(E80-10207) AS-BUILT DESIGN SPECIFICATION
FOR THE CAMS IMAGE-100 HYBRID SYSTEM.
VOLUME 1: SYSTEM DESIGN (Lockheed
Electronics Co.) 322 p HC A14/MF A01
CSCL 05B G3/43

N80-29786

Unclas
00207

T72-1440207.

NASA CR-
160644

"AS-BUILT" DESIGN SPECIFICATION
FOR THE
CAMS IMAGE-100 HYBRID SYSTEM

Job Order 71-195

VOLUME 1
SYSTEM DESIGN

Prepared By

Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15200
For

EARTH OBSERVATIONS DIVISION

SCIENCE AND APPLICATIONS DIRECTORATE

National Aeronautics and Space Administration
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas
August 1977

LEC-10822
Volume 1

JSC-13030
Volume 1

"AS-BUILT" DESIGN SPECIFICATION FOR THE CAMS IMAGE-100 HYBRID SYSTEM

Job Order 71-195

VOLUME 1

SYSTEM DESIGN

Assembled By

L. E. Giddings

From contributions of the following persons:

R. T. Minter
K. L. Pattison
P. S. Lin
G. J. Champagne
E. J. Hightower
W. A. Holley
J. S. Huang
T. R. Kell
D. L. Loe
L. F. Robinson
R. M. Rodriguez
J. K. Rowland
C. D. Shih
H. G. Thadani
S. G. Thadani
B. R. Thompson
E. L. Wilson

APPROVED BY

Philip L. Krumm

P. L. Krumm, Supervisor
Applications Software Section

Prepared By

Lockheed Electronics Company, Inc.

For

Earth Observations Division

Science and Applications Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

August 1977

LEC-10822
Volume 1

SPECIAL NOTE

The information in all three volumes of this document has been carefully checked. It is current at the time of publication, the end of August, 1977. This document will not be revised to show corrections and further changes. Rather, a new document will be issued toward the end of 1977 incorporating all changes, and making necessary corrections. The new volumes will be issued under the title: "As-Built Design Specifications for the CAMS Image-100 Hybrid System, as modified. The new document will be issued as LEC-11216 and JSC-13118.

Please bring errors and corrections to the attention of L. Giddings, 333,6311, mail code C42.

ABSTRACT

This document shows the CAMS Image-100 Hybrid System as it was actually built. Volume 1 lists the computer programs for each portion of the system, together with functional flow charts. Subroutines and function for each program are described in the summary. Volume 2 presents detailed flow charts and listings of all items listed in the first volume. The third volume presents brief descriptions and listings of subroutines shared by several programs. All three volumes close with an index of computer elements of the entire document.

CONTENTS

FIGURES

Unless specifically identified as reports or illustrations, all figures contain flowcharts.

Figure		Page
3-14	Field Definition.	.3-165
3-15	Dot Grid Crosshair Overlay.	.3-176
3-16	Dot Group Scatter Plots	.3-186
3-16a	Erase Window.	.3-194
3-16b	Theme Logical Operation	.3-197
3-17	Single Dot Labeling	.3-202
3-18	Automatic Brief Cluster Labeling (report)	.3-219
3-19	Automatic Cluster Labeling	.3-220
3-20	Cluster Map Display	.3-234
3-21	Recomputed Classification Summary (report)	.3-243
3-22	Classification Summary (report)	.3-244
3-23	Classification Map Display	.3-245
3-24	Dot Data Report (flow chart)	.3-256
3-25	Classified Category Percentages (report) (uncorrected)	3-260
3-26	Bias Correction Alpha Table (report)	.3-261
3-27	Corrected Proportions/Variances (report).	.3-262
3-28	Bias Correction/Classification Summary Flowchart.	.3-263
3-29	Brief Cluster Report (report)	.3-271
3-30	Cluster Mean/Standard Dev. Report (report).	.3-272
3-31	Intercluster Distance Report (report)	.3-273
3-32	Dots/Cluster Report (report).	.3-274
3-33	Cluster Nearest Neighbor Report (report).	.3-275
3-34	Cluster Reports Flowchart	.3-276
3-34a	Field Reports Flowchart	.3-279
3-35	Permanent Data Base Update.	.3-293

1. SCOPE

This document contains the design specification for software for the CAMS Image-100 Hybrid System. For economy, portions of existing Image-100 programs and interface modules were used whenever possible.

The CAMS Image-100 Hybrid System is implemented on the Building 17 PDP 11/45 Image processor to support CAMS Procedure 1 evaluation.

This document does not address software implemented on the LACIE system, except in reference to interfaces between the two systems. Further, this document assumes that the reader is familiar with both "Procedure 1" and the Image-100 analysis system.

2. APPLICABLE DOCUMENTS

- Job Order 63-1347-1195
- Requirements Document RECP 7M0001, dated January 5, 1977
- TIRF 76-0106
- GSFC/JSC INterface Control Document for the Large Area Crop Inventory Experiment. LACIE Level III Baseline Document 00701, December 29, 1974.
- JSC IMAGE-100 USERS' MANUAL, JSC-12586, LEC-10263, June 1977
- TEST SPECIFICATIONS for the CAMS Image-100 Hybrid System, LEC-10631, June 1977.

3. SYSTEM DESCRIPTION

This document describes the CAMS Image-100 Hybrid System. It presents design specifications for the software and the data base implemented on the PDP 11/45 and associated hardware in the Data Techniques Laboratory, Bldg. 17/JSC.

The system is designed to take advantage of unique I-100 display capabilities for interactive processing and the high speed of clustering and classification processes on the batch LACIE 6 system. The following paragraphs and Figure 3.1 briefly describe the procedures an analyst might follow using this system and the flow of data between the Image-100 and LACIE 6 systems. Current detailed LACIE analyst procedures are provided in the document.

Landsat imagery comes to JSC after certain preprocessing functions have been performed by the Goddard Space Flight Center. The data on tape is used to prepare film products, as well as to update the LACIE data bases, as shown in Figure 3-1.

Acquisitions for certain segments are transmitted by tape to the Image-100 data base. Once a decision has been made to process the accumulated data for a segment with the new acquisition, an analyst will normally first delineate DO/DU fields either on the Del Foster devices, using a photographic image; or at the Image-100 console, using the cursor and image on the television screen.

The analyst must label dots before performing clustering and classification on the LACIE 6 system. He may label dots either interactively or offline, but probably the most convenient will involve the Image-100 console. He can display the dots on the screen, superimposed on the image. He may then type is labels for each one that he can identify, using the Tektronix console. When completed, the Image-100 system will (indirectly) generate

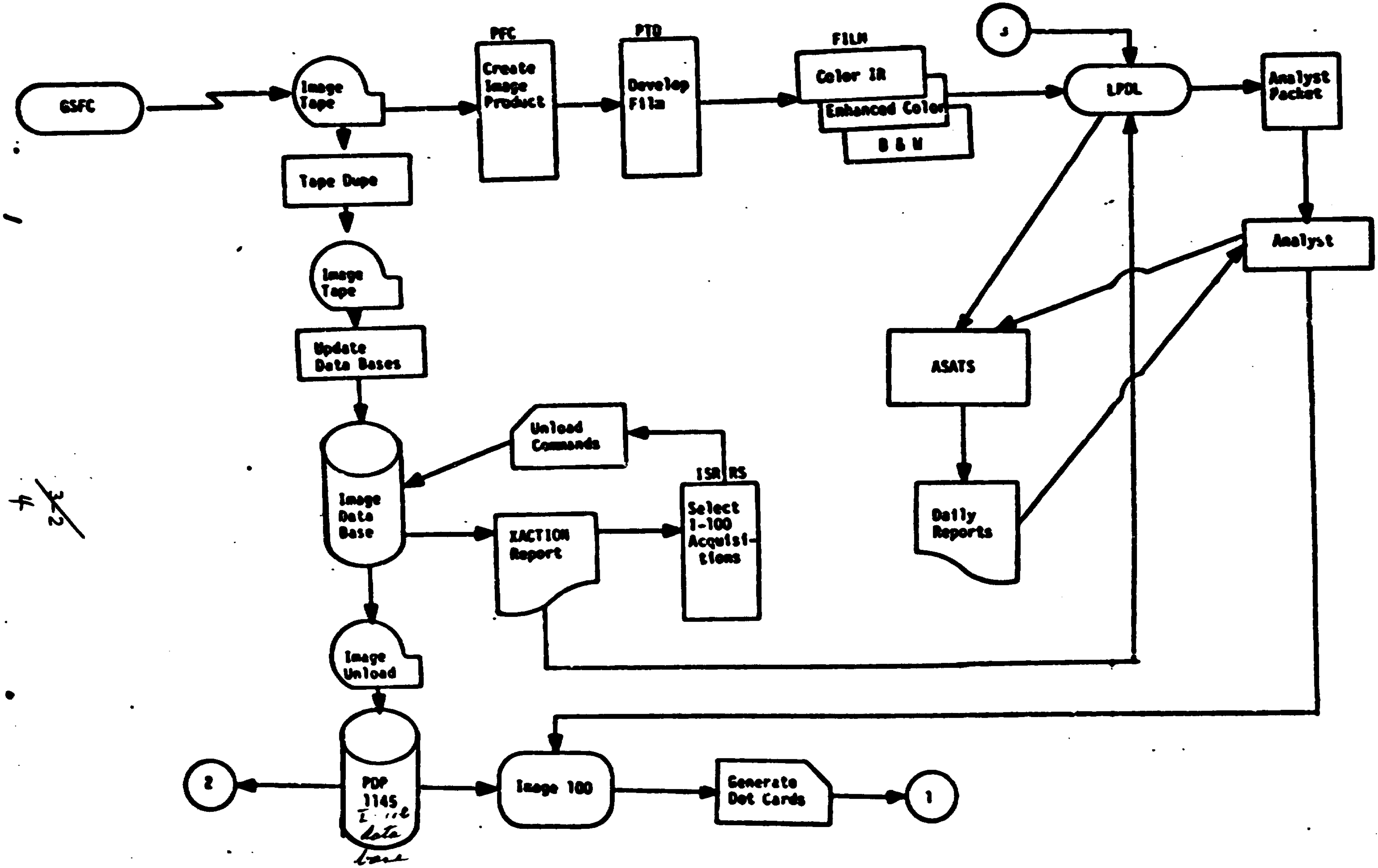


Figure 3-1.– CAMS Hybrid Functional Data Flow.

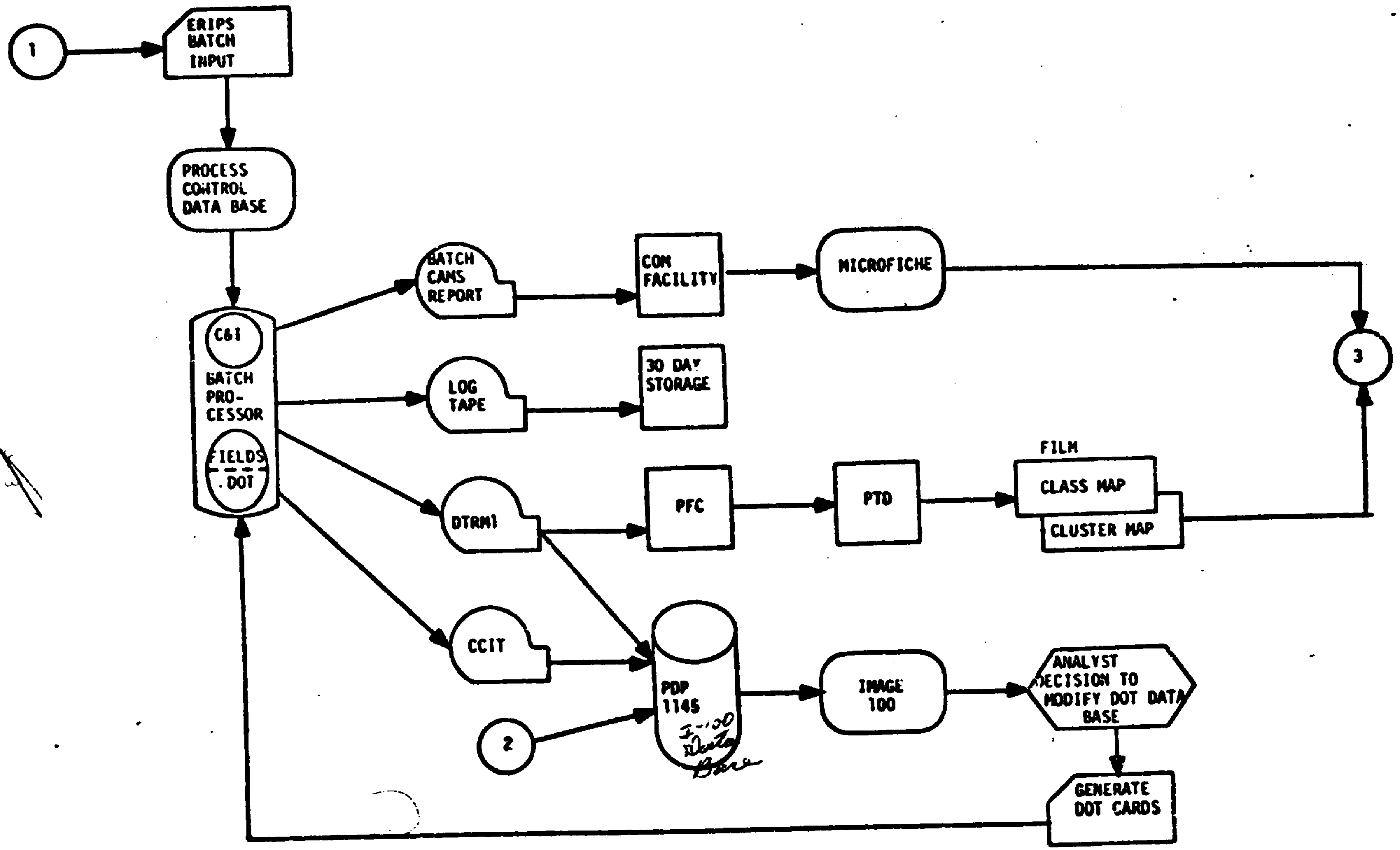


Figure 3-1.— Continued.

a card deck containing all labels to be used on the LACIE 6 system. The analyst might prepare the same cards manually if he wished, using photographic images and a transparent dot overlay.

This deck of cards, with a few others added, is submitted as a batch job to the LACIE 6 system. It contains all needed instructions for classifying the data for a give segment. The output products from the LACIE 6 system are the DTRM tape and the CAMS/CAS tape. The DTRM tape contains images with results of clustering (unconditional cluster map), results of an automatic cluster labelling procedure (conditional cluster map), and classification of the entire scene based on results of both of the above (classification map). (Figure 3-1 also shows the place of the DTRM tape in the flow of LACIE processing.) The CAMS/CAS tape will provide cluster statistics and classification summary information.

From this point, interactive operations allow the analyst to judge how well the entire procedure has worked. If he feels that the procedure has worked perfectly, he can accept results, have a report generated, update the data base, and exit the program.

If, on the other hand, the analyst notes deficiencies, he can analyze them at the console of the Image-100, interactively. Since defects will mostly relate to dot labels, he may display all dots, or any subset of them, in spectral coordinates as well as spatial coordinates. He may examine individual dots in several ways, to see if their current labels are correct. He may examine the "trajectory" of each dot, that is, its changes in spectral coordinates from acquisition to acquisition, to see if it has behaved as did other dots with the same label.

He may also change labels interactively. In some cases, if labelling is grossly defective, he will need to resubmit the segment for reclustering and classification. In most cases, it should only be necessary to use statistics already available for reassigning clusters to yield a new classification map without resubmission to the LACIE 6 system.

When the analyst is satisfied with results, he generates a report and updates the data base. His new results are then passed on to CAS for aggregation calculations.

3.1 HARDWARE DESCRIPTION

The hardware configuration for the CAMS Hybrid I-100 system is presented in Figure 3-2. Manuals for the PDP 11/45 computer and RSX 11D operating system are provided by the Digital Equipment Corporation (DEC). Manuals for the peripherals are provided by the vendors for the specific devices. It is beyond the scope of this document to define in detail the hardware components associated with the system.

3.2 DISK STRUCTURE

This section describes the structure of the two disks required for operational use of the system.

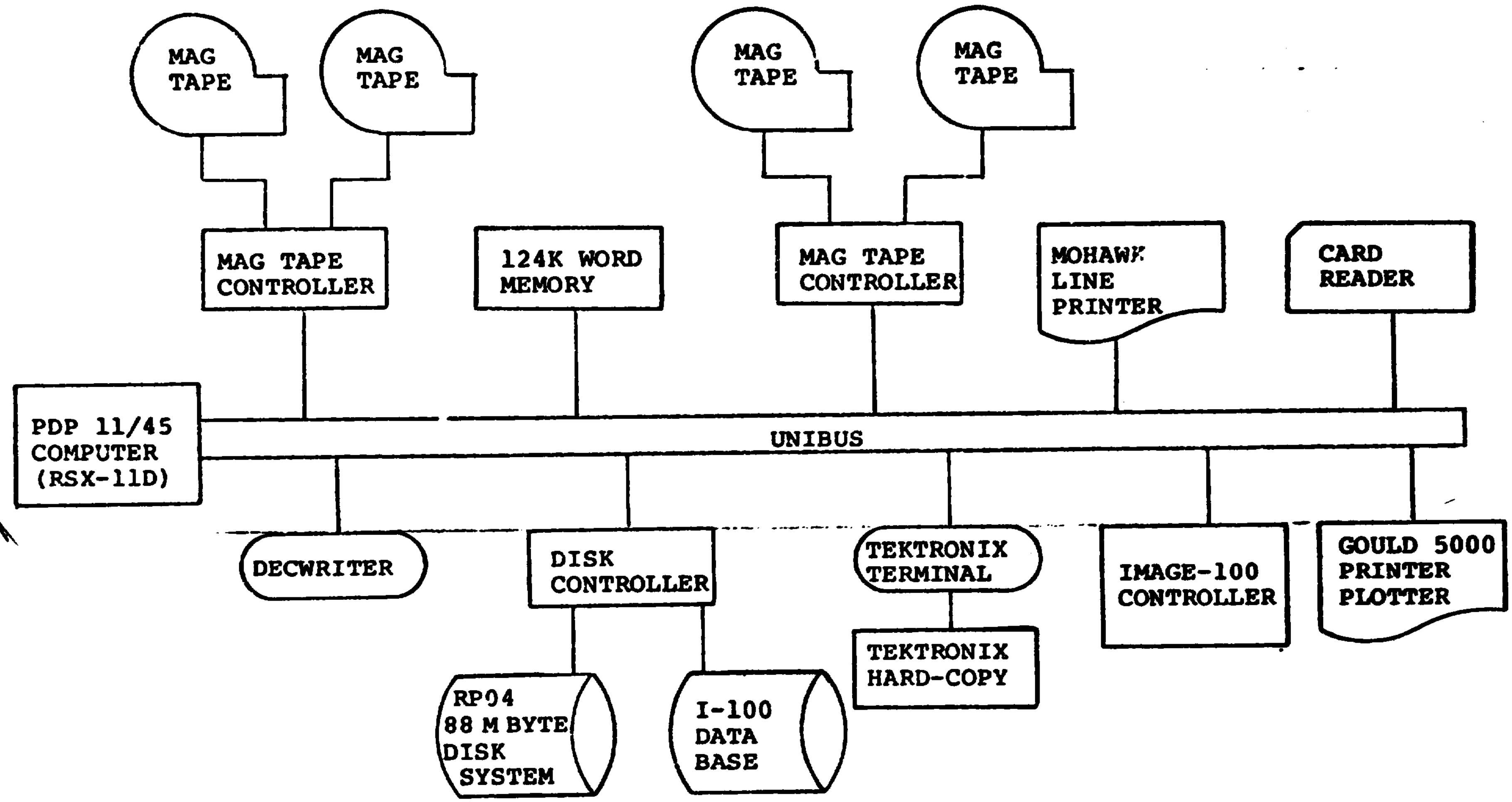


Figure 3-2.– CAMS/I-100 Hardware Configuration.

3.2.1 SYSTEM DISK

The system disk is an RP04 disk pack configured as shown in Figure 3-3A. The functions of the various segments residing on the system disk and critical to or supportive of the CAMS system are as follows:

- RSX-11D operating system - a resource-sharing executive designed to run on the PDP 11/45 digital computer in a multi-program environment.
- CAMS I-100/ERIPS Hybrid System - those computer programs required to accomplish the processing of the I-100/Procedure 1 Hybrid Interface and Interactive Displays.
- Image Library - those common functions originally developed by the General Electric Co. to support the Image-100 system.
- Terminal Control System (TCS) Library - those common functions developed by the General Electric Company to support the graphics display terminal.
- Image-100 System (IMA) - Those programs originally developed by the General Electric Company for the Image-100 and added to by Lockheed and Aeronutronic Ford Corporations.
- Data Base Directory and Disk Table File - The Data Base Directory contains segment level information pertaining to acquisition dates and classification results for all segments in the data base. Disk Table File contains the disk ID for each segment.
- Working Data Files for CAMS - a receptacle for those working results generated by interactive processing and for global common as written by each processing program just prior to its termination.

The following "User Identification Codes" (UIC's) are reserved for the CAMS I-100 Hybrid System on the system disk and contain the indicated data or software files.

[300,1] - All working files for the interactive programs.
(Defined in Section 3.4)

[300,2] - CAM run UIC

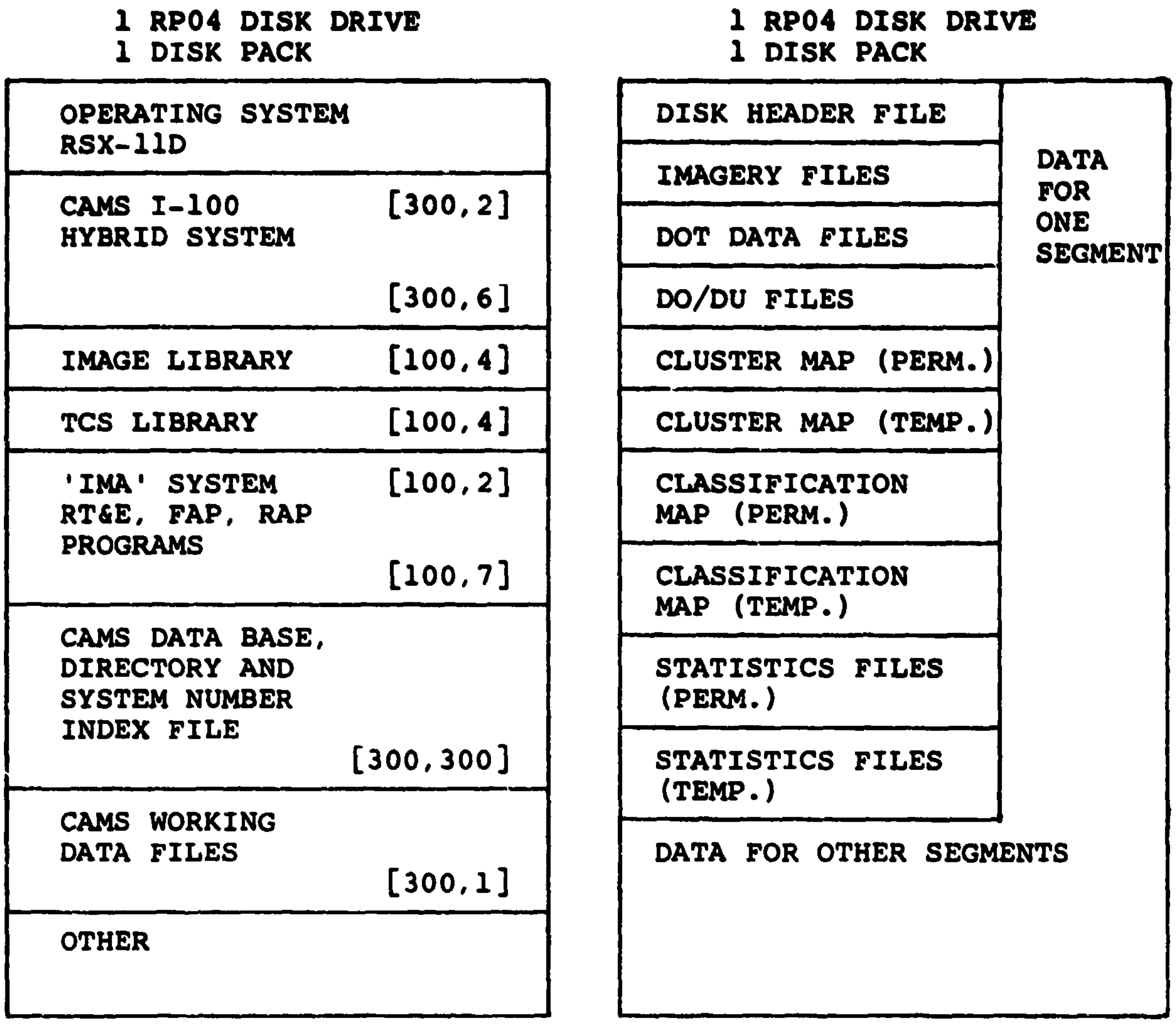
[300,3] - Source files for all programs and private subroutines

[300,4] - Object files for all programs and subroutines

[300,5] - Task and task build command files for all programs

[300,6] - Source files for all shareable and utility routines

[300,300] - Data base directory file and segment index file.
(Defined in Section 3.3.1)



SYSTEM DISK
3-3A

DATA DISK
3-3B

Figure 3-3: System Disk and Data Disk Configuration

3.2.2 DATA BASE DISK CONFIGURATION

The CAMS I-100 Hybrid System Data Base is maintained on multiple RP04 disk packs configured as shown in Figure 3-3. Each disk pack is initialized with a "disk identifier file" which contains a unique number identifying that disk pack. All files on all disk packs are maintained under the UIC [300,300]. Definitions of the various files are as follows. For more detail see Section 3.3.

- Disk Header File - File uniquely identifying the disk pack. File name is [300,300] HDRREC.DAT.

The following files may exist for each segment. Maximum number of segments per disk pack is 80.

- Imagery Files - No more than six acquisitions per segment of imagery data as read from the Universal formatted tape off-loaded from the LACIE data base.
- Dot Date Files - 209 dots per segment are described by spatial coordinates, category labels, type, and dot spectral information.
- DO/DU Files - a maximum of 50 DO/DU fields are described by field identification, field type and a maximum of ten vertices.
- Statistics Files - statistical information and supporting data as extracted from the CAMS/CAS Interface Tape.
- Cluster Map (Permanent or old) - the cluster map which the analyst has determined to be currently the most accurate.
- Cluster Map (Temporary or latest) - the cluster map which has been read from the DTRM tape and has not yet been examined by the analyst.
- Classification Map (Permanent) - the classification map which the analyst has determined to be currently the most accurate.
- Classification Map (Temporary) - the classification map which has been read from the DTRM tape and has not yet been examined by the analyst.

3.3 DATA BASE DESCRIPTION/FILE FORMAT

The CAMS Image-100 Hybrid System data base was designed to take advantage of the PDP 11/45 operating systems efficient filing system. All data is file oriented, a file naming convention was adopted for the data base so that the operating system could be used for quick retrieval of files by name.

The data is segment oriented and resides on multiple disk packs which must be mounted on device "DB2" one at a time as needed. The system disk has two controlling files which keep track of all segments and the disk pack that each segment is on.

A segment must be first introduced to the Image-100 data base by input of imagery data to the "Image Update Program" (IMAUPD). The system is programmed to handle a maximum of 200 segments on all disk packs. Approximately 80 segments can be loaded on to one disk pack.

The following Data Base File Naming Conventions were adopted.

Data Base Disk (DB2:) UIC = [300,300]

XXXX = Segment number
YYDDD = Julian Date

XXXXYYDDD.DAT - Imagery Files

XXXXPCLAS.MAP - Permanent classification map
XXXXTCLAS.MAP - Temporary classification map
XXXXPCLUS.MAP - Permanent cluster map
XXXXTCLUS.MAP - Temporary cluster map
XXXXPSTAT.DAT - Permanent stats and summary information
XXXXTSTAT.DAT - Temporary stats and summary information

XXXXDDOTS.DAT - Dot data
XXXXFIELD.DAT - Fields

System Disk UIC = [300,300]

DIRFILE.DAT - Directory file

DSKTBL.DAT - Segment disk table file

The remaining sections in 3.3 provide details on each of the above files.

3.3.1 DIRECTORY FILE (SY: [300,300]DIRFILE.DAT) AND DISK TABLE FILE (SY: [300,300]DSKTBL.DAT)

The Data Base Directory File is a multi-record direct access file. Each record is formatted as above and is segment oriented. The record for a segment is from this file at 'Initiate' time and moved, as is, to the common block COM2. The interactive programs update COM2 and the 'Data Base Update' program moves COM2 back to the appropriate record in the Directory File.

The Directory File interacts with the Disk Table File, to account for and locate any element (or elements) of data in the CAMS I-100 Hybrid System Data Base. Collectively they contain segment level information pertaining to acquisition dates, classification results and dates on which files were last updated. The directory is updated each time there is a change in the data base, whether the change is made in batch mode or on interactive command. The disk table file is updated by the Image Update and Image Delete programs.

Both files are resident on the system disk under the UIS [300,300]. The following programs access the files.

- Office Programs

  IMAUPD
  FLDUPD
  DOTUPD
  BSTAT
  DTERM
  SEGDEL

- Interactive Programs

  INIT
  PRMUPD

All of the above programs perform updates to the directory file except INIT. The directory file is unformatted and direct access. Records are segment oriented, so that once the record number for a given segment has been determined all information in the directory for that segment can be retrieved with a direct read of that record.

Most programs utilize the routine DSKCHK to determine the record number for the segment in the directory and whether or not the correct disk pack for a given segment is mounted. The routine DSKCHK determines this by reading the disk table file and searching for the indicated segment. The index of the successful search is the record number for the segment in the directory file.

The format and content of the individual records of the directory file is identical to common block COM2 which is documented in Section 3.4.

3.3.2 IMAGERY DATA FILES

Imagery data files are maintained for up to six acquisitions of each segment. The files are uniquely named by the four digit segment number followed by the five digit Julian date for the acquisition. The files are generated by the offline program "IMAUPD" from image unload tapes from the LACIE 6 system. The files are LACIE segment size images (i.e. 196 pixels by 117 lines and 4 channels).

3.3.3 DOT DATA FILES (DB2: [300,300]XXXXDDOTS.DAT)

Dot data files, consisting of 209 dots per segment, each of which is described by spatial coordinates, labels (analyst and classification), type and dot spectral information, are automatically created upon introduction of new segments into the system, and may be updated from that point on either batch mode (cards) or interactively.

The offline program IMAUPD originates and updates the dot data files by extracting the spectral values from each acquisition of the image data. The files may also be updated by the offline programs DOTUPD and FLDUPD. The dot update program "DOTUPD" sets the analyst label and type fields. The field update program "FLDUPD" flags dots lying in DO/DU fields.

The analyst label and type fields may be altered interactively. Interactive modifications to these fields are not reflected in the data base files unless the analyst executes the interactive data base update program "PRMUPD".

The following is a complete list of the programs which must access the data base dot data files.

- Offline Programs - IMAUPD, DOTUPD, FLDUPD, SEGDEL
- Interactive Programs - INIT, PRMUPD

The file content and format is identical to the working file DOTS.TMP documented in Section 3.4.4.

3.3.4 DO/DU FIELD DEFINITION FILES (DB2:[300,300]XXXXFIELD.DAT)

A DO/DU field definition file is maintained for each segment. For each segment there is a maximum of fifty fields allowed. These fields have the attributes of field identifier, type of field and a maximum of ten vertices. The files are created after the segment has been introduced into the system and upon receipt of a field definition card deck, or by interactive definition of the fields.

The following programs must access these data base files.

- Offline Programs - FLDUPD, SEGDEL
- Interactive Programs - INIT and PRMUPD.

The file format is identical to the working file FIELDS.TMP documented in Section 3.4.1.

3.3.5 CLASSIFICATION RESULTS FILES

Classification results consist of the following items: cluster map, classification map, cluster statistics and classification summary information. This information is retrieved from the CAMS/CAS Interface tape and the DTRM tape from the LACIE 6 system.

Two each of the following files are maintained in the data base for each segment.

a. Classification Map Files - contains one 196 pixel x 117 scan line image representing the results of classification to the category level. (DB2:[300,300]XXXXTCLAS.MAP and DB2[300,300] XXXXPCLAS.DAT

b. Cluster Map - contains one 196 pixel by 117 scan line image which is an image product of the LACIE clustering function. (DB2:[300,300]XXXXTCLUS.MAP and DB2:[300,300]XXXXPCLUS,MAP)

c. Statistics File - contains the following elements extracted from the CAMS/CAS Interface tape and DTRM tape. (DB2:[300,300]XXXXTSTAT.DAT and DB2:[300,300]XXXXPSTAT,DAT)

   - number of categories
   - number of subclasses (clusters)
   - number of channels
   - number of pixels in DO
   - number of pixels in DU
   - number of pixels thresholded out
   - number of pixels in the segment
   - number of pixels in each category
   - number of pixels in each subclass (cluster)
   - channels used in classification
   - acquisition dates used in classification
   - cluster/category cross index
   - cluster means and standard deviations
   - date of classification

- cluster number to which each dot was assigned
- category to which each dot was assigned during classification

The "temporary" files (i.e. those which have a T in the file name following the segment number) are updated <u>only</u> by the offline programs BSTAT and DTERM which read the latest results from LACIE. The only other program accessing the temporary files is the interactive program INIT, and the offline program SEGDEL.

The temporary files are made the "permanent" files (i.e. those which have a P in the file name) only by interactive decision by the analyst. This is done by the program PRMUPD. The programs INIT and SEGDEL also access these files.

The format and content of these files are identical to the working files presented in Section 3.4.

3.3.6 DISK PACK INITIALIZATION FILE - DB2:[300,300]HDRREC.DAT

Each of the data base disk packs must have this file on it before any segments can be added to the pack. The file is one record containing a unique identifier for that disk pack. The file may be generated by the PDP 11/45 system editor and PIP with the following steps.

1. MCR>HEL [300,2]
2. MCR><u>EDI</u>
3. EDI><u>HEAD.DAT</u>
   -CREATING NEW FILE-
   INPUT
4. 5
5. C/R
6. *EX
7. make sure that the disk pack labeled number 5 is mounted on DB2
8. MCR PIP DB2:[300,300]HDRREC.DAT = HEAD.DAT

Step 4 is the input of the disk pack number.

The following programs access this file to ascertain that the correct disk pack is mounted for a given segment. (The routine DSKCHK performs this function.)

- Offline programs

IMAUPD	BSTAT
DOTUPD	DTERM
FLDUPD	SEGDEL

- Interactive program - INIT

3.4 INTERACTIVE PROGRAMS WORKING FILES/GLOBAL COMMON DEFINITION

The files defined in this section are those working files which reside on the system disk during an interactive session. Some of the files are initialized by transferral from the data base in the Initiate Segment Analysis program, others may be generated during processing to pass information from one processing function to another. All working files reside under UIC [300,1] on the system disk.

The interactive Data Base Update program transfers those files which were updated back to the data base disk.

3.4.1 FIELD DEFINITION FILES

The 'Initiate Segment Analysis' program checks the data base existance flag for this file. If the file exist in the data base for the segment, that file (XXXXFIELD.DAT) is copied into the file [300,1] FIELDS.TMP. The existance flag, EFLAG3, in common COM3 is set accordingly.

The interactive program Field Definition reads this file and optionally displays, deletes and adds fields. The file [300,1] FIELDS.TMP is overwritten with the updated fields. If fields were deleted or added this program sets the update flag UFLAG2.

The Reports program reads this file and generates a report. The Recompute Proportion program must access this file.

The interactive Data Base Update program copies the file back to the data base file (XXXXFIELD.DAT) if UFLAG2 is set.

The file is written with unformatted FORTRAN write statements in the following format.

Record 1

Variable	Type	Definition
NOFLD	I	Number of fields in the file.

Records 2 through NOFLD + 1

FLDNAM	A	Name of the field. 6 bytes
LABEL	B	Number of vertices.
NV	B	-1 for DO fields, -2 for DU fields.
VERTEX	B	Coordinates of each vertex ordered as (x,y) or (pixel, line). These spatial coordinates reflect the vertex position in the data base image rather than the displayed image.

Example Usage:

```
   BYTE NV, VERTEX
   DIMENSION FLDNAM (3, MAXFLD), LABEL (MAXFLD), VERTEX (2, MAXV,
      MAXFLD), NV (MAXFLD)
   CALL ASSIGN (LUN, 'SY:[300,1]FIELDS.TMP')
   READ(LUN) NOFLD
   DO 10 I = 1, NOFLD
   READ(LUN) FLDNAM(I), LABEL (I), NV(I), VERTEX(J,K,I), J=1,2),
      K=1,MAXV)
10 CONTINUE
```

3.4.2 CLASSIFICATION MAP FILE

The Initiate Segment Analysis program asks the analyst whether he wants the permanent or temporary classification results from the data base. If he responds 'permanent', PFLAG is set to one, the existence date is checked and if it exists, the file XXXXPCLAS.MAP is copied from the data base into the working file SY:[300,1]CLASSMAP.TMP.

If he responds 'temporary', the data base existence date is checked for the file XXXXTCLAS.MAP. If it exists in the data base and the date matches the other classification results file dates, the file is copied into the working file SY:[300,1]CLASS MAP.TMP. The working file existence flag EFLAG2 is set accordingly.

The Classification Map Display program checks the flag EFLAG2 before attempting to assign the file. The Display portion of this program treats this file as a read only file. However, the Recompute Proportions module overwrites this file with a new classification map. When this happens, the program sets the update flag UFLAG1.

The interactive Data Base Update programs checks the flag, PFLAG; if equal 0, the file SY:[300,1]CLASSMAP.TMP is copied back to the data base file XXXXPCLAS.MAP. If PFLAG=1, (Common/COM3/), then the update flag UFLAG1 is checked, if the file has not been updated then the copy is not necessary.

The file is direct access by scan line number, with the following format.

Records 1 through NLIN

BUF(NPIX) - NPIX bytes in length
where
BUF(k) = category number to which pixel k was assigned in classification.

Example Usage:

```
   INCLUDE 'SY: [300,3]CAMSPARAM.INC'
   DIMENSION BUF (NPIX)
   LUN = 3
   CALL ASSIGN (LUN, 'SY:[300,1]CLASS MAP.TMP)
   DEFINE FILE LUN (NLIN, NPIX, U, NREC)
   DO 10 LINE = 1, NLIN
   READ (LUN 'LINE)(BUF(I), I = 1, NPIX)
   process this scan line
10 CONTINUE
```

3.4.3 CLUSTER MAP FILE

The Initiate Segment Analysis program asks the analyst whether he wants the permanent of temporary classification results from the data base. If he responds 'permanent', PFLAG is set to 1, the data base existence date is checked and if it exist, the file XXXXPCLUS.MAP is copied to the working file SY:[300,1] CLUSTERMP.TMP.

If he responds 'temporary', the data base existence date is checked against other classification results dates, and if it exist, the file XXXXTCLUS.MAP is copied to the working file SY:[300,1]CLUSTERMP.TMP.

The working file existence flag EFLAG1 is set accordingly.

The program Cluster Map Display checks the existence flag EFLAG1 before assigning this file. The file is a read-only file. It is never modified by the interactive programs. The Recompute Proportions module in Classification Map Display also checks EFLAG1 before attempting to use the file.

The Data Base Update program checks PFLAG. If equal 0, and EFLAG1=1, then this file will be copied back to the data base file XXXXPCLUS.MAP. Copy is not necessary if PFLAG=1.

The format of this file is identical to the CLASSMAP.TMP file, except the map will reflect cluster number rather than category number.

3.4.4 DOT DATA FILE

The Initiate Segment Analysis program copies the data base XXXXDOTS.DAT into the working file SY:[300,1]DOTS.TMP. Some information (dot types and analyst labels) in the data base file is placed in common/COM2/ for quick access. This file is always existent in the data base if the segment has been introduced to the directory. The data base file and the working file formats are identical. Records may be retrieved by dot number. The number of records is NDOTS (209).

Format of individual records in the file is as follows:

Record k (or dot k)

Byte 1 Spatial coordinate pixel number
2 Spatial coordinate line number
3 Type
4 Category index
5 Reflectance (Spectral) value - channel 1 acquisition 1
6 Reflectance (Spectral) value - channel 2 acquisition 1
7 Reflectance (Spectral) value - channel 3 acquisition 1
8 Reflectance (Spectral) value - channel 4 acquisition 1
9-10 Greenness - acquisition 1
11-12 Brightness - acquisition 1

(Repeated for each acquisition)

13-20 same as above for channels 1-4 acquisition 2
21-28 acquisition 3
29-36 acquisition 4
37-44 acquisition 5
45-52 acquisition 6

The Initiate Segment Analysis program stores the analyst labels from the data base file into 'DLABEL' and the type field into 'TYPE' of common/COM2/.

The <u>Data Base Update Programs</u> checks UFLAG3, if labels have been updated then the file XXXXDOTS in the data base is updated from common parameters 'DLABEL' and 'TYPE'. and an offload of the Dot Data file in card image format is effected automatically. An offload of the file may be found at analyst option if no updates to the file have been made. This is also true of DO/DU data.

3.4.5 CLUSTER STATISTICS FILE

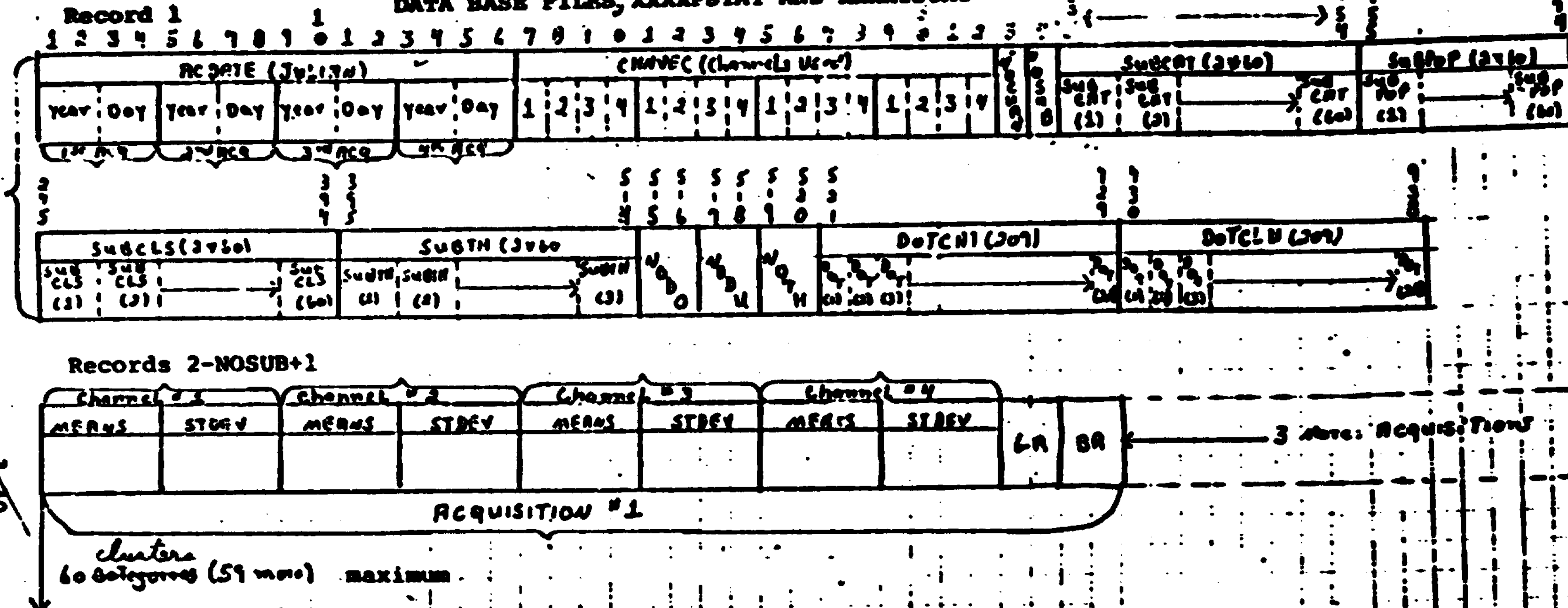
Global common/COM1/ will reflect record one from either the XXXPSTAT or XXXXTSTAT file, as retrieved from the data base.

The remaining records from that data base file are copied to the working file SY:[300,1]CLUSTATS.TMP and existence flag EFLAG5 is set by the 'Initiate Segment Analysis' program. The format of the data base file is presented on the next page.

The working file is a direct access file with records retrieved by cluster number.

The file SY:[300,1]CLUSTATS.TMP is never modified interactively. However, some of the parameters in /COM1/ may be updated. The Data Base Update program checks UFLAG4, if rework has occurred or if PFLAG=0, all records of the file XXXXPSTAT must be updated.

DATA BASE FILES, XXXXPSTAT AND XXXXTSTAT



The 'Initiate Segment Analysis' program moves record 1 into common block COM1 and records 2-NOSUB+1 are copies to the working file SY:[300,1]CLUSTATS.TMP. If less than MAXACC(4) acquisitions were used in classification, the dates, means, etc. are zero filled.

Figure 3-3a.- Data Base Files.

ORIGINAL PAGE IS
OF POOR QUALITY

3.4.6 NEAREST NEIGHBOR FILE - (NN.TMP)

This file is written by the 'Automatic Cluster Labelling' module and used in 'Cluster Map Display' and 'Reports'. Information in this file is not carried in the data base.

An existence flag (EFLAG4) is set by the creating program and checked by the using programs.

File Format

Record 1

Variable	Type	Definition
KNN	Integer	Number of nearest neighbors used.
ND	Integer	Number of labelling dots used.

Records 2-(NOSUB+1)

Variable	Type	Definition
DOTN	Integer	Dot number grid.
DLAB	Integer	Analyst label for this dot. Index to CATNAM.
DIST	Real	L-1 Distance from cluster mean to this dot.

Example Usage

```
   INCLUDE 'SY:[300,3]CAMSPARAM.INC'
   INTEGER DOTN, DLAB
   DIMENSION DOTN(NDOT), DLAB(NDOT), DIST(NDOT)
   LUN = ?
   CALL ASSIGN (LUN, 'SY:[300,1]NN.TMP, IER)
   READ(LUN) KNN, ND
   DO 10 I = 1, NOSUB
   READ(LUN)(DOTN(K), DLAB(K), DIST(K), K=1, KNN)
   processing for this cluster
10 CONTINUE
```

3.4.7 DOT GRID SCREEN COORDINATE FILE

The Initiate Segment Analysis program creates this file as [300,1]DOTGXY.TMP and initializes it to all ZERØES.

The interactive Image Display program will, after displaying the image, write the screen coordinates of all 209 dots to this file. If a dot physically occupies several pixels on screen, the screen coordinate of the upper left corner of the pixels will be stored in this file.

The interactive Dot Labelling program will search the stored coordinate pairs of this file in order to find the dot identification (dot grid number).

This file has one record which consists of 209 2-word entries. First word of each entry is the dot grid screen coordinate on horizontal axis, second word is the dot grid screen coordinate on vertical axis. Entries are ordered in dot grid screen. The value of each entry ranges between 0 and 511.

3.4.8 SPECTRAL PLOT SCREEN COORDINATE FILE

The Initiate Segment Analysis program will create this file as [300,1]SCATXY.TMP and initialize it to all

The interactive Dot Group Scatter Plot program will write the screen coordinates of all dots used to compose the scatter plot on a given window to a record of this file whose record number is corresponding to the window number, and this program will also set the same record for this given window to zero if this window is erased. If a dot physically occupies several pixels on screen, the screen coordinate of the upper left corner of the pixel will be stored in this file.

The interactive Dot Labelling program will search the stored coordinate pairs of this file in order to find the dot identification (dot grid number).

This file is a multiple record file. The number of records is equal to the number of spectral windows defined in the system (see NOSPWD in the system parameter list for this number) and the record number affects the window number. Each record of this fiel has the same forms as specified for the Dot Grid Screen Coordinate file. This file can be direct access by record number.

3.4.9 GLOBAL COMMON FILE - [300,1]GLOBAL.TMP

The purpose of this file is to provide a restart capability from a system crash. Before exiting every interactive program updates this file. The source code for doing this update is provided in source code files for inclusion in each interactive program for compilation.

An example of the source code provided in the file, SY:[300,3] CAMSCOMON.INC, follows:

```
INCLUDE 'SY:[300,3]CAMSPARAM.INC'
EQUIVALENCE (C1,ACDAT, (C2, ANALYSIS)
EQUIVALENCE (C3, ISEG), (C4, TX1)

DIMENSION C1 (    ), C2 (    ), C3 (    ), C4 (    )
COMMON/COM1/ACDATE, - - - - -
COMMON/COM2/ISEG, - - - - -
COMMON/COM3/PFLAG, - - - - -
COMMON/COM4/TX1, - - - - -
COMMON/COM5/DISKID
```

The above file is included in the FORTRAN source for each program with the FORTRAN statement:

```
INCLUDE 'SY:[300,3]CAMSCOMON.INC'
```

The following source code is provided in the file SY:[300,3]CAMSAVE.INC.

```
CALL ASSIGN (1, 'SY:[300,1]GLOBAL.TMP')
WRITE (1) C1
WRITE (1) C2
    .
    .
    .
```

This code is included before exiting program modules by the FORTRAN statement INCLUDE 'SY:[300,3]CAMSAVE.INC.

The 'Initiate Segment Analysis' program performs a similar read to restore the common area if the analyst has indicated a 'RESTART' mode of operation.

The subsections of 3.4.9 define each parameter in the five common blocks.

3.4.9.1 COM1

This common block will initially reflect Record 1 of either the XXXXPSTAT or XXXXTSTAT file as retrieved from the data base. Some of the parameters will reflect I-100 rework of the segment.

COMMON BLOCK NAME /COM1/

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
ACDATE	2,MAXACC	I	Julian dates for the acquisitions used in generating the classifications results. These dates reflect the 'permanent' or 'temporary' files as brought over from the data base. ACDATE(1,I)=year (2,I)=day	Retrieved from CAMS/CAS tape by the offline data base programs. Never modified.
CHNVEC	MAXCHN, MAXACC	B	Channels used in the classification. (=0 not used, =1 used) i.e., CHNVEC(2,3)=1 means the second channel of the third acquisition was used in classification.	As above.
NOCHAN		B	Number of channels used in the classification.	As above.
NOSUB		B	Number of subclasses. (clusters)	As above.

Figure 3-3b.– COM1.

COMMON BLOCK NAME COM1 (continued)

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
SUBCAT	MAXSUB	I	Index to CATNAM array, indicating the category to which each cluster was assigned. SUBCAT(15)=3 means cluster 15 was assigned to category 3.	Retrieved from CAMS/CAS tape. Permanent files will reflect I-100 rework, if any.
SUBPOP	MAXSUB	I	Population of each training subclass (cluster). (Number of pixels used in computing the statistics for training the classifier.)	Retrieved from CAMS/CAS, never modified by us.
CATKNT	MAXCAT	I	Number of pixels classified into each category.	Retrieved from CAMS/CAS for the temporary files. Permanent files may reflect I-100 rework, if any. (Recompute Proportions)
CATTH	MAXCAT	I	Pixels thresholded from each category.	As above. (Possibly reflect I-100 rework in Recompute Proportions.)

COMMON BLOCK NAME COM1 (continued)

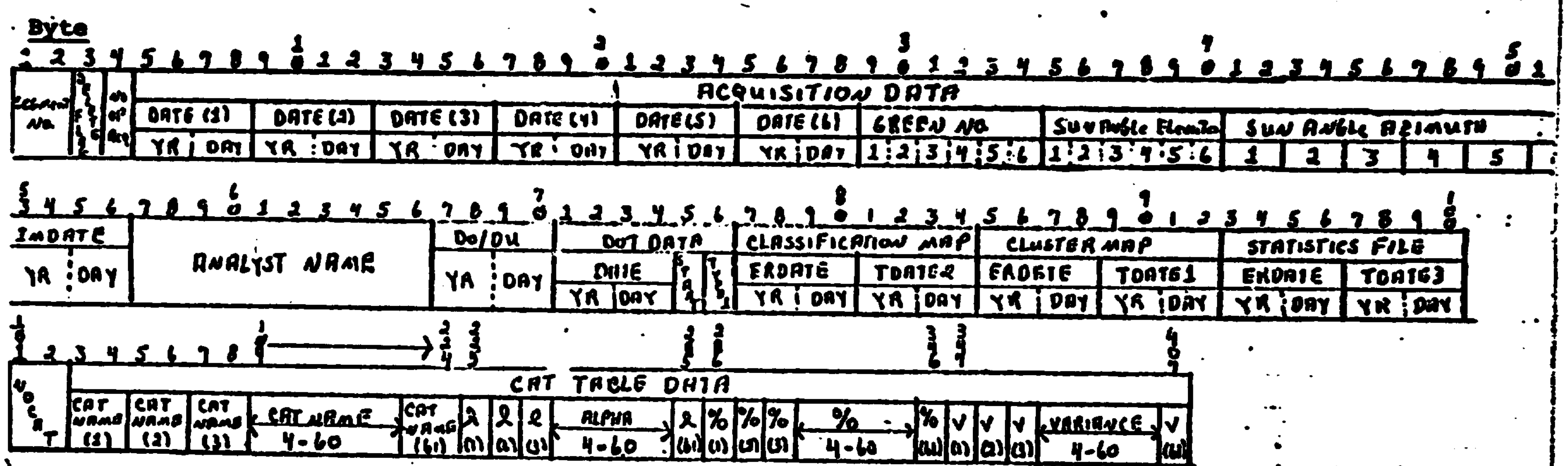
SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
NODO		I	Number of pixels in DO areas.	As above. May reflect I-100 rework.
NODU		I	Number of pixels in DU areas.	As above. May reflect I-100 rework.
NOTH		I	Total number of pixels threshold during classification.	Retrieved from CAMS/CAS. May reflect I-100 rework.
DOTCAT	NDOTS	B	Classification of each dot. Index to CATNAM table.	Retrieved from DTRM tape (Classification map) for temporary files. Permanent files may reflect I-100 rework. Recompute Proportions will update this array.
DOTCLU	NDOTS	B	Cluster to which each dot belongs.	Retrieved from DTRM tape (cluster map). Never modified by us.

3.4.9.2 COM2

The parameters in this common area will reflect initially information retrieved from the directory file. Many of the parameters in this common may be updated interactively and must be saved again in the appropriate records at data base update time.

The 'Initiate Segment Analysis' program allows the analyst to type in the segment number desired, then retrieves the appropriate record from the data base directory file and moves that record into the common block COM2. The format of directory file records is presented on the next page.



The Directory File is a multi-record direct access file. Each record is formatted as above and is segment oriented. The record for a segment is retrieved from this file at 'Initiate' time and moved, as is, to the common block COM2. The interactive programs update COM2 and the 'Data Base Update' program moves COM2 back to the appropriate record in the Directory File.

Figure 3-3c.– Data Base Directory File - SY:[300,300]DIRFILE.DAT.

COMMON BLOCK NAME COM2

SIZE

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
ISEG		I	Segment number being worked.	Copied from directory file at 'INIT' time, based on analyst key-in.
DELFLG		B	Delete flag for the next acquisition to be deleted from the data base. (Directory order.) DELFLG=2 means delete acquisition ADATES(2).	Reported to the analyst at 'INIT' time, updated by the analyst at 'Data Base Update' time.
NOACQ		B	Number of acquisitions currently in the data base for this segment. Maximum is MAXACD, now set at 6.	Updated by the offline imagery update program only.
ADATES	2,MAXACD	I	Julian dates for all acquisitions in the data base. ADATE(1,K)=year (2,K)=day	The offline imagery update program will retrieve this from the header of the image tape.
SOILGR	MAXACD	B	Soil green number for each acquisition in the data base.	As above.

Figure 3-3d.– COM2.

COMMON BLOCK NAME COM2

SIZE

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
SUNEL	MAXACD	B	Sun elevation.	As above.
SUNAZ	MAXACD	I	Sun azimuth.	As above.
IMDATE	2	I	Julian date for last I-100 work on this segment. IMDATE(1)=year (2)=day	Reported to the analyst at 'Initiate Segment Analysis' time, overwritten with current date at 'Data Base Update' time.
ANALXST	5	I	10 character name of the analyst who last worked the segment.	Reported to the analyst of 'INIT' time and overwritten by key-in from analyst.
FLDDAY	2	I	Date that the DO/DU field file was last updated in the data base.	Updated by the offline program 'Field Update' and the interactive Data Base Update program.
DOTDAY	2	I	Date that the dot data file was last updated in the data base.	Updated by the offline program 'Dot Data Update' and the interactive Data Base Update program.
NSTART		B	Number of starting vectors for clustering.	Updated by the offline program 'Dot Data Update' and the interactive Data Base Update.

COMMON BLOCK NAME COM2 (continued)

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
NTYPE1		B	Number of type 1 dots for dots in data base.	As above.
PDATE1	2	I	Date of the ERIPS classification for the 'permanent' classification map file.	Reported to the analyst at INIT time and, if PFLAG=0, overwritten with TDATE1 at Data Base Update time.
TDATE1	2	I	Date of the ERIPS classification for the temporary classification map file.	See above.
PDATE2	2	I	ERIPS date for permanent cluster map file.	
TDATE2	2	I	ERIPS date for temporary cluster map file.	Retrieved from DTRM tape, overrides PDATE2 if PFLAG=0, at Data Base Update time.
PDATE3	2	I	ERIPS date for permanent statistics file.	

COMMON BLOCK NAME COM2 (continued)

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
TDATE3	2	I	ERIPS date for temporary statistics file.	Retrieved from CAMS/CAS tape. Overrides PDATE3 if PFLAG=0 at Data Base Update time.
NOCAT		I	Number of categories in CATNAM array.	Updated offline by 'Dot Data Update' and possibly by CAMS/CAS program. Updated interactively by dot labelling. Used by almost every program.
CATNAM	(2,MAXCAT)	B	2 character names of all categories for this segment.	See above.
ALP	MAXCAT	B	Diagonal elements of the bias correction alpha matrix, (multiplied by 100 and rounded). One value for each category in order of CATNAM table.	The interactive 'Bias-Correction' Report program will update this matrix.
ALPO		B	Alpha for 'OTHER'.	Same as above.

COMMON BLOCK NAME COM2 (continued)

SIZE ____________

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
PCTCT	MAXCAT	B	Percent of the identifiable portion of the segment. 'Bias-Corrected' percentages.	Same as above.
PCTCTO		B	Percent for 'OTHER'.	
VAR	MAXCAT	B	Variance for each category, (multiplied by 100 and rounded).	Same as above.
VARO		B	Variance for 'OTHER'.	Same as above.

3.4.9.3 COM3

This common block reflects run-time parameters and flags which require no data base update themselves but flag those files which must be updated.

