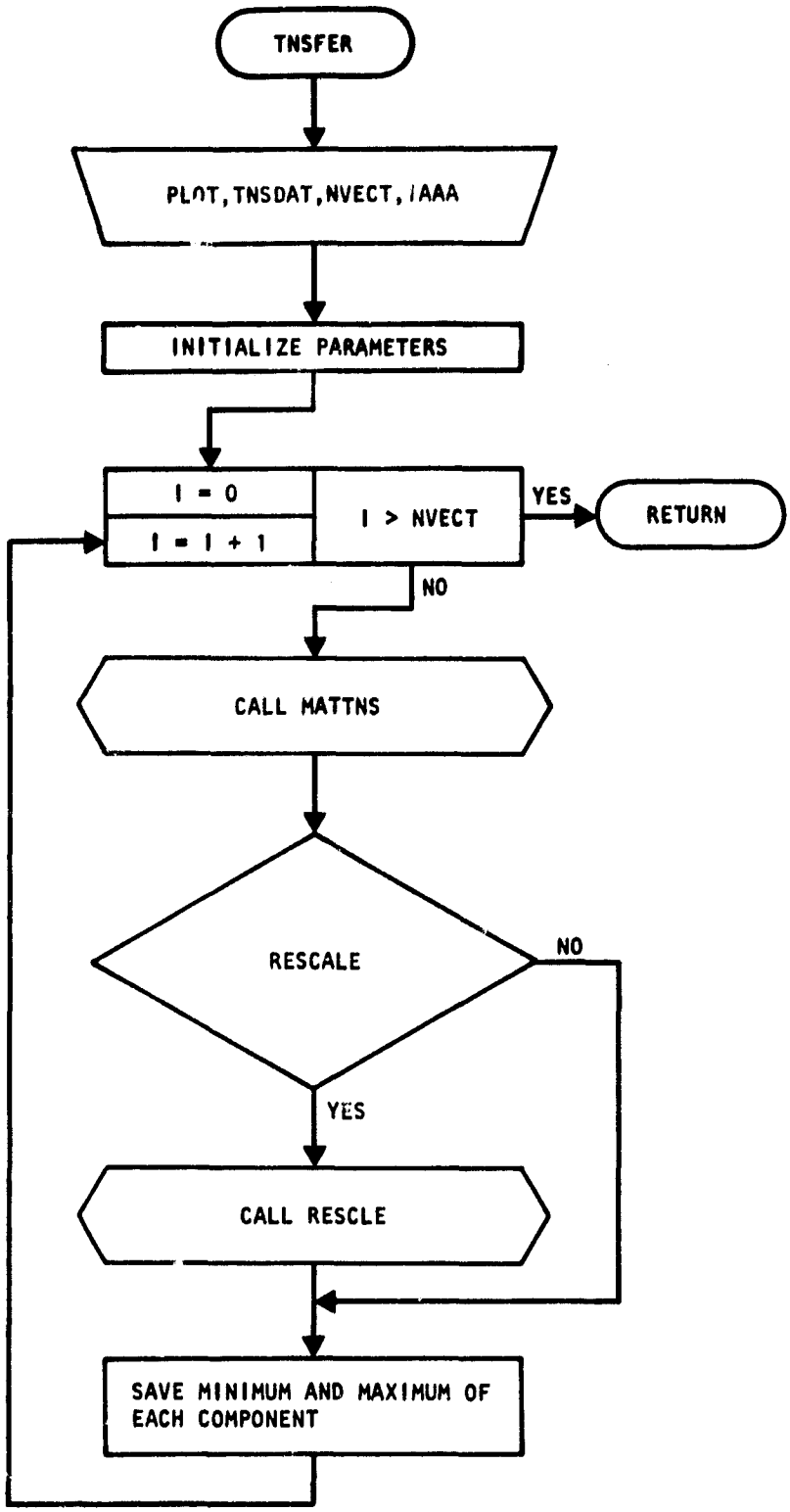

SETII
MEANS,MENS,BUFF
INITIALIZE PARAMETERS
A
READ CONTROL CARD
CHANNEL CARD
NO
YES
FETVEC = CHANNELS REFERRING TO THE SAVTAP FILE
STATFIL CARD
NO
YES
SAVTAP = UNIT NUMBER
SAVFIL = FILE NUMBER
HISFIL CARD
NO
B
YES
NHSTUN = UNIT NUMBER
NHSTF1 = FILE NUMBER
B
PIXPLT CARD
NO
YES
SET PARAMETERS LOG, FREQ, AND XYSCLE
COLOR CARD
NO
YES
FUNCTION VECSCN
SIZE CARD
NO
C
YES
SET PARAMETERS XH1, XLO, YH1, YLO, XSIZ, YSIZ, CSCALE
A

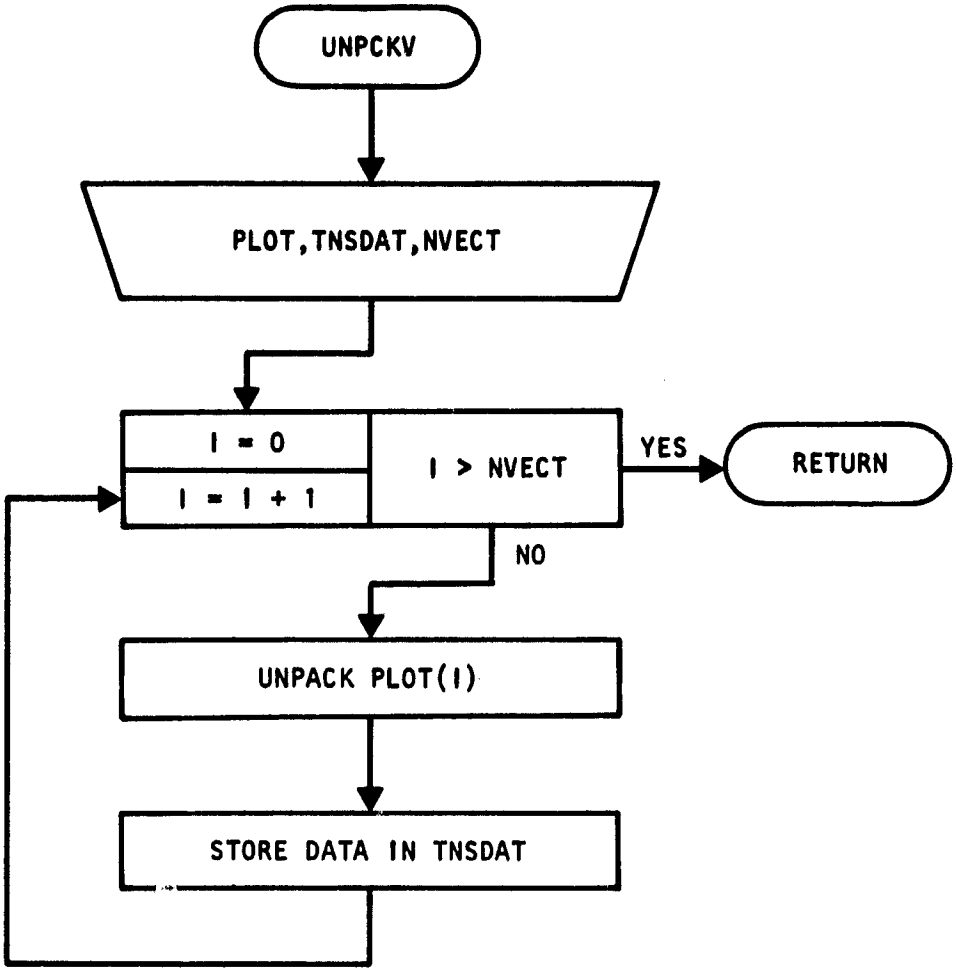

C
SYMBOL CARD
NO
YES
SYMTX = SYMBOLS FOR LINE PRINTER PLOT
MODULE CARD
NO
YES
CALL CRDSTA
DATE CARD
NO
YES
STORE INPUT DATE IN DATE ARRAY
COMMENT CARD
NO
D
YES
STORE INPUT COMMENT IN COMENT ARRAY
A
D
HED1 CARD
NO
YES
STORE INPUT LINE IN FIRST LINE OF LINE PRINTER HEADING
HED2 CARD
NO
YES
STORE INPUT LINE IN SECOND LINE OF LINE PRINTER HEADING
PLOTAP CARD
NO
E
YES
SCTRUN = UNIT NUMBER
A

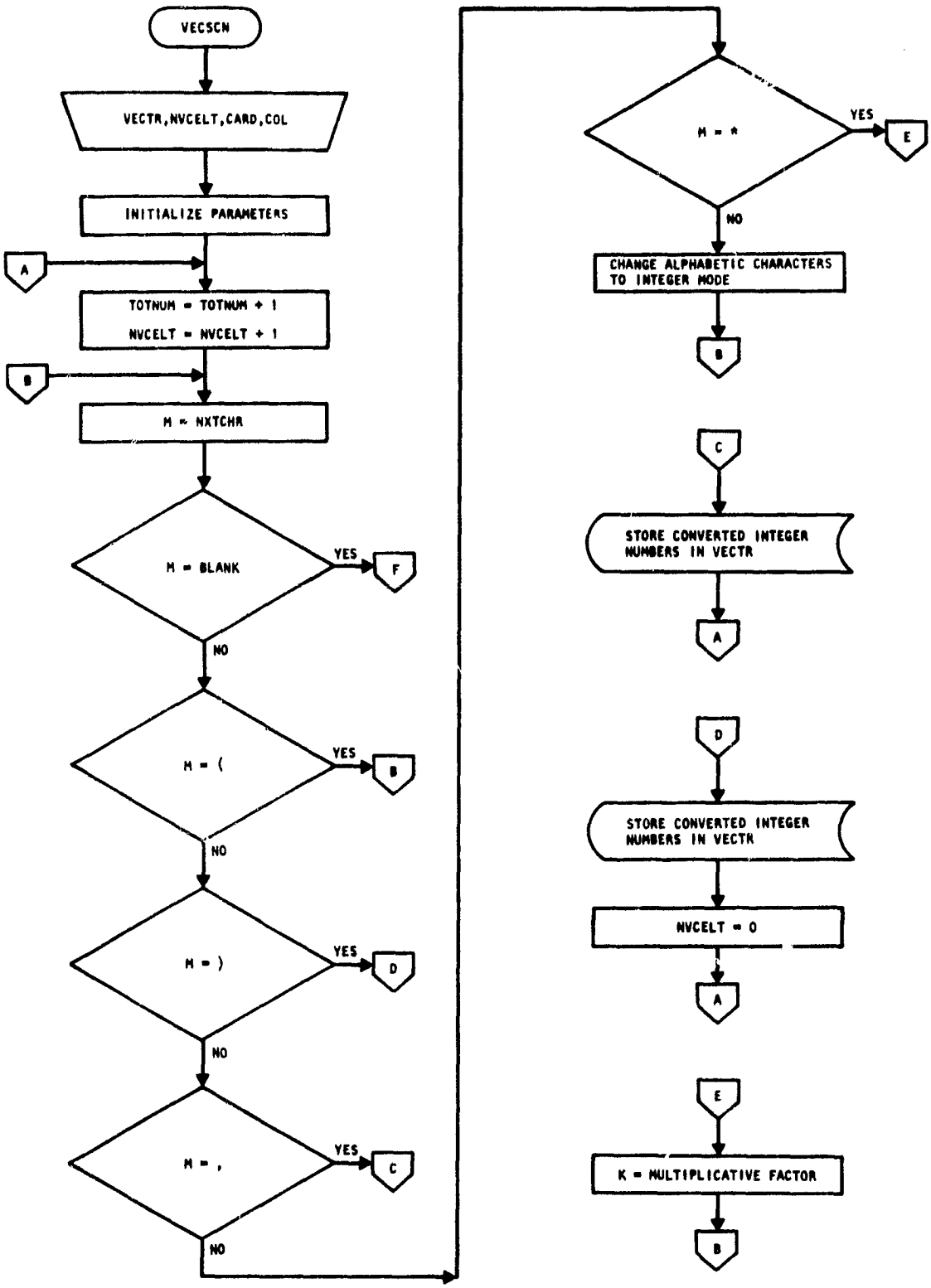

E
B-MATR CARD
NO
YES
CALL BMFIL
BCKGND CARD
NO
YES
SET PARAMETER BCKGND
BVEC CARD
NO
F
YES
STORE ELEMENTS IN NBVCHN
A
F
SCALE CARD
NO
YES
SET PARAMETERS XMAX, XMIN, YMAX, YMIN, SCALKY, RESCLE
A
END CARD
READ HEADER RECORD ON NHSTUN FILE
RETURN

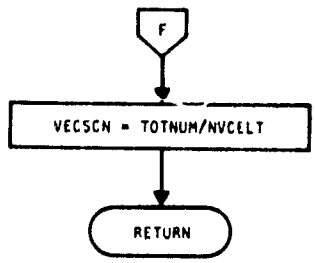

SORTVC
HIST,PNTR,ICOL,NOVEC,IBG,IEN,
II
INITIALIZE PARAMETERS
I = 0
I = I + 1
I > NOVEC
YES
NO
RETURN
A
NUMBR = HIST(ICOL,I)
IS NUMBR THE LARGEST NUMBER
NO
YES
STORE POINTER IN PNTR
A
IS NUMBR THE SMALLEST NUMBER
NO
B
YES
STORE POINTER IN PNTR
A
B
J = 0
J = J + 1
J > NOVEC - 1
NO
YES
PRINT ERROR MESSAGE
RETURN
NUMBR > HIST(ICOL,J)
NO
YES
STORE POINTER IN PNTR
A


STOFIL
LIMIT,MEANS,BUFF
MEANS RECORD
NO
YES
READ MEANS
READ DATA VECTORS
READ IDENTI-FICATION NUMBERS
READ FREQUENCY COUNTERS
NO
COLOR CODES RECORD
YES
READ COLOR CODES
RETURN


TNSFER
PLOT, TNSDAT, NVECT, IAAA
INITIALIZE PARAMETERS
I = 0
I = I + 1
I > NVECT
YES
RETURN
NO
CALL MATTNS
RESCALE
NO
YES
CALL RESCLE
SAVE MINIMUM AND MAXIMUM OF EACH COMPONENT


UNPCKV
PLOT,TNSDAT,NVECT
I = 0
I = I + 1
I > NVECT
YES
RETURN
NO
UNPACK PLOT(I)
STORE DATA IN TNSDAT


VECSCN
VECTR,NVCELT,CARD,COL
INITIALIZE PARAMETERS
A
TOTNUM = TOTNUM + 1
NVCELT = NVCELT + 1
B
M = NXTCHR
M = BLANK
YES
F
NO
M = (
YES
B
NO
M =)
YES
D
NO
M = ,
YES
C
NO
M = *
YES
E
NO
CHANGE ALPHABETIC CHARACTERS TO INTEGER MODE
B
C
STORE CONVERTED INTEGER NUMBERS IN VECTR
A
D
STORE CONVERTED INTEGER NUMBERS IN VECTR
NVCELT = 0
A
E
K = MULTIPLICATIVE FACTOR
B



17. DOTDATA PROCESSOR SUBPROGRAMS

The DOTDATA processor accepts field definition card images and an MSS DATAPE as input and creates a dot data file on the DOTUNT, which contains spatial and spectral information on the dots or pixels selected by the user. DOTDATA calls 5 subprograms within the processor and 20 utility subprograms (documented in section 19). Figure 17-1 is a linkage diagram of the DOTDATA processor.

DOTDATA PROCESSOR

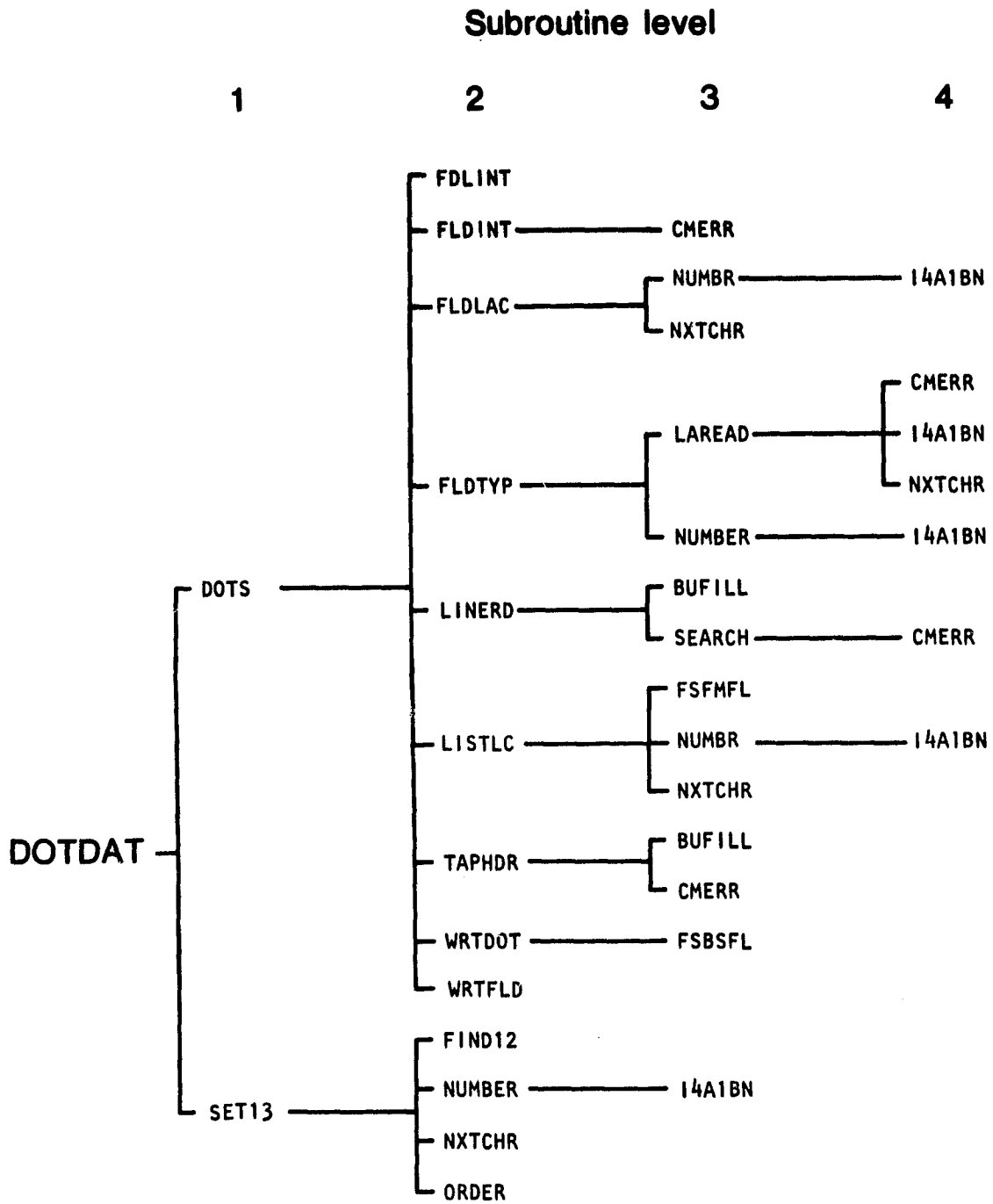


Figure 17-1. - Linkage diagram for the DOTDATA processor.

17.1 DOTDAT

The DOTDAT subprogram is the driver routine for the DOTDATA processor.

17.1.1 LINKAGES

The DOTDAT subprogram calls the DOTS and SET13 subprograms. It is called by MONTOR.

17.1.2 INTERFACES

The DOTDAT subprogram interfaces with other routines through the calling arguments.

17.1.3 INPUTS

Calling sequence: CALL DOTDAT(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.

17.1.4 OUTPUTS

Not applicable.

17.1.5 STORAGE REQUIREMENTS

This subprogram requires 384 bytes of storage.

17.1.6 DESCRIPTION

The subprogram DOTDAT calls SET13 to read and analyze the control cards. DOTS is called to coordinate the functions required to output the DOTUNT file.

17.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 17.6.

17.1.8 LISTING

The subprogram listing is provided in volume IV, section 17.

17.2 DOTS

The DOTS subprogram is the coordinator and supervisor of the functions required to output the DOTUNT file. This file is an interface to the ISOCLS, LABEL, and DISPLAY processors.

17.2.1 LINKAGES

The DOTS subprogram calls the FDLINT, FLDINT, FLDLAC, FLDTYP, LINERD, LISTLC, TAPHDR, WRTDOT, and WRTFLD subprograms. It is called by the DOTDAT driver routine.

17.2.2 INTERFACES

The DOTS subprogram interfaces with other routines through common blocks DOTVEC, GLOBAL, INFORM, and ISOLNK and through the calling arguments.

17.2.3 INPUTS

Calling sequence: CALL DOTS(DATA,FIELDS,VERTEX,TOP)

Parameter	Dimension	In/out	Definition
DATA	SIZE,1	Out	Array for storing the dot data (SIZE = 4 + NOFET2).
FIELDS	4,1	Out	Array for storing the field information for each dot.
VERTEX	1	Out	Array for storing the field vertices for type 1 or 2 dots.
TOP	1	In	Maximum usable working storage in ARRAY; TOP = 10 600.

17.2.4 OUTPUTS

This subprogram outputs results on tape and on the printer.

17.2.5 STORAGE REQUIREMENTS

This subprogram requires 43 514 bytes of storage.

17.2.6 DESCRIPTION

The DOTS subprogram calls FLDTYP to read in a field definition card image for a type i field and the appropriate utility routines to read the MSS image data for the field. Information, such as line, sample, type, and category numbers and the dot grid intersection, is collected. A maximum of $\frac{n \leq 5000}{(\text{number of channels} \leq 250)}$ dots may be collected. When all fields of type i have been processed, a file is written. Subprogram WRTDOT is called to output the DOTUNT file, and subprogram WRTFLD is called to print a summary of training or test fields. If applicable, the Sun angles from the input imagery data are output on the DOTUNT file. The dot data record is output on the line printer.

17.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 17.6.

17.2.8 LISTING

The subprogram listing is provided in volume IV, section 17.

17.3 FLDLAC

The FLDLAC subprogram reads the DOTUNT file and field definition card images, computes line and sample increments, and stores dot information.

17.3.1 LINKAGES

The FLDLAC subprogram calls the NUMBR and NXTCHR subprograms. It is called by DOTS.

17.3.2 INTERFACES

The FLDLAC subprogram interfaces with other routines through common blocks DOTVEC and INFORM and through the calling arguments.

17.3.3 INPUTS

Input to the FLDLAC subprogram consists of the DOTUNT file output by the DOTDATA processor.

Calling sequence: CALL FLDLAC(FIELDS,STAMNT,*,*,*,IPT,VERTEX)

Parameter	Dimension	In/out	Definition
FIELDS	4,1	Out	Array containing category name and dot type for dot I stored in FIELD(1,I) and FIELD(4,I).
STAMNT	1	In/out	Switch to indicate dots being taken from currently read card; initially set = 1 but reset = 2 if the DOT card has been read and is being processed.
*	1	Out	Exit route 1 (after all processing).
*	1	Out	Exit route 2 (if SWCHG $\leq$ 1).
*	1	Out	Exit route 3 (if a card is improperly formatted).

Parameter	Dimension	In/out	Definition
IPT	1	In/out	Pointer in VERTEX (incremented by 4).
VERTEX	1	Out	Array containing dot vertices (sample and line numbers repeated).

The control cards relevant to this routine are given in section 17.5.3 of volume II of this user guide.

17.3.4 OUTPUTS

The results are returned for use by the calling routine.

17.3.5 STORAGE REQUIREMENTS

This subprogram requires 3026 bytes of storage.

17.3.6 DESCRIPTION

Not required.

17.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 17.6.

17.3.8 LISTING

The subprogram listing is provided in volume IV, section 17.

17.4 FLDTYP

The subprogram FLDTYP initiates the reading of the field definition data set and signals the calling routine DOTS when all the fields for a given type have been processed.

17.4.1 LINKAGES

The FLDTYP subprogram calls the LAREAD and NUMBER subprograms. It is called by DOTS.

17.4.2 INTERFACES

The FLDTYP subprogram interfaces with other routines through common blocks DOTVEC and INFORM and through the calling arguments.

17.4.3 INPUTS

Calling sequence: CALL FLDTYP(FIELDS,STAMNT,*,*,*,IPT,VERTEX)

Parameter	Dimension	In/out	Definition
(1) FIELDS	4,1	Out	Array for storing the field data.
(2) STAMNT	1	In	Parameter for a GO TO Fortran statement.
(3) *	1	In	Exit route if a field definition card is encountered.
(4) *	1	In	Exit route if a TYPE card is encountered.
(5) *	1	In	Exit route if a $END card is encountered.
(6) IPT	1	In	Pointer for FIELDS array.
(7) VERTEX	1	Out	Array for storing the field vertices.

The field definition card image format relevant to this routine is given in section 3.2.3 of volume II of this user guide.

17.4.4 OUTPUTS

This subprogram stores the type and category names in common block DOTVEC.

17.4.5 STORAGE REQUIREMENTS

This subprogram requires 1318 bytes of storage.

17.4.6 DESCRIPTION

The subprogram FLDTYP signals the calling routine when a TYPE, field definition, or $END card image is encountered by exiting through calling arguments. If a TYPE card image is encountered, the subprogram exits by calling argument (4); if a field definition card image is encountered, the subprogram exits by calling argument (3); if a $END card is encountered, the subprogram exits by calling argument (5).

17.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 17.6.

17.4.8 LISTING

The subprogram listing is provided in volume IV, section 17.

17.5 SET13

Subprogram SET13 reads and analyzes the control card images for the DOTDATA processor.

17.5.1 LINKAGES

The SET13 subprogram calls the FIND12, NUMBER, NXTCHR, and ORDER subprograms. It is called by the DOTDAT driver routine.

17.5.2 INTERFACES

The SET13 subprogram interfaces with other routines through common blocks DOTVEC, GLOBAL, and INFORM.

17.5.3 INPUTS

Calling sequence: CALL SET13

The control cards relevant to this routine are given in section 17 (table 17-1) of volume II of this user guide.

17.5.4 OUTPUTS

This subprogram outputs a summary of user-requested options and, optionally, a list of selected channels and input data vectors.

17.5.5 STORAGE REQUIREMENTS

This subprogram requires 3588 bytes of storage.

17.5.6 DESCRIPTION

Each control card image is read and analyzed for format errors. If the control card is valid, the appropriate parameters and switches are set. When applicable, if a control card image for a specific option is not input, a default value is supplied.

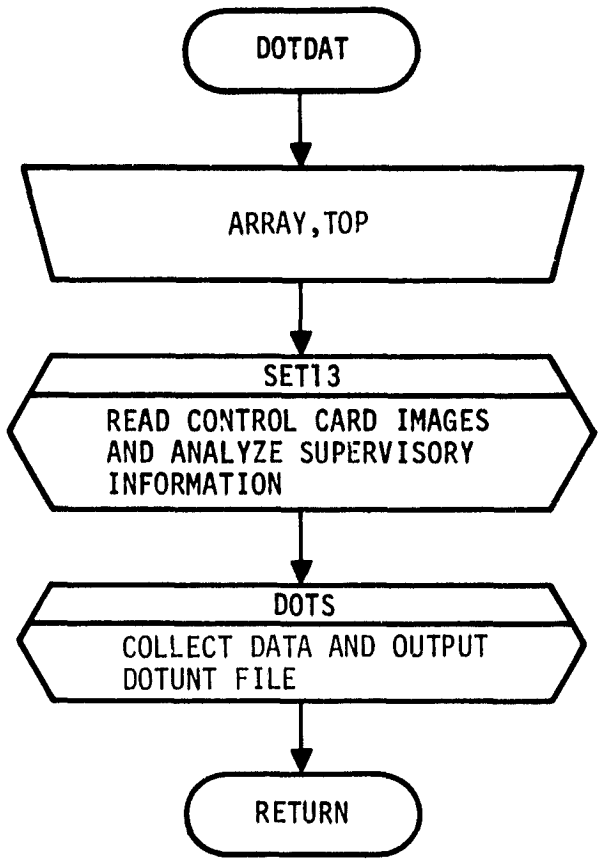
17.5.7 FLOW CHART

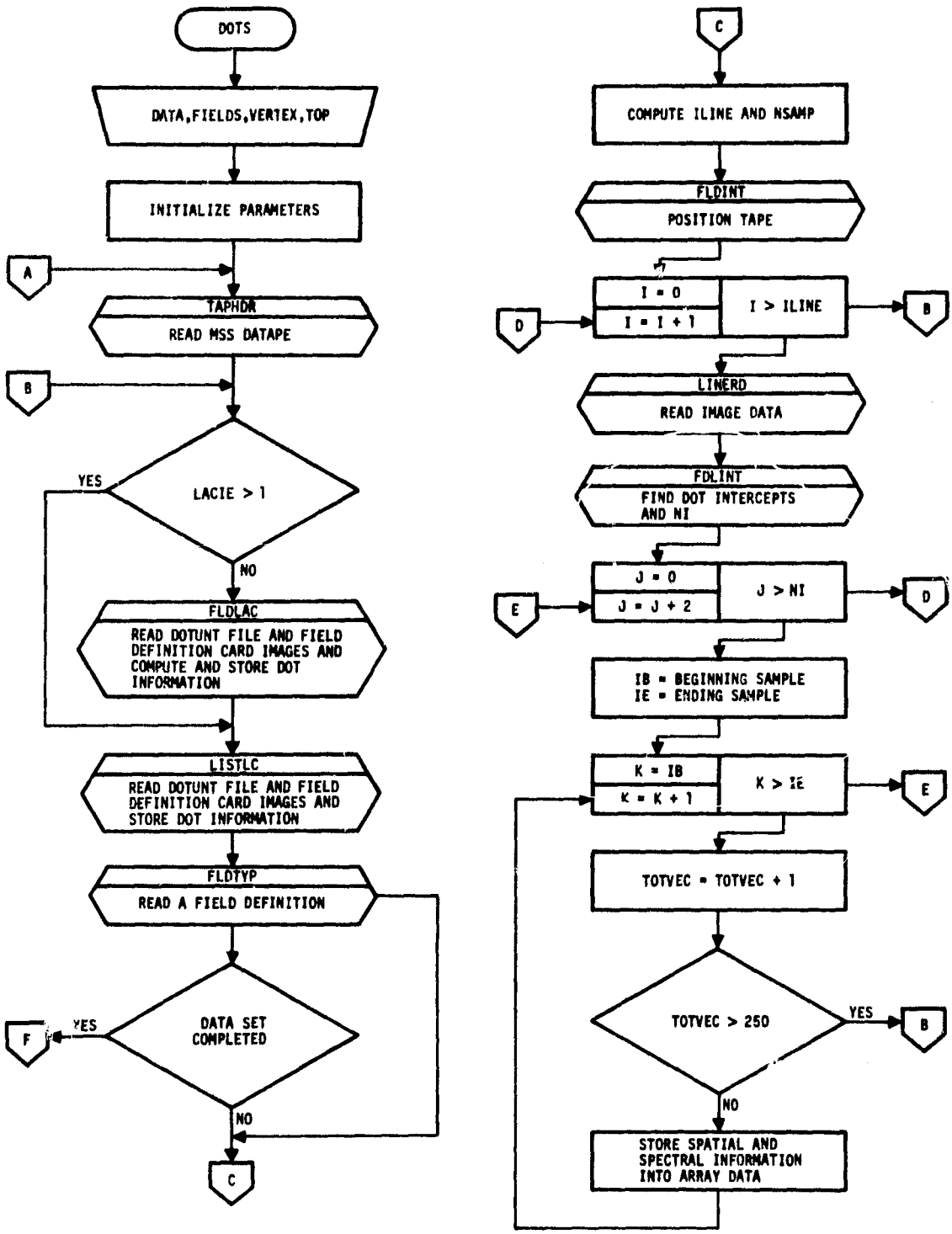
The available subprogram flow charts for this processor are provided in section 17.6.

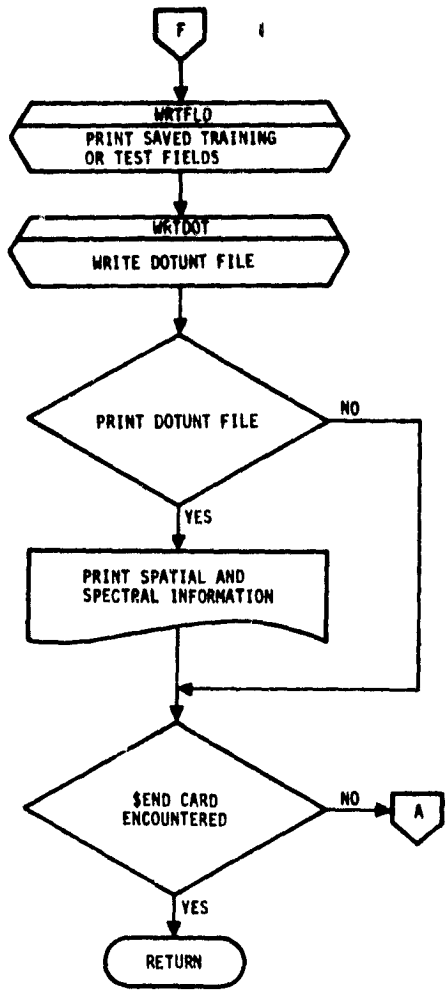
17.5.8 LISTING

The subprogram listing is provided in volume IV, section 17.

17.6 SUBPROGRAM FLOW CHARTS




DOTS
DATA,FIELDS,VERTEX,TOP
INITIALIZE PARAMETERS
A
TAPHDR
READ MSS DATAPE
B
LACIE > 1
YES
NO
FLDLAC
READ DOTUNT FILE AND FIELD DEFINITION CARD IMAGES AND COMPUTE AND STORE DOT INFORMATION
LISTLC
READ DOTUNT FILE AND FIELD DEFINITION CARD IMAGES AND STORE DOT INFORMATION
FLDTYP
READ A FIELD DEFINITION
DATA SET COMPLETED
YES
F
NO
C
C
COMPUTE ILINE AND NSAMP
FLDINT
POSITION TAPE
I = 0
I = I + 1
I > ILINE
D
B
LINERD
READ IMAGE DATA
FDLINT
FIND DOT INTERCEPTS AND NI
J = 0
J = J + 2
J > NI
E
D
IB = BEGINNING SAMPLE
IE = ENDING SAMPLE
K = IB
K = K + 1
K > IE
E
TOTVEC = TOTVEC + 1
TOTVEC > 250
YES
B
NO
STORE SPATIAL AND SPECTRAL INFORMATION INTO ARRAY DATA


F
WRTFLD
PRINT SAVED TRAINING
OR TEST FIELDS
WRTDOT
WRITE DOTUNT FILE
PRINT DOTUNT FILE
NO
YES
PRINT SPATIAL AND
SPECTRAL INFORMATION
$END CARD
ENCOUNTERED
NO
A
YES
RETURN

18. LABEL PROCESSOR SUBPROGRAMS

The LABEL processor aids the analyst in supervising the labeling of statistics obtained from the ISOCLS processor. The analyst may use one of two procedures for labeling: k-nearest-neighbor or all-of-a-kind. LABEL may also be used to relabel the previously processed DOTUNT or SAVTAP file. Optional output consists of a MAPUNT or MAPTAP file or line printer output of the relabeled DOTUNT or SAVTAP file. LABEL uses 19 processor subprograms and 36 utility subprograms (documented in section 19). Figure 18-1 is a linkage diagram of the LABEL processor.

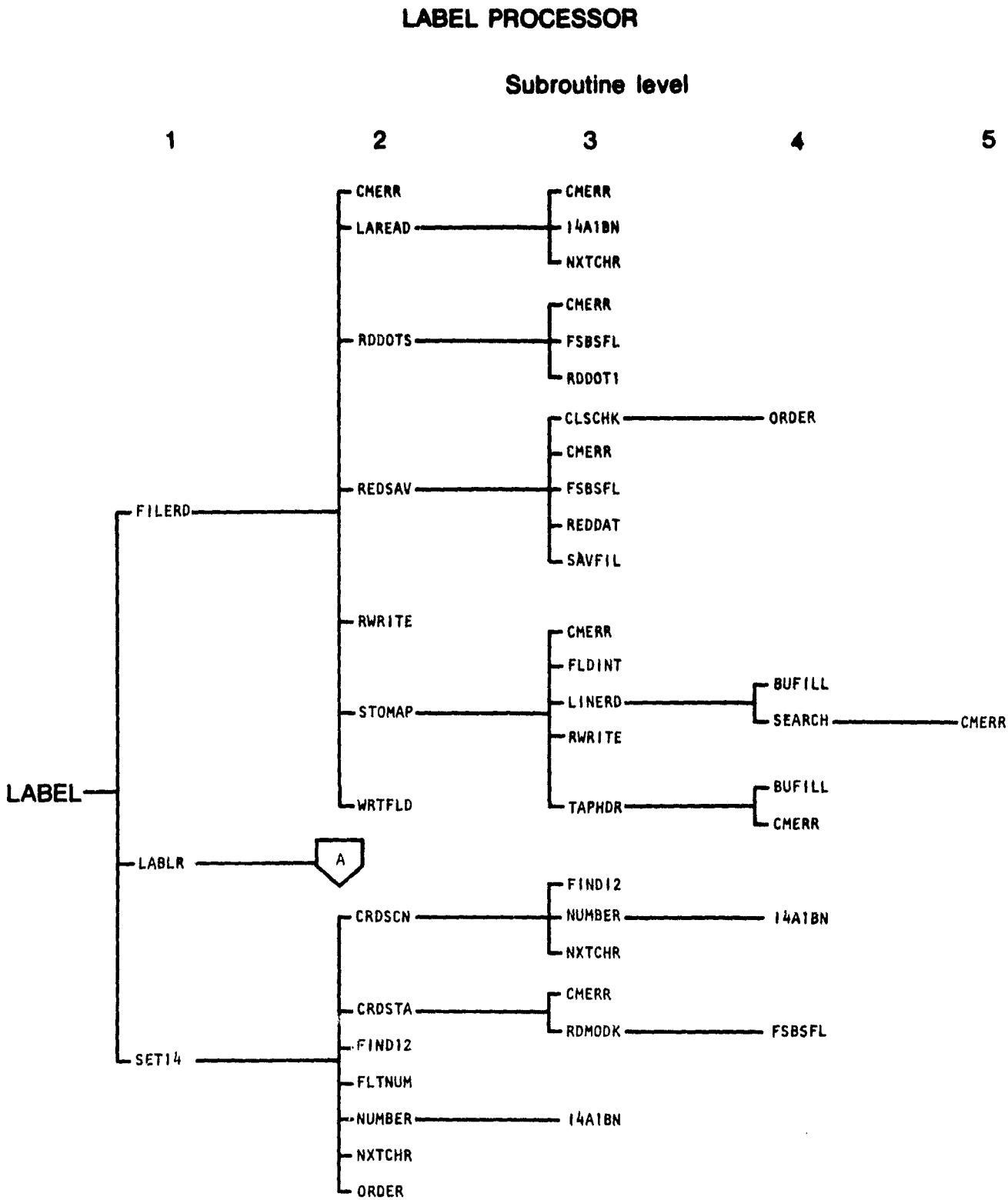
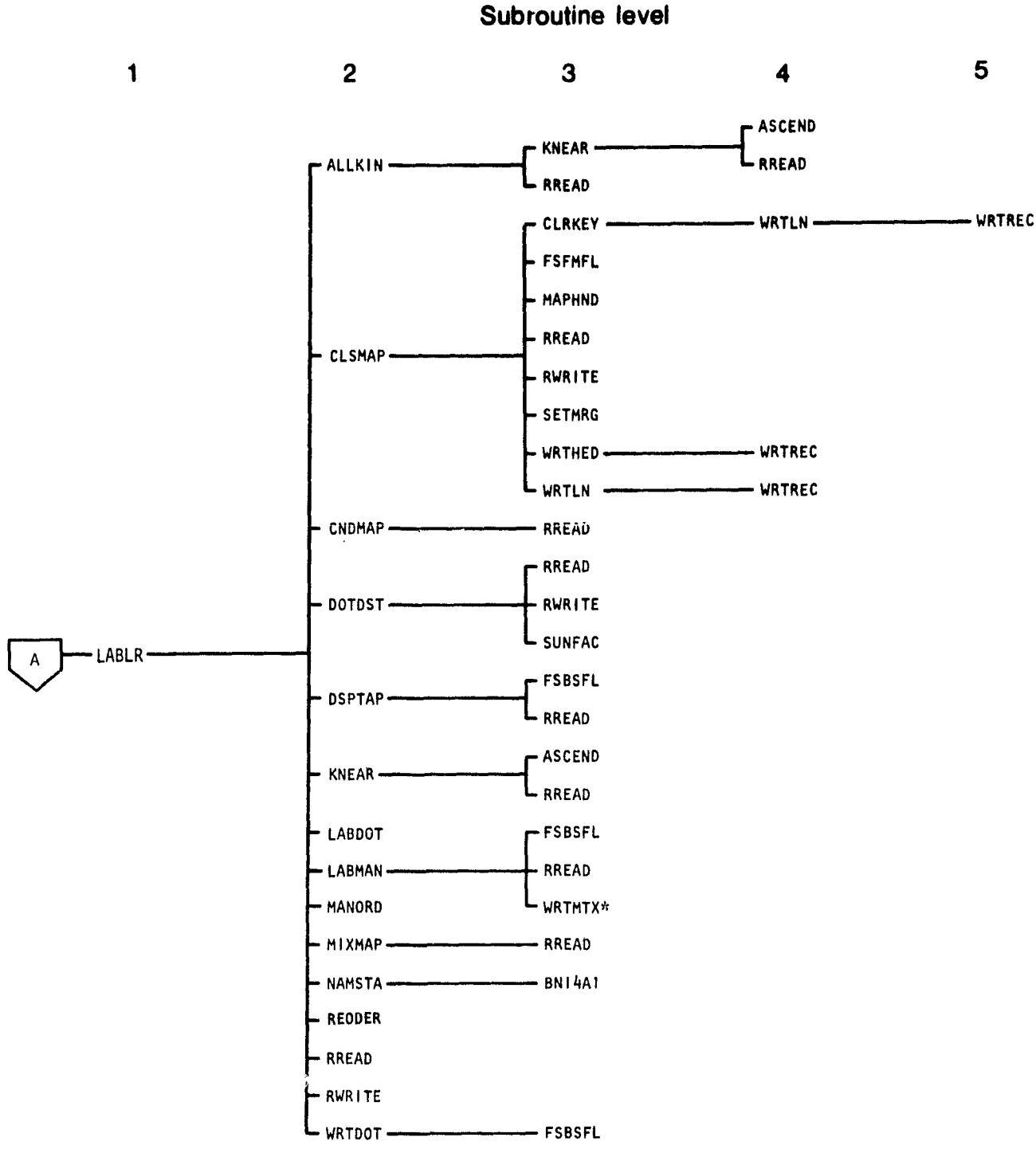


Figure 18-1.— Linkage diagram for the LABEL processor.