COMMON BLOCK NAME COM3

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
PFLAG		I	Permanent or temporary flag. = 1 if analyst has requested permanent files from the data base. = 0 if temporary	Must be set by the 'Initiate Segment Analysis'. The Data Base Update program must check this flag and take appropriate action.
DSKMNT		I	Identifier for the data disk pack currently mounted. (Originally set by the INIT program).	After COM2 is initiated for the segment. A check for 'sameness' of DSKMNT and DISKID should be made. If not the same, issue message to operator and 'PAUSE'. Reset the parameter 'DSKMNT' after operator key in and perform the check again. Check must be made again at data base update time.

Figure 3-3e.– COM3

COMMON BLOCK NAME COM3 (continued)

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
File existence flags 1 = existent 0 = non-existent				
EFLAG1			Cluster map file.	Set by 'INIT'.
EFLAG2			Classification map file.	Set by 'INIT'.
EFLAG3			Fields file.	Set by 'Initiate' or 'Field Definition'.
EFLAG4			Nearest Neighbor.	Set by 'Automatic Cluster Labelling'.
EFLAG5			Cluster Statistics File.	Set by 'INIT'.
File update flags = 1 means interactive update has occurred.				
UFLAG1			Classification map file.	Set by 'Classification Map Display/Recompute Prop.'
UFLAG2			Fields file.	Set by 'Field Definition'.
UFLAG3			Dot label and type update.	Set by Dot Labelling'.
UFLAG4			Bias-Correction Flag	Set by 'Bias-Correction'.

COMMON BLOCK NAME COM3 (continued)

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
NEWLAB	MAXSUB	I	Category index for each cluster as assigned either by our Automatic Cluster labelling or Manual overrride.	This table will be intialized by setting it equal to SUBCAT table at 'INIT' time. These labels will be transferred to 'SUBCAT' in COM1 in the Recompute Proportions program.

3.4.9.4 COM4

This common reflects current display information concerning refresh memory. Information in this common is carried in the data base.

COMMON BLOCK NAME COM4

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
TX1		I	Image display (refresh memory) and image (data base) source coordinate system.	These parameters are set by the 'Image Display' program and used by cluster and classification map display and grid overlay.
TY1		I		
TX2		I		
TY2		I		
IX1		I		
IY1		I		
IX2		I		
IY2		I		
ACDISP	2	I	Date of acquisition on display.	Same as above, also used by scatter plots.
II1	4	I	Image display and image source channel reference. II1(2)=3 means channel 3 from the source image is displayed on channel 2 of refresh memory.	Same as above.
G	4	I	Gains applied to image when displayed. Originally floating point numbers, multiplied by 100, truncated and used as integers.	Same as above.

Figure 3.3f.– COM4.

COMMON BLOCK NAME COM4 (continued)

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
B	4	I	Bias values applied to the image when displayed. $*Z_1 = G_1(X_1-b_1)$ $Z_2 = G_2(X_2-b_2)$ $Z_3 = G_3(X_3-b_3)$ $Z_4 = G_4(X_4-b_4)$	Same as above.

* Z_i is not necessarily on channel i in refresh memory.

COMMON BLOCK NAME COM4 (continued)

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
DTWIND (COM4)	5,NODTND	I	Dot blow-up window information for details ● Word 1=1 if this window is used	● In the Initiate Segment Analysis program, words 1-5 are set to 0. ● In the Dot Labelling program 1. Set IND to 1 for a new window. 2. Set XU,YU,XL and YL to the analyst's cursor input for a new window. 3. Set IND to 0 for an erased window.
(NODTWD=10, system parameter				
SPWIND	(5,NOSPWD)	I	Scatter plot window information.	

COMMON BLOCK NAME COM4 (continued)

SIZE ________

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
IMWIND (COM4)	4	I	Image Display Window • 4 words/window • Words 1,2 - XU,YU Coordinates of the upper left corner of the window • Words 3,4 - XL,YL Coordinates of the lower right corner of the window.	• In the Initiate Segment Analysis program 1. Set XU,YU,XL and YL to its default value as follows: XU = 0 YU = 20 XL = 392 YL = 254 • In the Image Display program 1. Set XU,YU,XL and YL to the analyst's keyboard input if the default value is overridden in selecting a new location. • Referenced by the cluster/class Map Display progams, the Dot Group Crosshair Overlay program, the Field Definition program and the Dot Labelling program.

COMMON BLOCK NAME COM4 (continued)

SIZE ____________

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
NUMDOT		I	• One byte • Number of dots selected • See the dot selection of the Dot Group Scatter Plot program for dot selecting methods	• Set to 0 by the Initiate Segment Analysis program. • Set to the number of dots selected in the Dot Group Cross-hair Overlay program and the Dot Group Scatter Plot program. • Referenced by the Dot Processing program.
DOTARY	NDOTS	I	• One byte each, range 1-209 • Selected dot grid number • See the dot selection of the Dot Group Scatter Plot program for dot selecting methods.	• Set to 0 by the Initiate Segment Analysis program. • Set to the dot grid numbers for all the selected dots in the Dot Group Crosshair Overlay program and the Dot Group Scatte Plot program. • Referenced by the Dot Processing Program.

COMMON BLOCK NAME COM4 (continued)

SIZE ______

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
GMIN		I	• The minimum data value for all dot data and cluster means of all channels and acquisition.	• Initialized by the Initiate Segment Analysis program. • Referenced by the Dot Group Scatter Plot program and the Dot Processing program.
GMAX		I	• The maximum data value for all dot data and cluster means of all channels and acquisition.	• See above.
FUL	(2,7)	I	• FUL(I,J) is the full scale for data values in channel J. • I=1 is the minimum data value I=2 is the maximum data value • J=1-4 are channels 1-4 J=5 is the greenness J=6 is the brightness J=7 is the green number	• See above.

COMMON BLOCK NAME COM4 (continued)

SIZE ____________

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
			• FUL(1,1) = 0	
			FUL(2,1) = 127	
			FUL(1,2) = 0	
			FUL(2,2) = 127	
			FUL(1,3) = 0	
			FUL(2,3) = 127	
			FUL(1,4) = 0	
			FUL(2,4) = 63	
			FUL(1,5) = 0	
			FUL(2,5) = 100	
			FUL(1,6) = 0	
			FUL(2,6) = 120	
			FUL(1,7) = -5	
			FUL(2,7) = 50	
CLAWND	8	I	Classification map window display	
CLUWND	8	I	Cluster map window display	

3.4.9.5 COM5

This common block contains dot information retrieved from the dot data file.

COMMON BLOCK NAME COM5

SIZE

Variable	Dimension	Type	Definition	Run Time and Data Base Considerations
DISKID		I	Disk pack identifier for this segment.	Set by the offline Image Update program in the file DSKTBL.DAT.
DLABEL	NDOTS	B	Analyst label for each dot. Index pointing to CATNAM array. Special indices: 0 - unlabelled dot -1 - the dot is in a DO area -2 - the dot is in a DU area Grid order	Updated offline by Dot Data and Field Update programs. Updated interactively by Field Definition and Dot Labelling.
TYPE	NDOTS	B	Dot type (grid order) 1=labelling dot 2=bias correction 0=unlabelled dot	Updated offline by Dot Data Update program and interactively by Dot Labelling.
RANDOM	NDOTS	I	Random/Grid index for dots. RANDOM(1)=15 means the first dot in the random sequence is dot number 15 from the grid.	This will be carried in a data statement in the 'Initiate' program and the offline program 'Dot Data Update'.
GRID	NDOTS	I	Grid/Random index for dots. The reverse of 'RANDOM' array. GRID(15)=1 means dot number 15 from the grid is the first dot in the random sequence.	See above.

Figure 3-3g.– COM5.

3.5 SOFTWARE DESCRIPTION

The software for the CAMS/I-100 system fall into two major groupings. The first contains those batch, or operator controlled, programs required to update the data base with new acquisitions, card dot data and DO/DU fields and classification results from ERIPS. The second contains those interactive programs with which the analyst must perform his analyses.

3.5.1 OFFLINE DATA BASE UPDATES

All programs in this section will be executed either batch or by an operator responding to queries at a terminal. In the latter case, he will have convenient access to the tape drives and card reader.

Since the offline data base update programs must open and use the same files that the interactive programs use, data base updates must be scheduled around interactive sessions at the I-100 terminal.

3.5.1.1 Imagery Update (IMAUPD)

IMAUPD maintains the data base file for the Image-100 Hybrid System. The program adds new segments and adds and deletes acquisitions.

3.5.1.1.1 Linkages

IMAUPD is the driver program for the Image-100 Hybrid Directory Update system. It calls the following programs or subroutines to perform its task: DELEAT, DIRCRE, FILEST, RDDISK, SRDISK, REPORT, and STRAYS. Other programs which are used at various times during program execution are RDHEAD, TAPSCN, and JULIAN. To read and write data, both from tape and the disk, the Imagery Read Utility routines are used.

3.5.1.1.2 Interfaces

DIRUPD does not interface with any other system. It is a stand-alone program.

3.5.1.1.3 Inputs

Input to DIRUPD is an Image Data tape in Universal format.

3.5.1.1.4 Outputs

Each time an acquisition is added to the data base, the Imagery Data file from the input tape is stored on the disk. The file name contains the segment number(s), the year (Y), and the Julian day (D). – SSSSYYDDD.DAT-.

3.5.1.1.5 Storage Requirements

IMAUPD requires 111 blocks of computer space.

3.5.1.1.6 Description

DIRUPD is primarily a driver program. It calls a group of subroutines which perform the required functions. These subroutines will check the header record of each Image Data file to obtain the segment number of that file. Having the segment number, the program can check its disk table file (DSKTBL) to determine if this segment has already been entered on this disk. If so, the program will update the existing file with the new acquisition. If it is a new segment, the program must add it to the data base and made a new entry in the DSKTBL file. The acquisition dates and other information concerning each acquisition are maintained on the directory file (DIRFILE). The system allows for six acquisitions per segment number. In order to keep the number of acquisitions within limits, a delete flag is provided in the DIRFILE. It can be set to 1-6, and that acquisition will be deleted. Acquisition information in the DIRFILE are kept left justified in their respective fields.

Upon completion of a run, the program prints a report of action it has taken.

3.5.1.1.7 Flowchart (See figure 3-4)

See figure 3-4.

3.5.1.1.8 Subroutines

SRDISK

SRDISK is the first subroutine to be called by DIRUPD. It in turn calls RDHEAD, which reads the header record from the Image Data tape. Imagery Read Utility routines are used to read the Image Data tape. At this point, if the data file does not contain the proper numbers of channels, a diagnostic is printed on the line printer, and a flag is set to stop processing of this segment. Otherwise processing continues by interviewing

IMAGE DIRECTORY UPDATE

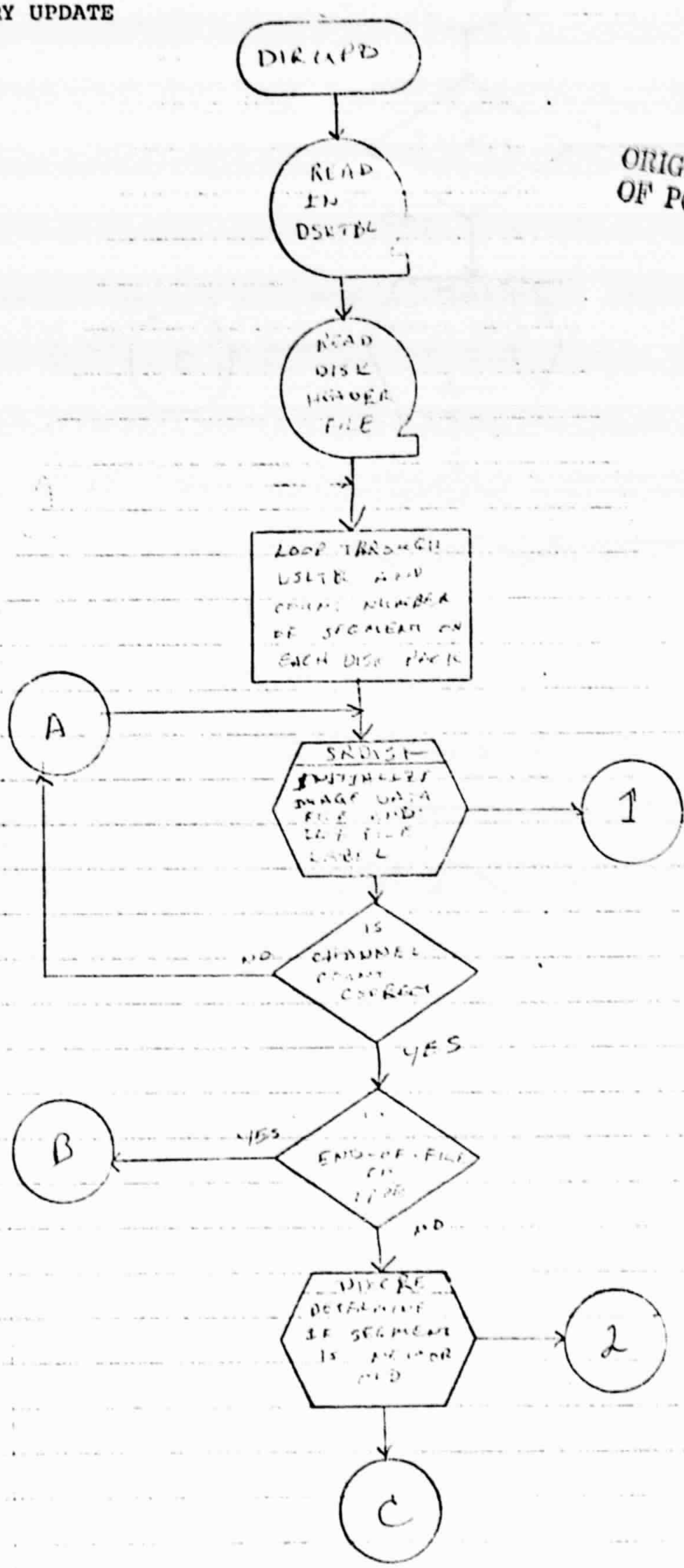
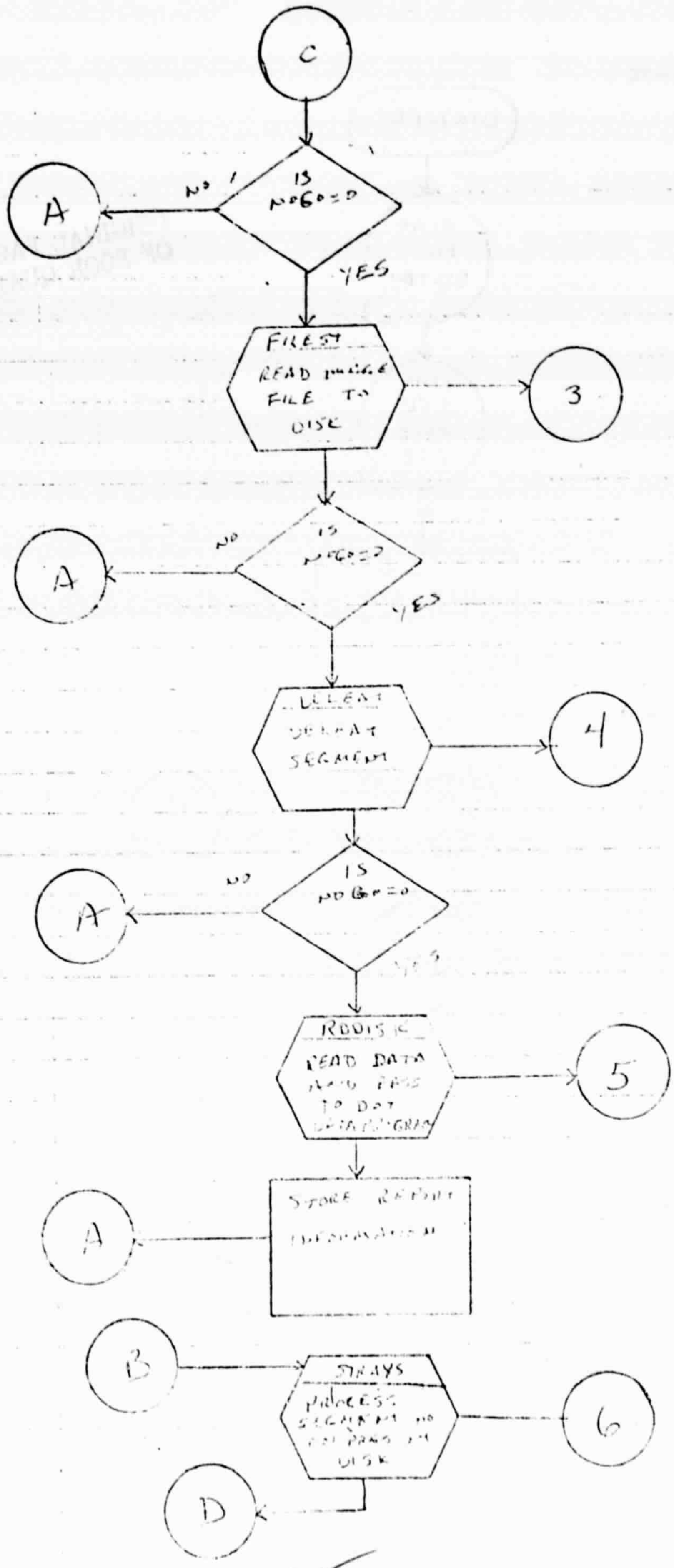
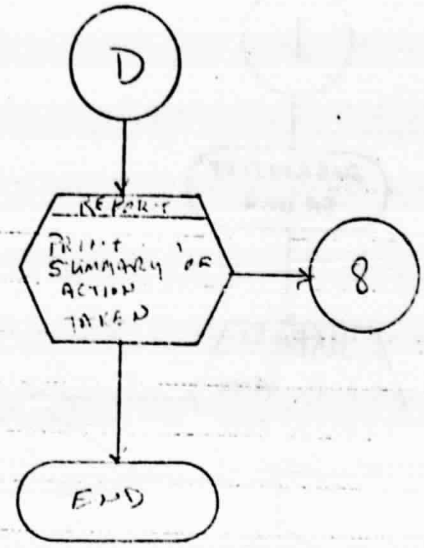
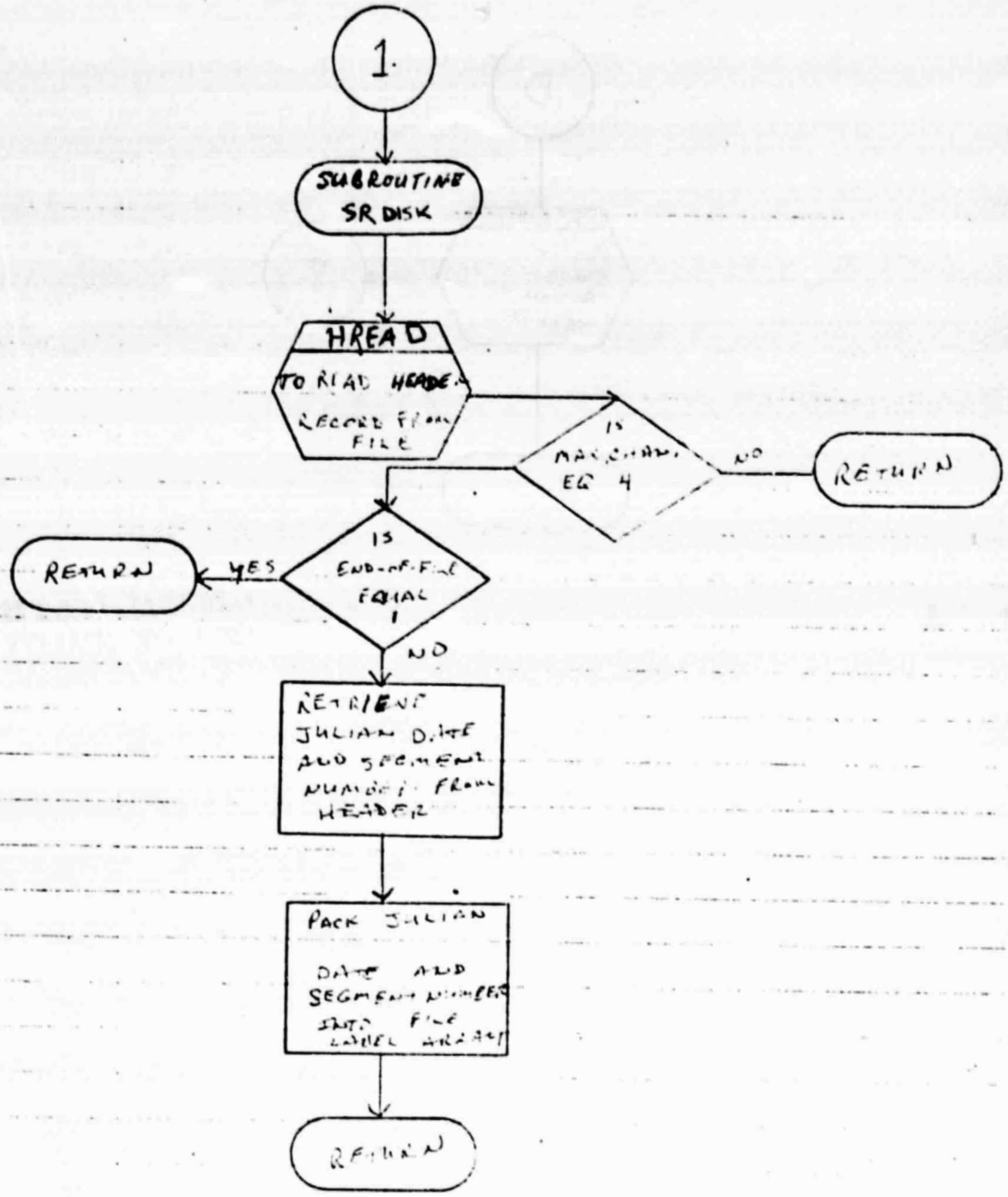


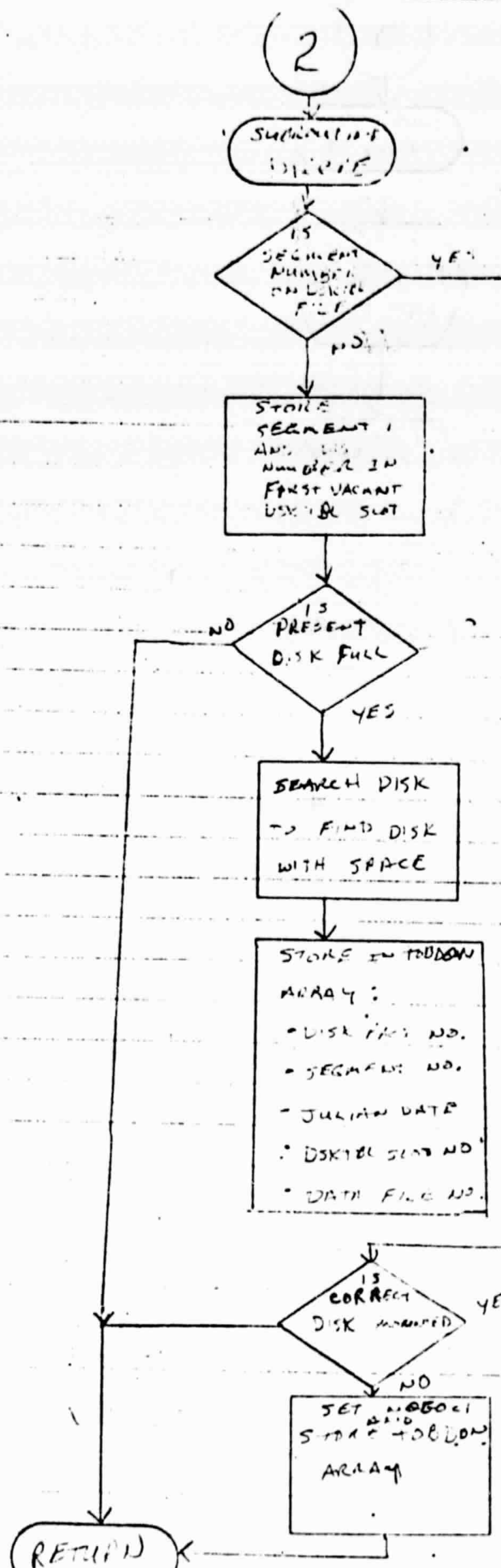
Figure 3-4

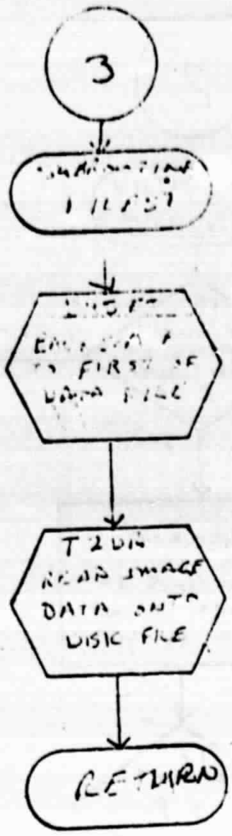

C
IS NOGOOD
NO
YES
A
FIRST
READ IMAGE FILE TO DISK
3
4
SEGMENT
5
ROOISK
READ DATA AND PASS TO DOT
STORE REPORT INFORMATION
B
STRAYS
PROCESS
DISK
6
D

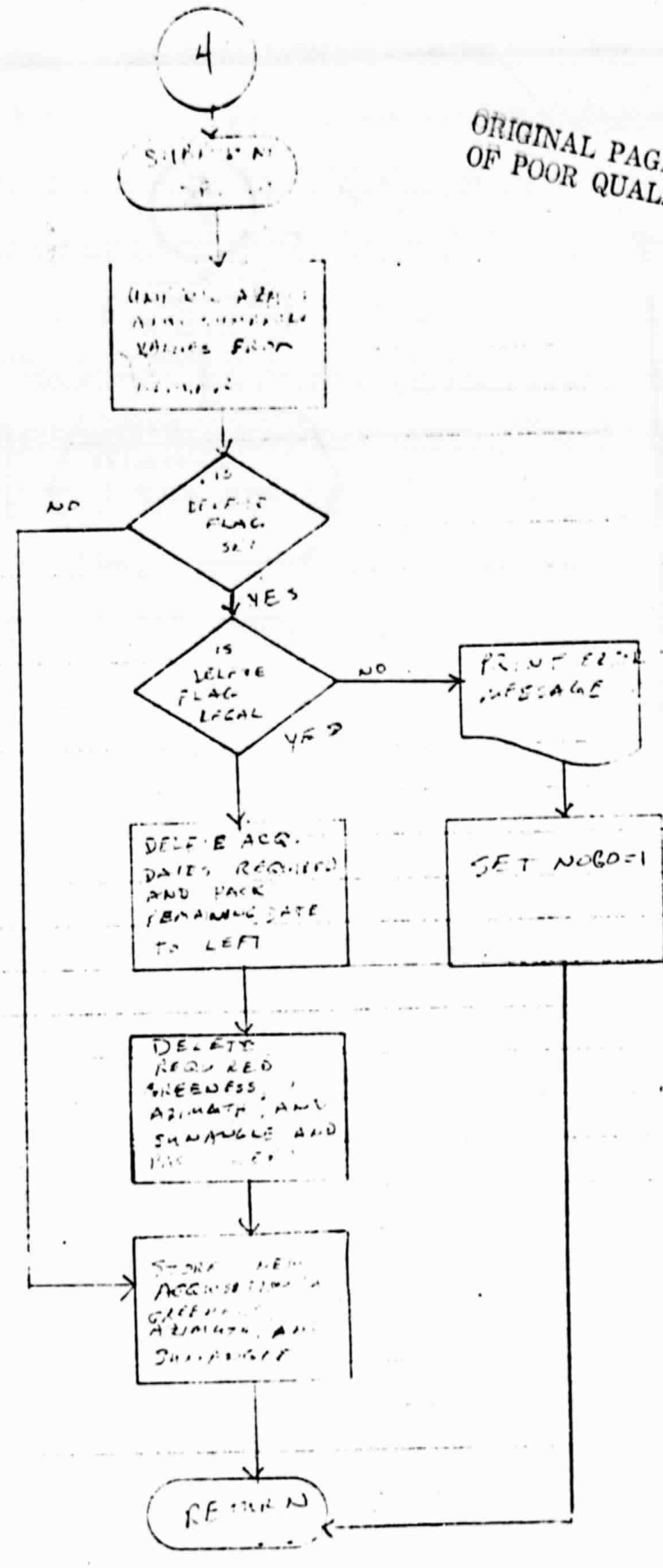

D
REPORT
PRINT SUMMARY OF ACTION TAKEN
8
END


1.
SUBROUTINE SRDISK
HREAD
TO READ HEADER RECORD FROM FILE
IS MAXCHAN EQ 4
NO
RETURN
IS END-OF-FILE EQUAL 1
YES
RETURN
NO
RETRIEVE JULIAN DATE AND SEGMENT NUMBER FROM HEADER
PACK JULIAN DATE AND SEGMENT NUMBER INTO FILE LABEL ARRAY
RETURN

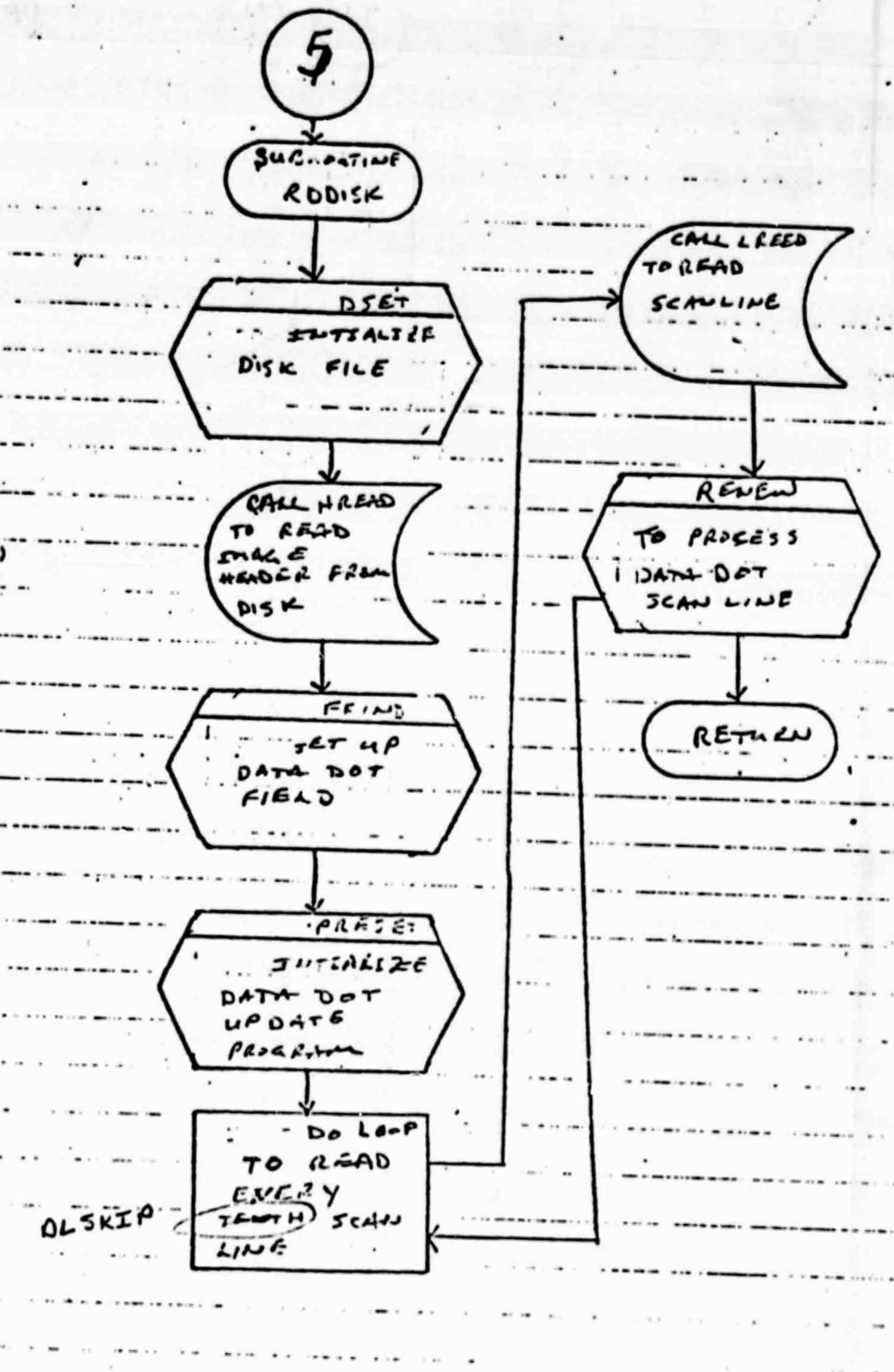
ORIGINAL PAGE IS
OF POOR QUALITY

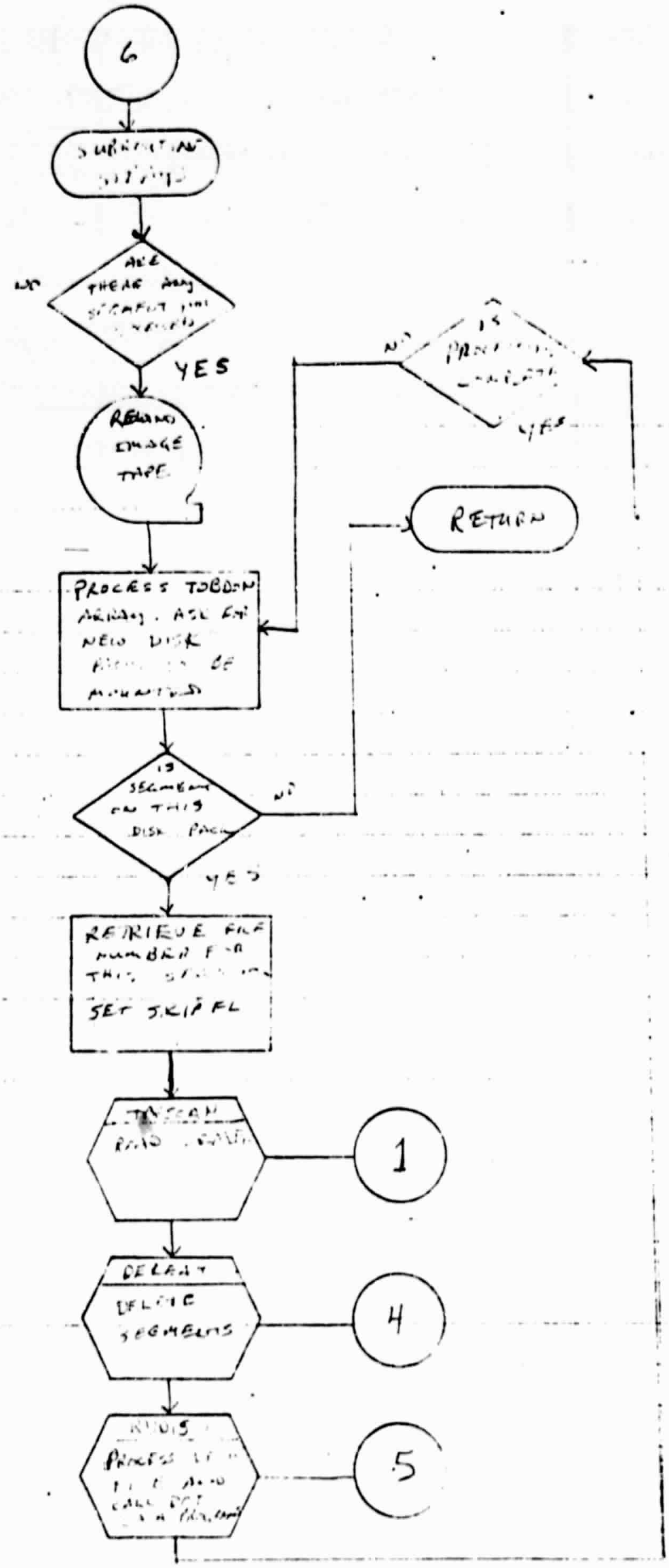


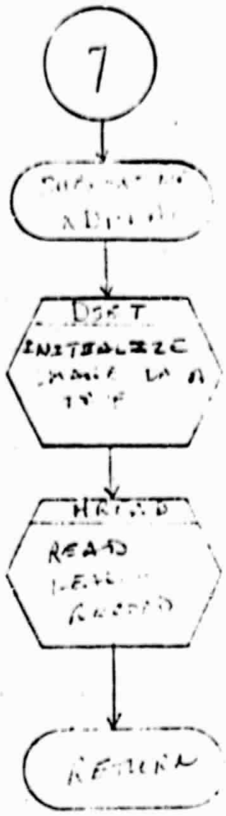

3
FIRST
T2DK
DATA
DISK FILE
RETURN

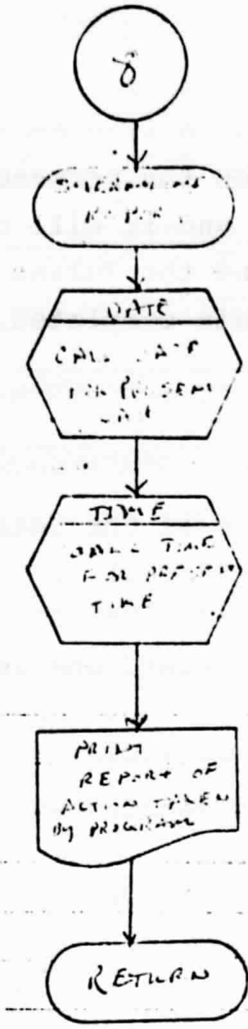


ORIGINAL PAGE IS
OF POOR QUALITY


5
SUBROUTINE RDDISK
DSET
INITIALIZE DISK FILE
CALL HREAD TO READ IMAGE HEADER FROM DISK
FFIND
SET UP DATA DOT FIELD
PRESET
INITIALIZE DATA DOT UPDATE PROGRAM
DO LOOP TO READ EVERY TENTH SCAN LINE
DLSKIP
CALL LREED TO READ SCANLINE
RENEW
TO PROCESS 1 DATA DOT SCAN LINE
RETURN


6
YES
REWIND IMAGE TAPE
RETURN
PROCESS TOBDON
NEW DISK
IS SEGMENT ON THIS DISK PACK
NO
YES
RETRIEVE FILE NUMBER FOR THIS
SET SKIP FL
1
DELETE SEGMENTS
4
5


7
INITIALIZE
READ
RETURN


8
DATE
TIME
PRINT
REPORT OF
ACTION TAKEN
BY PROGRAM
RETURN

the segment number and date. A file name is constructed for the data file to be stored on the permanent data base. This file name must be in ASCII, and it will contain the segment number (N), the year (Y), and the Julian day (D). The file name format is NNNNYYDDD.DAT. This completed, control is returned to DIRUPD.

DIRCRE

DIRCRE is called to determine if the segment is already on the data base. The disk table (DSKTBL) record is searched for the segment number being processed. DSKTBL is a double dimensional array, two words wide (N,2). Word one is the segment number and word two is the disk pack number on which that segment is stored. If the segment is not found, the now segment number is stored in the first empty segment word in the DSKTBL record. If a new segment is added, the program then determines if the disk pack which is mounted has space for this segment. If not, the program searches the other disk pack until it finds one on which to store the segment. To do this, the program has set up an array as large as the total number of segments. Then it goes through the DSKTBL and counts the number of segments on each disk. If a different disk pack is required, either for a new segment or an old segment, the necessary information needed to reactivate the segment is stored in an array (TOBDON), and a flag is set to stop processing of this segment. If the segment is to be placed on this disk pack which is mounted, processing continues with the next phase.

FILEST

FILEST stores the Image Data file on the disk pack using the Imagery Read Utility routines. The entire Image file, including header record, is read onto th permanent data base.

ORIGINAL PAGE IS
OF POOR QUALITY

DELEAT (the program for deleting acquisitions)

The data base will accommodate six acquisitions for each segment. To delete an acquisition, in order to add a new acquisition, the delete flag in the Directory record (DIRFILE) must be set to a value of 1-6. The delete flag is byte three of the DIRFILE record for each segment. If the delete flag is not set, the segment will be added in sequence. If the delete flag is set to a legal acquisition number, that acquisition is deleted and the remaining acquisitions are packed to the left. The new acquisition is then added in sequence. If the delete flag is larger than the number of acquisition stored but within the range of 1-6, the delete flag is set to zero, and the new acquisition is added. A delete flag which is out of range causes a diagnostic to be printed on the line printer, and processing on that segment is terminated. DELEAT also retrieves the green number, the sun angle, and the azimuth value from the header record and stores these values in the Directory record. The acquisition date which was retrieved earlier is also stored in the Directory record.

RDDISK

Using the Imagery Read Utility routines, RDDISK reads the data file from the permanent data base and passes the required data to the Data Dot programs. The Data Dot programs create or update the Data Dot file.

STRAYS

STRAYS is called to process segments which were not stored on the permanent data base which was mounted. The array TOBDON is used to determine which segments need to be processed and which disk packs need to be mounted. The subroutines which were used to store segments on the disk pack which was originally mounted when processing began are used by STRAYS to process the segments.

REPORT

Subroutine REPORT prints a 'summary of the action taken by DIRUPD. It lists the segment number and the date the data was recorded. The number of acquisitions which have been stored for that segment is listed. THe value of the delete flag and the date that was deleted is printed. The sun angle, azimuth, and green number are listed, followed by the number of records read from the data file, the number of parity records encountered, and the number of blocks used to store the file.

The information on pages 3-77 through 3-87 was found to be redundant and was removed.

3.5.1.2 FIELD FILE UPDATE (FLDUPD) (Programmed by Betsy Thompson; documented by Ken Pattison)

The purpose of this program is to process the DO/DU field definition card deck as generated in Del Foster format. This processed data will subsequently be used for two purposes:

a. Update the CAMS Image-100 Hybrid System permanent data base and directories.

b. After reformatting and/or interactive modification data will be offloaded in ERIPS card format for re-entry into the ERIPS system.

3.5.1.2.1 Linkages

a. Shared subroutines

1. ELAPSE
2. DSKCHK
3. FLGDOT
4. FLDINT

b. Private subroutines documented in section 3.5.1.2.8

1. CNTRL
2. NCNTRL
3. FLDST
4. NFLDST
5. FIELD
6. NFIELD
7. SEGEND
8. NSEGND
9. FLDEND
10. INDBRT
11. RDDIR
12. RDDOT
13. RDDODU

14. UPDOT
15. KOMBRT
16. EXPTD
17. DCOORD
18. JULIAN
19. WRDIR
20. WRDODU
21. WRDOT

3.5.1.2.2 Interfaces

a. Working File Name(s)

1. CARD.DAT - The file created by the batch stream deck and containing all card data input to the program.

2. OUTFILE.DAT - The formatted report file generated by the program.

b. Data Base Files

1. DIRFILE.DAT
2. DSKTBL.DAT

3.5.1.2.3 Inputs

a. Cards - Reference user's manual for example deck set-up. There are six card types.

1. SEG XXXX - Segment identifier.

2. DXXXXX or UXXXXX - Field name card and type identifier. D for DO fields, U for DU fields.

3. +XXXXXX - XXXXXX - Field vertex (pixel, line)

4. DEL XXXXX - Field delete card.

5. SEGDEL XXXX - Segment delete card

6. //END - Segment end card

3.5.1.2.4 Outputs

a. Reports - Field Definition Data Base Update Transaction Report-obtained by PIP listing of OUTFILE.DAT.

b. DIAGNOSTICS - (reference or duplicate user's manual)

c. Data base field definition files XXXXFIELD.DAT, where XXXX is the segment number. The file content and format is given in section 3.3.4.

d. Updated directory file.

3.5.1.2.5 Storage Requirements

102 blocks.

3.5.1.2.6 Description

The Field File Update program executes in a batch mode and processes the segment update deck from the Del Foster field definition system. The input deck has the general format of:

a. Control card with segment number

b. Field identification card with an action indication (default is 'ADD')

c. Up to ten vertices

d. An end of field/segment indicator

A field-defining portion of this deck will be read and edited. If all cards pass the editing and the proper disk pack is mounted, the DO/DU Fields File for this segment will be updated and the Directory File updated to reflect the addition or deletion. A maximum of 50 DO/DU fields, in combination, are allowed in the system. Each field may have a maximum of 10 vertices defined for it. If one of these vertices is not the closing vertice, the closing vertice will be provided by the program. To modify

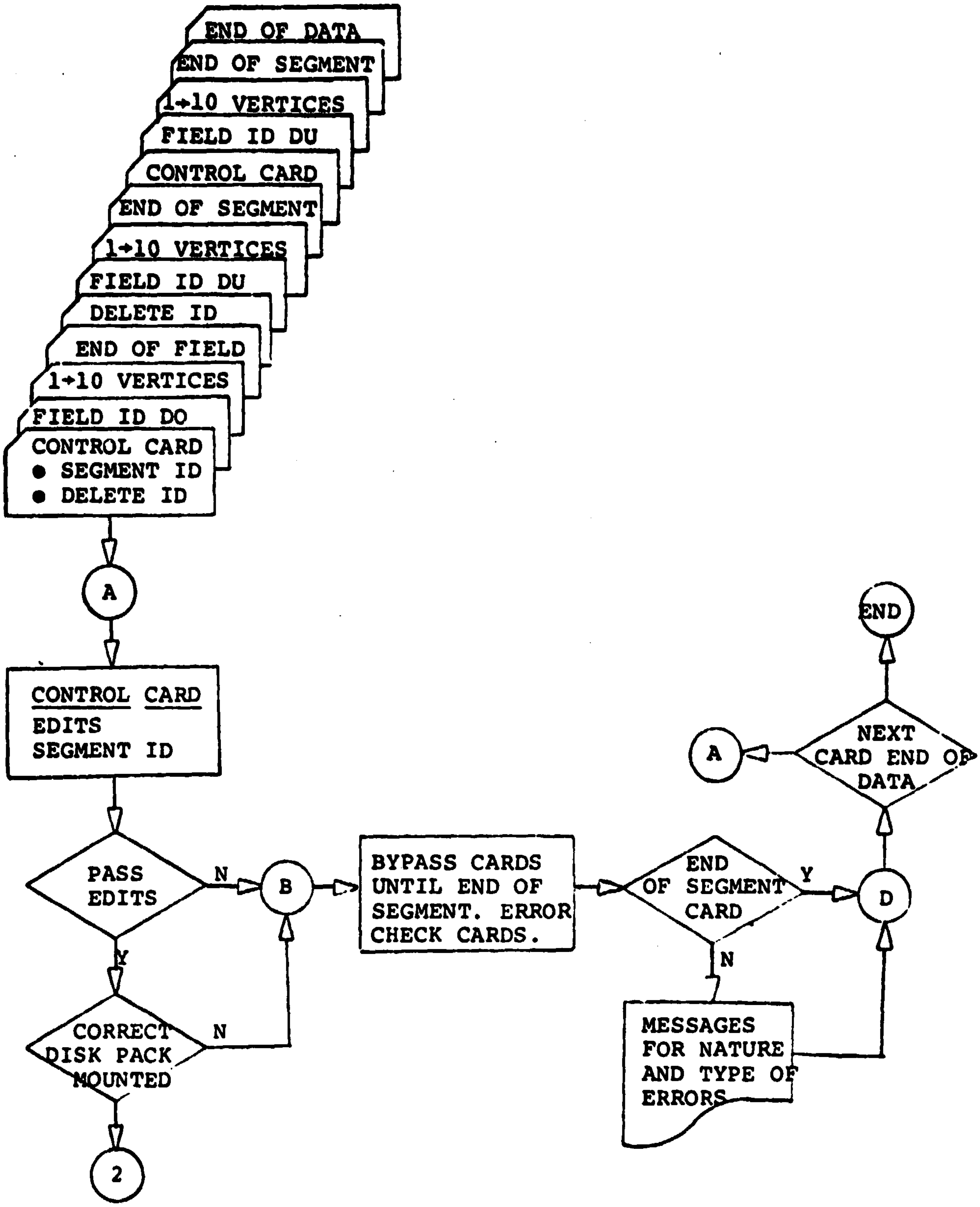


Figure 3-5.– DO/DU Field Update.

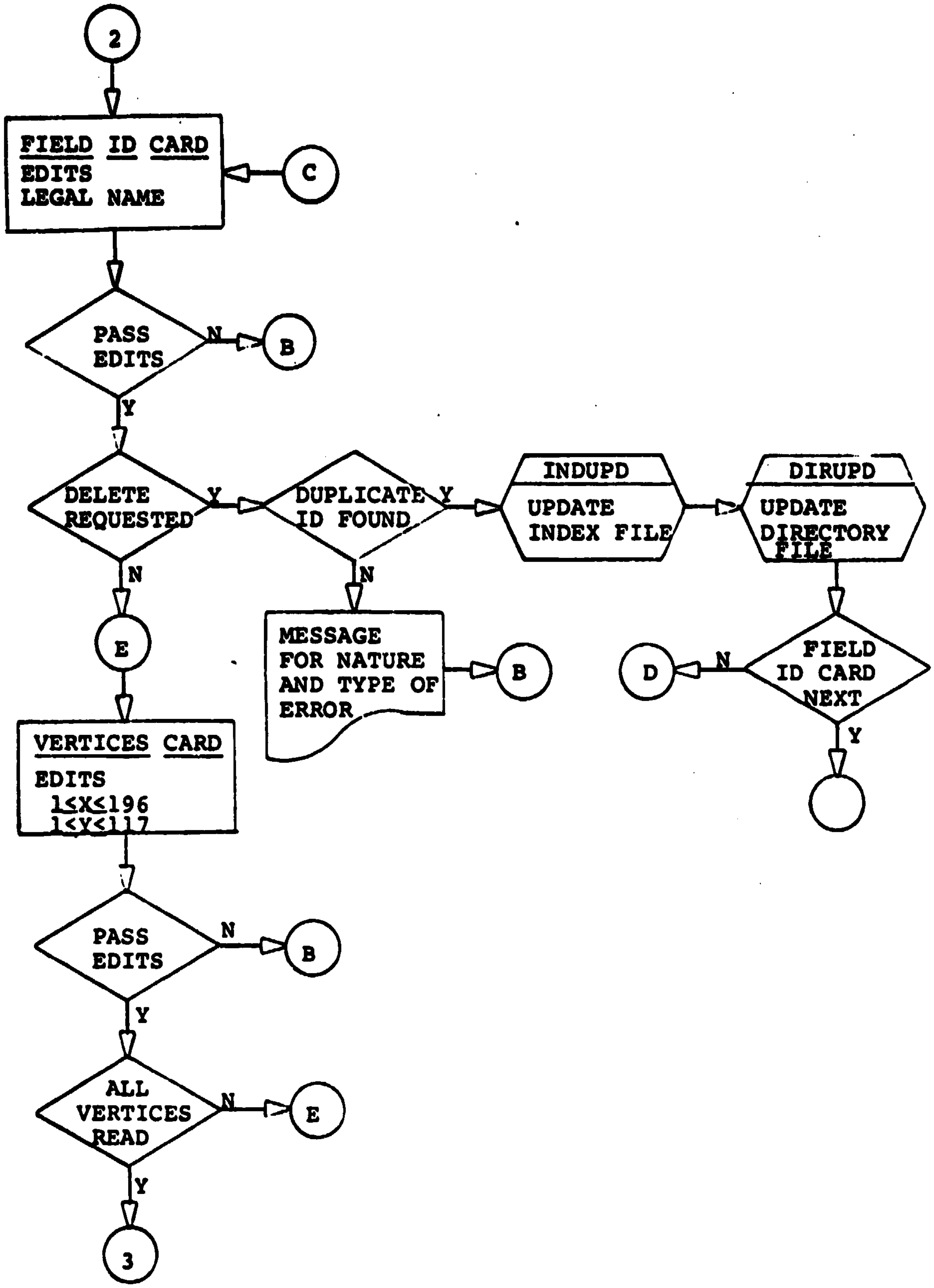


Figure 3-5.– Continued.

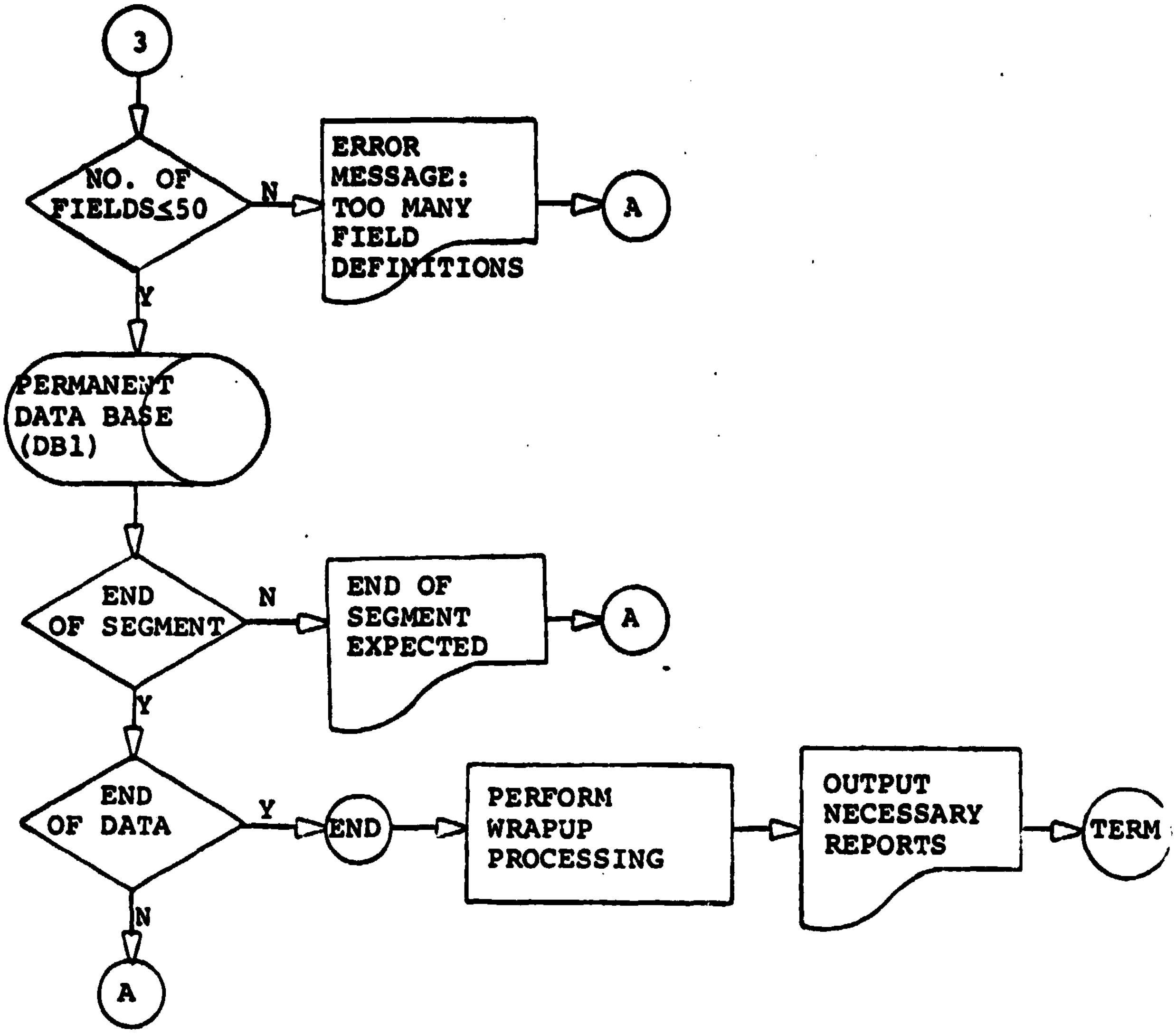


Figure 3-5.– Continued.

a vertex, the field containing this vertex must be deleted and the field reestablished in its entirety. This program is required to identify those dots in the Dot Data File that fall within a DO/DU area and update the Dot Data File accordingly.

3.5.1.2.7 Flowchart

See Figure 3-5.

3.5.1.2.8 Subroutines

3.5.1.2.8.1 CNTRL

a. Calling Sequence
   CALL CNTRL

b. Arguments as passed through local common blocks

c. Description

This subroutine is entered to process the CONTROL card. The action dictated by this control card is determined (ADD field or DELETE field) and flags set accordingly. The segment number is validated and the Directory entry accessed. If a Fields file exists for this segment, it is accessed for update, otherwise one is created. If a 'DELETE' action is scheduled, it is executed, otherwise the ADD is accomplished and the DOT Data file is updated if required.

This subroutine validates the CONTROL card, updates the required files and directories and performs the ADD or DELETE as specified for this update.

3.5.1.2.8.2 NCNTRL

This subroutine is entered where a control card was encountered out of sequence while processing the input deck.

a. Calling Sequence
   CALL NCNTRL

b. Arguments are passed through local common

c. Description

Since a CONTROL card was not expected at this point, a determination of what was expected is made. When the type of card that was expected is determined, the appropriate message is output to the printer and processing continues.

3.5.1.2.8.3 FLDST

The purpose of the subroutine is to prepare for the initialization of a new field (set of coordinate vertices) or to delete the field identified by the FIELD ID card.

a. Calling Sequence
   CALL FLDST

b. Arguments are passed through local common

c. Description

Upon entry, this subroutine determines whether this is to be the initiation of a new field or the deletion of an existing field and sets indicators as required. The field ID is then validated. If this is to be a field delete action, the field is deleted, the file is compacted and required housekeeping is accomplished. If this is to be the initiation of a new field, the next type of card expected is set and file totals are housekept.

3.5.1.2.8.4 NFLDST

This subroutine is entered when a field start card was encountered out of sequence when processing the input deck.

a. Calling Sequence
   CALL NFLDST

b. Arguments are passed through local common

c. Description

Since a field start card was encountered but not expected at this point, a determination of what was expected is made. When the type of card that was expected is determined, the appropriate message is output to the printer, required housekeeping is accomplished and processing continues.

3.5.1.2.8.5 FIELD

This subroutine is entered when a coordinate defining field card has been read and the specified coordinates must be verified.

a. Calling Sequence
   CALL FIELD

b. Arguments are passed through local common

c. Description

Upon entry, the 'X' and 'Y' coordinates are edited and if found to be in error or missing, the appropriate error message is output. The total number of coordinates is checked and if the maximum number have been processed, the flag is set to indicate the next card should be either the end of the segment or the start of the next field definition.

3.5.1.2.8.6 NFIELD

This subroutine is entered when a field definition card was encountered out of sequence while processing the input deck.

a. Calling Sequence
   CALL NFIELD

b. Arguments are passed through local common

c. Description

Since a field coordinate definition card was not expected at this point, a determination of what type of card was expected is made. When the type of card expected is determined, the appropriate message is output to the printer and processing continues.

3.5.1.2.8.7 SEGEND

The purpose of this subroutine is to process the 'END' of segment card and provide processing and report generation contingent in the end of a segment being reached.

a. Calling Sequence
   CALL SEGEND

b. Arguments are passed through local common

c. Description

Upon entry, this subroutine checks if the number of fields in this segment has been reduced to zero and, if so, this field file is deleted. If the file still exists, the field file is written (via a call to subroutine WRDODU) and the Segment Report is generated. If required, the DOT DATA FILE is updated for the segment and required housekeeping processing is accomplished (Directory update, next input expected, etc.).

3.5.1.2.8.8 NSEGND

This subroutine is entered when an end-of-segment card was encountered but was not expected while processing the input deck.

a. Calling Sequence
   CALL NSEGND

b. Arguments are passed through local common

c. Description

Since an end-of-segment card was not expected at this point, a determination of what type of card was expected is made. When the type of card expected is determined, the appropriate message is output to the printer and processing continues.

3.5.1.2.8.9 FLDEND

The purpose of this subroutine is to process the end of field definition card (//END), and provide wrap-up processing coincident with the end of a field being reached.

a. Calling Sequence
   CALL FLDEND

b. Arguments are passed through local common

c. Description

Upon entry, this subroutine forces the closing vector of the field to be established and increments the number of coordinates accordingly. If larger than 127, the 'X' coordinate value is transformed for storage in the logical buffer and then the closing vector coordinates are stored. The current update date is saved in the specified Directory entry and the indicator flags are set to expect either another field start card or an end of segment card next.

3.5.1.2.8.10 INDBRT

The purpose of this function is to find a substring of B imbedded in a substring of A

a. Calling Sequence
   K = INDBRT (A, I, N, B, J, M)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
A	A(1)		Out	The buffer to be searched.

Parameter	Dimension	Type	In/Out	Description
I		I*2	Out	Starting position of search in buffer.
N		I*2	Out	Length of buffer to be searched.
B	B(1)		In	Buffer containing string of characters
J		I*2	In	Starting position in buffer for search.
M		I*2	In	Length of string to search for.

This function searches string A looking for a match on string B. If a match is found, K is set to the starting location in A. If no match is found, K is set to zero.

3.5.1.2.8.11 RDDIR

The purpose of this subroutine is to read the direct access Directory File for the segment currently being processed.

a. Calling Sequence

```
CALL RDDIR (ID, IERR)
```

b. Arguments

Parameter	Dimension	Type	In/Out	Description
ID		I*2	In	Current segment number.
IERR		I*2	Out	Error status of Directory File read.

c. Description

Upon entry, this subroutine reads the specified Directory File entry. If during the read an error condition was setected, the appropriate error message is output to the printer.

3.5.1.2.8.12 RDDOT

The purpose of this subroutine is to read the direct access DOT DATA File for the segment currently being processed.

a. Calling Sequence

CALL RDDOT (ID, IERR)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
ID		I*2	In	Current segment number.
IERR		I*2	Out	Error status of Dot Data File read.

c. Description

Upon entry, this subroutine reads the specified Dot Data File in its entirety. If during the read an error condition was detected, the appropriate error message is output to the printer.

3.5.1.2.8.13 RDDODU

The purpose of this subroutine is to read the DO/DU (Field) File for the segment currently being processed.

a. Calling Sequence

CALL RDDODU (ID, IERR)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
ID		I*2	In	Current Segment number.
IERR		I*2	Out	Error status of Field File read.

c. Description

Upon entry, this subroutine reads the specified DO/DU (Field) File in its entirety. If during the read an error condition was detected, the appropriate error message is output to the printer.

3.5.1.2.8.14 UPDOT

The purpose of this subroutine is to update the Dot Data File when a Field File is created, updated or deleted.

a. Calling Sequence
CALL UPDOT (IOPT)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
IOPT		I*2	In	1=Segment deleted 2=Normal entry

c. Description

Upon entry, the current label for each dot is saved in temporary storage. The 'X' and 'Y' coordinates of all vertices are unpacked and saved in temporary storage. Dots falling within fields of the segment are labelled accordingly (FLGDOT) and the remaining labels restored.

3.5.1.2.8.15 KOMBRT

The purpose of this function is to compare the substrings of A and B according to the collating sequence of the hardware (PDP 11/45).

a. Calling Sequence
I=KOMBRT (A, C, N, B, J)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
A	A(1)		In	Name of buffer #1 to be compared.

Parameter	Dimension	Type	In/Out	Description
C		L*1	In	Starting position in buffer for comparison.
N		L*1	In	Length of string in comparison.
B	B(1)		In	Name of buffer #2 to be compared.
J		L*1	In	Starting position in buffer for comparison.

c. Description

This function compares two strings according to collating sequence. Neither string is changed. If they are not the same length, the short one is assumed to be filled with blanks.

3.5.1.2.8.16 EXPTD

The purpose of this subroutine is write messages to the printer according to the type of card expected but not found.

a. Calling Sequence
CALL EXPTD(I)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
I		I*2	In	Value indicating type of card expected.

c. Description

Upon entry, the type of card expected (I) is tested and, depending on the value of 'I', the appropriate message is output to the printer.

3.5.1.2.8.17 DCOORD

The purpose of this subroutine is to decode coordinate data as read from Del Foster cards.

a. Calling Sequence
CALL DCOORD (IBUF, INX, IXY, IERR)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
IBUF	IBUF(1)		In	Contains data to be decoded.
INX		I*2	In	Index vertice array (number of coordinates)
IXY		I*2	In	1=pixel 2=line
IERR		I*2	Out	Error status of coordinate processing.

c. Description

Upon entry, this subroutine checks to insure the maximum number of vertices has not been exceeded. All syntactically legal coordinates are decoded and stored in the vertex array. For all erroneous coordinate values, the error indicator is set and an appropriate error message is output.

3.5.1.2.8.18 JULIAN

The purpose of this subroutine is to correct the current Gregorian date to the current Julian date.

a. Calling Sequence
CALL JULIAN (YR, MO, DY, JULIO)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
YR		I*2	In	Gregorian year
MO		I*2	In	Gregorian month
DY		I*2	In	Gregorian day
JULIO		I*2	Out	Julian date (1-365)

c. Description

This subroutine generates the current Julian date in the following manner.

- Add current day to total number of days (by month) passed this year.
- Determine if this is a leap year or not.
- If not a leap year, subtract one day.

3.5.1.2.8.19 WRDIR

The purpose of this subroutine is to write the direct access Directory File entry for the segment currently being processed-

a. Calling Sequence
   CALL WRDIR (ID, IERR)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
ID		I*2	In	Current segment number.
IERR		I*2	Out	Error Status of Directory File write.

c. Description

Upon entry, this subroutine writes the specified Directory File entry. If during the write and error condition was deleted, the appropriate error message is output to the printer.

3.5.1.2.8.20 WRDODU

The purpose of this subroutine is to write the DO/DU (Field) File for the segment currently being processed.

a. Calling Sequence
   CALL WRDODU (ID, IERR)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
ID		I*2	In	Current segment number.
IERR		I*2	Out	Error status of Field File write.

c. Description

Upon entry, this subroutine writes the specified DO/DU (Field) File in its entirety. If during the write an error condition is detected, the appropriate error message is output to the printer.

3.5.1.2.8.21 WRDOT

The purpose of this subroutine is to write the direct acces Dot Data File for the segment currently being processed.

a. Calling Sequence
   CALL WRDOT (ID, IERR)

b. Arguments

Parameter	Dimension	Type	In/Out	Description
ID		I*2	In	Current segment number.
IERR		I*2	Out	Error status of Dot Data File write.

c. Description

Upon entry, this subroutine writes the specified Dot Data File in its entirety. If during the write an error condition is detected, the appropriate error message is output to the printer.

3.5.1.3 Dot Data Update/Generation (DOTUPD) (Programmed and documented by Don Loe

3.5.1.3.1 Linkages

a. FORTRAN

1. ASSIGN
2. CLOS$
3. OPEN$
4. IDATE

b. Shared subroutines and utilities

1. DSET
2. DSKCHK
3. ELAPSE
4. ERRMES
5. FSTVID
6. FSTVID
7. HVFY
8. LECTAP
9. RREAD
10. SUBSTR

c. Private subroutines

1. DIRLOD
2. DIRUPD
3. OPMESS
4. RDCARD
5. SKIP
6. UNPAK
7. UPDATE
8. CATLOG
9. JULIAN
10. ALLWPD
11. CARDIN
12. COMPAR

3.5.1.3.2 Interfaces External Interfaces

a. Common Name CAMSPARAM.INC

This common name is used as input only for parameters MAXCAT, MAXCHN, NPIX, NLIN, NDOTS, DLSKIP and DSSKIP.

b. Working file names

1. FOR00I.DAT
2. DIRFILE.DAT
3. XXXXDOTS.DAT

3.5.1.3.3 Inputs

a. Update card deck - see appendix

b. Dot data file

3.5.1.3.4 Outputs

a. Reports - a printer listing of the update card deck.

b. Diagnostics

1. Processing on segment 'XXXX' will be discontinued---Processing will continue with a new segment or halt with an E.O.F.

2. ERROR---Only one TYPE1 value allowed (Error message #1 is then printed)

3. ERROR---Only one START value allowed (error message #1 is then printed)

4. Error on start card---Value less than 1 or greater than 60 (message #1 is then printed)

5. Improper sequence or improper data or column misalignment on the following card. (The card is printed and then message #1 is printed.)

3.5.1.3.5 Storage Requirements

DOTUPD occupies 54 blocks.

3.5.1.3.6 Description

The Dot data update program executes in a batch mode and processes the batch update of the LABEL (analyst) and the type fields.

It processes a deck with the following parameters:

a. A control card with segment number.

b. A Type 1 value

c. A start value

d. A data deck with the option to include a Type 1 value and/or a label value.

3.5.1.3.7 Flow Charts

See figure 3-6.

3.5.1.3.8 Subroutines

3.5.1.3.8.1 DIRLOD(PTR)

a. Calling sequence
CALL DIRLOD(PTR)

Parameter	Dimension	Type	In/Out
PTR	(1)	Integer	In

PTR - a pointer to the proper record located in the directory file.

b. Description

This subroutine reads a record from the directory file and selects from this record the category names. These names are then stored into the 'CATNAM' array.

DOT DATA UPDATE/GENERATION

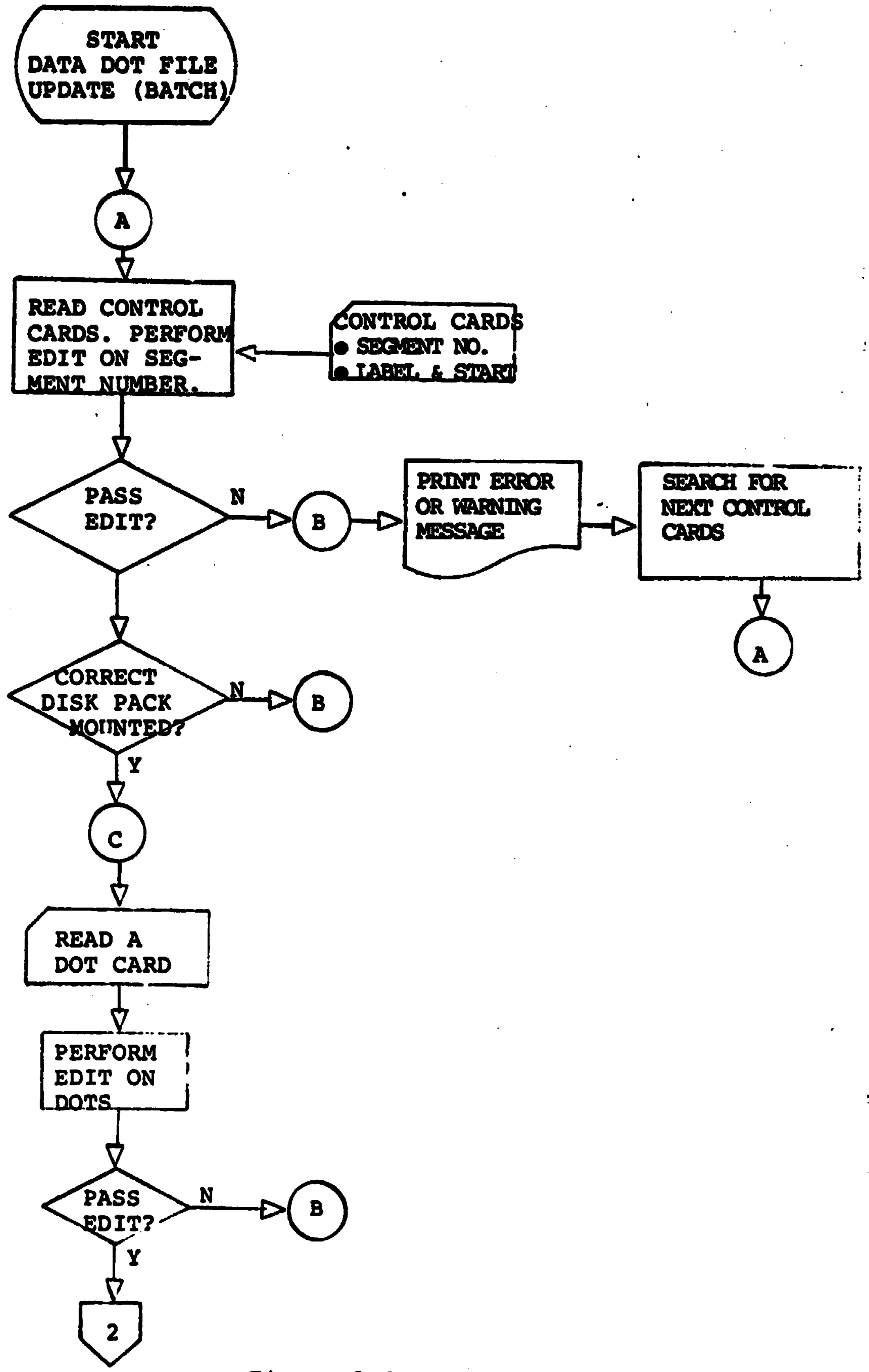


Figure 3-6

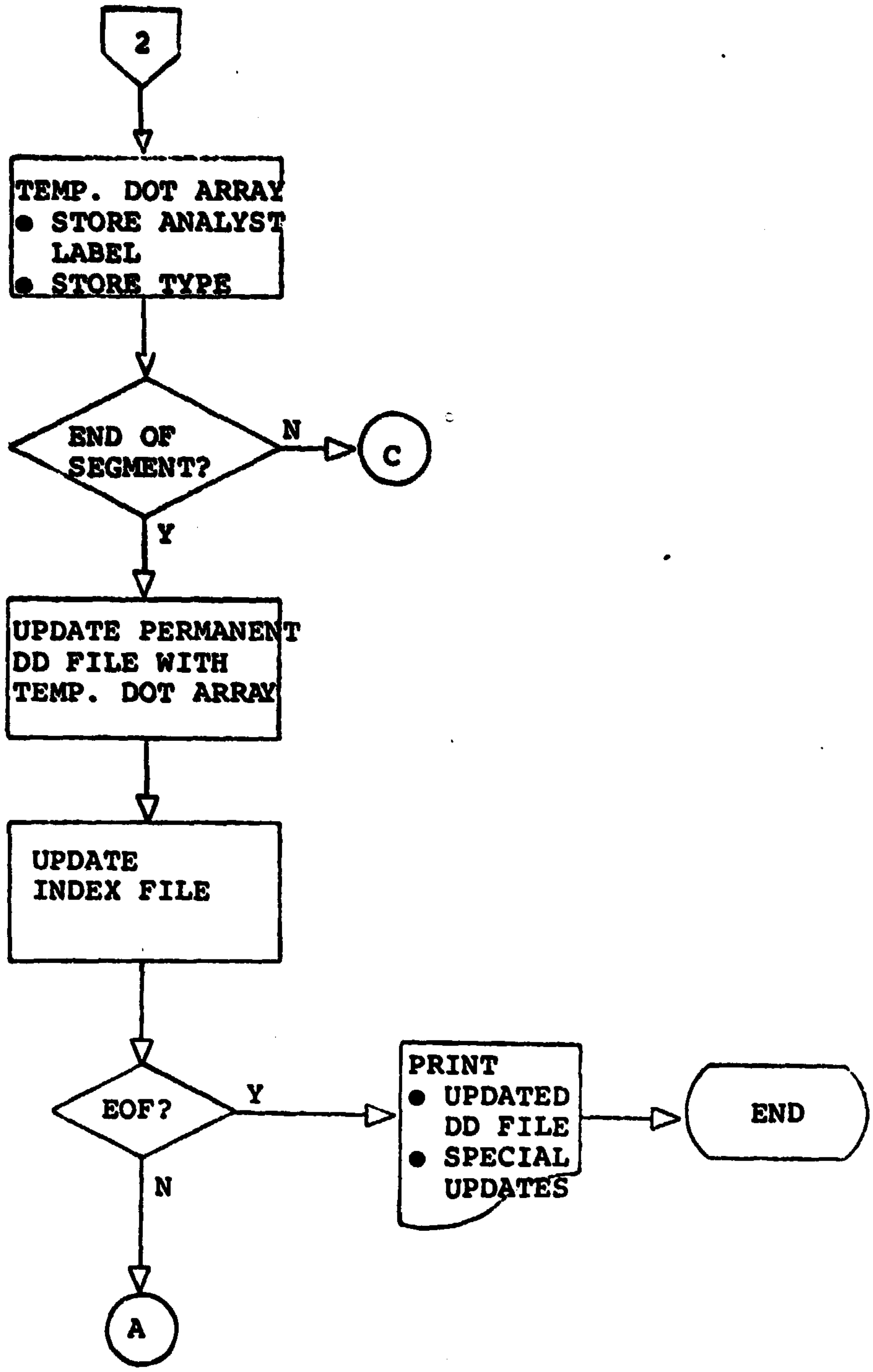


Figure 3-6 (cont)

3.5.1.3.8.2 Subroutine DIRUPD

a. CALL DIRUPD (START,TYPE1)

Parameter	Dimension	Type	In/Out
START	(1)	Integer	In
TYPE1	(1)	Integer	In

START - the START value read from cards

TYPE 1 - The TYPE1 value read from cards

b. Description

This subroutine updates a directory file record (see above) with the Julian day, year, START value, and the TYPE1 value. The record is then written back to the directory file.

3.5.1.3.8.3 OPMESS

a. Calling sequence

CALL OPMESS (DISKNM)

Parameter	Dimension	Type	In/Out
DISKNM	(1)	Integer	In

DISKNM - Disk number of the disk to be mounted.

b. Description

Prints the following message to the decwriter:

THE NEXT SEMENTS TO BE PROCESSED WILL BE FOUND ON DISK NUMBER __, IF YOU WISH TO CONTINUE, PLEASE MOUNT DISK NUMBER NUMBER __, THEN TYPE THE LETTER (C). IF YOU WISH TO ABORT THIS JOB THEN TYPE IN THE LETTER (X)---THANK YOU.

3.5.1.3.8.4 FUNCTION RDCARD

a. Calling sequence

RDCARD(CARD)

Parameter	Dimension	Type	In/Out
CARD	(40)	Integer	Out

b. Description

1. Checks status on card read

   a. IF STATUS = 1 - continue processing
      IF STATUS = 2 - Set function equal to 5 and return
      IF STATUS = 3 - Set function equal to 6 and return

2. Identifies the card read as one of the following - SEGMENT TYPE1, START, DOT.

3.5.1.3.8.5 SKIP

a. Calling sequence
   CALL SKIP

b. Description

   Skips to next segment card.

3.5.1.3.8.6 UNPAK

a. Calling sequence
   CALL UNPAK (CARD,DOTFLG,TYP1)

Parameter	Dimension	Type	In/Out
CARD	(40)	Integer	Out
DOTFLG	(1)	Integer	In

CARD - an array containing 'DOT' information.
DOTFLG - a flag indicating first card to be processed.

b. Description

   Unpacks a 'DOT' card, i.e., obtains the following values:

1. TYPE1, START, and all dot numbers on any one card.

3.5.1.3.8.7 UPDATE

a. Call sequence

CALL UPDATE (TYP1,INIT)

Parameter	Dimension	Type	In/Out
TYP1	(1)	Integer	In
INIT	(1)	Integer	In

TYP1 - same as TYPE1

INIT - if equal to a "c" then a complete update will be performed.

If equal to a zero, a partial update will be performed.

b. Description

1. Reads the DOT DATA file into a work array
2. Calls on 'ALLUPD' to update the work array
3. Writes the updated work array back to the DOT DATA file.

3.5.1.3.8.8 CATLOG

a. Call sequence

CALL CATLOG (LABEL,LABNUM)

Parameter	Dimension	Type	In/Out
LABEL	(1)	Integer	In
LABNUM	(1)	Integer	Out

LABEL - an alpha character(s) denoting a classification by an A.I.

LABNUM - a number indicating the position the LABEL has in the 'CATNAM' array.

b. Description

Search through the 'CATNAM' array to determine the position of the input alpha LABEL.

3.5.1.3.8.9.8 JULIAN

a. Calling sequence

CALL JULIAN (MO,DAY,YR)

Parameter	Dimension	Type	In/Out
MO	(1)	Integer	In
DAY	(1)	Integer	In
YR	(1)	Integer	In

MO - month

DAY - day

YR - year

b. Description

Calculates the Julian date

3.5.1.3.8.9.10 ALLUPD

a. Calling sequence

CALL ALLUPD (WORK,TYP1,INIT)

Parameter	Dimension	Type	In/Out
WORK	(2,209)	Integer	Input and Output
TYP1	(1)	Integer	In
INIT	(1)	Integer	In

WORK - the DOT DATA array to be updated

TYP1 - same or TYPE1

INIT - a flag to indicate complete or partial update

b. Description

Update the TYPE and label values in a DOT DATA record with the following conditions:

1. If a complete update is needed, all values of TYPE and LABEL are set to zero before any updates are made.

2. If a partial update is needed, then only those TYPE and LABEL values indicated on the card deck are changed.

3.5.1.3.8.10 CARDIN

a. Calling sequence

CALL CARDIN (CARD)

Parameter	Dimension	Type	In/Out
CARD	(40)	Integer	Out

CARD - an array to hold one card image

b. Description

Read a card.

3.5.1.4 CAMS/CAS Interface - Cluster Statistics (BSTAT) (Programmed and Documented by A. Holley)

The purpose of this program is to read the CAMS/CAS Interface Tape, extract classification results and cluster statistics for the given segment, and save this data on the data base disk.

3.5.1.4.1 Linkages

A. IMALB

NONE

B. FORTRAN (or system)

CLOSE, OPEN, DATE, SECONDS, TIME, ASNLUN, GETADR, QIO, WAITFR

C. Shared subroutines and utilities

DSKCHK

KAUTH

D. Private subroutines

CDRED

3.5.1.4.2 Interfaces

A. Common Name COM1

Parameter	Updated by Subroutine	Referenced by Subroutine
ACDATE	BSTAT	
CHNVEC	BSTAT	
NOCHAN	BSTAT	
NOSUB	BSTAT	
SUBCAT	BSTAT	
SUBPOP	BSTAT	
CATKNT	BSTAT	

CATTH	BSTAT
NODO	BSTAT
NODU	BSTAT
NOTH	BSTAT

B. Common Name COM2

Parameter	Updated by Subroutine	Referenced by Subroutine
TDATE3	BSTAT	
NOCAT	BSTAT	BSTAT
CATNAM	BSTAT	BSTAT

C. Common Name COM5

DISKID		BSTAT
		DSKCHK

D. Data Base File Names

SYO: [300,300] DIRFILE.DAT

DB2: [300,300] XXXXTSTAT.DAT

(XXXX is a four digit segment number)

3.5.1.4.3 Inputs

A. The CAMS/CAS Interface tape format specification is in Earth Resources Data Format Control Book (PHO-TR543 Rev. A, Change 3)

B. The program reads two input cards. The first card gives the input tape device code in column 1 - An M or X. The second card, also in column 1, has the unit number - 0 or 1.

3.5.1.4.4 Outputs

A. The program outputs the CAMS I-100 Data Base Transaction Report, listing segments and a processing summary.

B. DIAGNOSTICS:

CARD ERROR - this indicates the absence of the required inputs on the input cards. I/O STATUS BLOCK ERROR CODE. This indicates a fatal tape error condition (not a parity error).

TAPE ERRORS ENCOUNTERED = . This informational message is given at the end of processing and indicates that one or more tape parity errors were encountered.

3.5.1.4.5 <u>Storage</u>

BSTAT occupies 77 blocks.

3.5.1.4.6 BSTAT Description

The CAMS/CAS statistical file build program operates in a batch mode. The first action of the program is to output the processing summary header.

Subroutine CDRED is called to initialize tape operations. CDRED reads two input cards defining the tape device code and unit number. The program reads to a recognition segment record and obtains the segment number. DSKCHK is called to see if the segment is associated with the current disk. If not, a message is written for the processing summary report, and the program reads to the next recognition segment record on tape. If the segment is to be processed, the directory file is opened and a file record is read into common area COM1. Common variables are decoded. The program processes subclass related classification results, accumulating pixel counts by category. The program then reads the first statistics record for the segment. Statistics for each subclass are decoded, a division is performed according to the implied decimal point, and the data is stored in an array in the order in which the statistics data sets appear on the tape. Finally, the statistics file is opened. The first record written to the file is the COM1 common block. This is followed by means, standard deviations, greenness calculations, for each subclass, in groups of four channels. One record is written for each subclass or cluster.

The process is repeated for each segment on the tape, until the program reads the second end of file mark.

3.5.1.4.7 Flowchart

Figure 3-7. (For detailed flowchart see Volume 2).

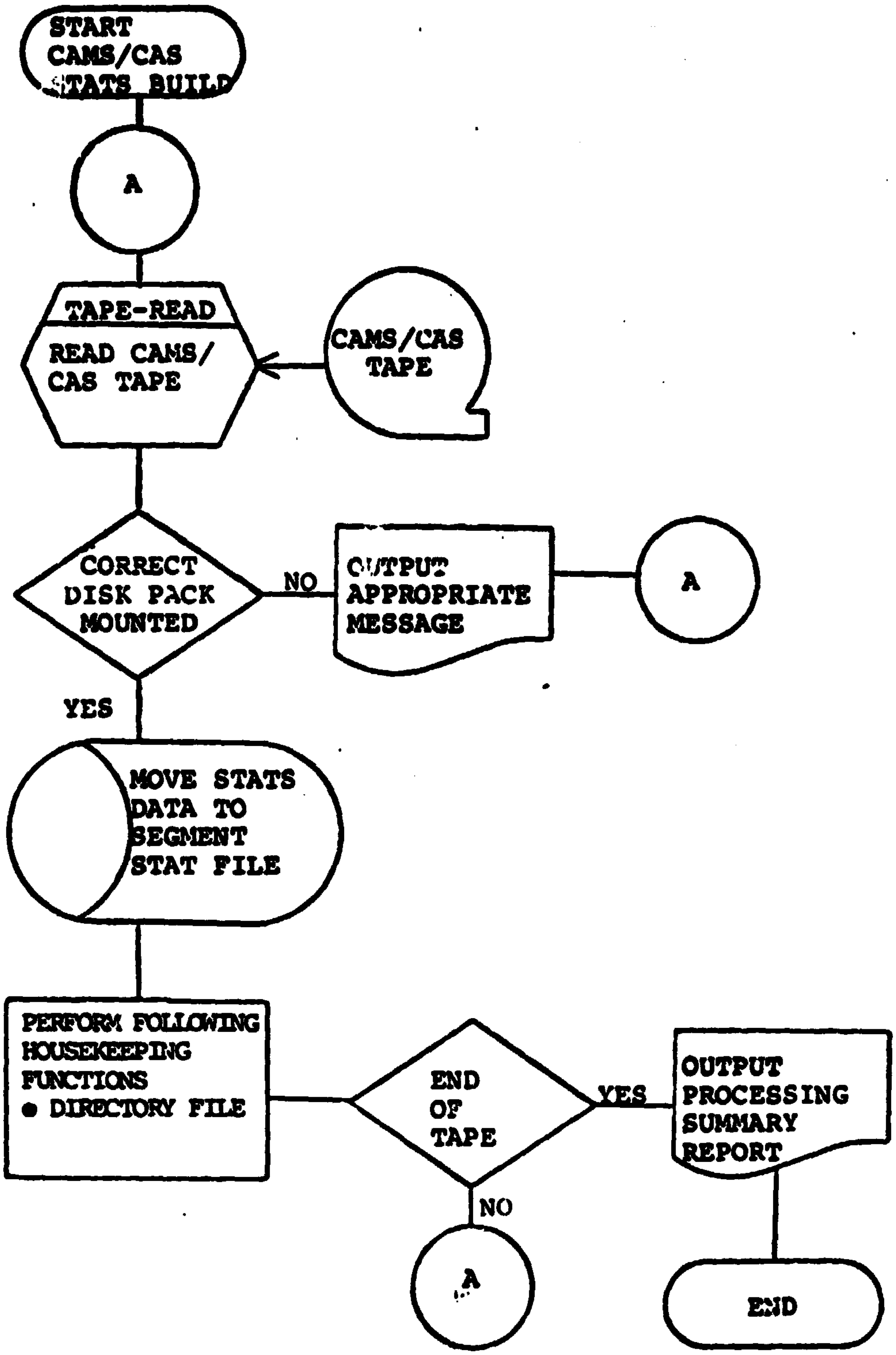


Figure 3-7.- CAMS/CAS Interface Tape.

3.5.1.4.8 Subroutines CDRED

CDRED handles tape operations for BSTAT.

- Calling sequence

  Call CDRED (IBUF,R,FILE)

- Arguments

Parameter	Dimension	Type	In/Out	Description
IBUF	800	BYTE	In	Input buffer
R		I	In	Control variable
File		I	Out	EOF count

- Description

If the argument R is 0, CDRED reads a card input from Unit 1. If the first column of the card is not 'M' or 'X', defining the tape device code, the program writes 'CARD ERROR' to the line printer and stops. CDRED reads a second card. If a '0' or '1' does not appear in Column 1, defining unit number, the same action is taken. The system subroutine ASNLUN is called to assign the unit number. The program calls GETADR to get the input buffer address set up in the IPRM array. The system routine QIO is called with a tape rewind function code. Next QIO is called with a read function code. If the read is successful the program returns. If an EOF was encountered, FILE is increased by 1. If an error code of -4 (parity) is indicated, an error count is incremented before a return is made. For an error code other than -4, the program prints a message and stops.

If, on entering the subroutine, R is greater than 0, the above logic is executed beginning with the read call to QIO.

On entering with a negative R, the subroutine does a rewind call to QIO. At this point the parity error count is checked. If the count is positive, a message is printed before returning to the main program.

3.5.1.5 Classification/Cluster Map Tapes (DTRM) (Programmed and documented by Don Loe)

3.5.1.5.1 Linkages

a. FORTRAN

1. ERRSET
2. SECNDS
3. CLOS$
4. OPEN$
5. DATE
6. TIME

b. Shared subroutines and utilities

1. DSET
2. DSKCHK
3. ELAPSE
4. ERRMES
5. FSTVID
6. HPROS
7. HVFY
8. LECTAP
9. RREAD
10. SUBSTR
11. LREED

c. Private subroutines

1. DIRUPD
2. FADE
3. FMAINT
4. HEADIN
5. MAPUPD
6. REPORT
7. OPMESS
8. LREED

3.5.1.5.2 Interfaces

a. External Interface

1. Common Name CAMSPARAM.INC

   This common name is used as input only for parameters MAXCHN,NPIX,NLIN,NDOT.

2. Working file names:

   TCLUNM.MAP
   TCLANM.MAP
   TSTAT.DAT
   DIRFILE.DAT

3.5.1.5.3 Inputs

The inputs to this program are:

a. DTRM Tape from ERIPS containing

   1. Classification map reflecting category level color codes

   2. The unconditional cluster map reflecting individual clusters with unique color codes.

3.5.1.5.4 Outputs

The outputs from this program are:

a. Updated temporary classification map and cluster map

b. Updated system directory

c. Updated system index file

d. Classification/cluster map update report

e. Appropriate error messages.

3.5.1.5.5 Storage Requirements

DTERM occupies 108 blocks.

3.5.1.5.6 Description

The classification/cluster map build program executes in an off line mode and processes the universal formatted data from the DTRM tape. To build these files, the DTRM tape is read, specific flags are checked and the indicated classification/cluster maps are transferred to the appropriate data bare disk pack. Upon completion of the transfer, the Index File will be updated or required.

3.5.1.5.7 Flow Chart

See Figure 3-8.

3.5.1.5.8 Subroutines

3.5.1.5.8.1 DIRUPD

a. Calling sequence

CALL DIRUPD (YR,JULDAT,ID,PTR)

Parameter	Dimension	Type	In/Out
YR	(1)	Integer	In
JULDAT	(1)	Integer	In
ID	(1)	Integer	In
PTR	(1)	Integer	In

YR - year
JULDAT - Julian date
ID - flag to indicate classification or cluster
PTR - pointer to records in the directory file

b. Description

Reads the appropriate record from Directory File, retrieve the category names, updates the record with new dates and writes the update record back to the Directory File.

DTRM TAPE READ AND BUILD

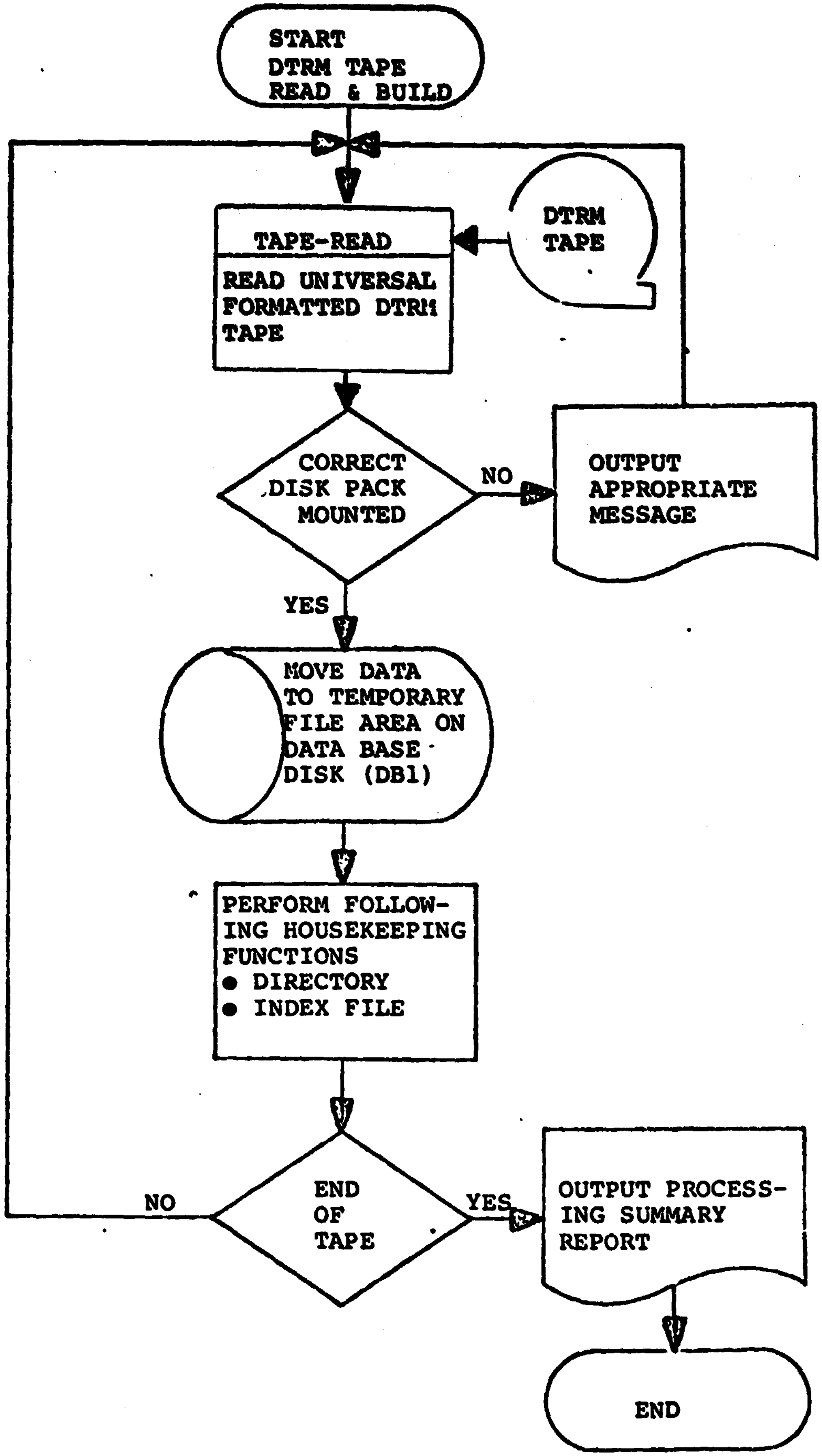


Figure 3-8

3.5.1.5.8.2 FADE

a. Calling sequence

CALL FADE

b. Description

Closes all files opened by the subroutine FMAINT.

3.5.1.5.8.3 FMAINT

a. Calling sequence

CALL FMAINT (SEGNUM,NPIX4,NLIN,ERROR)

Parameter	Dimension	Type	In/Out
SEGNUM	(1)	Integer	In
NPIX4	(1)	Integer	In
NLIN	(1)	Integer	In
ERROR	(1)	Integer	In

SEGNUM - segment number

NPIX4 - record size in double words

NLIN - maximum number of records

ERROR - error flag

b. Description

Opens the following files using the appropriate segment number:

1. TCLUNM.MAP
2. TCLANM.MAP
3. TSTAT.DAT

3.5.1.5.8.4 HEADIN

a. Calling sequence

CALL HEADIN (ID,SEGNUM, JULDAT,YR,EOF)

Parameter	Dimension	Type	In/Out
ID	(1)	Integer	Out
SEGNUM	(1)	Integer	Out
JULDAT	(1)	Integer	Out
YR	(1)	Integer	Out
EOF	(1)	Integer	Out

ID - flag indicating classification or cluster

SEGNUM - segment number

JULDAT - Julian date

YR - year

EOF - end of file flag

b. Description

Reads the Universal formatted DTRM tape, and from the header portion, selects the above indicated parameters.

3.5.1.5.8.5 MAPUPD

a. Calling sequence

CALL MAPUPD (ID)

Parameter	Dimension	Type	In/Out
ID	(1)	Integer	In

ID - a flag indicating classification or cluster

b. Description

Reads the data records from the Universal formatted DTRM tape. Based on the ID flag, classification or cluster data will be retrived and converted to cluster number values or category position values.

3.5.1.5.8.6 REPORT

a. Calling sequence

CALL REPORT (AID,ASEG,ADSK,AYR,AJUL,FILNUM)

Parameter	Dimension	Type	In/Out
AID	(30)	Integer	In
ASEG	(30)	Integer	In
ADSK	(30)	Integer	In
AYR	(30)	Integer	In
AJUL	(30)	Integer	In
FILNUM	((1)	Integer	In

AID - all ID numbers

ASEG - all segment numbers

ADSK - all disk numbers pertaining to segments

AYR - all year dates

AJUL - all Julian dates

FILNUM - number of files processed

b. Description

When 'DTERM' has processed all data files, a report is generated to the line printer using the above parameters.

3.5.1.5.8.6 OPMESS

a. Calling sequence

CALL OPMESS (DISKNM)

b. Arguments

Parameter	Dimension	Type	In/Out
DISKNM	(1)	Integer	In

DISKNM

DISKNM - The number of the next disk to be mounted.

c. Description

Prints the following message to the decwriter:

THE NEXT SEGMENTS TO BE PROCESSED WILL BE FOUND ON DISK NUMBER __; IF YOU WISH TO CONTINUE, PLEASE MOUNT DISK NUMBER __; THEN TYPE THE LETTER (C). IF YOU WISH TO ABORT THIS JOB, THEN TYPE IN THE LETTER (X)---THANK YOU.

3.5.1.6 Segment Delete (SEGDEL) (Programmed and documented by Ken Pattison

This non menu, interactive program accepts a segment number as input, validates the segment number, and for each valid segment number deletes all associated files on the Permanent Data Base Disk (DB2:). The user may delete as many segment associated files as he wishes.

3.5.1.6.1 Linkages

a. IMALIB

b. FORTRAN

c. Shared Subroutines

   1. ELAPSE
   2. DSKCHK

d. Private Subroutines

   1. SUBSTR

3.5.1.6.2 Interfaces

a. Data base names

   1. DSKTBL.DAT
   2. DIRFILE.DAT

3.5.1.6.3 Inputs

a. Key-In

   1. KEYIN to (D)ELETE ANOTHER SEGMENT
   2. PLEASE MOUNT DISK PACK NO. XXXX
   3. DO YOU WISH TO (C)ONTINUE
   4. INPUT SEGMENT NUMBER

3.5.1.6.4 Outputs

a. Reports

   1. Segment Delete CAMS I-100 Data Base Transaction Report (Reference User's Manual for contents)

b. Diagnostics

1. WRONG DISK PACK MOUNTED - Mount correct dispatch and rerun the job
2. SEGMENT NUMBER NOT ON DATA BASE - Input another segment number

3.5.1.6.5 Storage Requirements

SEGDEL occupies 45 blocks.

3.5.1.6.6 Description

The Segment Delete program accepts a segment number as input. For every valid segment number the following associated files are deleted:

- Dot Data File
- The imagery data associated with each acquisition
- Fields File (if present)
- Temporary Statistics File (if present)
- Permanent Statistics File (if present)
- Temporary Classification Map (if present)
- Permanent Classification Map (if present)
- Temporary Cluster Map (if present)
- Permanent Cluster Map (if present)
- Directory File (specific entry only)
- Disk Index File (specific entry only)

When the associated data base files have been deleted and the system files updated additional segments may be deleted or the job may be terminated at will.

3.5.1.6.7 Flowchart

See Figure 3-9.

3.5.1.6.8 Subroutine SUBSTR

To move a substring of A into a substring of B

- Calling Sequence

  CALL SUBSTR (A,I,N,B,J,M)

- Arguments

Parameter	Dimension	Type	In/Out	Description
A	A(1)		IN	String name containing data to be moved
I		I*2	IN	Starting position
N		I*2	IN	Length
B	B(1)		OUT	Receiving string name
J		I*2	OUT	Starting position
M		I*2	OUT	Length

- Description

The contents of string name 'A' starting at 'I' for 'N' characters is moved to string name 'B' starting at 'J' for 'M' characters. If string 'B' is shorter than string 'A', characters on the right will be truncated. If string 'B' is longer, it will be blank filled on the right.

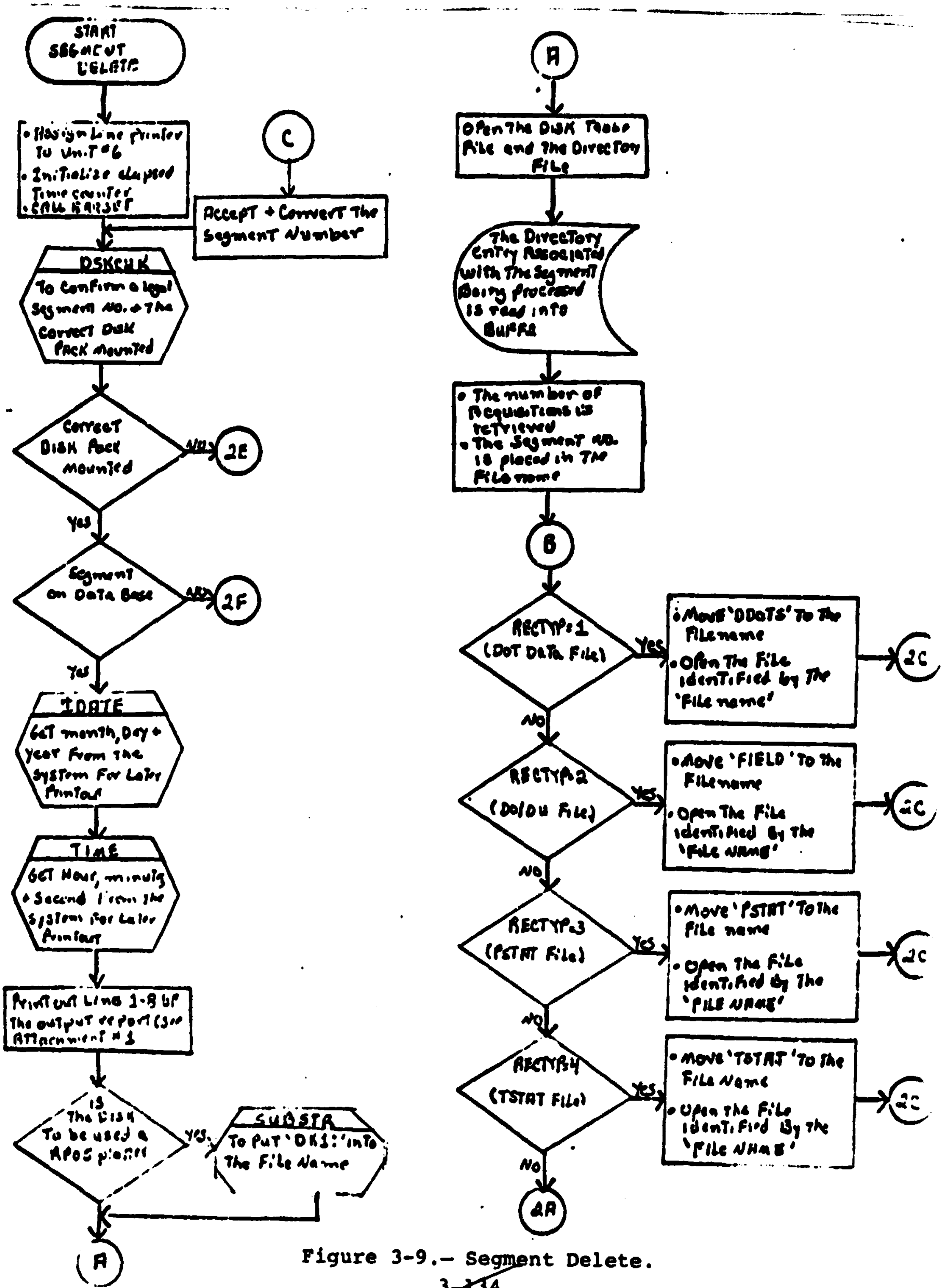
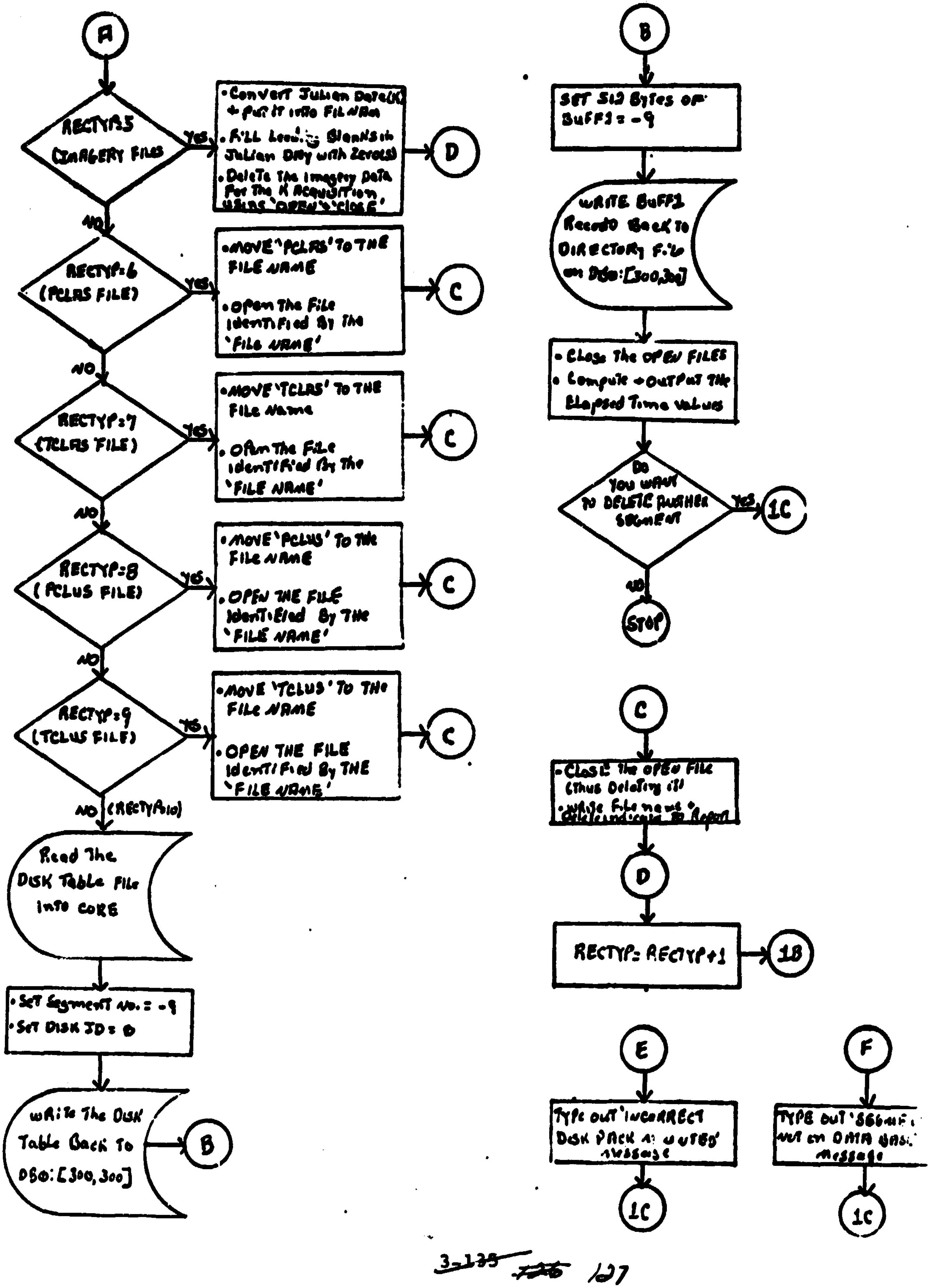


Figure 3-9.– Segment Delete.



A
RECTYP=5 (IMAGERY FILE)
YES
• Convert Julian Date(K) + put it into FILNAM
• Fill Leading Blanks in Julian Day with Zeros
• Delete the Imagery Data For the K Acquisition, using 'OPEN' & 'CLOSE'
D
NO
RECTYP=6 (PCLAS FILE)
YES
• MOVE 'PCLAS' TO THE FILE NAME
• Open The File identified By The 'File NAME'
C
NO
RECTYP=7 (TCLAS FILE)
YES
• MOVE 'TCLAS' To THE File Name
• Open The File identified By The 'FILE NAME'
C
NO
RECTYP=8 (PCLUS FILE)
YES
• MOVE 'PCLUS' To THE File NAME
• OPEN THE FILE Identified By THE 'FILE NAME'
C
NO
RECTYP=9 (TCLUS FILE)
YES
• MOVE 'TCLUS' TO THE FILE NAME
• OPEN THE FILE Identified By THE 'FILE NAME'
C
NO (RECTYP=10)
Read The DISK Table File into CORE
• Set Segment No. = -9
• Set DISK ID = 0
wRite The DISK Table Back To DB0:[300,300]
B
B
SET 512 Bytes of BUFF1 = -9
WRITE BUFF1 Record Back To DIRECTORY File on DB0:[300,300]
• Close The OPEN FILES
• Compute + OUTPUT The Elapsed Time Values
DO YOU WANT TO DELETE FURTHER SEGMENT
YES
1C
NO
STOP
C
• CLOSE The OPEN File (Thus Deleting it)
• Write File name + Deleted indicator To Report
D
RECTYP = RECTYP+1
1B
E
TYPE OUT 'INCORRECT DISK PACK MOUNTED' MESSAGE
1C
F
TYPE OUT 'SEG
NOT ON DATA BAS
Message
1C

3.5.2 INTERACTIVE ANALYSIS

The programs in this section are the programs with which the analyst must interact. He is prompted for inputs by menus and queries appearing on the Tektronix terminal. Brief reports needed for interactive decision making are output to the terminal screen, where hard copies may be made if desired. More lengthy reports are output to the Gould printer or line printer at the user's request.

Communication between these programs is accomplished through a core resident 'common' area and working files. The global common blocks are defined in section 3.4. The working files are defined in section 3.4 also.

The global common area and the working files are initialized from the data base at the time the analyst begins analysis for a specific segment. The interactive analysis procedures carried out by the analyst may change the content of the common area and working files repeatedly before results are satisfactory. The decision to permanently update the data base with the new results must be made by the analyst. He must activate the program which accomplishes this function.

To protect against system crashes and provide a restart capability, the global common area is preserved on a disk file at the completion of each execution for each major program module. In the event of a system crash, the analyst must indicate that he is restarting when he signs on again; then he must reactivate the program module which was in use when the system went down.

Except for a description of the control program (Section 3.5.2.1) the remainder of this section is organized according to the first menu the analyst will see when he has activated the CAMS/I-100 system. (See Figure 3-10)

```
     ***CAMS I-100 CONTROL PROGRAM***

 1 INITIATE SEGMENT ANALYSIS
 2 IMAGE DISPLAY
 3 FIELD DEFINITION
 4 DOT PROCESSING
   *DOT GROUP CROSSHAIR OVERLAYS
   *DOT GROUP SCATTER PLOTS
   *DOT LABELLING
 5 AUTOMATIC CLUSTER LABELLING
 6 CLUSTER DISPLAY/RELABEL
 7 CLASSIFICATION CORRECTION/MAP DISPLAY
 8 REPORTS
   *DOT DATA REPORT
   *BIAS CORRECTION/CLASSIFICATION SUMMARY
   *CLUSTER REPORTS
   *FIELD DATA REPORT
 9 DATA BASE UPDATE
   *PERMANENT DATA BASE UPDATE
   *CARD IMAGE FILE FOR OFFLOAD DOT DATA. DO/DU FIELD
99 CAMS TEST
 X EXIT
```

```
   ***CAMS I-100 DOT PROCESSING***

1 DOT GROUP CROSSHAIR OVERLAYS
2 DOT GROUP SCATTER PLOTS
3 DOT LABELLING
```

```
   ***CAMS I-100 REPORTS***

1 DOT DATA REPORT
2 BIAS CORRECTION/CLASSIFICATION SUMMARY
3 CLUSTER REPORTS
4 FIELD DATA REPORT
```

Figure 3-10.- Control Displays.

3.5.2.1 Storage CAMS/I-100 Control Program (CAMSEX) (Programmed and documented by Ethyl Hightower)

This program controls the program execution in response to the analyst's requests during the interactive imagery analysis. The program is installed as TASK = . . . CAM.

3.5.2.1.1 Linkages

A. IMALIB - routines used FRONT, INTFF, OUTPUT

B. System routines used CLREF, REQUES, WAITFR.

C. Shared subroutines - ELAPSE

D. Executable tasks using system event flag 50 for communication. INIT, FULOI3, FLDDEF, DOTOVR, SCPLOT, DOTPRO, ACLLAP, CLUDIS, RECPRO, DOTRPT, BIASCR, CLURPT, FLDRPT, PRMUPD.

3.5.2.1.2 Interfaces

This program communicates with other programs through the global common area (see Appendix B for details and system event flag 50.

3.5.2.1.3 Inputs

Inputs to this program are options requested by the analyst through keyboard to initiate the desired imagery analysis capability.

3.5.2.1.4 Outputs

DIAGNOSTICS

('$?') - User select a number less than 10 in the CONTROL PROGRAM menu

('INVALID OPTION') - If any number is selected under DOT PROCESSING which is greater than 3 an error message is printed.

('INVALID OPTION') - If any number is selected under REPORT which is greater than 4 an error message is printed.

3.5.2.1.5 <u>Storage Requirements</u>

CAMSEX occupies 28 blocks.

3.5.2.1.6 <u>Description</u>

The CAMS/I-100 control program is loaded into core immediately after the analyst signs on to the system, and remains in core until the end of one interactive imagery analysis session. After being loaded to core, the control program displays all functional capabilities in the form of options and waits for the analyst to input the option number from the keyboard. After the analyst keys in the desired option number, it activates the corresponding program and waits for program completion.

For the purpose of program communication, a system event flag is used as an indicator and is set to '0' by the control program prior to activating the requested program, and to '1' by the activated program when ready to return. As soon as the control program resumes from the wait state, it repeats the above process and waits for further input from the analyst.

3.5.2.1.7 <u>Flowchart</u>

See Figure 3-11.

3.5.2.1.8 <u>Subroutines</u>

None

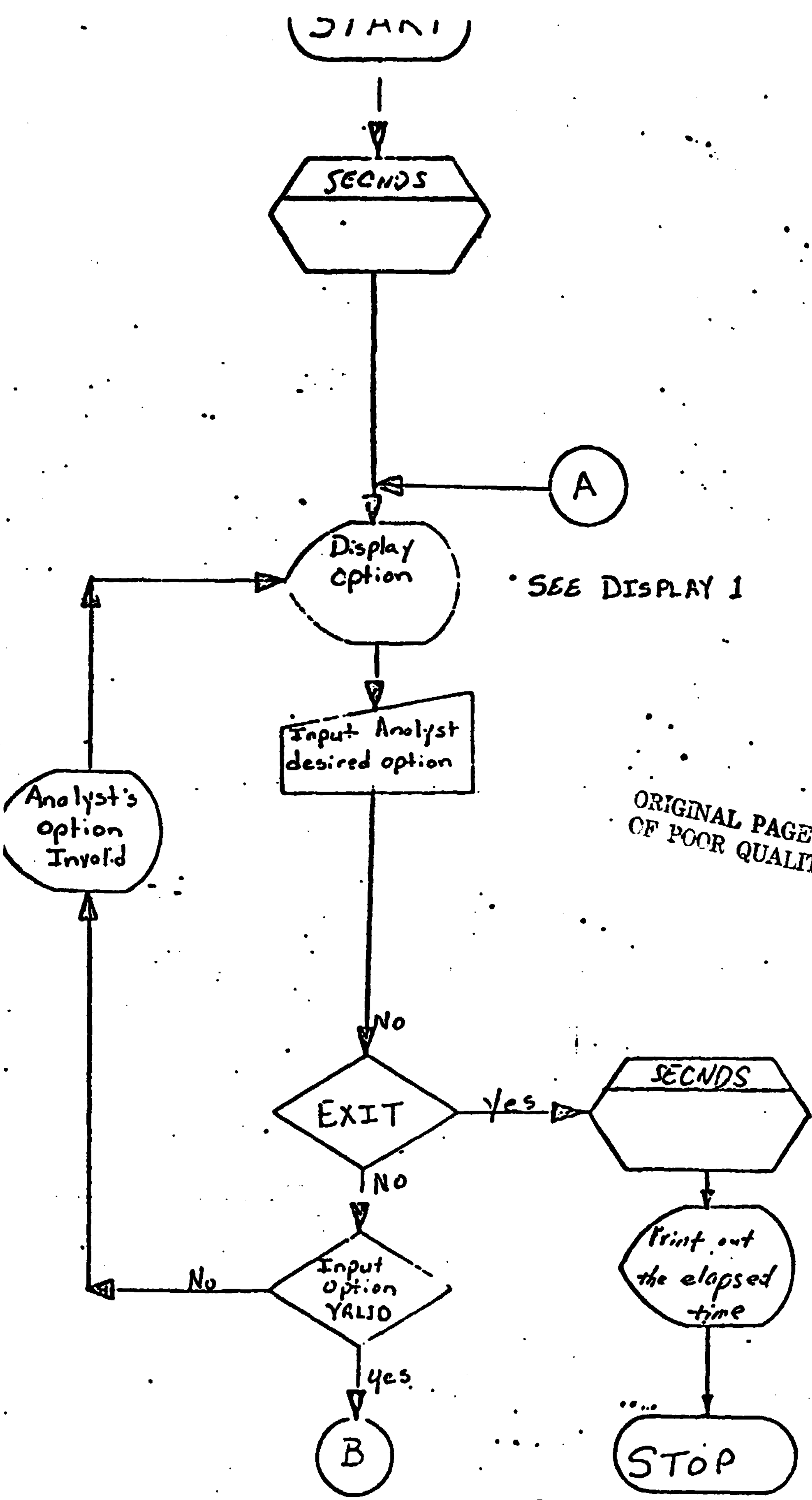


Figure 3-11.- Control Program.