*Entry point is DWRTMX.

Figure 18-1.– Concluded.

18.1 LABEL

The LABEL subprogram is the driver routine for the LABEL processor.

18.1.1 LINKAGES

The LABEL subprogram calls the FILERD, LABLR, and SET14 subprograms. It is called by MONTOR.

18.1.2 INTERFACES

The LABEL subprogram interfaces with other routines through the calling arguments.

18.1.3 INPUTS

Calling sequence: CALL LABEL(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.

18.1.4 OUTPUTS

Not applicable.

18.1.5 STORAGE REQUIREMENTS

This subprogram requires 8548 bytes of storage.

18.1.6 DESCRIPTION

LABEL calls the subprograms SET14 to read and analyze the control cards, FILERD to read in the specified files, and LABLR to

coordinate the functions required to execute the options specified by the analyst.

18.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.1.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.2 ALLKIN

The ALLKIN subprogram labels the statistics obtained from the clustering processor ISOCLS, using the all-of-a-kind procedure.

18.2.1 LINKAGES

The ALLKIN subprogram calls the KNEAR and RREAD subprograms. It is called by the LABLR subprogram.

18.2.2 INTERFACES

The ALLKIN subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and LABS and through the calling arguments.

18.2.3 INPUTS

Calling sequence: CALL ALLKIN(DOTS,SUBVEC,SUBNO,CATVEC,MEANS, DOTSUM)

Parameter	Dimension	In/out	Definition
DOTS	SIZE,TOTDT2	In	Array containing the dot data information (SIZE = 4 + NOFEAT; TOTDT2 = total dots on the DOTUNT file).
SUBVEC	1	In	Array containing the sequence number used for ordering the output statistics.
SUBNO	1	Out	Array containing the cluster numbers in each category.
CATVEC	1	Out	Array containing the category number assigned to each cluster.

Parameter	Dimension	In/out	Definition
MEANS	NOFET2,1	In	Array containing the means of the unlabeled statistics.
DOTSUM	60,60	In/out	Array containing dot summary information.

18.2.4 OUTPUTS

This subprogram outputs line printer summaries of the dot labeling details for each cluster and of the results of cluster labeling for all subclasses processed.

18.2.5 STORAGE REQUIREMENTS

This subprogram requires 7826 bytes of storage.

18.2.6 DESCRIPTION

For a given cluster, ALLKIN scans the labeling dots and groups all the dots which were assigned to that cluster during the clustering process.

The labels for the group of dots are polled. If the labels are of one category, the cluster will bear that category label. If more than one category is encountered, the cluster is labeled according to the category with the majority of dots. If a given cluster does not contain any labeling dots within the cluster, the k-nearest-neighbor procedure is applied.

18.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.2.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.3 ASCEND

The ASCEND subprogram sorts an array of floating-point numbers in ascending order.

18.3.1 LINKAGES

The ASCEND subprogram does not call any other subprogram. It is called by the KNEAR subprogram.

18.3.2 INTERFACES

The ASCEND subprogram interfaces with other routines through the calling arguments.

18.3.3 INPUTS

Calling sequence: CALL ASCEND(SCN,LNCAT,PTR1,PTR2)

Parameter	Dimension	In/out	Definition
SCN	LNCAT	In/out	Array of cluster numbers to be sorted.
LNCAT	1	In	Number of elements to sort.
PTR1	LNCAT	In/out	Array containing information relating to SCN.
PTR2	LNCAT	In/out	Array containing information relating to SCN.

18.3.4 OUTPUTS

The results are returned for use by the calling routine.

18.3.5 STORAGE REQUIREMENTS

This subprogram requires 790 bytes of storage.

18.3.6 DESCRIPTION

The subprogram ASCEND operates on three arrays – a floating-point and two integer arrays. The floating-point array is arranged in increasing order; the two integer arrays are sorted based on the ordering of the floating-point array.

18.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.3.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.4 CLRKEY

The CLRKEY subprogram writes the color keys on the mixed and conditional cluster map tapes.

18.4.1 LINKAGES

The CLRKEY subprogram calls the WRTLN subprogram. It is called by the CLSMAP subprogram.

18.4.2 INTERFACES

The CLRKEY subprogram interfaces with other routines through the calling arguments.

18.4.3 INPUTS

Calling sequence: CALL CLRKEY(XSIZ,IDATA,NOSUB2,CH,MEANS,NC)

Parameter	Dimension	In/out	Definition
XSIZ	1	In	Number of pixels per line (maximum 200).
IDATA	XSIZ,CH	Out	Array containing colors output per line.
NOSUB2	1	In	Number of subclasses.
CH	1	In	Channel number set = 1; in the multichannel case, set = NC + 1.
MEANS	NC,NOSUB2	In	Array of colors corresponding to subclasses.
NC	1	In	Number of channels set = 1.

18.4.4 OUTPUTS

This subprogram adds the color keys to the mixed or conditional cluster map tape.

18.4.5 STORAGE REQUIREMENTS

This subprogram requires 1220 bytes of storage.

18.4.6 DESCRIPTION

For each cluster (subclass), a 10-by-10 color number is written on the conditional and/or mixed cluster map tape. This cluster number is obtained from an internal table.

18.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.4.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.5 CLSMAP

The CLSMAP subprogram outputs conditional and/or mixed cluster maps.

18.5.1 LINKAGES

The CLSMAP subprogram calls the CLRKEY, FSFMFL, MAPHND, RREAD, RWRITE, SETMRG, WRTHED, and WRTLN subprograms. It is called by the LABLR subprogram.

18.5.2 INTERFACES

The CLSMAP subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and LABS and through the calling arguments.

18.5.3 INPUTS

Calling sequence: CALL CLSMAP(CATSUB,SWTCH,SUBNO,SUBVEC,SUBDES, CATVEC)

Parameter	Dimension	In/out	Definition
CATSUB	1	In	Array that flags the mixed and conditional clusters.
SWTCH	1	In	If = 1, the conditional map is output; if = 2, the mixed map is output.
SUBNO	1	In	Array containing the number of clusters in a category.
SUBVEC	1	In	Array containing the sequence number used for ordering the output statistics.
SUBDES	60	In	Array containing the cluster names.
CATVEC	1	In	Array containing the category number assigned to each cluster.

18.5.4 OUTPUTS

This subprogram outputs a conditional and/or mixed cluster map on the specified unit. If both conditional and mixed cluster maps are requested, the conditional map is output first and then the mixed map.

A line printer symbol map of the conditional and/or mixed cluster map is printed.

18.5.5 STORAGE REQUIREMENTS

This subprogram requires 18 712 bytes of storage.

18.5.6 DESCRIPTION

For the line printer conditional or mixed map, each unconditional labeled cluster is assigned a category symbol and each conditional or mixed cluster is assigned a unique symbol.

For the output production film converter (PFC) tape the unconditional labeled clusters are arranged in alphabetical order, followed by the conditional or mixed clusters arranged by cluster number in ascending order. Each cluster is assigned a color by a table look-up technique.

The previously stored MAPUNT file is read into core from disk a line at a time. Each pixel of the line is assigned the appropriate predetermined symbol or color and output to the line printer and PFC tape, respectively.

18.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.5.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.6 CNDMAP

The CNDMAP subprogram flags conditional clusters.

18.6.1 LINKAGES

The CNDMAP subprogram calls the RREAD subprogram. It is called by the LABLR subprogram.

18.6.2 INTERFACES

The CNDMAP subprogram interfaces with other routines through common blocks INFORM and LABS and through the calling arguments.

18.6.3 INPUTS

Calling sequence: CALL CNDMAP(DOTS,CNDSUB,CATVEC)

Parameter	Dimension	In/out	Definition
DOTS	SIZE,TOTDT2	In	Dot data for each dot: sample, line, type, and category numbers.
CNDSUB	60	Out	Array for flagging conditional clusters. Each subclass is set equal to entry in CATVEC if not conditional; otherwise, it is set equal to 62 minus the number of conditional subclasses flagged previously.
CATVEC	1	In	Array containing category number for each subclass.

18.6.4 OUTPUTS

The results are returned for use by the calling routine.

18.6.5 STORAGE REQUIREMENTS

This subprogram requires 1808 bytes of storage.

18.6.6 DESCRIPTION

For each cluster I, the subprogram CNDMAP reads into core from disk that portion of the distance table for cluster I. The minimum distance between the labeled cluster I and dot of like label is found and compared with a threshold value. If the distance is greater than the threshold, the cluster is flagged as conditional. This flagging is done by assigning it a category number equal to 62 minus the number of clusters already flagged.

18.6.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.6.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.7 CRDSCN

The CRDSCN function interprets the control cards DOTLABEL and STATLABEL.

18.7.1 LINKAGES

The CRDSCN function calls the FIND12, NUMBER, and NXTCHR subprograms. It is called by the SET14 subprogram.

18.7.2 INTERFACES

The CRDSCN function interfaces with other routines through the calling arguments.

18.7.3 INPUTS

Calling sequence: CRDSCN(CARD,GRPDEX,GRPNAM,GROUPS,NOGRP,GRPTR)

Parameter	Dimension	In/out	Definition
CARD	62	In	Array containing the string of characters from control card DOTLABEL or STATLABEL.
GRPDEX	1	Out	Array containing the pointers for retrieving information from the GROUPS array.
GRPNAM	1	Out	Array containing input category names.
GROUPS	1	Out	Array containing the number of clusters associated with an input category name, followed by the actual cluster numbers.
NOGRP	60	Out	Number of category names encountered on an input card image at a given time.

Parameter	Dimension	In/out	Definition
GRPTR	1	In/out	Pointer for storing into the GROUPS array.

The control cards relevant to this routine are given in section 18 (table 18-1) of volume II of this user guide.

18.7.4 OUTPUTS

The results are returned for use by the calling routine.

18.7.5 STORAGE REQUIREMENTS

This subprogram requires 1310 bytes of storage.

18.7.6 DESCRIPTION

The function CRDSCN interprets an alphanumeric string, one character per word. The first seven characters are assumed to be alphabetic characters, followed by at least one numeric character.

The alphabetic characters define a category name plus a comma; the numeric characters are cluster numbers.

Three arrays are returned: One contains the labels (category names), one the numbers of the clusters or dots to relabel, and the other index pointers.

For the following control cards,

```
DOTLABEL     WHEAT, 3, 4, 7
DOTLABEL     NONWHT, 10, 12
```

the array will contain the following.

GRPNAM(1) = WHEAT	GRPDEX(1) = 1	GROUPS(1) = 3
GRPNAM(2) = NONWHT	GRPDEX(2) = 5	GROUPS(2) = 3
		GROUPS(3) = 4
		GROUPS(4) = 7
		GROUPS(5) = 7
		GROUPS(6) = 10
		GROUPS(7) = 12

18.7.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.7.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.8 DOTDST

The DOTDST subprogram computes the intercluster dot distance. The distances for all the labeling dots are computed and saved on disk for later referencing.

18.8.1 LINKAGES

The DOTDST subprogram calls the RREAD, RWRITE, and SUNFAC subprograms. It is called by the LABLR subprogram.

18.8.2 INTERFACES

The DOTDST subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and LABS and through the calling arguments.

18.8.3 INPUTS

Calling sequence: CALL DOTDST(MEANS,DOTS,TABLE,TOP)

Parameter	Dimension	In/out	Definition
MEANS	NOFET2,1	In	Array containing unlabeled means.
DOTS	SIZE,1	In	Array containing dot data.
TABLE	TOTDT2,1	In	Array containing the distances of all dots for each cluster.
TOP	1	In	Maximum number of words in TABLE.

18.8.4 OUTPUTS

This subprogram outputs the intercluster dot distance table on the line printer.

18.8.5 STORAGE REQUIREMENTS

This subprogram requires 2982 bytes of storage.

18.8.6 DESCRIPTION

The distance (L1 or L2) between each dot cluster pair is computed using a default Sun-angle correction of 60° or using an analyst-specified Sun-angle correction.

When the distances for a given cluster are computed, they are stored on high-speed disk one group at a time (a group contains the number of labeling dots). The number of writes to disk depends upon the number of cluster means.

To print the intercluster dot distance table, the program reads the distances into core and then prints them on the line printer.

18.8.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.8.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.9 DSPTAP

The DSPTAP subprogram writes an unformatted unconditional cluster map tape to be used as input to the DISPLAY processor to obtain certain general performance summaries.

18.9.1 LINKAGES

The DSPTAP subprogram calls the FSBSFL and RREAD subprograms. It is called by the LABLR subprogram.

18.9.2 INTERFACES

The DSPTAP subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and LABS and through the calling arguments.

18.9.3 INPUTS

Calling sequence: CALL DSPTAP(SUBNO,SUBDES,FLDSAV,VERTX,CATVEC, SUBVEC,MEANS,COVAR,TOP,DATA,NOFLD,TOTVRT)

Parameter	Dimension	In/out	Definition
SUBNO	1	In	Array containing numbers of subclasses in each category.
SUBDES	1	In	Array of subclass names.
FLDSAV	4,1	In	Array containing the following information for each training field: field name, class and subclass numbers, and number of field vertices.
VERTX	2,1	In	Array of field vertices.
CATVEC	1	In	Array containing category number for each subclass.

Parameter	Dimension	In/out	Definition
SUBVEC	1	In	Array containing reordering of original subclasses by category number.
MEANS	NOFET2,1	In	Array containing subclass mean vectors.
COVAR	VARSZ2,1	In	Array containing subclass covariance matrices by row as lower triangular matrix.
TOP	1	In	Maximum storage location in ARRAY; TOP = 10 600.
DATA	1	In	Array containing original cluster number to which th ixel was assigned during clustering (in ISOCLS).
NOFLD	1	In	Total number of fields.
TOTVRT	1	In	Total number of vertices.

18.9.4 OUTPUTS

This subprogram outputs an unformatted tape (MAPTAP).

18.9.5 STORAGE REQUIREMENTS

This subprogram requires 13 108 bytes of storage.

18.9.6 DESCRIPTION

Input of the MAPUNT control card to the LABEL processor initiates the call to subprogram DSPTAP, which writes an unformatted, unconditional, cluster map tape for use by the DISPLAY processor. The ' yout is the same as that for the MAPTAP written by the CLASSIFY processor (see section 4.4 of volume II of this user guide). However, only category and subclass information is

provided; and there is no inclusion of class/subclass and category/class associations. The information obtained is the result of cluster labeling in the LABEL processor, using either the k-nearest-neighbor or the all-of-a-kind procedure.

18.9.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.9.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.10 FILERD

The FILERD subprogram computes the needed high-speed disk addresses and reads in the input files.

18.10.1 LINKAGES

The FILERD subprogram calls the CMERR, LAREAD, RDDOTS, REDSAV, RWRITE, STOMAP, and WRTFLD subprograms. It is called by the LABEL driver routine.

18.10.2 INTERFACES

The FILERD subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and LABS and through the calling arguments.

18.10.3 INPUTS

Input to the FILERD subprogram consists of the DOTUNT, MAPUNT, and SAVTAP files output by the various processors. (See table 5-1, section 5, of volume II of this user guide for an overview of EOD-LARSYS files and processors.)

Calling sequence: CALL FILERD(ARRAY,TOP,NOFLD,TOTVRT,FLDSAV, VERTX)

Parameter	Dimension	In/out	Definition
ARRAY	1	Out	Working storage array (see section 18.1.3); contains the dot data information.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
NOFLD	1	Out	Number of fields on the DOTUNT file.
TOTVRT	1	Out	Number of field vertices on the DOTUNT file.

Parameter	Dimension	In/out	Definition
FLDSAV	4,1	Out	Array containing the field information on the DOTUNT file.
VERTX	2,1	Out	Array containing the field vertices on the DOTUNT file.

The field definition card image format relevant to this routine is described in section 3.2.3 of volume II of this user guide.

18.10.4 OUTPUTS

The results are stored for use by the calling routine.

18.10.5 STORAGE REQUIREMENTS

This subprogram requires 1672 bytes of storage.

18.10.6 DESCRIPTION

The FILERD subprogram accepts the MAPUNT, SAVTAP, or DOTUNT file as input.

If a MAPUNT file is input, the subprogram FILERD initiates the reading of the field card, computes the dimensions of the input file, reads in the file, and stores the entire image on high-speed disk.

If a SAVTAP file is input, the statistics are read into the upper partition of the array. Only the means and covariances are stored on high-speed disk.

If a DOTUNT file is input, the file is read in and stored starting in the location of ARRAY previously occupied by the covariances.

Based on the number of files read in, disk addresses that will be needed in other routines are computed and stored in common block LABS.

18.10.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.10.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.11 KNEAR

The KNEAR subprogram labels the statistics obtained from the clustering processor ISOCLS, using the k-nearest-neighbor procedure.

18.11.1 LINKAGES

The KNEAR subprogram calls the ASCEND and RREAD subprograms. It is called by the ALLKIN and LABLR subprograms.

18.11.2 INTERFACES

The KNEAR subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and LABS and through the calling arguments.

18.11.3 INPUTS

Calling sequence: CALL KNEAR(DOTS,SUBVEC,SUBNO,CATVEC,ITER, TAB1,SWTCH,CATNUM,CLUNUM,MEANS,DOTSUM)

Parameter	Dimension	In/out	Definition
DOTS	SIZE,1	In	Array containing the dot data information.
SUBVEC	60	In	Array containing the sequence number used for ordering the output statistics.
SUBNO	1	Out	Array containing the number of clusters in each category.
CATVEC	1	Out	Array containing the category number assigned to each cluster.
ITER	1	In	Number of iterations.
TAB1	1	In	Starting disk address for retrieving distances.

Parameter	Dimension	In/out	Definition
SWTCH	1	In	If = 0, a header record is written.
CATNUM	1	In	Category number assigned to cluster CLUNUM.
CLUNUM	1	In	Number of unlabeled cluster being labeled.
MEANS	NOFET2,1	In	Array containing means.
DOTSUM	60,60	In/out	Array containing dot summary information.

18.11.4 OUTPUTS

This subprogram outputs a line printer summary of the k-nearest labeling dots and Cluster Labeling Details and Cluster Labeling Summary reports.

18.11.5 STORAGE REQUIREMENTS

This subprogram requires 6470 bytes of storage.

18.11.6 DESCRIPTION

For a given cluster, a list of the category labels of the k-nearest dots to the cluster is computed. The label with the majority of the dots labels the cluster. If a tie occurs, k - 1 dots are used. The default value of k is 1.

18.11.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.11.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.12 LABDOT

The LABDOT subprogram labels or relabels the DOTUNT file.

18.12.1 LINKAGES

The LABDOT subprogram does not call any other subprogram. It is called by the LABLR subprogram.

18.12.2 INTERFACES

The LABDOT subprogram interfaces with other routines through common block LABS and through the calling arguments.

18.12.3 INPUTS

Calling sequence: CALL LABDOT(DOTS)

Parameter	Dimension	In/out	Definition
DOTS	SIZE,1	In	Array containing the dot data information.

18.12.4 OUTPUTS

The results are stored for use by the calling routine.

18.12.5 STORAGE REQUIREMENTS

This subprogram requires 958 bytes of storage.

18.12.6 DESCRIPTION

The category names input through control cards are checked for new name entries. If a new category has been input, the arrays concerning the category information are adjusted.

Each dot number input through a control card in subprogram SET14 is relabeled to the appropriate label. No dots may be deleted from the set of dots being updated.

18.12.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.12.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.13 LABLR

The LABLR subprogram is the coordinator and supervisor of the routines required to perform the operations specified by the analyst.

The analyst may either update the DOTUNT and/or SAVTAP files or label a given set of statistics by one of two procedures (k-nearest-neighbor or all-of-a-kind) in one execution of the LABEL processor.

18.13.1 LINKAGES

The LABLR subprogram calls the ALLKIN, CLSMAP, CNDMAP, DOTDST, DSPTAP, KNEAR, LABDOT, LABMAN, MANORD, MIXMAP, NAMSTA, REODER, RREAD, RWRITE, and WRTDOT subprograms. It is called by the LABEL driver routine.

18.13.2 INTERFACES

The LABLR subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and LABS and through the calling arguments.

18.13.3 INPUTS

Calling sequence: CALL LABLR(ARRAY,TOP,NOFLD,TOTVRT,FLDSAV, VERTX,MEANS,EXITT)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	Working storage array (see section 18.1.3).
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
NOFLD	1	In	Number of fields on the DOTUNT file.

Parameter	Dimension	In/out	Definition
TOTVRT	1	In	Number of vertices of fields on the DOTUNT file.
FLDSAV	4,1	In	Array containing the field information on the DOTUNT file.
VERTX	2,1	In	Array containing the field vertices of the fields on the DOTUNT file.
MEANS	NOFET2,1	In	Array for storing the means.
EXITT	1	In	Switch which allows program to exit if any labels were not used.

18.13.4 OUTPUTS

At the analyst's option, this subprogram outputs a conditional cluster map, a mixed cluster map, or the DISPLAY interface tape, MAPTAP.

18.13.5 STORAGE REQUIREMENTS

This subprogram requires 19 182 bytes of storage.

18.13.6 DESCRIPTION

Depending on the analyst-selected options, LABLR initializes the DOTSUM array to zero and supervises the following functions:

a. Relabeling of the SAVTAP file.

b. Relabeling of the DOTUNT file.

c. Execution of the k-nearest-neighbor procedure.

d. Execution of the all-of-a-kind procedure.

e. Output of the conditional cluster map.

f. Output of the mixed cluster map.

g. Output of the DISPLAY interface tape, MAPTAP.

18.13.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.13.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.14 MANORD

The MANORD subprogram relabels the SAVTAP file.

18.14.1 LINKAGES

This routine does not call any other subprogram. It is called by the LABLR subprogram.

18.14.2 INTERFACES

The MANORD subprogram interfaces with other routines through common block LABS and through the calling arguments.

18.14.3 INPUTS

Calling sequence: CALL MANORD(CLSNAM,CLSVEC,SUBVEC,NOCLS2,SUBNO, NOSUB2)

Parameter	Dimension	In/out	Definition
CLSNAM	1	In	Array containing class names (category names) on the SAVTAP file.
CLSVEC	1	In	Array containing the class-subclass correspondence table.
SUBVEC	1	Out	Array containing the sequence number used for ordering the output statistics.
NOCLS2	1	In	Number of class names on the SAVTAP file.
SUBNO	60	Out	Array containing the cluster numbers in each class (category).
NOSUB2	1	In	Number of clusters.

18.14.4 OUTPUTS

This subprogram outputs a list of input class names. The results are returned for use by the calling routine.

18.14.5 STORAGE REQUIREMENTS

This subprogram requires 1500 bytes of storage.

18.14.6 DESCRIPTION

The MANORD subprogram allows the user to relabel the SAVTAP file manually by inputting CLASSNAME card images (section 3.1.3, volume II). The MANORD subprogram compares the class names on the user-input card images with those from the SAVTAP file. If an error occurs on input, it generates a message and a list of names from the SAVTAP file. The SUBVC2 array is rearranged so that all subclasses (clusters) for a given class are grouped together. In this way, the user may regroup clusters into another class. (For example, if the cluster WHT1 belonged to the class WHEAT and the user decided it should belong to NONWHT, he could assign the cluster WHT1 to the class NONWHT.) The new class number for the relabeled subclass is stored, and the number of subclasses in each new class is computed. Subclass numbers are ordered according to the newly assigned class number and are stored in SUBVEC. The regrouping of clusters causes the SAVTAP file to be rearranged, but the regrouped clusters are not renamed.

18.14.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.14.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.15 MAPHND

The MAPHND subprogram prints out the heading for the conditional and/or mixed cluster map.

18.15.1 LINKAGES

The MAPHND subprogram does not call any other subprogram. It is called by the CLSMAP subprogram.

18.15.2 INTERFACES

The MAPHND subprogram interfaces with other routines through common block INFORM and through the calling arguments.

18.15.3 INPUTS

Calling sequence: CALL MAPHND(NOCAT,CLSSYM,CATNAM,KATNO,SUBDES, CATSUB)

Parameter	Dimension	In/out	Definition
NOCAT	1	In	Number of categories.
CLSSYM	1	In	Array containing symbols for categories or subclasses.
CATNAM	1	In	Array containing category names.
KATNO	1	In	Array containing the category number assigned to each cluster.
SUBDES	1	In	Array containing cluster names.
CATSUB	1	In	Array containing flagged conditional or mixed clusters.

18.15.4 OUTPUTS

This subprogram prints the heading for the conditional and/or mixed cluster map.

18.15.5 STORAGE REQUIREMENTS

This subprogram requires 1128 bytes of storage.

18.15.6 DESCRIPTION

All the clusters for a category are grouped together. Information such as the category name, cluster number, cluster name, and symbol for the cluster is printed on the heading of the conditional and/or mixed cluster map.

18.15.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.15.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.16 MIXMAP

The MIXMAP subprogram flags the conditional clusters.

18.16.1 LINKAGES

The MIXMAP subprogram calls the RREAD subprogram. It is called by the LABLR subprogram.

18.16.2 INTERFACES

The MIXMAP subprogram interfaces with other routines through common blocks GLOBAL and LABS and through the calling arguments.

18.16.3 INPUTS

Calling sequence: CALL MIXMAP(DOTS,MIXSUB,NOSUB2,CATVEC)

Parameter	Dimension	In/out	Definition
DOTS	SIZE,TOTDT2	In	Array containing the dot data.
MIXSUB	1	Out	Array used for flagging a mixed cluster on the output mixed map.
NOSUB2	1	In	Number of clusters.
CATVEC	1	In	Array containing the category number of each cluster.

18.16.4 OUTPUTS

The results are returned for use by the calling routine.

18.16.5 STORAGE REQUIREMENTS

This subprogram requires 2020 bytes of storage.

18.16.6 DESCRIPTION

The MIXMAP subprogram retrieves, from the high-speed disk, the cluster number assigned to each dot during clustering. All the dots belonging to cluster I are collected, and their labels are polled. If the labels define more than one category, cluster I is flagged as a mixed cluster. A mixed cluster is assigned a cluster of 62 - J, where J = the number of mixed clusters encountered at that time.

18.16.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.16.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.17 REODER

The REODER subprogram provides a reordering of subclass names and numbers of associated pixels, as part of the manual relabeling of statistics option in the LABEL processor.

18.17.1 LINKAGES

This routine does not call any other subprogram. It is called by the LABLR subprogram.

18.17.2 INTERFACES

The REODER subprogram interfaces with other routines through common block INFORM and through the calling arguments.

18.17.3 INPUTS

Calling sequence: CALL REODER(ARRAY,SUBVEC,N)

Parameter	Dimension	In/out	Definition
ARRAY	1	In/out	Working storage (see section 18.1.3) containing subclass names, starting at index SUBDS2.
SUBVEC	1	In	Array containing reordering of subclass pixels by category.
N	1	Out	Array containing number of pixels in each subclass, as reordered in SUBVEC.

18.17.4 OUTPUTS

The results are returned for use by the calling routine.

18.17.5 STORAGE REQUIREMENTS

This subprogram requires 820 bytes of storage.

18.17.6 DESCRIPTION

The subclass names and numbers of pixels per subclass are reordered according to SUBVEC. The new ordering places all subclasses associated with the first category first in order, all subclasses associated with the second category second in order, and so on.

18.17.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.17.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.18 SET14

The SET14 subprogram reads and analyzes the control cards for the LABEL processor.

18.18.1 LINKAGES

The SET14 subprogram calls the CRDSCN, CRDSTA, FIND12, FLTNUM, NUMBER, NXTCHR, and ORDER subprograms. It is called by the LABEL driver routine.

18.18.2 INTERFACES

The SET14 subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and LABS and through the calling arguments.

18.18.3 INPUTS

Calling sequence: CALL SET14(ARRAY,TOP,EXIT)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	Working storage (see section 18.1.3).
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
EXIT	1	Out	Switch which allows program to exit if any labels were not used.

The control cards relevant to this routine are given in section 18 (table 18-1) of volume II of this user guide.

18.18.4 OUTPUTS

This subprogram outputs a line printer summary of the selected options.

18.18.5 STORAGE REQUIREMENTS

This subprogram requires 9972 bytes of storage.

18.18.6 DESCRIPTION

Each control card is read and analyzed for format errors. If the control card is valid, the appropriate parameters and switches are set. If the card image is invalid, an appropriate error message is generated. When applicable, if a control card for a specific option is not input, a default value is supplied.

18.18.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 18.20.

18.18.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.19 STOMAP

The STOMAP subprogram reads the input MAPUNT tape and stores the image on high-speed disk.

18.19.1 LINKAGES

The STOMAP subprogram calls the CMERR, FLDINT, LINERD, RWRITE, and TAPHDR subprograms. It is called by the FILERD subprogram.

18.19.2 INTERFACES

The STOMAP subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

18.19.3 INPUTS

Input to the STOMAP subprogram consists of the MAPUNT tape containing the clustered or classified map output by the ISOCLS or DISPLAY processor.

Calling sequence: CALL STOMAP(ILINE,NSAMP,HIST,LIMIT,BEGIN1)

Parameter	Dimension	In/out	Definition
ILINE	1	In	Number of lines on the MAPUNT file.
NSAMP	1	In	Number of samples per scan line.
HIST	LIMIT	In	Scratch area.
LIMIT	1	In	Parameter dimensioning HIST.
BEGIN1	1	In	Beginning disk address for storing the MAPUNT file.

18.19.4 OUTPUTS

The results are stored on high-speed disk.

18.19.5 STORAGE REQUIREMENTS

This subprogram requires 1234 bytes of storage.

18.19.6 DESCRIPTION

Based on the size of both the array HIST and the input MAPUNT file, the subprogram STOMAP computes how many scan lines of the file can be read into core at one time. When the buffer is filled, the data are dumped onto high-speed disk. (The number of dumps to the disk is a function of the number of lines that can reside in core.)

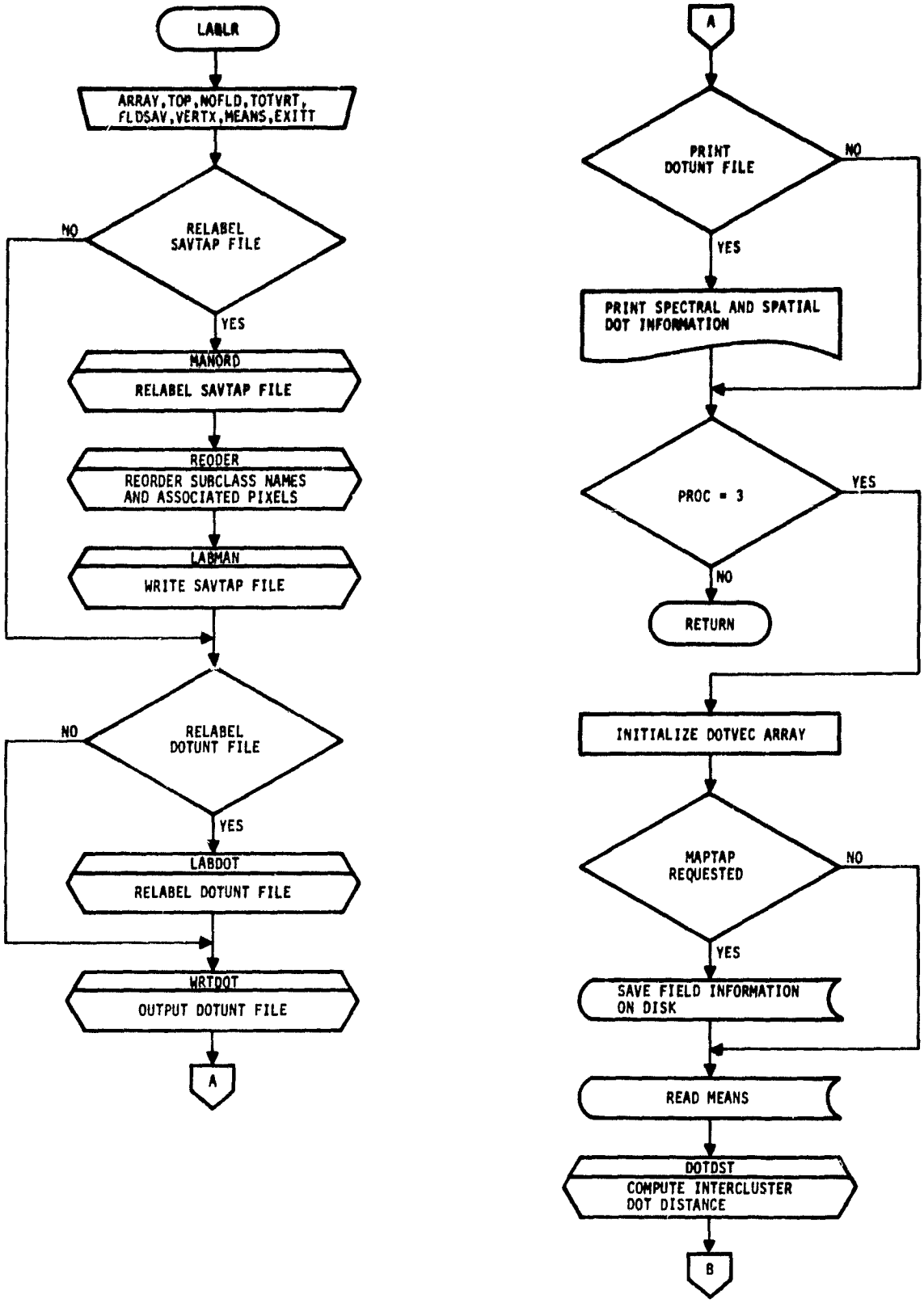
18.19.7 FLOW CHART

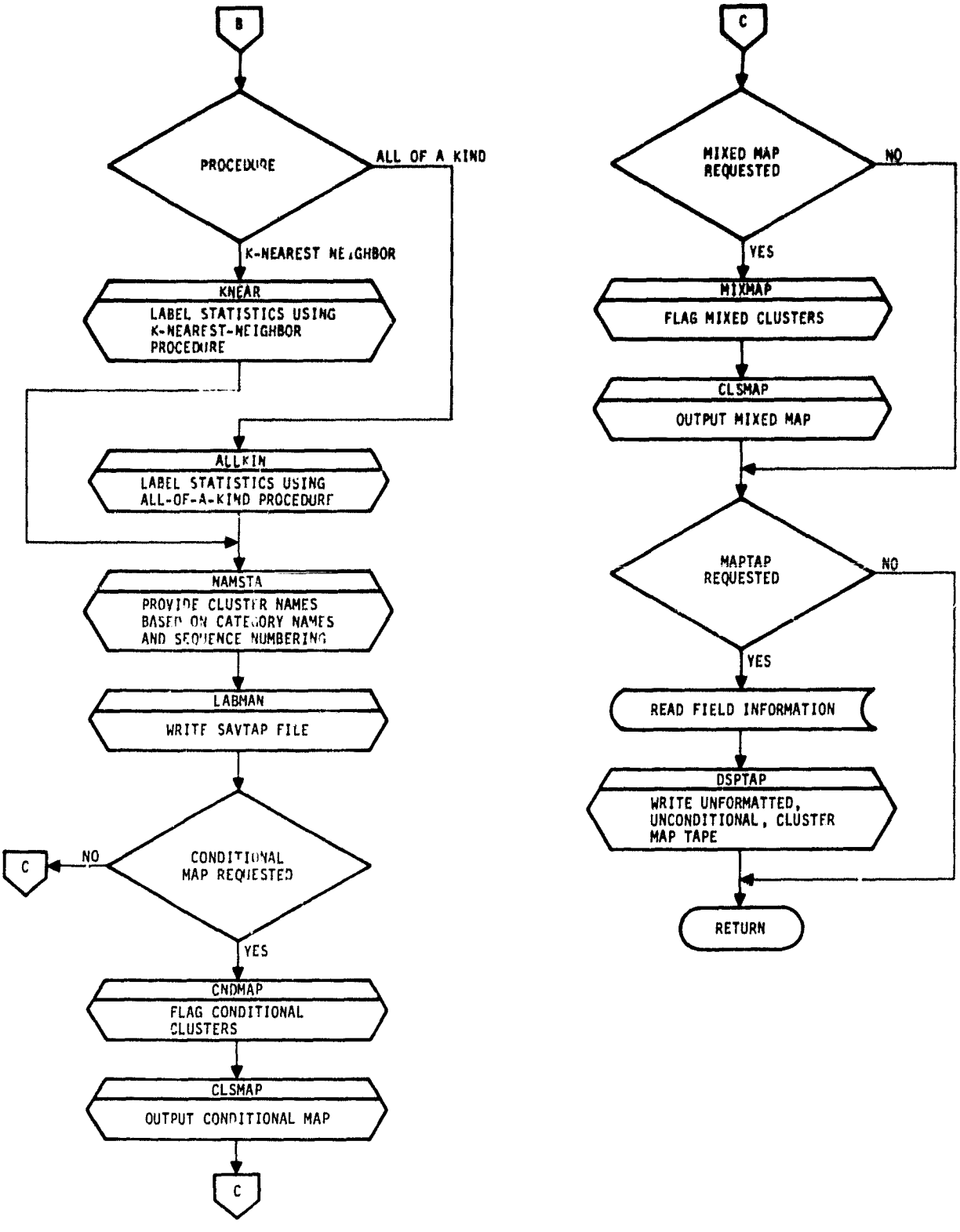
The available subprogram flow charts for this processor are provided in section 18.20.

18.19.8 LISTING

The subprogram listing is provided in volume IV, section 18.