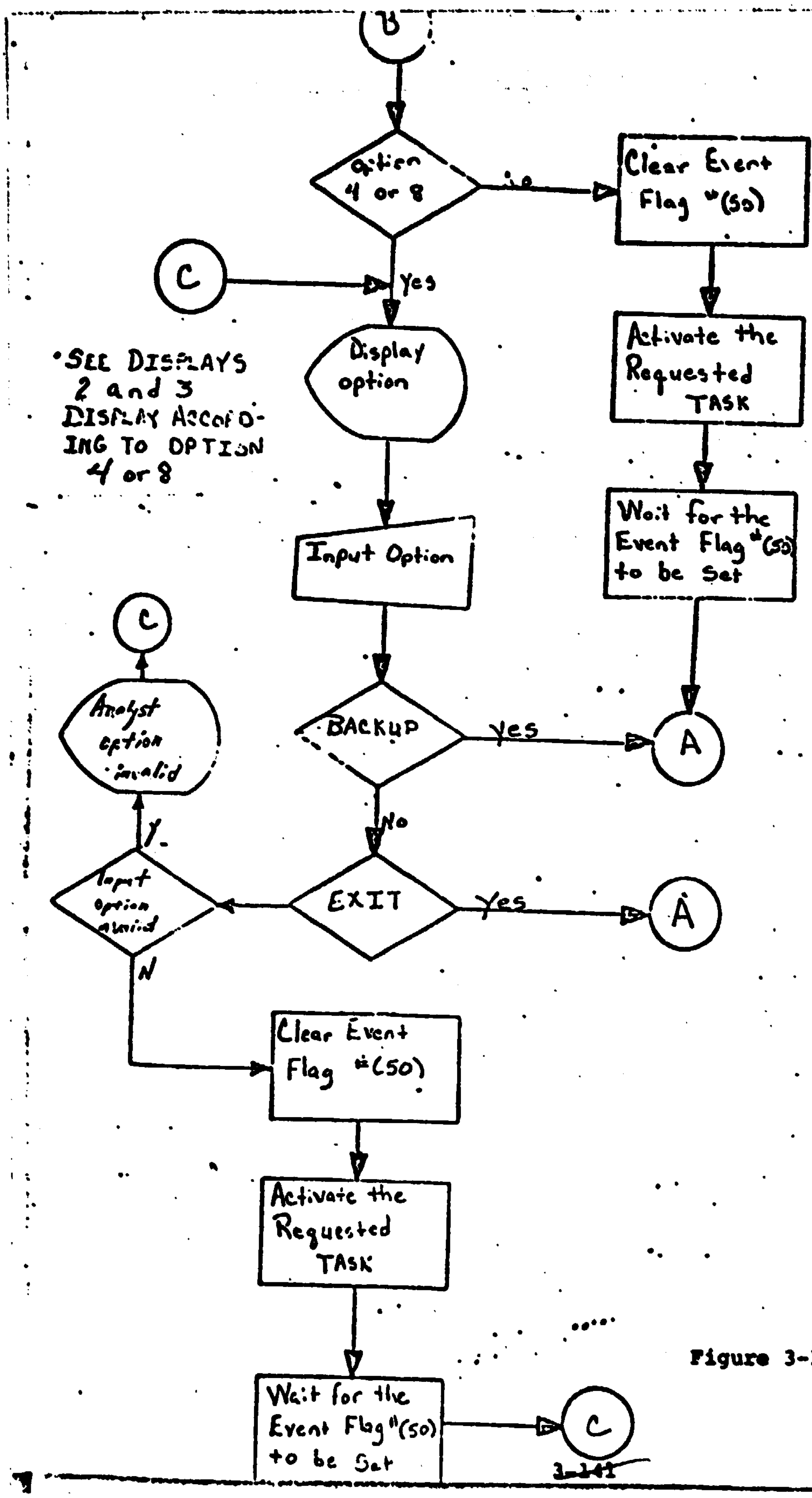


Figure 3-11.- Continued.

3.5.2.2 Initiate Segment Analysis (INIT) (Programmed and documented by Gene Wilson)

3.5.2.2.1 Linkages

A. IMALIB

1. FRONT
2. IBYTE

B. FORTRAN (OR SYSTEM)

1. CLOS$
2. OPEN$
3. SETEF
4. $MAXO
5. $MINO

C. SHARED SUBROUTINES AND OUTLINES

1. CSGDPH
2. DSKCHK
3. ELAPSE

D. PRIVATE SUBROUTINES

1. FTRNFR
2. INTLZE

E. TASK OVERLAY LINKS

NONE

3.5.2.2.2 Interfaces

EXTERNAL

A. FILES

1. 'DBO:[300,1]CLASSMAP.TMP;1'
2. 'DBO:[300,1]CLUSTERMP.TMP;1'
3. 'DB2:[300,300]XXXXTCLAS.MAP'
4. 'DB2:[300,300]XXXXPCLAS,MAP'
5. 'DB2:[300,300]XXXXTCLUS.MAP'
6. 'DB2:[300,300]XXXXPCLUS.MAP'
7. 'DB2:[300,300]XXXXTSTAT.DAT'
8. 'DB2:[300,300]XXXXPSTAT.DAT'
9. 'DB2:[300,300]XXXXFIELD.DAT'

10. 'DB2:[300,300]XXXXDDOTS.DAT'
11. 'DBO:[300,1]DOTS.TMP:1'
12. 'DBO:[300,1]FIELDS.TMP;1'
13. '[300,1]DOTGXY.TMP;1'
14. '[300,1]SCATXY.TMP;1'
15. 'DBO:[300,1]STATFIL.TMP;1'
16. 'DBO:[300,1]CLUSTATS.TMP;1'
17. 'DBO:[300,300]DIRFILE.DAT'
18. 'DBO:[300,1]GLOBAL.TMP;1'

WHERE XXXX = SEGMENT NUMBER

B. COM1, COM2, COM3, COM4, AND CAM5 ARE ALL COMPLETELY OUTPUT.

3.5.2.2.3 Inputs

KEYIN (REFERENCE USER'S MANUAL)

3.5.2.2.4 Outputs

A. Reports - segment report display
B. Diagnostics
   1. Initiate segment analysis mode requested is other than N, R, or X: try again.
      Action: program loops back for another mode try.
   2. XXXX Illegal segment number -- must be four integers - try again (where XXXX is a KEYIN)
      Action: Program loops back for another segment number.
   3. Error encountered in transfer of temporary classification map.
      Action: Exit logic
   4. Permanent classification map does not exist. (Exit logic)
   5. Error encountered in transfer of permanent classification map.
      Action: Program loops back for another segment number.

6. Error encountered in transfer of temporary cluster map. Exit logic.
7. Permanent cluster map does not exist. Exit logic.
8. Error encountered in transfer of permanent cluster map. Exit logic.
9. Error encountered in transfer of temporary statistics file (Record 1). Exit logic.
10. Error encountered in transfer of temporary statistics file. Exit logic.
11. Permanent statistics file does not exist. Exit logic.
12. Error encountered in transfer of permanent statistics file (Record 1). Exit logic.
13. Error encounter in transfer of permenent statistics file. Exit logic.

3.5.2.2.5 Storage

INIT occupies 67 blocks.

3.5.2.2.6 Description - same as before

The initiate segment analysis program allows the analyst to initiate segment analysis under a variety of situations.

After the analyst inputs the desired segment number, a segment summary report is displayed on the terminal to inform the analyst the latest segment working status. This report consists of the acquisition dates (up to 6), previous computation results, existent DO/DU files, dot data file availability, classification and cluster map creation dates, etc.

Having displayed the above report, the program then allows the analyst to select one of the following run modes to initiate segment analysis. Case 1 is the normal segment analysis initiation. It is designed for the analyst to perform normal segment analysis whenever it is due to the arrival of the new acquisition data or the reassurance of the previous work. This mode is the normal mode of operation. If the analyst selects this mode, the program will read in the acquisition date from the analyst and retrieve the desired portion of the data base to build the global common and create the temporary working files.

Case 2 is the continuation from the previous segment analysis. It is designed to allow the analyst to continue the previous unfinished work in an orderly manner. If the analyst selects this mode, the program will read in the acquisition date from the analyst and retrieve the desired data from the temporary results save tape to restore the global common and the temporary working files.

Case 3 is the restart from system crash. It is designed to help the analyst to recover from accidental system crash. If the analyst selects this mode, the program will read in the saved global common area from disk to restore the global common. Note that the global common is always saved to disk by all interactive imagery analysis programs during any imagery analysis session. However, there is no restore on the temporary working files (the cluster and classification maps). It is the analyst's responsibility to ensure their validity.

After the program completes the segment initialization, it exists.

3.5.2.2.7 Functional Flow Chart

See Frame 3-12.

3.5.2.2.8 <u>Subroutines</u>

A. Subroutine FTRNFR

1. This subroutine transfers (loads) a file.
2. Call FTRNFR (FILEIN, FILEOT, NR, KRW, MR, FLAG)

   WHERE: FILEIN is input file name
   FILEOT is output file name
   NR is number of records in the file
   KRW = length of each record
   MR - maximum number of records

B. Subroutine INTLZE

1. This subroutine initializes COM1 - COM5 to zero or other values.
2. Call INTLZE

ORIGINAL PAGE IS OF POOR QUALITY

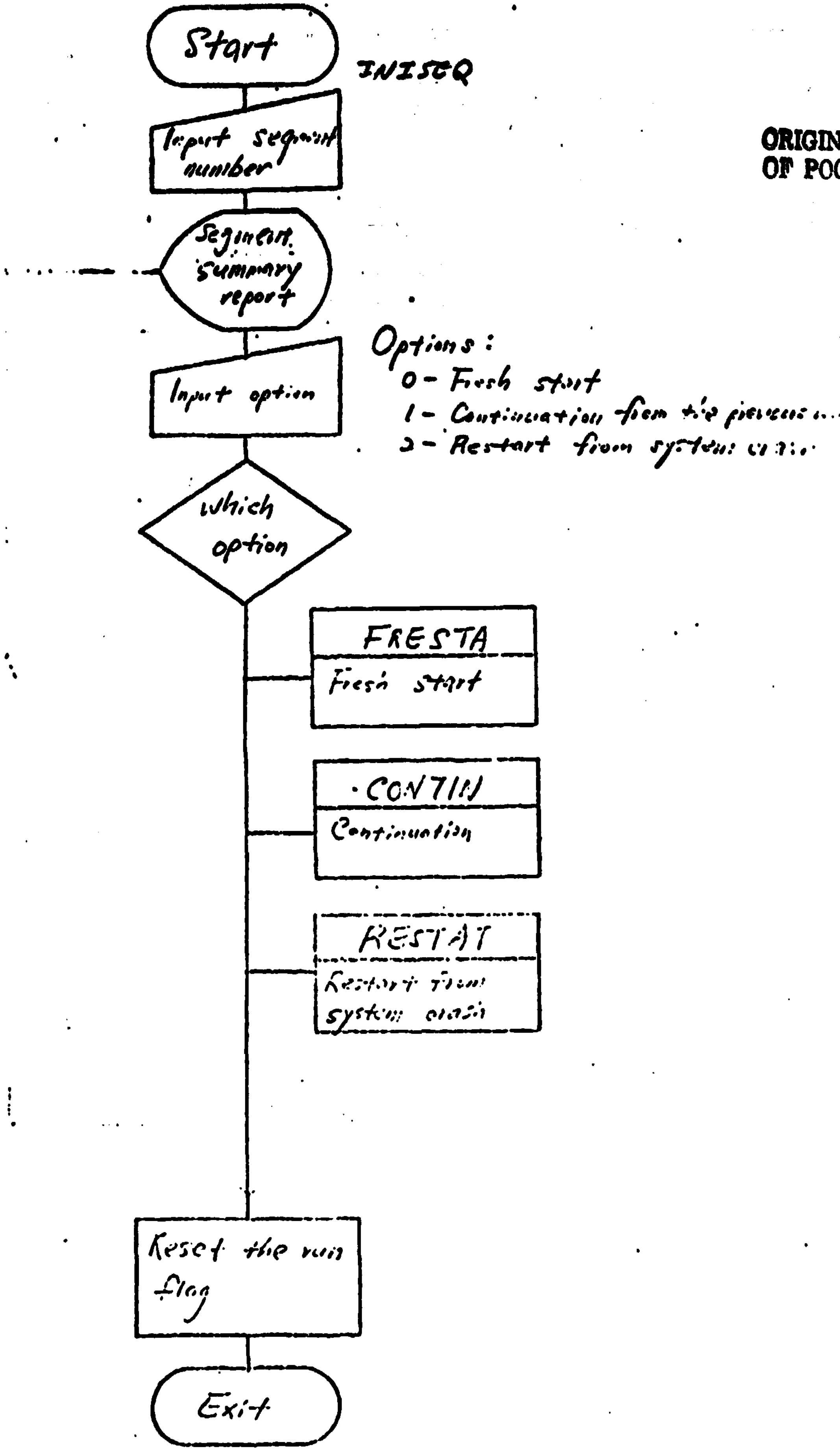


Figure 3-12.— Initiate Segment Analysis.

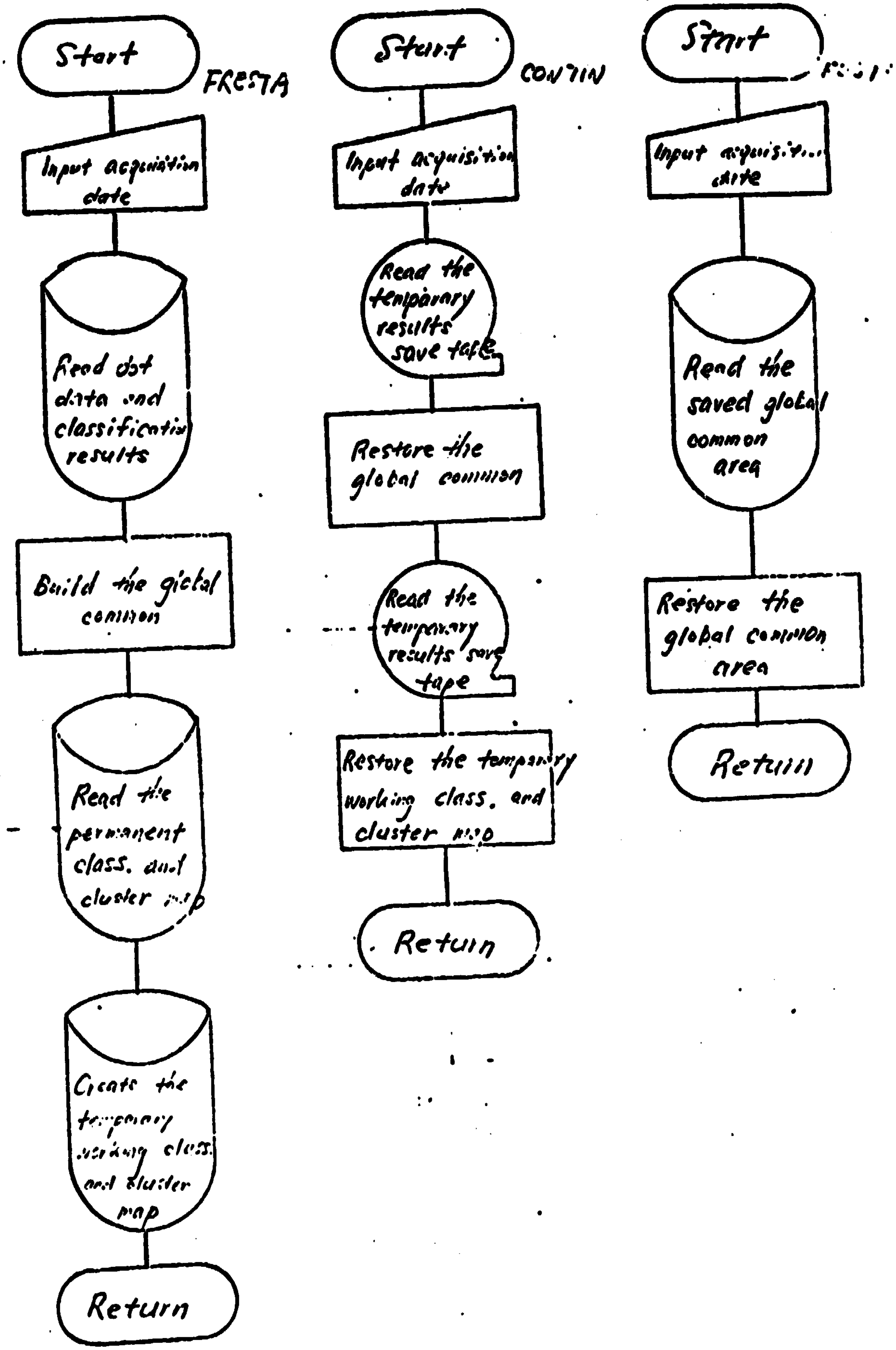


Figure 3-12.— Continued.

3.5.2.3 Image display (FULOI3) (Programmed and documented by T. Kell)

This program will read an image data file, apply user supplied transformations and write the data to the Image-100 refresh memory for CRT display.

3.5.2.3.1 Linkages

A. IMALIB

B. FORTRAN: F4POTS

C. SHARED SUBROUTINES: CSGDPH, DSLT, DUSET, FFFPI, HREAD, LIN, TCHLST

D. PRIVATE SUBROUTINES: COMLUT, GABI, ZOOOOM

E. TASK OVERLAY LINKS: FULOI4

3.5.2.3.2 Interfaces

Common name FCOM

Parameter	Updated by Subroutine	Referenced by Subroutine
FILE	*	DSET
WC	TCHLST	---
MGL	*	---
MGF	*	---
LGL	*	---
LGF	*	---
HFL	*	---
HFG	*	---
CHL	TCHLST	---
CHP	TCHLST	---
MXC	---	---
NF	*	---
F11	*	DSET, DUSET, HRBAD
MTXTFG	*	DSET
TUNIT	*	DSBT
TUN	---	---

Parameter	Updated by Subroutine	Referenced by Subroutine
FLN	*	---
FORM	*	DSET, HRBAD
TTXI	*	ZOOOOM
TTY1	*	
TTX2	*	
TTY2	*	
IIX1	*	
IIY1	*	
IIX2	*	
IIY2	*	ZOOOOM
IX	ZOOOOM	---
IY		---
TX		---
TY		---
MX		---
MY		---
NX		---
NCMAX	*	---
LGBC	*	COMLUT
LUN	---	---
GB	---	---
LUT	COMLUT	---
NCR	---	---
RFAC	*	---
X2	ZOOOOM	---
YZ	ZOOOOM	---

Note: 1) * means main

2) - means none

COMMON NAME HCOM

Parameter	Updated by Subroutine	Referenced by Subroutine
SS	HPROS	*
SE		*
LS		*
LE		HREAD, LREED, FFIND
NRPDS		
NDSPR		
NCPR		
ANCL		
NC		
NS		
NBIT		
DOI		
NCAR		
SUD		
RSIZ		
RSKIP		
HSIZ		
CALP		
CBRR	DSET, HREAD	*

COMMON NAME COM2

ISEG	?	*
ADATES	?	*

COMMON NAME COM4

TX1	*	?
TY1		
TX2		
TY2		
IX1		
IY1		
IX2		

Parameter	Updated by Subroutine	Referenced by Subroutine
IY2	*	?
ACDISP	*	?
II1	*	?
G	*	?
B	*	?

FILES

FULOI.DAT

DOTGXY.TMP

3.5.2.3.3 Inputs

A. NA

B. NA

C. KEY-IN (SEE USER'S MANUAL)

D. NA

The image data to be displayed for LACIE/CAMS will be a file resident in the data base and retrieved by user specified segment number and acquisition date. This program will also accept image data on tape for the non-LACIE production user in one of the three formats: UNIVERSAL, ERTS, or LARSYS.

Default conditions for tape and image coordinates, gains and biases and image file name will be maintained for LACIE but each default may be overridden by keyboard input.

3.5.2.3.4 Outputs

A. Reports
   Summary - See User's Manual
B. Diagnostics
   1. "CATASTROPHIC ERROR NUMBER"
      Actions available - Restart or exit.

2. ZOOOOM ERROR - BAD IMAGERY/CRT POSITIONS
   Actions available - reenter scene corners.
3. "CHANNEL NUMBER MUST BE POSITIVE."
   Actions available - respecify gain/biases
4. "CHANNEL # NOT IN IMAGERY"
   Actions available - ditto.
5. "OPEN ERROR ON FILE FULOI.DAT"
   Actions available - restart or exit.

C. Other

Image on CRT

In addition to the CRT display of the image, this program will output to a global communication area all parameters required by other programs to locate the image on the display and to apply transformations.

3.5.2.3.5 Storage Requirements

FUL013 occupies 97 blocks and FUL014 occupies 92 blocks.

3.5.2.3.6 Description

Program requests desired acquisition date, if LACIE defaults and if not, the Imagery/CRT positions, and gain bias locations/values. A summary is then given and if proceed is yes, FULOI4 is "REQUES"ted. When EF53 is set program asks restart of exit and responds appropriately.

The image display program has the following capabilities:

A. Input image data may be on tape or disk (i.e., from the data base).

B. The image may be in either UNIVERSAL, LARSYS or ERTS format.

C. The tape and display coordinates (i.e., pixel and line numbers) are input to the program, allowing the scene to be displayed anywhere on the screen, and to be blown-up or reduced in size according to the analyst's desires.

D. The following transformations may be applied to the data as it is transferred from tape or disk file to refresh memory for image display.

$z = G(y-b)$

where: z = the vector written to refresh memory (maximum dimension 5)

G = gains supplied by the analyst or read from the header of a UNIVERSAL formatted image

b = bias values supplied by the analyst or read from the header of a UNIVERSAL formatted image

$y = Bx+c$

where: B = analyst supplied transformation matrix of maximum dimension 5 x 60

x = raw data vector of maximum dimension 60

c = analyst supplied bias vector of maximum dimension 5.

If B and C are not input then $y = x$ and x is limited to a dimension of 5.

It is up to the analyst when using these transformations to ensure that the transformation itself scales the data to a 0-255 range. The $y = Bx+c$ transformation will be done in floating point. However, since y must be an 8-bit integer number, fractions will be truncated, negative numbers will be set to 0, and numbers greater than 255 will be set to 255.

E. There are five display channels on the Image-100. The fifth is normally reserved for theme track displays but it may also be used for video display. The analyst may specify which channel from his image file is to be viewed on each channel of the display.

Default conditions will be provided for each of these input parameters.

3.5.2.3.7 Flowcharts

This flowchart is presented in Figure 3.13.

3.5.2.3.8 Subroutines

3.5.2.3.8.1 COMLUT

Compute gain bias lookup tables

- Calling sequence
  CALL COMLUT
- Arguments
  None
- Description
  Computes
  LUT(J,I) = (LGBC(I,2)/100)*(J + LGBC(I,3)/100.) for I = 1 to 5
  J = 0 to 255
  Within limits of 0 to 255 on LUT

3.5.2.3.8.2 GABI

Accesses gains and biases from leader and converts from PCF Gain/biases to FULOI gain/biases.

- Calling sequence
  CALL GABI
- Arguments
  None

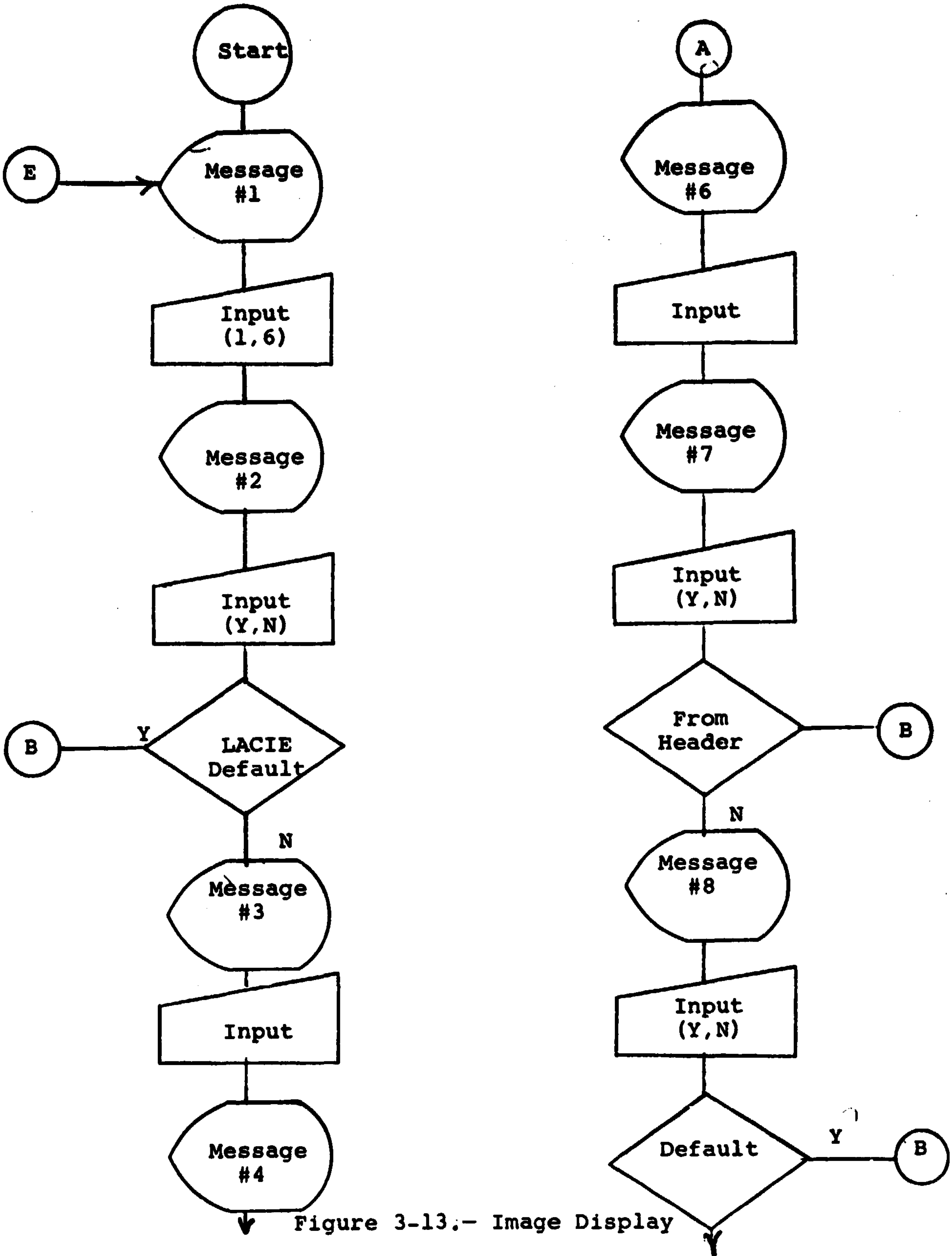
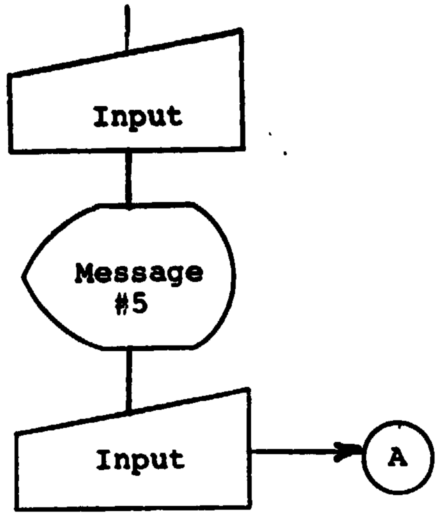


Figure 3-13.– Image Display



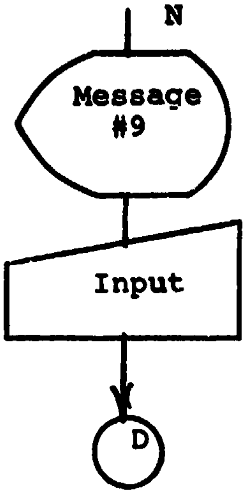
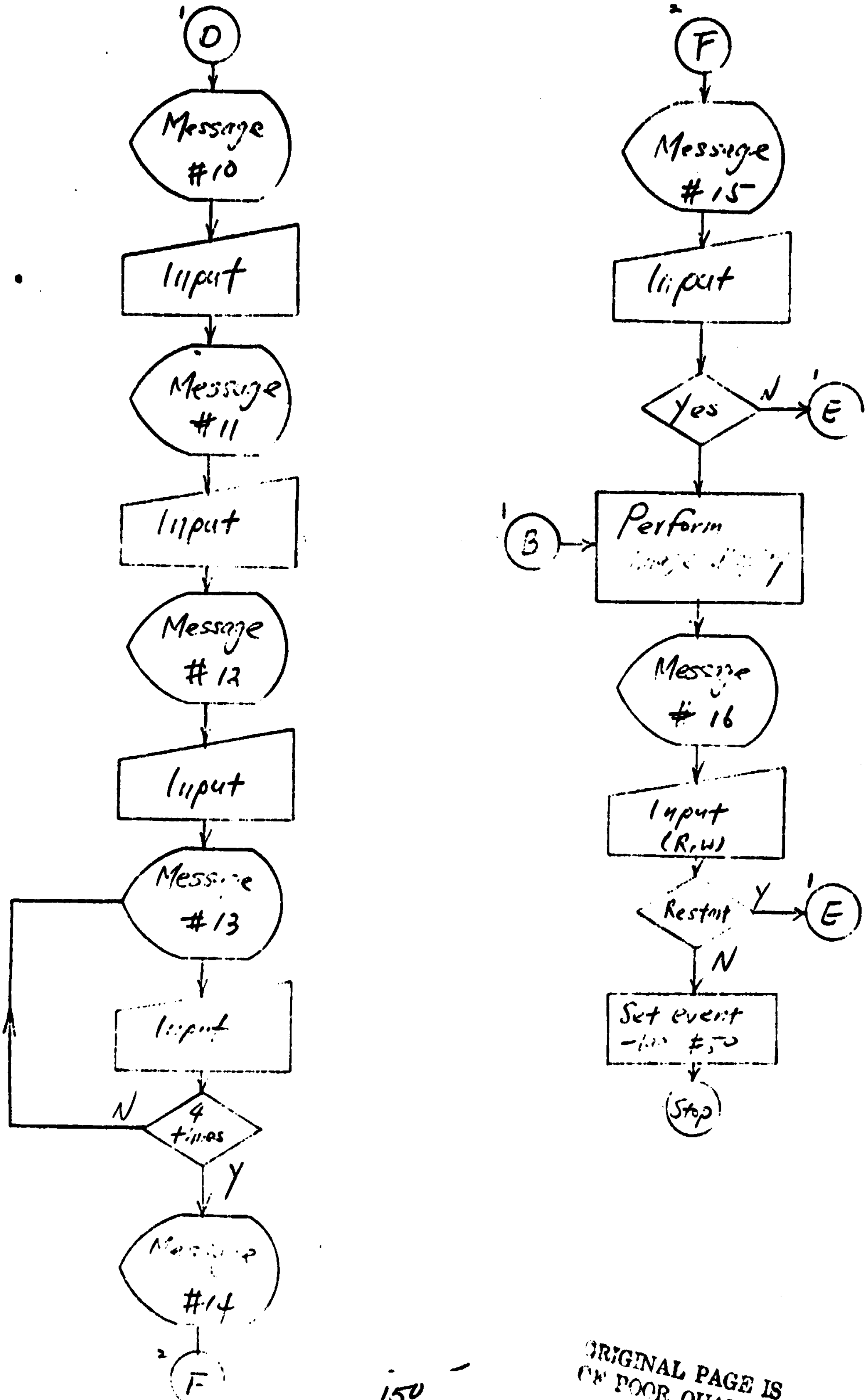


Figure 3-13.- (Continued)


D
Message #10
Input
Message #11
Input
Message #12
Input
Message #13
Input
N
4 times
Y
#14
F
F
Message #15
Input
Yes
N
E
B
Perform
Message #16
Input (R,W)
Restart
Y
E
N
Set event
Stop

- Description
  1. Gets PCF gains/biases from leader
     These are Y = A*X + B
  2. Convert to FULOI gain/biases
     These are Y = A*(X+B)

3.5.2.3.8.3 ZOOOOM

Converts scene/CRT corners to a set of "registration" look up tables.

- Calling sequence
  CALL ZOOOOM (IX1,IY1,IX2,IY2,TX1,TY1,TX2,TY2,IX,IY,TX,TY, XZ,YZ,MX,MY,NX)

Parameter	Dimension	Type	In/Out	Description
IX1	1	I	I	Scene/CRT
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
TY2	1	I	I	Corners
IX	512	I	O	Pixel/Line
IY	512	I	O	Scene/CRT
TX	512	I	O	Conversion
TY	512	I	O	Tables
XZ	1	R	O	X Zoom factor
YZ	1	R	O	Y Zoom factor
MX	1	I	O	CRT width
MY	1	I	O	CRT length
NX	1	I	O	Scene width

This program generates four tables IX, IY, TX, TY such that i varies from 1 to MX. IX_i is the pixel on the CRT where imagery pixel TX_i is to be placed. The same applies in the Y direction.

3.5.2.3.9 FULOI4

This task takes the parameters generated by FULOI3 and places an imagery upon the CRT.

3.5.2.3.9.1 Linkages

A. IMALIB

B. F4POTS

C. DSET, DUSET, FFIND, HREAD, LREED

D. GBCALC

E. None

3.5.2.3.9.2 Interfaces

See 3.5.2.3.2

3.5.2.3.9.3 Inputs

A. Tapes - none

B. Cards - none

C. Key-in - none

D. Other - A. FULOI.DAT

B. Imagery file

3.5.2.3.9.4 Outputs

A. Reports - none

B. Diagnostics - none

C. Other - CRT display

3.5.2.3.9.5 Storage Requirements

3.5.2.3.9.6 Description

For i from 1 to MY, the desired output channels are read for line

TY_i, the desired input channels are read for line TY_i. Then for j from 1 to MX, input pixels as specified in TX_j are moved to output pixels IX_j after being properly scaled. Then the output is written and i is incremented.

3.5.2.3.9.7 Flowchart

Presented in Figure 3.13A.

3.5.2.3.9.8 Subroutines

3.5.2.3.9.8.1 GBCALC

This routine moves the imagery data and scales it for both the vector mode and the matrix mode (not used).

- Calling sequence
  CALL GBCALC (TP, IP, B, V, NX, NC)

Parameter	Dimension	Type	In/Out	Description
TP	1	I	I	TX_i
IP	1	I	I	IX_i
B	(NX,NC)	B	I	Data read from imagery file
V	(512,5)	B	IO	Data read from/to be written to CRT
NX	1	I	I	See 3.5.2.3.8.3
NC	1	I	I	Number of input imagery channels

Pixel data B (TP,LGBC(I,1)) is used as an index to the tables of 3.5.2.3.8.1 and is stored in V(IP,I) for I from 1 to 5 where LGBC(I,4) is not zero or less than zero.

3.5.2.4 Field Definition (FLDDEF) (Programmed and Documented by T. Kell)

This task prints out the field file, displays fields from the field file, adds fields to the field file, and removes fields from the field file.

3.5.2.4.1 Linkages

A. IMALIB

B. FORTRAN: F4POTS

C. SHARED: CSGDPH,LIN

D. PRIVATE: DELFLD,FLDRPT,FLGDOT

E. OVERLAY: IRREG3

3.5.2.4.2 Interfaces

Common name COM3

Parameter	Updated by	Referenced by
EFLAG3	*	DELFLD,FLDRPT
UFLAG2	*,DELFLD	---

Common name COM4

Parameter	Updated by	Referenced by
DOLABEL	*	---

Working files

IRRDEF.TMP

FIELDS.TMP

3.5.2.4.3 Inputs

A. Tapes name

B. Cards name

C. Key-in - See User's Manual

D. Other - IRRDEF.TMP - data returned from IRRBG3

For field deletion, the analyst inputs the field name through the keyboard.

For field display, the analyst inputs the field name or the class label through the keyboard.

For defining new fields, the analyst enters the field vertices (limited to 10 vertices) either through the keyboard or the cursor.

3.5.2.4.4 Outputs

A. Reports - See User's Manual

B. Diagnostics
   1. FIELD DEFINITION FILE EMPTY - Asks restart or exit
   2. EXISTANCE FLAG FOR FILE "FIELDS.TMP" NOTSET CLEARLY NO FIELDS CAN BE DELETED. . . exits
   3. USER TYPED "X", GOOD BYE NO FIELDS DELETED - exits
   4. NO FIELDS TO DELETE - exits

C. Other IRRDEF.TMP,FIELDS.TMP

For field deletion, the requested field data is deleted from a temporary file as read from the data base at segment initialization. The dot DO/DU flags are updated similarly. For defining new fields, the temporary files are updated to reflect the inputed field data and the dot DO/DU flags.

3.5.2.4.5 Storage Requirements

FLDDEF occupies 58 blocks. IRREG3 occupies 80 blocks.

3.5.2.4.6 Description

Program asks if to report, display, delete, or add files.

Report - program asks output device, prints report, asks restart or exit.

Display - program asks for a selection of field(s) to display as well as operational parameters, displays it (them) and asks if more files to display.

Delete - program asks for a selection of field(s) to delete, asks yes/no for each field before deleting it.

Add - program asks for operational parameters then requests up to 10 corners via

1. Cursor
2. TI
3. File

3.5.2.4.7 Flowcharts

Presented in Figure 3-14.

3.5.2.4.8 Subroutines

3.5.2.4.8.1 DELFLD

Delete fields from "FIELDS.TMP"

- Calling sequence
  CALL DELFLD(IO)
- Arguments

3.5.2.4.8.2 FLDRPT

Print field report

- Calling sequence

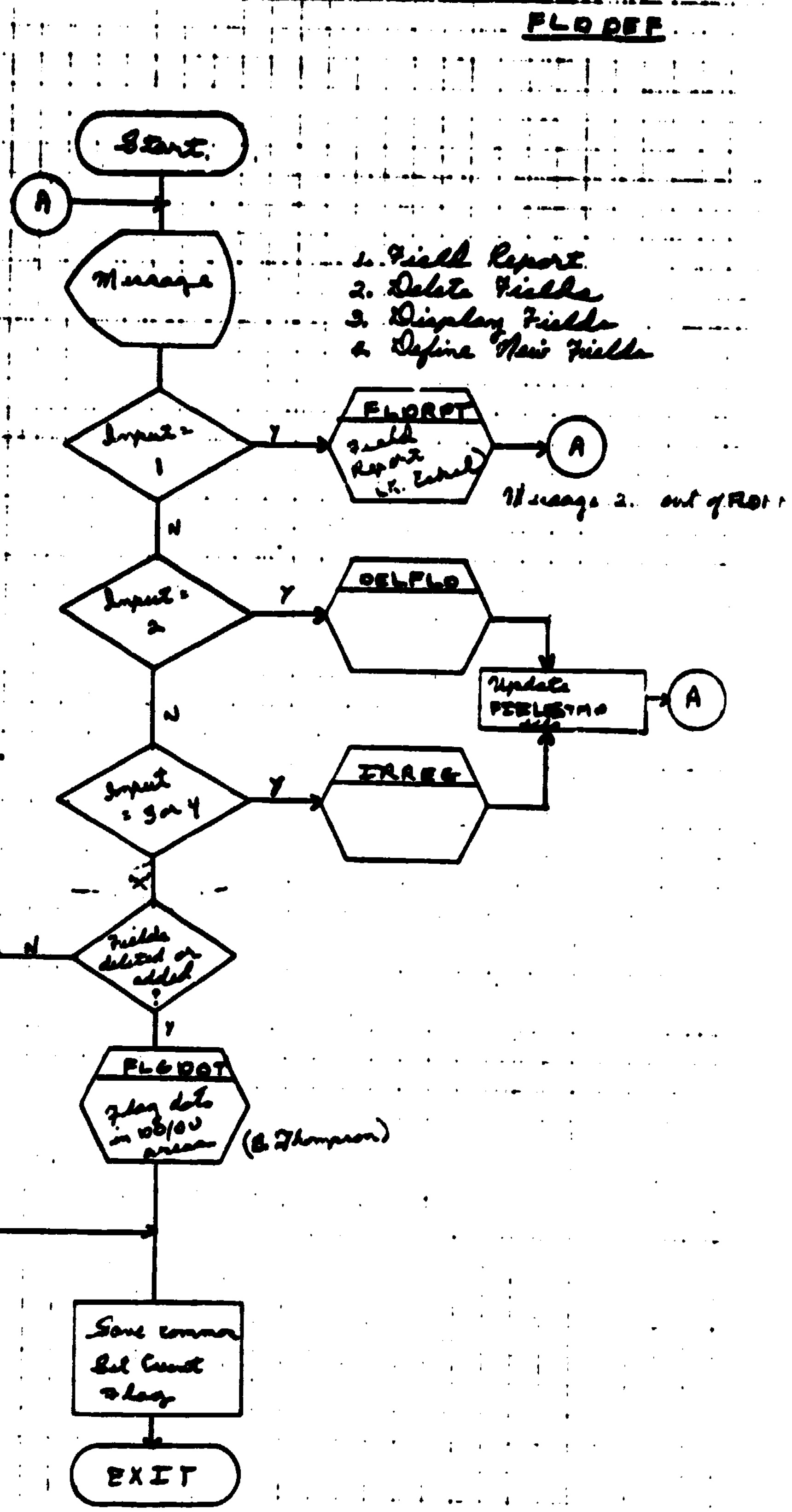
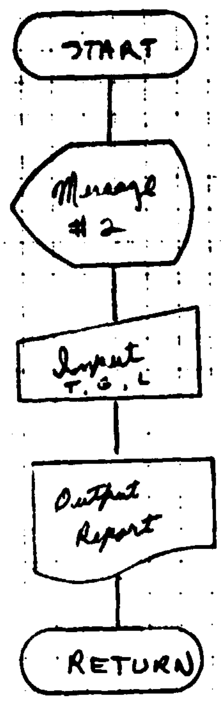


Fig. ;-14

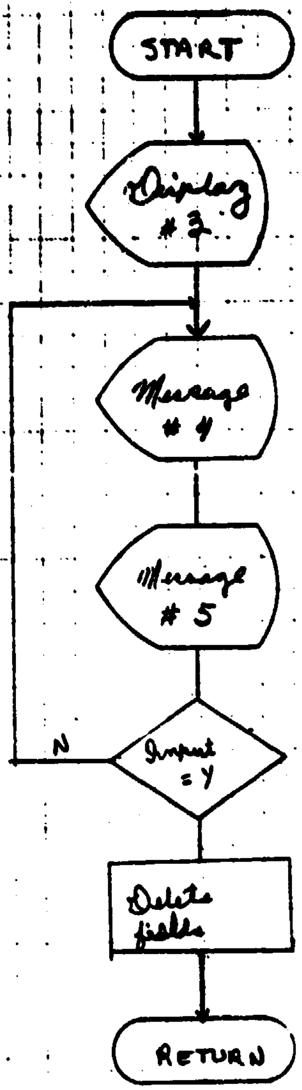
ORIGINAL PAGE IS OF POOR QUALITY

FLOAPT

K. Eckel

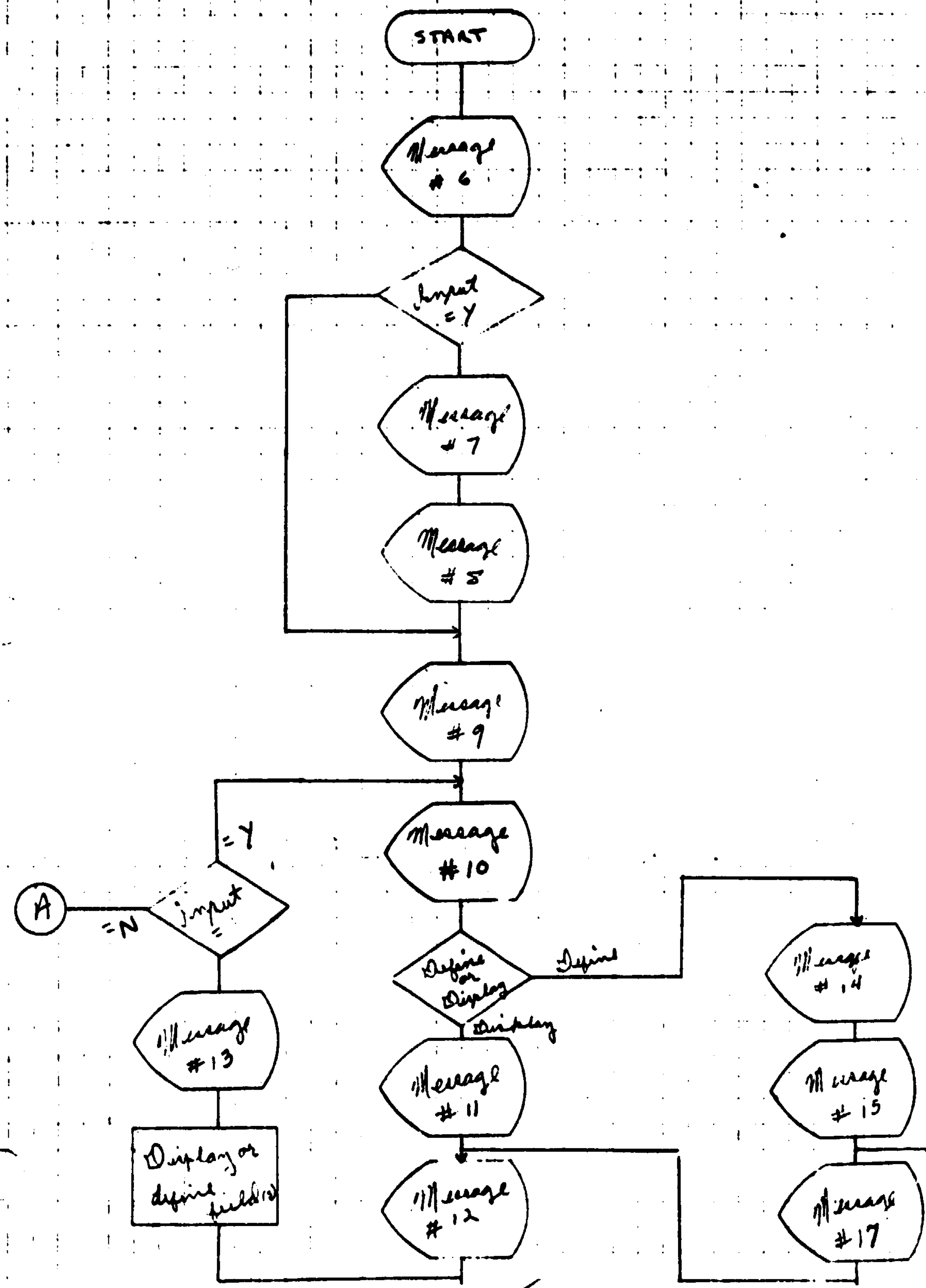


DELFLD



ORIGINAL PAGE IS
OF POOR QUALITY

IRREG


START
Message #6
Input =Y
Message #7
Message #8
Message #9
=Y
Message #10
A
=N
input =
Define or Display
Define
Display
Message #14
Message #13
Message #11
Message #15
Display or define field(s)
Message #12
Message #17

Messages		Default
1.	(1) Field Report (2) Delete Fields (3) Display Fields (4) Define New Fields (x) EXIT	N/A
2.	REPORT DEVICE (T)ERMINAL, (G)OULD OR (L)INE PRINTER >	T
3.	CURRENT FIELDS ARE (1) NAME1 (2) NAME2 (3) NAME3 (4) NAME4 (5) NAME5 (6) NAME5	
4.	TYPE NUMERS OF FIELDS TO BE DELETED >	N/A
5.	DELETE FIELDS (N)NAMEN (M)NAMEM - - - CONTINUE (Y)ES/(N)O >	 Y
6.	USE LATEST IMAGE DISP,AY COORDINATES (Y)ES/(N)O >	Y
7.	COORDINATES FOR DATA BASE IMAGE ---->	TX1,TY1 TX2,TY2
8.	COORDINATES FOR DISPLAY IMAGE ---->	IX1,IY1 IX2,IY2
9.	(O)UTLINE FIELD OR (S)HADE AREA 0>	0
10.	CURRENT FIELDS ARE: NO. CATEGORY NAME (1) DO NAME1 (2) ____________ (3) ________	
11.	INPUT CATEOGRY NAME OR FIELD NUMBERS TO BE DISPLAYED > (DO and DU only acceptable responses)	
12.	DISPLAY ON THEME NUMBER -__? >	
13.	MORE FIELDS? (Y)ES/(N)O >	
14.	INPUT SIX CHARACTER FIELD NAME FOR NEW FIELD >	

Messages	Default
15. INPUT CATEGORY OF NEW FIELD (DO) OR (DU) >	
16. FIELD NAME ________ CATEGORY _______ CONTINUE? (Y)ES/(N)O >	
17. DEFINE FIELD VERTICES BY (C)URSOR OR (K)EYBOARD? >	

CALL FLDRPT(IO)

- Arguments

Parameter	Dimension	Type	In/Out	Description
IO	1	I	I	TI LUN

- Description

See 3.5.2.4.6

3.5.2.4.8.3 FLGDOT

Flag dots as do order.

- Calling sequence

CALL FLGDOT (NFLD,NV,FIELD,DODU,DLABEL)

- Arguments

Parameter	Dimension	Type	In/Out	Description
NFLD	1	I	I	# of fields
NV	NFLD	I	I	# of vertices in each field
FIELD	(2*MAXV,NFLD)	I	I	X,Y positions of each vertex
DODU	NFLD	I	I	indicate of dodu status of each field
DLABEL	NDOTS	I	O	indicate of dodu status of each dot

3.5.2.4.9 IRREG3

Display portions of, or add to file FIELDSTMP.

3.5.2.4.9.1 Linkage

A. IMALIB

B. F4POTS

C. Standard IRREG routine package

D. FLDNAM

E. None

3.5.2.4.9.2 Interfaces

Common name CURCOM

See no new usage beyond IRREG.

Working files

IRRDEF.TMP,FIELDS.TMP

3.5.2.4.9.3 Inputs

A. TAPES - none

B. CARDS - none

C. Key-in - See User's Manual

D. Other - IRRDEF.TMP

3.5.2.4.9.4 Outputs

A. Reports - none

B. Diagnostics -

   1. NO FIELDS SELECTED. TRY AGAIN. . .
      request field selection again

   2. TOO MANY FIELDS. DELETE ONE FIRST.
      Exits

   3. Standard IRREG diagnostics

C. Other IRRDEF.TMP, FIELDS.TMP

3.5.2.4.9.5 Storage requirements

3.5.2.4.9.6 Description

Beyond the standard IRREG description, if you are in display mode, the program asks for your selection of fields to display. The vertices of these fields are then fed to IRREG to process.

If in definition mode, the program asks for the field name/type, and input methods, the input field(s) are placed in FIELDS.TMP upon exit.

3.5.2.4.9.7 Flowcharts

See LEC-6063, IRREG program document.

3.5.2.4.9.8 Subroutines

3.5.2.4.9.8.1 FLDNAM

Controls field selection

- Calling sequence

  CALL FLDNAM (MFLDS,FPTR,NFL,CURDEF,DDIESF,IFST,FIELD,IO,
  NFLDS,BLOCK)

Parameter	Dimension	Type	IN/OUT	Description
MFLDS	1	I	O	1 = more fields 0 = no more fields
FPTR	1	I	O	Index to FIELD (current field)
NFL	1	I	IO	Number of fields left to display input 0 to initialize
CURDEF	1	I	O	vertex input method 0 = cursor 1 = TI
DDIESF	1	I	IO	Display/define flag 0 = define 1 = display
IFST	1	I	I	0 = initialization (don't change)

Parameter	Dimension	Type	In/Out	Description
FIELD	10+2*MAXU, MAXFLD	B	O	FIELDS.TMP in memory
IO	1	I	I	TI LUN
BLOCK	15	I	I	IRRDEF.TMP in memory

- Description

  See 3.5.2.4.9.6

3.5.2.4.9.8.2 Other routines

See 3.5.2.4.9.7

3.5.2.5 Dot Processing

The programs in this section allow the analyst to view dots over the image or in scatter plots. He may identify individual dots by cursor, or he may enter dot numbers at the console. He may relabel dots at will. To aid the analyst, individual dots can be alarmed and reports can be generated. If he wishes, he may see trajectory plots; he may also examine magnifications of individual dots and their surroundings.

3.5.2.5.1 Dot Group Crosshair Overlay (DOTOVR) (Programmed and Documented by R. Rodriguez and S. Thadani)

This program allows the user to select a group of dots and to overlay them as crosshairs on the video image.

3.5.2.5.1.1 Linkages

*ATTACH BLKTHM *DATE *DETACH DOTIN *FRONT *INTFF *IRT *IWT *OUTPUT *SECNDS *SETBIT *SETEF *TIME *WAIT

*-Image 100 (IMALIB) LIBRARY system routines

3.5.2.5.1.2 Interfaces

A. COM4

Parameter	Update by Subroutine	Referenced by Subroutine
TX1,TY1,TX2,TY2		DOTOVR
IMWIND		DOTOVR
CLUWND		DOTOVR
CLAWND		DOTOVR
NUMDOT	DOTIN	DOTOVR
DOTARY	DOTIN	DOTOVR

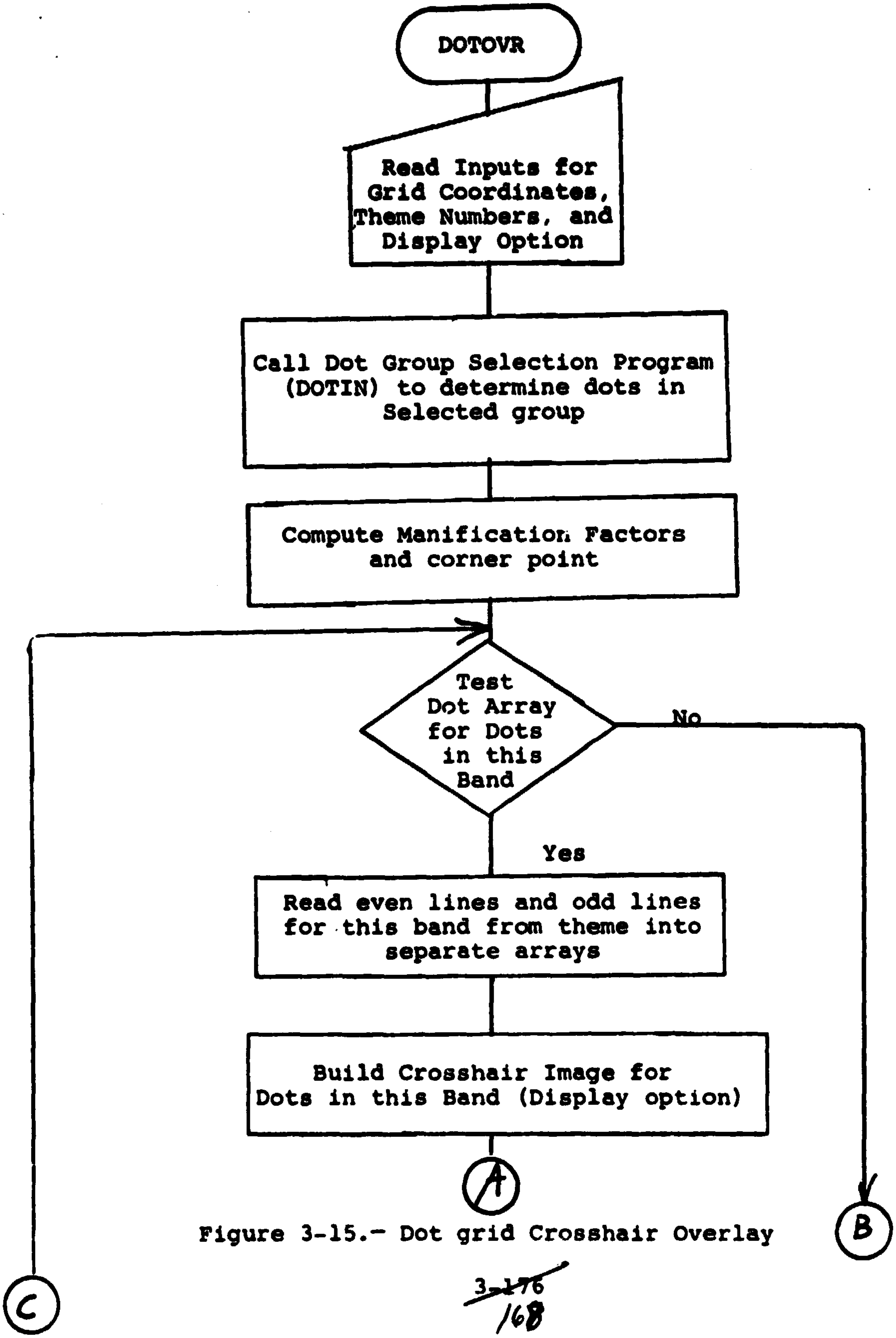


Figure 3-15.- Dot grid Crosshair Overlay

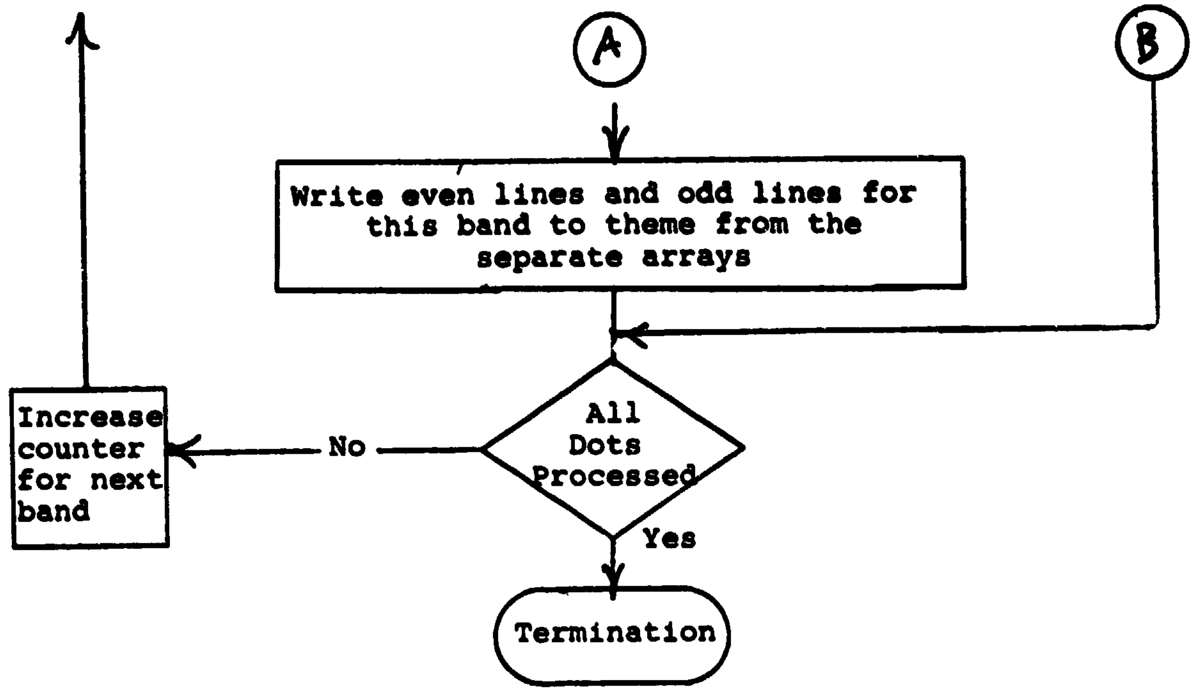


Figure 3-15.- (Continued)

3.5.2.5.1.3 Inputs

The user inputs, to program queries via the DISPLAY terminal, identify the particular subset of dots to be displayed, the video image coordinates, the type of crosshair display, and the theme on which the dots are to be written. (See JSC-IMAGE-100 USERS MANUAL, Part III CAMS Section 6)

3.5.2.5.1.4 Outputs

A. The output is a selected subset of dots written on a theme on the video display.

B. Diagnostics

   **** INVALID INPUT **** (Input must be transmitted again)

   NO DOTS FOUND (program goes to termination)

   Dot crosshair overlay completed (End of Run)

3.5.2.5.1.5 Storage Requirements

DOTOVR occupies 66 blocks.