18.20 SUBPROGRAM FLOW CHARTS




B
PROCEDURE
ALL OF A KIND
K-NEAREST NEIGHBOR
KNEAR
LABEL STATISTICS USING K-NEAREST-NEIGHBOR PROCEDURE
ALLKIN
LABEL STATISTICS USING ALL-OF-A-KIND PROCEDURE
NAMSTA
PROVIDE CLUSTER NAMES BASED ON CATEGORY NAMES AND SEQUENCE NUMBERING
LABMAN
WRITE SAVTAP FILE
CONDITIONAL MAP REQUESTED
NO
C
YES
CNDMAP
FLAG CONDITIONAL CLUSTERS
CLSMAP
OUTPUT CONDITIONAL MAP
C
C
MIXED MAP REQUESTED
NO
YES
MIXMAP
FLAG MIXED CLUSTERS
CLSMAP
OUTPUT MIXED MAP
MAPTAP REQUESTED
NO
YES
READ FIELD INFORMATION
DSPTAP
WRITE UNFORMATTED, UNCONDITIONAL, CLUSTER MAP TAPE
RETURN

19. UTILITY SUBPROGRAMS

The utility subprograms may be used by two or more processors. Sixty utility subprograms and their functions are listed in table 19-1. The subprograms are documented in sections 19.1 through 19.60.

TABLE 19-1.- UTILITY SUBPROGRAMS AND FUNCTIONS

Subprogram	Function
BMFIL	Performs input/output functions with respect to the B-matrix.
BNI4A1	Converts internal binary to EBCDIC characters.
BUFILL	Reads MSS DATAPE one record at a time.
CHAIN	Links all clusters having means less than a specified distance apart.
CHLDET	Computes the modified Cholesky decomposition and determinant of the covariance matrix.
CLDIST	Calculates weighted distance between cluster means.
CLSCHK	Checks user-input classes, subclasses, groupings, and channels.
CLSHIS	Scales and prints a histogram of training subclass and field data.
CMERR	Prints an error message and terminates execution.
COMHST	Entry to CLSHIS.
CRDSTA	Reads card image input, computes base addresses, and stores data.
DESCEN	Arranges a set of integers is descending order.
DSTAPE	Generates a formatted cluster map file on MAPUNT.
DWRTMX	Entry to WRTMTX.
FDLINT	Accepts a nonrectangular field table as input and returns the x-intercepts for the given scan line.
FIND12	Scans card image input and locates special symbols.
FLDHIS	Entry to CLSHIS.
FLDINT	Unpacks pixels from specified lines on the MSS DATAPE.
FLTNUM	Scans card images, interprets read numbers separated by commas, and returns them in an array.

TABLE 19-1.- Continued.

Subprogram	Function
FSBSFL	Positions file for an unformatted read or write.
FSFMFL	Positions file for a formatted read or write.
GETINF	Retrieves stored information.
GETST	Retrieves, formats, writes and stores statistics from the SAVTAP file.
GRPSCN	Scans field definition card images and determines if classes or subclasses have been assigned to groups.
HISTGM	Calculates histograms and writes histogrammed statistics on file.
HISTIC	Computes and displays statistics for the histogram.
HSTGRM	Entry to CLSHIS.
I4A1BN	Converts EBCDIC digits to internal binary integers.
LABMAN	Writes a formatted SAVTAP file and creates a module STAT file.
LAREAD	Reads field definition card images.
LINERD	Reads one scan line of data from the MSS DATAPE.
LISTLC	Reads dot data files, computes line and sample increments, and stores data.
MATVEC	Multiplies a matrix A by a vector B and stores the result in a vector C.
MTMDAT	Multiplies a matrix A by the transpose of a matrix B and stores the result in a lower triangular matrix DD.
MTMLS6	Multiplies a matrix A by a vector B and stores the result in matrix C.
NAMSTA	Provides cluster names based on category names and sequence numbering.
NUMBER	Scans control card images searching for integers.

TABLE 19-1.- Continued.

Subprogram	Function
NUMBR	Processes one field definition card image at a time and reads and stores all numbers.
NXTCHR	Locates and enters the next nonblank character on a card image being read.
ORDER	Arranges a set of N integers in ascending order.
PRINT	Generates most of the output for the ISOCLS and TESTSP processors.
PRTCOV	Writes the transformed covariance matrix.
RANK	Calculates greenness value for each cluster and outputs an ordered list of color keys, cluster numbers, and greenness values.
RDDOTS	Reads the DOTUNT file.
RDDOT1	Retrieves spectral and/or spatial information from the DOTUNT file.
RDFILE	Entry to RDMEAN.
RDMEAN	Reads a card image file, stores requested channels, and prints a summary of means for requested channels.
RDMODK	Reads in remainder of the module STAT file and writes statistics on the SAVTAP file.
REDDAT	Reads covariances and means from the SAVTAP file and reduces statistics.
REDSAV	Reads the SAVTAP file and reduces statistics to a user-requested subset.
RREAD	Simulates the random read of a work file used to store data temporarily during execution.
RWRITE	Simulates the random write of a work file used to store data temporarily during execution.
SAVFIL	Reads records from the SAVTAP file.
SEARCH	Searches for the correct scan line to process.
SETMRG	Inactive.

TABLE 19-1.- Continued.

Subprogram	Function
SETUP7	Reads control cards, analyzes supervisory information, and sets options for the ISOCLS and TESTSP processors.
SUNFAC	Computes Sun-angle gain corrections for pixel radiance values for each channel based on input Sun angles.
TAPHDR	Reads the header record of the MSS DATAPE (formatted read).
WRTBM	Entry to WRTBMT.
WRTBMT	Writes the transformed B-matrix.
WRTDOT	Outputs the DOTUNT file for the LABEL and DOTDATA processors.
WRTFLD	Prints saved training or test fields.
WRTHED	Writes the header record for each data tape (formatted write).
WRTLN	Writes the data for each data tape (formatted write).
WRTMTX	Prints the single-precision covariance matrices; entry DWRTMX prints the double-precision covariance matrix.
WRTREC	Prints a scan line (one record) of data.

19.1 BMFIL

The BMFIL subprogram performs one of five input/output functions with regard to the B-matrix.

19.1.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	REDIF2
DATA-TR	SETUP8
SELECT	DAVIDN, SETUP4, and USERIN
SCTRPL	SET11

19.1.2 INTERFACES

The BMFIL subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

19.1.3 INPUTS

Calling sequence: CALL BMFIL(BMAT,LCOMB,NOFET,VEC,KEY)

Parameter	Dimension	In/out	Definition
BMAT	LCOMB,NOFET	In/out	Linear transformation matrix.
LCOMB	1	In/out	Number of linear combinations in BMAT.
NOFET	1	In/out	Number of features in BMAT.
VEC	NOFET	In/out	Vector containing features used in obtaining the B-matrix.

Parameter	Dimension	In/out	Definition
KEY	1	In	Determines the input/output function to be performed.

19.1.4 OUTPUTS

This subprogram outputs the B-matrix on file or cards.

19.1.5 STORAGE REQUIREMENTS

This subprogram requires 1372 bytes of storage.

19.1.6 DESCRIPTION

Depending on the value of the parameter KEY, one of five input/output functions is performed for the calling routine: If KEY = 1, the B-matrix is read from card images and stored on file; if KEY = 2, the B-matrix is read from the file; if KEY = 3, the values of LCOMB, NOFET, and VEC are read from the file (used to establish B-matrix dimensions); if KEY = 4, the B-matrix is output on cards; and if KEY = 5, the B-matrix is output to tape or disk file.

19.1.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.1.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.2 BNI4A1

The BNI4A1 subprogram converts internal binary numbers from the integer-4 form to a string of EBCDIC characters. The characters are stored one per word in the high-order bit positions.

19.2.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	NAMSTA
LABEL	NAMSTA
TESTSP	NAMSTA

19.2.2 INTERFACES

The BNI4A1 subprogram interfaces with other routines through the calling arguments.

19.2.3 INPUTS

Calling sequence: CALL BNI4A1(IFLD,INCHR,IBN)

Parameter	Dimension	In/out	Definition
IFLD	20	Out	First word of a field in an array for storage of output EBCDIC characters, one character per word in A1 format with blank fill to the right.
INCHR	1	In	Number of EBCDIC characters (width of the field); should be integer-4 form also.

Parameter	Dimension	In/out	Definition
IBN	1	In	Integer in internal binary form, positive or negative in the legal range -2^{31} to $(2^{31}-1)$.

19.2.4 OUTPUTS

The results are returned for use by the calling routine.

19.2.5 STORAGE REQUIREMENTS

This subprogram requires 722 bytes of storage.

19.2.6 DESCRIPTION

The BNI4A1 subprogram accepts integers in internal binary, positive or negative, integer-4 form. It converts the integers to a string of EBCDIC characters and stores them left justified, one character per word, with blank fill to the right.

19.2.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.2.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.3 BUFILL

The BUFILL subprogram reads the MSS DATAPE one record at a time.

19.3.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	LINERD and TAPHDR
DAMRG	LINERD and TAPHDR
DATA-TR	LINERD and TAPHDR
DOTDATA	LINERD and TAPHDR
GRAYMAP	LINERD and TAPHDR
GTDDM	LINERD and TAPHDR
GTTCN	LINERD and TAPHDR
HIST	LINERD and TAPHDR
ISOCLS	LINERD and TAPHDR
LABEL	LINERD and TAPHDR
NDHIST	LINERD and TAPHDR
STAT	LINERD and TAPHDR
TESTSP	LINERD and TAPHDR

19.3.2 INTERFACES

The BUFILL subprogram interfaces with other routines through the calling arguments.

19.3.3 INPUTS

Calling sequence: CALL BUFILL(IREC,IUNIT,MAXREC,IBUF,NRPDS, ENDTAP,IERR)

Parameter	Dimension	In/out	Definition
IREC	1	In/out	Counter for the record being read.
IUNIT	1	In	Unit number of the record being read.
MAXREC	1	In	Length of one record to be read.
IBUF	765	In/out	Dimension of MAXREC.
NRPDS	1	In	Number of records per data set.
ENDTAP	1	Out	If = -1, EOF was encountered.
IERR	1	Out	Error flag; if a tape error occurs, a message is written and control returns to the calling routine.

19.3.4 OUTPUTS

The results are returned for use by the calling routine.

19.3.5 STORAGE REQUIREMENTS

This subprogram requires 812 bytes of storage.

19.3.6 DESCRIPTION

The subprogram BUFILL reads one record at a time from the MSS DATAPE. It reads NRPDS records up to a maximum of 10 records in an execution.

19.3.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.3.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.4 CHAIN

The CHAIN subprogram links all clusters with names less than a specified distance apart.

19.4.1 LINKAGES

The CHAIN subprogram does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	ISOCLS
TESTSP	TESTSP

19.4.2 INTERFACES

The CHAIN subprogram interfaces with other routines through common blocks GLOBAL and PASS and through the calling arguments.

19.4.3 INPUTS

Calling sequence: CALL CHAIN(CLD)

Parameter	Dimension	In/out	Definition
CLD	MAXCLS,MAXCLS	In	Array containing cluster distances; MAXCLS = maximum number of clusters per class.

19.4.4 OUTPUTS

This subprogram outputs a printed summary of clusters which were chained.

19.4.5 STORAGE REQUIREMENTS

This subprogram requires 1804 bytes of storage.

19.4.6 DESCRIPTION

This subprogram chains all clusters, the means of which are less than DLMIN units apart. CHAIN builds an array ICHAIN to contain the numbers of chained clusters and prints a summary of chained clusters. (The chaining process is described in detail in volume II, section 9.1.4.)

19.4.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.4.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.5 CHLDET

The CHLDET subprogram computes the modified Cholesky decomposition of the covariance matrix. To ascertain if the covariance matrix is singular, CHLDET computes and returns the determinant.

19.5.1 LINKAGES

The CHLDET subprogram does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	COVAR1
TESTSP	COVPAT

19.5.2 INTERFACES

The CHLDET subprogram interfaces with other routines through the calling arguments.

19.5.3 INPUTS

Calling sequence: CALL CHLDET(KKK,NV,DUM,DET)

Parameter	Dimension	In/out	Definition
KKK	1	In	Array containing the covariance matrix stored symmetrically.
NV	1	In	Number of channels to be processed.
DUM	1	Out	Working storage array of size NV - 1.
DET	1	Out	Determinant of the covariance matrix.

19.5.4 OUTPUTS

The results are returned for use by the calling routine.

19.5.5 STORAGE REQUIREMENTS

This subprogram requires 2986 bytes of storage.

19.5.6 DESCRIPTION

The CHLDET subprogram transfers the input covariance matrix KKK into KK. It sets up a loop over all input channels and (1) computes the elements of the diagonal matrix D and stores them in the diagonal elements of the covariance matrix KK, (2) computes the elements of the lower triangular matrix L and stores them in the off-diagonal elements of KK, and (3) obtains the modified Cholesky factorization of the input matrix and stores the result in symmetric notation in KK. The determinant is returned in DET.

19.5.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.5.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.6 CLDIST

The CLDIST subprogram calculates the weighted distance between cluster means.

19.6.1 LINKAGES

The CLDIST subprogram does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	ISODAT
TESTSP	ISOPAT

19.6.2 INTERFACES

The CLDIST subprogram interfaces with other routines through common block PASS and through the calling arguments.

19.6.3 INPUTS

Calling sequence: CALL CLDIST(CLD,STDEV,MEANS)

Parameter	Dimension	In/out	Definition
CLD	MAXCLS,MAXCLS	Out	Array containing distances between clusters; CLD(N,M) = distance between clusters N and M.
STDEV	NOFEAT,MAXCLS	In	Array containing standard deviations for each feature/ cluster.
MEANS	NOFEAT,MAXCLS	In	Array containing means of each feature/cluster.

19.6.4 OUTPUTS

Not applicable.

19.6.5 STORAGE REQUIREMENTS

This subprogram requires 1228 bytes of storage.

19.6.6 DESCRIPTION

CLDIST calculates the distance between clusters using the following equation:

$$CLD_{ij} = \left[\frac{(\mu_{ki} - \mu_{kj})^2}{\sigma_{ki}\sigma_{kj}} \right]^{1/2}$$

where

i,j = index for the specific cluster

k = index for the specific feature

μ = mean

σ = standard deviation

If σ_{ki} or σ_{kj} = 0, CLD_{ij} is arbitrarily set to 999.99 to prevent the cluster from being cha ned to other clusters. A standard deviation of zero usually indicates bad data.

19.6.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.6.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.7 CLSCHK

The CLSCHK subprogram checks the validity of user requests regarding classes, s bclasses, groupings, and channels and saves the results.

19.7.1 LINKAGES

This routine calls the ORDER subprogram. The subprogram which calls it, along with the processors, is listed below.

Processor	Calling subprogram
CLASSIFY	REDSAV
DATA-ER	REDSAV
LABEL	REDSAV
SELECT	REDSAV
TRSTAT	REDSAV

19.7.2 INTERFACES

The CLSCHK subprogram interfaces with other routines through common block INFORM and through the calling arguments.

19.7.3 INPUTS

Calling sequence: CALL CLSCHK(CLSDES,SUBDES,FLDSAV,VERTEX,SUBNO, NOFEAT,FETVEC,NOCLS,NOFLD,BMFLG,NOSUB)

Parameter	Dimension	In/out	Definition
CLSDES	1	In/out	Array of class names.
SUBDES	1	In/out	Array of subclass names.
FLDSAV	4,NOFLD	In/out	Array containing field information for classes and subclasses to be used.

Parameter	Dimension	In/out	Definition
VERTEX	1	In/out	Array containing field vertices.
SUBNO	1	In	Array containing number of subclasses in each class.
NOFEAT	1	In	Number of features (channels).
FETVEC	30	In	Array containing channels.
NOCLS	1	In	Number of classes to be processed.
NOFLD	1	In	Number of fields to be processed.
BMFLG	1	In	Flag indicating that the B-matrix has been input.
NOSUB	1	In	Number of subclasses to be processed.

19.7.4 OUTPUTS

The results are returned for use by the calling routine.

19.7.5 STORAGE REQUIREMENTS

This subprogram requires 5640 bytes of storage.

19.7.6 DESCRIPTION

The CLSCHK subprogram performs the following functions:

a. Checks to assure that each requested subclass is in the input statistics; if not, it prints a diagnostic message and proceeds to the next subclass.

b. If the subclass has been assigned to a group, checks to assure that the group is in the input statistics.

c. Deletes all group subclasses, adds back the first subclass from each group, and constructs a group for each subclass not explicitly grouped.

d. Calls subroutine ORDER to arrange subclasses in ascending order.

e. Initializes an array, SUBPTR, which contains the new index for each subclass to be used in processing.

f. Checks the validity of the requested channels, writes a message if the channel is not valid, and ignores it in classification.

g. Checks the B-matrix channels to assure that they are a subset of available training data channels; if channels are incorrect, an error message is printed and the program terminates.

h. If channels are correct, sets up a revised inverse table of channels.

i. Sets up the CLSVC2 array for storing the class numbers to which corresponding subclasses belong.

j. Saves field descriptions for classes and subclasses to be used.

19.7.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.7.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.8 CLSHIS

The CLSHIS subprogram scales and prints a histogram of training subclass and training field data.

19.8.1 LINKAGES

This routine calls the SETMRG subprogram. The subprograms which call it, along with the respective processors, are listed below. CLSHIS has three entry points: COMHST, FLDHIS, and HSTGRM.

Processor	Calling subprogram
DATA-TR	LNTRAN
GRAYMAP	HISTGM
HIST	HISTGM
STAT	LEARN

19.8.2 INTERFACES

The CLSHIS subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

19.8.3 INPUTS

Calling sequences:

a. CALL CLSHIS(TALLY,HISBUF,TTL,XSIZ,XHGH,XLOW,YSIZ,NOHIST, FLDPTS,HISVEC)

Parameter	Dimension	In/out	Definition
TALLY	NOHIST,XSIZ	In	Array containing data to be histogrammed.
HISBUF	XSIZ	Out	Storage for histogrammed data.
TTL	1	In	Title.

Parameter	Dimension	In/out	Definition
XSIZ	1	In	XHGH - XLOW.
XHGH	1	In	Maximum x-value to be histogrammed.
XLOW	1	In	Minimum x-value to be histogrammed.
YSIZ	1	In	Height of the y-axis in the histogram.
NOHIST	1	In	Number of channels to display in the histogram.
FLDPTS	1	In	Number of points in the field.
HISVEC	30	Out	Vector of histogrammed data.

b. ENTRY COMHST(TALLY,HISBUF,TTL,NOHIST,HISVEC,XSIZ,XHGH,XLOW, YSIZ)

c. ENTRY FLDHIS(TALLY,HISBUF,TTL,XSIZ,XHGH,XLOW,YSIZ,NOHIST, FLDPTS,TITLE,HISVEC)

Parameter	Dimension	In/out	Definition
TITLE	1	In	Title.

d. ENTRY HSTGRM(TALLY,HISBUF,TTL,PRINT,XSIZ,XHGH,XLOW,YSIZ, NOHIST,FLDPTS,HISVEC)

Parameter	Dimension	In/out	Definition
PRINT	1	In	Determines format to use in printing title.

19.8.4 OUTPUTS

This subprogram outputs a histogram of training field and subclass data.

19.8.5 STORAGE REQUIREMENTS

This subprogram requires 4050 bytes of storage.

19.8.6 DESCRIPTION

Not required.

19.8.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.8.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.9 CMERR

The CMERR subprogram generates an error message.

19.9.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	CLSFY2, CRDSTA, FLDINT, LAREAD, REDIF2, REDSAV, SEARCH, SETUP2, and TAPHDR
DAMRG	FLDINT, LAREAD, SEARCH, SET18, and TAPHDR
DATA-TR	CRDSTA, FLDINT, LAREAD, REDSAV, SEARCH, SETREM, and TAPHDR
DISPLAY	DSPLY1, EMTHRS, LAREAD, LISTSM, REDIF3, and SETUP3
DOTDATA	FLDINT, LAREAD, SEARCH, and TAPHDR
GRAYMAP	FLDINT, LAREAD, PICT, SEARCH, and TAPHDR
GTDDM	FLDINT, SEARCH, and TAPHDR
GTTCN	FLDINT, SEARCH, and TAPHDR
HIST	FLDINT, LAREAD, SEARCH, and TAPHDR
ISOCLS	COVAR1, CRDSTA, DSTAPE, FLDINT, GETST, ISOCLS, LAREAD, RDDATA, RDDOTS, SEARCH, and TAPHDR
LABEL	CRDSTA, FILERD, LAREAD, RDDOTS, REDSAV, SEARCH, STOMAP, and TAPHDR
NDHIST	ADDRES, FLDINT, LAREAD, NDHST1, RESTO (RESTOR), SEARCH, STODAT, and TAPHDR
SCTRPL	CRDSTA, GETST, SETADR, and SET11

Processor	Calling subprograms
SELECT	AVEDIV, BHTCHR, CRDSTA, DAVIDN, DAVDN1, DIVERG (DIVRG1), GTSTAT, REDSAV, SELECT, SETUP4, and TRNDIV
STAT	FLDINT, LAREAD, LEARN, SEARCH, and TAPHDR
TESTSP	COVPAT, CRDSTA, DSTAPE, FLDINT, GETST, LAREAD, RDDOTS, RDDPAT, SEARCH, and TESTSP
TRSTAT	CRDSTA, REDSAV, SETUP9, and TRAMTX

19.9.2 INTERFACES

The CMERR subprogram interfaces with other routines through the calling sequence.

19.9.3 INPUTS

Calling sequence: CALL CMERR

19.9.4 OUTPUTS

This subprogram outputs the following error message: "ERROR HAS OCCURRED."

19.9.5 STORAGE REQUIREMENTS

This subprogram requires 300 bytes of storage.

19.9.6 DESCRIPTION

Not required.

19.9.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.9.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.10 CRDSTA

The CRDSTA subprogram reads card image input, computes base addresses, and stores data in ARRAY.

19.10.1 LINKAGES

This routine calls the CMERR and RDMODK subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
CLASSIFY	REDIF2
DATA-TR	SETUP8
ISOCLS	SETUP7
LABEL	SET14
SCTRPL	SET11
SELECT	SETUP4
TESTSP	SETUP7
TRSTAT	SETUP9

19.10.2 INTERFACES

The CRDSTA subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and PASSB and through the calling arguments.

19.10.3 INPUTS

Calling sequence: CALL CRDSTA(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	Out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.

19.10.4 OUTPUTS

This subprogram stores data in ARRAY.

19.10.5 STORAGE REQUIREMENTS

This subprogram requires 1058 bytes of storage.

19.10.6 DESCRIPTION

The CRDSTA subprogram reads keywords to be used in base addresses and computes base addresses for data input on cards. If the module STAT deck is input, subprogram RDMODK is called to read in the statistical data and write them on the SAVTAP file. CRDSTA prints an error message and exits if input exceeds core limits.

19.10.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.10.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.11 DESCEN

The DESCEN subprogram arranges a set of integers in descending order.

19.11.1 LINKAGES

The DESCEN subprogram does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	ISODAT
TESTSP	ISOPAT

19.11.2 INTERFACES

The DESCEN subprogram interfaces with other routines through the calling arguments.

19.11.3 INPUTS

Calling sequence: CALL DESCEN(SCN,LNCAT,PTR1,PTR2)

Parameter	Dimension	In/out	Definition
SCN	LNCAT	In/out	Array containing integers to be sorted.
LNCAT	1	In	Current number of clusters.
PTR1	LNCAT	In/out	Array containing pointers to combine clusters.
PTR2	LNCAT	In/out	Array containing pointers to split clusters.

19.11.4 OUTPUTS

The results are returned for use by the calling routine.

19.11.5 STORAGE REQUIREMENTS

This subprogram requires 790 bytes of storage.

19.11.6 DESCRIPTION

Not required.

19.11.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.11.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.12 DSTAPE

The DSTAPE subprogram generates a MAPUNT file in either Universal or LARSYS II format, if requested by the user.

19.12.1 LINKAGES

The DSTAPE subprogram calls the CMERR, FDLINT, RANK, RREAD, WRTHED, and WRTLN subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	ISOCLS
TESTSP	TESTSP

19.12.2 INTERFACES

The DSTAPE subprogram interfaces with other routines through common blocks GLOBAL and PASS and through the calling arguments.

19.12.3 INPUTS

Calling sequence: CALL DSTAPE(IPLACE,IBUF,MEANS,FLDINF)

Parameter	Dimension	In/out	Definition
IPLACE	NOPTS	In	Vector of cluster numbers to which corresponding data points are assigned.
IBUF	1	Out	Storage buffer used for one scan line of data taken from IPLACE.
MEANS	NOFEAT,MAXCLS	In	Array containing means of each feature for each cluster.

Parameter	Dimension	In/out	Definition
FLDINF	1	In	Array containing field information [name, class and subclass numbers, and vertices (sample and line numbers)].

19.12.4 OUTPUTS

This subprogram outputs a MAPUNT file.

19.12.5 STORAGE REQUIREMENTS

This subprogram requires 4274 bytes of storage.

19.12.6 DESCRIPTION

Beginning with the first field input by the user, a buffer is filled with the cluster identification for one scan line of data. Subprogram FDLINT is called to retrieve the pixel numbers and return the ordered pixel intercepts on the scan line. The line is written on tape until all lines from a given field have been written on MAPUNT. One file is output for each input field.

19.12.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.12.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.13 FDLINT

The FDLINT subprogram returns the pixel numbers of those pixels on a given line that are contained within the boundaries of a nonrectangular field.

19.13.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	CATGRY and CONTEX
DATA-TR	LNTRAN and TRHIST
DISPLAY	DESIG, FLDBOR, and PCT (PCTT)
DOTDATA	DOTS
GRAYMAP	HISTGM and PICT
HIST	HISTGM
ISOCLS	DSTAPE, PRINT, and RDDATA
NDHIST	NDHST1
STAT	LEARN
TESTSP	DSTAPE, PRINT, and RDDPAT

19.13.2 INTERFACES

The FDLINT subprogram interfaces with other routines through the calling arguments.

19.13.3 INPUTS

Calling sequence: CALL FDLINT(FIELD,NPTS,FL,YLINE,NSAMP,JJ)

Parameter	Dimension	In/out	Definition
FIELD	2,NPTS	In	Nonrectangular field table. (All vertices must be in clockwise order, and the last vertex must equal the first vertex for field closure. The first vertex must have a minimum pixel value.)
NPTS	1	In	Number of points in FIELD.
FL	8	Out	Array containing the ordered pixel intercepts.
YLINE	1	In	Scan line number.
NSAMP	1	Out	Number of samples in the field of a given scan line.
JJ	1	Out	Length of the array FL.

19.13.4 OUTPUTS

The results are returned for use by the calling routine.

19.13.5 STORAGE REQUIREMENTS

This subprogram requires 3284 bytes of storage.

19.13.6 DESCRIPTION

The subprogram FDLINT accepts a nonrectangular field table as input and returns the x-intercepts for the given scan line y.

19.13.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.13.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.14 FIND12

The FIND12 function scans card input and locates special symbols.

19.14.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	GRPSCN, REDIF2, and SETUP2
DAMRG	SET18
DATA-TR	SETUP8
DISPLAY	REDIF3 and SETUP3
DOTDATA	SET13
GRAYMAP	SETUP6
GTDDM	SET19
GTTCN	SET17
HIST	SETUP5
ISOCLS	SETUP7
LABEL	CRDSCN and SET14
NDHIST	SET10
SCTRPL	SET11 and VECSCN
SELECT	GRPSCN, SETUP4, and WGTSCN
STAT	SETUP1
TESTSP	SETUP7
TRSTAT	SETUP9

19.14.2 INTERFACES

The FIND12 subprogram interfaces with other routines through the calling arguments.

19.14.3 INPUTS

Calling sequence: FIND12(CARD,COL,VECTOR)

Parameter	Dimension	In/out	Definition
CARD	1	In	BCD buffer.
COL	1	In/out	Pointer to position in card.
VECTOR	1	In	Vector of N symbols to be located in CARD [N is given in VECTOR(1)].

19.14.4 OUTPUTS

The results are returned for use by the calling routine.

19.14.5 STORAGE REQUIREMENTS

This subprogram requires 548 bytes of storage.

19.14.6 DESCRIPTION

The input card is scanned for any of the symbols in the given array.

19.14.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.14.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.15 FLDINT

The FLDINT subprogram unpacks pixels from specified lines on the MSS DATAPE.

19.15.1 LINKAGES

This routine calls the CMERR subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	CLSFY2
DAMRG	DAMRG
DATA-TR	LNTRAN and TRHIST
DOTDATA	DOTS
GRAYMAP	HISTGM and PICT
GTDDM	DDM
GTTCN	TCN
HIST	HISTGM
IBOCLS	RDDATA
LABEL	STOMAP
NDHIST	NDHIST1 and STODAT
STAT	LEARN
TESTSP	RDDPAT

19.15.2 INTERFACES

The FLDINT subprogram interfaces with other routines through common block TAPERD and through the calling arguments.

19.15.3 INPUTS

Calling sequence: CALL FLDINT(BLOCK,FETVEC,NOFEAT)

Parameter	Dimension	In/out	Definition
BLOCK	6	In	Rectangular field description used in unpacking pixels from MSS DATAPE.
FETVEC	30	In	Array containing channels.
NOFEAT	1	In	Number of channels.

19.15.4 OUTPUTS

The results are returned for use by the calling routine.

19.15.5 STORAGE REQUIREMENTS

This subprogram requires 2586 bytes of storage.

19.15.6 DESCRIPTION

The FLDINT subprogram positions the input tape at the correct scan line for a specific field, establishes areas on the scan line to unpack, and unpacks data for each channel in FETVEC. If a requested input channel number is greater than the total number of channels on the input file, the subprogram exits by calling the CMERR subprogram.

19.15.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.15.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.16 FLTNUM

The FLTNUM subprogram is a function that scans cards, interprets real numbers separated by commas, and returns them in the array NUMVEC.

19.16.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
CLASSIFY	REDIF2
DATA-TR	SETUP8
DISPLAY	REDIF3
ISOCLS	SETUP7
LABEL	SET14
SCTRPL	SET11
SELECT	WGTSCN
TESTSP	SETUP7

19.16.2 INTERFACES

The FLTNUM subprogram interfaces with other routines through the calling arguments.

19.16.3 INPUTS

Calling sequence: FLTNUM(CARD,COL,NUMVEC,VECMAX)

Parameter	Dimension	In/out	Definition
CARD	62	In	62-column card buffer.

Parameter	Dimension	In/out	Definition
COL	1	In	Pointer to position in CARD.
NUMVEC	20	Out	Array in which to return the numbers.
VECMAX	1	Out	Length of NUMVEC.

19.16.4 OUTPUTS

This subprogram stores results in NUMVEC.

19.16.5 STORAGE REQUIREMENTS

This subprogram requires 1452 bytes of storage.

19.16.6 DESCRIPTION

The FLTNUM subprogram accepts integers input as card images, interprets them, and stops at the first nonnumeric character, except when the characters are used to denote a data statement format.

19.16.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.16.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.17 FSBSFL

The FSBSFL subprogram is a file-positioning routine for unformatted reads or writes.

19.17.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	RDMODK, REDSAV, and SETUP2
DATA-TR	RDMODK and REDSAV
DISPLAY	EMTHRS and SETUP3
DOTDATA	WRTDOT
ISOCLS	GETST, LABMAN, RDDOTS, and RDMODK
LABEL	DSPTAP, LABMAN, RDDOTS, RDMODK, REDSAV, and WRTDOT
SCTRPL	GETST, RDMODK, SET11, and STOFIL
SELECT	RDMODK and REDSAV
STAT	LEARN
TESTSP	GETST, LABMAN, RDDOTS, and RDMODK
TRSTAT	RDMODK, REDSAV, and TRAMTX

19.17.2 INTERFACES

The FSBSFL subprogram interfaces with other routines through the calling arguments.

19.17.3 INPUTS

Calling sequence: CALL FSBSFL(UNIT,FILE,ISTAT)

Parameter	Dimension	In/out	Definition
UNIT	1	In	Number of the Fortran unit on which the requested file is located.
FILE	1	In	Number of files to skip.
ISTAT	1	Out	If = 0, file position is correct; if = 2, incorrect.

19.17.4 OUTPUTS

Not applicable.

19.17.5 STORAGE REQUIREMENTS

This subprogram requires 552 bytes of storage.

19.17.6 DESCRIPTION

The FSBSFL subprogram moves forward on the tape until it reaches the requested file number. If the parameter FILE is negative, it generates an error message stating that the unit only moves forward. Control returns to the calling program when FILE $\leq$ 0 or when the unit is positioned on the requested file.

19.17.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.17.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.18 FSFMFL

The FSFMFL subprogram is a file-positioning routine for formatted reads or writes. (The formats of interest in the system are Universal and LARSYS III.)

19.18.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
DAMRG	DAMRG
DATA-TR	LNTRAN
DISPLAY	LISTSM
DOTDATA	LISTLC
GTDDM	DDM
GTTCN	TCN
LABEL	CLSMAP

19.18.2 INTERFACES

The FSFMFL subprogram interfaces with other routines through the calling arguments.

19.18.3 INPUTS

Calling sequence: CALL FSFMFL(UNIT,FILE,ISTAT)

Parameter	Dimension	In/out	Definition
UNIT	1	In	Number of the Fortran unit on which the requested file is located.
FILE	1	In	Number of files to skip.

Parameter	Dimension	In/out	Definition
ISTAT	1	Out	If = 0, file position is correct; if = 2, incorrect.

19.18.4 OUTPUTS

Not applicable.

19.18.5 STORAGE REQUIREMENTS

This subprogram requires 580 bytes of storage.

19.18.6 DESCRIPTION

The FSFMFL subprogram positions the tape on UNIT at FILE and returns the status in ISTAT.

19.18.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.18.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.19 GETINF

The GETINF subprogram retrieves stored information from ARRAY.

19.19.1 LINKAGES

The GETINF subprogram calls the NAMSTA subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	ISOCLS
TESTSP	TESTSP

19.19.2 INTERFACES

The GETINF subprogram interfaces with other routines through the calling arguments.

19.19.3 INPUTS

Calling sequence: CALL GETINF(ARRAY,FLDSAV,VERTEX,CLSNMS,NOSUBS, SUBNM,NOCLS,TOTSUB)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
FLDSAV	4,1	Out	Storage array for field information.
VERTEX	1	Out	Storage array for field vertices.
CLSNMS	1	Out	Storage array for class names.
NOSUBS	1	Out	Storage array for subclass numbers.
SUBNM	1	In/out	Storage array for subclass names.

Parameter	Dimension	In/out	Definition
NOCLS	1	In	Number of classes.
TOTSUB	1	In/out	Total number of subclasses.

19.19.4 OUTPUTS

The results are returned for use by the calling routine.

19.19.5 STORAGE REQUIREMENTS

This subprogram requires 1480 bytes of storage.

19.19.6 DESCRIPTION

Subprogram GETINF retrieves field information, vertices, class and subclass names, and subclass numbers from ARRAY and stores them in the specified output arrays. It calls the NAMSTA subprogram to assign names to clusters and update statistical information.

19.19.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.19.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.20 GETST

The GETST subprogram retrieves statistics from the SAVTAP file, rewrites them in a specified format, and stores them in the arrays MENS and STDEV.

19.20.1 LINKAGES

This routine calls the CMERR and FSBSFL subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	ISOCLS
SCTRPL	SCATTR
TESTSP	TESTSP

19.20.2 INTERFACES

The GETST subprogram interfaces with other routines through the calling arguments.

19.20.3 INPUTS

Input to the GETST subprogram consists of the SAVTAP file output by the STAT processor.

Calling sequence: CALL GETST(UNIT,FILE,MENS,STDEV,NOSUB2,SUBVEC, NOCHAN,CHNVEC,MEANS,COVAR,ITRIG)

Parameter	Dimension	In/out	Definition
UNIT	1	In	Number of the Fortran unit from which statistics are to be retrieved.

Parameter	Dimension	In/out	Definition
FILE	1	In	File number on UNIT from which statistics are to be retrieved.
MENS	1	Out	Array containing the mean vector for each subclass.
STDEV	1	Out	Array containing the subset of standard deviations for requested channels in each subclass.
NOSUB2	1	Out	Number of subclasses on the SAVTAP file.
SUBVEC	1	In	Vector of subclass names.
NOCHAN	1	In	Number of channels to be used in processing.
CHNVEC	1	In	Array containing actual channels requested from training segment.
MEANS	1	In	Vector of subclass means.
COVAR	465	In	Matrix of covariances.
ITRIG	1	In	If = 1, standard deviations will be returned along with the means.

19.20.4 OUTPUTS

This subprogram outputs the statistics in the arrays MENS and STDEV.

19.20.5 STORAGE REQUIREMENTS

This subprogram requires 3126 bytes of storage.

19.20.6 DESCRIPTION

The storage arrays passed to this subroutine for the means and standard deviations should be dimensioned singularly in the calling routine. The statistics are stored in the following manner.

```
MEANS(1)              — CHANNEL 1,       SUBCLASS 1
MEANS(2)              — CHANNEL 2,       SUBCLASS 1
MEANS(3)              — CHANNEL 3,       SUBCLASS 1
        ⋮                    ⋮                 ⋮
MEANS(NOCHAN)         — CHANNEL NOCHAN,  SUBCLASS 1
MEANS(NOCHAN+1)       — CHANNEL 1,       SUBCLASS 2
MEANS(NOCHAN+2)       — CHANNEL 2,       SUBCLASS 2
MEANS(NOCHAN+3)       — CHANNEL 3,       SUBCLASS 2
        ⋮                    ⋮                 ⋮
MEANS(NOCHAN*2)       — CHANNEL NOCHAN,  SUBCLASS 2
        ⋮                    ⋮                 ⋮
MEANS(NOCHAN*NOSUB)   — CHANNEL NOCHAN,  SUBCLASS NOSUB
```

The standard deviations (STDEV) are stored in the same manner.

19.20.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.20.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.21 GRPSCN

The GRPSCN subprogram is a function which scans training and/or test field definition card images and determines if classes or subclasses have been assigned to groups.

19.21.1 LINKAGES

This routine calls the FIND12, NUMBER, and NXTCHR subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
CLASSIFY	REDIF2
SELECT	SETUP4

19.21.2 INTERFACES

The GRPSCN function interfaces with other routines through common block INFORM and through the calling arguments.

19.21.3 INPUTS

Calling sequence: GRPSCN(CARD,NNCLAS,GRPTR)

Parameter	Dimension	In/out	Definition
CARD	62	In	62-column card buffer.
NNCLAS	1	In	Maximum number of classes to allow.
GRPTR	1	Out	Pointer to groups.

19.21.4 OUTPUTS

The results are returned for use by the calling routine.

19.21.5 STORAGE REQUIREMENTS

This subprogram requires 1644 bytes of storage.

19.21.6 DESCRIPTION

Field definition cards are input and scanned, and an error message is generated if the end of a group is reached and a class is not found.

19.21.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.21.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.22 HISTGM

The HISTGM subprogram calculates histograms and writes total histogrammed statistics on the HISFIL for use by the GRAYMAP processor.

19.22.1 LINKAGES

This routine calls the CLSHIS, FDLINT, FLDINT, HISTIC, LAREAD, LINERD, and TAPHDR subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
GRAYMAP	GRAYMP
HIST	HIST

19.22.2 INTERFACES

The HISTGM subprogram interfaces with other routines through common blocks GLOBAL, GRCBLK, and HISTOR and through the calling arguments.

19.22.3 INPUTS

Calling sequence: CALL HISTGM(FILHIS,FLDTAL,TOTTAL)

Parameter	Dimension	In/out	Definition
FILHIS	NOFEAT,256	In	Array of fields to be histogrammed.
FLDTAL	NOHIST,XSIZ	In	Number of field to be histogrammed.
TOTTAL	NOHIST,XSIZ	Out	Array containing total histogrammed statistics.

19.22.4 OUTPUTS

This subprogram outputs field and histogram statistics on the line printer, histogram statistics on tape, and histograms on the pen plotter.

19.22.5 STORAGE REQUIREMENTS

This subprogram requires 57 224 bytes of storage.

19.22.6 DESCRIPTION

The HISTGM subprogram accepts data from the MSS DATAPE (via subprogram TAPHDR) and field definition cards. It stores field information, zeroes out part of the field histogram array, scales factors for plotting, and prints field statistics. It calls FLDINT to unpack pixels, LINERD to read each scan line of data, FDLINT to return pixel numbers, and HISTIC to compute and display statistics for the histogram. It writes each scan line of data on tape and plots a histogram for each field. After all data are read in, it writes the total histogram on tape, prints total statistics, and plots the total histogram.