3.5.2.6.1.6 Description

DOTOVR will set constants and initiate times, clear the display terminal screen, and query user for grid coordinates. The user may input a set of coordinates or select from common the standard, cluster, or classification image coordinates. The program checks to see if the Tape Coordinates are missing and sets them to a standard if they are. The magnification factors are computer based on the cooruinates. The cross hair mark size is set. The user next select the theme number (default = 8). Subroutine DOTIN is called for the dot subset. It returns the number of data and a Dot array. The user now selects a display option (1 = write over, 2 = add-on do not erase dot, 3 = add on full cross) (default = 2). The magnification factors and the corner point are computed.

The Dot array is tested to determine if any data on a particular band are to be processed. If there are some, the even lines and the odd lines in the band are read from the theme into separate arrays. The arrays are modified to reflect the display option and the dot to be displayed inside the specified grid. The even lines and the odd lines for the band are written on the theme. The desired dot are displayed by crosshairs on the theme. Repeat for all 11 bands.

3.5.2.5.1.7 Flowchart

Presented in Figure 3-15.

3.5.2.5.1.8 Subroutine

3.5.2.5.1.8.1 BLKTHM

This program puts labels, blanks, or a pattern in a particular area of a video theme.

- Calling sequence
  CALL BLKTHM (ML,MU,MR,MB,NT,IBUF,IOP)

- Arguments

Parameter	Dimension	Type	In/Out	Description
ML	1	Integer	In	left coord. of block
MU	1	"	In	upper coord. of block
MR	1	"	In	right coord. of block
MB	1	"	In	button coord. of block
NT	1	"	In	theme number
IBUF	(To be determined by calling program set to 1.	"	In	Data array (buffer)

Parameter	Dimension	Type	In/Out	Description
IOP	1	Integer	In	Option for output =0, fill block with data in IBUF =1, fill block with 0's, if IBUF (1)=0 fill block with 1's, if IBUF (1) =1

- Description

BLKTHM receives the block coordinates, the theme number, the buffer address, and the option thru its arguments. Various control constants such as the number of even lines, the number of odd lines, and the starting line are computed using the coordinates. The selected lines are read from the selected theme into an array. If the option is zero the contents of the buffer are shifted into the array at the block defined from the coordinates. If the option is one, the first work of the buffer is checked. If it is zero, IFILL is set to all 0's. If it is one, IFILL is set to all 1's. The contents of IFILL are shifted into the array rather than the contents of the buffer. The modified array is then written back to the theme.

This process is repeated in bands of up to 32 lines at a time. First the even lines are done, then the odd lines and again the up to 16 even and up to 16 odd lines until the required block is filled.

3.5.2.5.1.8.2 DOTIN

This program selects the dot subset to be displayed.

- Calling sequence

CALL DOTIN (IO,II)

- Arguments

Parameter	Dimension	Type	In/Out	Description
IO	1	Integer	In	Unit number
II	1	Integer	Out	Status word

- Description

  DOTIN queries the user via a display terminal for dot subset desired. The subset is located and the dot listed. If acceptable, the program returns. If not, the selection process is repeated.

3.5.2.5.2 Dot Group Scatter Plot (SCPLOT) (Programmed and documented by T. Kell)

This program

A. Generates dot and cluster scatter plots

B. Erases windows

C. Does whole screen logical operations on themes.

3.5.2.5.2.1 Generate Scatter Plots (DGSCPL) (Programmed and documented by T. Kell)

3.5.2.5.2.1.1 Linkages

A. IMALIB

B. F4POTS

C. CSGDQH, LIN, FFUNC, SHELL, CLUSEL, DOTIN (See Vol. 3)

D. DGSCPL,PLOT,SETVID, SETWIN

E. Task 'WINDRM' performs operation 2 and task 'THLOPM' performs operation 3.

3.5.2.5.2.1.2 Interfaces

A. Common name COM1

Parameter	Set by	Referenced by
ACDATE	---	DGSCPL
CHNVBC	---	VALCK

B. Common name COM2

Parameter	Set by	Referenced by
ADATES	---	DSCPL
SOILGR	---	DGSCPL

C. Common name COM4

Parameter	Set by	Referenced by
ACDISP	FULOI3	DGSCPL
II1	"	"
G	"	"
B	"	"
SPWIND	SETWIN	SETWIN
NUMDOT	DOTIN	DESCPL
DOTARY	"	DGSLPL
GMIN	---	"
GMAX	---	"
FULL	---	"

3.5.2.5.2.1.3 Inputs

A. Tapes - none

B. Cards - none

C. Key-in - see User's Manual

D. Other - DOTS.TMP
CLUSTATP.TMP

3.5.2.5.2.1.4 Outputs

A. Reports - scatter plot report - see User's Manual

B. Diagnostics
"CAN NOT PROCEED" - re-requests dot or cluster option
"NO DOT SELECTED. TRY AGAIN. . ."
asks for dot,cluster selection again.

C. Other - File SCATXY.TMP

3.5.2.5.2.1.5 Storage Requirements

SCPLOT occupies 88 blocks.

3.5.2.5.2.1.6 Description

Only option 1 of 3.5.2.5.2.1 will be described. The user is requested to select the desired window, dots or clusters and which acquisition, the scaling for output, the desired channels, the desired theme and then calls DOTIN or CLUSTEL to select the appropriate items. The scatter plot points are then read from DOTS.TMP or CLUSTAB.TMP, scaled and plotted.

3.5.2.5.2.1.7 Flowcharts

3.5.2.5.2.1.8 Subroutines

3.5.2.5.2.1.8.1 SETWIN

Set scatter plot window

- Calling sequence
  CALL SETWIN (IO)

Parameter	Dimension	Type	In/Out	Description
IO	1	I	I	TILUN

- Description
  Program requests N,C, or C/R to get next window, cursor designated window or current window after displaying current window.

  If N is entered process restarts with next available window.

If C is entered the cursor coordinates are reentered and the entered window becomes the current window.

If C/R is entered, the PGM terminates.

3.5.2.5.2.1.8.2 SETVID

Display a line of constant value on the CRT.

- Calling sequence

  Call SETVID (Y,D)

Parameter	Dimension	Type	In/Out	Description
Y	1	I	I	CRT Line #
D	1	I	I	Output data

- Description

  Line Y is read for channels II1≠0, D is stored in the line over the desired range, and the line is re-output.

3.5.2.5.2.1.8.3 Plot

Plots scatterplot

- Calling sequence

  Call PLOT(K,I)

Parameter	Dimension	Type	In/Out	Description
K	1	I	IO	Index to first dot to be plotted this pass
I	1	I	I	Index to last dot to be plotted this pass

- Description

  On entry this program checks to see if the set of dots to be plotted on this call has the same vertical position as the line currently in the buffer. If not, the current buffer is written to the CRT and succeeding lines are read, axised, and written until the dot line is about to be modified. Then, or if the first test is ok, the next line is read and axised.

This is not done if the line has already been read. Then the dot data is plotted, the line is written and the next line is read, axised, plotted, and saved for the next pass. This process is repeated in successive calls.

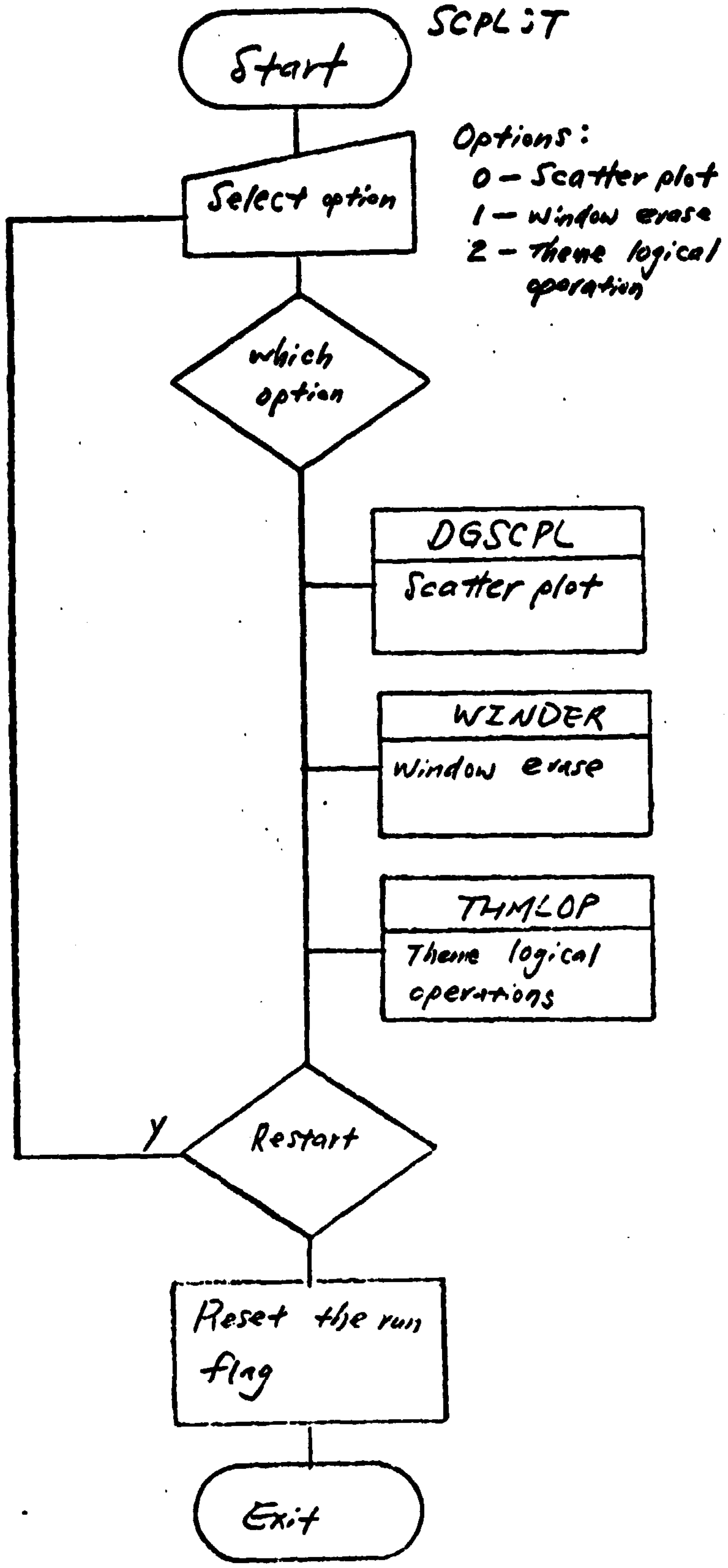


Figure 3-16.– Dot Group Scatter Plots.

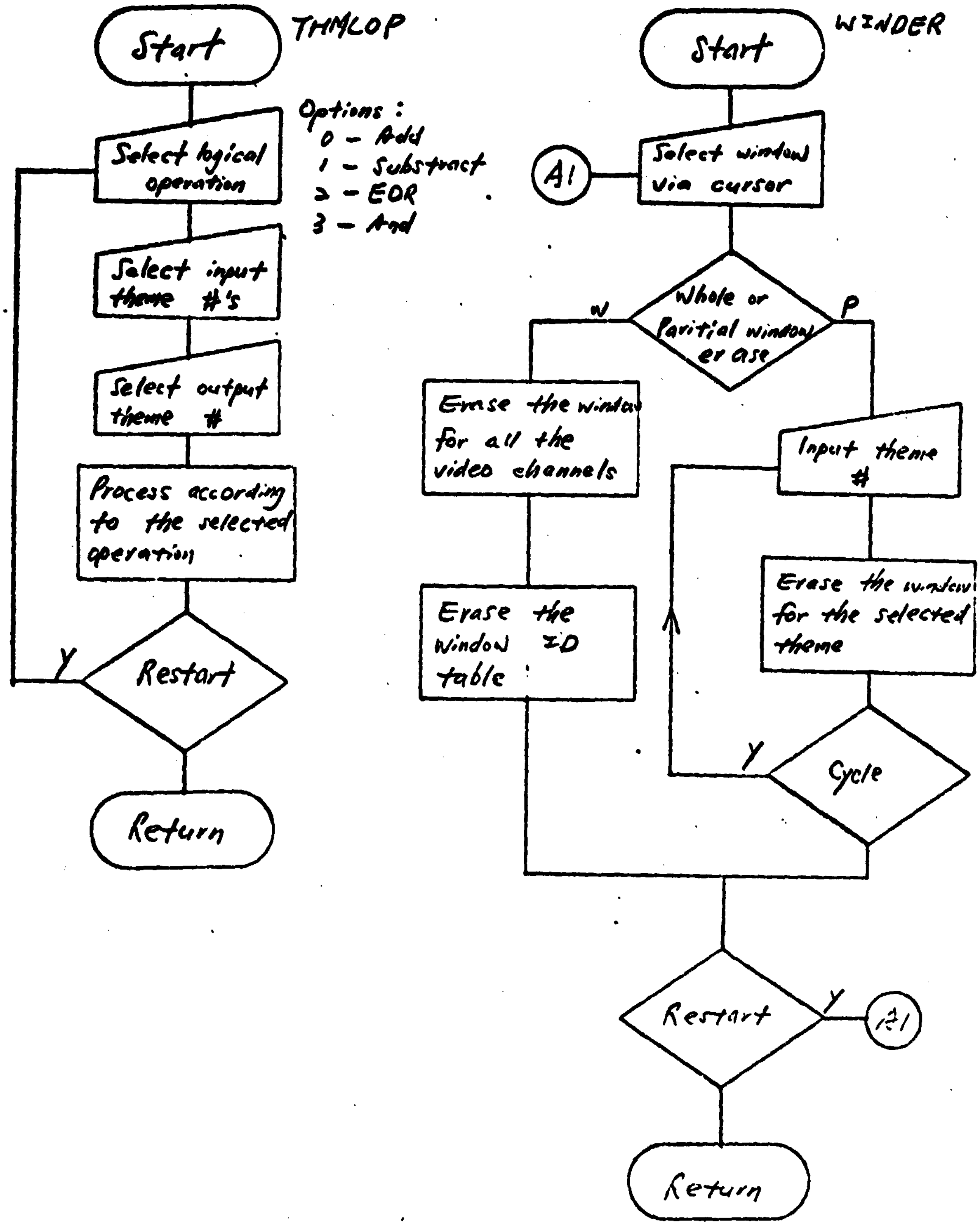


Figure 3-16.— Continued.

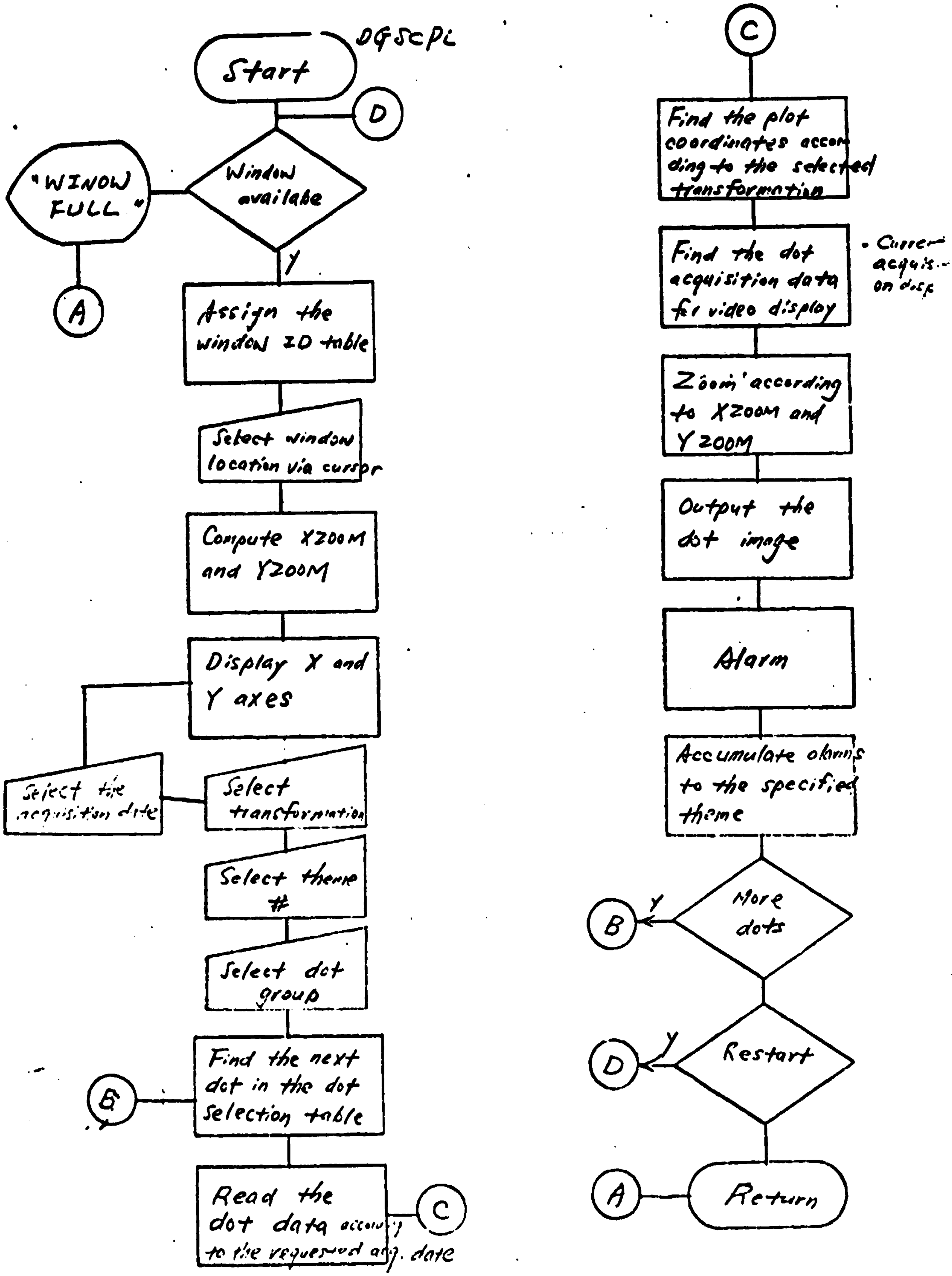


Figure 3-16. Continued.

SUBROUTINE: Dot Group Scatter Plot

	Messages	Type	Range	Error Message for Range Error	Default
1.	Select options: (S)catter plot, (W)indow erase or (T)heme logical operations. >	Q	S,W,T	N/A	N/A
2.	Select window location for scatter plot: ENTER "CR" to accept window as displayed, enter (N) to display next default window or enter (C) to use cursor to select desired location and size. >	Q	N,C	N/A	Display default window
3.	(Select window by cursor) Place cursor and enter "CR" when ready. >	Q	N/A	N/A	N/A
4.	No window available.	I	N/A	N/A	N/A
5.	Select options for scatter plots: (D)ot group or (C)luster means. N >	Q	D,C	N/A	N/A
6.	Select one acquisition date from the following dates for dot group scatter plot. (1) ______ (2) ______ (3) ______ (at most 6 acq.) (4) ______ (5) ______ (6) ______ N >	Q	1-MAXACD	Yes	Acq on display or PS (Previous selections)
7.	(If overridden in 6) Selected acquisition date = XXXXXX	I	N/A	N/A	N/A

181

SUBROUTINE: Dot Group Scatter Plot (cont)

	Messages	Type	Range	Error Message for Range Error	Default
8.	Select the coordinate system: Channels 1-4, (G)reenness, (B)rightness or green (N)umber. N1, N2 >	Q	1-4,B G,N	Yes	N, B or PS
9.	(If overridden in 8) Selected channels = X,X	I	N/A	N/A	N/A
10.	Select one acquisition date from the following dates for cluster means scatter plot. (1) ______ (2) ______ (at most 4 acq.) (3) ______ (4) ______ N >	Q	1-4	Yes	Acq. on display or PS
11.	(If overriden in 10) Selected acquisition date = XXXXXX	I	N/A	N/A	N/A
12.	Selected channels not available.	E	N/A	N/A	N/A
13.	Enter scaling: (F)ixed, (G)lobal or (A)nalyst input. >	Q	F.G.A	N/A	G or PS
14.	(If analyst inputs scale) Input XMAX, XMIN, YMAX and YMIN. >	Q	-5,127	Yes	N/A
15.	Selected scale: XMAX = , XMIN = , YMAX = , YMIN =	I	N/A	N/A	N/A
16.	Select output theme track number. N>	Q	1,8	Yes	2 or PS

182

SUBROUTINE: Dot Group Scatter Plot (cont)

	Messages	Type	Range	Error Message for Range Error	Default
17.	Scatter Plot Report: Dot Grid Number / X / Y XX / XX / XX (ordered in Y axis)	I	N/A	N/A	N/A
18.	Scatter Plot Report: Cluster Number / X / Y XX / XX / XX (ordered in Y axis)	I	N/A	N/A	N/A
19.	For next scatter plot. (S)ame window or (N)ew window >	Q	S,N	N/A	N/A
20.	(For same window) Acquisitions available for dot group scatter plot: Acq. dates / Channel XXX all XXX all XXX all XXX all XXX all XXX all (at most 6 acq.) For the current window in use: Acquisition date = ______ Channels = ______	I	N/A	N/A	N/A
21.	(For same window) Acquisitions available for cluster mean scatter plot: Acq. dates / Channel XXX XX XXX XX XXX XX XXX XX (at most 4 acq.)	I	N/A	N/A	N/A

SUBROUTINE: Dot Group Scatter Plot (cont)

	Messages	Type	Range	Error Message for Range Error	Default
	For the current window in use: Acquisition date = ______ Channels = ______				
22.	Proceed. (Y)es/(N)o?>	Q	Y,N	N/A	N/A
23.	Can not proceed.	I	N/A	N/A	N/A
24.	(Error message for all range check) Out of range.	I	N/A	N/A	N/A
25.	No soil green number for this acquisition. Channel N therefore, same as channel G.	I	N/A	N/A	N/A

184

3.5.2.5.2.2 Erase Windows (WINDER, WINDRM)

n.1 Linkage

A. IMALIB: IRK,IWK,INTFF,FRONT.

B. FORTRAN: CLOSE,OPEN

C. SHARED SUBROUTINES AND UTILITIES: BLKTHM,VDALTR

D. PRIVATE SUBROUTINE: NONE

n.2 Interfaces

A. COMMON NAME: COM4
PARAMETER UPDATE: NONE

n.3 Inputs

N/A

n.4 Output

A. N/A

B. N/A

C. FILE: [300,1]SCATXY.TMP.

n.5 Storage Requirements

n.6 The subroutine Winder consists of two subroutines BLKTHM and VDALTR which reads the cursor coordinates from the Keyboard input, searches the window from the globe common (COM4), accepts the theme track number or FULL to be erased from the Keyboard, then calls the subroutine BLKTHM to erase that particular theme track in the window or calls VDALTR to erase the full window.

n.7 Flow chart

See Figure 3-16a.

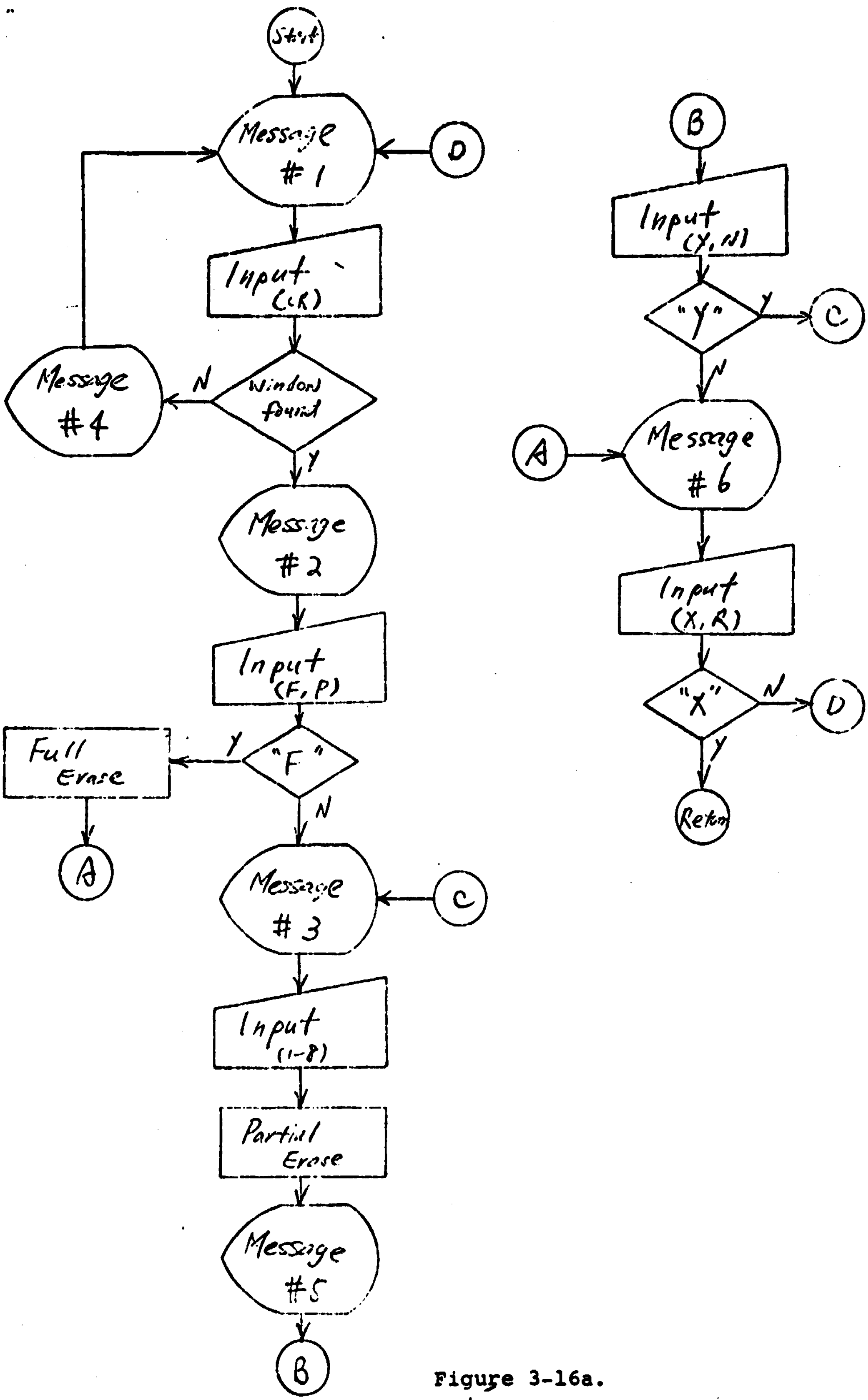

Start
Message #1
D
Input (CR)
Window found
N
Y
Message #4
Message #2
Input (F, P)
"F"
Full Erase
A
Message #3
C
Input (1-8)
Partial Erase
Message #5
B
Input (Y, N)
"Y"
Message #6
Input (X, R)
"X"
Return

Figure 3-16a.

SUBROUTINE: Window Erase

	Messages	Type	Range	Error Message for Range Error	Default
1.	Select window by cursor and enter "CR" when ready. >	Q	N/A	N/A	Shrink the cursor to a crosshair and display at the center of the screen.
2.	Select option for window erase, (P)artial or (F)ull. >	Q	P,F	N/A	N/A
3.	(Partial erase) Select theme track number to be erased. >	Q	1-8	Yes	N/A
4.	Cursor not in the window.	E	N/A	N/A	N/A
5.	(Partial erase) More theme tracks to be erased. (Y)es/(N)o? >	Q	Y,N	N/A	N/A
6.	E(R)ase another window or e(X)it >	Q	R,X	N/A	N/A
7.	Out of range. (Error message for range error)	E	N/A	N/A	N/A

3.5.2.5.2.3 Logical Operations on Themes (THLOPM, THMLOP)

n.1 Linkage

A. IMALIB; ICØM, JAND, INTFF, IWT, FRONT, WAIT, IRT

B. FORTRAN; IØR, IEØR

C. NONE

D. NONE

n.2 Interfaces

N/A

n.3 Inputs

N/A

n.4 Outputs

N/A

n.5 Storage Requirement

n.6 The subroutine THMLØP performs the logical operation (i.e. AND, OR, Exclusive OR, or substraction) between two input theme tracks which user inputs from the Keyboard, then output to the user selective theme track.

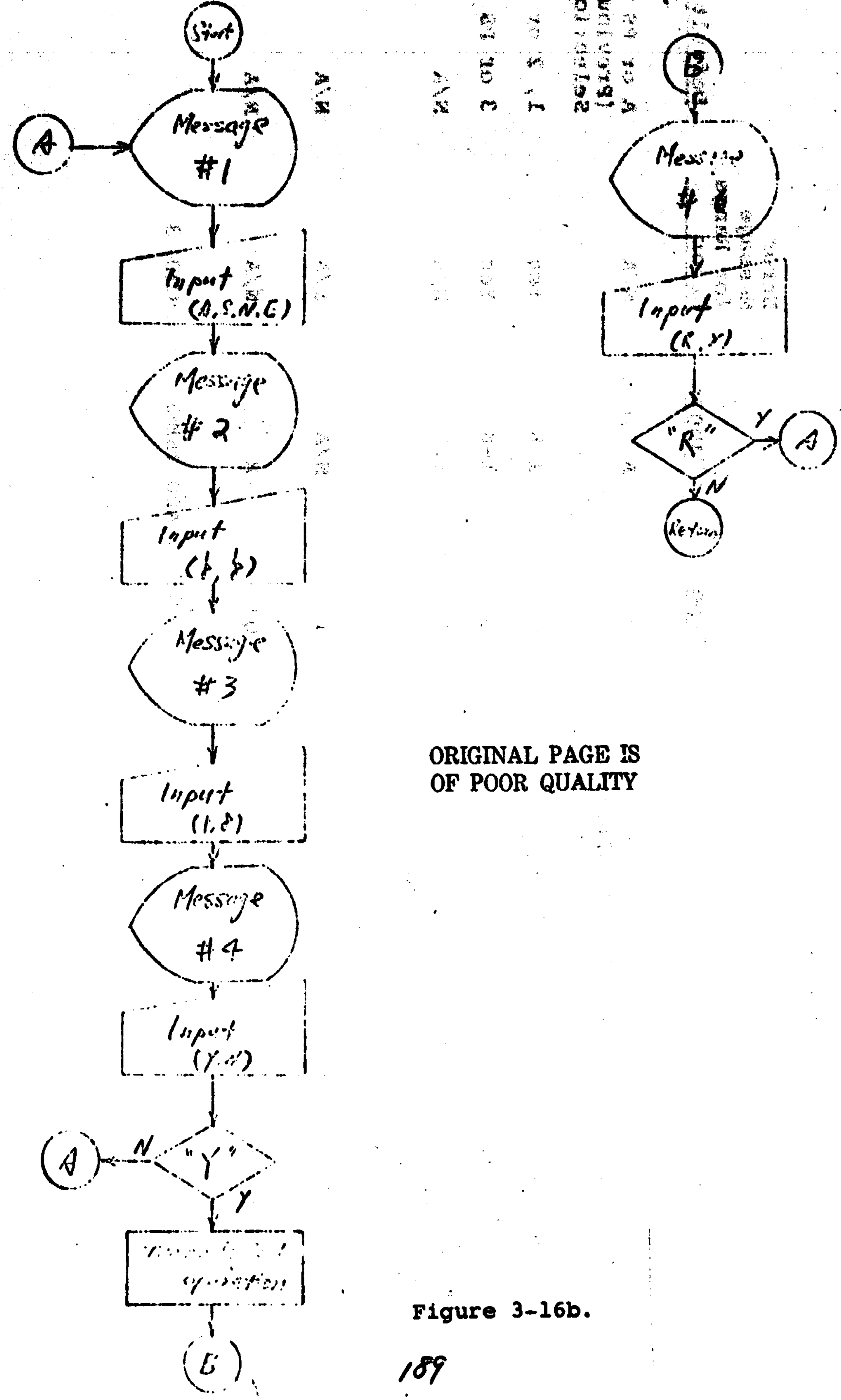
n.7 Flow Chart

See Figure 3-16b.

n.8 Subroutines

NONE

Flow logical operation indication



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 3-16b.

SUBROUTINE: Theme Logical Operation

	Messages	Type	Range	Error Message for Range Error	Default
1.	Select logical operation: (A)dd, (S)ubtract, a(N)d, (E)xclusive or N >	Q	A,S,N,E	N/A	A or PS (Previous Selection)
2.	Select Input Themes: N1,N2 >	Q	1-8	Yes	1, 2 or PS
3.	Select Output Theme: N >	Q	1-8	Yes	3 or PS
4.	Theme logical operation: Theme (1-8) <u>ADD, AND, SUB, EOR</u> Theme (1-8) = Theme (1-8) Proceed (Y)es/(N)o? >	Q	Y,N	N/A	N/A
5.	(For range check) Out of range!!!	I	N/A	N/A	N/A
6.	(R)estart or E(X)it >	Q	R	N/A	N/A

THEME 1 ADD THEME 2 = THEME 3

3.5.2.5.3 Single Dot Labelling (DTOPRD) (Programmed and documented by L. F. Robinson)

This program will perform the following functions:

- Dot selection for processing
- Window Erase
- Dot Blowup
- Trajectory plot(s)
- Group typing for current dot selection
- Single Dot labeling
- Single Dot typing

3.5.2.5.3.1 Linkages

*ATTACH BLOWUP *CLOSE *DETACH DOTLAB **ELAPS *EXIT FINDOT *FRONT GTYPE HSEKPC *IBYTE *IDATE *IRV *IWL *OUTPUT **SETEF STYPE *TIME TRAJPL VDALTR *WAIT **WINDER

*IMAGE 100 (IMALIB) LIBRARY system routines

**CAMS HYBRID system utility routines

3.5.2.5.3.2 Interfaces

A. COM1

Parameter	Set by	Reference by
ACDATE		DOTPRO
CATTH		DOTLAB

COM2

Parameter	Set by	Reference by
NOACQ		DOTPRO
ADATES		TRASPL, DOTPRO
SØILGR		DOTPRO
NTYPE1		STYPE, GTYPE

Parameter	Set by	Referenced by
CATNAM		DOTLAB, DOTPRO
NOCAT		DOTLAB

COM3

Parameter	Set by	Referenced by
UFLAG3	DOTLAB	DOTLAB

COM4

Parameter	Set by	Referenced by
ACDISP		DOTPRO
DTWIND		BLOWUP
SPWIND	TRAJPL	TRAJPL, FINDOT
NUMDOT		GTYPE, FINDOT
DOTARY		DTYPE, FINDOT
GMIN		TRAJPL
GMAX		TRAJPL
DLABEL	DOTLAB	DOTLAB, DOTPRO, GTYPE
TYPE		GTYPE
IMWIND		FINDOT
FUL		TRAJPL

B. External Files

1. [1]DBO: [300,1]DOTS.TMP
2. [1] [300,1]DOTGXY,TMP
3. [1] [300,1]SCATXY.TMP
4. [1]DBO: [300,1]GLOBAL.TMP;1

3.5.2.5.3.3 Inputs

User inputs to this program determine the type of dot processing to be accomplished: Erasing of window space, dot area enlargement, display of dot acquisition data in a trajectory plot, group typing of the dots in the current dot selection, single dot labeling and single dot typing. (See JSC - IMAGE - 100 USERS MANUAL, PART III CAMS Section 8).

3.5.2.5.3.4 Outputs

A. The outputs from this program consist of a Dot Data Report, alarmed Video data, Englarged Dot Video display, Dot Trajectory plots, altered Dot type and label information.

B. Diagnostics

   WARNING NO ACQUISITION ON DISPLAY (Program continues)

   Other diagnostics consist of a message being repeated if the reply to it is considered invalid.

3.5.2.5.3.5 Storage Requirements

DOTPRO occupies 86 blocks.

3.5.2.5.3.6 Description

DOTPRO clears the terminal screen, displays date/time, and calls FINDOT. FINDOT assists the user in selecting a particular dot for processing and returns that grid number to the main routine. DOTPRO then produces a Dot Data Report and alarms the Video screen with the selected dots' four channel data from the CIR image, or, if that is not available, from the dot data file. Next the user is asked to select an option from the list described in 3.5.2.5.3 above. These options are handled by subroutines described further on in this section.

3.5.2.5.3.7 Flowchart

Presented in figure 3-17.

3.5.2.5.3.8 Subroutines

3.5.2.5.3.8.1 FINDOT

This subroutine provides the user with the capability of selecting a particular dot grid number for processing.

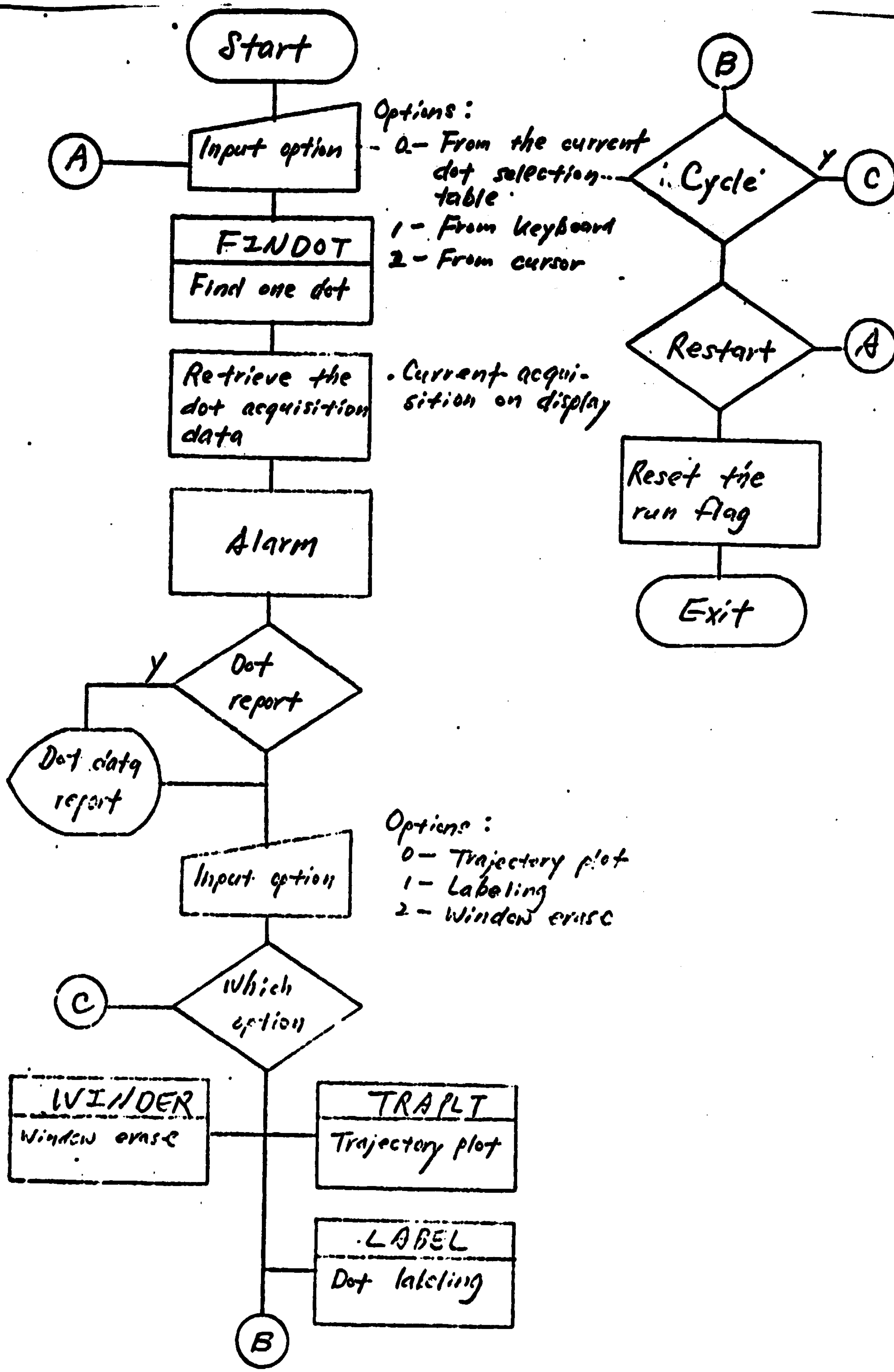


Figure 3-17.- Single Dot Labelling.

SINDOT-2

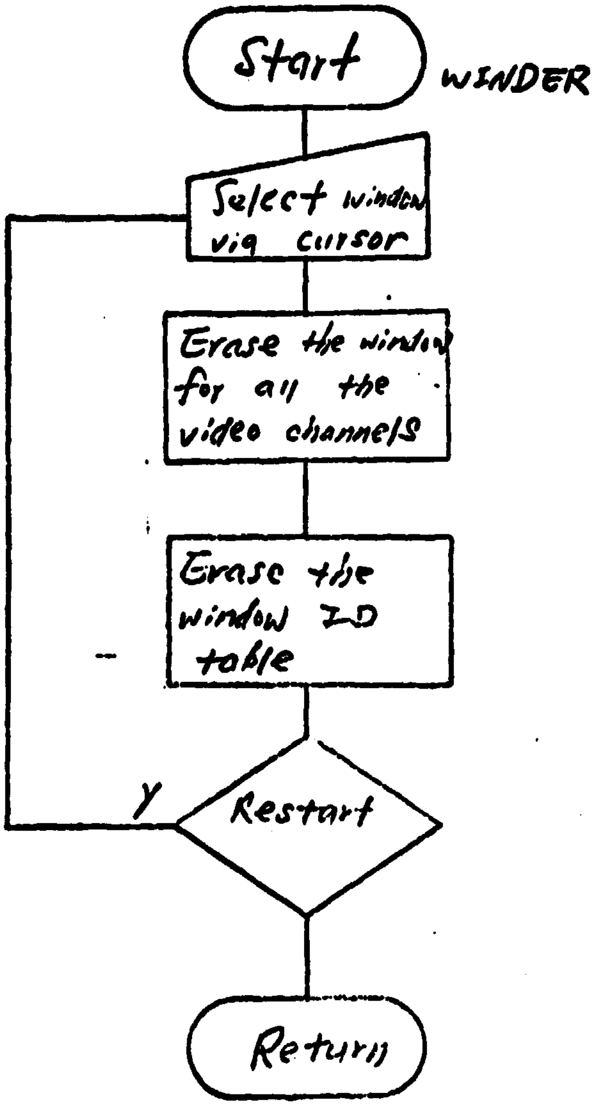
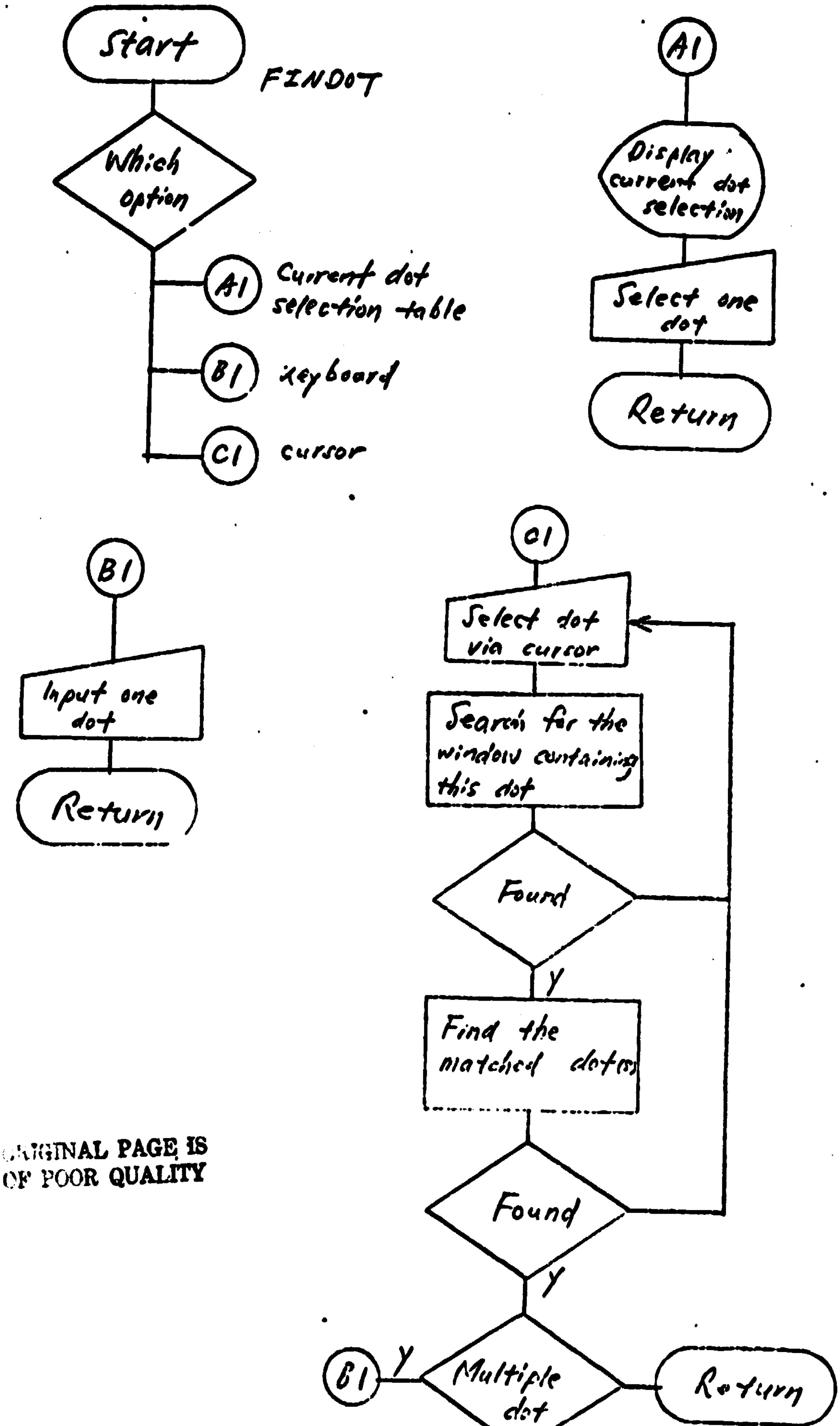


Figure 3-17.- Continued.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 3-17.– Continued.

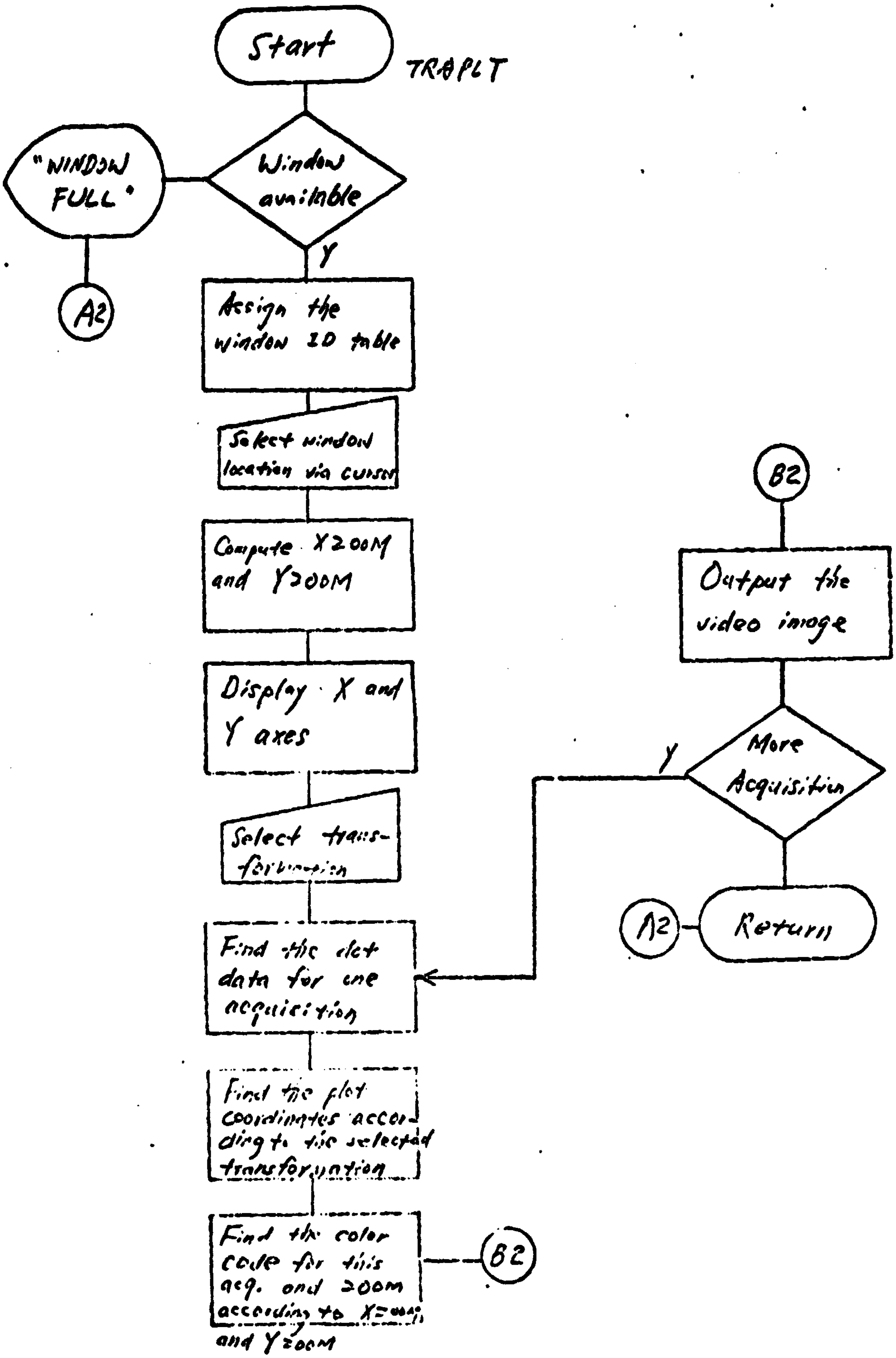


Figure 3-17.— Continued.

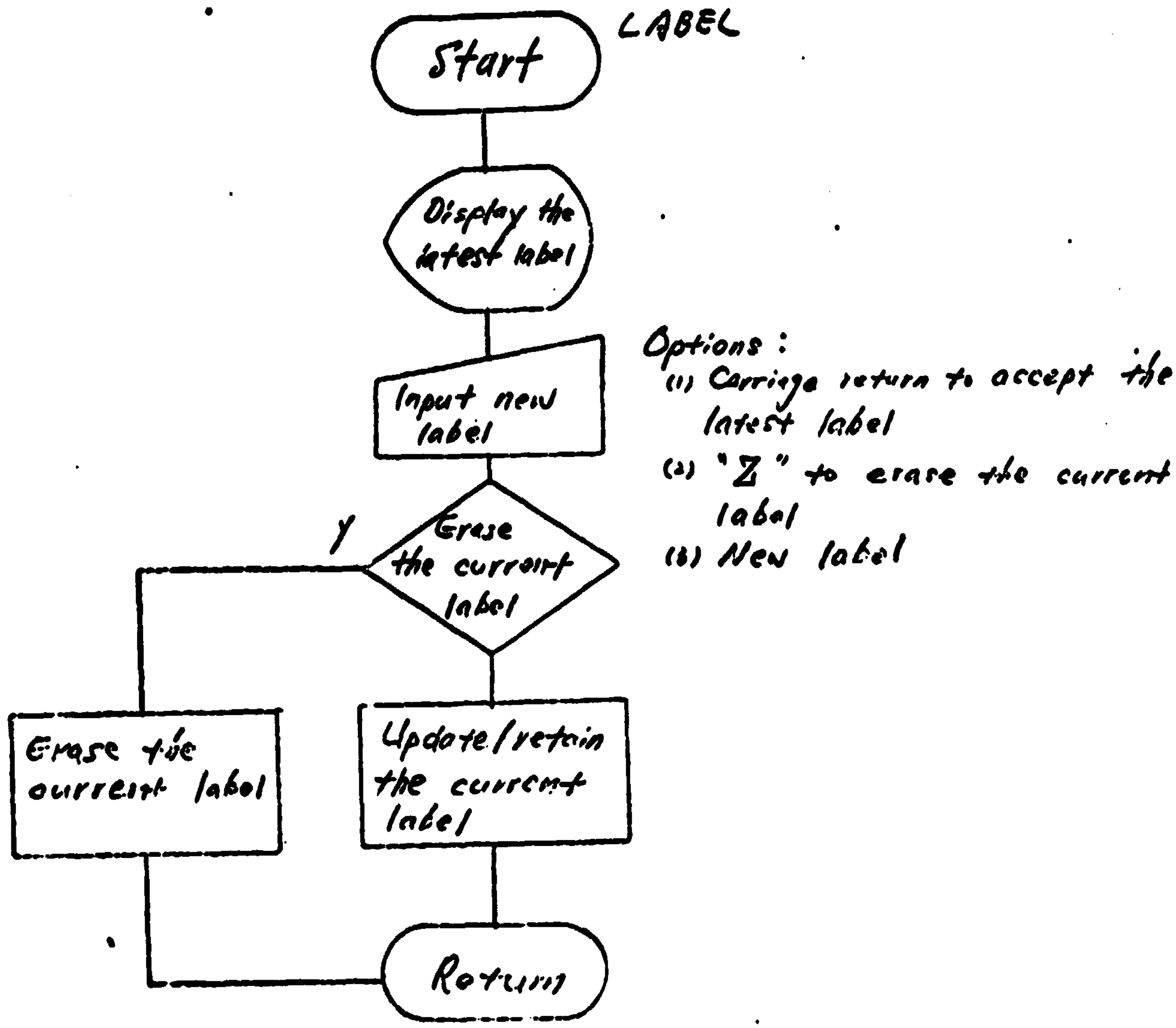


Figure 3-17.– Continued.

SUBROUTINE: Single Dot Labelling

	Messages	Type	Range	Error Message for Range Error	Default
1.	Select single dot from the current dot (S)election, (K)eyboard or (C)ursor. N >	Q	S,K,C	N/A	K,PS (Previous selections)
2.	(From the current dot selection) Current dot selection: X XX XX XX XX XXX XX XX X XX (10 dots/line) X X XXX X X X X X XX XXX Select one from the above dots. >	Q	One of the selected dots	Yes	N/A
3.	Selected dot not in current selection!!!	E	N/A	N/A	N/A
4.	(Input dot number by keyboard) Input dot grid number. >	Q	1-209	Yes	N/A
5.	(Error message for all range check) Out of range!!!	E	N/A	N/A	N/A
6.	(Select a dot by cursor) Select a dot by cursor and enter "CR" when ready. >	Q	CR	N/A	N/A
7.	Selected dot not found!!!	E	N/A	N/A	N/A
8.	Multiple dots found within cursor. XX XX XX XXX X X X XX X XX (10 dots/line) Select one from the above dots. >	Q	One of the multiple /dots	Yes	N/A
9.	Selected dot is not a multiple dot.	E	N/A	N/A	N/A

SUBROUTINE: Single Dot Labelling (cont)

	Messages	Type	Range	Error Message for Range Error	Default
10.	Dot data report: Random index = XX Grid index = XX Analyst label = XX Type = XX Special coordinates = X Green number = XX XX XX XX XX XX Acquisition data = XX XX XX XX XX XX XX XX	I	N/A	N/A	N/A
11.	Selection option from the following: (1) Window erase (2) Dot blow-up (3) Trajectory plot (4) Group typing for current dot selection (5) Single dot labelling (6) Single dot typing >	Q	1-5	Yes	N/A
12.	(R)es+art or E(X)it. >	Q	R,X	N/A	N/A
13.	Place the cursor around the dot for blow-up, adjust size and enter "CR" when ready. >	Q	N/A	N/A	N/A
14.	Place the cursor at upper left corner of the desired location for dot blow-up and enter "CR" when ready. >	Q	N/A	N/A	Set cursor to '+'.

200
'11

SUBROUTINE: Single Dot Labelling (cont)

	Messages	Type	Range	Error Message for Range Error	Default
15.	Proceed (Y)es/(N)o >	Q	Y,N	N/A	N/A
16.	Input the number for first labelled dots to be set to type 1 from current dot selection. (The remaining labelled dots are automatically set to type 2). >	Q	number of labelled dots selected	Yes	N/A
17.	M is more than the number of selected labelled dots.	E	N/A	N/A	N/A
18.	Previous label is P. Input new label for dot number M. >	Q	2 char. (Left justified) ZZ (unlabelled)	N/A	W or PS
19.	Label M is a new category.	I	N/A	N/A	N/A
19a.	{Label M for dot number N. / No label for dot number N.} (Y)es/(N)o? >	Q	Y,N	N/A	N/A
20.	No window available.	I	N/A	N/A	N/A
21.	Select window location for trajectory plot: Enter "CR" to accept window as displayed, enter (N) to display next default window or enter (C) to use cursor to select desired location and size. >	Q	N,C	N/A	Display default window
22.	Select the coordinate system: Channels 1-4, (G)reenness, (B)rightness or green (N)umber. N1,N2 >	Q	1-4,B G,N	Yes	N#,B or PS

201

SUBROUTINE: Single Dot Labelling (cont)

	Messages	Type	Range	Error Message for Range Error	Default
23.	(If overridden in 22) Selected channels = X,X	I	N/A	N/A	N/A
24.	Enter scaling: (F)ixed, (G)lobal or (A)nalyst input. >	Q	F,G,A	N/A	G or PS
25.	(If analyst inputs scale) Input XMAX, XMIN, YMAX and YMIN. >	Q	-5,127	Yes	N/A
26.	Selected scale: XMAX = , XMIN = , YMAX = , YMIN =	I	N/A	N/A	N/A
27.	Trajectory plot report for dot number M:	I	N/A	N/A	N/A

Acquisition date	Horizontal	Vertical	Color
XX (earliest)	X	XX	violet
XX	X	XX	blue
XX	X	XX	green
XX	X	XX	yellow
XX	X	XX	orange
XX(latest)	X	XX	red

(ordered in acq. date)

28.	(Select window by cursor) Place cursor and enter "CR" when ready. >	Q	CR	N/A	N/A
29.	Previous type is P. Input new type for dot number M. N >	Q	0,1,2	Yes	1 or PS
30.	(If overridden in 29) Type P for dot number M.	I	N/A	N/A	N/A

- Calling sequence

  CALL FINDOT (N,EXFL)

- Arguments

Parameter	Dimension	Type	In/Out	Description
N	1	INTEGER	OUT	dot grid number
EXFL	1	INTEGER	OUT	exit flag

- Description

FINDOT gives the user the option of selecting a dot via KEYBOARD entry, CURSOR encirclement or from the CURRENT DOT SELECTION. If the KEYBOARD option is used the entry is checked for $\geq$ or $\leq$ 209. In CURSOR encirclement the X,Y tables for scatter plots and the CIR iamge are compared to the cursor X,Y coordinated for a match and the resulting dot grid number(s) are supplied for verification. CURRENT DOT SELECTION causes a display of those dots contained in the current dot selection and the user is asked to choose from them. Error checking is done and one of the displayed dots must be chosen. At any point in this subroutine the user may EXIT, the subroutine (enter an "X" on the terminal) or back up one logical operation (enter a "B" or the terminal).

3.5.2.5.3.8.2 BLOWUP

This subroutine provides a 2 for 1 blowup of a designated area on the VIDEO display.

- Calling sequence

  CALL BLOWUP (EXFL)

- Arguments

Parameter	Dimension	Type	In/Out	Description
EXFL	1	INTEGER	OUT	exit flag

• Description

BLOWUP requests that the user size and position the cursor around the area to be enlarged and signal readiness with a "CR" entry. Upon receiving this "CR" the subroutine calculates the area to be blownup and reduces the cursor to a 2 x 2 pixel display. It then asks the user to position it where the enlargement is required. Another "CR" signals user readiness. At this point the cursor is expanded to show the size of the area and the message "PROCEED (Y)ES or (N)O" is displayed. A negative reply simply starts the subroutine at the beginning, a positive reply causes the blownup to occur.

3.5.2.5.3.8.3 TRAJPL

This subroutine will output a trajectory plot for up to six acquisitions of a given dot.

• Calling sequence

CALL TRAJPL (EXFL,N,GRNØB)

• Arguments

Parameter	Dimension	Type	In/out	Description
EXFL	1	INTEGER	OUT	exit flag
N	1	INTEGER	IN	dot grid number
GRNOB	6	INTEGER	IN	green number for up to 6 acquisitions of the dot grid number (N).

• Description

TRAJPL allows the user to select the display area for the plot either by using an available window of by cursor definition. A coordinate system is requested and then scaling factor are required of the user. The plot is then generated from the selected coordinates on an acquisition date hierarchy i.e.: latest date to earliest date.

3.5.2.5.3.8.4 GTYPE

This subroutine accomplishes group typing using the dots from the CURRENT DOT SELECTION.

- Calling sequence

CALL GTYPE (EXFL)

- Arguments

Parameter	Dimension	Type	In/Out	Description
EXFL	1	INTEGER	OUT	exit flag

- Description

GTYPE requests the user indicate, via the terminal keyboard, the number of dots from the CURRENT DOT SELECTION required to be set to type 1. The rest, if any, of the dots are set to type 2. The only exception to this rule is that unlabeled dots are ignored as are DO and DU dots.

3.5.2.5.3.8.5 DOTLAB

This subroutine is used to label a specified dot.

- Calling sequence

CALL DOTLAB (EXFL,N)

- Arguments

Parameters	Dimension	Type	In/Out	Description
EXFL	1	INTEGER	OUT	exit flag
N	1	INTEGER	IN	dot grid number

- Description

DOTLAB informs the user of the previous label, if any, for the dot in question and requests the new label. If the new label is a NEW CATEGORY the user is so informed. The user is then shown the label just requested and asked wheather to proceed or not. A negative reply recycles the subroutine, a positive reply gets the dot labeled and the subroutine then returns.

3.5.2.5.3.8.6 STYPE

This subroutine will change the type for a specified dot.

- Calling sequence

  CALL STYPE (EXFL,N)

- Arguments

Parameter	Dimension	Type	In/Out	Description
EXFL	1	INTEGER	OUT	exit flag
N	1	INTEGER	IN	dot grid number

- Description

STYPE informs the user of the previous type, if any for the dot in question and asks for the new type. If the new type is one the number of type one dots is increased by one, if the old type is one the number is decreased by one. The type change flag is then set and the subroutine does a return.

3.5.2.6 Automatic Cluster Labelling (ACLLAP) (Programmed and Documented by H. Thadani)

This algorithm labels clusters according to the majority label of the 'k' nearest type 1 labelling dots. Ties are resolved by discarding the *kth* dot in the random sequence and repolling.

3.5.2.6.1 Linkages

A. IMALIB - ATTACH, DETACH, FRONT, INTFF and OUTPUT.

B. FORTRAN - ASSIGN, CLOSE, ELAPSE, UPDATE, OPEN and TIME.

C. PRIVATE SUBROUTINES - CLABEL, KNNPRN, RDODAT, RDCLMN and ALSORT.

3.5.2.6.2 Interfaces

None

A. Common name COM1

Parameter	Updated by Subroutine	Referenced by Subroutine
ACDATE		ACLLAP
		RDODAT
		KNNPRN
CHNVEC		RDODAT
		RDCLMN
		CLABEL
		KNNPRN

B. Common name COM2

Parameter	Updated by Subroutine	Referenced by Subroutine
ISEG		ACLLAP
		KNNPRN
ADATES		ACLLAP
		RDODAT
		KNNPRN

Parameter	Updated by Subroutine	Referenced by Subroutine
SUNEL		CLABEL
NTYPE1		ACLLAP
		KNNPRN
CATNAM		ACLLAP

C. Common name COM3

Parameter	Updated by Subroutine	Referenced by Subroutine
EFLAG4	ACLLAP	ACLLAP
NEWLAB	ACLLAP	ACLLAP

D. Common name COM5

Parameter	Updated by Subroutine	Referenced by Subroutine
RANDOM		ACLLAP
DLABEL		ACLLAP
TYPE		ACLLAP

E. Common name HGT

Parameter/aray	Updated by Subroutine	Referenced by Subroutine
BUFDOT	RDODAT	RDODAT
BUFCLM	RDCLMN	RDCLMN
TDIS	CLABEL	CLABEL
CHANVC	RDODAT	RDODAT
NACQ	RDODAT	RDODAT
ALABEL	ACLLAP	ACLLAP
CLLAB	ACLLAP	ACLLAP
ILABEL	CLABEL	CLABEL
FLABEL	CLABEL	CLABEL
ARIND	ACLLAP	ACLLAP
	CLABEL	CLABEL

F. Working file name(s).

1. Dot data file - [300,1]DOTS.TMP
2. Cluster stat. file - [300,1]CLUSTATS.TMP

3. Nearest neighbor file - [300,1]NN.TMP;1

3.5.2.6.3 Inputs

A. Files - DOTS.TMP and CLUSTATS.TMP

B. Key-in -

   1. User keys in number of type 1 labelling data. Default = NTYPE1
   2. User keys in number of nearest neighbors. Default = k.
   3. User keys in report and output device.

3.5.2.6.4 Output

A. Reports

   1. Brief cluster labeling report (See Figure 3-18)
   2. Detailed nearest neighbor report

B. Files - NN.TMP file

C. Diagnostics - 1. If acquisitions of data base do not match those of classification message printed out on user console will be as follows: 'FATAL ERROR!!! CLASSIFICATION ACQUISITIONS DO NOT MATCH DATA BASE ACQUISITIONS'

3.5.2.6.5 Storage

ACLLAP occupies 93 blocks.

3.5.2.6.6 Description

This program labels clusters automatically using the following steps:

1. Compute L1 distances (corrected by the appropriate sun angle factor 5) to each of 'k' type 1 dots using the formula:

$$\text{DISTANCE (ISUB,K)} = \sum_{I=1}^{NOCHAN} (\text{MEAN(I,ISUB)} -$$

where MEAN (I,ISUB) - mean vector for channel I, subclass ISUB

DOTDATA (I,K) - dot data vector for channel I, dot K.

S(I) - sun angle correction factor for channel I

DISTANCE (ISUB,K) - L1 distance of mean ISUB to dot K.

2. For subclass ISUB sort distance vector DISTANCE in ascending order. Sort corresponding labels of dots maintaining random sequence.

3. Find majority label of nearest k distances. In the case of a tie, discard the *kth* dot and repoll.

4. Label cluster ISUB according to majority label.

5. Repeat procedure for all clusters.

3.5.2.6.7 Flow chart

See Figure 3-19.

3.5.2.6.8 Subroutines

3.5.2.6.8.1 RDODAT

This subroutine reads the dot data working file (DOTS.TMP) and returns dot data vector DOTDAT for all 'n' type 1 dots and user channels. It will also return diagnostic described in 3.5.2.6.4.C.

- Calling sequence

  CALL RDODAT (N, ARAND, DOTDAT, DFLAG)

- Arguments

Parameter	Dimension	Type	In/Out	Description
DOTDAT	16,NDOTS	I	O	dot data vector

BRIEF CLUSTER LABELLING REPORT

SEGMENT ID=9738
ACQUISITION(S)= 76148 76166 Ø Ø Ø Ø
CLUSTERING CHANNELS= 1 2 3 4 5 6 7 8
NUMBER OF TYPE 1 LABELLING DOTS= 1Ø
NUMBER OF NEAREST NEIGHBORS USED= 2

CLUSTER	NUMBER OF MAJORITY NEAREST NEIGHBORS	AUTOMATIC LABEL
1	1	W
2	2	N
3	2	N
4	2	N
5	1	N

E(X)IT, (R)ETURN, OR PAGE (F)ORWARD >F

BRIEF CLUSTER LABELLING REPORT

SEGMENT ID=9738
ACQUISITION(S)= 76148 76166 Ø Ø Ø Ø
CLUSTERING CHANNELS= 1 2 3 4 5 6 7 8
NUMBER OF TYPE 1 LABELLING DOTS= 1Ø
NUMBER OF NEAREST NEIGHBORS USED= 2

CLUSTER	NUMBER OF MAJORITY NEAREST NEIGHTORS	AUTOMATIC LABEL
6	2	N
7	2	W
8	2	N
9	1	N
10	2	N

E(X)IT, (R)ETURN, OR PAGE (F)ORWARD >X
E(X)IT OR (R)ECYCLE >X

Figure 3-18.— Automatic Brief Cluster Labelling Report.

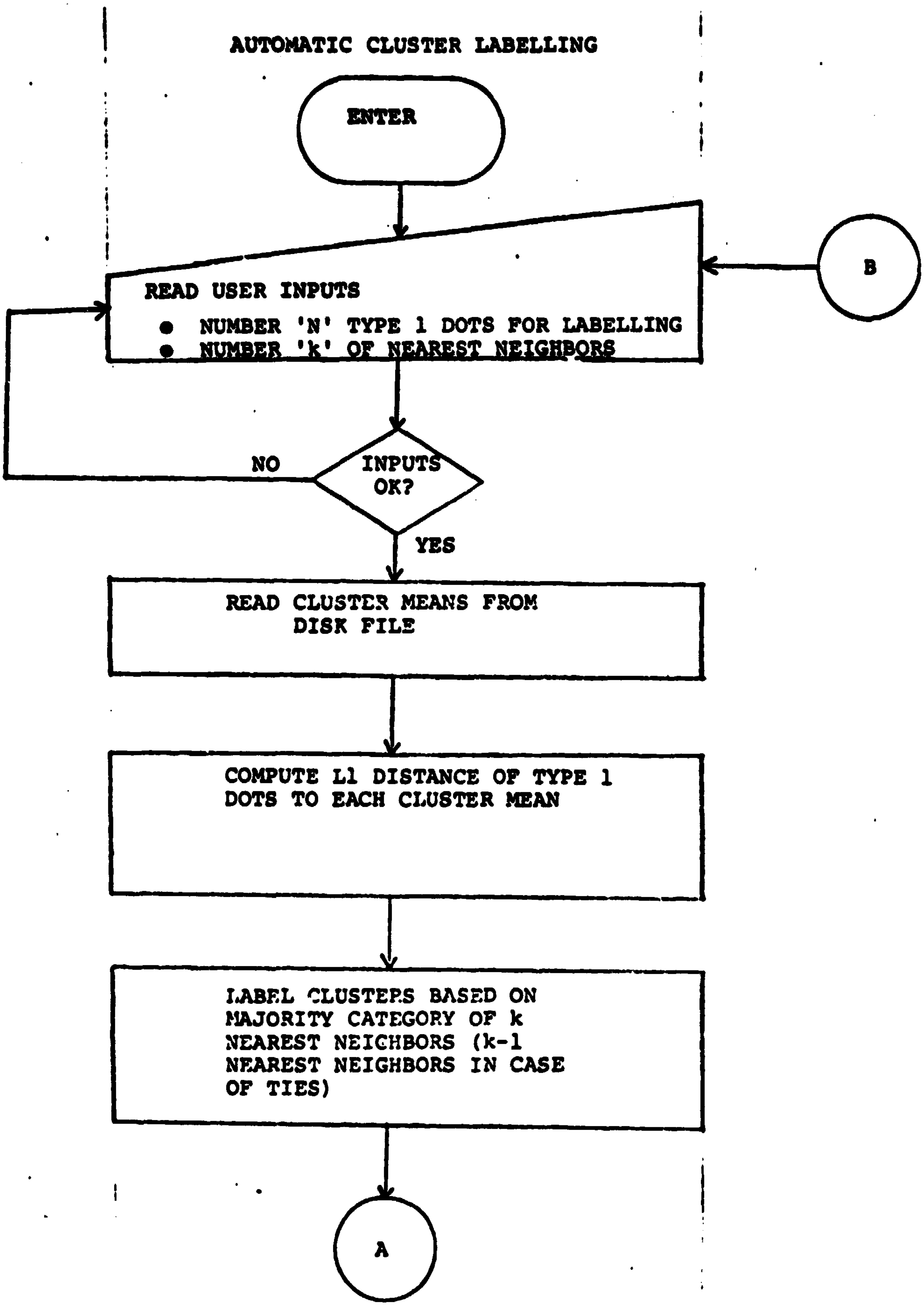


Figure 3-19.- Automatic Cluster Labelling.

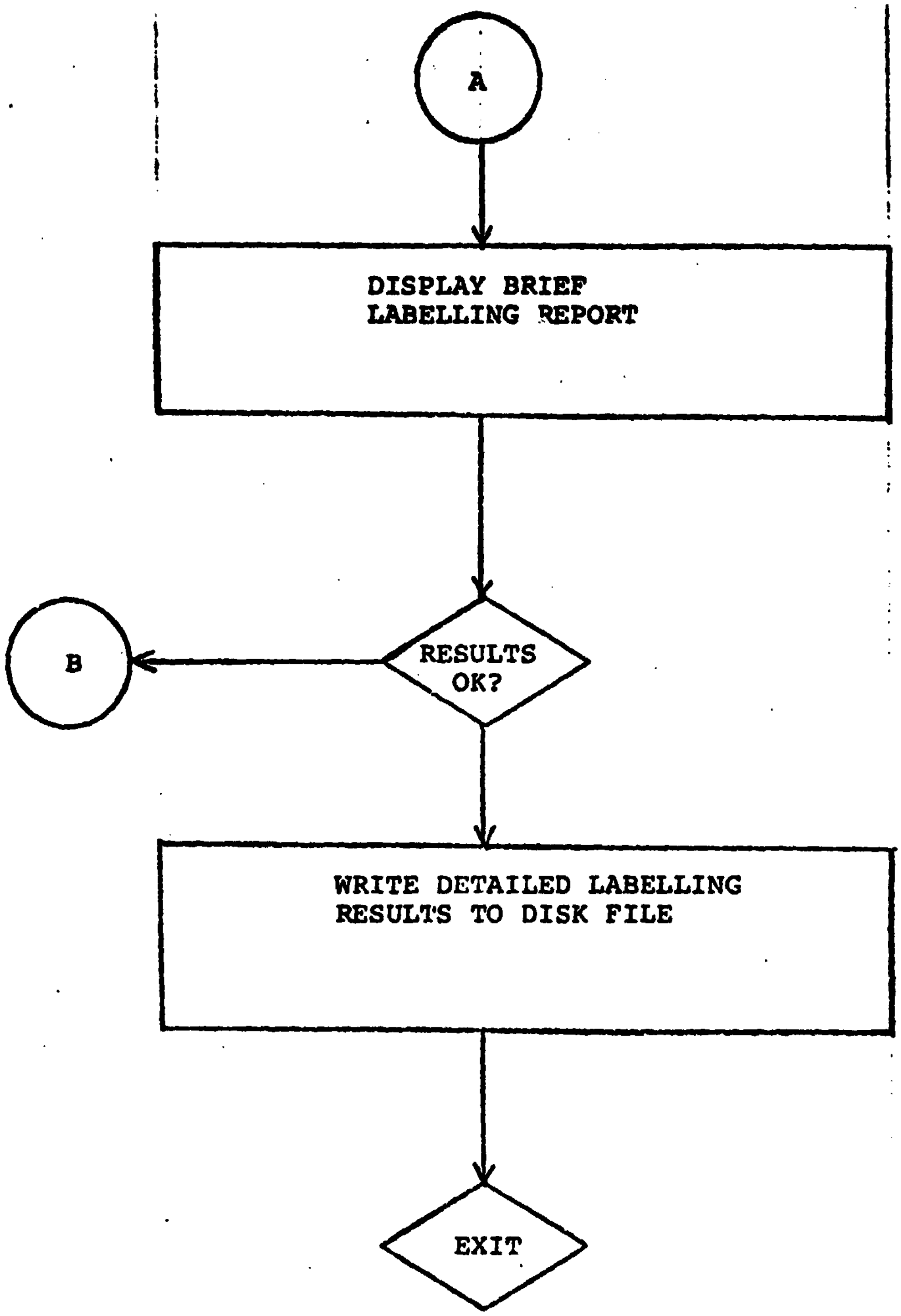


Figure 3-19.- Continued.

Parameter	Dimension	Type	In/Out	Description
N	1	I	I	number of type 1 dots
DFLAG	1	I	O	flag for diagnostic
ARAND	NDOTS	I	I	random dot index vector

3.5.2.6.8.2 RDCLMN

This subroutine reads the cluster is filed (CLUSTATS.TMP) and returns mean vector AMN for sublcass ISUB and all user channels.

- Calling sequence

  CALL RDCLMN (ISUB, ANM, ITOCHN, CHAN)

- Arguments

Parameter	Dimension	Type	In/Out	Description
ISUB	2	I	I	Subclass
AMN	16,MAXSUB	R	O	Mean vector
ITOCHN	1	I	I	No. of channels
CHAN	16	I	I	Channel vector

3.5.2.6.8.3 CLABEL

This subroutine computes and sorts the L1 distance vector in ascending order and labels the cluster according to the majority label. The sorting is done via subroutine ALSORT.

- Calling sequence

  CALL CLABEL (ISUB, N, AMN, DOTDAT, NCAT, ARAND, KNN, ITOCHN, ACAT)

- Arguments

Parameter	Dimension	Type	In/Out	Description
ISUB	1	I	I	Subclass #
N	1	I	I	No. of type 1 dots
AMN	16,MAXSUB	R	I	Mean vector
DOTDAT	16,NDOTS	I	I	Dot data vector
NCAT	1	I	I	No. of categories
ARAND	NDOTS	I	I	Random dot index vector

Parameter	Dimension	Type	In/Out	Description
KNN	1	I	I	No. of nearest neighbors
ITUCHN	1	I	I	No. of channels
ACAT	MAXSUB	I	I	Array containing category indices
TDIS	NDOTS	R	I	L1 distance vector

3.5.2.6.8.4 KNNPRN

This subroutine provides the option to and prints either of the 2 reports on one of the following devices:

1. The Gould Printer
2. The line printer
3. The users terminal

- Calling sequence

  CALL KNNPRN (A)

- Arguments

Parameter	Dimension	Type	In/Out	Description
A	1	I	I/O	A = 0 exit A = 1 recycle

3.5.2.7 Cluster Map Display (CLUDIS) (Programmed and Documented by G. Champagne)

This program produces unconditional, conditional, and mix cluster maps that may be displayed on the themes on the I-100.

3.5.2.7.1 Linkages

A. IMALIB

B. FORTRAN

C. SHARED: GETCOO, TWRITE, ZOOOM, BLKTHM

D. PRIVATE: EFWARN, UNCDIS, CONDIS, MIXDIS, LIST1, CATTHM, CLUTHM, CLABEL, DEFALT, CONDIT, LIST2, and MIXED

3.5.2.7.2 Interfaces

A. COM1

Parameter	Updated by Subroutine	Referenced by Subroutine
NOSUB		LIST1
		CATTHM
		CLUTHM
		CLABEL
		DEFALT
		CONDIT
		LIST2
		MIXED

B. COM2

Parameter	Updated by Subroutine	Referenced by Subroutine
NOCAT		CATTHM
		CLABEL
		DEFALT
CATNAM		LIST1
		CATTHM
		CLABEL
		DEFALT
		LIST1

C. COM3

Parameter	Updated by Subroutine	Referenced by Subroutine
EFLAG1		CLUDIS
EFLAG4		CONDIS
		MIXDIS
NEWLAB		LIST1
		CATTHM
		CLABEL
		DEFALT
		LIST2

D. COM4

Parameter	Updated by Subroutine	Referenced by Subroutine
TX1		GETCOO
TY1		GETCOO
TX2		GETCOO
TY2		GETCOO
IX1		GETCOO
IX2		GETCOO
IY1		GETCOO
IY2		GETCOO
CLUWND	UNCDIS	
	CONDIS	
	MIXDIS	

E. LOCOM2

Parameter	Updated by Subroutine	Referenced by Subroutine
CMASK		UNCDIS
		CATTHM
		CLUTHM
		TWRITE
		CLABEL
		DEFALT

Parameter	Updated by Subroutine	Referenced by Subroutine
		CONDIS
		MIXDIS

F. ZOOM

Parameter	Updated by Subroutine	Referenced by Subroutine
IC		UNCDIS
		CATTHM
		CLUTHM
		TWRITE
		CONDIS
		MIXDIS
TC		Same as above
IX		Same as above
IY		Same as above
TX		Same as above
TY		Same as above
MX		Save as above
MY		Same as above

G. FATAL

Parameter	Updated by Subroutine	Referenced by Subroutine
ZO		UNCDIS
		CONDIS
		MIXDIS
		ZOOM

H. Working File Names

CLUSTERMP.TMP

NN.TMP

3.5.2.7.3 Input

A. N/A

B. N/A

C. See User's Manual

3.5.2.7.4 Outputs

A. N/A

B. ' *** INPUT ERROR *** '

3.5.2.7.5 Storage

CLUDIS occupies 94 blocks.

3.5.2.7.6 Description

The module is a driver and requests options from the user for determining the kind of cluster map desired. NN.TMP and CLUSTERMP.TMP are opened and the existence flag for CLUSTERMP is checked to see if it is on. Upon termination, global common is saved.

3.5.2.7.7 Flow Chart

Functional flow chart is to be found in Figure 3-20.

3.5.2.7.8 Subroutines

3.5.2.7.8.1 EFWARN

Warns user that CLUSTERMP.TMP is not available.

- Call EFWARN
- N/A
- The program tells the user that the cluster map file is not on working storage and that the user may get cluster map reports if he wants them.

PRECEDING PAGE BLANK NOT FILMED

3.5.2.7.8.2 UNCDIS

Displays unconditional cluster map

- CALL UNCDIS
- N/A
- The program allows the user to change the default input coordinates of the cluster map file and the default output coordinates for display of the themes. The user is given options to list the clusters and categories, assign categories to themes, assign clusters to theme, and change cluster labels. The user may write the clusters to theme.

3.5.2.7.8.3 CONDIS

Finds the conditional clusters

- CALL CONDIS
- N/A
- Everything that applies to UNCDIS applies to this program. In addition, this program finds the conditional clusters.

3.5.2.7.8.4 MIXDIS

Finds the mixed clusters

- CALL MIXDIS
- N/A
- Everything that applies to UNCDIS applies to this program. In addition, this program finds the mixed clusters.

3.5.2.7.8.5 GETCOO

Gets coordinates from the user.

- CALL GETCOO (IC, TC, ISET)
- Arguments

Parameters	Dimension	Type	In/Out	Description
IC	4	Integer	Out	Output coordinates
TC	4	Integer	Out	Input coordinates
ISET	1	Integer	Out	Flag

- The default input and output coordinates are displayed to the user. The user may accept the default coordinates or change them to some other acceptable coordinates.

3.5.2.7.8.6 LIST1

Displays cluster numbers and their categories

- CALL LIST1
- N/A
- The program determines how many lines will be needed for display of cluster numbers and their categories and their proceeds to display those clusters and categories on TI.

3.5.2.7.8.7 CATTHM

Assigns categories to theme for display

- CALL CATTHM
- N/A
- The user tells the program which categories are to be assigned to which themes. In addition, the user has the option of adding to or replacing what is in the theme and he may also have category displayed as is or have the negative of category displayed.

3.5.2.7.8.8 CLUTHM

Assigns clusters to theme for display

- CALL CLUTHM
- N/A
- The user tells the program which clusters are to be assigned

to which themes. In addition, the user has the option of adding to or replacing what is in the theme and he may also have the clusters displayed as is or have the negative of the cluster displayed.

3.5.2.7.8.9 CLABEL

Changes cluster labels

- CALL CLABEL
- N/A
- The clusters and their categories are displayed to the user. The user has the option to change the category of any cluster. The change is made on global common.

3.5.2.7.8.10 TWRITE

Writes out the clusters to themes.

- CALL TWRITE
- N/A
- The clusters and categories which have been assigned to themes are written to the themes on the I-100 by this program. The cluster map file is read to locate the cluster location and the corresponding bytes are turned on on the I-100.

3.5.2.7.8.11 DEFALT

Assigns categories to themes

- CALL DEFALT (CMASK)
- Parameter / Dimension / Type / In/Out / Description

Parameter	Dimension	Type	In/Out	Description
CMASK	60	Byte	Input	Cluster mask

- If the user decides to accept the defaults in category to theme assignment, the program will assign the categories of that segment to the themes, beginning with category one and theme one.

3.5.2.7.8.12 CONDIT

Determines which of the clusters are conditional

- CALL CONDIT (COND, ISET)

Parameter	Dimension	Type	In/Out	Description
COND	MAXSUB	Integer	Output	Conditional clusters
ISET	1	Integer	Output	Flag

- This program reads the nearest neighbor file and with that information determines which of the clusters are conditional. When a conditional cluster is found, the corresponding element is set to '**' in the array COND.

3.5.2.7.8.13 LIST2

Displays clusters, categories, and conditional or mixed clusters.

- CALL LIST2 (DATA, IFLG)

Parameter	Dimension	Type	In/Out	Description
DATA	MAXSUB	Integer	Input	Mixed or conditional clusters
ISET	1	Integer	Input	Flag

- Everything that applies to LIST1 applies to this program. In addition, mixed or conditional clusters, as indicated by the flag, is displayed also.

3.5.2.7.8.14 MIXED

Determines which of the clusters are mixed.

- CALL MIXED (MIX, ISET)

Parameter	Dimension	Type	In/Out	Description
MIX	MAXSUB	Integer	Output	Mixed clusters
ISET	1	Integer	Output	Flag

- This program reads the nearest neighbor file and with that information determines which of the clusters are mixed.

When a mixed cluster is found, the corresponding element is set to '**' in the array MIX.

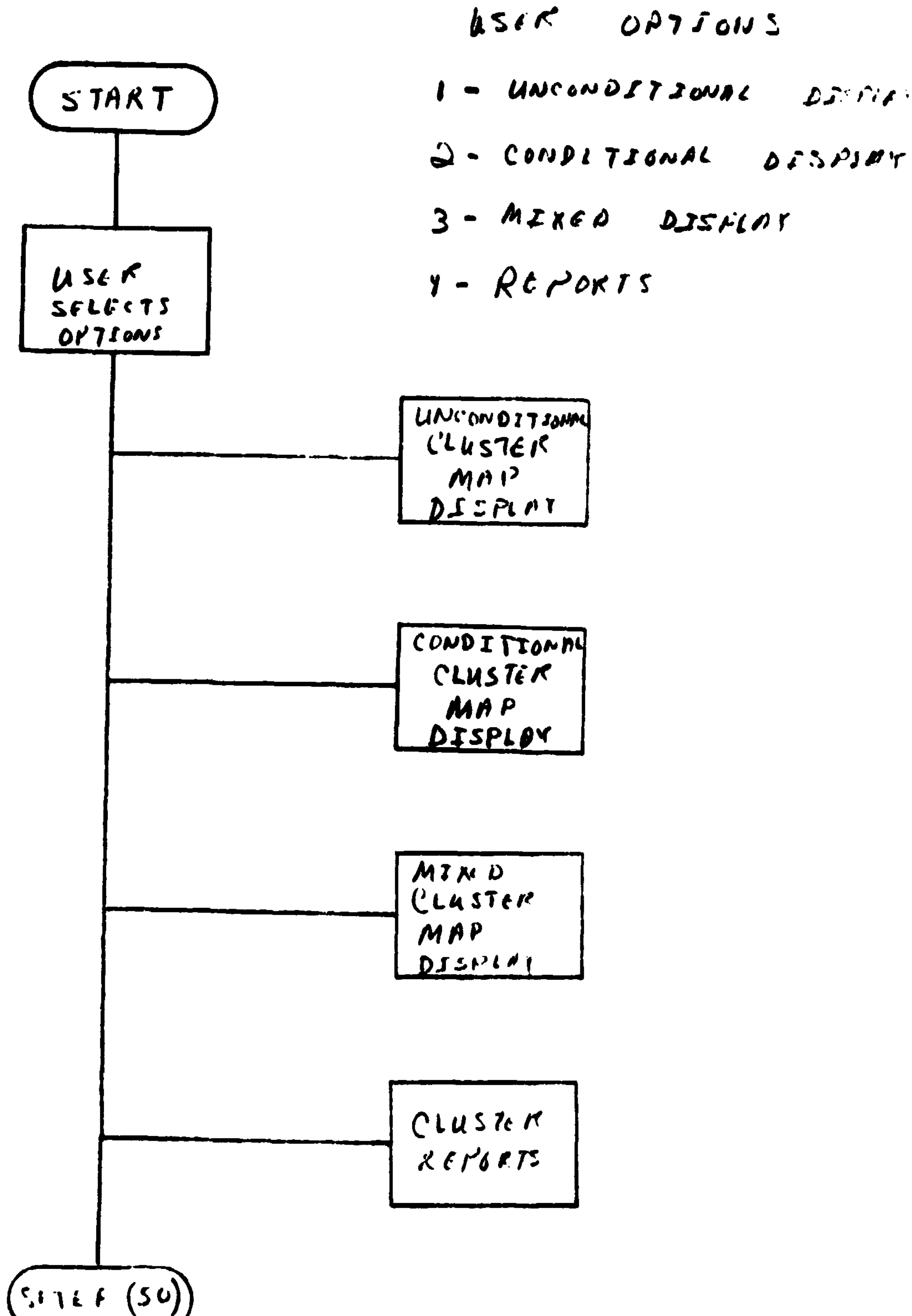


Figure 3-20.- Cluster Map Display

UNCONDITIONAL CLUSTER MAP DISPLAY

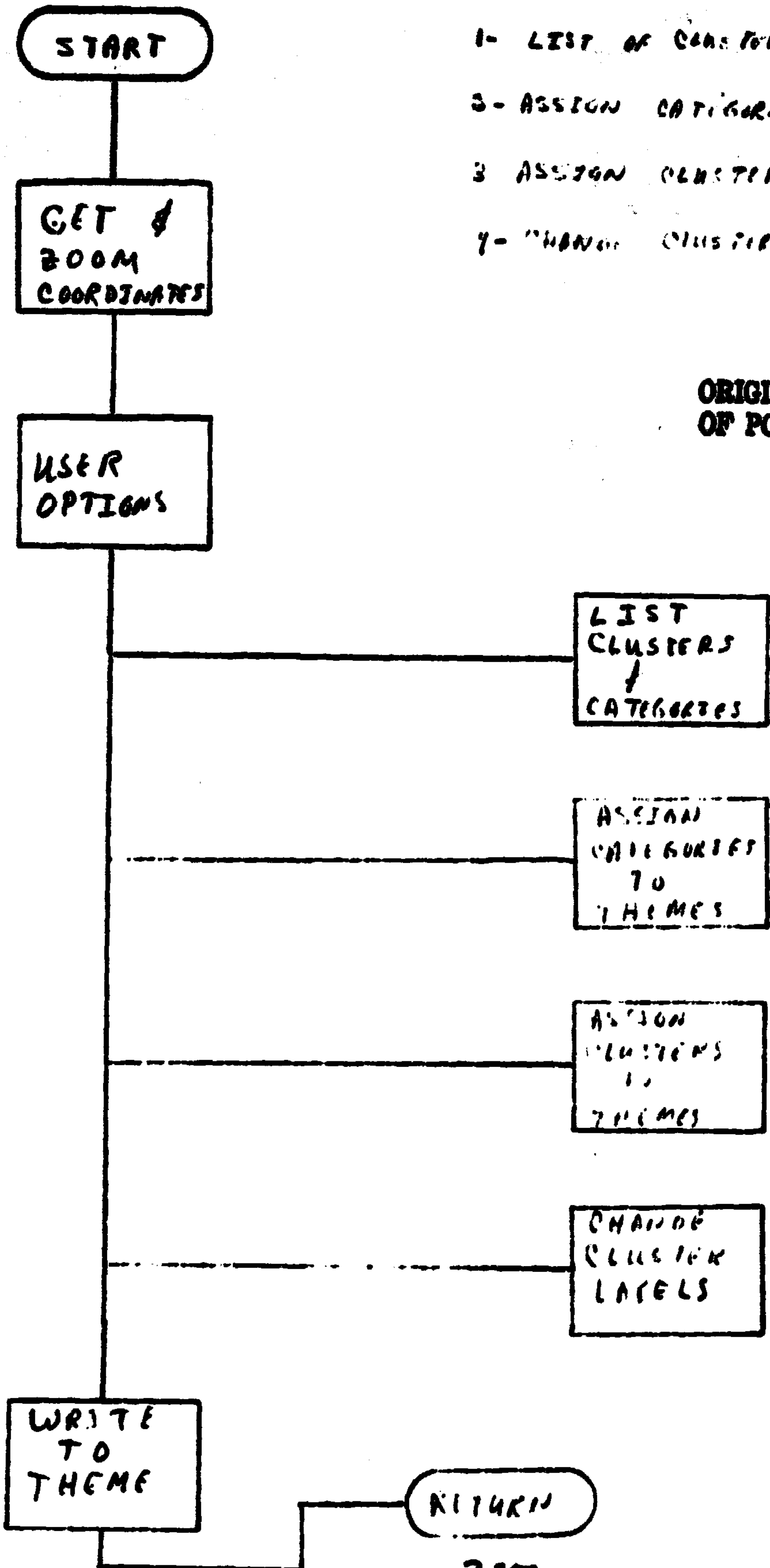
USER OPTIONS:

1- LIST OF CLUSTERS & CATEGORIES

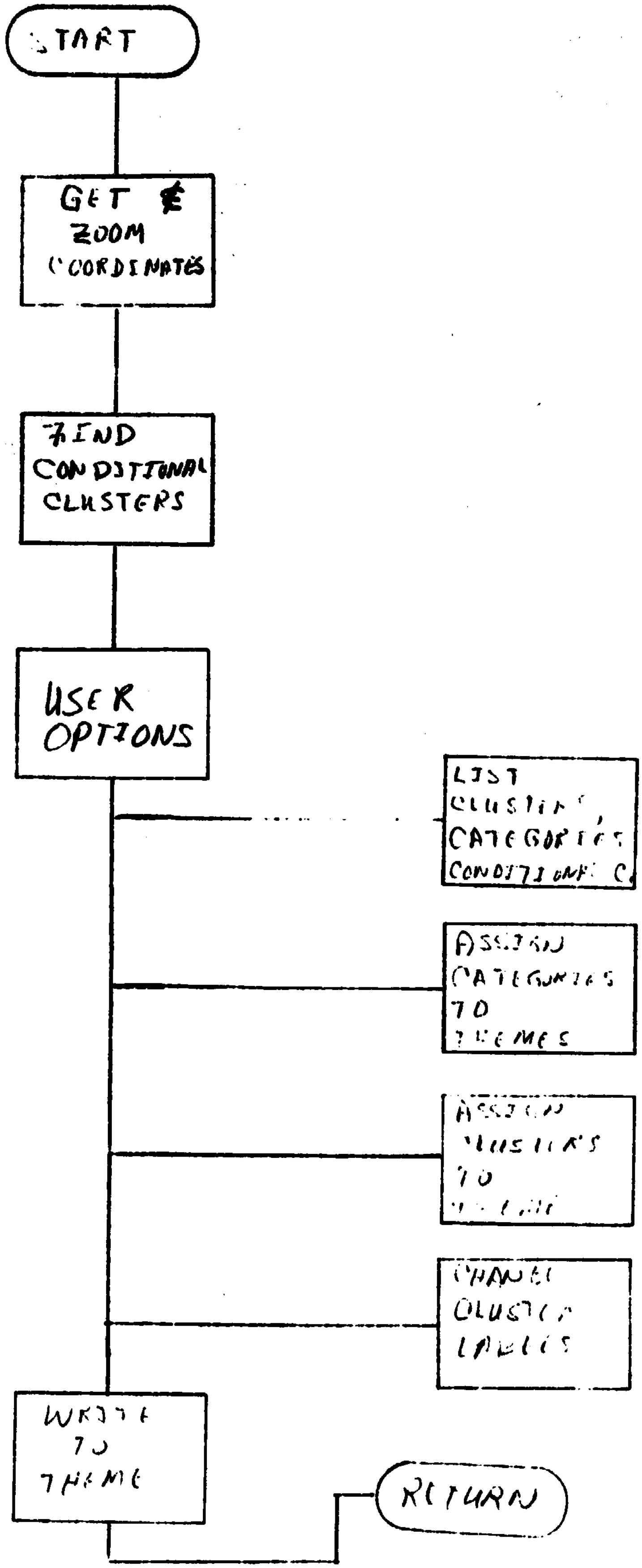
2- ASSIGN CATEGORIES TO THEMES

3 ASSIGN CLUSTER TO THEMES

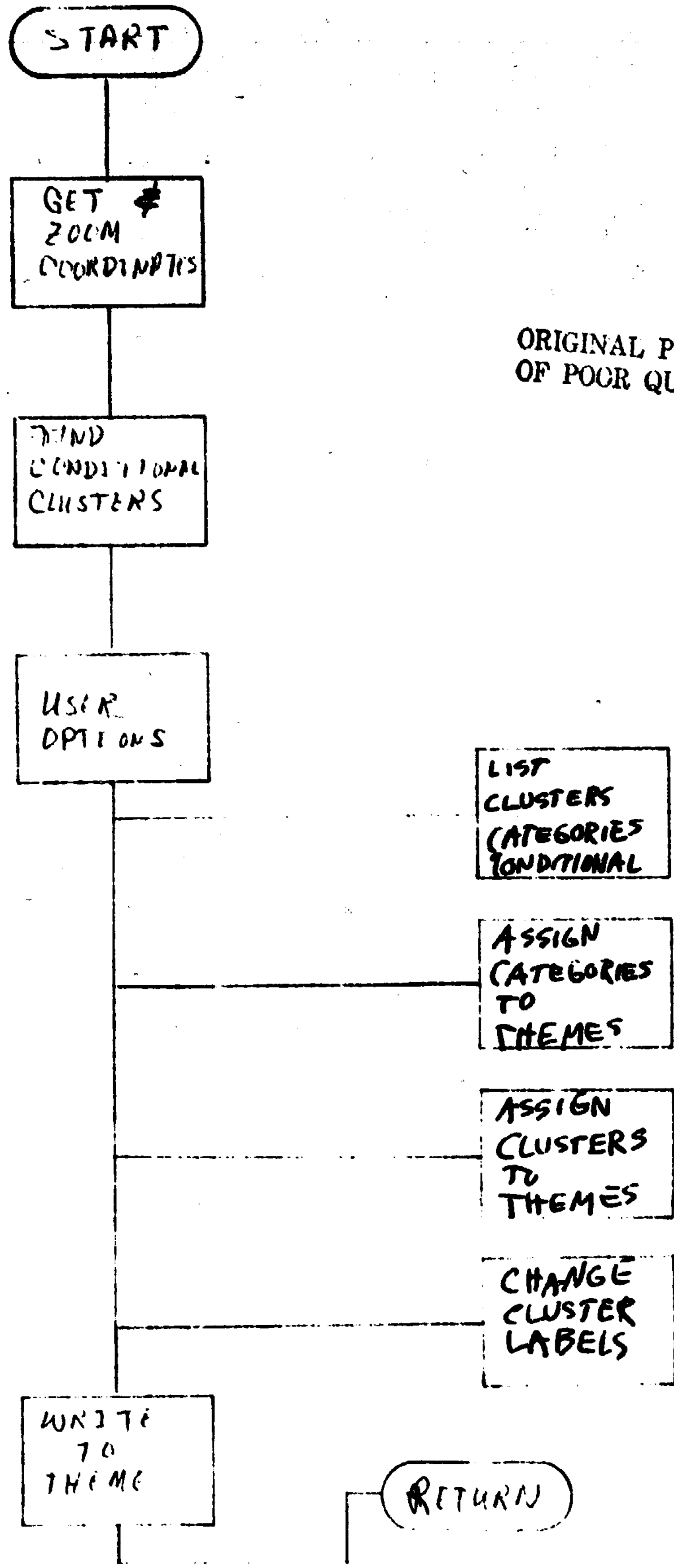
4- CHANGE CLUSTER LABELS



CONDITIONAL CLUSTER MAP DISPLAY



MIXED CLUSTER MAP DISPLAY



3.5.2.8 Recompute Proportions/Classification Map Display (RECPRO,REPROP,CLASS) (Programmed by H. Thadani and G. Champagne and documented by H. Thadani)

This algorithm reads the classification and cluster maps and reassigns pixels to the category they have been relabelled to and updates the corresponding category proportions. It also gives the option of displaying the classification map before or after the

3.5.2.8.1 Linkages

A. IMALIB - ATTACH, DETACH, FRONT, INTFF, and OUTPUT, CLATHM, ZOOOOM, BLKTHM.

B. FORTRAN - ASSIGN, CLOSE, CLASS, ELAPSE, IDENTI, OPEN, and TIME

C. PRIVATE SUBROUTINES
   1. Recompute proportions - REPROP, RECPRN, FDLINT
   2. Classification map display - CLADIS, LIST3, CLATHM, and DIFLT2.

3.5.2.8.2 Interfaces

A. Common name RCP

Parameter	Updated by Subroutines	Referenced by Subroutines
ICAKNT	REPROP	REPROP
IRESUB	REPROP,RECPRN	REPROP,RECPRN
FLDNAM	REPROP,FDLINT	REPROP,FDLINT
LABEL	REPROP	REPROP,RECPRN
VERTEX	REPROP	REPROP,FDLINT
NV		
FIELD		
LOWHI		
FL		
FLL		

Parameter	Updated by Subroutines	Referenced by Subroutines
DIRCAT		
CHAN	REPROP	REPROP,RECPRN
X		REPROP,RECPRM
RECPCT		RECPRN
PERDO		RECPRN
PERDU		RECPRN
PERTH		RECPRN
PERUND		RECPRN
NLP		RECPRN
NDOPIX		RECPRN
NDUPIX		RECPRN
NTH		RECPRN
UNID		RECPRN

The recomputed proportions application program references all of common blocks COM1 and COM3 and a few parameters from the COM2, COM4, and COM5 common blocks.

B. Working Files

1. Recompute proportions
    a. CLASSMAP.TMP
    b. CLUSTERMP.TMP
    c. FIELDS.TMP
2. Classification map display
    a. CLASSMAP.TMP

3.5.2.8.3 Inputs

A. Key-in - refer to User's Manual

3.5.2.8.4 Outputs

A. Reports

1. Recompute Proportions
    There are two reports displayed by this module.

a. The relabelled cluster report

b. Recomputed proportions report

2. Classification Map Display

Refer to User's Manual

3.5.2.8.5 Storage

RECPRO, REPROP, and CLASS occupy 34, 68, and 68 blocks respectively.

3.5.2.8.6 Description

This module drives two separate application programs - the recomputed proportions application program and the classification map display application program. This is done via a driver (acronym: PLEASE). Description of the classification map display application program will not be given here since it is identical to the unconditional cluster map display application program which has been documented elsewhere.

Description of the recomputed proportion application program follows. This program begins by determining the relabelled clusters. It then checks the existence of the classification map, cluster map and fields working files. The classification and cluster maps are then read simultaneously a line at a time. For every line read the DO/DU field intercept information is stored using subroutine fdlint. The category counts are updated for every line read. Finally, dot categories are updated and a detailed report displayed.

3.5.2.8.7 Flow Charts

Figure 3-23.

3.5.2.8.8 Subroutines

3.5.2.8.8.1 FDLINT

- Calling sequence

CALL FDLINT (FIELD, JJL, FLL, NOLIN, WSAMP, JJ)

Parameter	Dimension	Type	In/Out	Description
FIELD	2XNPTS	I	I	Field vertices array
JJL	1	I	I	No. of vertices
FLL	8	I	O	Array containing the ordered pixel intercepts
NOLIN	1	I	I	Sure line numbers
NSAMP	1	I	O	No. of samples contained in the field of a given line
JJ	1	I	O	The length of the array FLL

3.5.2.8.8.2 RECPRN

This subroutine prints the recomputed proportions. The formulas used are:

1. NLP = NLIN x NPIX
2. UNID(X) = NLP - ICAKNT(X) - NDUPIX - NTH
3. $\text{RECPCT(I)} = \dfrac{100 \times (\text{ICAKNT(X)})}{\text{UNID}}$
4. $\text{RECPCT(X)} = 100 \times \dfrac{(\text{ICAKNT(L)})}{\text{NLP}}$
5. $\text{PERDO} = \dfrac{\text{NDOPIX}}{\text{NLP}} \times 100$
6. $\text{PERDU} = \dfrac{\text{NDVPIX}}{\text{NLP}} \times 100$
7. $\text{PERTH} = \dfrac{\text{NTH}}{\text{NLP}} \times 100$
8. $\text{PERUND} = \dfrac{\text{UNID}}{\text{NLP}} \times 100$

where NLP = total no. of pixels in scene

NLIN = no. of lines

NPIX = no. of pixels per line

UNID = no. of identifiable pixels

ICAKNT(X) = no. of pixels in category X

NDUPIX = no. of DU pixels

NTH = no. of thresholded pixels

RECPCT(I) = percent pixels belonging to category I

PERDO = percent DO pixels

PERDU = percent DU pixels

PERTH = percent thresholded pixels

PERUND = percent identifiable pixels

NDOPIX = no. of DO pixels

Note: DO and DU pixels encountered are not updated to reflect new category. All other type pixels are updated.

DIAGNOSTICS Please refer to User's Manual

DATE ______________

TIME ______________

TITLE

RECOMPUTED CLASSIFICATION
SUMMARY REPORT

ORIGINAL PAGE IS
OF POOR QUALITY

SEGMENT ID ______________

ACQUISITION DATE(S) ______________

CLASSIFICATION CHANNELS ______________

NUMBER OF PIXELS IN SCENE ______________

NUMBER OF DU PIXELS ______________

NUMBER OF DO PIXELS ______________

NUMBER OF THRESHOLDED PIXELS ______________

PERCENTAGE OF THRESHOLDED PIXELS ______________%

NUMBER OF IDENTIFIABLE PIXELS ______________

NUMBER OF CATEGORY 'X' PIXELS ______________

Figure 3-21.— Recomputed Classification Summary Report.

CLASSIFICATION SUMMARY

CATEGORY	NUMBER OF PIXELS IN CATEGORY	PERCENTAGE OF IDENTIFIABLE PIXELS
1	29	----
2	38	----

Figure 3-22.- Classification Summary.

CLASSIFICATION MAP DISPLAY

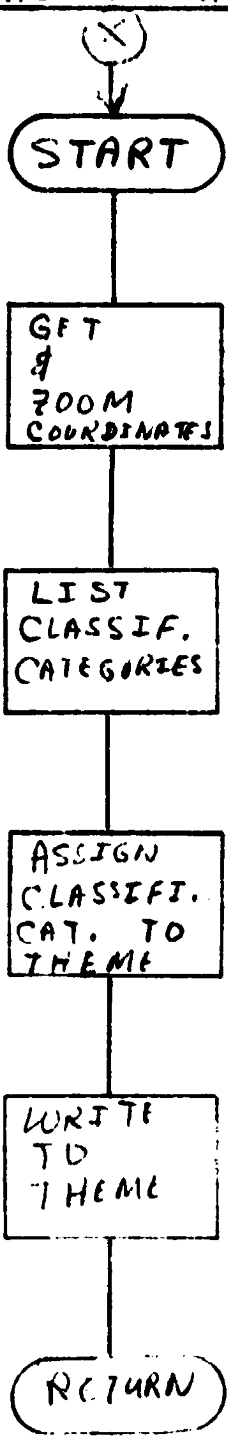
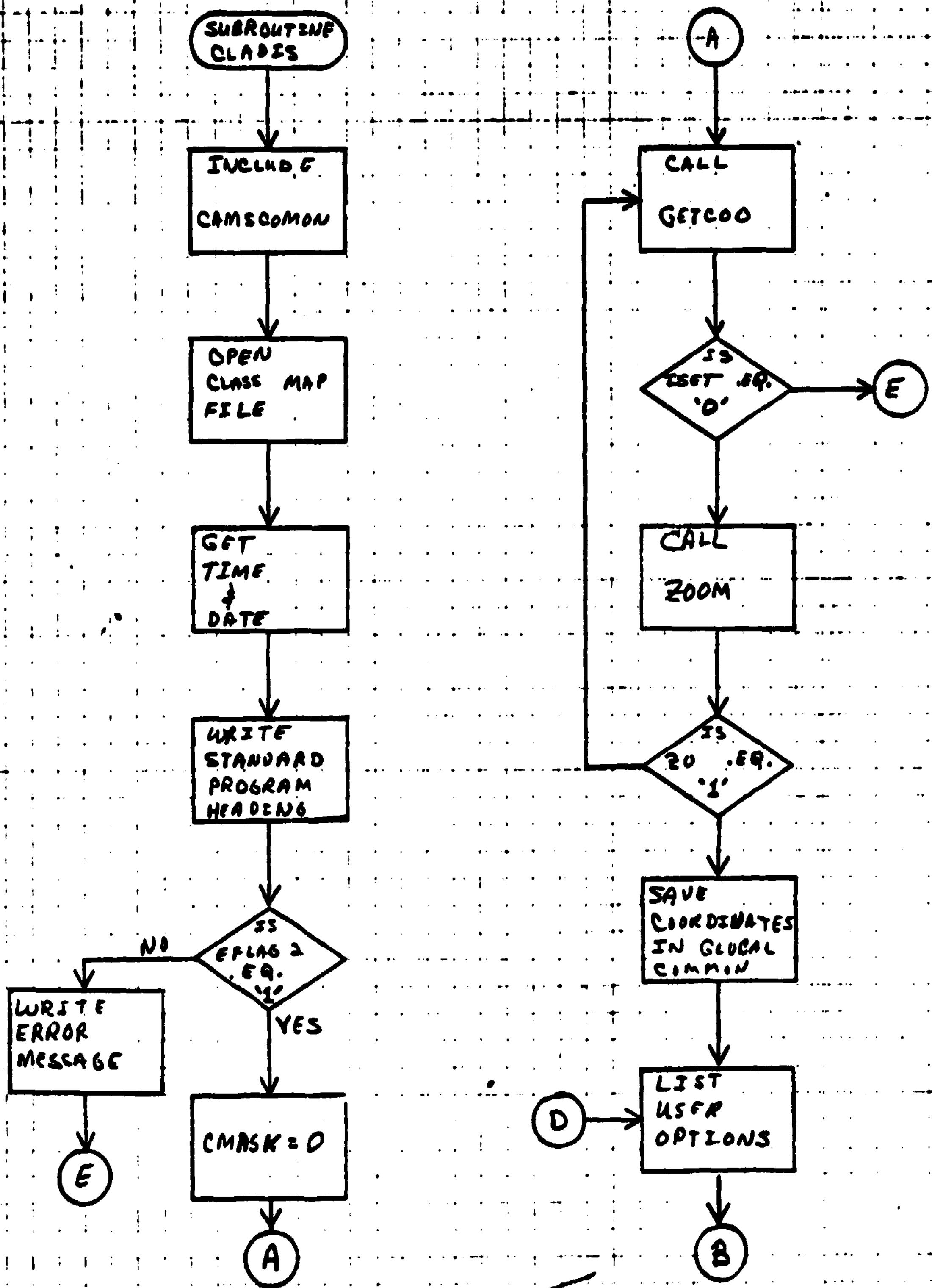


Figure 3-23

CLADIS.FTN

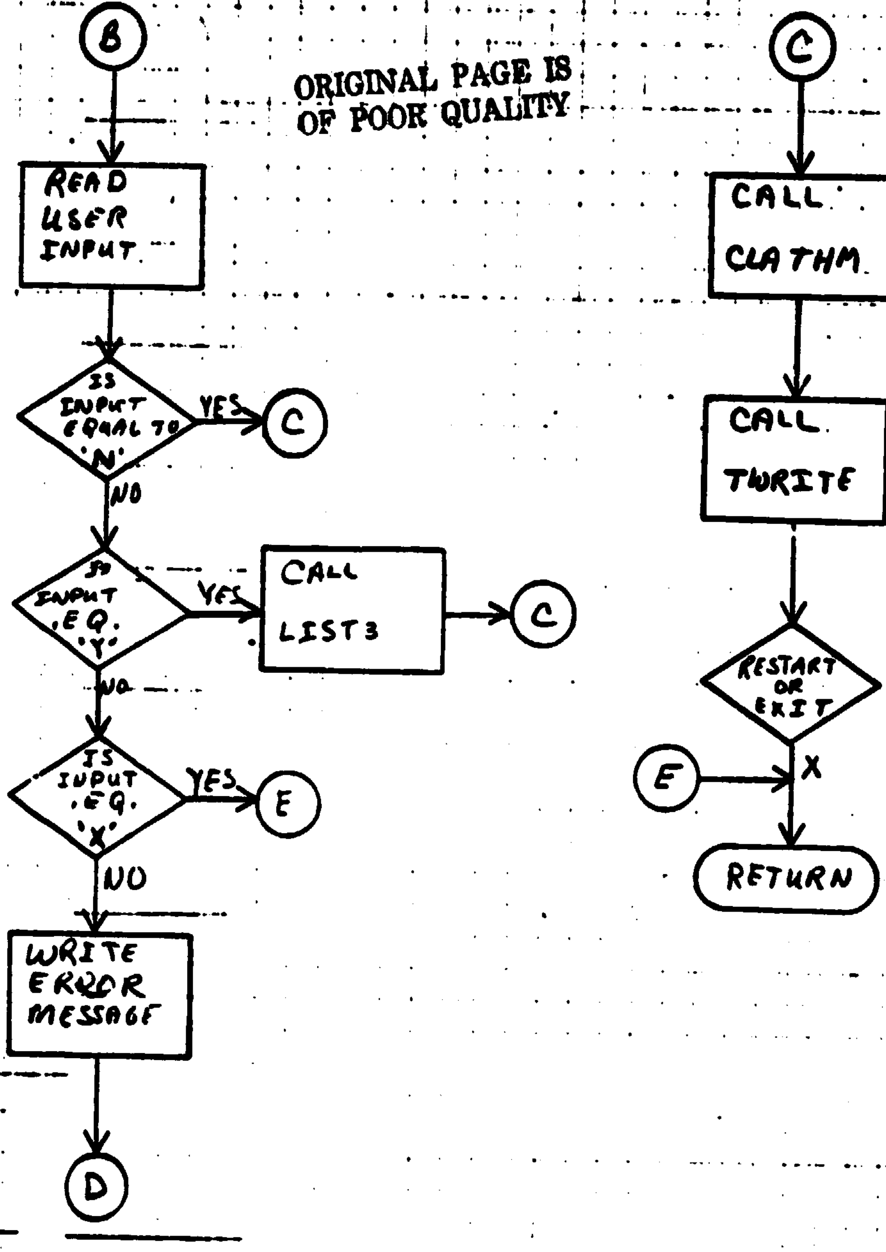
(CLASSIFICATION MAP DISPLAY)