19.22.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.22.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.23 HISTIC

The HISTIC subprogram computes and displays statistics for the histogram.

19.23.1 LINKAGES

This routine does not call any other subprogram. The subprogram which calls it, along with the processors, is listed below.

Processor	Calling subprogram
GRAYMAP	HISTGM
HIST	HISTGM

19.23.2 INTERFACES

The HISTIC subprogram interfaces with other routines through common blocks GLOBAL and GRCBLK and through the calling arguments.

19.23.3 INPUTS

Calling sequence: CALL HISTIC(IHG,NI,IFLD,VERTCS,NC)

Parameter	Dimension	In/out	Definition
IHG	NOFEAT,256	In	Array containing data to be histogrammed.
NI	1	In	Print flag; = -1 suppresses part of printer output.
IFLD	50,24	In	Array containing field information.
VERTCS	1	In	Array containing field vertices.
NC	1	In	Number of channels.

19.23.4 OUTPUTS

The results are returned for use by the calling routine.

19.23.5 STORAGE REQUIREMENTS

This subprogram requires 3814 bytes of storage.

19.23.6 DESCRIPTION

The HISTIC subprogram computes data ranges, means, standard deviations, and normalized ranges for plotting histograms. This information is returned to the HISTGM subprogram, which plots histograms for the HIST and GRAPMAP processors.

19.23.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.23.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.24 I4A1BN

The I4A1BN subprogram converts an array of EBCDIC digits to internal binary integers.

19.24.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	LAREAD and NUMBER
DAMRG	LAREAD and NUMBER
DATA-TR	LAREAD and NUMBER
DISPLAY	LAREAD and NUMBER
DOTDATA	LAREAD, NUMBER, and NUMBR
GRAYMAP	LAREAD and NUMBER
GTDDM	NUMBER
GTTCN	NUMBER
HIST	LAREAD and NUMBER
ISOCLS	LAREAD and NUMBER
LABEL	LAREAD and NUMBER
NDHIST	LAREAD and NUMBER
SCTRPL	NUMBER and VECSCN
SELECT	NUMBER
STAT	LAREAD and NUMBER
TESTSP	LAREAD and NUMBER

19.24.2 INTERFACES

The I4A1BN subprogram interfaces with other routines through the calling arguments.

19.24.3 INPUTS

Calling sequence: CALL I4A1BN(IFLD,NCHFLD,NCVTED)

Parameter	Dimension	In/out	Definition
IFLD	20	In	First word of an array of EBCDIC characters.
NCHFLD	1	In	Number of characters in the field.
NCVTED	1	Out	One-word result in binary.

19.24.4 OUTPUTS

The results are returned for use by the calling routine.

19.24.5 STORAGE REQUIREMENTS

This subprogram requires 1384 bytes of storage.

19.24.6 DESCRIPTION

This subprogram accepts an array of EBCDIC characters, looks for the first nonblank character, and determines if it is a digit or other character specified by the calling routine. It sets a minus flag for negative digits and converts digits to internal binary. If any character other than a blank or a digit is encountered, an error message is generated.

19.24.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.24.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.25 LABMAN

Subprogram LABMAN writes a statistics tape (SAVTAP) in the Universal or LARSYS II format and, optionally, creates the module STAT file.

19.25.1 LINKAGES

This routine calls the FSBSFL, RREAD, and WRTMTX subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	ISOCLS
LABEL	LABLR
TESTSP	TESTSP

19.25.2 INTERFACES

The LABMAN subprogram interfaces with other routines through the calling arguments.

19.25.3 INPUTS

Input to the LABMAN subprogram consists of the SAVTAP file output by the ISOCLS, LABEL, STAT, or TRSTAT processor.

Calling sequence: CALL LABMAN(UNIT,FILE,NOCLS,TOTSUB,NOFEAT, TOTFLD,TOTVRT,FETVEC,FLDSAV,VERTEX,CLSNMS,NOSUBS,SUBNM,N,STADRS, VARSIZ,PUNCH,SUBVEC,PRNSTS,SWTCH)

Parameter	Dimension	In/out	Definition
UNIT	1	In	Fortran unit number of tape on which file is to be written.
FILE	1	In	File number on UNIT for this output file.

Parameter	Dimension	In/out	Definition
NOCLS	1	In	Number of classes to be processed.
TOTSUB	1	In	Number of subclasses for all channels.
NOFEAT	1	In	Number of features (channels).
TOTFLD	1	In	Number of training fields.
TOTVRT	1	In	Number of vertices for all training fields.
FETVEC	NOFEAT	In	Vector containing channel numbers for which the statistics were computed.
FLDSAV	4,TOTFLD	In	Training field information (name of field, numbers of subclass and class to which the field belongs, and number of vertices in the fields, including the closure point).
VERTEX	TOTVRT	In	Array containing vertices from all training fields stored by sample and line number for each vertex.
CLSNMS	NOCLS	In	Array of class names.
NOSUBS	NOCLS	In	Number of clusters in each subclass.
SUBNM	TOTSUB	In	Array of reordered subclass names.
N	TOTSUB	In	Array of reordered pixel numbers in each subclass.
STADRS	1	In	Starting address for disk storage access.
VARSIZ	1	In	Storage size for lower triangular part of covariance matrix.

Parameter	Dimension	In/out	Definition
PUNCH	1	In	Request to create card image file.
SUBVEC	1	In	Array of reordered subclasses by category.
PRNSTS	1	In	Request to print statistics summary.
SWTCH	1	In	= 1, program calculates the mean using statistics from the ISOCLS processor; = 2, statistics are from the LABEL processor; = 3, statistics are from the STAT processor; and, = 4, statistics are from the TRSTAT processor.

19.25.4 OUTPUTS

An unformatted statistics tape (SAVTAP) is output. Optionally, a card image file and/or a line printer summary is produced.

19.25.5 STORAGE REQUIREMENTS

This subprogram requires 6594 bytes of storage.

19.25.6 DESCRIPTION

The LABMAN subprogram accepts training field information and statistics from the ISOCLS, LABEL, STAT, or TRSTAT processor. It reads means and covariances into core from disk storage. Statistics are rewritten to reflect manual relabeling of clusters and are output on the SAVTAP file; optionally, a card image file and/or line printer summary is produced.

19.25.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.25.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.26 LAREAD

The LAREAD subprogram is a function that reads field definition card images.

19.26.1 LINKAGES

This routine calls the CMERR, I4A1BN, and NXTCHR subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	CLSFY2 and SETUP2
DAMRG	SET18
DATA-TR	LNTRAN and TRHIST
DISPLAY	REDIF3
DOTDATA	FLDTYP
GRAYMAP	HISTGM and PICT
HIST	HISTGM
ISOCLS	RDDATA
LABEL	FILERD
NDHIST	FLDCLS, FLDFLD, and FLDSUB
STAT	LEARN
TESTSP	RDDPAT

19.26.2 INTERFACES

The LAREAD subprogram interfaces with other routines through the calling arguments.

19.26.3 INPUTS

Calling sequence: LAREAD(FLDNAM,VERTCS,FLDINF,NC)

Parameter	Dimension	In/out	Definition
FLDNAM	1	In/out	Alphanumeric designation for the field name.
VERTCS	2,11	In/out	Array of field vertices.
FLDINF	6	Out	Array of field information (name, class).
NC	1	In/out	Number of cards input ($\leq$10).

19.26.4 OUTPUTS

The results are returned for use by the calling routine.

19.26.5 STORAGE REQUIREMENTS

This subprogram requires 3222 bytes of storage.

19.26.6 DESCRIPTION

LAREAD reads the name and vertex line and sample numbers from field definition card images. Rectangular field coordinates are determined, and vertices are arranged in ascending clockwise order (smallest sample first).

19.26.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.26.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.27 LINERD

The LINERD subprogram reads one scan line of data from the MSS DATAPE.

19.27.1 LINKAGES

This routine calls the BUFILL and SEARCH subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	CLSFY2
DAMRG	DAMRG
DATA-TR	LNTRAN and TRHIST
DOTDATA	DOTS
GRAYMAP	HISTGM and PLOT
GTDDM	DDM
GTTCN	TCN
HIST	HISTGM
ISOCLS	RDDATA
LABEL	STOMAP
NDHIST	NDHST1 and STODAT
STAT	LEARN
TESTSP	RDDPAT

19.27.2 INTERFACES

The LINERD subprogram interfaces with other routines through common blocks BUFF and TAPERD and through the calling arguments.

19.27.3 INPUTS

Calling sequence: CALL LINERD(IDATA,ENDTAP)

Parameter	Dimension	In/out	Definition
IDATA	-	Out	Storage array for unpacked data. (Dimension depends upon buffer size in calling routine.)
ENDTAP	1	Out	End of tape: If = -1, an EOF was encountered.

19.27.4 OUTPUTS

The results are returned for use by the calling routine.

19.27.5 STORAGE REQUIREMENTS

This subprogram requires 15 768 bytes of storage.

19.27.6 DESCRIPTION

The information to be unpacked from the Universal-formatted MSS DATAPE occupies the following byte positions on the tape header record:

Byte numbers	Description
109-110	Pixel start number
111-112	Pixel stop number
1789-1790	Pixel skip factor
1791-1792	Line skip factor
2201-2202	Sun angle for pass 1 (channels 1-4)
2203-2204	Sun angle for pass 2 (channels 5-8)
⋮	⋮
2215	Sun angle for pass 8 (channels 29-30)

The information just listed is placed in labeled common TAPERD.

19.27.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.27.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.28 LISTLC

The LISTLC subprogram reads the PPTC, GT, and AI dot data files for the DISPLAY and DOTDATA processors, as specified by control card images PPUN, GTUN, and AIUN. It computes line and sample increments and stores the dot data.

19.28.1 LINKAGES

The LISTLC subprogram calls the FSFMFL, NUMBR, and NXTCHR subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
DISPLAY	LISTSM
DOTDATA	DOTS

19.28.2 INTERFACES

The LISTLC subprogram interfaces with other routines through common blocks DOTVEC and INFORM and through the calling arguments.

19.28.3 INPUTS

Input to the LISTLC subprogram consists of the PPTC, GT, and/or AI dot data files (see section 12.13 for a description of these files).

Calling sequence: CALL LISTLC(FIELDS,STAMNT,*,*,*,SWCHG,INIT, IUNIT,IFILE,IPT,VERTEX)

Parameter	Dimension	In/out	Definition
FIELDS	4,1	Out	Array containing dots indexed by NOFLD2; (1,b) = category name; (4,b) = 2 (number of vertices).

Parameter	Dimension	In/out	Definition
STAMNT	1	In/out	Switch to indicate dots are taken from currently read card; initially set = 1 but reset = 2 if DOT card has been read and is being processed.
*	1	Out	Exit route 1 (after all processing).
*	1	Out	Exit route 2 (if SWCHG $\leq$ 1).
*	1	Out	Exit route 3 (if a card is improperly formatted).
SWCHG	1	In/out	Counter for number of times dot type changes; initially set = 1.
INIT	1	In/out	Set = 0 initially; reset = 1 after dot data files have been read in.
IUNIT	1	In	Unit number of input dot data.
IFILE	1	In	Relative file number of input dot data.
IPT	1	In/out	Pointer in VERTEX (incremented by 4).
VERTEX	1	Out	Array containing dot vertices (sample and line numbers repeated).

The field definition card images relevant to this routine are given in section 17.5.4 of volume II of this user guide.

19.28.4 OUTPUTS

The results are stored for use by the calling routine.

19.28.5 STORAGE REQUIREMENTS

This subprogram requires 3026 bytes of storage.

19.28.6 DESCRIPTION

Not required.

19.28.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.28.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.29 MATVEC

The MATVEC subprogram multiplies matrix A by vector B and stores the result in vector C.

19.29.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
DATA-TR	KBTRAN
TRSTAT	TRAMTX

19.29.2 INTERFACES

The MATVEC subprogram interfaces with other routines through the calling arguments.

19.29.3 INPUTS

Calling sequence: CALL MATVEC(A,B,C,L,M)

Parameter	Dimension	In/out	Definition
A	L,M	In	Matrix.
B	M	In	Vector.
C	L	Out	Storage array for product A × B.
L	1	In	Number of rows in A and C.
M	1	In	Number of columns in A and B.

19.29.4 OUTPUTS

The results are returned for use by the calling routine.

19.29.5 STORAGE REQUIREMENTS

This subprogram uses 588 bytes of storage.

19.29.6 DESCRIPTION

A matrix A and a vector B are input via calling argument. The product of A times B is stored in vector C.

19.29.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.29.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.30 MTMDAT

The MTMDAT subprogram multiplies matrix A by the transpose of matrix B and stores the result in a lower triangular matrix DD.

19.30.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
DATA-TR	KBTRAN
TRSTAT	TRAMTX

19.30.2 INTERFACES

The MTMDAT subprogram interfaces with other routines through the calling arguments.

19.30.3 INPUTS

Calling sequence: CALL MTMDAT(A,B,C,L,M,N,D,DD)

Parameter	Dimension	In/out	Definition
A	L,M	In	Matrix.
B	N,M	In	Transpose matrix.
C	L,N	Out	Product of A × B.
L	1	In	Number of rows in A, C, and D
M	1	In	Number of columns in A and B.
N	1	In	Number of rows in B and number of columns in C.
D	L	Out	Vector.

Parameter	Dimension	In/out	Definition
DD	1	Out	Array for storage of lower triangular matrix, product of A × B.

19.30.4 OUTPUTS

The results are returned for use by the calling routine.

19.30.5 STORAGE REQUIREMENTS

This subprogram requires 1020 bytes of storage.

19.30.6 DESCRIPTION

A matrix A and a transposed matrix B are input via calling argument. The program multiplies A times B to obtain a result C, which is stored in a lower triangular array DD.

19.30.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.30.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.31 MTMLS6

The MTMLS6 subprogram multiplies a matrix A by a vector B and stores the result in matrix C.

19.31.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
DATA-TR	KBTRAN
TRSTAT	TRAMTX

19.31.2 INTERFACES

The MTMLS6 subprogram interfaces with other routines through the calling arguments.

19.31.3 INPUTS

Calling sequence: CALL MTMLS6(A,B,C,M,N)

Parameter	Dimension	In/out	Definition
A	M,N	In	Matrix.
B	1	In	Vector stored in symmetric notation.
C	M,N	Out	Matrix for storage of product of A × B.
M	1	In	Number of rows in A and C.
N	1	In	Number of columns in A and C.

19.31.4 OUTPUTS

The results are returned for use by the calling routine.

19.31.5 STORAGE REQUIREMENTS

This subprogram requires 880 bytes of storage.

19.31.6 DESCRIPTION

Not required.

19.31.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.31.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.32 NAMSTA

Subprogram NAMSTA provides cluster names based on category names and sequence numbering. A check is made to ensure that each category is involved in the naming of at least one cluster. If not, that category is deleted from the total count of number of categories.

19.32.1 LINKAGES

This routine calls the BNI4A1 subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	GETINF
LABEL	LABLR
TESTSP	GETINF

19.32.2 INTERFACES

The NAMSTA subprogram interfaces with other routines through the calling arguments.

19.32.3 INPUTS

Calling sequence: CALL NAMSTA(SUBNAM,CATVEC,SUBNO,NOSUB2,CATNAM, NOCAT)

Parameter	Dimension	In/out	Definition
SUBNAM	60	Out	Array containing cluster (subclass) names.
CATVEC	60	In	Array containing category numbers of subclasses.
SUBNO	1	In	Array of subclass numbers within each class.

Parameter	Dimension	In/out	Definition
NOSUB2	1	In	Number of subclasses.
CATNAM	60	Out	Array containing category names.
NOCAT	1	In	Number of categories.

19.32.4 OUTPUTS

The results are stored for use by the calling routine.

19.32.5 STORAGE REQUIREMENTS

This subprogram requires 860 bytes of storage.

19.32.6 DESCRIPTION

NAMSTA assigns names to clusters using the first two characters of the category name and two digits. The cluster names are stored in the vector SUBNAM on the basis of the category assignment to each cluster. If a category has no associated clusters, SUBNO and NOCAT are adjusted to remove that category from subsequent consideration.

19.32.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.32.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.33 NUMBER

The NUMBER subprogram is a function that scans control cards looking for integer numbers.

19.33.1 LINKAGES

This routine calls the I4A1B subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	GRPSCN, REDIF2, and SETUP2
DAMRG	SET18
DATA-TR	SETUP8
DISPLAY	REDIF3 and SETUP3
DOTDATA	FLDTYP and SET13
GRAYMAP	SETUP6
GTDDM	SET19
GTTCN	SET17
HIST	SETUP5
ISOCLS	SETUP7
LABEL	CRDSCN and SET14
MONPAC	MSCAN
MONTOR	MSCAN
NDHIST	SET10
SCTRPL	SET11
SELECT	GRPSCN and SETUP4
STAT	SETUP1
TESTSP	SETUP7
TRSTAT	SETUP9

19.33.2 INTERFACES

The NUMBER subprogram interfaces with other routines through the calling arguments.

19.33.3 INPUTS

Calling sequence: NUMBER(CARD,COL,NUMVEC,NOW)

Parameter	Dimension	In/out	Definition
CARD	1	In	62-column card buffer.
COL	1	Out	Pointer to position in CARD.
NUMVEC	1	Out	Array in which to return the number.
NOW	1	In	Input column.

19.33.4 OUTPUTS

The results are returned for use by the calling routine.

19.33.5 STORAGE REQUIREMENTS

This subprogram requires 986 bytes of storage.

19.33.6 DESCRIPTION

Not required.

19.33.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.33.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.34 NUMBR

The NUMBR subprogram processes one field definition card image at a time. It reads and stores all numbers in array NDOTS, with NDCARD as an index.

19.34.1 LINKAGES

This routine calls the I4A1BN subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
DISPLAY	LISTLC
DOTDATA	FLDLAC

19.34.2 INTERFACES

The NUMBR subprogram interfaces with other routines through the calling arguments.

19.34.3 INPUTS

Calling sequence: CALL NUMBR(NDOTS,NDCARD,CARD,COL)

Parameter	Dimension	In/out	Definition
NDOTS	1	Out	Array containing numbers read from card images.
NDCARD	1	Out	Index to NDOTS.
CARD	1	In	Array containing input field definition card image.
COL	1	In	Index to CARD.

The field definition card images relevant to this routine are given in section 17.5.4 of volume II of this user guide.

19.34.4 OUTPUTS

The results are returned for use by the calling routine.

19.34.5 STORAGE REQUIREMENTS

This subprogram requires 792 bytes of storage.

19.34.6 DESCRIPTION

Not required.

19.34.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.34.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.35 NXTCHR

The NXTCHR subprogram is a function which locates and enters the next nonblank character in a card being read.

19.35.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	CATSCN, GRPSCN, LAREAD, and REDIF2
DAMRG	SET18
DATA-TR	LAREAD and SETUP8
DISPLAY	LAREAD, LISTLC, REDIF3, and SETUP3
DOTDATA	FLDLAC, LAREAD, LISTLC, and SET13
GRAYMAP	LAREAD and SETUP6
GTDDM	SET19
GTTCN	SET17
HIST	LAREAD and SETUP5
ISOCLS	LAREAD and SETUP7
LABEL	CRDSCN, LAREAD, and SET14
MONPAC	MSCAN
MONTOR	MSCAN
NDHIST	LAREAD and SET10
SCTRPL	SET11 and VECSCN
SELECT	GRPSCN, SETUP4, and WGTSCN
STAT	LAREAD and SETUP1

Processor	Calling subprograms
TESTSP	LAREAD and SETUP7
TRSTAT	SETUP9

19.35.2 INTERFACES

The NXTCHR subprogram interfaces with other routines through the calling arguments.

19.35.3 INPUTS

Calling sequence: NXTCHR(CARD,COL)

Parameter	Dimension	In/out	Definition
CARD	1	In	BCD buffer.
COL	1	In/out	Pointer to position in CARD.

19.35.4 OUTPUTS

The results are returned for use by the calling routine.

19.35.5 STORAGE REQUIREMENTS

This subprogram requires 510 bytes of storage.

19.35.6 DESCRIPTION

Not required.

19.35.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.35.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.36 ORDER

The ORDER subprogram arranges a set of N integers in ascending order.

19.36.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	CLSCHK and REDIF2
DATA-TR	CLSCHK and SETUP8
DOTDATA	SET13
ISOCLS	SETUP7
LABEL	CLSCHK and SET14
NDHIST	SET10
SCTRPL	SET11
SELECT	CLSCHK, DAVIDN, SELECT, SETUP4, and WHRPLC
TESTSP	SETUP7
TRSTAT	CLSCHK and SETUP9

19.36.2 INTERFACES

The ORDER subprogram interfaces with other routines through the calling arguments.

19.36.3 INPUTS

Calling sequence: CALL ORDER(VEC,N)

Parameter	Dimension	In/out	Definition
VEC	1	In/out	Vector of N integers.
N	1	In	Number of integers in VEC.

19.36.4 OUTPUTS

The results are returned for use by the calling routine.

19.36.5 STORAGE REQUIREMENTS

This subprogram requires 484 bytes of storage.

19.36.6 DESCRIPTION

The subprogram ORDER operates on an input vector of N unsorted integers by comparing and switching values in pairs until all integers are in logical ascending order. It returns the values in the array VEC.

19.36.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.36.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.37 PRINT

The PRINT subprogram generates most of the output for the ISOCLS and TESTSP processors.

19.37.1 LINKAGES

The PRINT subprogram calls the FDLINT, RREAD, and SETMRG subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
ISOCLS	ISOCLS and ISODAT
TESTSP	TESTSP and ISOPAT

19.37.2 INTERFACES

The PRINT subprogram interfaces with other routines through common blocks GLOBAL and PASS and through the calling arguments.

19.37.3 INPUTS

Calling sequence: CALL PRINT(KKT,IPLACE,MEANS,STDEV,CLD, FLDINF,N)

Parameter	Dimension	In/out	Definition
KKT	1	In	Number of iterations.
IPLACE	NOPTS	In	Vector of cluster numbers to which the corresponding data points are assigned.
MEANS	NOFEAT,MAXCLS	In	Array containing means of each feature/cluster.
STDEV	NOFEAT,MAXCLS	In	Array containing standard deviations for each feature/ cluster.

Parameter	Dimension	In/out	Definition
CLD	MAXCLS,1	In	Array containing distances between clusters.
FLDINF	1	In	Array containing field information.
N	MAXCLS	In	Number of pixels per cluster.

19.37.4 OUTPUTS

This subprogram outputs the information in section 19.37.6 on the specified unit.

19.37.5 STORAGE REQUIREMENTS

This subprogram requires 6568 bytes of storage.

19.37.6 DESCRIPTION

The PRINT subprogram prints the following information each time it is called: header record, total clusters, total data points, summary of points in each cluster, means, standard deviations, and distances between clusters.

If requested by the user, the following information is printed for each field: header record, field name, total points in the field, field boundaries (line and sample numbers), the symbol for each pixel in the field, and a summary of points for each cluster in the field.

19.37.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.37.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.38 PRTCOV

The PRTCOV subprogram writes the transformed covariance matrix.

19.38.1 LINKAGES

This routine calls the WRTMTX subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
DATA-TR	KBTRAN and SETUP8
TRSTAT	SETUP9 and TRAMTX

19.38.2 INTERFACES

The PRTCOV subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and TRBLCK and through the calling arguments.

19.38.3 INPUTS

Calling sequence: CALL PRTCOV(COVMTX,AVEMTX,CV1,AV1,CLSMTX)

Parameter	Dimension	In/out	Definition
COVMTX	CV1,NOSUB2	In	Array of covariance matrices for NOSUB2 training subclasses.
AVEMTX	AV1,NOSUB2	In	Array of subclass means.
CV1	1	In	Number of rows in COVMTX.
AV1	1	In	Number of rows in AVEMTX.
CLSMTX	NOSUB2	In	Array of training class names.

19.38.4 OUTPUTS

This subprogram outputs the transformed covariance matrix on the specified unit.

19.38.5 STORAGE REQUIREMENTS

This subprogram requires 1032 bytes of storage.

19.38.6 DESCRIPTION

The subprogram PRTCOV writes the heading for the transformed covariance matrix and calls subprogram WRTMTX to write the matrix.

19.38.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.38.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.39 RANK

The RANK subprogram calculates the greenness value for each cluster and outputs an ordered list of color keys, cluster numbers, and greenness values.

19.39.1 LINKAGES

The RANK subprogram does not call any other subprogram. The subprogram which calls it, along with the processors, is listed below.

Processor	Calling subprogram
ISOCLS	DSTAPE
TESTSP	DSTAPE

19.39.2 INTERFACES

The RANK subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

19.39.3 INPUTS

Calling sequence: CALL RANK(NOFEAT,FETVC2,LNCAT,MEANS,IPTT)

Parameter	Dimension	In/out	Definition
NOFEAT	1	In	Number of features (channels).
FETVC2	28	In	Array of channels; not used.
LNCAT	1	In	Number of clusters.
MEANS	NOFEAT,LNCAT	In	Array of cluster means.
IPTT	LNCAT	In/out	Array of clustered data.

19.39.4 OUTPUTS

This subprogram outputs results on the line printer.

19.39.5 STORAGE REQUIREMENTS

This subprogram requires 1712 bytes of storage.

19.39.6 DESCRIPTION

The RANK subprogram checks to see if the number of input channels is a multiple of 4; if not, it prints out color keys ordered by cluster number. If channels are a multiple of 4, RANK sorts data according to the average greenness of each cluster over all acquisitions and prints the color keys, cluster numbers, and greenness values.

19.39.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.39.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.40 RDDOTS

The RDDOTS subprogram reads information from the DOTUNT file.

19.40.1 LINKAGES

This routine calls the CMERR, FSBSFL, and RDDOT1 subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	ISOCLS
LABEL	FILERD
TESTSP	TESTSP

19.40.2 INTERFACES

The RDDOTS subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

19.40.3 INPUTS

RDDOTS accepts a multifile unformatted Fortran-created tape in the format defined in appendix H of the "As-Built" Design Specification for EOD-LARSYS Procedure 1 (JSC-13143, LEC-11293, October 1977).

Calling sequence: CALL RDDOTS(DOTS,DOTVEC,TOTDT3,TYPSWT,SIZES, TOTDT2,NOCAT,CATNAM,NOFET2,FETVC2,NOFEAT,FETVEC,NOSUN,ANGLE, NOFLD,TOTVRT,FLDSAV,VERTEX,KVAR)

Parameter	Dimension	In/out	Definition
DOTS	SIZES,1	Out	Spectral and/or spatial dot information, depending upon option value of TYPSWT.

Parameter	Dimension	In/out	Definition
DOTVEC	1	In	Dot vector, depending on value of TYPSWT; if TYPSWT = 1, list of dots for which spectral information is wanted. = 2, ignored, except for a check if TOTDT3 ± 0. = 3, list of dots to be excluded; all other dots on the DOTFIL are passed back to the calling program.
TOTDT3	1	In	Total number of resolution elements for which data are requested, depending on value of TYPSWT; if TYPSWT = 1, number of dots requested. = 2, should be preset to 0. = 3, number of dots to be excluded.
TYPSWT	1	In	Option switch; must be preset to a value of 1, 2, or 3: = 1, spectral information requested. = 2, spatial information requested (sample, line, type, and category numbers). = 3, both spatial and spectral information requested.

Parameter	Dimension	In/out	Definition
SIZES	1	Out	Number of rows of dots in DOTS and KVAR arrays, depending on value of TYPSWT; if TYPSWT = 1, set to NOFET2. = 2, set to 4. = 3, set to 4 + NOFET2.
TOTDT2	1	Out	Total number of resolution elements, depending on value of TYPSWT; if TYPSWT = 1, set to TOTDT3. = 2, set to TOTDOT (number of dots on DOTFIL). = 3, set to TOTDOT - TOTDT3.
NOCAT	1	Out	Number of categories, value extracted from DOTFIL.
CATNAM	1	Out	Array of category names, values extracted from DOTFIL.
NOFET2	1	In/out	Number of channels for which information is requested. If set to 0, reset to NOFEAT. If TYPSWT = 2, the value of NOFET2 is ignored.
FETVC2	1	In	Array of channel numbers for which information is requested; ignored if TYPSWT = 2.
NOFEAT	1	In	Number of channels used in processing.
FETVEC	30	In	Array containing channel numbers used in processing.

Parameter	Dimension	In/out	Definition
NOSUN	1	Out	Number of Sun angles on DOTFIL.
ANGLE	NOSUN	Out	Array of Sun-angle values on DOTFIL.
NOFLD	1	Out	Number of fields on DOTFIL.
TOTVRT	1	Out	Number of vertices on DOTFIL.
FLDSAV	4,1	Out	Array of field information on DOTFIL.
VERTEX	2,1	Out	Array containing field vertices on DOTFIL.
KVAR	SIZES,TOTDT2	Out	Spectral dot information in floating point; set if TYPSWT = 1.

19.40.4 OUTPUTS

The results are returned for use by the calling routine.

19.40.5 STORAGE REQUIREMENTS

This subprogram requires 22 836 bytes of storage.

19.40.6 DESCRIPTION

The dot data tape is read and appropriate information is extracted. There are three modes of extraction:

a. Dot spectral information

b. Dot spatial information

c. Both spectral and spatial information

In all cases, the count information (first record) is made available.

Each file on the dot data tape consists of three records. Please refer to the flow chart provided in section 19.61 for details concerning contents of these records and the file description in appendix H of the "As-Built" Design Specification for EOD-LARSYS Procedure 1 (JSC-13143, LEC-11293, October 1977).

Dot spectral information consists of radiance values for specified dots and channels. Dot spatial information consists of sample number, line number, type number (1 for label/starting dots, 2 for bias dots), and category number for those dots specified by the user.

RDDOTS extracts data for three processors — LABEL, ISOCLS, and TESTSP. Since ISOCLS and TESTSP need different portions of the dot data, a switch passed via a calling argument controls which portions of the file to return to the calling routine.

19.40.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.40.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.41 RDDOT1

The RDDOT1 subprogram retrieves spectral and/or spatial information from the DOTFIL.

19.41.1 LINKAGES

This routine does not call any other subprogram. The subprogram which calls it, along with the processors, is listed below.

Processor	Calling subprogram
ISOCLS	RDDOTS
LABEL	RDDOTS
TESTSP	RDDOTS

19.41.2 INTERFACES

The RDDOT1 subprogram interfaces with other routines through the calling arguments.

19.41.3 INPUTS

Calling sequence: CALL RDDOT1(TEMDOT,DOTS,KVAR,SIZES,TOTDT2, DOTVEC,FETVC3,SIZE,TOTDOT,TOTDT3,NOFET2,TYPSWT)

Parameter	Dimension	In/out	Definition
TEMDOT	5000	In	Temporary location for dot data during subprogram execution.
DOTS	SIZES,1	Out	Spectral and/or spatial dot information, depending on option value of TYPSWT.
KVAR	SIZES,1	Out	Dot spectral information in floating point; set if TYPSWT = 1.

Parameter	Dimension	In/out	Definition
SIZES	1	In	Number of rows of dots in DOTS and KVAR arrays; depends on value of TYPSWT.
TOTDT2	1	In	Total number of resolution elements, depending on value of TYPSWT; if TYPSWT = 1, set to TOTDT3, spectral data. = 2, set to TOTDOT, spatial data. = 3, set to TOTDOT - TOTDT3, spectral and spatial data.
DOTVEC	1	In	Dot vector, depending on value of TYPSWT; if TYPSWT = 1, dots for which spectral data are requested. = 2, ignored. = 3, dots to be excluded; remaining dots are passed back to the calling routine.
FETVC3	1	In	Channel numbers for which information is requested; ignored if TYPSWT = 2.
SIZE	1	In	Used to calculate current index to TEMDOT.
TOTDOT	1	In	Total number of resolution elements for spectral and spatial data retrieval.

Parameter	Dimension	In/out	Definition
TOTDT3	1	In	Total number of resolution elements for which data are requested, depending on value of TYPSWT; if TYPSWT = 1, number of dots requested. = 2, preset to zero. = 3, number of dots to be excluded.
NOFET2	1	In	Number of channels for which data are requested; if 0, reset to NOFEAT; ignored if TYPSWT = 2.
TYPSWT	1	In	Option switch: = 1, spectral information requested. = 2, spatial information (sample, line, type, and category numbers) requested. = 3, both spectral and spatial information requested.

19.41.4 OUTPUTS

The results are returned for use by the calling routine.

19.41.5 STORAGE REQUIREMENTS

This subprogram requires 1440 bytes of storage.

19.41.6 DESCRIPTION

According to the value of the parameter TYPSWT, RDDOT1 retrieves the following information from the DOTFIL (which has been read in by the calling routine, RDDOTS).

a. If TYPSWT = 1, spectral (radiance) values for the specified dots and channels are extracted.

b. If TYPSWT = 2, spatial (sample, line, type, and category) numbers are extracted.

c. If TYPSWT = 3, both spectral and spatial data are extracted.

19.41.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.41.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.42 RDMEAN

The RDMEAN subprogram reads the MEAN card image file for the ISOCLS and TESTSP processors. Another entry point, RDFILE, reads the card file, stores only requested channels, and prints a summary of means for requested channels.

19.42.1 LINKAGES

The RDMEAN subprogram calls the RREAD and RWRITE subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
ISOCLS	ISOCLS and SETUP7
TESTSP	SETUP7 and TESTSP

19.42.2 INTERFACES

The RDMEAN subprogram interfaces with other routines through common blocks GLOBAL, PASS, and PASSA and through the calling arguments.

19.42.3 INPUTS

Calling sequences:

a. CALL RDMEAN(MENS)

Parameter	Dimension	In/out	Definition
MENS	30,1	In	Array of cluster means to be read in.

b. ENTRY RDFILE(MEANS,MENS)

Parameter	Dimension	In/out	Definition
MEANS	NOFEAT,MAXCLS	Out	Array containing means of each feature for each cluster.

19.42.4 OUTPUTS

This subprogram outputs initial cluster means on the specified unit.

19.42.5 STORAGE REQUIREMENTS

This subprogram requires 2112 bytes of storage.

19.42.6 DESCRIPTION

The RDMEAN subprogram goes to the beginning disk address where card images are stored and calls RWRITE to extract unformatted image data, place them in the buffer, and write the entire record.

Entry RDFILE reads the entire record, stores only the means for user-requested channels, and prints initial cluster centers for these channels on the specified unit. If the means for a specified channel are not on file, a message to that effect is printed and dummy values are used.

19.42.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.42.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.43 RDMODK

The RDMODK subprogram reads in the remainder of the module STAT file and writes the following statistics on the SAVTAP file: means, covariances, class and subclass descriptions, number of subclasses in each class, field information, and vertices.

19.43.1 LINKAGES

This routine calls the FSBSFL subprogram. The subprogram which calls it, along with the processors, is listed below.

Processor	Calling subprogram
CLASSIFY	CRDSTA
DATA-TR	CRDSTA
ISOCLS	CRDSTA
LABEL	CRDSTA
SCTRPL	CRDSTA
SELECT	CRDSTA
TESTSP	CRDSTA
TRSTAT	CRDSTA

19.43.2 INTERFACES

The RDMODK subprogram interfaces with other routines through common blocks GLOBAL, INFORM, and PASSB and through the calling arguments.

19.43.3 INPUTS

Input to the RDMODK subprogram consists of the module STAT file output by the STAT processor.

Calling sequence: CALL RDMODK(AVAR,COVAR,CLSDES,SUBNO,SUBDES, FLDSAV,VERTEX,ARRAY)

Parameter	Dimension	In/out	Definition
AVAR	NOFEAT	In	Array containing channel means.
COVAR	VARSIZ	In	Matrix of covariances.
CLSDES	NOCLS	In	Array containing class descriptions.
SUBNO	NOCLS	In	Array containing subclass numbers within each class.
SUBDES	NOSUB	In	Array containing subclass descriptions.
FLDSAV	4,NOFLD	In	Array of field information.
VERTEX	2,TOTVRT	In	Array containing field vertices.
ARRAY	1	In	Variably dimensioned working storage.

19.43.4 OUTPUTS

This subprogram outputs results on the SAVTAP file.

19.43.5 STORAGE REQUIREMENTS

This subprogram requires 2328 bytes of storage.

19.43.6 DESCRIPTION

The RDMODK subprogram calls FSBSFL to position the SAVTAP to the desired file. If the correct file is not found, it generates an error message and reads the module STAT deck to check for input errors. If the file is found, the field information and vertices are written on SAVTAP in the Universal or LARSYS II format.

19.43.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.43.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.44 REDDAT

Subprogram REDDAT reads covariances and means from the SAVTAP file and reduces statistics.

19.44.1 LINKAGES

This routine does not call any other subprogram. The subprogram which calls it, along with the processors, is listed below.

Processor	Calling subprogram
CLASSIFY	REDSAV
DATA-TR	REDSAV
LABEL	REDSAV
SELECT	REDSAV
TRSTAT	REDSAV

19.44.2 INTERFACES

The REDDAT subprogram interfaces with other routines through common blocks BESTKN, GLOBAL, and INFORM and through the calling arguments.

19.44.3 INPUTS

Calling sequence: CALL REDDAT(COVAR,AVAR,CLSDES,SUBNO,SUBDES, FLDSAV,VERTEX,COV,AVEN,CLSDS,SUBNOS,SUBDS,FLDSV,VERTX,NOFEAT, VARSIZ,NOCLS,NOFLD,NOSUB,FETVEC)

Parameter	Dimension	In/out	Definition
COVAR	VARSZ2,NOSUB2	In/out	Matrix of covariances.
AVAR	NOFEAT,NOSUB	In/out	Array containing channel means.
CLSDES	NOCLS	In	Array containing class descriptions.

Parameter	Dimension	In/out	Definition
SUBNO	NOCLS	In	Array containing subclass numbers within each class.
SUBDES	NOSUB	In	Array containing subclass descriptions.
FLDSAV	4,NOFLD	In	Array containing field information.
VERTEX	2,TOTVT2	In	Array containing field vertices.
COV	VARSIZ	In/out	Storage areas for output statistics.
AVEN	NOFET2,NOSUB2	Out	
CLSDS	NOCLS2	Out	
SUBNOS	NOCLS2	Out	
SUBDS	NOSUB2	Out	
FLDSV	4,NOFLD2	Out	
VERTX	2,TOTVT2	Out	
NOFEAT	1	In	Number of channels.
VARSIZ	1	In	Storage size for lower triangular part of covariance matrix.
NOCLS	1	In	Number of classes.
NOFLD	1	In	Number of fields.
NOSUB	1	In	Number of subclasses.
FETVEC	30	In	Array containing channel numbers used in processing.

19.44.4 OUTPUTS

The results are returned for use by the calling routine.

19.44.5 STORAGE REQUIREMENTS

This subprogram requires 3946 bytes of storage.

19.44.6 DESCRIPTION

The REDDAT subprogram reads statistics from the SAVTAP file and

a. Reduces means and covariances.

b. Reduces class description and array containing number of subclasses.

c. Reduces subclass descriptions, field information, and vertices.

d. Zeroes out a portion of covariance array containing subclasses that have been grouped.

e. Checks classification channels against training channels.

f. Reduces subclasses by channels.

g. Groups subclasses and means.

h. After grouping, stores the number of points per subclass.

19.44.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.44.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.45 REDSAV

The REDSAV subprogram reads the SAVTAP file (which was written by the STAT or ISOCLS processor or by the input of a module STAT file) and reduces the statistics to a user-requested subset.

19.45.1 LINKAGES

This routine calls the CLSCHK, CMERR, FSBSFL, REDDAT, and SAVFIL subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
CLASSIFY	SETUP2
DATA-TR	SETUP8
LABEL	FILERD
SELECT	SETUP4
TRSTAT	SETUP9

19.45.2 INTERFACES

The REDSAV subprogram interfaces with other routines through common blocks GLOBAL and INFORM and through the calling arguments.

19.45.3 INPUTS

Input to the REDSAV subprogram consists of the SAVTAP file output by the ISOCLS, STAT, or TESTSP processor.

Calling sequence: CALL REDSAV(ARRAY,TOP,BMFLG)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	Working storage array for statistics.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
BMFLG	1	In	Flag indicating input of the B-matrix.

19.45.4 OUTPUTS

This subprogram stores reduced statistics in ARRAY.

19.45.5 STORAGE REQUIREMENTS

This subprogram requires 12 170 bytes of storage.

19.45.6 DESCRIPTION

The REDSAV subprogram reads statistics from the SAVTAP file, computes bases and reduced bases, and calls the REDDAT subprogram for statistics reduction.

19.45.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.45.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.46 RREAD

The RREAD subprogram simulates the random read of a work file used to store data temporarily during execution of a processor.

19.46.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
DAMRG	DAMRG
DOTDATA	FLDLAC
DISPLAY	DSPLY2
GRAYMAP	PICT
ISOCLS	COVAR1, DSTAPE, LABMAN, PRINT, PSPLIT, and RDMEAN
LABEL	ALLKIN, CLSMAP, CNDMAP, DOTDST, DSPTAP. KNEAR, LABLR, LABMAN, and MIXMAP
NDHIST	NDHST1, NDHST2, RESTO (RESTOR), and WRTFIL
SCTRPL	CLRCOD, CNTER, LINPLT (PRTPLT, STOPTS), and SCATTR
SELECT	DAVDN3, DAVIDN, and GENRPT
TESTSP	COVPAT, DSTAPE, LABMAN, PRINT, PSPPAT, and RDMEAN

19.46.2 INTERFACES

The RREAD subprogram interfaces with other routines through the calling arguments.

19.46.3 INPUTS

Calling sequence: CALL RREAD(BEGADD,WHERE,TOTWDS,STATUS)

Parameter	Dimension	In/out	Definition
BEGADD	1	In	Number of words from the beginning of the file to the point where the read is to start.
WHERE	1	Out	Storage location for data read.
TOTWDS	1	In	Total number of data words to be read.
STATUS	1	Out	Set to 0 when input/output is complete; not used but returned as 0 for compatibility.

19.46.4 OUTPUTS

This subprogram outputs the data exactly as read from the scratch file into the output area defined by the calling argument WHERE.

19.46.5 STORAGE REQUIREMENTS

This subprogram requires 1982 bytes of storage.

19.46.6 DESCRIPTION

Using calling arguments BEGADD and TOTWDS, subprogram RREAD calculates the number of records of size BUFSIZ that are required to store the requested data. The records are read one at a time, and their contents are stored serially in WHERE.

19.46.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.46.8 LISTING

The subprogram listing is given in volume IV, section 19.

19.47 RWRITE

The RWRITE subprogram simulates the random write of a work file used to store data temporarily during the execution of a processor.

19.47.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
DAMRG	DAMRG
DISPLAY	DSPLY2
GRAYMAP	PICT
ISOCLS	ISOCLS, PSPLIT, RDDATA, and RDMEAN (RDFILE)
LABEL	CLSMAP, DOTDST, FILERD, LABLR, and STOMAP
NDHIST	NDHST1, NDHST2, and STODAT
SCTRPL	LINPLT (PRTPLT, STOPTS), SCATTR, SETADR, and STOFIL
SELECT	DAVDN1, DAVDN3, DAVIDN, and PRELIM
TESTSP	RDDPAT, RDMEAN (RDFILE), PSPPAT, and TESTSP

19.47.2 INTERFACES

The RWRITE subprogram interfaces with other routines through the calling arguments.

19.47.3 INPUTS

Calling sequence: CALL RWRITE(BEGADD,WHERE,TOTWDS,STATUS)

Parameter	Dimension	In/out	Definition
BEGADD	1	In	Number of words from the beginning of the file to the point where the write is to start.
WHERE	1	Out	Storage location for written data.
TOTWDS	1	In	Total number of data words to be written.
STATUS	1	Out	Set to 0 when input/output is complete; not used but returned as 0 for compatibility.

19.47.4 OUTPUTS

This subprogram outputs the data exactly as read from the scratch file into the output area defined by the calling argument WHERE.

19.47.5 STORAGE REQUIREMENTS

This subprogram requires 2084 bytes of storage.

19.47.6 DESCRIPTION

Using BEGADD and TOTWDS, subprogram RWRITE calculates the number of records of size BUFSIZ that are required to store the data. The records are read one at a time, and their contents are stored serially in WHERE. After data are stored, control is returned to the calling routine.

19.47.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.47.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.48 SAVFIL

The SAVFIL subprogram reads records from the SAVTAP file.

19.48.1 LINKAGES

This routine does not call any other subprogram. The subprogram which calls it, along with the processors, is listed below.

Processor	Calling subprogram
CLASSIFY	REDSAV
DATA-TR	REDSAV
LABEL	REDSAV
SELECT	REDSAV
TRSTAT	REDSAV

19.48.2 INTERFACES

The SAVFIL subprogram interfaces with other routines through common blocks GLOBAL and INFORM and through the calling arguments.

19.48.3 INPUTS

Input to the SAVFIL subprogram consists of the SAVTAP file output by the ISOCLS, STAT, or TESTSP processor.

Calling sequence: CALL SAVFIL(FLDSAV,VERTEX,CLSID,SUBNO,SUBDES, NOFLD,NOCLS,NOSUB)

Parameter	Dimension	In/out	Definition
FLDSAV	4,NOFLD	In/out	Array containing field information (name, class and subclass numbers, and number of vertices).
VERTEX	1	In/out	Array containing field vertices.

Parameter	Dimension	In/out	Definition
CLSID	1	In/out	Array containing class descriptions.
SUBNO	1	In/out	Array containing subclass numbers in each class.
SUBDES	1	In/out	Array containing subclass descriptions.
NOFLD	1	In	Number of fields used in processing.
NOCLS	1	In	Number of classes used in processing.
NOSUB	1	In	Number of subclasses used in processing.

19.48.4 OUTPUTS

The results are returned for use by the calling routine.

19.48.5 STORAGE REQUIREMENTS

This subprogram requires 900 bytes of storage.

19.48.6 DESCRIPTION

Not required.

19.48.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.48.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.49 SEARCH

The SEARCH subprogram searches for the correct scan line to be processed.

19.49.1 LINKAGES

This routine calls the CMERR subprogram. The subprogram which calls it, along with the processors, is listed below.

Processor	Calling subprogram
CLASSIFY	LINERD
DAMRG	LINERD
DATA-TR	LINERD
DOTDATA	LINERD
GRAYMAP	LINERD
GTDDM	LINERD
GTTCN	LINERD
HIST	LINERD
ISOCLS	LINERD
LABEL	LINERD
NDHIST	LINERD
STAT	LINERD
TESTSP	LINERD

19.49.2 INTERFACES

The SEARCH subprogram interfaces with other routines through common block TAPERD and through the calling arguments.

19.49.3 INPUTS

Calling sequence: CALL SEARCH(*,*,ENDTAP,IBUF,NRPDS,NDSPR)

Parameter	Dimension	In/out	Definition
*	1	Out	Exit route 1 (after all processing).
*	1	Out	Exit route 2 (in event of a missing scan line).
ENDTAP	1	Out	If = -1, EOF was encountered.
IBUF	765	In/out	Array containing record to be scanned.
NRPDS	1	In	Number of records per data set.
NDSPR	1	In	Number of data sets per record.

19.49.4 OUTPUTS

The results are returned for use by the calling routine.

19.49.5 STORAGE REQUIREMENTS

This subprogram requires 1450 bytes of storage.

19.49.6 DESCRIPTION

SEARCH looks for a specified scan line of data and informs the user how many times it scans each record and how many records are scanned during each search. If it is unable to locate the appropriate line, it so informs the user and either furnishes a previous scan (if available) or aborts via CMERR.

19.49.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.49.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.50 SETMRG

The SETMRG subprogram is an inactive routine.

19.50.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	CLSFY2
DISPLAY	DISTCV, DSPLY2, and PRTSUM
GRAYMAP	CLSHIS (COMHST, FLDHIS, HSTGRM) and HEADNG
HIST	CLSHIS
ISOCLS	PRINT
LABEL	CLSMAP
SCTRPL	LINPLT (PRTPLT, STOPTS)
SELECT	PLOT
STAT	CLSHIS (COMHST, FLDHIS, HSTGRM) and LEARN
TESTSP	PRINT

19.50.2 INTERFACES

The SETMRG subprogram interfaces with other routines through the calling arguments.

19.50.3 INPUTS

Calling sequence: CALL SETMRG(A,B,C)

Parameter	Dimension	In/out	Definition
A	1	In/out	Inactive.
B	1	In/out	Inactive.
C	1	In/out	Inactive.

19.50.4 OUTPUTS

Not applicable.

19.50.5 STORAGE REQUIREMENTS

This subprogram requires 324 bytes of storage.

19.50.6 DESCRIPTION

Not required.

19.50.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.50.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.51 SETUP7

The SETUP7 subprogram reads and analyzes all control card input and sets options for the ISOCLS and TESTSP processors.

19.51.1 LINKAGES

The SETUP7 subprogram calls the CRDSTA, FIND12, FLTNUM, NUMBER, NXTCHR, ORDER, and RDMEAN subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
ISOCLS	ISOCLS
TESTSP	TESTSP

19.51.2 INTERFACES

The SETUP7 subprogram interfaces with other routines through common blocks GLOBAL, ISOLNK, and PASS and through the calling arguments.

19.51.3 INPUTS

Optional input to the SETUP7 subprogram consists of SAVTAP and DOTUNT files output by the STAT and DOTDATA processors, respectively.

Calling sequence: CALL SETUP7(ARRAY,TOP,ITIME)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	Out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
ITIME	1	In	Counts number of times SETUP7 is called.

The control cards relevant to this routine are given in volume II, section 9 (table 9-1), of this user guide. Class and field definitions (volume II, section 3.2.3) and (optionally) initial cluster centers are input by the user.

19.51.4 OUTPUTS

The SETUP7 subprogram prints messages if errors occur on the CHANNELS, DATAFILE, STATFILE, and DOTFILE cards or in the event of any invalid input; a list of user-requested parameter values and options; and other diagnostic messages (volume II, section 9).

19.51.5 STORAGE REQUIREMENTS

This subprogram requires 9888 bytes of storage.

19.51.6 DESCRIPTION

The SETUP7 subprogram initializes default values for input parameters and sets up the reread buffer for card input. It reads each control card, identifies the keyword, and branches to the statement number which processes that input. After the card is processed, the program branches back to the read statement and reads the next card. SETUP7 sets up supervisory procedures for clustering, reads in the module STAT file and stores statistics on the SAVTAP file for subsequent processing, and prints a list of user-requested parameter values and options.

19.51.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.51.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.52 SUNFAC

Subprogram SUNFAC computes Sun-angle gain corrections for pixel radiance values for each channel based on input Sun angles.

19.52.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
DAMRG	DAMRG
ISOCLS	ISODAT
LABEL	DOTDST
TESTSP	ISOPAT

19.52.2 INTERFACES

The SUNFAC subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

19.52.3 INPUTS

Calling sequence: CALL SUNFAC(SUNCOR,SUNANG,FETVEC,NOFEAT,ISUNC, ISUNT)

Parameter	Dimension	In/out	Definition
SUNCOR	1	Out	Sun-angle corrections corresponding to channels in FETVEC.
SUNANG	1	In	Array of Sun angles.
FETVEC	1	In	Array of channel numbers.
NOFEAT	1	In	Number of channels.

Parameter	Dimension	In/out	Definition
ISUNC	1	In	Flag indicating Sun angles came from card images.
ISUNT	1	In	Flag indicating Sun angles were extracted from the MSS DATAPE.

19.52.4 OUTPUTS

The results are returned for use by the calling routine.

19.52.5 STORAGE REQUIREMENTS

This subprogram requires 2492 bytes of storage.

19.52.6 DESCRIPTION

The Sun-angle corrections are computed from an internal table covering four channels at a haze factor of 25 percent. The corrections correspond to the channels contained in FETVEC. If the ISUNT flag is on, the first Sun angle in SUNANG is assumed to correspond to pass 1. If the ISUNC flag is on, the first Sun angle in SUNANG is assumed to correspond to the first pass in FETVEC.

The Sun-angle corrections, which basically are gain factors, are used in the computation of cluster mean-dot distances. The Sun-angle gain corrections computed by SUNFAC are used in computing pixel/cluster mean distances in the clustering routines PSPLIT and PSPPAT in determining the largest standard deviations in routines ISODAT and ISOPAT.

19.52.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.52.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.53 TAPHDR

The TAPHDR subprogram reads the header record of the MSS DATAPE in either the Universal or LARSYS II format.

19.53.1 LINKAGES

This routine calls the BUFILL and CMERR subprograms. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
CLASSIFY	CLSFY2
DAMRG	DAMRG
DATA-TR	LNTRAN and TRHIST
DOTDATA	DOTS
GRAYMAP	HISTGM and PICT
GTDDM	DDM
GTTCN	TCN
HIST	HISTGM
ISOCLS	RDDATA
LABEL	STOMAP
NDHIST	NDHST1 and STODAT
STAT	LEARN
TESTSP	RDDPAT

19.53.2 INTERFACES

The TAPHDR subprogram interfaces with other routines through common blocks IDSTOR, IDLNK, and TAPERD and through the calling arguments.

19.53.3 INPUTS

Input to the TAPHDR subprogram consists of a multifile MSS DATAPE in Universal or LARSYS II format.

Calling sequence: CALL TAPHDR(DATAPE,IFILE)

Parameter	Dimension	In/out	Definition
DATAPE	1	In	Tape unit number.
IFILE	1	In	Number of EOF's to be skipped in positioning tape.

19.53.4 OUTPUTS

The results are returned for use by the calling routine.

19.53.5 STORAGE REQUIREMENTS

This subprogram requires 34 382 bytes of storage.

19.53.6 DESCRIPTION

The TAPHDR subprogram unpacks the following information from the header record of the MSS DATAPE:

Byte numbers	Description
109-110	Pixel start number
111-112	Pixel stop number
1789-1790	Pixel skip factor
1791-1792	Line skip factor
2201-2202	Sun angle for pass 1 (channels 1-4)
2203-2204	Sun angle for pass 2 (channels 5-8)
⋮	⋮
2215	Sun angle for pass 8 (channels 29-30)

This information is unpacked simply by extending the TAPHDR subprogram data vectors HWRD, BIT, and NB; expanding the dimension of the ID vector; and expanding an existing unpacking code block. The above information will be placed in labeled common ISOLNK.

The arrays HWRD and NB are precalculated word and bit positions which must be extracted from the header record.

The subprograms FLDINT (section 19.15) and LINERD (section 19.27) are required, also, to read the MSS DATAPE.

19.53.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.53.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.54 WRTBMT

The WRTBMT subprogram writes the B-matrix. The subprogram WRTBMT has two entries: WRTBMT, which is used to write the double-precision B-matrix (BMAT); and WRTBM, which is used to write the single-precision B-matrix (BBMAT).

19.54.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
CLASSIFY	SETUP2
DATA-TR	SETUP8
SCTRPL	SET11
SELECT	SELECT

19.54.2 INTERFACES

The WRTBMT subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

19.54.3 INPUTS

Calling sequences:

a. CALL WRTBMT(BMAT,NOFET4,NOFET2,FETVC2)

Parameter	Dimension	In/out	Definition
BMAT	NOFET4,NOFET2	In	Array containing the double-precision B-transformation matrix.
NOFET4	1	In	Number of linear combinations.
NOFET2	1	In	Number of channels.

Parameter	Dimension	In/out	Definition
FETVC2	30	In	Array of channels.

b. ENTRY WRTBM(BBMAT,NOFET4,NOFET2,FETVC2)

Parameter	Dimension	In/out	Definition
BBMAT	NOFET4,NOFET2	In	Array containing the single-precision B-transformation matrix.

19.54.4 OUTPUTS

This subprogram outputs the linear transformation B-matrix, including the number of channels and number of linear combinations, on the specified unit.

19.54.5 STORAGE REQUIREMENTS

This subprogram requires 1416 bytes of storage.

19.54.6 DESCRIPTION

Not required.

19.54.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.54.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.55 WRTDOT

The subprogram WRTDOT outputs the dot data tape, DOTFIL, for the processors DOTDATA and LABEL.

19.55.1 LINKAGES

This routine calls the FSBSFL subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
DOTDATA	DOTS
LABEL	LABLR

19.55.2 INTERFACES

The WRTDOT subprogram interfaces with other routines through the calling arguments.

19.55.3 INPUTS

Calling sequence: CALL WRTDOT(TOTDOT,NOSUN,FLDSAV,VERTEX,ANGLE, DOTS,NOCAT,CATNAM,SIZE,NOFET2,FETVC2,TOTVT2,NOFLD2,UNIT,FILE)

Parameter	Dimension	In/out	Definition
TOTDOT	1	In	Total number of dots to be written on DOTFIL.
NOSUN	1	In	Number of Sun angles.
FLDSAV	4,1	In	Array containing field information.
VERTEX	2,1	In	Array containing field vertices.
ANGLE	1	In	Array containing Sun-angle values.

Parameter	Dimension	In/out	Definition
DOTS	SIZE,TOTDOT	In	Array containing dot information.
NOCAT	1	In	Number of categories.
CA"NAM	NOCAT	In	Array containing category names.
SIZE	1	In	4 + NOFET2.
NOFET2	1	In	Number of features (channels).
FETVC2	30	In	Array containing channel numbers.
TOTVT2	1	In	Total number of vertices.
NOFLD2	1	In	Number of fields.
UNIT	1	In	Number of the Fortran unit on which file is to be written.
FILE	1	In	File number on UNIT for this output file.

19.55.4 OUTPUTS

WRTDOT outputs a multifile unformatted Fortran tape with three records per file.

19.55.5 STORAGE REQUIREMENTS

This subprogram requires 1308 bytes of storage.

19.55.6 DESCRIPTION

WRTDOT outputs the DOTFIL; one file is written per call to WRTDOT. In general, two types of files are written, the first containing labeling/starting dots and the second containing bias correction dots. A file is created for each type of dot.

19.55.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.55.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.56 WRTFLD

The WRTFLD subprogram prints saved training or test fields.

19.56.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
CLASSIFY	CLSFY1
DISPLAY	SETUP3
DOTDATA	DOTS
LABEL	FILERD
NDHIST	WRTFIL
SCTRPL	SCATTR
SELECT	PRTFLD

19.56.2 INTERFACES

The WRTFLD subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

19.56.3 INPUTS

Calling sequence: CALL WRTFLD(FLDSAV,VERTEX,NOFLD,KEY,CLSNAM, SUBNAM)

Parameter	Dimension	In/out	Definition
FLDSAV	4,NOFLD	In	Array containing field information.
VERTEX	2,1	In	Array containing field vertices.
NOFLD	1	In	Number of fields.

Parameter	Dimension	In/out	Definition
KEY	1	In	Indicator as to type of data requested (see section 19.55.6).
CLSNAM	1	In	Array containing class names.
SUBNAM	1	In	Array containing subclass names.

19.56.4 OUTPUTS

This subprogram outputs saved training or test fields on the specified unit.

19.56.5 STORAGE REQUIREMENTS

This subprogram requires 2060 bytes of storage.

19.56.6 DESCRIPTION

Subprogram WRTFLD prints out saved training or test fields in accordance with the value of the parameter KEY.

If KEY = 1, saved training fields are output.

If KEY = 2, input test fields are output.

Training and test fields are printed out in the following columnar form:

FIELD CLASS SUBCLASS VERTICES (SAMPLE,LINE)

If KEY = 3, designated fields are printed out in the following columnar form:

DESIGNATED FIELDS

FIELD DESIGNATED VERTICES (SAMPLE,LINE)

Also, upon user request, this subprogram will print out a separate list of DO/DU fields.

19.56.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.56.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.57 WRTHED

The WRTHED subprogram writes the header record for each data tape in Universal or LARSYS III format.

19.57.1 LINKAGES

This routine calls the WRTREC subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
DAMRG	DAMRG
DATA-TR	LNTRAN
DISPLAY	DSPLY2
GTTCN	TCN
ISOCLS	DSTAPE
LABEL	CLSMAP
SCTRPL	SCATTR
TESTSP	DSTAPE

19.57.2 INTERFACES

The WRTHED subprogram interfaces with other routines through common blocks IDSTOR, TAPERD, and WRTAP and through the calling arguments.

19.57.3 INPUTS

Calling sequence: CALL WRTHED(NCHAN,FEAT,NSAMP,FRMAT,IUNIT)

Parameter	Dimension	In/out	Definition
NCHAN	1	In	Number of channels to be written for each data set.

Parameter	Dimension	In/out	Definition
FEAT	30	In	Array containing channels to be written.
NSAMP	1	In	Number of samples per channel.
FRMAT	1	In	If = 1, Universal format.
IUNIT	1	In	Number of tape output unit; if = 2, LARSYS III format.

19.57.4 OUTPUTS

This subprogram outputs the header record on tape.

19.57.5 STORAGE REQUIREMENTS

This subprogram requires 5912 bytes of storage.

19.57.6 DESCRIPTION

The WRTHED subprogram writes the header record in 32-bit bytes for LARSYS III and 8-bit bytes for Universal format. This information is packed. One call is made to this subprogram for each reel of tape.

19.57.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.57.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.58 WRTLN

The WRTLN subprogram writes the data for each data tape in either Universal or LARSYS II format.

19.58.1 LINKAGES

This routine calls the WRTREC subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
DAMRG	DAMRG
DATA-TR	LNTRAN
DISPLAY	DSPLY2
GTTCN	TCN
ISOCLS	DSTAPE
LABEL	CLRKEY and CLSMAP
SCTRPL	CLRKYS and SCATTR
TESTSP	DSTAPE

19.58.2 INTERFACES

The WRTLN subprogram interfaces with other routines through common block WRTAP and through the calling arguments.

19.58.3 INPUTS

Calling sequence: CALL WRTLN(IDATA,LSTLIN)

Parameter	Dimension	In/out	Definition
IDATA	1	In	Storage array for data to be written.
LSTLIN	1	In	= 0 for (N - 1) data sets; = -1, for last data set.

19.58.4 OUTPUTS

This subprogram outputs the packed data in 8-bit bytes on tape.

19.58.5 STORAGE REQUIREMENTS

This subprogram requires 15 132 bytes of storage.

19.58.6 DESCRIPTION

The WRTLN subprogram writes the data in 8-bit bytes. WRTLN must be called for each data set to be written. Packed data are written on tape in LARSYS II format, one set of packed data per record. Packed data are written also in Universal format. Ancillary information is packed into the array PACRAY and then written on tape by calling WRTREC.

19.58.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.58.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.59 WRTMTX

The WRTMTX subprogram prints the single-precision covariance matrices. Entry DWRTMX prints the double-precision covariance matrices.

19.59.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprogram
CLASSIFY	CLSFY1
DATA-TR	PRTCOV
DISPLAY	DSPLY1
ISOCLS	LABMAN
LABEL	LABMAN
SELECT	PRTFLD
STAT	FLDCOV
TESTSP	LABMAN
TRSTAT	PRTCOV

19.59.2 INTERFACES

The WRTMTX subprogram interfaces with other routines through the calling arguments.

19.59.3 INPUTS

Calling sequences:

a. CALL WRTMTX(MATICE,SIZE,BCD)

Parameter	Dimension	In/out	Definition
MATICE	1	In	Array containing the single-precision covariance matrix.
SIZE	1	In	Rank of MATICE (DMATICE)
BCD	1	In	Contains BCD precision for printout.

b. ENTRY DWRTMX(DMATIC,SIZE,BCD)

Parameter	Dimension	In/out	Definition
DMATIC	1	In	Array containing the double-precision covariance matrix.

19.59.4 OUTPUTS

This subprogram outputs the single- or double-precision covariance matrix on the specified unit.

19.59.5 STORAGE REQUIREMENTS

This subprogram requires 1040 bytes of storage.

19.59.6 DESCRIPTION

The subprogram WRTMTX has two entry points: WRTMTX, which prints the single-precision covariance matrix; and DWRTMX, which prints the double-precision covariance matrix.

19.59.7 FLOW CHART

The available flow charts for the utility subprograms are provided in section 19.61.

19.59.8 LISTING

The subprogram listing is provided in volume IV, section 19.

19.60 WRTREC

The WRTREC subprogram outputs a scan line (one record) of data.

19.60.1 LINKAGES

This routine does not call any other subprogram. The subprograms which call it, along with the respective processors, are listed below.

Processor	Calling subprograms
DAMRG	WRTHED and WRTLN
DATA-TR	WRTHED and WRTLN
DISPLAY	WRTHED and WRTLN
GTTCN	WRTHED and WRTLN
ISOCLS	WRTHED and WRTLN
LABEL	WRTHED and WRTLN
SCTRPL	WRTHED and WRTLN
TESTSP	WRTHED and WRTLN

19.60.2 INTERFACES

The WRTREC subprogram interfaces with other routines through the calling arguments.

19.60.3 INPUTS

Calling sequence: CALL WRTREC(UNIT,LENGTH,IBUF)

Parameter	Dimension	In/out	Definition
UNIT	1	In	Number of the Fortran unit on which the record is to be written.
LENGTH	1	In	Number of lines or records.

Parameter	Dimension	In/out	Definition
IBUF	3000	In	Array containing records to be written.

19.60.4 OUTPUTS

This subprogram outputs a scan line of data on the specified unit.

19.60.5 STORAGE REQUIREMENTS

This subprogram requires 448 bytes of storage.

19.60.6 DESCRIPTION

Not required.

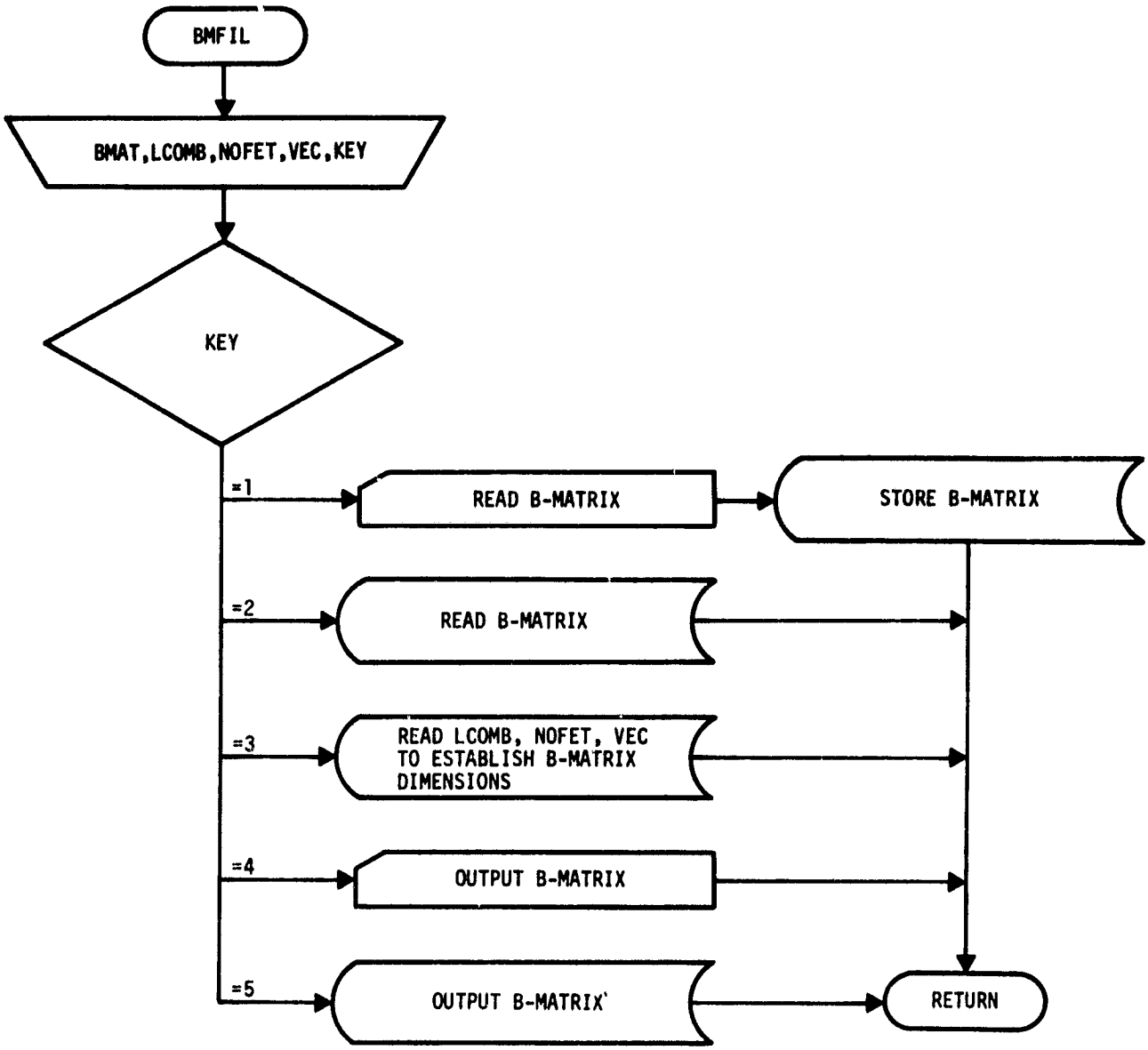
19.60.7 FLOW CHART

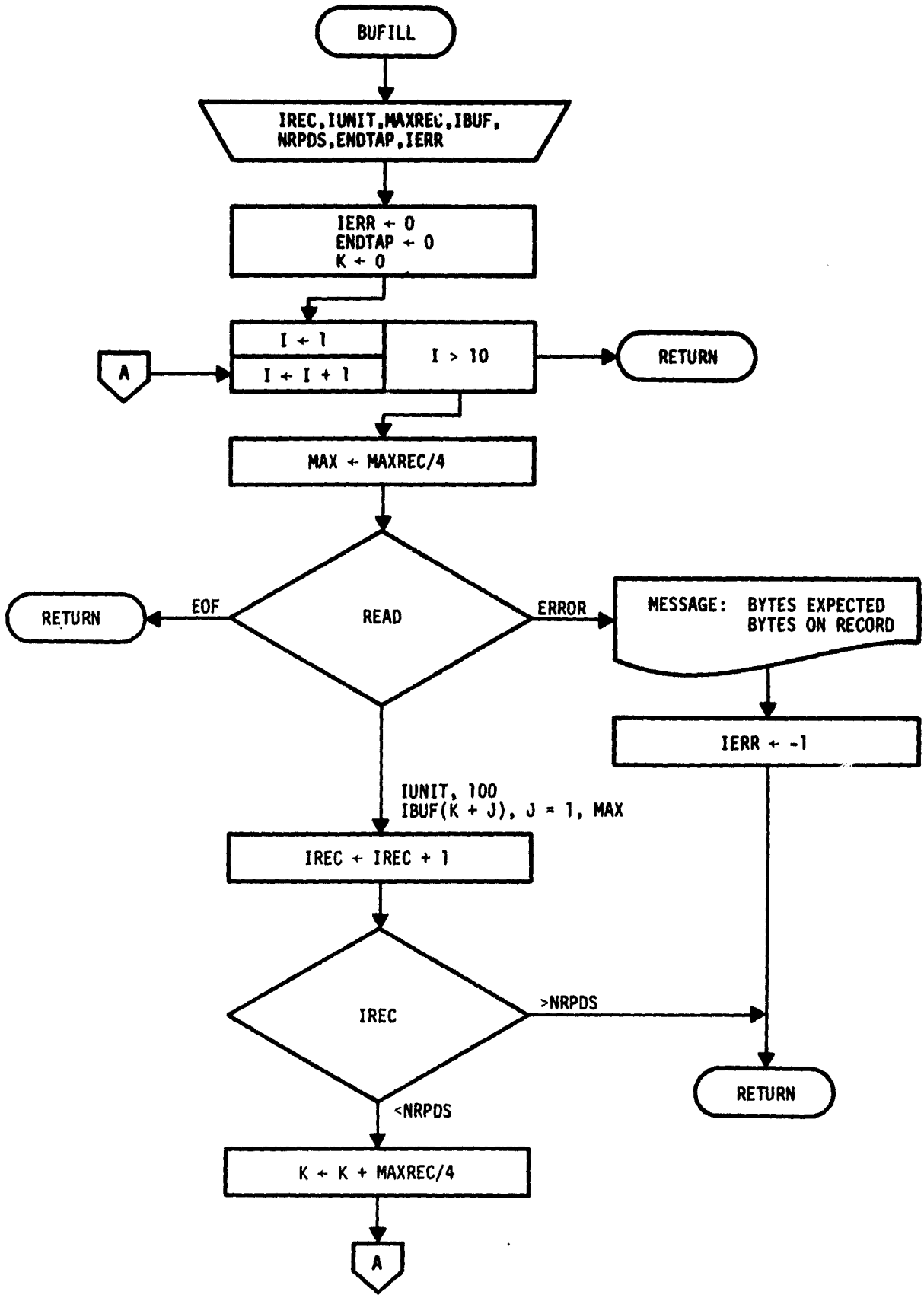
The available flow charts for the utility subprograms are provided in section 19.61.

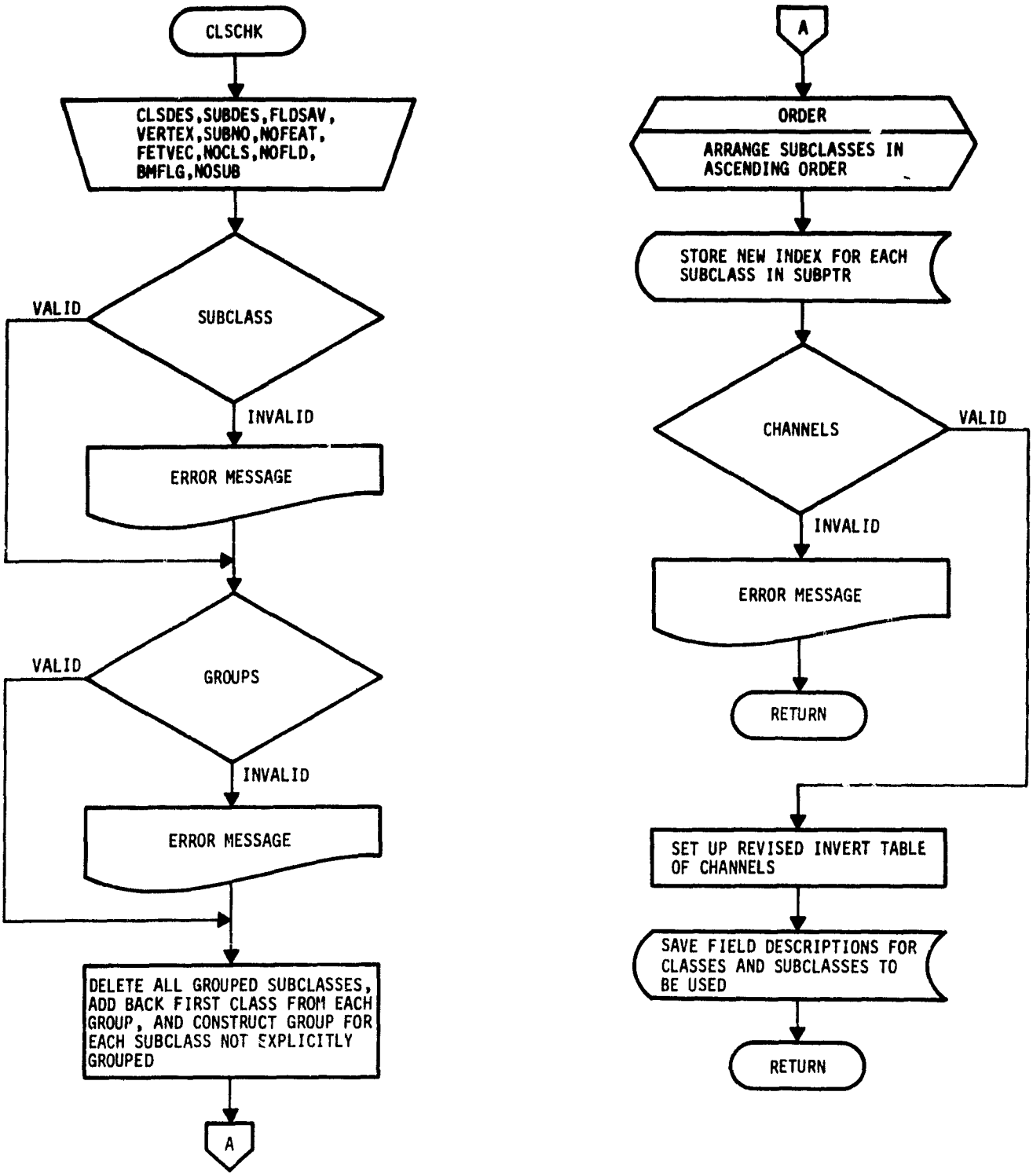
19.60.8 LISTING

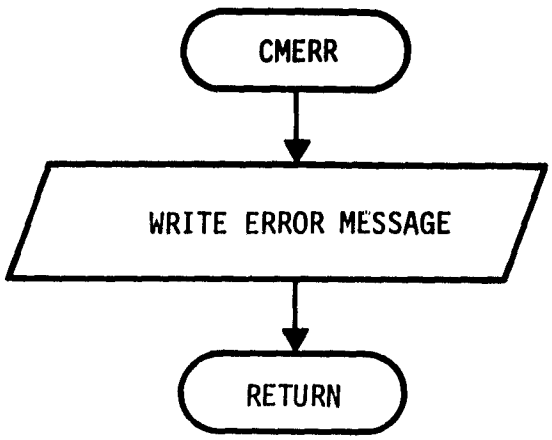
The subprogram listing is provided in volume IV, section 19.