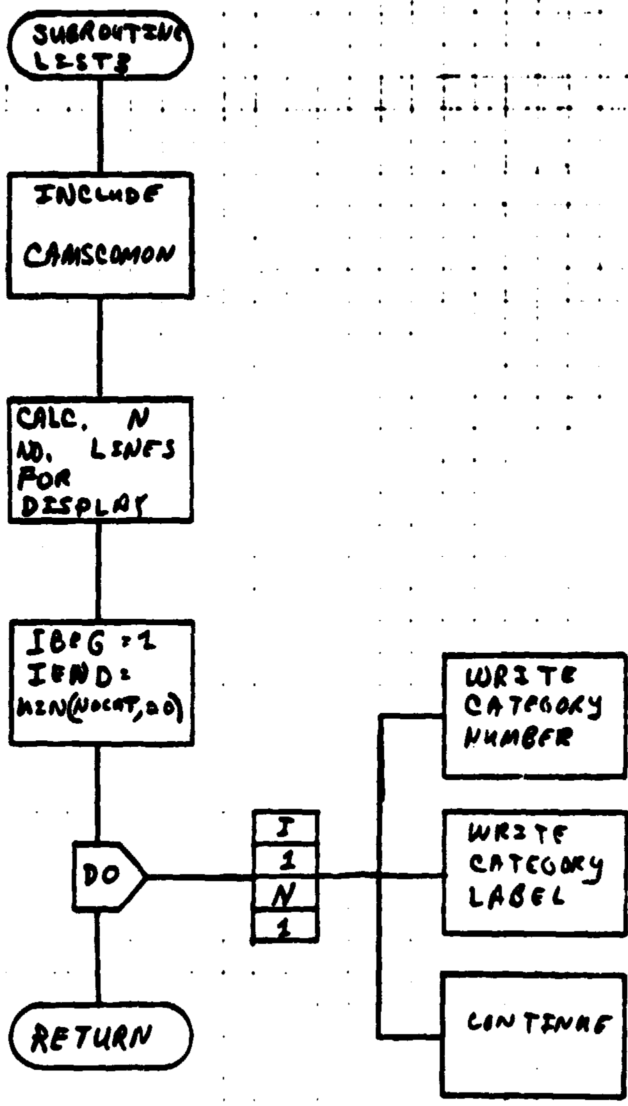
CLADIS.FTN

(CLASSIFICATION MAP DISPLAY)

ORIGINAL PAGE IS
OF POOR QUALITY

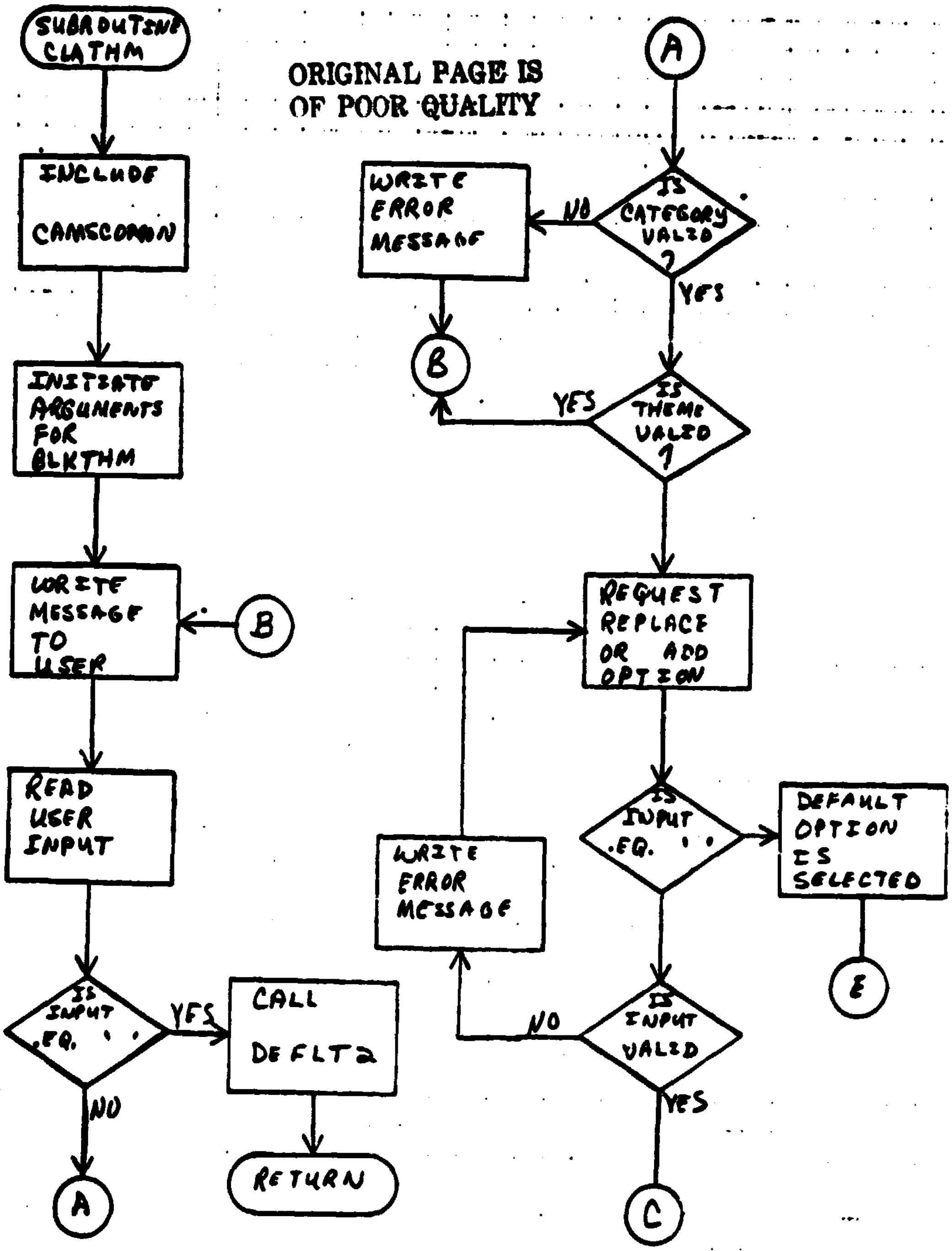


LISTS.FTN
(LISTS CATEGORIES)



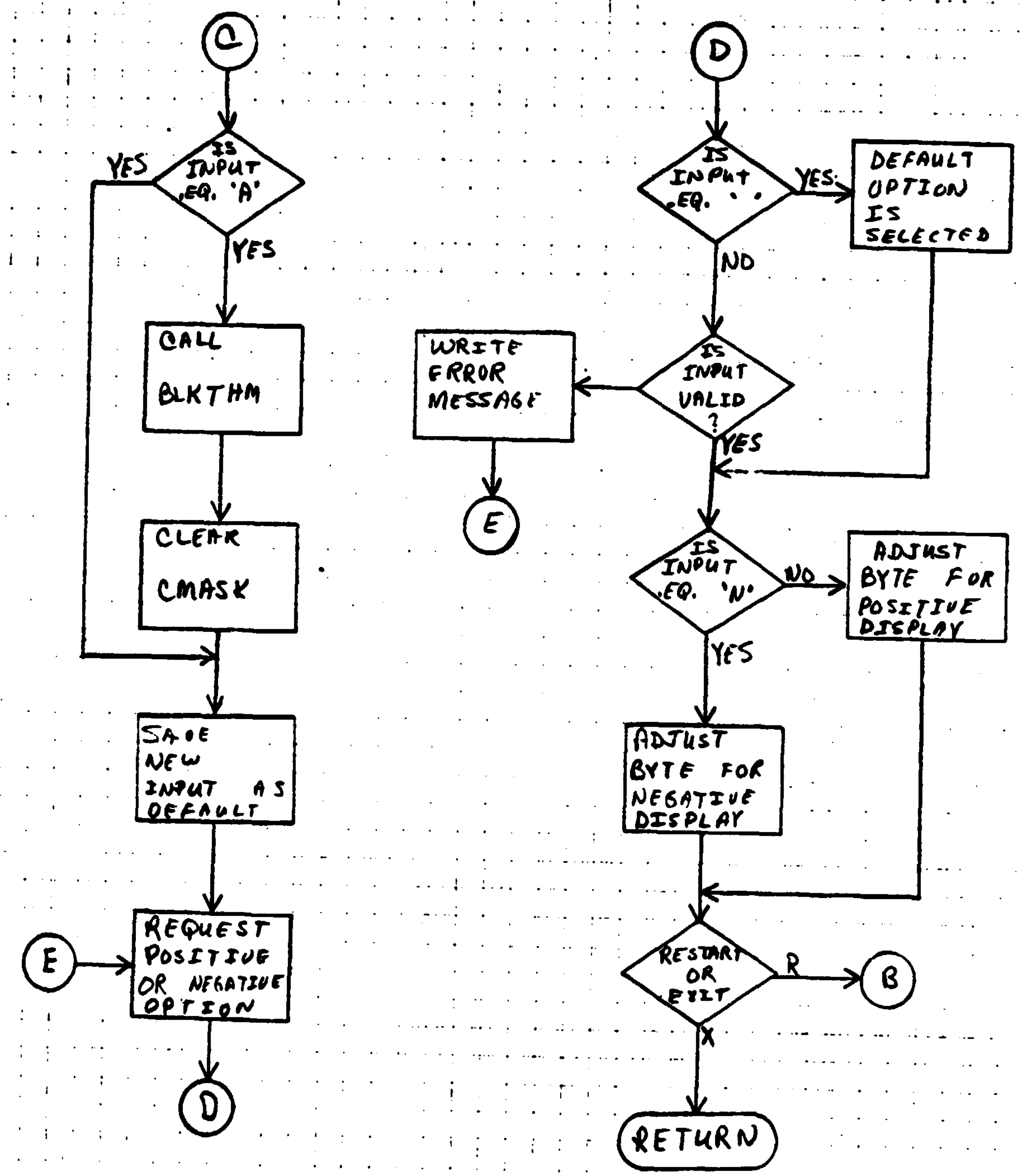
CLATHM.FTN

(ASSIGNS CATEGORIES TO THEMES)



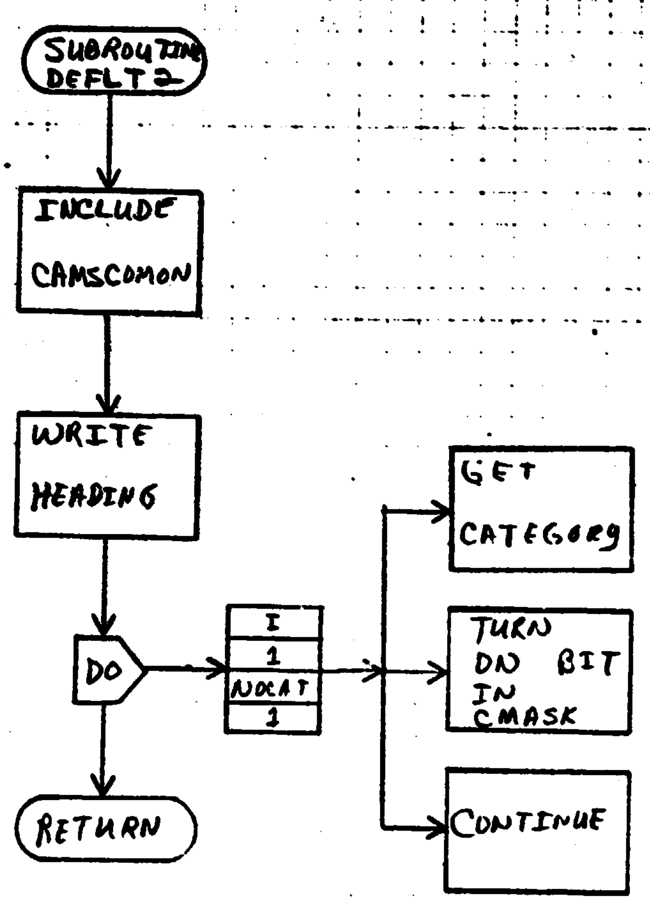
CLATHM.FTN

(ASSIGNS CLASSIFICATION CATEGORIES TO THEME)



DEFLT2.FTN

(ASSIGNS CLASSIFICATION CATEGORIES TO THEME)



3.5.2.9 Reports

The various interactive analysis processors in the system will in general generate intermediate results. T ese intermediate results will aid the user in making various processing decisions within a particular processor. Brief reports containing intermediate results will be displayed directly to the user (on the terminal screen or Gould printer) by the processor that generated these results. The reports module will display more lengthy and complete results obtained normally at the conclusion of segment analysis. Results for generation of these reports will exist on temporary disk files or in common areas. The user will have the option of viewing the results either on the terminal screen (from where they may be hardcopied), the Gould printer or line printer. Where possible, the user will be provided with the option of specifying the display of appropriate subsets of results. There will be foru major report modules. They are:

- Dot Data Report
- Bias Correction/Classification Summary Report
- Cluster Report
- DO/DU Field Definition Report

These modules are described in the following sections.

ORIGINAL PAGE IS
OF POOR QUALITY

3.5.2.9.1 Dot Data Report (DOTRPT) (Programmed and documented by Jane Huang)

DOT DATA REPORT (DOTRPT) prints out the dot selection report through subroutine DOTIN.

3.5.2.9.1.1 Linkages

A. IMALIB - FRONT, IBYTE, OUTPUT

B. FORTRAN or SYSTEM - SETEF, IDATE, TIME, CLOSE, CLOS$, ASSIGN AND OPEN$

C. Shared subroutine and utilities - DTCLIO, ELAPSE

D. Private subroutine - DOTIN

3.5.2.9.1.2 Interfaces

A. Common name COM2

Parameter	Updated by subroutines	Referenced by subroutines
ADATES(2,MAXACD)		
NOACQ		
CATNAM(MAXCAT)		
COILGR(MAXACD)		
NOCAT		

B. Common name COM4

Parameter	Updated by subroutine	Referenced by subroutine
NUMDOT	DOTIN	
DOTARY(NDOTS)	DOTIN	

C. Common name COM5

Parameter	Updated by subroutine	Referenced by subroutine
RANDOM(NDOTS)		
GRID(NDOTS)		
DLABEL(NDOTS)		

D. Working file name: DOT working file

3.5.2.9.1.3 Inputs

None

3.5.2.9.1.4 Outputs

A. Report - DOT DATA REPORT
B. Diagnostics
   END OF DATA REPORT!!!---Enter (R)estart or E(X)

3.5.2.9.1.5 Storage

DOTRPT occupies 60 blocks.

3.5.2.9.1.6 Description

GRENO(JJ1)=GRENSS(JJ1)-ARRY(JJ1)
where GRENO---green number
      GRENSS--- greeness
      ARRY---soil green number

3.5.2.9.1.7 Flow Charts

See figure 3-24

3.5.2.9.1.8 Subroutines

3.5.2.9.1.8.1 DOTIN

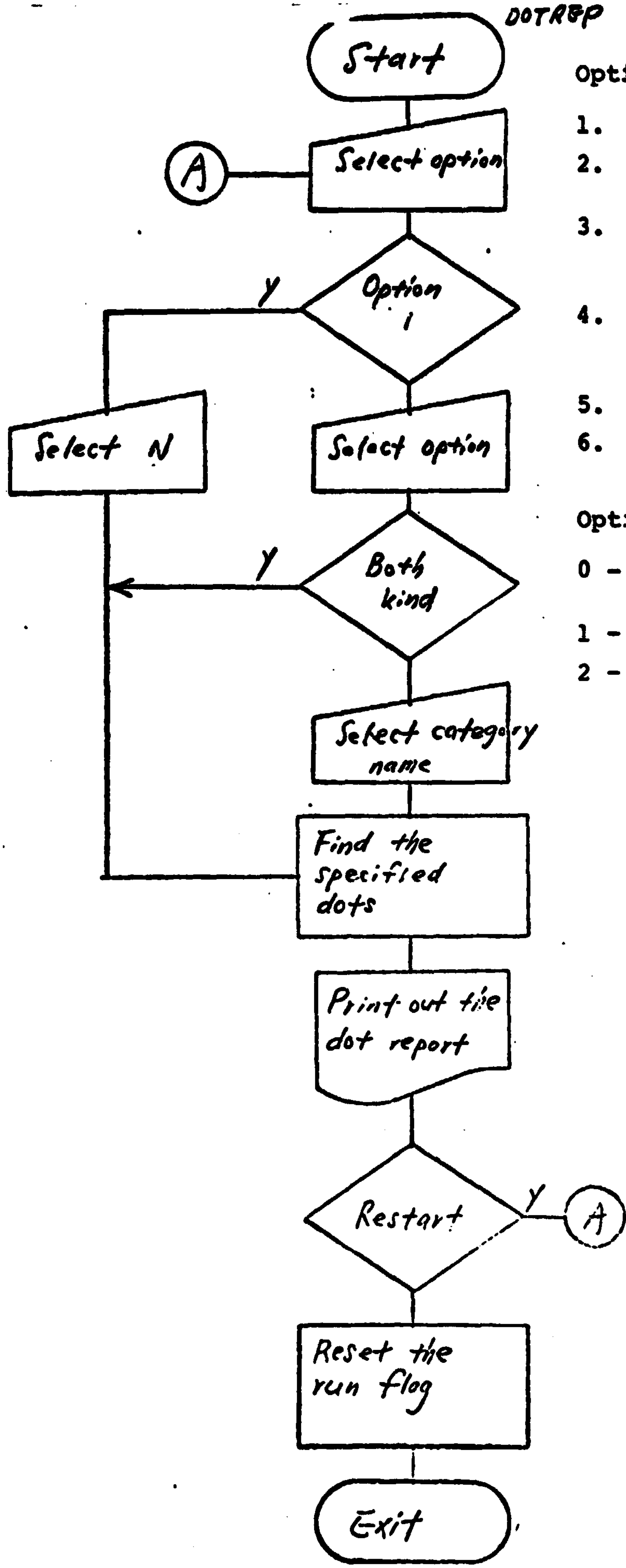
CALL DOTIN (IO,II)

Parameter	Dimension	Type	In/Out	Description
IO			In	Input output unit
II			Out	II=1---normal return II=2---Exit II=3---Backup

Description:

Subroutine DOTIN allows the analyst to select the dot group of interest according to one of the following dot selection rules:

1. All 209 dots
2. Unlabeled dots from the random sequence
3. Dots by type, analyst label and classifier label
4. Bias correction dots by the classification proportion
5. Starting dots
6. DO/DU dots



Options:

1. All 209 dots
2. Unlabeled dots from the random sequence
3. Dots by type, analyst label and classifier label
4. Bias correction dots by the classification proportion
5. Starting dots
6. DO/DU dots

Options:

0 - Both analyst and class labels

1 - Analyst label

2 - Class label

Figure 3-24.— Dot Data Report.

3.5.2.9.2 Bias Correction/Classification Summary (BIASCR) (Programmed and documented by Jane Huang)

The BIASCR program is the driver for the uncorrected proportions report and bias corrected computation report.

3.5.2.9.2.1 Linkages

A. IMALIB - FRONT, OUTPUT

B. FORTRAN - SETEF

C. Shared subroutines and utilities - ELAPSE

D. Private subroutines - BIASCP, UNBIAS, SELDOT, ROFF

3.5.2.9.2.2 Interfaces

A. Common name COM1

Parameter	Updated by subroutine	Referenced by subroutine
CATKNT		UNBIAS
NODO		UNBIAS
NODU		UNBIAS
NOTH		UNBIAS
DOTCLU(NDOTS)		UNBIAS

B. Common name COM2

Parameter	Updated by subroutine	Reference by subroutine
NOACQ		UNBIAS
ADATES		UNBIAS

C. Common name COM5

Parameter	Updated by subroutine	Reference by subroutine
DLABEL(NDOTS)		SELDOT
TYPE(NDOTS)		SELDOT

3.5.2.9.2.3 Input

None

3.5.2.9.2.4 Outputs

A. Title of report - The bias corrected and uncorrected report
B. Diagnostics - END OF UNCORRECTED PROPORTIONS REPORT!!!---the program will return to the main driver

3.5.2.9.2.5 Storage

BIASCR occupies 95 blocks.

3.5.2.9.2.6 Description

The driver computes % of DO and unidentified area, using the following equations:

DOPCT=NODO*100./TP1XL

where DOPCT--% of DO area

NODO--total DO pixels

TPIXL--total pixels in the segment in this case is 22932

UPCT=DUPCT+TPCT+XPCT

where UPCT--% of total unidentified area

DUPCT--% of unidentified area

TPCT--% of threshold area

XPCT--% of cloud or cloud shadow area

TPCT=NOTH*100./TP1XL

where NOTH--pixels in threshold area

DUPCT--NODU*100./TP1XL

DUPCT=NODU*100./TP1XL

where NODU--pixels in DU area

XPCT=CATKNT of category XX*100./(TP1XL-NODU-NOTH-XP1XL)

3.5.2.9.2.7 Flow Charts

See figure 3-28.

3.5.2.9.3.8 Subroutines

3.5.2.9.3.8.1 UNBIAS(II)

- Calling sequence

  Call UNBIAS(II)

Parameter	Dimension	Type	In/Out	Description
II		I	Out	In this case II always equals 3, which is normal return, backup, or exit

Description:

Subroutine UNIBAS prints out the report of uncorrected proportions.

A. SELDOT(-128,-128,-2,-128, DIDU)

Description:

Subroutine SELDOT has exactly the same function as subroutine DOTSEL.

B. ROFF(RNUM,ROFND)

Parameter	Dimension	Type	In/Out	Description
RNUM		R	In	Real number to be rounded off
ROFNO		R	Out	The rounded off real number

Description:

Subroutine ROFF can round off a real number to one significant decimal place.

DATE ________

TIME ________

CLASSIFIED CATEGORY PERCENTAGES

(UNCORRECTED)

SEGMENT ID ________

ACQUISITION DATE(S) ________

CLASSIFICATION CHANNELS ________

PIXEL CATEGORIES	PIXELS IN CATEGORY	% OF PIXELS IN CATEGORY
C_1(W)	--	--
C_2(S)	--	--
C_3(N)	--	--
...		
C_N(-)	--	--
X	--	--
DO	--	--
DU	--	--
THRESHOLD	--	--
X + DU + THRESHOLD	--	--

Figure 3-25.– Classified Category Percentages.

DATE ________

TIME ________

BIAS CORRECTION

ALPHA TABLE

SEGMENT ID ______

ACQUISITION DATE(S) ________

CLASSIFICATION CHANNELS ______

NUMBER OF TYPE 2 LABELLED DOTS ________

ANALYST LABELLED CATEGORY	CLASSIFIER LABELLED CATEGORY C_1	C_2	...	C_N
C_1	$\alpha_{K_1 K_1}$	$\alpha_{K_1 K_2}$	...	$\alpha_{K_1 K_{S+1}}$
C_2	$\alpha_{K_2 K_1}$	$\alpha_{K_2 K_2}$	...	$\alpha_{K_2 K_{S+1}}$
⋮				
C_N	$\alpha_{K_S K_1}$	$\alpha_{K_S K_2}$	...	$\alpha_{K_S K_{S+1}}$

Figure 3-26.— Bias Correction Alpha Table.

DATE ________

TIME ________

CORRECTED PROPORTIONS AND VARIANCES

SEGMENT ID ________

ACQUISITION DATE(S) ________

CLASSIFICATION CHANNELS ________

CATEGORY	UNCORRECTED PROPORTION	CORRECTED PROPORTION	ESTIMATED VARIANCE
K_1	--	--	--
K_2	--	--	--
⋮			
K_S	--	--	--
K_{S+1}	--	--	--

Figure 3-27.— Corrected Proportions and Variances.

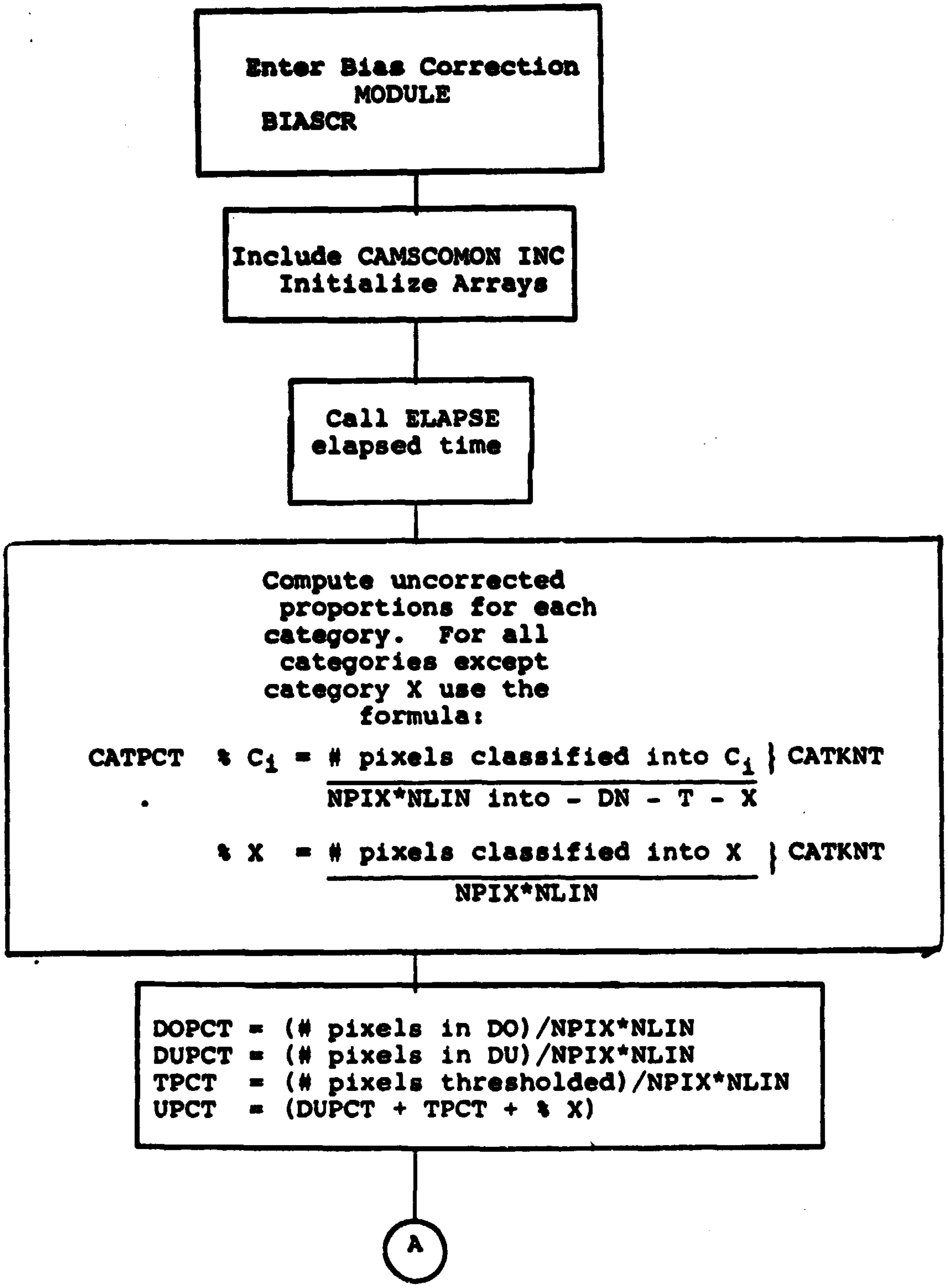


Figure 3-28.– Bias Correction/Classification Flow Chart

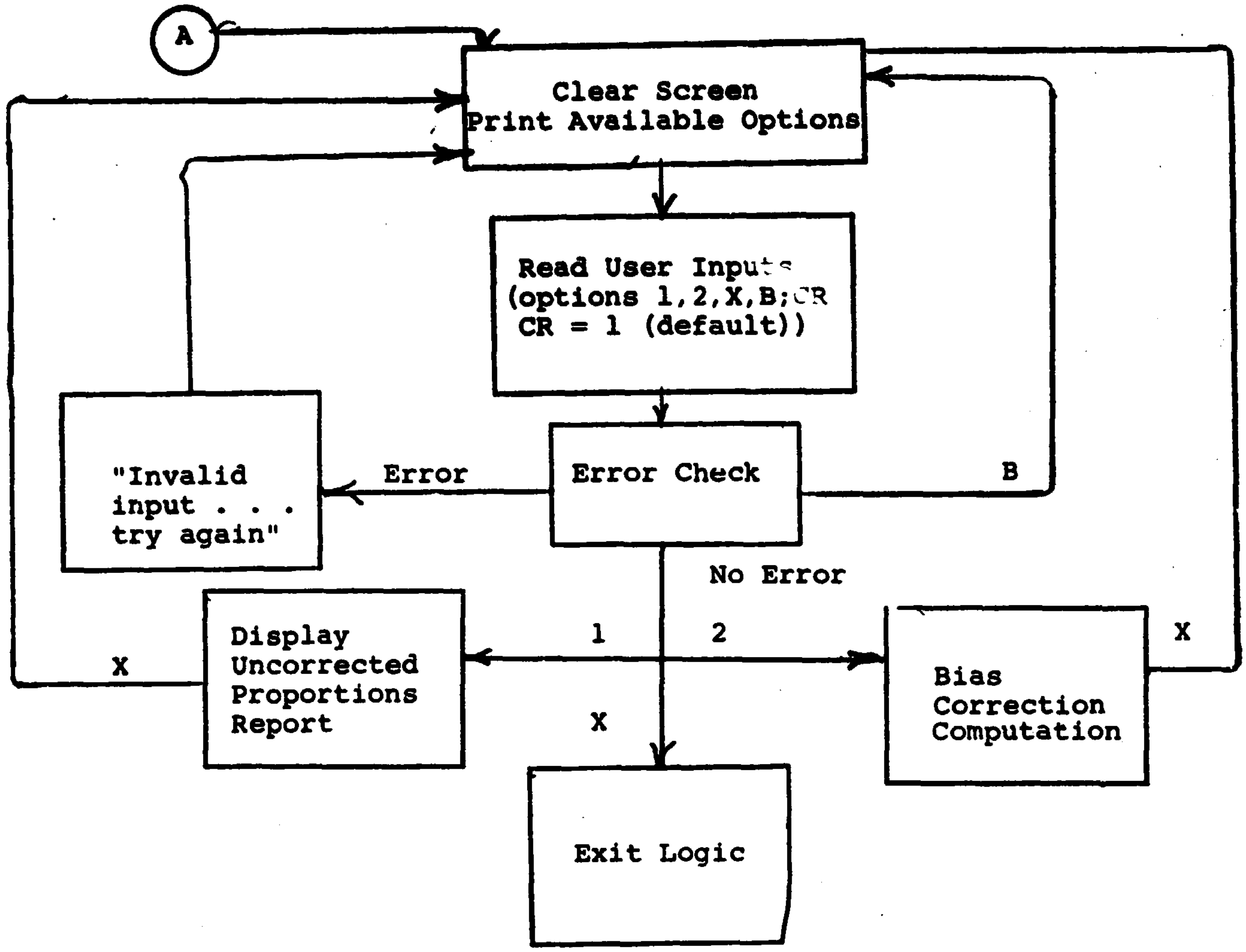


Figure 3-28.– (Continued)

3.5.2.9.3 Cluster Reports (CLURPT) (Programmed and documented by L. F. Robinson and E. L. Wilson)

This program is designed to provide the following report(s) upon request:

- Brief Cluster
- Cluster Mean/Standard Deviation
- Intercluster Distance
- Cluster Nearest Neighbor

3.5.2.9.3.1 Linkages

BRFCLU *CLOSØ CLUSNN **ELAPSE *FRONT *IDATE INCLDS,*INTFF MENSTD *OPENØ *OUTPUT *SETEF *TIME

*IMAGE 100 *IMALIB) LIBRARY system routines.

**CAMS HYBRID SYSTEM UTILITY ROUTINES.

3.5.2.9.3.2 Interfaces

A. CAMS Common Parameters

Parameter	Set by	Referenced by
MAXACC	INIT	BRFCLU, CLUSNN, INCLDS, MENSTD

COM 1

Parameter	Set by	Referenced by
NOSUB	——	BRFCLU, CLUSNN, INCLDS, MENSTD
ACDATE	——	" "
CHNVEC	——	" "
SUBCAT	——	" "

COM2

Parameter	Set by	Referenced by
ISEG	——	BRFCLU, CLUSNN, INCLDS, MENSTD
CATNAM	——	" "
NOCAT	——	" "

B. External File

1. 'DBO: [300,1]CLUSTATS.TMP'
2. 'DBP: [300,1]GLOBAL.TMP;1'

3.5.2.9.3.3 Inputs

User inputs to this program determine which of the cluster reports are to be displayed and upon what device, Terminal Screen, Line Printer or the Gould Printer. (See JSC-IMAGE-100 User's Manual, Part III CAMS Section 12.4)

3.5.2.9.3.4 Outputs

A. The outputs of this program consist solely of the following cluster reports: 1. Brief Cluster 2. Cluster Mean/Standard Deviation 3. Intercluster Distance 4. Cluster Nearest Neighbor. These reports may be routed to any of three display devices, the terminal CRT, line printer or the gould printer.

B. Diagnostics

1. "NO CLUSTER DATA AVAILABLE FOR SEGMENT ____"
   (program then exits)

2. Other diagnostics consist of a message being repeated if the reply to it is considered invalide.

3.5.2.9.3.5 Storage Requirements

CLURPT occupies 72 blocks.

3.5.2.9.3.6 Description

CLURPT clears the terminal screen, displays date/time, and checks to see if there is cluster data available for the segment being worked. If not, the diagnostic (3.5.2.9.3.4 BI) is displayed and the program exist. If data are available, the user is asked which report is desired and what device is to be used. Upon receiving valid replys to these querries, the subroutine responsible for generating the requested report is called. When the subroutine in question returns, a check is made to see if further reports are required; if not, the program exists. If other reports are desired, the subroutines necessary are called in order and then the program exists.

3.5.2.9.3.7 Flowchart

Presented in figure 3-29.

3.5.2.9.3.8 Subroutines

3.5.2.9.3.8.1 MENSTD

This subroutine produces the Mean/Standard Deviation cluster report.

- Calling sequence

  CALL MENSTD (DO1, DO2, DO3)

- Arguments

Parameter	Dimension	Type	In/Out	Description
DO1	1	Integer	In	Display device
DO2	1	Integer	In	Display device
DO3	1	Integer	In	Display device

- Description

MENSTD displays upon the output device specified the Mean/Standard Deviation cluster report, for the segment being worked on, from data contained in the file CLUSTATS.TMP.

If the terminal is the display device the report will be broken up into pages such that no data will be lost due to screen overload. After each page is full the message "ENTER "CR" TO PROCEED" is displayed and processing is halted until a "CR" is input.

3.5.2.9.3.8.2 INCLDS

This subroutine produces the Intercluster Distance cluster report.

- Calling sequence

  CALL INCLDS (DO1, DO2, DO3)

- Arguments

Parameter	Dimension	Type	In/Out	Description
DO1	1	Integer	In	Display device
DO2	1	Integer	In	Display device
DO3	1	Integer	In	Display device

- Description

INCLDS displays upon the output device specified the Intercluster Distance cluster report, for the segment being worked on, from the data in the file CLUSTATS.TMP.

If the terminal is the display device the report will be broken up into pages such that no data will be lost due to screen overload. After each page is full the message "ENTER "CR" TO PROCEED" is displayed and processing is halted until a "CR" is input.

3.5.2.9.3.8.3 BRFCLU

This subroutine produces the Brief Cluster cluster report.

- Calling sequence

  CALL BRFCLU (DO1, DO2, DO3)

- **Arguments**

Parameter	Dimension	Type	In/Out	Description
DO1	1	Integer	In	Display device
DO2	1	Integer	In	Display device
DO3	1	Integer	In	Display device

- **Description**

BRFCLU displays upon the output device specified the Brief Cluster cluster report, for the segment being worked on, from data contained in the file CLUSTATS.TMP.

If the terminal is the display device the report will be broken up into pages such that no data will be lost due to screen overload. After each page is full the message "ENTER "CR" TO PROCEED" is displayed and processing is halted until a "CR" is input.

3.5.2.9.3.8.4 CLUSNN

This subroutine produces the Cluster Nearest Neighbor cluster report.

- Calling sequence

  CALL CLUSNN (DO1, CO2, DO3)

- **Arguments**

Parameter	Dimension	Type	In/Out	Description
DO1	1	Integer	In	Display device
DO2	1	Integer	In	Display device
DO3	1	Integer	In	Display device

- **Description**

CLUSNN displays upon the output device specified the Cluster Nearest Neighbor cluster report, for the segment being worked on, from data contained in the file CLUSTATS.TMP.

If the terminal is the display device the report will be broken up into pages such that no data will be lost due to screen overload. After each page is full the message "ENTER "CR" TO PROCEED" is displayed and processing is halted until a "CR" is input.

DATE ______________

TIME ______________

BRIEF CLUSTER REPORT

CLUSTER REPORT FOR SEGMENT NUMBER ______________

ACQUISITION DATE(S) ______________

CLUSTERING CHANNELS ______________

LINE SKIP ______________ PIXEL SKIP ______________

CLUSTER	CATEGORY	POPULATION	PERCENTAGE OF SEGMENT	GREEN NUMBER
1	-	-	-	-
2	-	-	-	-
.	-	-	-	-
.	-	-	-	-
.	-	-	-	-
.	-	-	-	-
.	-	-	-	-
.	-	-	-	-
N	-	-	-	-

*DO/DU FIELDS INCLUDED/EXCLUDED FROM CLUSTERING.

Figure 3-29.— Brief Cluster Report.

DATE ______________

TIME ______________

CLUSTER MEAN/STANDARD DEVIATION REPORT

SEGMENT ID ______________

ACQUISITION DATE(S) ______________

	CLUSTER '1'		CLUSTER'N'	
CHANNEL	MEAN	ST. DEV.	MEAN	ST. DEV.
1	XXX.XX	XXX.XX	XXX.XX	XXX.XX
2	XXX.XX	XXX.XX	XXX.XX	XXX.XX
.	XXX.XX	XXX.XX	XXX.XX	XXX.XX
.	XXX.XX	XXX.XX	XXX.XX	XXX.XX
.	XXX.XX	XXX.XX	XXX.XX	XXX.XX
.	XXX.XX	XXX.XX	XXX.XX	XXX.XX
.	XXX.XX	XXX.XX	XXX.XX	XXX.XX
N	XXX.XX	XXX.XX	XXX.XX	XXX.XX

Figure 3-30.– Cluster Mean/Standard Deviation Report.

DATE ______________

TIME ______________

INTERCLUSTER DISTANCE REPORT

SEGMENT ID ______________

ACQUISITION DATE(S) ______________

CLUSTERING CHANNELS ______________

CATEGORY/ CLUSTER	W 1	NW 2	SW 3	W 4	- - - - - - - - - - - - - - - - - - - -	W N
W 1	0					
NW 2	X	0				
SW 3	X	X	0			
W 4	X	X	X	0		
.	.	.	.	.		
.	.	.	.	.		
.	.	.	.	.		
.	.	.	.	.		
W N	X	X	X	X	- - - - - - - - - -	0

Figure 3-31.– Intercluster Distance Report.

DATE ______________

TIME ______________

DOTS/CLUSTER REPORT

SEGMENT ID ______________

ACQUISITION DATE(S) ______________

CLUSTERING CHANNELS ______________

DOT GRID NUMBER	CLUSTER ASSIGNED	ANALYST LABEL	CLUSTER LABEL
1	9	W	NW
2	10	NW	NW
.	.	.	.
.	.	.	.
.	.	.	.
209	3	W	W

Figure 3-32.— Dots/Cluster Report.

DATE ____________

TIME ____________

CLUSTER NEAREST NEIGHBOR REPORT

SEGMENT ID ____________

ACQUISITION DATE(S) ____________

CLUSTERING CHANNELS ____________

	NEAREST NEIGHBOR DISTANCE						
		1					
CLUSTER	DOT	LABEL	DIST	2/NW	3/W	- - - - - -	K/SW
1	xxx	xx	xxxx				
2		x		x	x	- - - - - -	x
3		x		x	x	- - - - - -	x
		x		x	x	- - - - - -	x
		x		x	x	- - - - - -	x
N		xx		x	x	- - - - - -	x
		x		x	x	- - - - - -	x

Figure 3-33.– Cluster Nearest Neighbor Report.

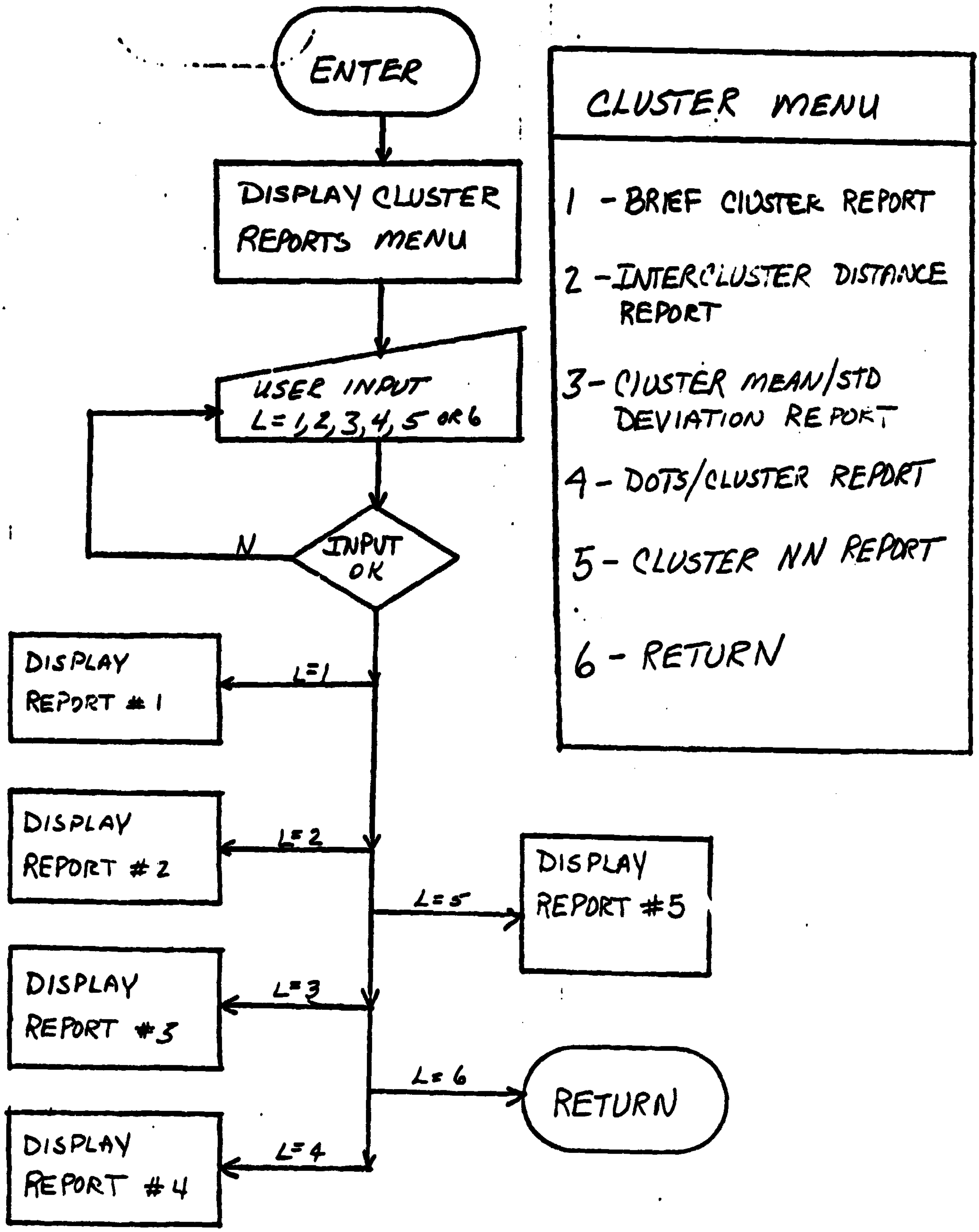


Figure 3-34.– Cluster Reports.

3.5.2.9.4 Field Definition Report (FLDRPT)

FLDRPT - Field definition report - this routine will generate an output report containing all data in the FIELDS.TMP file. The report can be output on any of the following: line printer, Gould printer or terminal.

3.5.2.9.4.1 Linkages

A. ASSIGN, CLOS$, ELAPSE, IBYTE, OPEN$, OUTPUT, SETEF

B. ?

C. GSGDPH

D. N/A

E. N/A

3.5.2.9.4.2 Interfaces

A. N/A

B. N/A

C. Working files - [300,1] FIELDS.TMP - input only

3.5.2.9.4.3 Inputs

A. N/A

B. N/A

C. KEY-IN - T for terminal, G for Gould, or L for line printer to direct report to proper device.

D. N/A

3.5.2.9.4.4 Outputs

A. Reports - Field Definition Report - Reference??

B. ?

C. ?

3.5.2.9.4.5 Storage requirement

FLDRPT occupies 35 blocks.

3.5.2.9.4.6 Description:

The routine queries the terminal for which device (Gould Printer, line printer or line printer) to output the report on. Next it opens the FIELDS.TMP file and using the data in the first records sets up a DO-loop to handle all the data. A heading is printed to the output device; the data is read, formatted and output to the appropriate device. When the bottom of the page for the line printer and Gould printer is reached, a new header is printed and additional data output until all data is processed. Then an "END of report" is output to the terminal followed by "Restart or EXIT?" which is self explanatory. For the terminal output when the bottom of the page is reached, the message "ENTER F for page forward, R for Restart or X for exit" is output. If "F" is input, a new header is output and output continues to line bottom of the page or end of data as above. Otherwise the routine exits or restarts.

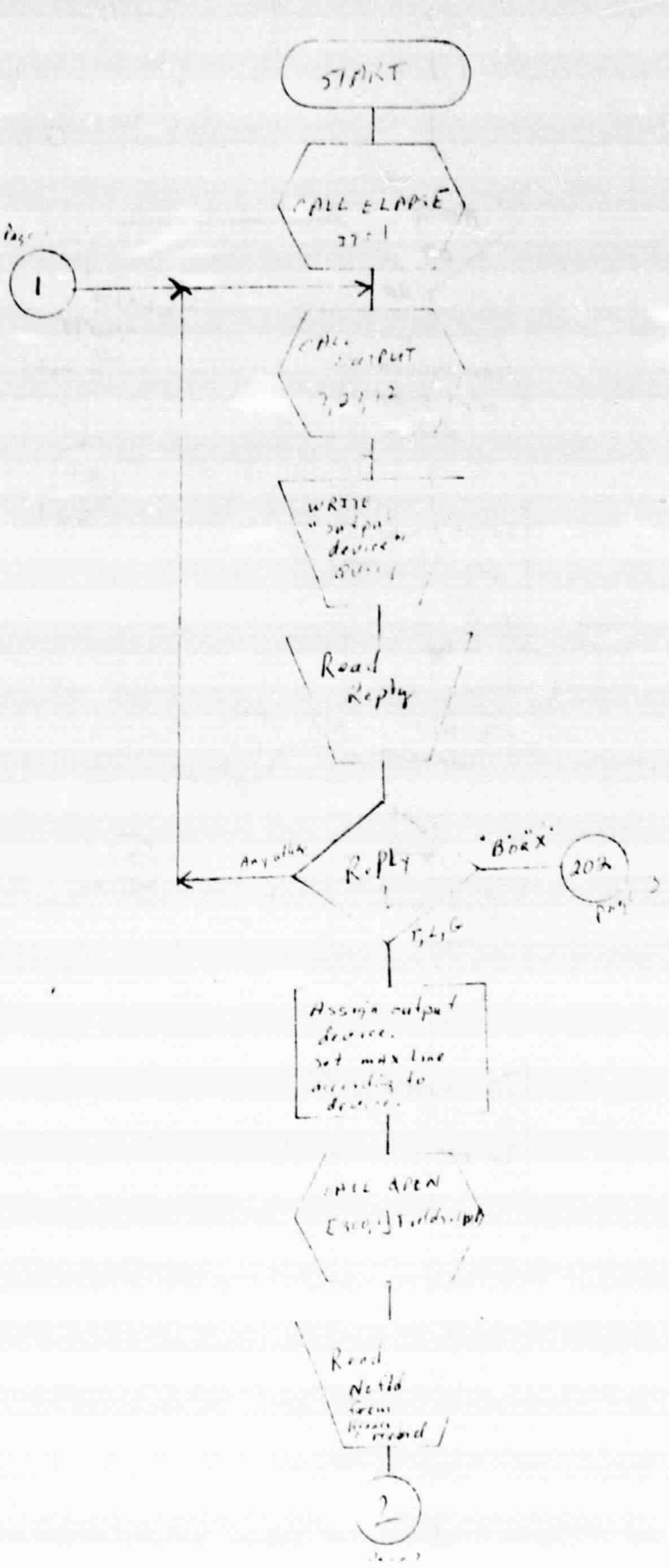
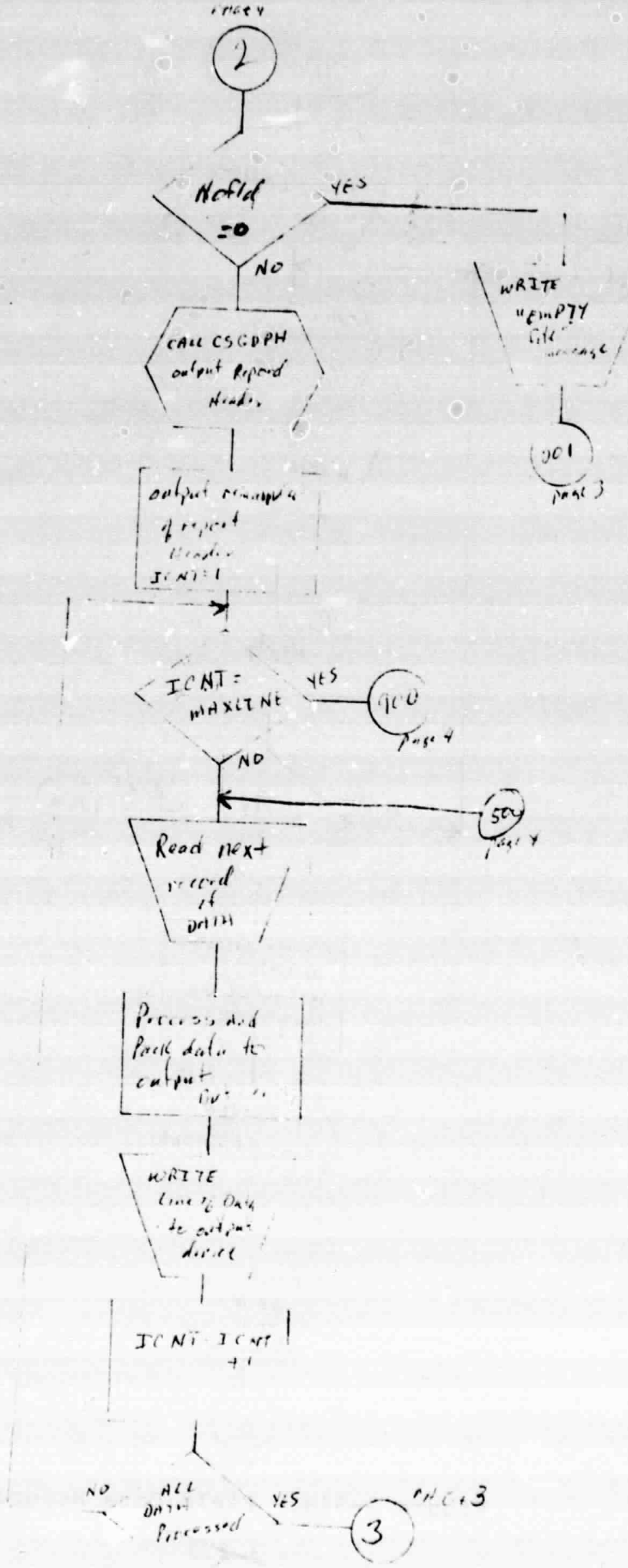


Figure 3-34a.– Field Data Report.



ORIGINAL PAGE IS OF POOR QUALITY

Figure 3-34a.- Continued

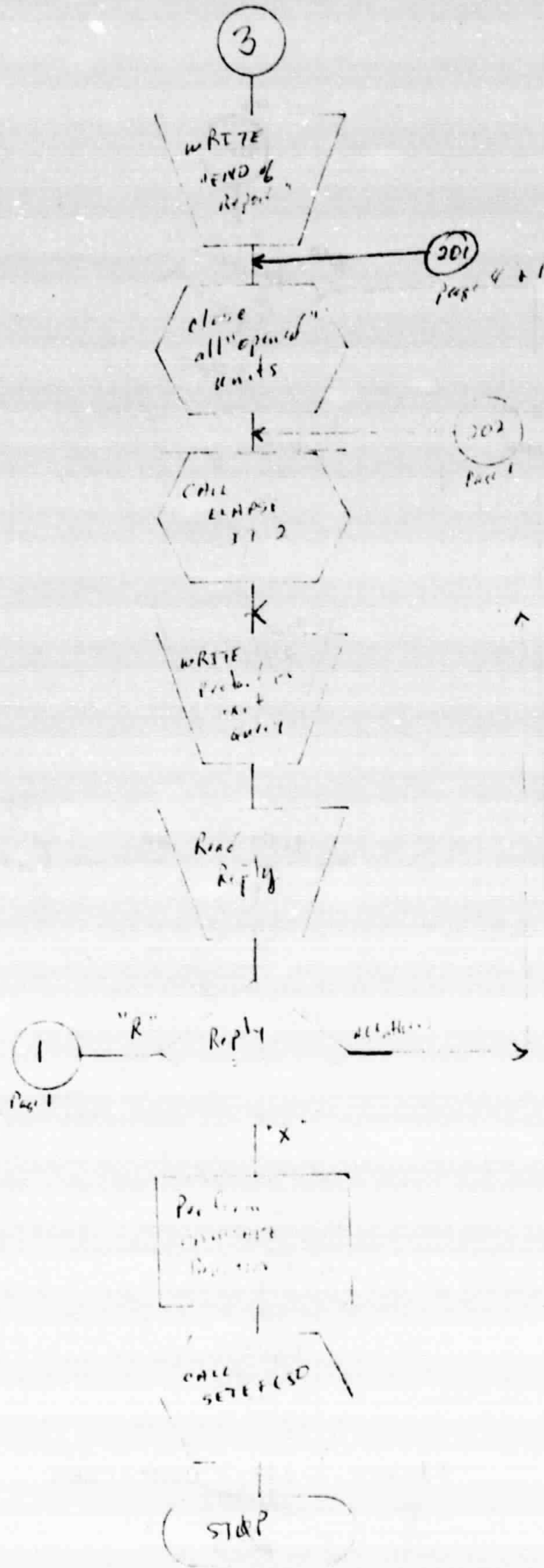


Figure 3-34a. - Continued

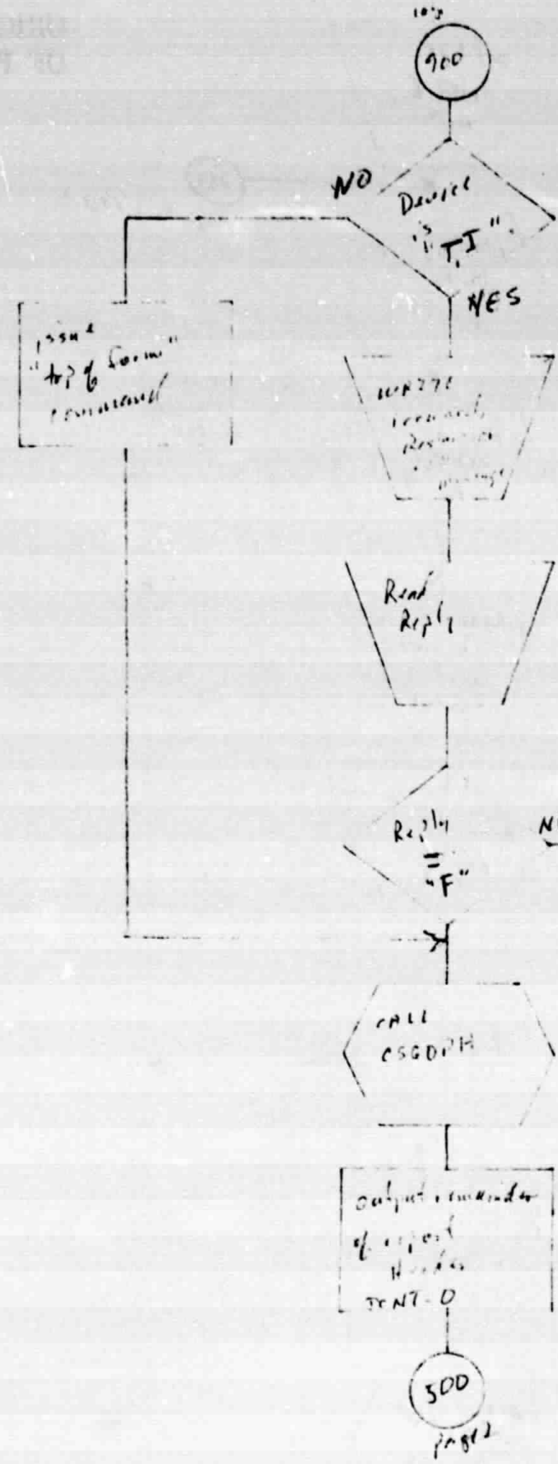


Figure 3-34a.- Continued

3.5.2.10 Permanent Data Base Update (PRMUPD) (Programmed and Documented by K. Pattison)

This program transfers the specified set of files currently residing in the temporary file area on the System Disk (DBO) to the Permanent Data Base Disk (DB2). If updates have been made to the Dot Data File and/or the Fields File the files are offloaded in card format; if not the option is provided to force an offload. The acquisition delete flag is set at either the user's discretion or by system requirement.

3.5.2.10.1 Linkage

A. IMALIB

B. FORTRAN

C. Shared Subroutines

1. ELAPSE

D. Private Subroutines

1. SUBSTR
2. UNLDOT
3. AQLFST
4. CLASAV
5. DOTSAV
6. FLDSAV
7. JULIAN
8. RPTGEN
9. STASAV
10. FLDOFF
11. RDFLD

3.5.2.10.2 Interfaces

A. COM1

Parameter	Updated by Subroutine	Referenced by Subroutine

B. COM2

Parameter	Updated by Subroutine	Referenced by Subroutine
DELFLG	AQLPST	AQLFST
NOACQ		AQLFST
ADATES		AQLFST
DOTDAY	DOTSAV	RPTGEN
NSTART	DOTSAV	DOTSAV
FLDDAY	FLDSAV	RPTGEN
IMDATE	CLASAV STASAV	RPTGEN
NSTART		
PDATEL	CLASAV	RPTGEN RPTGEN
TDATE1		CLASAV
PDAIE2	CLASAV	RPTGEN RPTGEN
TDATE2		CLASAV
PDATE3	STASAV	RPTGEN RPTGEN
TDATE3		STASAV
NOCAT		STASAV

C. COM3

Parameter	Updated by Subroutine	Referenced by Subroutine
PFLAG		RPTGEN CLASAV MAIN STASAV
EFLAG1		MAIN
EFLAG2		MAIN
EFLAG3		MAIN
EFLAG5		MAIN
UFLAG1		CLASAV MAIN RPTGEN
UFLAG2	MAIN	RPTGEN
UFLAG3	MAIN	MAIN RPTGEN

Parameter	Updated by Subroutine	Referenced by Subroutine
UFLAG4		STASAV MAIN RPTGEN

D. COM5

Parameter	Updated by Subroutine	Referenced by Subroutine
TYPE		DOTSAV
DLABEL		DOTSAV

E. Working File Name(s)

1. DIRFILE.DAT
2. DOTS.TMP
3. XXXXDOOTS.DAT
4. FIELDS.TMP
5. XXXXFIELD.DAT
6. CLASSMAP.TMP
7. XXXXPCLAS.MAP
8. XXXXTCLAS.MAP
9. CLUSTERMP.TMP
10. XXXXTCLUS.MAP
11. XXXXPCLUS.MAP
12. CLUSTATS.TMP
13. XXXXPSTAT.DAT
14. XXXXTSTAT.DAT
15. DSKTBL.DAT

3.5.2.10.3 Inputs

Key-In (reference User's Manual or other attachment to this package.)