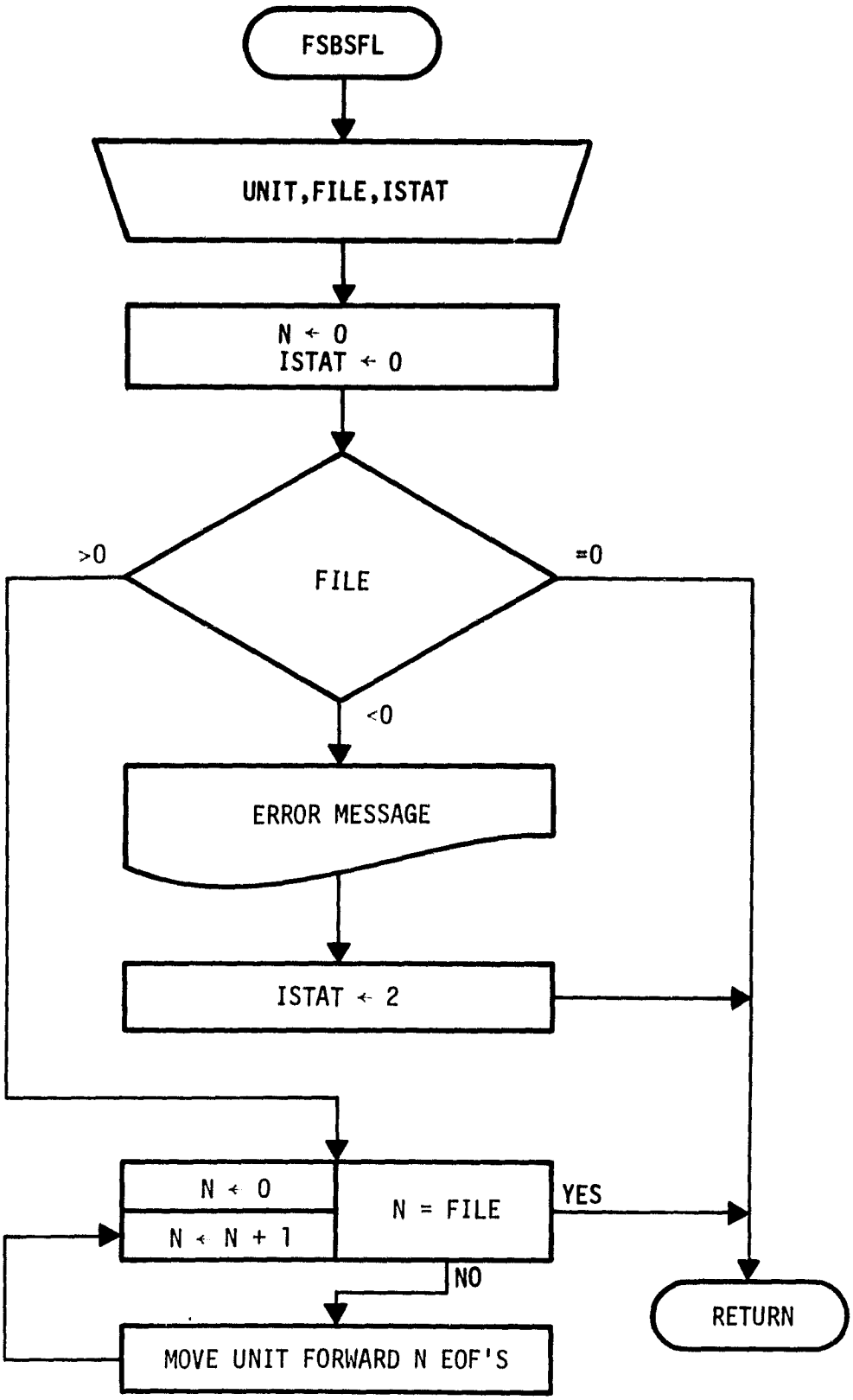
19.61 UTILITY SUBPROGRAM FLOW CHARTS

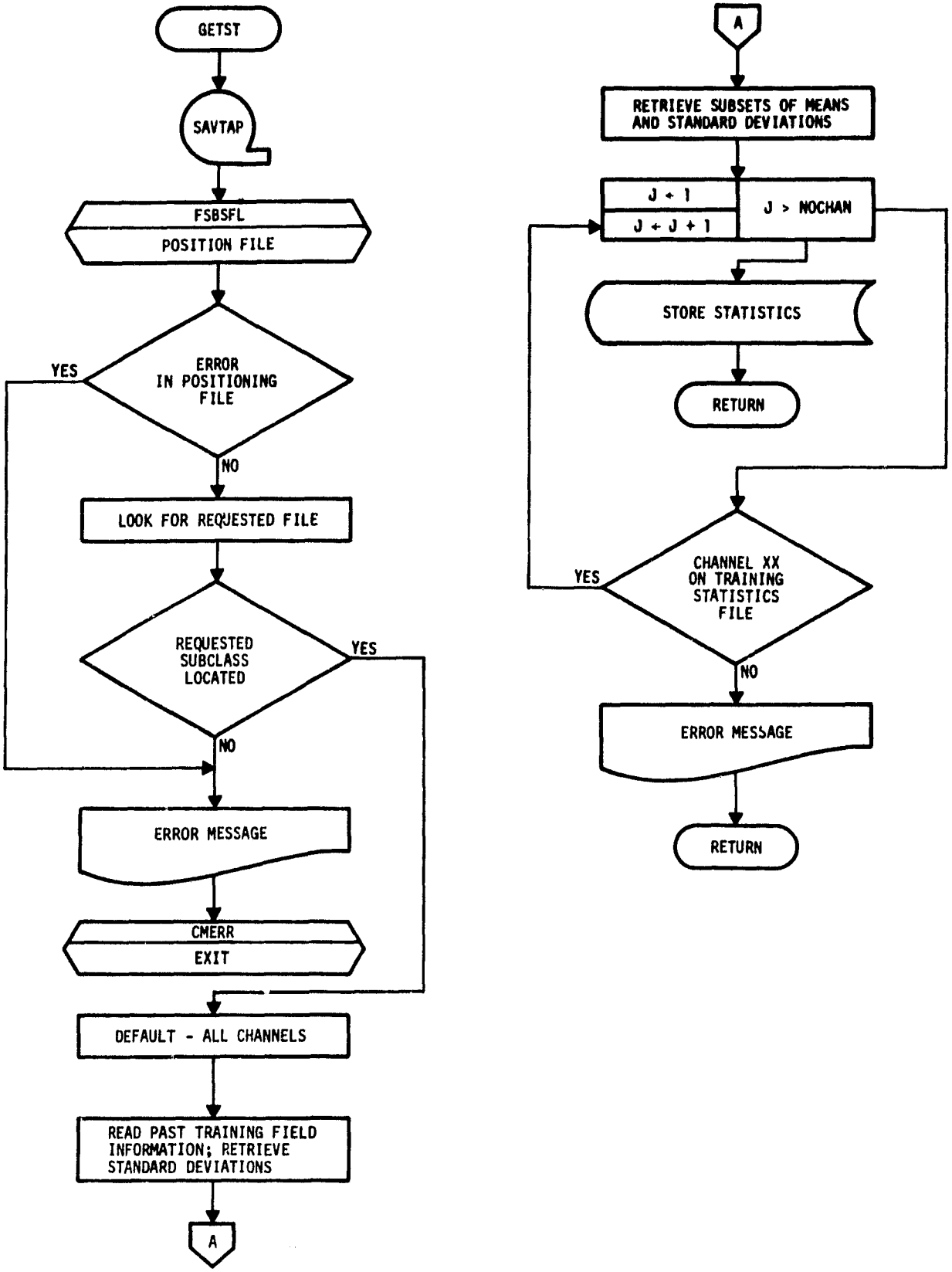


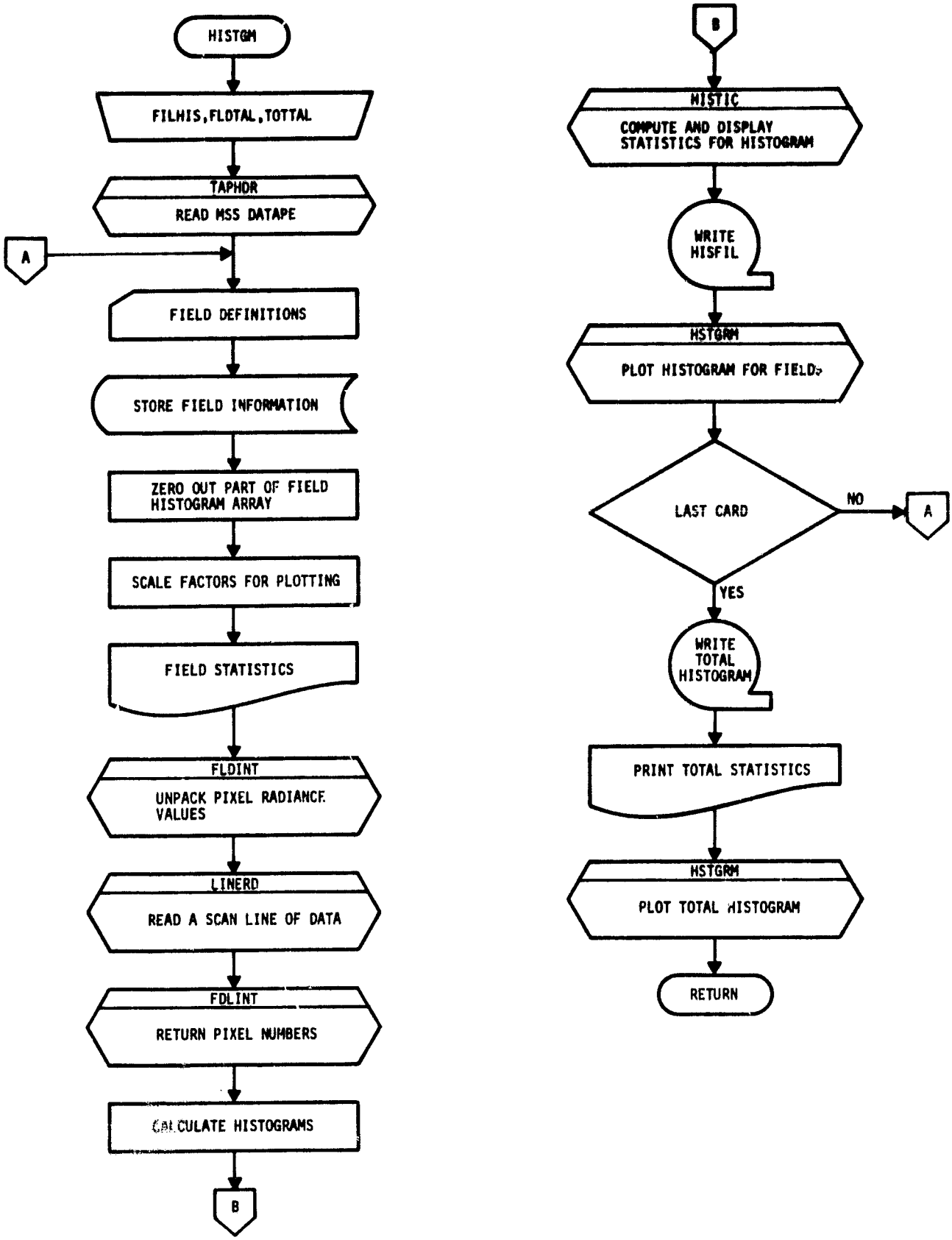

BUFILL
IREC,IUNIT,MAXREC,IBUF,
NRPDS,ENDTAP,IERR
IERR ← 0
ENDTAP ← 0
K ← 0
I ← 1
I ← I + 1
A
I > 10
RETURN
MAX ← MAXREC/4
READ
EOF
RETURN
ERROR
MESSAGE: BYTES EXPECTED
BYTES ON RECORD
IERR ← -1
IUNIT, 100
IBUF(K + J), J = 1, MAX
IREC ← IREC + 1
IREC
>NRPDS
RETURN
<NRPDS
K ← K + MAXREC/4
A

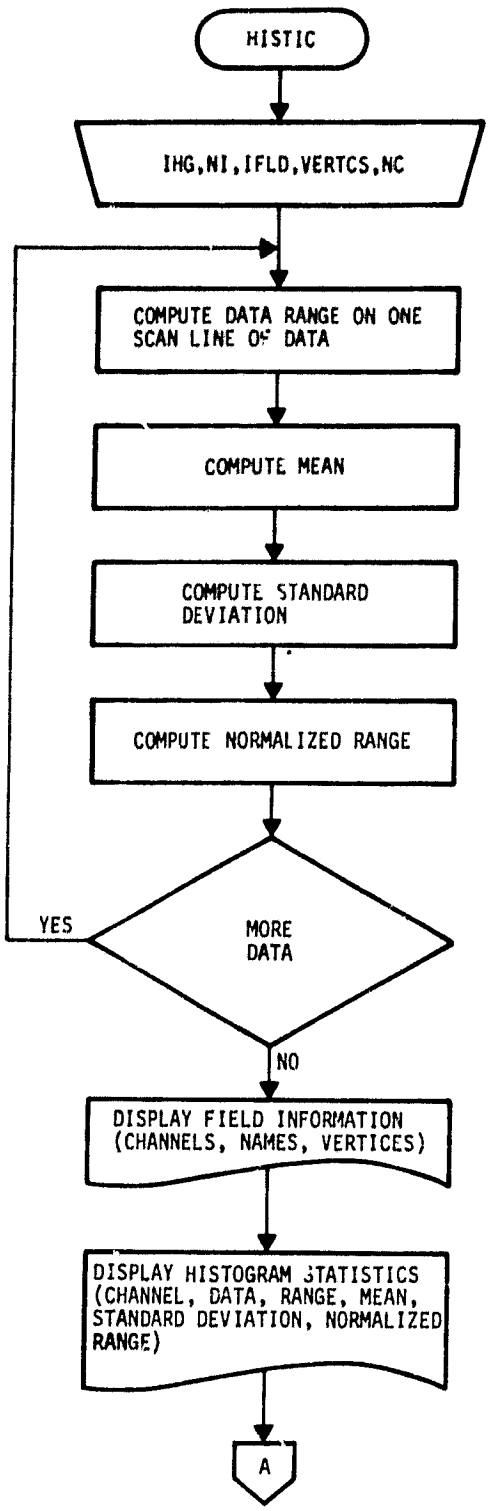

CLSCHK
CLSDES,SUBDES,FLDSAV,
VERTEX,SUBNO,NOFEAT,
FETVEC,NOCLS,NOFLD,
BMFLG,NOSUB
SUBCLASS
VALID
INVALID
ERROR MESSAGE
GROUPS
VALID
INVALID
ERROR MESSAGE
DELETE ALL GROUPED SUBCLASSES, ADD BACK FIRST CLASS FROM EACH GROUP, AND CONSTRUCT GROUP FOR EACH SUBCLASS NOT EXPLICITLY GROUPED
A
A
ORDER
ARRANGE SUBCLASSES IN ASCENDING ORDER
STORE NEW INDEX FOR EACH SUBCLASS IN SUBPTR
CHANNELS
VALID
INVALID
ERROR MESSAGE
RETURN
SET UP REVISED INVERT TABLE OF CHANNELS
SAVE FIELD DESCRIPTIONS FOR CLASSES AND SUBCLASSES TO BE USED
RETURN

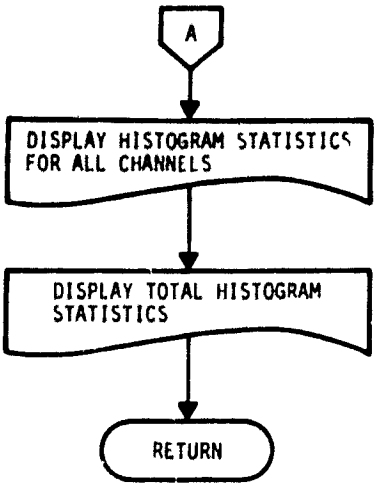

CMERR
WRITE ERROR MESSAGE
RETURN

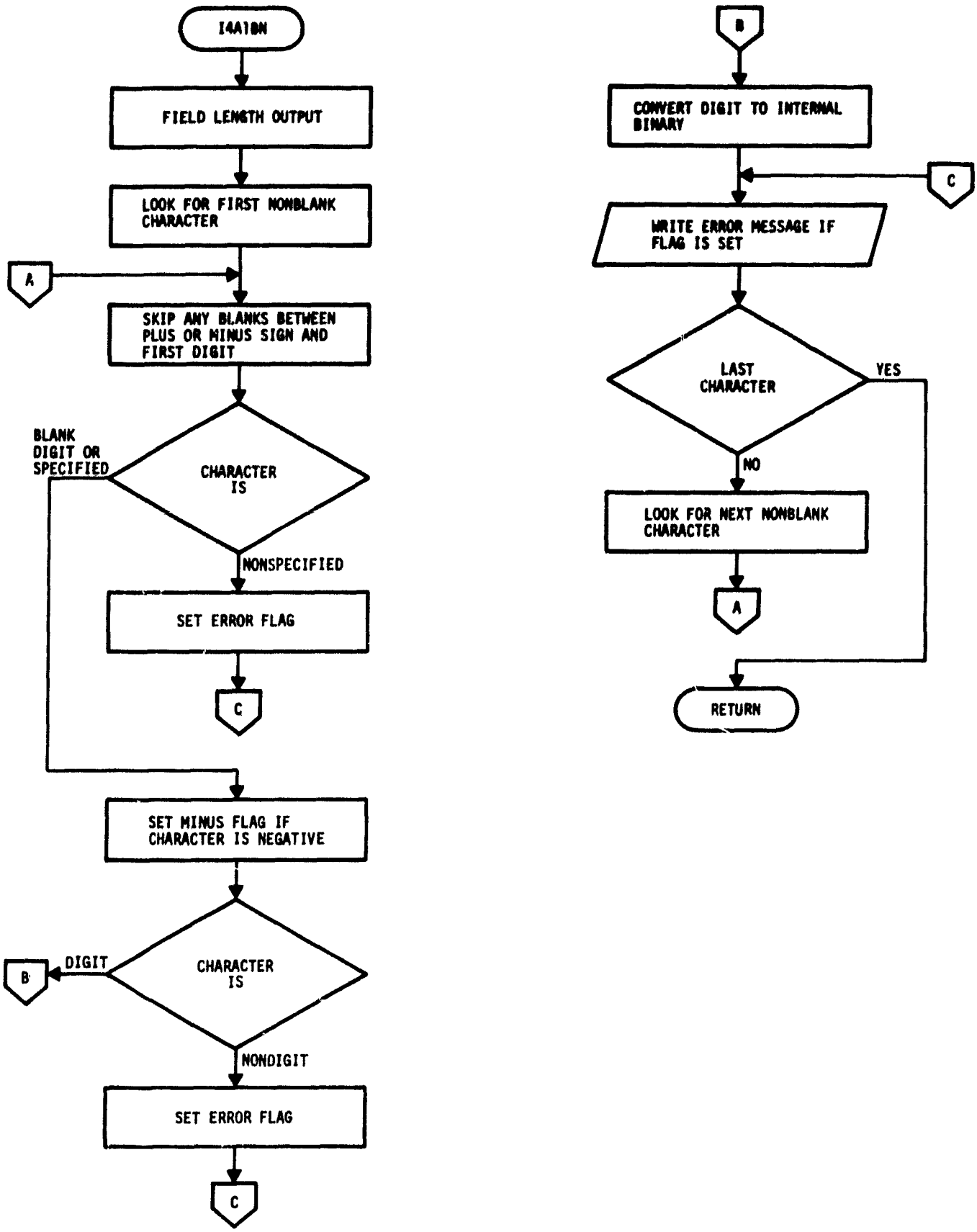

FSBSFL
UNIT,FILE,ISTAT
N ← 0
ISTAT ← 0
FILE
>0
=0
<0
ERROR MESSAGE
ISTAT ← 2
N ← 0
N ← N + 1
N = FILE
YES
NO
MOVE UNIT FORWARD N EOF'S
RETURN

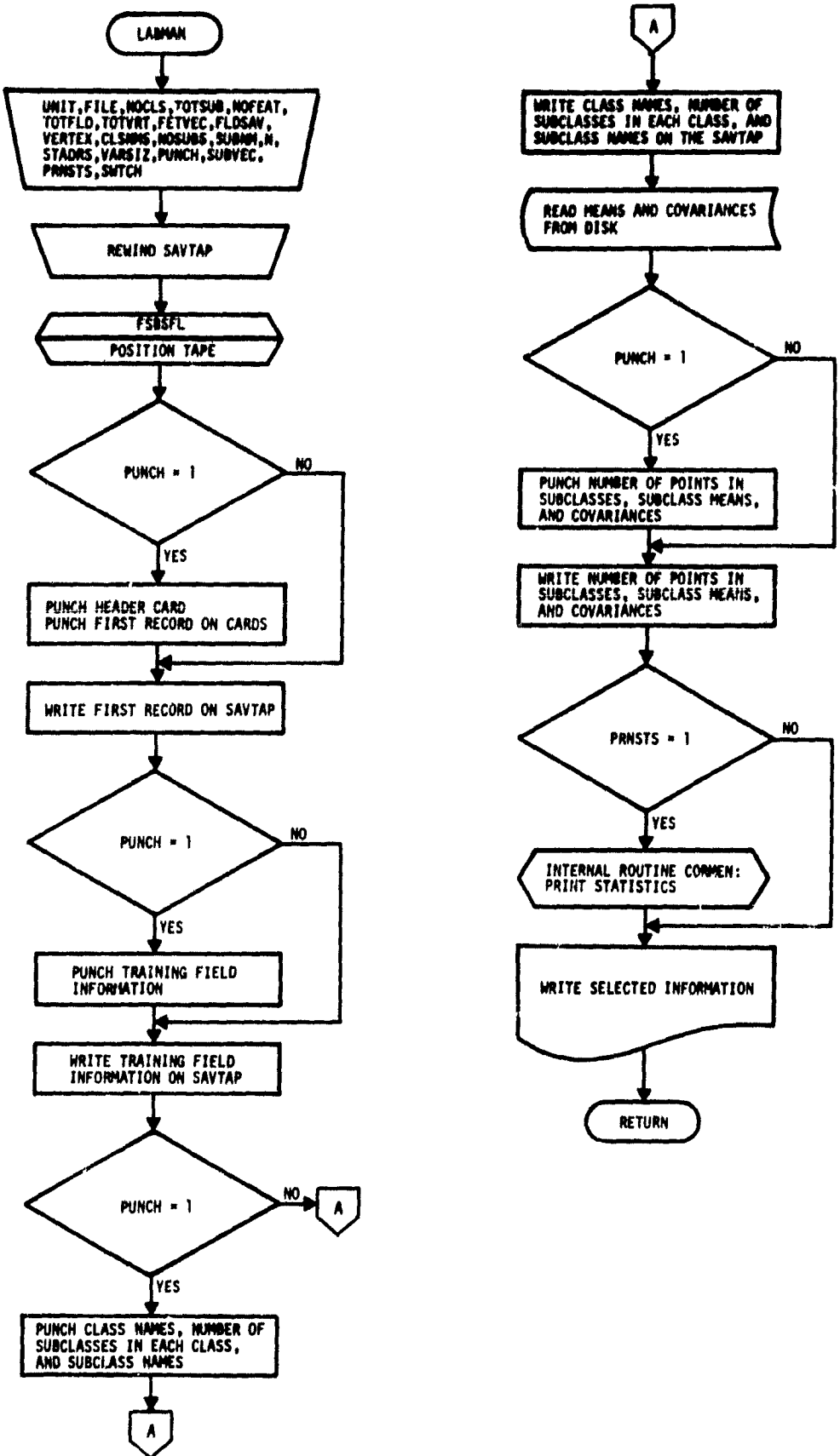

GETST
SAVTAP
FSBSFL
POSITION FILE
ERROR IN POSITIONING FILE
YES
NO
LOOK FOR REQUESTED FILE
REQUESTED SUBCLASS LOCATED
YES
NO
ERROR MESSAGE
CMERR
EXIT
DEFAULT - ALL CHANNELS
READ PAST TRAINING FIELD INFORMATION; RETRIEVE STANDARD DEVIATIONS
A
A
RETRIEVE SUBSETS OF MEANS AND STANDARD DEVIATIONS
J ← 1
J ← J + 1
J > NOCHAN
STORE STATISTICS
RETURN
CHANNEL XX ON TRAINING STATISTICS FILE
YES
NO
ERROR MESSAGE
RETURN

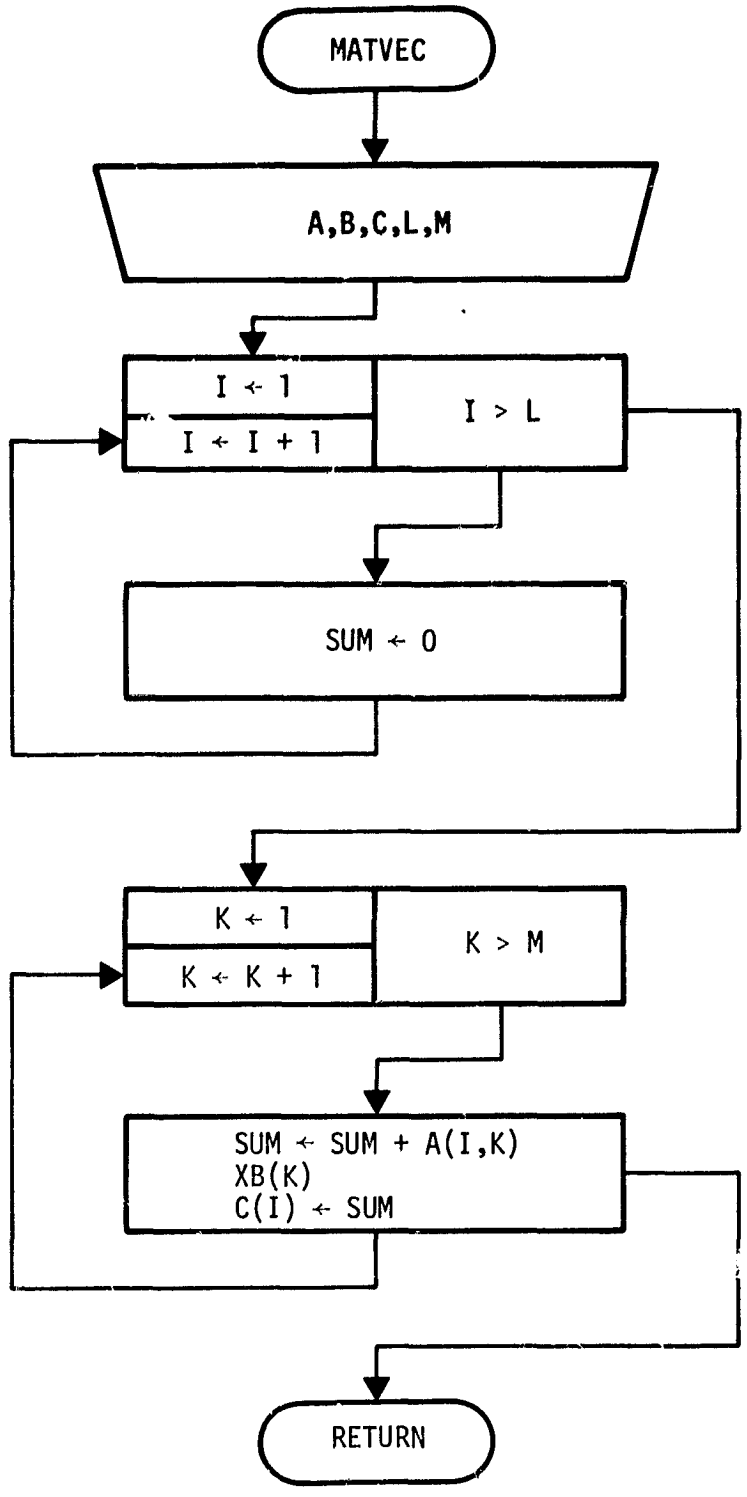

HISTGM
FILHIS,FLOTAL,TOTTAL
TAPHDR
READ MSS DATAPE
A
FIELD DEFINITIONS
STORE FIELD INFORMATION
ZERO OUT PART OF FIELD HISTOGRAM ARRAY
SCALE FACTORS FOR PLOTTING
FIELD STATISTICS
FLDINT
UNPACK PIXEL RADIANCE VALUES
LINERD
READ A SCAN LINE OF DATA
FDLINT
RETURN PIXEL NUMBERS
CALCULATE HISTOGRAMS
B
B
HISTIC
COMPUTE AND DISPLAY STATISTICS FOR HISTOGRAM
WRITE HISFIL
HSTGRM
PLOT HISTOGRAM FOR FIELD
LAST CARD
NO
A
YES
WRITE TOTAL HISTOGRAM
PRINT TOTAL STATISTICS
HSTGRM
PLOT TOTAL HISTOGRAM
RETURN

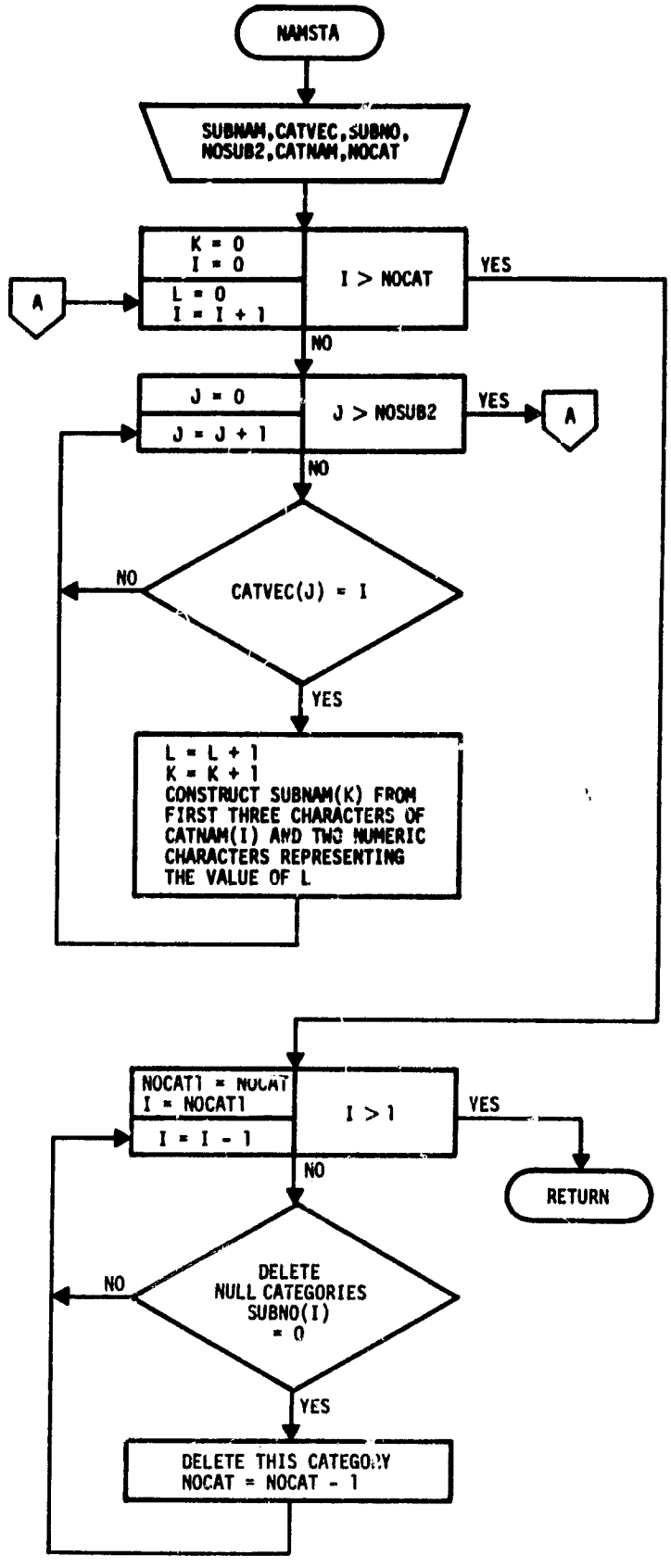

HISTIC
IHG,NI,IFLD,VERTCS,NC
COMPUTE DATA RANGE ON ONE SCAN LINE OF DATA
COMPUTE MEAN
COMPUTE STANDARD DEVIATION
COMPUTE NORMALIZED RANGE
MORE DATA
YES
NO
DISPLAY FIELD INFORMATION (CHANNELS, NAMES, VERTICES)
DISPLAY HISTOGRAM STATISTICS (CHANNEL, DATA, RANGE, MEAN, STANDARD DEVIATION, NORMALIZED RANGE)
A

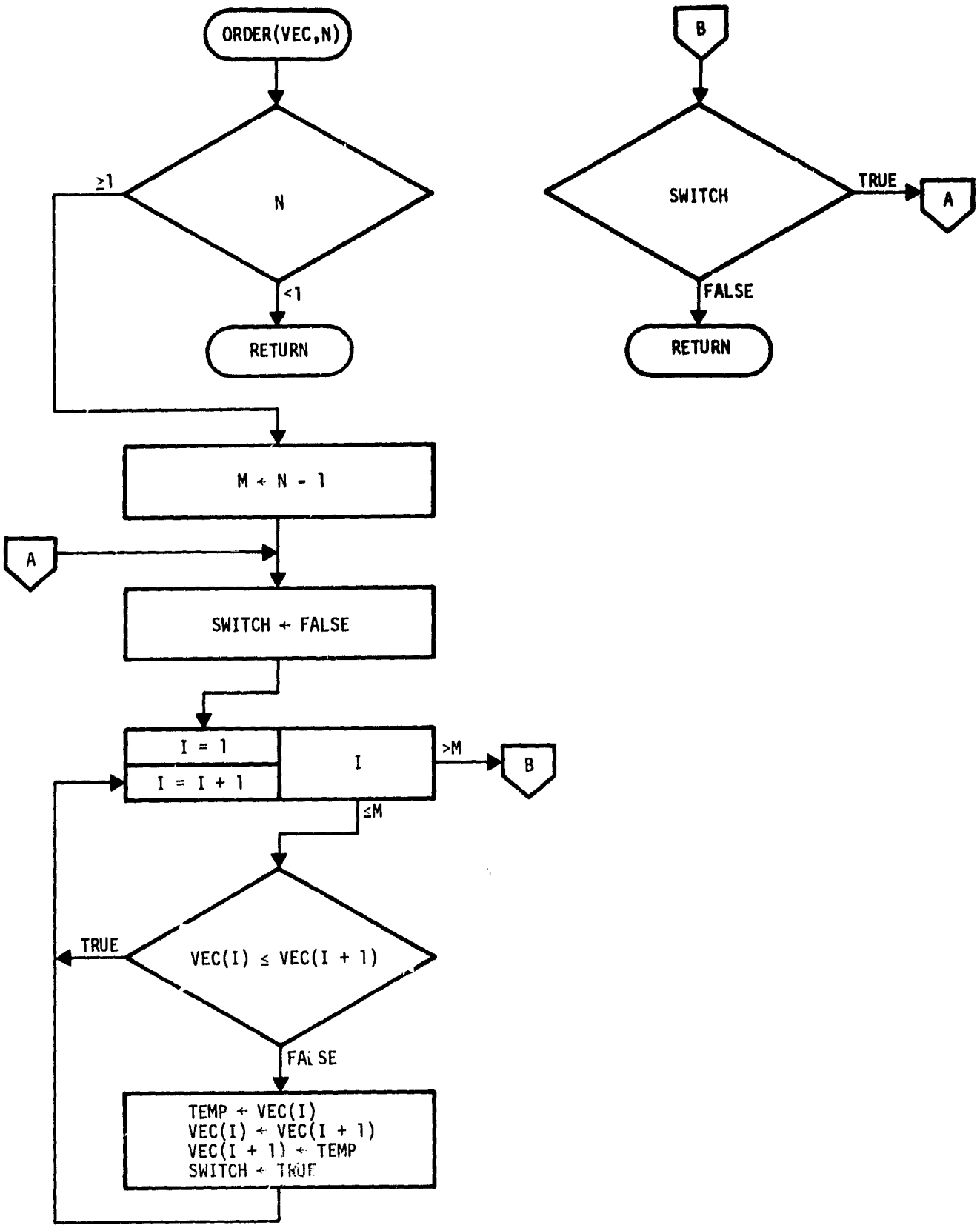

A
DISPLAY HISTOGRAM STATISTICS FOR ALL CHANNELS
DISPLAY TOTAL HISTOGRAM STATISTICS
RETURN

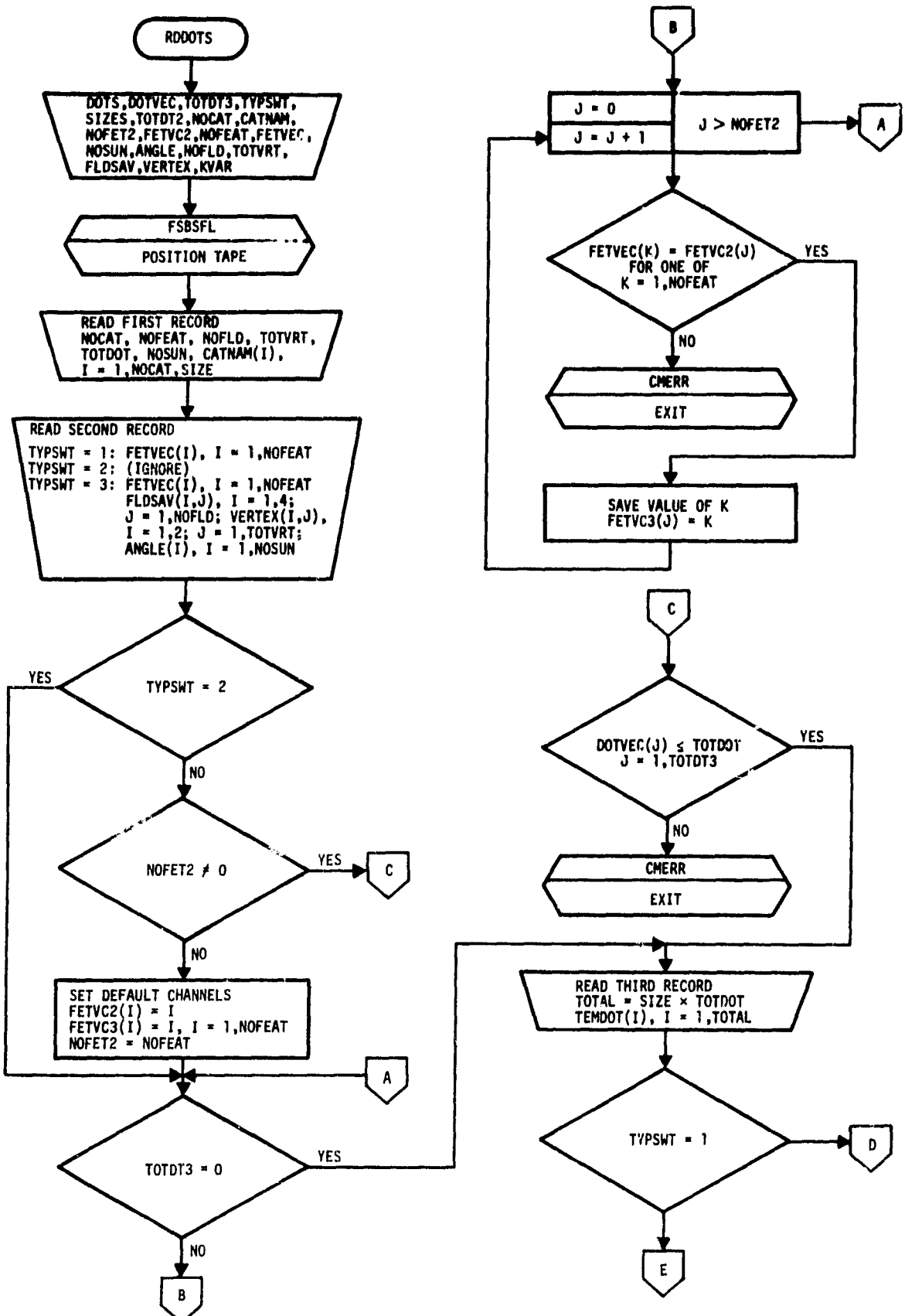

I4A1BN
FIELD LENGTH OUTPUT
LOOK FOR FIRST NONBLANK CHARACTER
A
SKIP ANY BLANKS BETWEEN PLUS OR MINUS SIGN AND FIRST DIGIT
BLANK DIGIT OR SPECIFIED
CHARACTER IS
NONSPECIFIED
SET ERROR FLAG
C
SET MINUS FLAG IF CHARACTER IS NEGATIVE
B
DIGIT
CHARACTER IS
NONDIGIT
SET ERROR FLAG
C
B
CONVERT DIGIT TO INTERNAL BINARY
C
WRITE ERROR MESSAGE IF FLAG IS SET
LAST CHARACTER
YES
NO
LOOK FOR NEXT NONBLANK CHARACTER
A
RETURN

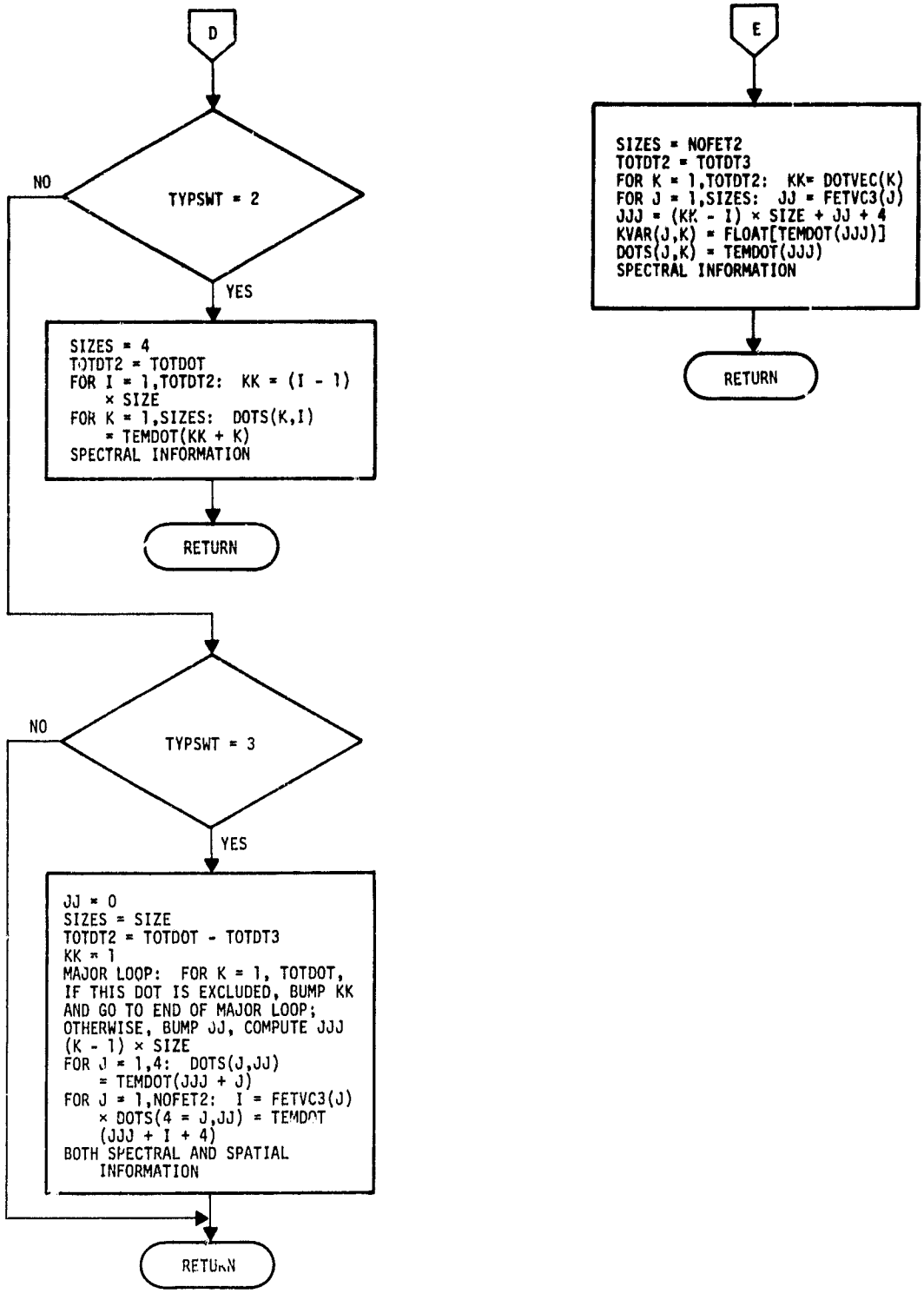

LABMAN
UNIT,FILE,NOCLS,TOTSUB,NOFEAT,
TOTFLD,TOTVRT,FETVEC,FLDSAV,
VERTEX,CLSNMS,NOSUBS,SUBNM,N,
STADRS,VARSIZ,PUNCH,SUBVEC,
PRNSTS,SWTCH
REWIND SAVTAP
FSBSFL
POSITION TAPE
PUNCH = 1
NO
YES
PUNCH HEADER CARD
PUNCH FIRST RECORD ON CARDS
WRITE FIRST RECORD ON SAVTAP
PUNCH = 1
NO
YES
PUNCH TRAINING FIELD
INFORMATION
WRITE TRAINING FIELD
INFORMATION ON SAVTAP
PUNCH = 1
NO
A
YES
PUNCH CLASS NAMES, NUMBER OF
SUBCLASSES IN EACH CLASS,
AND SUBCLASS NAMES
A
A
WRITE CLASS NAMES, NUMBER OF
SUBCLASSES IN EACH CLASS, AND
SUBCLASS NAMES ON THE SAVTAP
READ MEANS AND COVARIANCES
FROM DISK
PUNCH = 1
NO
YES
PUNCH NUMBER OF POINTS IN
SUBCLASSES, SUBCLASS MEANS,
AND COVARIANCES
WRITE NUMBER OF POINTS IN
SUBCLASSES, SUBCLASS MEANS,
AND COVARIANCES
PRNSTS = 1
NO
YES
INTERNAL ROUTINE COMMEN:
PRINT STATISTICS
WRITE SELECTED INFORMATION
RETURN

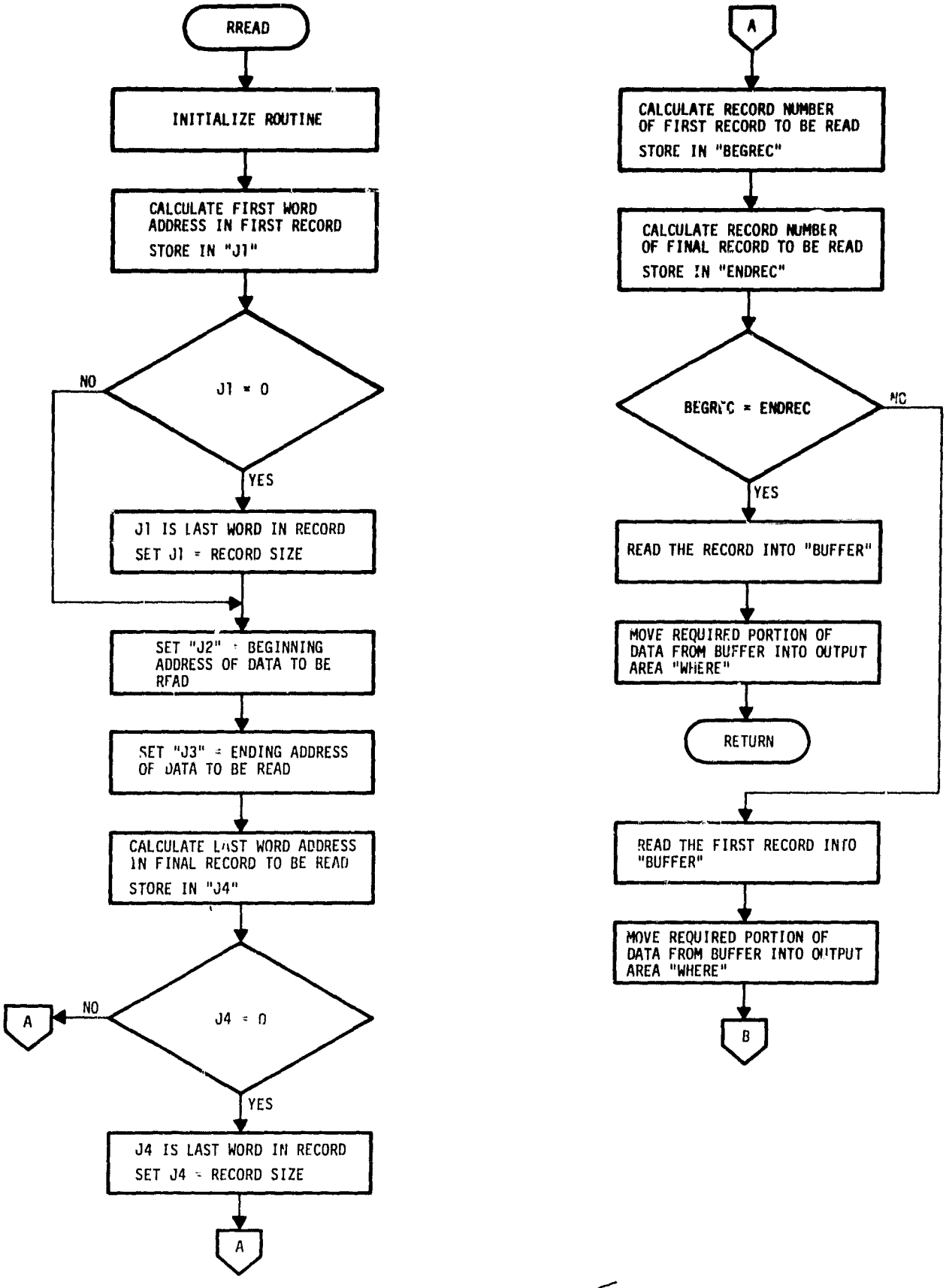

MATVEC
A,B,C,L,M
I ← 1
I ← I + 1
I > L
SUM ← 0
K ← 1
K ← K + 1
K > M
SUM ← SUM + A(I,K)
XB(K)
C(I) ← SUM
RETURN

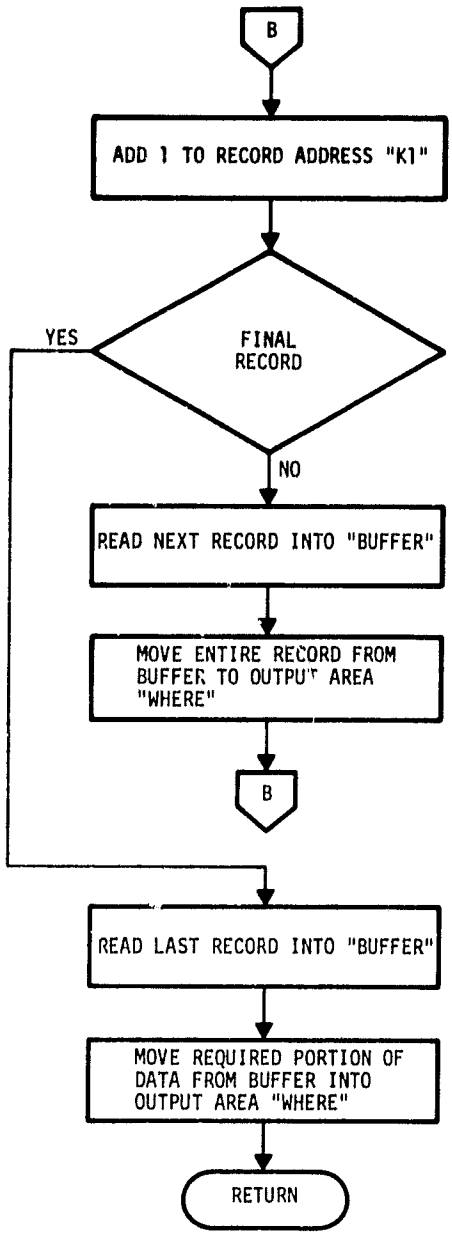

NAMSTA
SUBNAM,CATVEC,SUBNO,
NOSUB2,CATNAM,NOCAT
K = 0
I = 0
L = 0
I = I + 1
I > NOCAT
YES
NO
A
J = 0
J = J + 1
J > NOSUB2
YES
A
NO
NO
CATVEC(J) = I
YES
L = L + 1
K = K + 1
CONSTRUCT SUBNAM(K) FROM
FIRST THREE CHARACTERS OF
CATNAM(I) AND TWO NUMERIC
CHARACTERS REPRESENTING
THE VALUE OF L
NOCAT1 = NOCAT
I = NOCAT1
I = I - 1
I > 1
YES
NO
RETURN
NO
DELETE
NULL CATEGORIES
SUBNO(I)
= 0
YES
DELETE THIS CATEGORY
NOCAT = NOCAT - 1

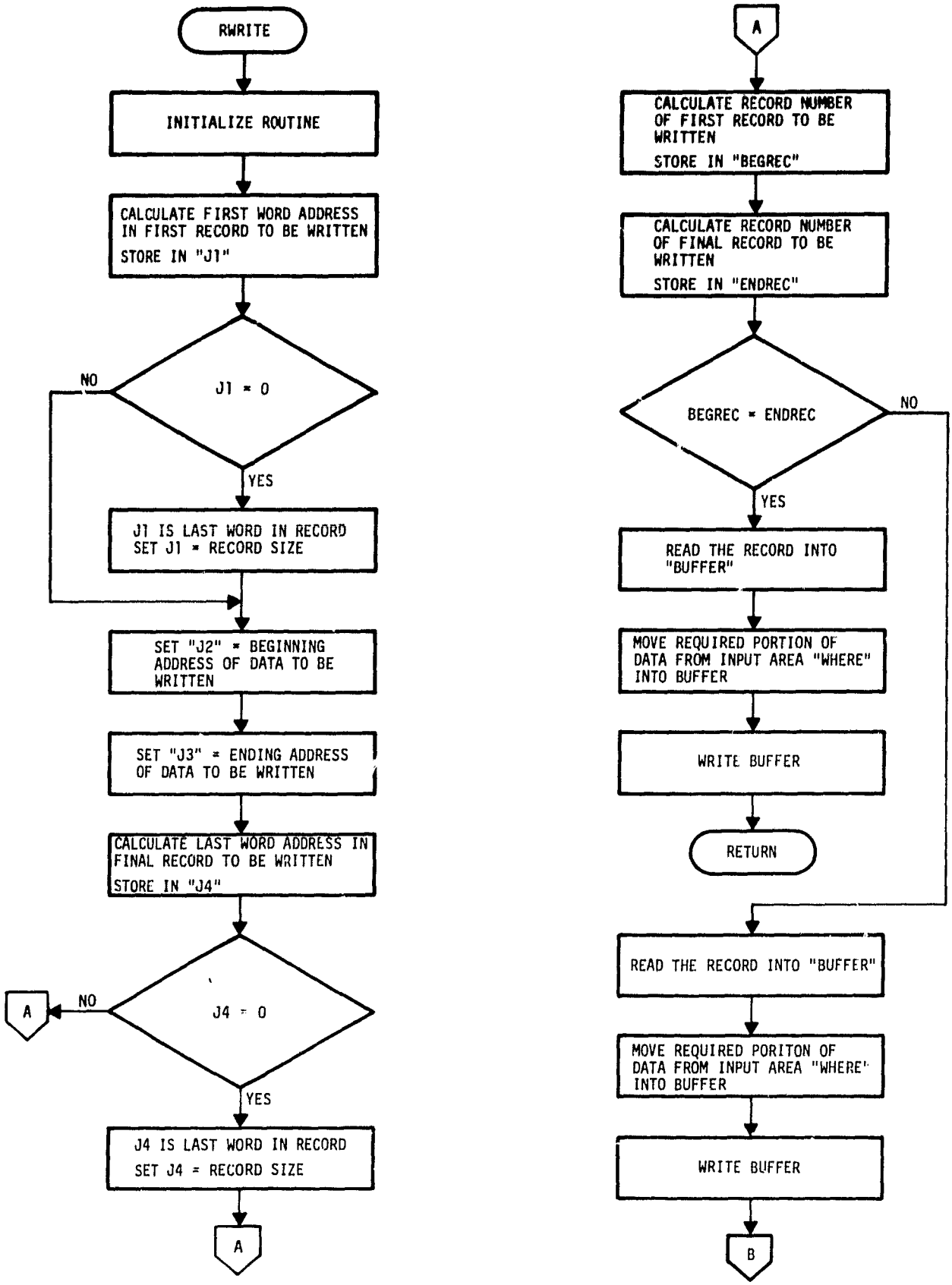

ORDER(VEC,N)
N
≥1
<1
RETURN
M ← N - 1
A
SWITCH ← FALSE
I = 1
I = I + 1
I
>M
B
≤M
VEC(I) ≤ VEC(I + 1)
TRUE
FALSE
TEMP ← VEC(I)
VEC(I) ← VEC(I + 1)
VEC(I + 1) ← TEMP
SWITCH ← TRUE
B
SWITCH
TRUE
A
FALSE
RETURN

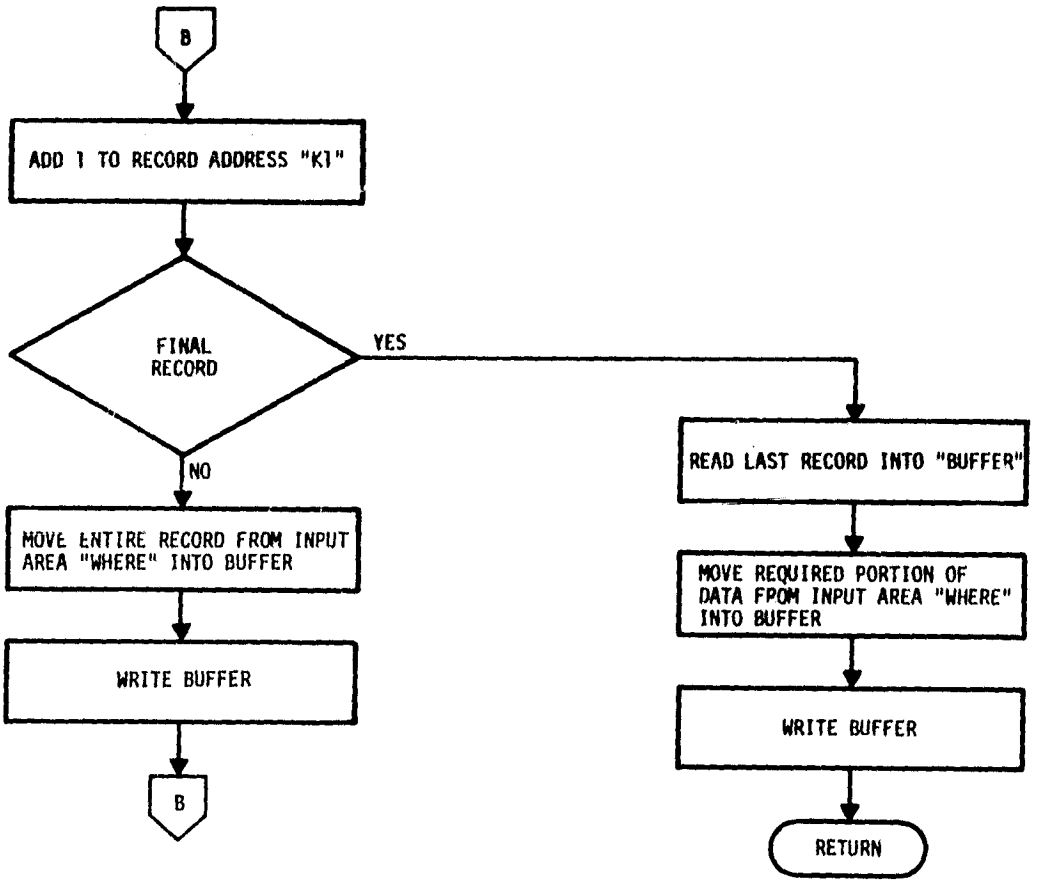

RDDOTS
DOTS,DOTVEC,TOTDT3,TYPSWT,
SIZES,TOTDT2,NOCAT,CATNAM,
NOFET2,FETVC2,NOFEAT,FETVEC,
NOSUN,ANGLE,NOFLD,TOTVRT,
FLDSAV,VERTEX,KVAR
FSBSFL
POSITION TAPE
READ FIRST RECORD
NOCAT, NOFEAT, NOFLD, TOTVRT,
TOTDOT, NOSUN, CATNAM(I),
I = 1,NOCAT,SIZE
READ SECOND RECORD
TYPSWT = 1: FETVEC(I), I = 1,NOFEAT
TYPSWT = 2: (IGNORE)
TYPSWT = 3: FETVEC(I), I = 1,NOFEAT
FLDSAV(I,J), I = 1,4;
J = 1,NOFLD; VERTEX(I,J),
I = 1,2; J = 1,TOTVRT;
ANGLE(I), I = 1,NOSUN
TYPSWT = 2
YES
NO
NOFET2 ≠ 0
YES
C
NO
SET DEFAULT CHANNELS
FETVC2(I) = I
FETVC3(I) = I, I = 1,NOFEAT
NOFET2 = NOFEAT
A
TOTDT3 = 0
YES
NO
B
B
J = 0
J = J + 1
J > NOFET2
A
FETVEC(K) = FETVC2(J)
FOR ONE OF
K = 1,NOFEAT
YES
NO
CMERR
EXIT
SAVE VALUE OF K
FETVC3(J) = K
C
DOTVEC(J) ≤ TOTDOT
J = 1,TOTDT3
YES
NO
CMERR
EXIT
READ THIRD RECORD
TOTAL = SIZE × TOTDOT
TEMDOT(I), I = 1,TOTAL
TYPSWT = 1
D
E

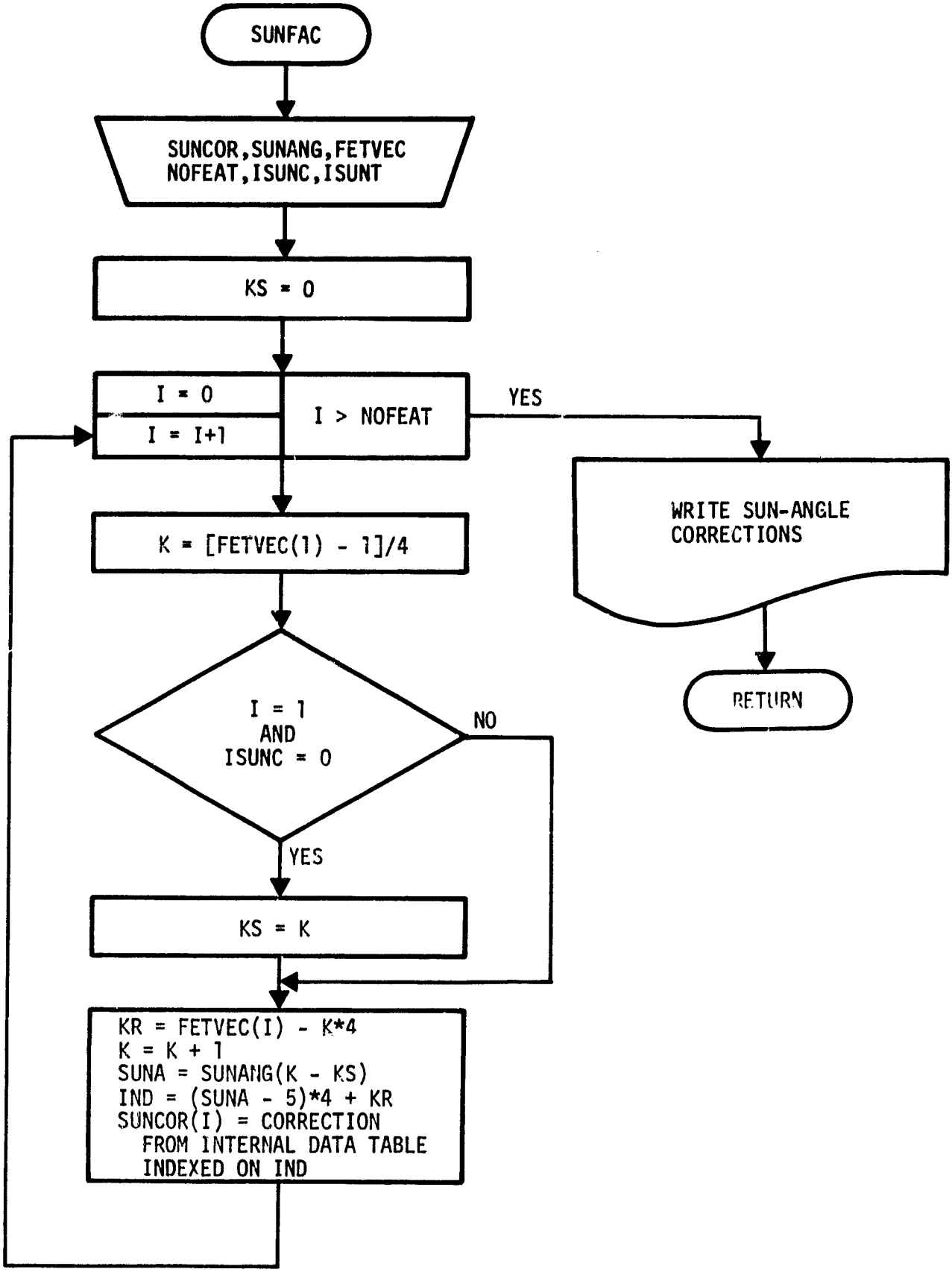

D
TYPSWT = 2
NO
YES
SIZES = 4
TOTDT2 = TOTDOT
FOR I = 1,TOTDT2: KK = (I - 1)
× SIZE
FOR K = 1,SIZES: DOTS(K,I)
= TEMDOT(KK + K)
SPECTRAL INFORMATION
RETURN
TYPSWT = 3
NO
YES
JJ = 0
SIZES = SIZE
TOTDT2 = TOTDOT - TOTDT3
KK = 1
MAJOR LOOP: FOR K = 1, TOTDOT,
IF THIS DOT IS EXCLUDED, BUMP KK
AND GO TO END OF MAJOR LOOP;
OTHERWISE, BUMP JJ, COMPUTE JJJ
(K - 1) × SIZE
FOR J = 1,4: DOTS(J,JJ)
= TEMDOT(JJJ + J)
FOR J = 1,NOFET2: I = FETVC3(J)
× DOTS(4 = J,JJ) = TEMDOT
(JJJ + I + 4)
BOTH SPECTRAL AND SPATIAL
INFORMATION
RETURN
E
SIZES = NOFET2
TOTDT2 = TOTDT3
FOR K = 1,TOTDT2: KK= DOTVEC(K)
FOR J = 1,SIZES: JJ = FETVC3(J)
JJJ = (KK - I) × SIZE + JJ + 4
KVAR(J,K) = FLOAT[TEMDOT(JJJ)]
DOTS(J,K) = TEMDOT(JJJ)
SPECTRAL INFORMATION
RETURN

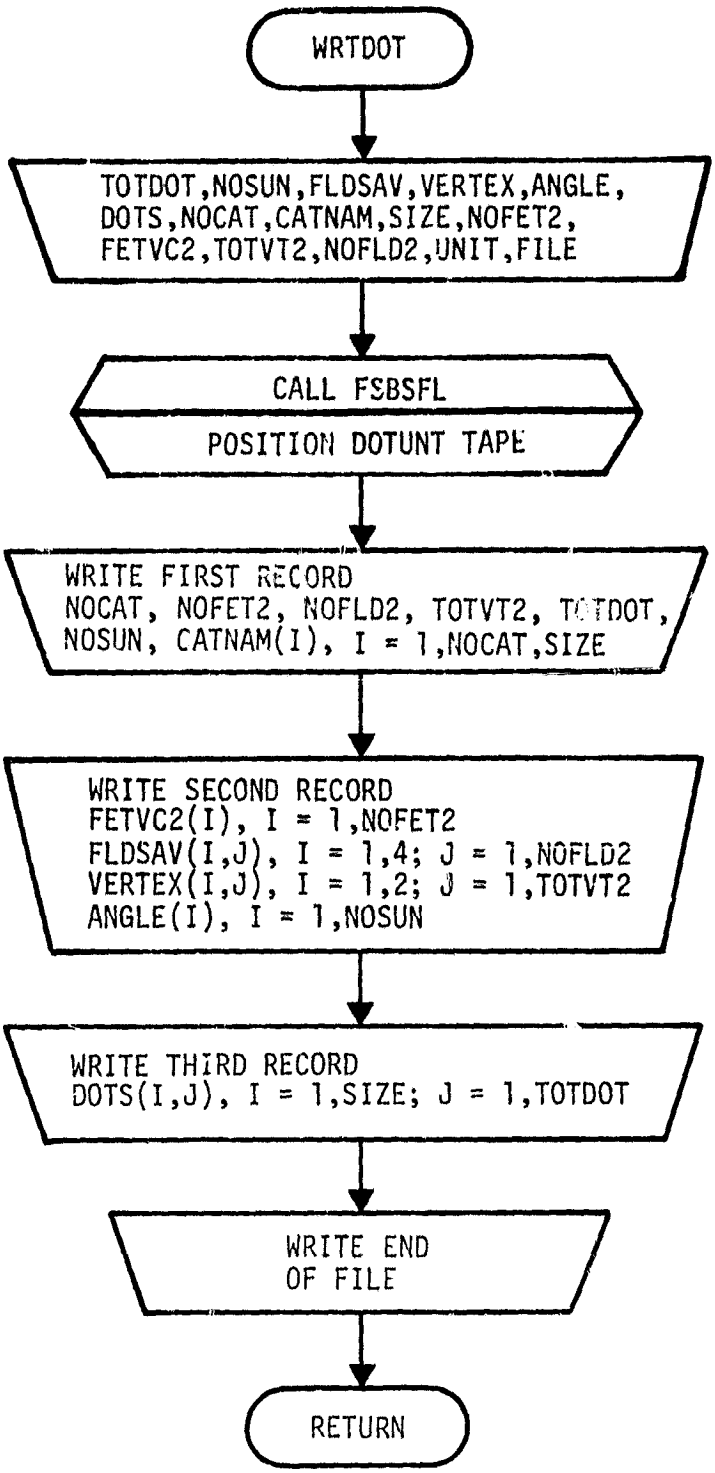

RREAD
INITIALIZE ROUTINE
CALCULATE FIRST WORD ADDRESS IN FIRST RECORD
STORE IN "J1"
J1 = 0
NO
YES
J1 IS LAST WORD IN RECORD
SET J1 = RECORD SIZE
SET "J2" = BEGINNING ADDRESS OF DATA TO BE READ
SET "J3" = ENDING ADDRESS OF DATA TO BE READ
CALCULATE LAST WORD ADDRESS IN FINAL RECORD TO BE READ
STORE IN "J4"
J4 = 0
NO
A
YES
J4 IS LAST WORD IN RECORD
SET J4 = RECORD SIZE
A
A
CALCULATE RECORD NUMBER OF FIRST RECORD TO BE READ
STORE IN "BEGREC"
CALCULATE RECORD NUMBER OF FINAL RECORD TO BE READ
STORE IN "ENDREC"
BEGREC = ENDREC
NO
YES
READ THE RECORD INTO "BUFFER"
MOVE REQUIRED PORTION OF DATA FROM BUFFER INTO OUTPUT AREA "WHERE"
RETURN
READ THE FIRST RECORD INTO "BUFFER"
MOVE REQUIRED PORTION OF DATA FROM BUFFER INTO OUTPUT AREA "WHERE"
B




RWRITE
INITIALIZE ROUTINE
CALCULATE FIRST WORD ADDRESS IN FIRST RECORD TO BE WRITTEN
STORE IN "J1"
J1 = 0
NO
YES
J1 IS LAST WORD IN RECORD
SET J1 = RECORD SIZE
SET "J2" = BEGINNING ADDRESS OF DATA TO BE WRITTEN
SET "J3" = ENDING ADDRESS OF DATA TO BE WRITTEN
CALCULATE LAST WORD ADDRESS IN FINAL RECORD TO BE WRITTEN
STORE IN "J4"
J4 = 0
NO
YES
A
J4 IS LAST WORD IN RECORD
SET J4 = RECORD SIZE
A
A
CALCULATE RECORD NUMBER OF FIRST RECORD TO BE WRITTEN
STORE IN "BEGREC"
CALCULATE RECORD NUMBER OF FINAL RECORD TO BE WRITTEN
STORE IN "ENDREC"
BEGREC = ENDREC
NO
YES
READ THE RECORD INTO "BUFFER"
MOVE REQUIRED PORTION OF DATA FROM INPUT AREA "WHERE" INTO BUFFER
WRITE BUFFER
RETURN
READ THE RECORD INTO "BUFFER"
MOVE REQUIRED PORITON OF DATA FROM INPUT AREA "WHERE" INTO BUFFER
WRITE BUFFER
B


B
ADD 1 TO RECORD ADDRESS "K1"
FINAL RECORD
YES
NO
READ LAST RECORD INTO "BUFFER"
MOVE REQUIRED PORTION OF DATA FROM INPUT AREA "WHERE" INTO BUFFER
WRITE BUFFER
RETURN
MOVE ENTIRE RECORD FROM INPUT AREA "WHERE" INTO BUFFER
WRITE BUFFER
B


SUNFAC
SUNCOR,SUNANG,FETVEC
NOFEAT,ISUNC,ISUNT
KS = 0
I = 0
I = I+1
I > NOFEAT
YES
WRITE SUN-ANGLE
CORRECTIONS
RETURN
K = [FETVEC(1) - 1]/4
I = 1
AND
ISUNC = 0
NO
YES
KS = K
KR = FETVEC(I) - K*4
K = K + 1
SUNA = SUNANG(K - KS)
IND = (SUNA - 5)*4 + KR
SUNCOR(I) = CORRECTION
FROM INTERNAL DATA TABLE
INDEXED ON IND


WRTDOT
TOTDOT,NOSUN,FLDSAV,VERTEX,ANGLE,
DOTS,NOCAT,CATNAM,SIZE,NOFET2,
FETVC2,TOTVT2,NOFLD2,UNIT,FILE
CALL FSBSFL
POSITION DOTUNT TAPE
WRITE FIRST RECORD
NOCAT, NOFET2, NOFLD2, TOTVT2, TOTDOT,
NOSUN, CATNAM(I), I = 1,NOCAT,SIZE
WRITE SECOND RECORD
FETVC2(I), I = 1,NOFET2
FLDSAV(I,J), I = 1,4; J = 1,NOFLD2
VERTEX(I,J), I = 1,2; J = 1,TOTVT2
ANGLE(I), I = 1,NOSUN
WRITE THIRD RECORD
DOTS(I,J), I = 1,SIZE; J = 1,TOTDOT
WRITE END
OF FILE
RETURN

20. DAMRG PROCESSOR SUBPROGRAMS

The DAMRG processor merges formatted MSS DATAPE files in one of three ways: channel, spatial, or line merge. All merging is based on user-specified fields, and output is in Universal or LARSYS III format. The DAMRG processor has two subprograms within the processor and uses 18 utility subprograms (documented in section 19). Figure 20-1 is a linkage diagram of the DAMRG processor.

DAMRG PROCESSOR

Subroutine level

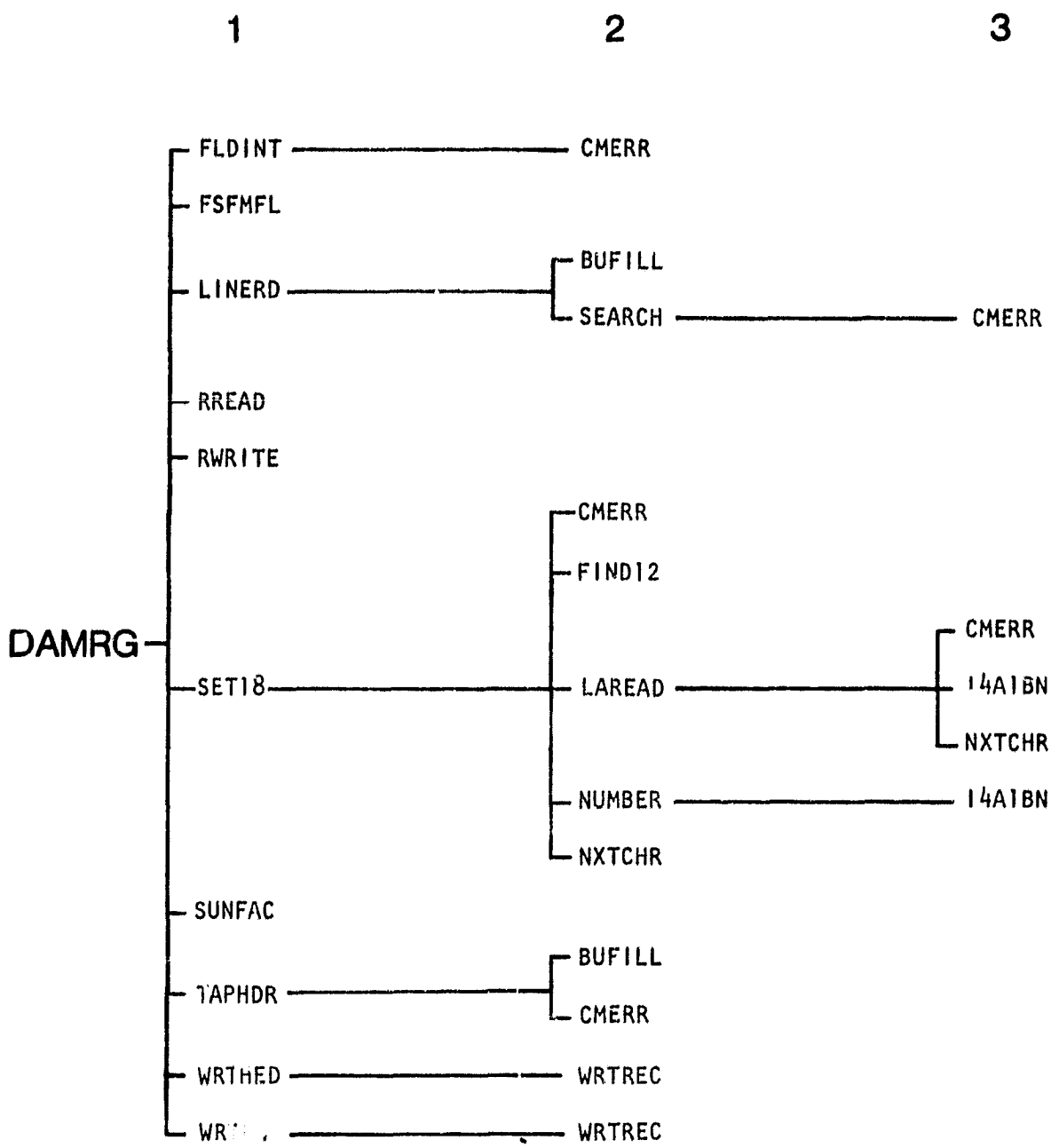


Figure 20-1.- Linkage diagram for the DAMRG processor.

20.1 DAMRG

The DAMRG subprogram is the driver routine for the DAMRG processor.

20.1.1 LINKAGES

This routine calls the FLDINT, FSFMFL, LINERD, RREAD, RWRITE, SET18, SUNFAC, TAPHDR, WRTHED, and WRTLN subprograms. It is called by MONTOR.

20.1.2 INTERFACES

The DAMRG subprogram interfaces with other routines through common blocks GLOBAL, ISOLNK, MRGDAT, TAPERD, and WRTAP and through the calling arguments.

20.1.3 INPUTS

Calling sequence: CALL DAMRG(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.

20.1.4 OUTPUTS

This subprogram outputs an MSS data file in either LARSYS III or Universal format.

20.1.5 STORAGE REQUIREMENTS

This subprogram requires 4600 bytes of storage.

20.1.6 DESCRIPTION

The DAMRG subprogram calls SET18 to obtain decoded input data and then processes each input file in order of occurrence in the control card image file. Each input file is written to the random-access disk file. After all input files have been processed, the output MSS data file is written. (The detailed processing steps are set out in the subprogram listing, volume IV, section 20.)

20.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 20.3.

20.1.8 LISTING

The subprogram listing is provided in volume IV, section 20.

20.2 SET18

The SET18 subprogram reads and analyzes control card and field definition card images.

20.2.1 LINKAGES

This routine calls the CMERR, FIND12, LAREAD, NUMBER, and NXTCHR subprograms. It is called by the DAMRG driver routine.

20.2.2 INTERFACES

The SET18 subprogram interfaces with other routines through common blocks GLOBAL, ISOLNK, MRGDAT, TAPERD, and WRTAP and through the calling arguments.

20.2.3 INPUTS

Calling sequence: CALL SET18(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	In/out	See section 20.1.3 for a description of these parameters.
TOP	1	In	

The control and field definition card images relevant to this routine are given in section 20 (table 20-1) and section 3.2.3, respectively, of volume II of this user guide.

20.2.4 OUTPUTS

The results are returned for use by the calling routine.

20.2.5 STORAGE REQUIREMENTS

This subprogram requires 6126 bytes of storage.

20.2.6 DESCRIPTION

The SET18 subprogram processes control card and field definition card images provided by the user and initializes variables in the GLOBAL, ISOLNK, and MRGDAT common blocks.

SET18 reads, stores, and compares the first four characters of each control card image against a data array. When a match is found, transfer is made to the corresponding code to decode information starting at column 11.

A count will be made in all cards of the form

DATAPE INPUT/_ _ _

to determine the number of input files. Upon encountering the *END delimiter, the field definition card images will be read and decoded by repeated calls to LAREAD (one call if channel or pseudomerge option). The $END card delimits card input and triggers a return to the calling routine.

Continuation of information from one card image to another will occur only on LINES control cards (pseudomerge option). The way of handling these card images is sketched as follows:

NOLINE = 0 (initially)

Upon encountering a LINES card image,

NOLINE = NUMBER(CARD,COL,LINES,NOLINE)

This keeps a running total of lines read in and stores line numbers in LINES. Lines associated with input files will be discerned by use of NLINES(6), the count for each file in order of input.

20.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 20.3.

20.2.8 LISTING

The subprogram listing is provided in volume IV, section 20.

20.3 SUBPROGRAM FLOW CHARTS

No flow charts are available for the DAMRG processor.

21. GTDDM PROCESSOR SUBPROGRAMS

The GTDDM processor accepts the converted ground-truth crop code file generated by the GTTCN processor as input and labels the 209 dots. It generates a LACIE-formatted dot card image file for each ground-truth image file. This processor calls 8 dedicated routines and 11 utility subprograms. Figure 21-1 is a linkage diagram of the GTDDM processor.

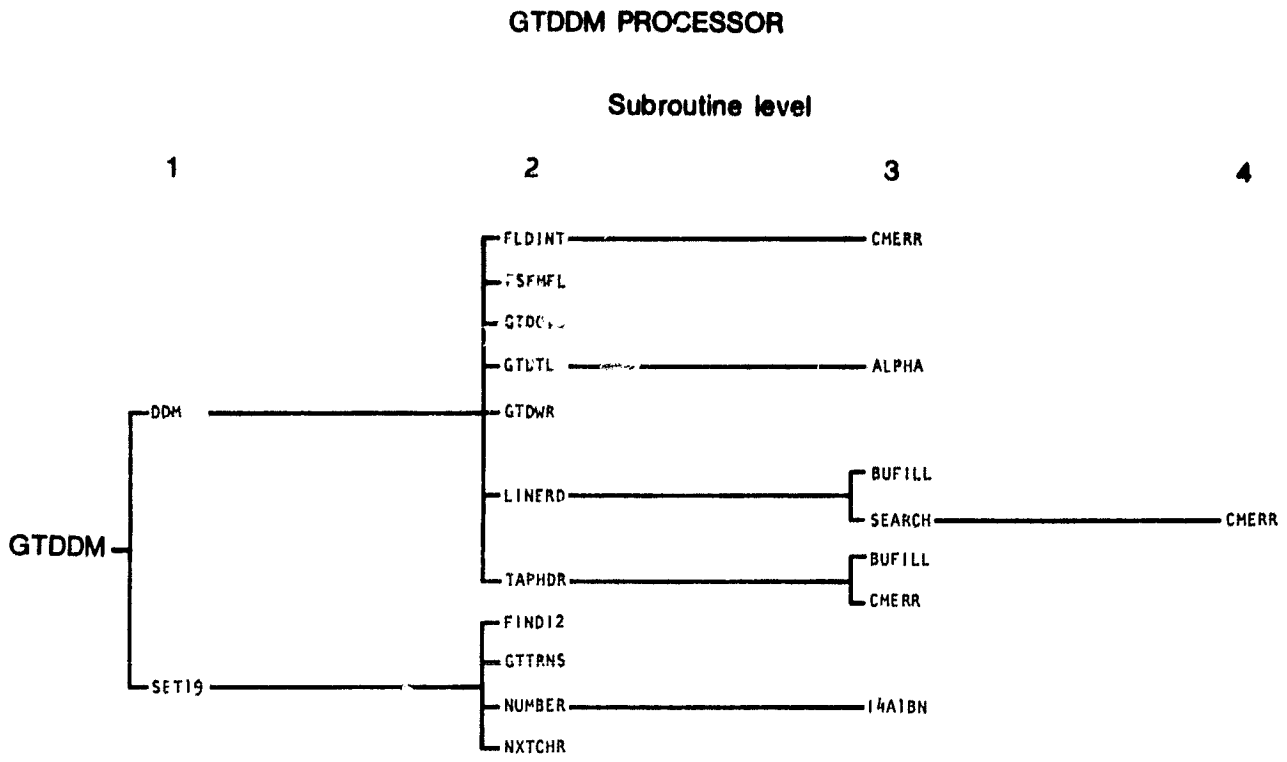


Figure 21-1.- Linkage diagram for the GTDDM processor.

21.1 GTDDM

The GTDDM subprogram is the driver routine for the GTDDM processor.

21.1.1 LINKAGES

This routine calls the DDM and SET19 subprograms. It is called by MONTOR.

21.1.2 INTERFACES

The GTDDM subprogram interfaces with other routines through the calling arguments.

21.1.3 INPUTS

Calling sequence: CALL GTDDM(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In/out	Maximum usable storage in ARRAY; TOP = 10 600.

21.1.4 OUTPUTS

The GTDDM subprogram generates a LACIE-formatted dot card image file. (See "As-Built" Design Specification for LACIE-Formatted Dot Cards in EOD-LARSYS, JSC-13972, LEC-12154, April 1978.)

21.1.5 STORAGE REQUIREMENTS

This subprogram requires approximately 1600 bytes of storage.

21.1.6 DESCRIPTION

The GTDDM subprogram calls the SET19 subprogram to read and analyze control cards and set options and the DDM subprogram to label the 209 dots from the ground-truth image file.

21.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 21.9.

21.1.8 LISTING

The subprogram listing is provided in volume IV, section 21.

21.2 ALPHA

The ALPHA function accepts a symbol S as input and checks to see if S is an alphabetic character. As soon as S is found as a valid alphabetic character, the integer value of the character is returned in ALPHA. If S is not an alphabetic character, an error message is generated.

21.2.1 LINKAGES

The ALPHA function does not call any other subprogram. It is called by the GTDTL subprogram.

21.2.2 INTERFACES

The ALPHA function interfaces with other routines through the calling arguments.

21.2.3 INPUTS

Calling sequence: ALPHA(S)

Parameter	Dimension	In/out	Definition
S	1	In	Input symbol.

21.2.4 OUTPUTS

The results are returned for use by the calling routine.

21.2.5 STORAGE REQUIREMENTS

This subprogram requires approximately 2400 bytes of storage.

21.2.6 DESCRIPTION

Not required.

21.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 21.9.

21.2.8 LISTING

The subprogram listing is provided in volume IV, section 21.

21.3 DDM

The DDM subprogram processes and labels the 209 dots provided by the GTTCN processor.

21.3.1 LINKAGES

This routine calls the FLDINT, FSFMFL, GTDOTS, GTDTL, GTDWR, LINERD, and TAPHDR subprograms. It is called by the GTDDM driver routine.

21.3.2 INTERFACES

The DDM subprogram interfaces with other routines through common blocks GTBK and TAPERD and through the calling arguments.

21.3.3 INPUTS

Input to the DDM subprogram consists of the GTRDU and GTWRU files output by the GTTCN processor.

Calling sequence: CALL DDM(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	In/out	See section 21.1.3 for a description of these parameters.
TOP	1	In	

21.3.4 OUTPUTS

This subprogram outputs the labeled dot data file.

21.3.5 STORAGE REQUIREMENTS

This subprogram requires approximately 5600 bytes of storage.

21.3.6 DESCRIPTION

The DDM subprogram reads the input crop code file one line at a time and, for each line containing pixels on the LACIE grid, adds this information to the LACIE-formatted ground-truth file which is output.

21.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 21.9.

21.3.8 LISTING

The subprogram listing is provided in volume IV, section 21.

21.4 GTDOTS

For each line on the LACIE grid, the GTDOTS subprogram extracts crop code values for samples on the LACIE grid (input in array IDATA) and stores them in array DMTX.

21.4.1 LINKAGES

This routine does not call any other subprogram. It is called by the DDM subprogram.

21.4.2 INTERFACES

The GTDOTS subprogram interfaces with other routines through common block GTBK and through the calling arguments.

21.4.3 INPUTS

Calling sequence: CALL GTDOTS(IDATA,DMTX,LINE)

Parameter	Dimension	In/out	Definition
IDATA	1	In	Array containing input scan line of crop codes.
DMTX	11,19	Out	Array containing extracted crop codes for the LACIE dots on the scan line.
LINE	1	Out	Counter incremented by 1 upon entry to GTDOTS; it is the first index for the DMTX array.

21.4.4 OUTPUTS

Not applicable.

21.4.5 STORAGE REQUIREMENTS

This subprogram requires approximately 1600 bytes of storage.

21.4.6 DESCRIPTION

Not required.

21.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 21.9.

21.4.8 LISTING

The subprogram listing is provided in volume IV, section 21.

21.5 GTDTL

The GTDTL subprogram replaces the crop codes in the DMTX array with one-character names using the transformation table TRNS1 in common block TR. It also builds the tables TRNS2 and TRNS3, which are used to print out the names of existing categories.

21.5.1 LINKAGES

This routine calls the ALPHA subprogram. It is called by the DDM subprogram.

21.5.2 INTERFACES

The GTDTL subprogram interfaces with other routines through common block TR and through the calling arguments.

21.5.3 INPUTS

Calling sequence: CALL GTDTL(DMTX,NSYM)

Parameter	Dimension	In/out	Definition
DMTX	11,19	In/out	Array containing extracted crop codes for the LACIE dots on the scan line.
NSYM	1	Out	Number of categories found.

21.5.4 OUTPUTS

This subprogram outputs the number of categories.

21.5.5 STORAGE REQUIREMENTS

This subprogram requires approximately 3200 bytes of storage.

21.5.6 DESCRIPTION

Not required.

21.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 21.9.

21.5.8 LISTING

The subprogram listing is provided in volume IV, section 21.

21.6 GTDWR

The GTDWR subprogram accepts a matrix of dot labels and a type mask matrix as input and outputs LACIE-formatted dot files on both the printer and the ground-truth output unit.

21.6.1 LINKAGES

This routine does not call any other subprogram. It is called by the DDM subprogram.

21.6.2 INTERFACES

The GTDWR subprogram interfaces with other routines through common blocks GTBK and TR and through the calling arguments.

21.6.3 INPUTS

Calling sequence: CALL GTDWR(DMTX,TYPE,NSYM)

Parameter	Dimension	In/out	Definition
DMTX	11,19	In	Matrix of one-character dot labels.
TYPE	1	In	Key indicating type mask: if = 1, transition year; if = 2, Phase III; or if = 3, user input.
NSYM	1	In	Number of categories found.

21.6.4 OUTPUTS

This subprogram outputs LACIE-formatted dot data on the line printer and the ground-truth output unit.

21.6.5 STORAGE REQUIREMENTS

This subprogram requires approximately 4000 bytes of storage.

21.6.6 DESCRIPTION

Not required.

21.6.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 21.9.

21.6.8 LISTING

The subprogram listing is provided in volume IV, section 21.

21.7 GTTRNS

The GTTRNS subprogram constructs the default transformation of crop code numbers to one-character crop names.

21.7.1 LINKAGES

This routine does not call any other subprogram. It is called by the SET19 subprogram.

21.7.2 INTERFACES

The GTTRNS subprogram interfaces with other routines through common block TR.

21.7.3 INPUTS

Calling sequence: CALL GTTRNS

21.7.4 OUTPUTS

The results are returned for use by the calling routine.

21.7.5 STORAGE REQUIREMENTS

This subprogram requires approximately 2400 bytes of storage.

21.7.6 DESCRIPTION

Not required.

21.7.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 21.9.

21.7.8 LISTING

The subprogram listing is provided in volume IV, section 21.

21.8 SET19

The SET19 subprogram reads and analyzes control card images and sets options.

21.8.1 LINKAGES

This routine calls the FIND12, GTTRNS, NUMBER, and NXTCHR subprograms. It is called by the GTDDM driver routine.

21.8.2 INTERFACES

The SET19 subprogram interfaces with other routines through common blocks GLOBAL, GTBK, INFORM, TAPERD, and TR.

21.8.3 INPUTS

Calling sequence: CALL SET19

The control cards relevant to this routine are given in section 21 (table 21-1) of volume II of this user guide.

21.8.4 OUTPUTS

This subprogram outputs a list of user-requested options.

21.8.5 STORAGE REQUIREMENTS

This subprogram requires approximately 12 800 bytes of storage.

21.8.6 DESCRIPTION

Not required.

21.8.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 21.9.

21.8.8 LISTING

The subprogram listing is provided in volume IV, section 21.

21.9 SUBPROGRAM FLOW CHARTS

No flow charts are provided for the GTDDM processor.

22. GTTCN PROCESSOR SUBPROGRAMS

The GTTCN processor is a ground-truth file conversion routine. It accepts Universal-formatted 351-sample by 392-line ground-truth image files as input, reduces the data to 117 samples by 196 lines, and outputs the results in Universal format. The GTTCN processor utilizes 6 routines that are exclusive to the processor and 14 utility subprograms. Figure 22-1 is a linkage diagram of the GTTCN processor.

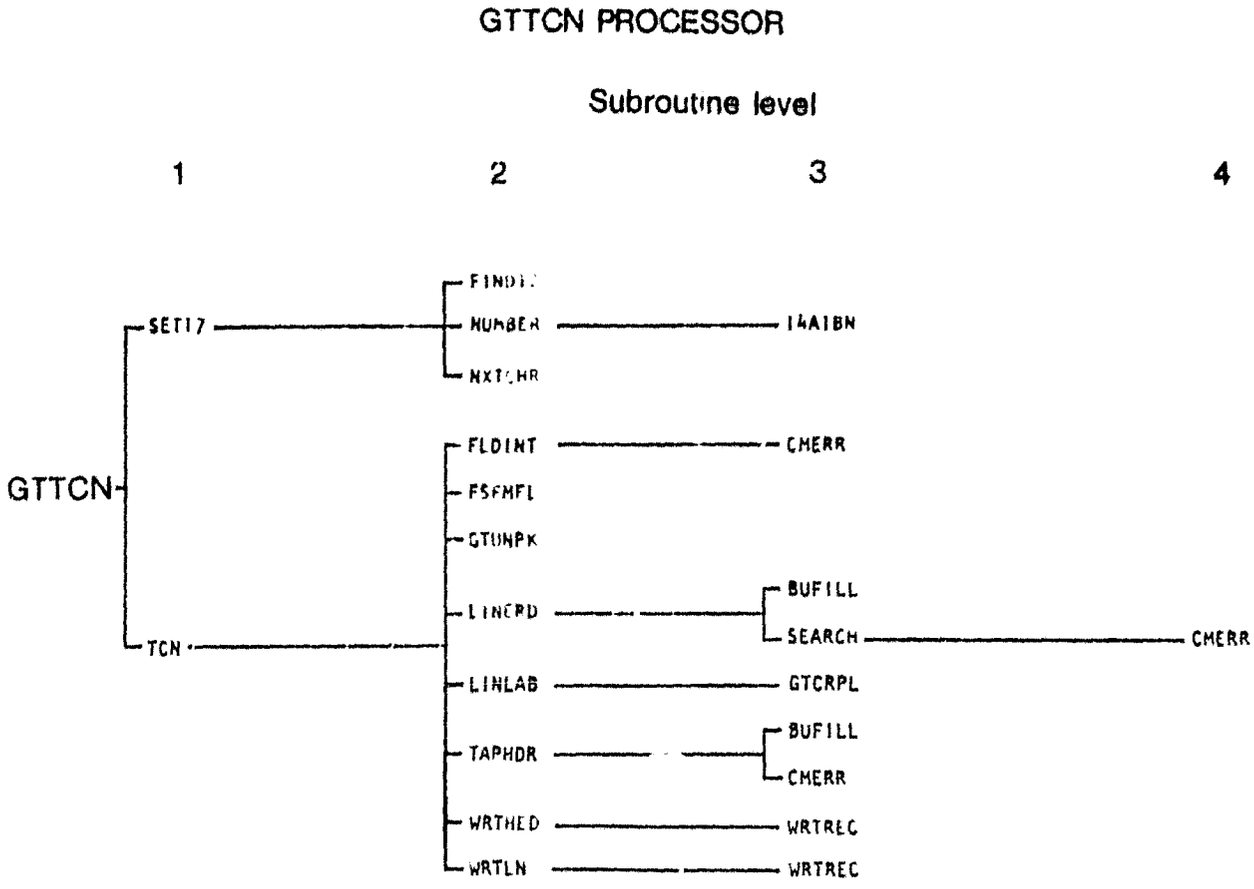


Figure 22-1.- Linkage diagram for the GTTCN processor.

22.1 GTTCN

The GTTCN subprogram is the driver routine for the GTTCN processor.

22.1.1 LINKAGES

This routine calls the SET17 and TCN subprograms. It is called by MONTOR.

22.1.2 INTERFACES

The GTTCN subprogram interfaces with other routines through the calling arguments.

22.1.3 INPUTS

Calling sequence: CALL GTTCN(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In/out	Maximum usable storage in ARRAY; TOP = 10 600.

22.1.4 OUTPUTS

Not applicable.

22.1.5 STORAGE REQUIREMENTS

This subprogram requires approximately 1600 bytes of storage.

22.1.6 DESCRIPTION

The GTTCN subprogram calls the SET17 subprogram to analyze control cards and set options and the TCN subprogram to perform the ground-truth tape conversion.

22.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 22.7.

22.1.8 LISTING

The subprogram listing is provided in volume IV, section 22.

22.2 GTCRPL

The GTCRPL subprogram examines the six subpixel labels and labels the pixel.

22.2.1 LINKAGES

This routine does not call any other subprogram. It is called by the LINLAB subprogram.

22.2.2 INTERFACES

The GTCRPL subprogram interfaces with other routines through common block GTBK and through the calling arguments.

22.2.3 INPUTS

Calling sequence: CALL GTCRPL(CROP,MT,NC)

Parameter	Dimension	In/out	Definition
CROP	1	Out	Crop code value.
MT	6	In	Array containing subpixel crop code numbers.
NC	1	Out	Counter which provides maximum number of subpixels having the same crop code.

22.2.4 OUTPUTS

The results are returned for use by the calling routine.

22.2.5 STORAGE REQUIREMENTS

This subprogram requires approximately 2400 bytes of storage.

22.2.6 DESCRIPTION

Not required.

22.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 22.7.

22.2.8 LISTING

The subprogram listing is provided in volume IV, section 22.

22.3 GTUNPK

The GTUNPK subprogram converts the crop code to a numerical designation.

22.3.1 LINKAGES

This routine does not call any other subprogram. It is called by the TCN subprogram.

22.3.2 INTERFACES

The GTUNPK subprogram interfaces with other routines through the calling arguments.

22.3.3 INPUTS

Calling sequence: CALL GTUNPK(IDATA,N1,N2,OS,NRPDS,LENGTH)

Parameter	Dimension	In/out	Definition
IDATA	1	In/out	Array containing three lines of data from the subpixel-level ground-truth file.
N1	1	In	Starting count number for subpixel modification; set = 1.
N2	1	In	Passed as the variable INPX, which is set = 392 (number of subpixels on a line).
OS	1	In/out	Offset on the index in IDATA; set initially = 72.
NRPDS	1	In	Number of records per data set.
LENGTH	1	In	Constant offset of 540 (length of output data records). In Universal format, the data record length in bytes must be divisible by 180.

22.3.4 OUTPUTS

The results are returned for use by the calling routine.

22.3.5 STORAGE REQUIREMENTS

This subprogram requires 2400 bytes of storage.

22.3.6 DESCRIPTION

The numerical crop code designation is calculated in the following manner: If the input number is greater than 128, 128 is subtracted from the input number; if the input number is less than or equal to 128, 128 is added to the input number.

22.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 22.7.

22.3.8 LISTING

The subprogram listing is provided in volume IV, section 22.

22.4 LINLAB

For each three lines of input data, the LINLAB subprogram produces one line of output data, with the crop code determined by the GTCRPL subprogram.

22.4.1 LINKAGES

This routine calls the GTCRPL subprogram. It is called by the TCN subprogram.

22.4.2 INTERFACES

The LINLAB subprogram interfaces with other routines through the calling arguments.

22.4.3 INPUTS

Calling sequence: CALL LINLAB(IDATA,MAXREC)

Parameter	Dimension	In/out	Definition
IDATA	3060	In/out	Array containing three lines of data as input; as output, the first line is replaced by a line of converted data and the remaining two lines remain as inputs.
MAXREC	1	In	Maximum record length; corresponds to GTREC, which is set = 540.

22.4.4 OUTPUTS

This subprogram outputs pixel values for one scan line in IDATA.

22.4.5 STORAGE REQUIREMENTS

This subprogram requires approximately 2400 bytes of storage.

22.4.6 DESCRIPTION

Not required.

22.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 22.7.

22.4.8 LISTING

The subprogram listing is provided in volume IV, section 22.

22.5 SET17

The SET17 subprogram analyzes control card images and sets options.

22.5.1 LINKAGES

This routine calls the FIND12, NUMBER, and NXTCHR subprograms and the Fortran REREAD subprogram. It is called by the GTTCN driver routine.

22.5.2 INTERFACES

The SET17 subprogram interfaces with other routines through common blocks GLOBAL, GTBK, INFORM, and TAPERD.

22.5.3 INPUTS

Calling sequence: CALL SET17

The control cards relevant to this routine are given in section 22 (table 22-1) of volume II of this user guide.

22.5.4 OUTPUTS

This subprogram outputs an input summary, a list of user-requested options, and a list of files to be converted.

22.5.5 STORAGE REQUIREMENTS

This subprogram requires approximately 8000 bytes of storage.

22.5.6 DESCRIPTION

The SET17 subprogram accepts card image and tape input. It analyzes each control card and generates a message if an error occurs. An input summary and lists of user-requested options and files to be converted are generated.

22.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 22.7.

22.5.8 LISTING

The subprogram listing is provided in volume IV, section 22.

22.6 TCN

The TCN subprogram converts Universal-formatted 351-sample by 392-line ground-truth image tape files to Universal-formatted 117-sample by 196-line files.

22.6.1 LINKAGES

This routine calls the FLDINT, FSFMFL, GTUNPK, LINERD, LINLAB, TAPHDR, WRTHED, and WRTLN subprograms. It is called by the GTTCN driver routine.

22.6.2 INTERFACES

The TCN subprogram interfaces with other routines through common blocks GTBK, TAPERD, and WRTAP and through the calling arguments.

22.6.3 INPUTS

Input to the TCN subprogram consists of the MSS DATAPE in Universal format.

Calling sequence: CALL TCN(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	See section 22.1.3 for a description of these parameters.
TOP	1	In	

22.6.4 OUTPUTS

This subprogram outputs a Universal-formatted image tape.

22.6.5 STORAGE REQUIREMENTS

This subprogram requires approximately 4800 bytes of storage.

22.6.6 DESCRIPTION

The conversion of the 351-sample by 392-line ground-truth image files to 117-sample by 196-line image files requires a labeling procedure for mapping six subpixel classifications as one pixel. The user may select any or all of the six subpixels in the labeling process. Pixels are labeled by majority rule, with the first label taking precedence in the event of a tie. The TCN subprogram calls utility subprograms FLDINT, FSFMFL, LINERD, and TAPHDR to read in image data; subprograms GTUNPK and LINLAB to perform the conversion; and utility subprograms WRTHED and WRTLN to output the reduced image files.

22.6.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 22.7.

22.6.8 LISTING

The subprogram listing is provided in volume IV, section 22.

22.7 SUBPROGRAM FLOW CHARTS

No flow charts are provided for the GTTCN processor.

23. TESTSP PROCESSOR SUBPROGRAMS

The TESTSP processor is an iterative self-organizing clustering procedure which uses sample values of pixels clustered in packed form on disk storage. Basically, TESTSP performs the same functions as the ISOCLS processor except that the storage required by TESTSP is only one-fourth of that used by ISOCLS. TESTSP utilizes 5 processor subprograms and 43 utility subprograms (documented in section 19). Figure 23-1 is a linkage diagram for the TESTSP processor.

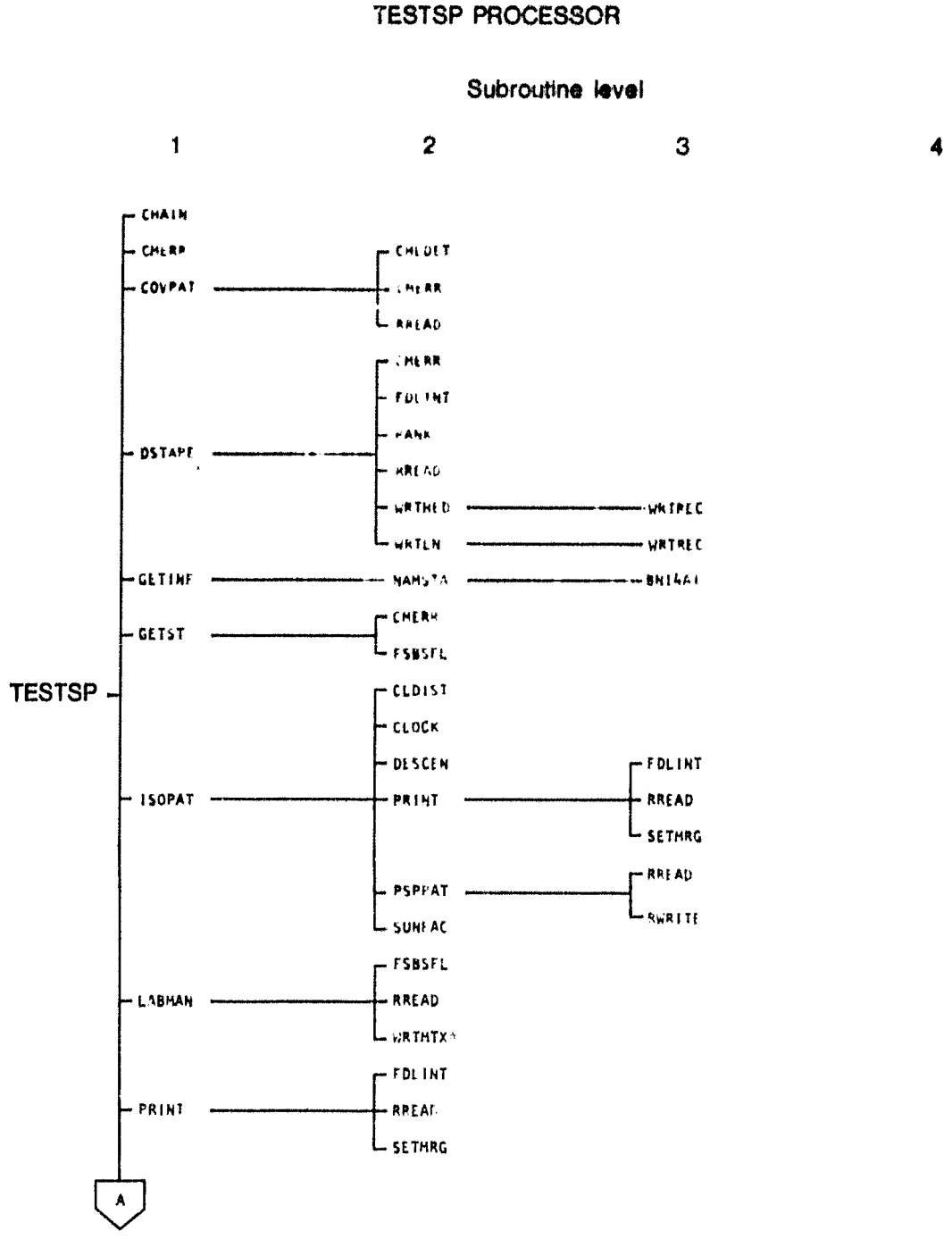
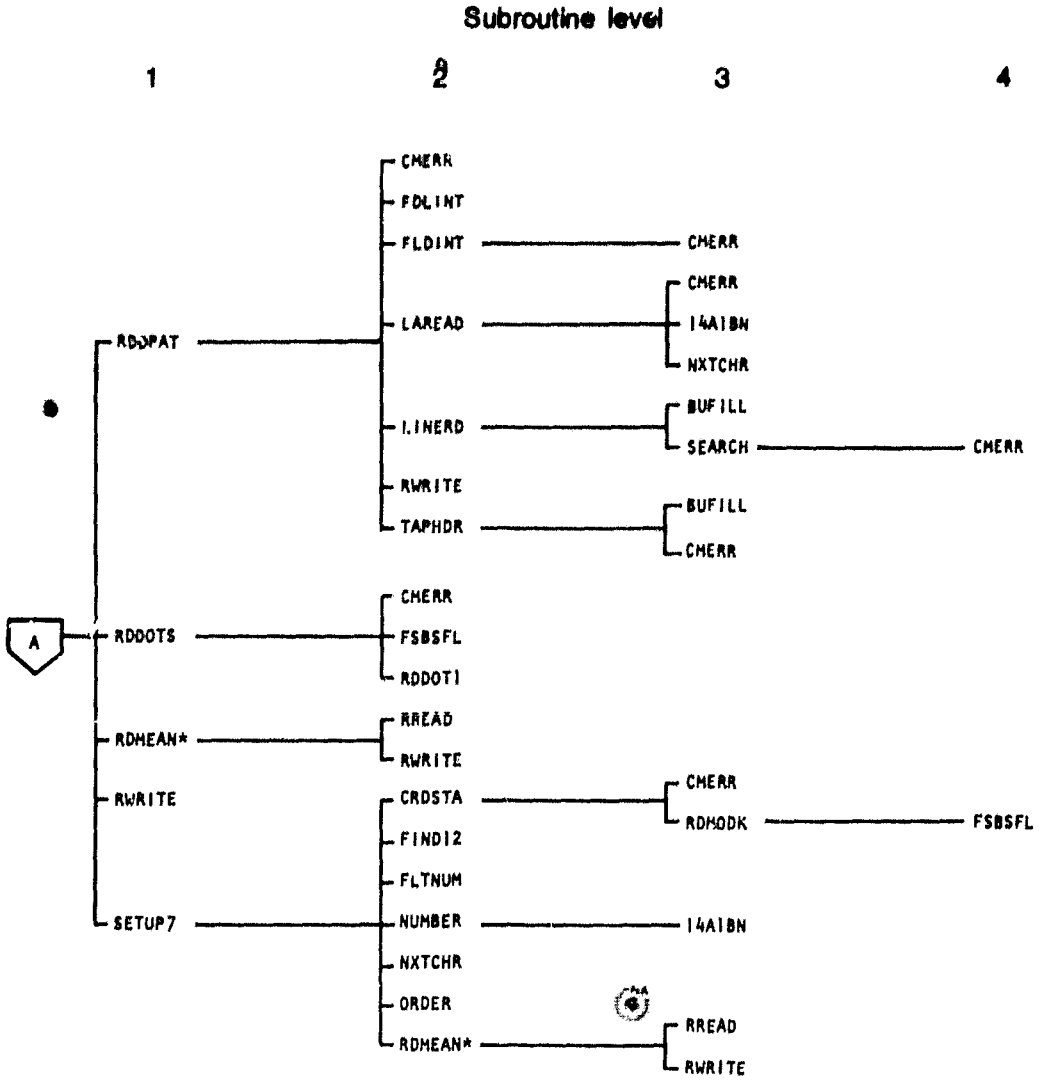


Figure 23-1.- Linkage diagram for the TESTSP processor.



*Entry point is RDFILE.

Figure 23-1.- Concluded.

23.1 TESTSP

The TESTSP subprogram performs a modified version of the clustering algorithm (ISOCLS, section 9.1). It is the driver routine for the TESTSP processor.

23.1.1 LINKAGES

The TESTSP routine calls the CHAIN, CMERR, COVPAT, DSTAPE, GETINF, GETST, ISOPAT, LABMAN, PRINT, RDDOTS, RDDPAT, RDMEAN, RWRITE, and SETUP7 subprograms. It is called by MONPAC, which is a modified version of the MONTOR system monitor for the EOD-LARSYS.

23.1.2 INTERFACES

The TESTSP subprogram interfaces with other routines through common blocks GLOBAL, ISOLNK, and PASS and through the calling arguments.

23.1.3 INPUTS

Calling sequence: CALL TESTSP(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In/out	Maximum usable storage in ARRAY; TOP = 10 600.

23.1.4 OUTPUTS

This subprogram outputs the SAVTAP file and (optionally) creates a MAPUNT file.

23.1.5 STORAGE REQUIREMENTS

This subprogram requires 8000 bytes of storage.

23.1.6 DESCRIPTION

With the exception of storage space allocated, the TESTSP subprogram performs the same operations as the ISOCLS subprogram (section 9.1). Processor subprograms COVPAT, ISOPAT, PSPPAT, and RDDPAT are modified renditions of the ISOCLS subprograms COVAR1, ISODAT, PSPLIT, and RDDATA, respectively.

23.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 23.6.

23.1.8 LISTING

The subprogram listing is provided in volume IV, section 23.

23.2 COVPAT

The COVPAT subprogram calculates and prints the covariance matrix for each cluster.

23.2.1 LINKAGES

This routine calls the CHLDET, CMERR, and RREAD subprograms. It is called by the TESTSP driver routine.

23.2.2 INTERFACES

The COVPAT subprogram interfaces with other routines through common blocks GLOBAL and PASS and through the calling arguments.

23.2.3 INPUTS

Calling sequence: CALL COVPAT(COVAR,IDAT1,IPLACE,MEANS,N,IBAD)

Parameter	Dimension	In/out	Definition
COVAR	VARSIZ,LNCAT	Out	Array containing covariance matrix for each cluster; VARSIZ = size of each covariance matrix; LNCAT = number of clusters.
IDAT1	1	In	Storage location in ARRAY for data read.
IPLACE	NOPTS	In	Vector of cluster numbers to which corresponding data points (pixels) are assigned.
MEANS	NOFEAT,MAXCLS	In	Array containing means of each feature for each cluster.
N	MAXCLS	In	Array containing field and class information for the cluster being processed.

Parameter	Dimension	In/out	Definition
IBAD	1	Out	Indication that covariance matrix was singular.

23.2.4 OUTPUTS

This subprogram outputs the covariance matrix on the specified unit.

23.2.5 STORAGE REQUIREMENTS

This subprogram requires 4800 bytes of storage.

23.2.6 DESCRIPTION

The COVPAT subprogram calculates the lower triangular portion of the covariance matrix and stores each element in the COVAR array in consecutive locations. (See section 9.2.6 for a description of the equation and matrix.) It calls subprogram CHLDET to compute the determinant and ascertain if the covariance matrix for each cluster is singular. If so, the cluster is deleted.

23.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 23.6.

23.2.8 LISTING

The subprogram listing is provided in volume IV, section 23.

23.3 ISOPAT

The ISOPAT subprogram performs the clustering process for the TESTSP processor.

23.3.1 LINKAGES

This routine calls the CLDIST, CLOCK, DESCEN, PRINT, PSPPAT, and SUNFAC subprograms. It is called by the TESTSP driver routine.

23.3.2 INTERFACES

The ISOPAT subprogram interfaces with other routines through common blocks GLOBAL, ISOLNK, and PASS and through the calling arguments.

23.3.3 INPUTS

Calling sequence: CALL ISOPAT(IDAT1,IPLACE,MEANS,N,STDEV,CLD, FLDINF,AVP,AMN)

Parameter	Dimension	In/out	Definition
IDAT1	1	In/out	Storage location in ARRAY for data read.
IPLACE	NOPTS	Out	Vector of cluster numbers to which corresponding data points are assigned.
MEANS	NOFEAT,MAXCLS	Out	Array containing means of each feature for each cluster.
N	MAXCLS	In	Number of pixels per cluster.
STDEV	NOFEAT,MAXCLS	In	Array containing standard deviations for each feature and/or cluster.
CLD	MAXCLS,MAXCLS	In	Array containing distance between clusters.

Parameter	Dimension	In/out	Definition
FLDINF	1	In	Array containing field information.
AVP	NOFEAT,MAXCLS	Out	Intermediate storage for compared cluster standard deviations.
AMN	NOFEAT,MAXCLS	Out	Intermediate storage for computed cluster means.

23.3.4 OUTPUTS

Not applicable.

23.3.5 STORAGE REQUIREMENTS

This subprogram requires 16 304 bytes of storage.

23.3.6 DESCRIPTION

The ISOPAT subprogram accepts field and statistical information as input, removes DO/DU pixels from the pixels to be clustered, applies a Sun-angle correction by channel to each pixel, and assigns data to clusters. Subprogram CLDIST is called to calculate distances between cluster centers. If the maximum number of iterations (ISTOP) equals zero, small clusters are deleted. If ISTOP does not equal zero, subprogram PRINT is called to output results.

ISOPAT continues by deleting clusters with fewer than the minimum allowable number (NMIN) of pixels. The current number of clusters (LNCAT) is reduced accordingly. The maximum standard deviation for each cluster is found, and a beginning sequence of SPLIT iterations is performed until at least 80 percent of the clusters processed have standard deviations less than the threshold parameter STDMAX. Then, iterations alternate between COMBINE

and SPLIT until the last iteration (ISTOP), which is always a SPLIT iteration. Parameters are reinitialized and the iteration procedure begins again. If ISTOP is not reached, the user must initialize the next iteration by entering "CLASSIFY AND CALCULATE NEW STATISTICS." (The iteration procedure is described in detail in volume II, section 9.1.3.)

23.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 23.6.

23.3.8 LISTING

The subprogram listing is provided in volume IV, section 23.

23.4 PSPPAT

Subprogram PSPPAT performs much of the computational work associated with the TESTSP processor. For each iteration of the clustering algorithm, it assigns pixels to clusters and computes the new mean and standard deviation vectors.

23.4.1 LINKAGES

The PSPPAT subprogram calls the RREAD and RWRITE subprograms. It is called by the ISOPAT subprogram.

23.4.2 INTERFACES

The PSPPAT subprogram interfaces with other routines through common blocks ARRAY, ISOLNK, and PASS and through the calling arguments.

23.4.3 INPUTS

Calling sequence: CALL PSPPAT(MEANS,STDEV,N,CLD,IDAT1,IPLACE, AVP,AMN,MEN)

Parameter	Dimension	In/out	Definition
MEANS	NOFEAT,MAXCLS	In/out	Array containing cluster means.
STDEV	NOFEAT,MAXCLS	Out	Array containing cluster standard deviations.
N	MAXCLS	Out	Number of pixels in each cluster.
CLD	MAXCLS,MAXCLS	In	Array containing intercluster distances; not used.
IDAT1	1	In/out	Storage location in ARRAY for data read.
IPLACE	NOPTS	In/out	Vector of cluster numbers to which pixels are assigned.

Parameter	Dimension	In/out	Definition
AVP	NOFEAT,MAXCLS	Out	Intermediate storage for computed cluster standard deviations.
AMN	NOFEAT,MAXCLS	Out	Intermediate storage for computed cluster means.
MEN	NOFEAT,MAXCLS	Out	Not used; in ISOPAT call, the same storage as MEANS.

23.4.4 OUTPUTS

The results are returned for use by the calling routine.

23.4.5 STORAGE REQUIREMENTS

This subprogram requires 6400 bytes of storage.

23.4.6 DESCRIPTION

As the pixel radiance values are read from disk, they are first tested for 0 or 255 values over all channels. DO and DU pixels are assigned cluster numbers greater than the cluster numbers assigned to classes. Any pixel not DO or DU is assigned to a cluster based on the minimum distance L_1 from the means. At the user's option, Sun-angle correction factors can be applied to this distance computation. After all pixels are assigned to clusters, PSPPAT recomputes the cluster means and standard deviations.

In order to save calculation time, PSPPAT uses two basically identical cluster assignment loops, one which does and one which does not incorporate the Sun-angle corrections. Thus, this multiplication (with a default value of 1) is not performed for runs without Sun-angle corrections.

23.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 23.6.

23.4.8 LISTING

The subprogram listing is provided in volume IV, section 23.

23.5 RDDPAT

The RDDPAT subprogram coordinates the routines which read fields of data from the MSS DATAPE and stores the data on disk for processing by TESTSP.

23.5.1 LINKAGES

The RDDPAT subprogram calls the CMERR, FDLINT, FLDINT, LAREAD, LINERD, RWRITE, and TAPHDR subprograms. It is called by the TESTSP driver routine.

23.5.2 INTERFACES

The RDDPAT subprogram interfaces with other routines through common blocks ARRAY, GLOBAL, and PASS and through the calling arguments.

23.5.3 INPUTS

Input to the RDDPAT subprogram consists of the MSS DATAPE and field definition card images (volume II, section 3.2.3).

Calling sequence: CALL RDDPAT(FD1,TOP,IDATA,IDIM,LAST)

Parameter	Dimension	In/out	Definition
FD1	1	In	Location in ARRAY for storing class names.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
IDATA	IDIM	In	Array for storing pixel values for each scan line.
IDIM	1	In	Maximum usable storage for IDATA.
LAST	1	Out	Flag indicating (when set to 1) that all classes for one set of control cards have been processed.

23.5.4 OUTPUTS

The RDDPAT subprogram outputs listings of DO/DU fields and fields to be clustered for each given class.

23.5.5 STORAGE REQUIREMENTS

This subprogram requires 10 670 bytes of storage.

23.5.6 DESCRIPTION

The RDDPAT subprogram reserves 2000 locations in ARRAY for storing field definition information. Field definition cards are read for each class and/or field and are stored as follows:

ARRAY(1) = class name

ARRAY(2) = index pointer to next class name

ARRAY(3) = number of clusters in this class

ARRAY(4) = number of fields for this class

ARRAY(5) = first field name

ARRAY(6) = number of vertices

ARRAY(7) = actual vertex numbers

ARRAY(8) = total pixels in the field

ARRAY(9) = field information block for this field

RDDPAT reads DO and DU CLASSNAME cards and the associated field definition cards, calculates field information, and computes information concerning DO/DU pixels for use by other subprograms; e.g., the number of DO/DU pixels in each field for the field being processed and the disk addresses for the DO/DU pixels. It also assigns printing symbols and special mean values (0 or 255) to DO and DU pixel radiance values. Special cluster numbers are assigned; i.e., numbers having values 1 and 2 greater than the numbers of other clusters are given to DO/DU pixels.

23.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 23.6.

23.5.8 LISTING

The subprogram listing is provided in volume IV, section 23.

23.6 SUBPROGRAM FLOW CHARTS

No flow charts are provided for the TESTSP processor.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1980—671-099/936

~~23-17~~
K33