3.5.2.10.4 Outputs

A. Reports-Permanent Data Base Update Transaction Report (Reference User's Manual for details.)

B. Diagnostics (Should be in User's Manual - redundant here.)

3.5.2.10.5 Storage Requirements

PRMUPD uses 83 blocks.

3.5.2.10.6 Description

The Permanent Data Base Update program uses a single file format to establish file names, making minimal character changes as file names change. The specific files are transferred in the following manner:

- Dot Data File:
  - If there have been updates to this file the transfer is made and an offload is forced.
  - If there have been no updates the analyst may force an offload.
- Fields File
  - If a Fields File does not exist, this processing area is bypassed.
  - If there have been updates to this file the transfer is made and an offload is forced.
  - If there have been no updates the analyst may force an offload.
- Classification and cluster maps
  - If either of these files do not exist this area is bypassed.
  - If at initialization time permanent files were requested and during processing they have been modified, the updated files are saved.
  - If temporary files were requested they are saved as permanent files irregardless of whether they were modified or not.
- Statistics File
  - If a statistics file does not exist, this processing area

is bypassed.

- If at initialization time permanent files were requested and during processing the statistics file was modified, the updated file is saved.
- If temporary files were requested the statistics file is saved as a permanent file irregardless of whether it was modified or not.

When all files have transferred successfully the analyst is given the opportunity to set the delete flag if the number of acquisitions for this segment exceeds three.

3.5.2.10.7 Flow Chart

Flow chart is presented in Figure 3-35.

3.5.2.10.8 Subroutines

3.5.2.10.8.1 SUBSTR

To move a substring of A into a substring of B

- Calling sequence
  CALL SUBSTR (A.I,N,B,J,M)
- Arguments

Parameter	Dimension	Type	In/Out	Description
A	A(1)		In	String name containing data to be moved
I		Ix2	In	Starting position
N		Ix2	In	Length
B	B(1)		Out	Receiving string name
J		Ix2	Out	Starting Position
M		Ix2	Out	Length

- Description
  The contents of string name 'A' starting at 'I' for 'N' characters is moved to string name 'B' starting at 'O' for

'M' characters. If string 'B' is shorter than string 'A', characters on the right will be truncated. If string 'B' is longer, it will be blank filled on the right.

3.5.2.10.8.2

To reformat file DOTS.TMP on the System Disk (DBO) into 80 column card images for subsequent processing by system subroutine CARD-OUT card eventual card punching in the PMIS DAS.

- Calling sequence
  CALL UNLDOT
- Arguments are passed thru Global Common
- Description
  The 'SEGMENT', 'TYPE', and 'START' cards are formulated and written to file 'CARD.DAT' (which will subsequently be processed for off line card punching). The individual 'DOT' card formats are built with the associated labels and the card images written to the output file.

3.5.2.10.8.3 AQLFST

To set the acquisition delete flag upon either the analyst request or when forced to by the system.

- Calling sequence
  CALL AQLFST
- Arguments are passed thru Global Common
- Description
  If the acquisition delete flag is already set or the number of acquisitions for the segment is thru or less an immediate exit is effected. If the number of acquisitions is four or five the analyst is given a choice of setting the flag or delaying his decision. When the number of acquisitions reaches six and the flag is not set, the analyst is forced to set the flag. The only other alternative is to abort the

run. When setting the delete flag the analyst is shown which acquisition dates are eligible for deletion.

3.5.2.10.8.4 CLASAV

To transfer the classification and cluster maps residing in the temporary file on the system disk to the permanent Data Base Disk properly identified.

- Calling sequence
  CALL CLASAV
- Arguments are passed thru Global Common
- Description
  The cluster map is transferred first. If temporary files were originally loaded the permanent date in the Directory File is set from the ERIPS date. The classification map is transferred next. If temporary files were originally loaded the permanent date in the Directory File is set from the ERIPS date. The Julian date for the last I-100 work on this segment is set into the Directory File.

3.5.2.10.8.5 DOTSAV

To transfer the Dot Data File residing in the temporary file on the system disk to the permanent data base file properly identified.

- Calling sequence
  CALL DOTSAV
- Arguments are passed thru Global Common
- Description
  The Dot Data File is transferred from the temporary file on the system disk to the permanent data base disk. The update Julian date is saved in the Directory File. The analyst is given the option of forcing a starting value vector different from the one displayed. The offload (UNLDOT) program is called to perform its assigned tasks. (See 3.5.2.10.2.)

3.5.2.10.8.6 FLOSAV

To transfer the FIELDS File residing in the temporary file on the system disk to the permanent data base file properly identified.

- Calling sequence
  CALL FLOSAV
- Arguments are passed thru Global Common
- Description
  The FIELDS File is transferred from the temporary file on the system disk to the permanent data base disk. The update Julian date is saved in the Directory File. The offload (FLDOFF) program is call to perform its assigned tasks. (See 3.5.2.10.10.)

3.5.2.10.8.7 JULIAN

To convert the current year, month, day, values to a Julian date.

- Calling sequence
  CALL JULIAN (YR. MO, DY, JULIO)
- Arguments

Parameter	Dimension	Type	In/Out	Description
YR		Ix2	In	Current year (ASCH format)
MO		Ix2	In	Current month (ASCH format)
DAY		Ix2	In	Current Day (ASCH format)
JULIO		Ix2	Out	Converted Julian date

- Description
  A preset table of Julian date values is accessed and the retrieval value is added to the current day. If the current year is a leap year the necessary Julian date adjustment is made.

3.5.2.10.8.8 RPTGEN

To output the Permanent Data Base Update Transaction Report

- Calling sequence

  CALL RPTGEN (SEGNO, I, J, K, FILTYP, A)

- Arguments

Parameter	Dimension	Type	In/Out	Description
SEGNO	(2)		In	Segment number being processed
I		Ix2	In	Current month
J		Ix2	In	Current day
K		Ix2	In	Current year
FILTYP	(23)		In	Name of file going to Data Base
A	(4)		In	Current time

- Description

  The transaction report header information (titles, date, time, segment number and disk number) are output to the Gould printer. Based on the existance of, the update status, and the status of intialized files (temporary or permanent) the files that were actually transferred to the permanent data base are listed with their associated Julian date.

3.5.2.10.8.9 STASAV

To transfer the segmented statistics file residing in Common Block #1 and in the temporary file on the system disk to the permanent data base file properly identified.

- Calling sequence

  CALL STASAV

- Arguments are passed thru Global Common
- Description

  Common Block #1, which contains the statistical information in the header record for this segment, is written to the permanent data base disk. The additional statistics file data is transferred from the temporary file on the system disk to the permanent data base disk. The updated Julian date is saved in the Directory File. If temporary files were loaded at initialization time the permanent date in the Directory File is set from the ERIPS date.

3.5.2.10.8.10 FLDOFF

To reformat file DOTS.TMP on the System Disk (DBO) into 80 column cards, processable by the ERIPS system, for subsequent input to system subroutine CARD-OUT and eventual card punching by the PMIS DAS.

- Calling sequence
  CALL FLDOFF
- Arguments are passed thru Global Common
- Description
  The header cards are formulated and written to the output file being constructed for CARD-OUT (XXXXFLDOF.DAT). The individual field card (LINENO and PIXELNO) card formats are built with the associated vertice number and the card images written to the output file. Every pair of vertices is checked to determine if it is also a specified 'DOT' and if so the type of field ('DO' or 'DU') is indicated in the Dot Data File.

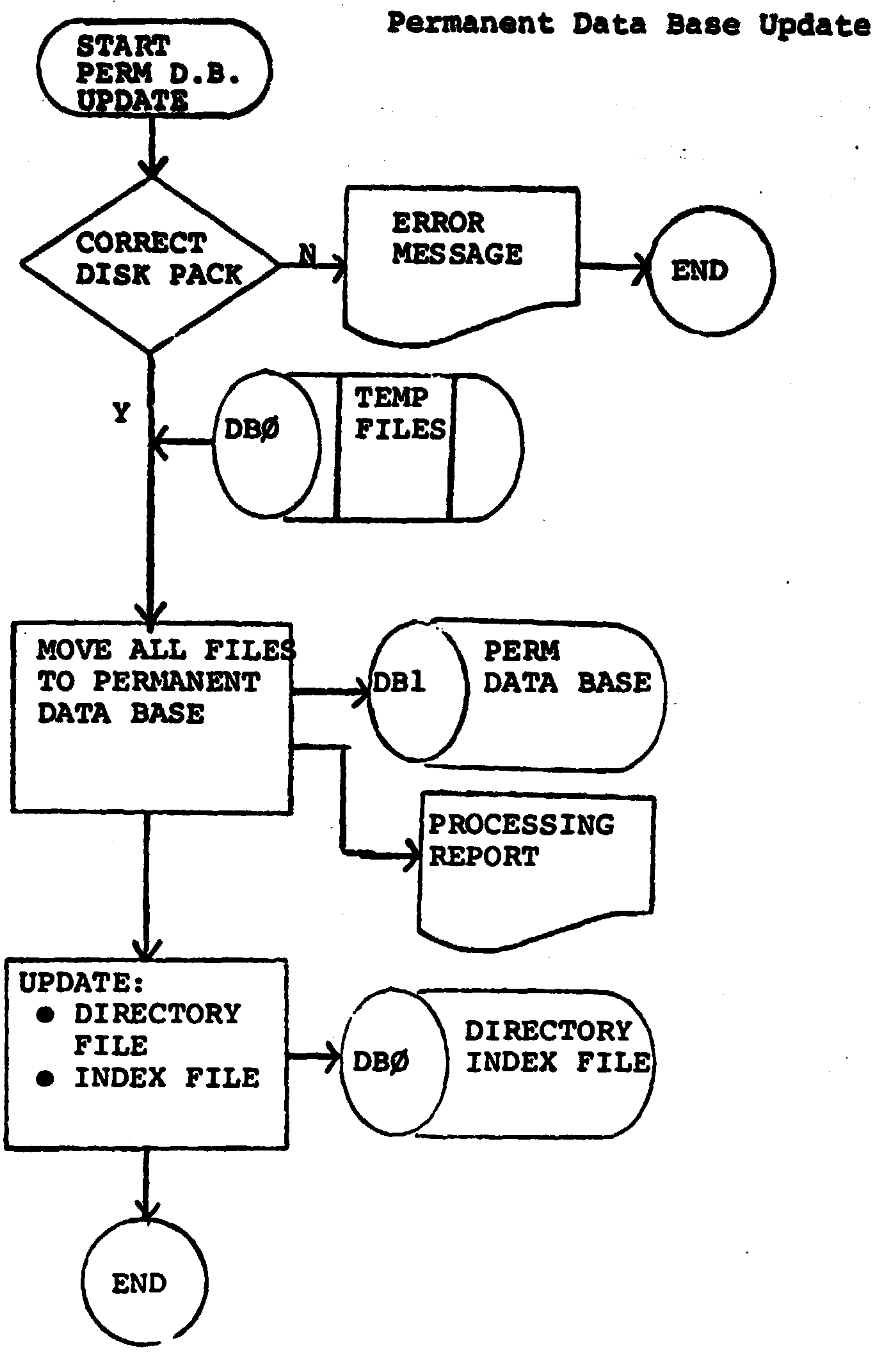


Figure 3-35.– Permanent Data Base Update.

APPENDIX A

CODING STANDARDS

- All coding will be done in FORTAN IV and compiled with the F4P compiler. (Exceptions may be granted if assembly language will make a considerable time difference.)

- Upon entry, each program will perform the following functions:

  1. Interactive programs will clear the Tektronic screen - CALL OUTPUT (27,12).
  2. Print the program name and version/date. See Example 3.
  3. Print the current date and time. See Example 3.
  4. Initiate timing.

- Each program should have a comment header giving the name of the program or subroutine and its primary function.

  Example:

```
CCC ------------------------------------------------------------------- C
C                                                                       C
C           IMAGE  DATA DISPLAY - IMDIS.FTN(MAIN)                       C
C                                                                       C
CCC ------------------------------------------------------------------- C
```

  or

```
                       SUBROUTINE RIMDA

C                                                                       C
CCC ------------------------------------------------------------------- C
C                                                                       C
C                                                                       C
C                                                                       C
C                                                                       C
C --------------------------------------------------------------------- C
```

- Comments should be present for every local major function.

- Queries: (See examples 1 and 2)

  1. Brief and meaningful.
  2. Print default condition and allow answer on same line. Use '$' as carriage control in the format statement to keep the cursor on the same line.

3. CR ' ' accept default (if no default, repeat query).

4. 'B' back-up to last query (not allowed on 1st query or at program exit logic).

5. 'X' branch to program exit logic.

6. Read responses with alphanumeric format and use IMALIB routines 'FRONT', 'INTFF', etc. to interpret answers. See examples 1 and 2.

7. Test on alphanumeric characters rather than octal numbers (i.e., 'Y', 'N', 'B' rather than '130, '120).

8. If input is unacceptable, print error message and return to print query again.

9. The symbol '>' will be used to indicate to the analyst an input is expected. The default, if any, should be printed immediately preceding the '>' symbol.

10. In the case where 'B' and 'X' are acceptable responses to a query, the printed message should include: '(Back-up and Exit Options Not Allowed)'.

- Exit logic at the end of an I-100 interactive program should print the elapsed time, then ask 'E(X)IT or (R)ESTART. If 'R' response, return to the logical start of the program. If 'X' response, save global common and set event flag 50. (INCLUDE '[300,3]CAMSSAV').

- All interactive programs will include the global common blocks via the statement INCLUDE '[300,3]CAMSCOMON'. Offline programs will include if needed.

- Interactively, passage of data from one module to another will be accomplished by the GLOBAL common blocks and working data files. All files are defined in section 3.4.

- There are no restrictions on common blocks within the individual modules, however they must be defined in the design document.

- Image library routines must be utilized. These utilities include routines for interpreting user input, magnetic tape utilities and many others. A list has been provided to each team member. It is each team members responsibility to familiarize himself with these utilities.

EXAMPLE 1

```
      BYTE FILNAM (32), W(74)
10    WRITE   (6,1000) FILNAM
1000  FORMAT('$ FILE NAME:', 32A1, 2X, '>')
C*    RING BELL
      CALL OUTPUT (7)
      READ (6,1010) W
1010  FORMAT (74A1)
C*    REMOVE LEADING BLANKS
      CALL FRONT (W,74)
      IF (W(1).EQ.'X') GO TO exit logic
      IF (W(1).EQ.'B') GO TO previous query (not statement 10)
      IF (W(1).EQ.' ') GO TO accept default
      DO 20 I = 1,30
20    FILNAM(I) = W(I)
      GO TO 10
```

Tektronix printout from the above would look like the following:

```
FILE NAME:  DK1:[100,2]518176032 > MTO:[170,1]IMAGE.DAT
FILE NAME:  MTO:[170,1]IMAGE.DAT > C/R to accept
```

EXAMPLE 2

```
      BYTE W(74)
      INTEGER CHNVEC(4)
 10   WRITE (6,100) CHNVEC
100   FORMAT('$ CHANNELS', 413 '?>')
      CALL OUTPUT (7)
      READ(6,101)W
101   FORMAT(74A1)
      CALL FRONT(W,74)
      IF(W(1).EQ.'X') GO TO exit logic
      IF(W(1).EQ.'B') GO TO previous query (not statement 10)
      IF(W(1).EQ.' ') GO TO accept default
C*    Decode input string
      IP = 0
      DO 20 I = 1,4
      CALL INTFF(IP, W, 74, CHNVEC(I))
 20   CONTINUE
      perform error checking if applicable and return to 10 if
         an error is detected.
```

EXAMPLE 3: HEADING PRINT

DATE: MM/DD/YY

TIME: HR:MM:SS

IMAGE DISPLAY PROGRAM/MAY 1977

APPENDIX A

DEVELOPMENT STANDARDS

- All coding will be done in FORTRAN 4 and complied with the F4P compiler. (Exceptions may be granted if assembly language will make a considerable time difference.)
- Upon entry, each program will
  1. Clear the Tektronic screen - CALL OUTPUT (27,12). Interactive programs only.)
  2. Print the program name and version/date.
  3. Print the current date and time.
  4. Initiate timing.
- Each program should have a comment header giving the name of the program or subroutine, its primary function and name of programmer.
- Comments should be present for every local major function.
- Queries
  1. Brief and meaningful, with all acceptable responses indicated in parenthesis if possible.
  2. Print default condition and allow answer on same line. The symbol '>' will be used to prompt the user for input. (See example 1)
  3. C/R ' ' accept default (if no default, repeat query). Default conditions should be printed immediately preceeding the input prompt '>'.
  4. 'B' back-up to last query.
  5. 'X' branch to program exit logic.

6. Read responses with Alphanumeric free form format and use IMALIB routines 'FRONT', 'INTEF', etc. to interpret answers. Blanks and/or commas should be used as delimiters.

7. Test on alphanumeric characters rather than octal numbers (i.e., 'Y', 'N', 'B' rather than '130, '120).

8. In order to provide 'B' (backup option), logic between two queries should be compatible.

- At the end of an I-100 interactive program, ask 'E(X)IT or (R)ESTART, i.e., 'R' to return to the logical start of the program, 'X' to return to control program. Before exit, print elapsed time and save the GLOBAL common area on disk file GLOBAL.TMP.
- Global common areas will reside in memory at all times.
- Passage of data from one module to another will be accomplished by the GLOBAL common area and working data files.
- There will be no restrictions on common blocks within the individual modules, however they must be defined in the design document.

EXAMPLE 1

```
BYTE FILNAM (32), W(74)

  10    WRITE (6,1000) FILNAM

1000    FORMAT ('$ FILE NAME:', 32A1, 2X, '>')

        CALL OUTPUT (7)

        READ (6,1010) W
1010    FORMAT (74A1)
        CALL FRONT (W,74)
        IF (W(1).EQ.'W') GO TO exit logic

        IF (W(1).EQ.'B') GO TO previous query (not statement 10)
        IF (W(1).EQ.' ') GO TO accept default
        DO 20 I = 1,20
  20    FILNAM(I) = W(I)
```

Tektronix printout from the above would look something like the below:

```
FILE NAME:  DK1:[100,2]518176032 > MTO:[170,1]IMAGE.DAT

FILE NAME:  MTO:[170,1]IMAGE.DAT > C/R to accept
```

APPENDIX B

GLOSSARY

The following definitions and identifications refer only to LACIE and CAMS Procedure 1. For more comprehensive definitions see the "First LACIE Dictionary of Remote Sensing Terminology," JSC-11332, or other general works on remote sensing.

Acquisition - The portion of a Landsat scene which represents a LACIE segment at a specific point in time.

Alarm - A bit image which is used for transient storage of results of operations on images. Alarms are displayed by pushbutton on the Image-100. (See Theme)

Automatic Labelling - This is a procedure for determining the most probable identity of a cluster according to the label of the nearest labelled dots.

C&I - Cataloging and indexing.

CAMS - Classification and Mensuration Subsystem of LACIE.

CAMS/CAS Interface Tape - A digital computer tape primarily used to transmit information from CAMS to CAS.

CAS - Crop Assessment Subsystem of LACIE.

Category - In essence, a name to be used for identifying clusters and classes.

CIR - Color infrared image.

Class - Normally the groupings of clusters that are known to belong to given categories, such as wheat, non-wheat, and spring wheat.

Classification Map - The Universal Format classified image, found on the DTRM tape, in which all pixels are assigned to one or another category.

Cluster - A group of pixels that are similar in spectral values, as generated by automatic (non-supervised) classification programs.

Cluster Map - A Universal Format image, to be found on the DTRM tape, which reflects (spatially) how pixels were assigned to the generated clusters.

Control Program - The basic subunit of the CAMS Image-100 Hybrid System which calls all other interactive programs in the system; implicitly or explicitly, an analyst returns to this program whenever he exits other programs.

CRT - Cathode ray tube, a color television tube used for display of images on the Image-100.

DEC - Digital Equipment Corporation.

DO - "Designated other" pixels; usually those areas in the scene known not to be the category of interest (rivers, cities, etc.).

DO/DU Cards - A deck of punched cards which list vertices of fields containing all DO and DU pixels.

Dot - A pixel selected in some systematic way for use in Procedure 1 classification or clustering schemes.

DTRM Tapes - (sometimes written DTERM) - digital computer tapes primarily used to transmit imagery data from ERIPS to the Production Film Converter, Building 30, for preparation of photographic images.

DU - "Designated unidentifiable" pixels; those pixels which are to be excluded from proportion calculations; usually clouds and cloud shadows.

EOD - Earth Observations Division of NASA at the Johnson Space Center.

ERIPS - Earth Resources Interactive Processing System, the primary analysis system for LACIE, implemented on an IBM-360 computer system.

ERTS Format - A raw-data tape format used for Landsat-1 and -2 imagery data.

Global Common - A portion of computer memory reserved for common use by all subunits of the CAMS Image-100 Hybrid System.

Gould Printer - A user-oriented immediate access printer physically located adjacent to the Image-100 console.

GSFC - Goddard Space Flight Center, NASA, Greenbelt, Maryland.

Hybrid - In the CAMS Image-100 Hybrid System this term refers to a means of performing certain operations on ERIPS to save time.

Image-100 - The General Electric Interactive Multispectral Analysis system, Model 100, which incorporates a PDP 11/45 computer.

I-100 - See Image-100.

ISOCLS - Interactive Self-Organizing Clustering Program, a non-supervised classification program.

IR - Infrared.

JSC - Lyndon B. Johnson Space Center, Houston, Texas.

Kauth Transformation - A mathematical transformation used to reduce the four highly-redundant Landsat channels to two very independent channels, called "greenness" and "brightness," and two others which carry little information.

LACIE - Large Area Crop Inventory Experiment. Also used as an acronym (with a version number) to identify the primary computer system for processing imagery data. The LACIE computer system includes ERIPS as well as the batch processing functions on the IBM computers in Bldg. 30, JSC.

LARSYS - Laboratory for the Application of Remote Sensing System, referring to programs somehow related to the remote sensing organization in Purdue University. The term is locally used to refer to the EOD-LARSYS program for processing of remotely sensed imagery data, which now bears little resemblance to the original programs.

LARSYS Format - A digital computer tape format for imagery, based on interleaved channels.

LPDL - The LACIE Physical Data Laboratory in Building 17.

MAG Tape - Digital computer tape.

Multitemporal Analysis - Operations in which several acquisitions are analyzed as a single multi-channel image to improve precision of calculation; called multitemporal because the acquisitions are images of the same scene taken at different times.

PFC - Production Film Generator in Building 30.

Procedure 1 - A procedure for performing the classification and mensuration function of LACIE.

RP04 - Trade name for the disk drives used in the current hardware configuration for the PDP 11/45.

RSX-11D - Multiprogramming operating system for the PDP 11/45 computer.

Scatter Plot - A graph of the location of pixels in spectral coordinates, which may be raw spectral data or Kauth data.

Segment - A fixed geographical area which is analyzed in the LACIE project. Segments are identified by numbers. Individual acquisitions of each segment are analyzed as they become available.

Spatial Coordinates - Coordinates in space, such as line number and pixel number.

Spectral Coordinates - Coordinates in terms of spectral intensity, such as numbers for each of the four Landsat-2 channels.

TBD - 'To Be Determined.'

Tektronix - The brand name of the interactive terminal with graphics screen associated with a printer for the contents of the same screen, both physically associated with the Image-100.

Theme - A bit image used for semipermanent storage of results of operations on images themes; can be displayed by pushbutton on the screen of the Image-100. (See Alarm)

Trajectory Plot - A graph showing the location in spectral coordinates of a dot (or other pixel) as a function of time.

Universal Format - An interleaved-pixel format for images which is specified for use in the LACIE program (see LACIE Level III baseline document number 00701).

Vector - As used in this document, the set of spectral values associated with a pixel. For a single (four-channel) acquisition there will be as many as four such values. For a four-acquisition multitemporal image there could be as many as sixteen values associated with each pixel.

Window - An area in the refresh memory and on the screen that is used for display purposes.

Zoom - Magnify by repeating pixels.

INDEX TO ALL VOLUMES

The following index lists all computer programs and subroutines found in the text and printouts, and the variables listed in the text. The first number of each description is the volume number, and the remaining number refers to the section in that volume. A preceding L indicates the position of a listing. Therefore, 1-3.3.1 indicates a reference in section 3.3.1 of volume 1, and L2-14.6 indicates that a listing can be found in section 14.6 of volume 2. The list of programs and subroutines is definitive. The list of variables is not complete since those not mentioned in text are not included.

NAME	DESCRIPTION		LINE	COMMENT	OCCURRENCES					LOCATION OF LISTING
A		VARIABLE	1		1-3.5.1.2	1-3.5.1.6	1-3.5.2.6	1-3.5.2.X	3-1.1	
			2		3-9.					
ACAT		VARIABLE	1		1-3.5.2.6					
ACDAT	COM1	VARIABLE	1		1-4...9					
ACDATE		VARIABLE	1		1-3.5.1.4	1-3.5.2.5	1-3.5.2.6	1-3.5.2.9		
ACDISP	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.5			
ACLLAP		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	1-3.5.2.6	2-14.		L2-14.
ADATES	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.5	1-3.5.2.6	1-3.5.2.9	
			2		1-3.5.2.X					
ADSK		VARIABLE	1		1-3.5.1.5					
AID		VARIABLE	1		1-3.5.1.5					
AJUL		VARIABLE	1		1-3.5.1.5					
ALABEL		VARIABLE	1		1-3.5.2.6					
ALLUPD	PRIVT	SUBROUTINE	1		1-3.5.1.3					L2-3.7
ALP	COM2	VARIABLE	1		1-3.4.9					
ALP0	COM2	VARIABLE	1		1-3.4.9					
ALPTAB	PRIVT	SUBROUTINE	1							L2-18.4
ALSORT	PRIVT	SUBROUTINE	1		1-3.5.2.6					L2-14.4
ANACAT		VARIABLE	1		3-4.4					
ANALYST		VARIABLE	1		1-3.4.9					
ANCL		VARIABLE	1		1-3.5.2.3					
AQLFST	PRIVT	SUBROUTINE	1		1-3.5.2.X					L2-21.1
ARAND		VARIABLE	1		1-3.5.2.6					
ARIND		VARIABLE	1		1-3.5.2.6					
ARRY		VARIABLE	1		1-3.5.2.9					
ASEG		VARIABLE	1		1-3.5.1.5					
ASNLUN	F4PLIB	SUBROUTINE	1		1-3.5.1.4					
ASSIGN	F4PLIB	SUBROUTINE	1		1-3.5.2.6	1-3.5.2.8				
ATTACH	F4PLIB	SUBROUTINE	1		1-3.5.2.5	1-3.5.2.6	1-3.5.2.8			
AYR		VARIABLE	1		1-3.5.1.5					
B	COM4	VARIABLE	1		1-3.4.9	1-3.5.1.2	1-3.5.1.6	1-3.5.2.3	1-3.5.2.5	
			2		1-3.5.2.X	3-10.	3-19.			
BIASCR		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	1-3.5.2.9	2-18.		L2-18.
BITSET	SHARE	SUBROUTINE	1		3-1.					L3-1
BLKTHM	SHARE	SUBROUTINE	1		1-3.5.2.5	1-3.5.2.7	1-3.5.2.8	3-2.		L3-2
BLOCK		VARIABLE	1		1-3.5.2.4					
BLOWUP	PRVT	SUBROUTINE	1		1-3.5.2.5					L2-13.6
BRFCLU	PRVT	SUBROUTINE	1		1-3.5.2.9					L2-14.6
BSTAT	OFFICE	PROGRAM	1	TASK	1-CONTNTS	1-3.3.1	1-3.3.5	1-3.5.1.4	2-4.	L2-4.
BUFCLM		VARIABLE	1		1-3.5.2.6					
BUFDOT		VARIABLE	1		1-3.5.2.6					
C		VARIABLE	1		1-3.5.1.2	1-3.5.2.3				
C1		VARIABLE	1		1-3.4.9					
C2		VARIABLE	1		1-3.4.9					
C3		VARIABLE	1		1-3.4.9					
C4		VARIABLE	1		1-3.4.9					
CALP		VARIABLE	1		1-3.5.2.3					
CAMSAVE.INC		FILE	1		1-3.4.9					
CAMSCOMUN.INC		FILE	1		1-3.4.9					
CAMSEX		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	2-7.			L2-7.
CAMSPARAM.INC		FILE	1		1-3.4.6	1-3.4.9	1-3.5.1.3	1-3.5.1.5		
CARD		VARIABLE	1		1-3.5.1.3					

ORIGINAL PAGE IS
OF POOR QUALITY

NAME	DESCRIPTION		LINE	COMMENT	OCCURRENCES					LOCATION OF LISTING
CARD.DAT		FILE	1		1-3.5.1.2					
CARDIN	PRIVT	FUNCTION	1		1-3.5.1.3					L2-3.10
CAT		VARIABLE	1		1-3.5.1.3					
CATKNT	COM1	VARIABLE	1		1-3.4.9	1-3.5.1.4	1-3.5.2.9			
CATLOG	PRIVT	FUNCTION	1		1-3.5.1.3					L2-3.2
CATNAM	COM2	VARIABLE	1		1-3.4.9	1-3.5.1.3	1-3.5.1.4	1-3.5.2.5	1-3.5.2.6	
			2		1-3.5.2.7	1-3.5.2.9				
CATTH	COM1	VARIABLE	1		1-3.4.9	1-3.5.1.4	1-3.5.2.5			
CATTHM	PRIVT	SUBROUTINE	1		1-3.5.2.7	2-15.4				L2-15.4
CBRR		VARIABLE	1		1-3.5.2.3					
CDRED	PRIVT	SUBROUTINE	1		1-3.5.1.4					L2-4.1
CHAN		VARIABLE	1		1-3.5.2.8					
CHANVC		VARIABLE	1		1-3.5.2.6					
CHL		VARIABLE	1		1-3.5.2.3					
CHNVBC		VARIABLE	1		1-3.5.2.5					
CHNVEC	COM1	VARIABLE	1		1-3.4.9	1-3.5.1.4	1-3.5.2.6	1-3.5.2.9		
CHP		VARIABLE	1		1-3.5.2.3					
CLABEL	PRIVT	SUBROUTINE	1		1-3.5.2.6	1-3.5.2.7	2-15.7			L2-14.3
			2							L2-15.7
CLACAT		VARIABLE	1		3-4.4					
CLADIS		PROGRAM	1							L2-16.
CLAD:S	PRIVT	SUBROUTINE	1							L2-16.3
CLASAV	PRIVT	SUBROUTINE	1		1-3.5.2.X					L2-21.4
CLASS		SUBROUTINE	1	TASK	1-CONTNTS	1-3.5.2.8				L2-5.3
CLASSMAP.TMP		FILE	1		1-3.4.3	1-3.5.2.2	1-3.5.2.8	1-3.5.2.X		
CLATHM		SUBROUTINE	1		1-3.5.2.8					L2-16.5
CLAWND	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.5				
CLLAB		VARIABLE	1		1-3.5.2.6					
CLOSE	F4PLIB	SUBROUTINE	1		1-3.5.1.4	1-3.5.2.5	1-3.5.2.6	1-3.5.2.8		
CLREF	F4PLIB	SUBROUTINE	1		1-3.5.2.1					
CLSREP	PRIVT	SUBROUTINE	1		2-19.					L2-19.
CLUARY		VARIABLE	1		3-4.2					
CLUDIS		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	1-3.5.2.7	2-15.		L2-15.
CLURPT		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	1-3.5.2.9	2-19.		L2-19.
CLUSEL		SUBROUTINE	1		1-3.5.2.5					L3-4.2
CLUSNN	PRVT	SUBROUTINE	1		1-3.5.2.9					L2-14.8
CLUSTATS.TMP		FILE	1		1-3.4.5	1-3.5.2.2	1-3.5.2.6	1-3.5.2.9	1-3.5.2.X	
CLUSTATP.TMP		FILE	1		1-3.5.2.5					
CLUSTERMP.TMP		FILE	1		1-3.4.3	1-3.5.2.2	1-3.5.2.7	1-3.5.2.8	1-3.5.2.X	
CLUSTR	PRVT	SUBROUTINE	1							L2-5.4
CLUTHM	PRIVT	SUBROUTINE	1		1-3.5.2.7	2-15.5				L2-15.5
CLUWND	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.5	1-3.5.2.7			
CMASK		VARIABLE	1		1-3.5.2.7					
CNTRL	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.2				L2-2.2
COM1	GLOBAL	COMMON	1		1-CONTNTS	1-3.4.5	1-3.4.9	1-3.5.1.4	1-3.5.2.2	1-F3-38
			2		1-3.5.2.7	1-3.5.2.9	1-3.5.2.X			
COM2	GLOBAL	COMMON	1		1-CONTNTS	1-3.3.1	1-3.4.4	1-3.4.9	1-3.5.2.2	1-F3-3D
COM3	GLOBAL	COMMON	1		1-CONTNTS	1-3.4.9	1-3.5.2.2	1-3.5.2.X		1-F3-3E
COM4	GLOBAL	COMMON	1		1-CONTNTS	1-3.4.9	1-3.5.2.2	1-3.5.2.9		1-F3-3F
			2		1-3.5.2.7	1-3.5.2.9	1-3.5.2.X			
COM5	GLOBAL	COMMON	1		1-CONTNTS	1-3.4.9	1-3.5.2.2	1-3.5.2.9	1-3.5.2.X	1-F3-36
COMLUT	PRIVT	SUBROUTINE	1		1-3.5.2.3					L2-9.2
COMPAR	PRIVT	FUNCTIONNE	1		1-3.5.1.3					L2-3.11
CONDIS	PRIVT	SUBROUTINE	1		1-3.5.2.7	2-15.12				L215.12

NAME	DESCRIPTION		LINE COMMENT	OCCURRENCES					LOCATION OF LISTING
CONDIT	PRIVT	SUBROUTINE	1	1-[illegible].5.2.7	2-15.13				L215.13
CRUNCH	PRVT	SUBROUTINE	1						L2-9
CSGDPH	SHARE	SUBROUTINE	1	1-[illegible].2.2	1-3.5.2.3	1-3.[illegible].2.4	1-3.5.2.5	1-3.5.2.9	L3-3.
			2	3-[illegible]					
CURDEF	PRVT	SUBROUTINE	1						L2-10.4
CURDEF		VARIABLE	1	1-3.5.2.4					
D		VARIABLE	1	1-3.5.2.5					
DATARD	PRVT	SUBROUTINE	1						L2-19.3
DATE	F4PLIB	SUBROUTINE	1	1-3.5.1.4	1-3.5.2.5				
DAY		VARIABLE	1	1-3.5.1.3					
DCOORD	PRIVT	SUBROUTINE	1	1-3.5.1.2	2-2.6				L2-2.6
DDDOT	PRIVT	SUBROUTINE	1						L2-1.12
DDIESF		VARIABLE	1	1-3.5.2.4					
DEFALT	PRIVT	SUBROUTINE	1	1-3.5.2.7	2-15.11				L215.11
DEFLT2	PRVT	SUBROUTINE	1						L2-16.6
DELBAT	PRIVT	SUBROUTINE	1.	1-3.5.1					L2-1.
DELFLD	PRIVT	SUBROUTINE	1	1-3.5.2.4					L2-10.1
DELFLG	COM2	VARIABLE	1	1-3.4.9	1-3.5.2.X				
DETACH	IMALIB	SUBROUTINE	1	1-3.5.2.5	1-3.5.2.6	1-3.5.2.8			
DFLAG		VARIABLE	1	1-3.5.2.6					
DGSCPL		SUBROUTINE	1	1-CONTNTS	1-3.5.2.5	2-12.1			L2-12.1
DIRCAT		VARIABLE	1	1-3.5.2.8					
DIRCRE	PRIVT	SUBROUTINE	1	1-3.5.1.1					L2-1.
DIRCRE		VARIABLE	1	1-3.5.1.1					
DIRFIL		VARIABLE	1	1-3.5.1.1					
DIRFILE.DAT		FILE	1	1-3.3	1-3.3.1	1-3.5.1.2	1-3.5.[illegible].4	1-3.5.1.5	
			2	1-3.5.1.6	1-3.5.2.2	1-3.5.2.X			
DIRLOD	PRIVT	SUBROUTINE	1	1-3.5.1.3					L2-3.3
DIRUPD	IMAUPD	PROGRAM	1	1-3.5.1.1					L2-1.
DIRUPD	PRIVT	SUBROUTINE	1	1-3.5.1.3	1-3.5.1.5				L2-5.1
DIRUPD		VARIABLE	1	1-3.5.1.1					
DISKID	COM5	VARIABLE	1	1-3.4.9	1-3.5.1.4				
DISKNM		VARIABLE	1	1-3.5.1.3	1-3.5.1.5				
DIST		VARIABLE	1	1-3.4.6					
DK		VARIABLE	1	3-4.2					
DLAB		VARIABLE	1	1-3.4.6					
DLABEL	COM5	VARIABLE	1	1-3.4.4	1-3.4.9	1-3.5.2.4	1-3.5.2.5	1-3.5.2.6	
			2	1-3.5.2.9	1-3.5.2.X	3-11.			
DLABE.		VARIABLE	1	1-3.5.2.4					
DLABE.	COM5	VARIABLE	1	1-3.5.2.6	1-3.5.2.X				
DLSKIP		VARIABLE	1	1-3.5.1.3					
DO1		VARIABLE	1	1-3.5.2.9					
DO2		VARIABLE	1	1-3.6.2.9					
DO3		VARIABLE	1	1-3.5.2.9					
DODU		VARIABLE	1	1-3.5.2.4					
DOI		VARIABLE	1	1-3.5.2.3					
DOLABEL		VARIABLE	1	1-3.5.2.4					
DOPCT		VARIABLE	1	1-3.5.2.9					
DOTARY	COM4	VARIABLE	1	1-3.4.9	1-3.5.2.5				
DOTCAT	COM1	VARIABLE	1	1-3.4.9					
DOTCLU	COM1	VARIABLE	1	1-3.4.9	1-3.5.2.9				
DOTDAT		VARIABLE	1	1-3.5.2.6					
DOTDAY	COM2	VARIABLE	1	1-3.4.9	1-3.5.2.X				

NAME		DESCRIPTION	LINE	COMMENT	OCCURRENCES					LOCATION OF LISTING
DOTFLG		VARIABLE	1		1-[illegible].1.3					
DOTGXY.TMP		FILE	1		1-[illegible].7	1-3.5.2.2	1-3.[illegible] 2.5			
DOTIN		SUBROUTINE	1		1-[illegible].2.5	1-3.5.2.9				L3-4.1
DOTLAB		SUBROUTINE	1		1-[illegible].2.5					L2-13.4
DOTN		VARIABLE	1		1-[illegible].4.6					
DOTOFF	PRIVT	SUBROUTINE	1	=UNLDOT						
DOTOVR		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	1-3.5.2.5	2-11.		L2-11.
DOTPRO		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	1-3.5.2.5			L2-13.
DOTRPT		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	2-17.			L2-17.
DOTSAV	PRIVT	SUBROUTINE	1		1-3.5.2.X					L2-21.2
DOTSEL		SUBROUTINE	1		1-3.5.2.9	3-4.4				L3-4.4
DOTS.TMP		FILE	1		1-3.3	1-3.4.4	1-3.5.2.2	1-3.5.2.5	1-3.5.2.6	
			2		1-3.5.2.X					
DOTUPD	OFFICE	PROGRAM	1	TASK	1-CONTNTS	1-3.3.1	1-3.3.3	1-3.5.[illegible].3	2-3.	L2-3.
DOTYPE		VARIABLE	1		3-4.4					
DSET	SHARE	SUBROUTINE	1		1-3.5.1.3	1-3.5.1.5	1-3.5.2.3	3-5.		L3-5.
DSKCHK	SHARE	SUBROUTINE	1		1-3.3.1	1-3.5.1.2	1-3.5.1.3	1-3.5.[illegible].4	1-3.5.1.5	L3-6.
			2		1-3.5.1.6	1-3.5.2.2	3-6.			
DSKID		VARIABLE	1		3-6.					
DSKMNT	COM3	VARIABLE	1		1-3.4.9					
DSKTBL		VARIABLE	1		1-3.5.1.1					
DSKTBL.DAT		FILE	1		1-3.3	1-3.3.1	1-3.5.1.6	1-3.5.[illegible].2	1-3.5.2.X	
DSSKIP		VARIABLE	1		1-3.5.1.3					
DTCL10	SHARE	SUBROUTINE	1		3-4.					L3-4.0
DTERM	OFFICE	PROGRAM	1	TASK	1-CONTNTS	1-3.3.1	1-3.3.5			L2-5.
DTRM		PROGRAM	1	=DTERM	1-3.5.1.5	2-5				
DTWIND	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.4				
DUPCT		VARIABLE	1		1-3.5.2.9					
DUSET		SUBROUTINE	1	=DSET	1-3.5.2.3					
DY		VARIABLE	1		1-3.5.1.2	1-3.5.2.X				
E		VARIABLE	1		3-4.3					
EFLAG1	COM3	VARIABLE	1		1-3.4.3	1-3.4.9	1-3.5.2.7	1-3.5.2.X		
EFLAG2	COM3	VARIABLE	1		1-3.4.9	1-3.5.2.X				
EFLAG3	COM3	VARIABLE	1		1-3.4.9	1-3.5.2.4	1-3.5.2.X			
EFLAG4		VARIABLE	1		1-3.4.6	1-3.4.9	1-3.5.2.6	1-3.5.2.7	1-3.5.2.X	
EFLAG5	COM3	VARIABLE	1		1-3.4.5	1-3.4.9	1-3.5.2.X			
EFWARN	PRIVT	SUBROUTINE	1		1-3.5.2.7	2-15.1				L2-15.1
ELAPSE	SHARE	SUBROUTINE	1		1-3.5.1.2	1-3.5.1.3	1-3.1.5.5	1-3.5.1.6	1-3.5.2.1	L3-6.1+
			2		1-3.5.2.2	1-3.5.2.5	1-3.5.2.6	1-3.5.2.8	1-3.5.2.9	L3-7.
			3		1-3.5.2.X	3-7.				
EOF		VARIABLE	1		1-3.5.1.5					
ERRMES	SHARE	SUBROUTINE	1		1-3.5.1.3	1-3.5.1.5	3-8.			L3-8
ERROR		VARIABLE	1		1-3.5.1.5					
EXFIL		VARIABLE	1		1-3.5.2.5					
EXIT	F4PLIB	SUBROUTINE	1		1-3.5.2.5					
EXPTD	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.3				L2-2.3
F		VARIABLE	1		3-4.3					
F11		VARIABLE	1		1-3.5.2.3	3-5.				
FADE	PRIVT	SUBROUTINE	1	=FMAINT	1-3.5.1.5					
FAKCLG	PRIVT	SUBROUTINE	1							L2-10
FDLINT	PRIVT	SUBROUTINE	1							L2-16.1
FFFPI	SHARE	SUBROUTINE	1		1-3.5.2.3	3-9.				L3-9
FFUNC	SHARE	SUBROUTINE	1		1-3.5.2.5	3-10.				L3-10.
FIELD	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.5				L2-2.5

NAME		DESCRIPTION	LINE	COMMENT	OCCURRENCES					LOCATION OF LISTING
FIELD		VARIABLE	1		1-3.5.2.4	1-3.5.2.8				
FIELDS.TMP		FILE	1		1-3.5.2.4	1-3.5.2.2	1-3.5.2.4	1-3.5.2.8	1-3.5.2.9	
			2		1-3.5.2.X					
FILE		VARIABLE	1		1-3.5.1.4	3-5.	1-3.5.2.3			
FILE		VARIABLE	1		1-3.5.2.3					
FILEIN		VARIABLE	1		1-3.5.2.2					
FILEOUT		VARIABLE	1		1-3.5.2.2					
FILEST	PRIVT	SUBROUTINE	1		1-3.5.1.1					L2-1.
FILNUM		VARIABLE	1		1-3.5.1.5					
FILTYP		VARIABLE	1		1-3.5.2.X					
FINDOT	PRVT	SUBROUTINE	1		1-3.5.2.5					L2-13.2
FL		VARIABLE	1		1-3.5.2.8	3-11.2				
FLABEL		VARIABLE	1		1-3.5.2.6					
FLAG		VARIABLE	1		1-3.5.2.2	3-4.4	3-6.			
FLDDAY	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.X				
FLDDEF		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	2-10.			L2-10.
FLDEND	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.7				L2-2.7
FLDINT		SUBROUTINE	1		1-3.5.1.2	1-3.5.2.8	2-2.17	3-11.		L3-11.+
			2							L2-2.17
FLDLAB		VARIABLE	1		3-11.1					
FLDNAM	PRIVT	SUBROUTINE	1							L2-10.
FLDNAM		VARIABLE	1		1-3.5.2.8					
FLDOFF	PRIVT	SUBROUTINE	1		1-3.5.2.X					L2-21.8
FLDRPT		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	2-20.			L2-20.
FLDRPT	PRIVT	SUBROUTINE	1		1-3.5.2.4	1-3.5.2.9				L2-10.2
FLDSAV	PRIVT	SUBROUTINE	1		1-3.5.2.X					L2-21.3
FLDST	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.4				L2-2.4
FLDUPD	OFFICE	PROGRAM	1	TASK	1-CONTNTS	1-3.3.1	1-3.3.3	1-3.3.4	1-3.5.1.2	L2-2.1
			2		2-2.1					
FLGDOT	SHARE	SUBROUTINE	1		1-3.5.1.2	1-3.5.2.4	2-2.18	3-11.1		L3-11.1
			2							+L22.18
FLL		VARIABLE	1		1-3.5.2.8					
FLN		VARIABLE	1		1-3.5.2.3					
FMAINT	PRIVT	SUBROUTINE	1		1-3.5.1.5					L2-5.5
FN		VARIABLE	1		3-5.					
FORM		VARIABLE	1		1-3.5.2.3					
FOR001.DAT		FILE	1	SYSTEM	1-3.5.1.3					
FPTR		VARIABLE	1		1-3.5.2.4					
FRONT	I-100	SUBROUTINE	1		1-3.5.2.5	1-3.5.2.6	1-3.5.2.8	1-3.5.2.9		
FSTVID	SHARE	SUBROUTINE	1		1-3.5.1.3	1-3.5.1.5	3-12.			L3-12.
FTRNFR	PRIVT	SUBROUTINE	1		1-3.5.2.2					L2-8.1
FUL	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.5				
FULL		VARIABLE	1		1-3.5.2.5					
FUL012	PRIVT	SUBROUTINE	1	=FUL014						L2-9.5
FUL013		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1	1-3.5.2.3	2-9.		L2-9.
FUL014	PRIVT	SUBROUTINE	1	TASK	1-3.5.2.3					L2-9.4
FVCLOS	I-100	SUBROUTINE	1		3-12.					
FVDLTE	I-100	SUBROUTINE	1		3-12.					
FVDSET	I-100	SUBROUTINE	1		3-12.					
FVOPEN	I-100	SUBROUTINE	1		3-12.					
FVREAD	I-100	SUBROUTINE	1		3-12.					
FVRWND	I-100	SUBROUTINE	1		3-12.					
FVWAIT	I-100	SUBROUTINE	1		3-12.					
FVWRIT	I-100	SUBROUTINE	1		3-12.					

ORIGINAL PAGE IS OF POOR QUALITY

NAME	DESCRIPTION		LINE	COMMENT	OCCURRENCES	LOCATION OF LISTING
G	COM4	VARIABLE	1		1-3.4.9 1-3.5.2.3 1-3.5.2.5	L2-9.1
GABI	PRIVT	SUBROUTINE	1		1-3.5.2.3	
GB		VARIABLE	1		1-3.5.2.3	L2-9.6
GBCALC	PRIVT	SUBROUTINE	1			
GETADR	F4PLIB	SUBROUTINE	1		1-3.5.1.4	
GETO00	SHARE	SUBROUTINE	1		1-3.5.2.7 2-15.8 3-13.	L3-11.+
			2			L2-15.8
GLOBAL.TMP		FILE	1		1-3.4.9 1-3.5.2.2 1-3.5.2.5 1-3.5.2.9	
GMAX	COM4	VARIABLE	1		1-3.4.9 1-3.5.2.5	
GMIN	COM4	VARIABLE	1		1-3.4.9 1-3.5.2.5	
GREND		VARIABLE	1		1-3.5.2.9	
GRENDB		VARIABLE	1		1-3.5.2.5	
GRENSS		VARIABLE	1		1-3.5.2.9	
GRID	COMS	VARIABLE	1		1-3.4.9	L2-13.7
GTYPE	PRVT	SUBROUTINE	1		1-3.5.2.5	
HORREC.DAT		FILE	1		1-3.2.2	
HEXD	PRIVT	SUBROUTINE	1		1-3.5.1.5	L2-4.3 L2-5.9
HEAD'N	PRIVT	SUBROUTINE	1		1-3.5.2.3	
HFG		VARIABLE	1		1-3.5.2.3	L2-9.4
HFL		VARIABLE	1			
HOCUTT	PRIVT	SUBROUTINE	1			L3-14.
HPROS	SHARE	SUBROUTINE	1	=DSET	1-3.5.1.5 3-14.	
HREAD		SUBROUTINE	1		1-3.5.2.3	L2-13.5
HSEKPG	PRIVT	SUBROUTINE	1		1-3.5.2.5	L2-19.4
			2			
HSIZ		VARIABLE	1		1-3.5.2.3	L3-15.
HVFY	SHARE	SUBROUTINE	1		1-3.5.1.5 3-15.	
I		VARIABLE	1		1-3.5.1.2 1-3.5.1.6 1-3.5.2.5 1-3.5.2.X 3-1.1	
			2		3-9.1 3-17.	
IBUF		VARIABLE	1		1-3.5.1.2 1-3.5.1.4 1-3.5.2.5 3-2.	
IBYTE	IMALIB	SUBROUTINE	1		1-3.5.2.5	
IC		VARIABLE	1		1-3.5.2.7 3-13.	
ICAKNT		VARIABLE	1		1-3.5.2.8	
ID		VARIABLE	1		1-3.5.1.2 1-3.5.1.5	
IDATE	F4PLIB	SUBROUTINE	1		1-3.5.2.5 1-3.5.2.9	
IDENT1	PRIVT	SUBROUTINE	1		1-3.5.2.8	
IER		VARIABLE	1		1-3.4.6	
IERR		VARIABLE	1		1-3.5.1.2	
IFST		VARIABLE	1		1-3.5.2.4	
II		VARIABLE	1		1-3.5.2.5 1-3.5.2.9 3-7.	
III	COM4	VARIABLE	1		1-3.4.9 1-3.5.2.3 1-3.5.2.5	
IIX1		VARIABLE	1		1-3.5.2.3	
IIX2		VARIABLE	1		1-3.5.2.3	
IIY1		VARIABLE	1		1-3.5.2.3	
IIY2		VARIABLE	1		1-3.5.2.3	
ILABEL		VARIABLE	1		1-3.5.2.6	
IMAUPD	OFFICE	PROGRAM	1	TASK	1-3.3 1-3.3.1 1-3.3.2 1-3.3.3 1-3.5.1	L2-1.
IMDATE	COM2	VARIABLE	1		1-3.4.9 1-3.5.2.X	
IMWIND	COM4	VARIABLE	1		1-3.4.9 1-3.5.2.5	L2-19.2
INCLDS	PRIVT	SUBROUTINE	1		1-3.5.2.9	L2-2.2
INDBRT	PRIVT	SUBROUTINE	1		1-3.5.1.2	L2-8.
INIT	INTRAC	PROGRAM	1	TASK	1-CONTNTS 1-3.3.1 1-3.3.3 1-3.3.4 1-3.3.5	
			2		1-3.5.2.1 1-3.5.2.2 2-8	
INIT		VARIABLE	1		1-3.5.1.3	

NAME	DESCRIPTION		LINE	COMMENT	OCCURRENCES					LOCATION OF LISTING
INTFF	F4PLIB	SUBROUTINE	1		1-3.5.2.5	1-3.5.2.6	1-3.5.2.8	1-3.5.2.9		
INTLZE	PRIVT	SUBROUTINE	1		1-3.5.2.2					L2-8.2
INX		VARIABLE	1		1-3.5.1.2					
IO		VARIABLE	1		1-3.5.2.4	1-3.5.2.5	1-3.5.2.9	3-3.	3-4.1	
IOP		VARIABLE	1		1-3.5.2.5	3-2.				
IOPT		VARIABLE	1		1-3.5.1.2					
IP		VARIABLE	1		1-3.5.2.3					
IRODEF.TMP		FILE	1		1-3.5.2.4					
IRESUB		VARIABLE	1		1-3.5.2.8					
IRRE63		SUBROUTINE	1	TASK	1-3.5.2.4					
IRT	IMALIB	SUBROUTINE	1		1-3.5.2.5					
IRU	IMALIB	SUBROUTINE	1		1-3.5.2.5					
IOPRNT	PRIVT	SUBROUTINE	1 2		3-4.2					L2-4.2
ISEG	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.6	1-3.5.2.9		
ISET		VARIABLE	1		3-13.					
ITUCHN		VARIABLE	1		1-3.5.2.6					
IU		VARIABLE	1		3-17.					
IWL	IMALIB	SUBROUTINE	1		1-3.5.2.5					
IWT	IMALIB	SUBROUTINE	1		1-3.5.2.5					
IX		VARIABLE	1		1-3.5.2.3	1-3.5.2.7	3-4.2			
IX1	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.7			
IX2	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.7			
IXY		VARIABLE	1		1-3.5.1.2					
IY		VARIABLE	1		1-3.5.2.3	1-3.5.2.7				
IY1	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.7			
IY2	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.7			
J		VARIABLE	1		1-3.5.1.2	1-3.5.1.6	1-3.5.2.X			
JDIR		PROGRAM	1	*IMAUPD						L2-1.
JJ		VARIABLE	1		3-11.2					
JJ		VARIABLE	1		1-3.5.2.8					
JUL		VARIABLE	1		1-3.5.2.8					
JUL10		VARIABLE	1		1-3.5.1.2	1-3.5.2.X				
JULDAT		VARIABLE	1		1-3.5.1.5					
JULIAN	PRIVT	FUNCTION	1 2							L2-3. + L2-5.8 L2-1 + L2-2.16
JULIAN	PRIVT	SUBROUTINE	1 3		1-3.5.1.2	1-3.5.1.3	1-3.5.2.X	2-2.16		
X		VARIABLE	1		1-3.5.2.5	1-3.5.2.X				
KAUTH		SUBROUTINE	1		1-3.5.1.4					L2-1.
KK		VARIABLE	1		3-4.1					
KNN		VARIABLE	1		1-3.4.6	1-3.5.2.6				
KNNPRN	PRIVT	SUBROUTINE	1		1-3.5.2.6					L2-14.5
KIMBRT	PRIVT	SUBROUTINE	1		1-3.5.1.2					L2-2.2
KRW		VARIABLE	1		1-3.5.2.2					
L		VARIABLE	1		3-17.					
LABEL		VARIABLE	1		1-3.5.1.3	1-3.5.2.8				
LABNUM		VARIABLE	1		1-3.5.1.3					
LE		VARIABLE	1		1-3.5.2.3					
LECTAP	SHARE	SUBROUTINE	1		1-3.5.1.3	1-3.5.1.5	3-16.			L3-16.
LGBC		VARIABLE	1		1-3.5.2.3					
LGF		VARIABLE	1		1-3.5.2.3					
LGL		VARIABLE	1		1-3.5.2.3					

NAME	DESCRIPTION	LINE	COMMENT	OCCURRENCES	LOCATION OF LISTING
LIN	SHARE SUBROUTINE	1		1-3.5.2.3 1-3.5.2.4 1-3.5.2.5 3-17.	L3-17.
LIST1	PRIVT SUBROUTINE	1		1-3.5.2.7 2-15.3	L2-15.3
LIST2	PRIVT SUBROUTINE	1		1-3.5.2.7 2-15.15	L2-15.5
LIST3	PRIVT SUBROUTINE	1			L3-16.4
LOOKUP	VARIABLE	1			L2-5.11
LOWL	VARIABLE	1		1-3.5.2.8	
LREED	SUBROUTINE	1	-DSET	1-3.5.1.5	
LRJUNK	PRIVT SUBROUTINE	1			L2-12.2
LS	VARIABLE	1		1-3.5.2.3	
LUN	VARIABLE	1		1-3.4.6 1-3.5.2.3 3-5.	
LUN	VARIABLE	1		1-3.5.2.3	
LUT	VARIABLE	1		1-3.5.2.3	
M	VARIABLE	1		1-3.5.1.2 1-3.5.1.6 1-3.5.2.X	
MAFUPD	PRIVT SUBROUTINE	1		1-3.5.1.5	L2-5.2
MAXACC	VARIABLE	1		1-3.4.5 1-3.4.9 1-3.5.2.9	
MAXACD	VARIABLE	1		1-3.4.9	
MAXCAT	VARIABLE	1		1-3.5.1.3	
MAXCH	VARIABLE	1		1-3.4.9 1-3.5.1.3 1-3.5.1.5	
MAXSUB	VARIABLE	1		1-3.4.9	
MB	VARIABLE	1		1-3.5.2.5 3-2.	
MDODU	PROGRAM	1	-FLDUPD		L2-2.1
MENSTD	PRIVT SUBROUTINE	1		1-3.5.2.9	L2-19.1
MFLDS	VARIABLE	1		1-3.5.2.4	
MGL	VARIABLE	1		1-3.5.2.3	
MIXDIS	PRIVT SUBROUTINE	1		1-3.5.2.7 2-15.14	L215.14
MIXED	PRIVT SUBROUTINE	1		1-3.5.2.7 2-15.16	L215.16
ML	VARIABLE	1		1-3.5.2.5 3-2.	
MD	VARIABLE	1		1-3.5.1.2 1-3.5.2.X	
MR	VARIABLE	1		1-3.5.2.2 1-3.5.2.5 3-2.	
MTXTFG	VARIABLE	1		1-3.5.2.3 3-5.	
MU	VARIABLE	1		1-3.5.2.5 3-2.	
MX	VARIABLE	1		1-3.5.2.3 1-3.5.2.7	
MXC	VARIABLE	1		1-3.5.2.3	
MY	VARIABLE	1		1-3.5.2.3 1-3.5.2.7	
N	VARIABLE	1		1-3.5.1.2 1-3.5.1.6 1-3.5.2.5 1-3.5.2.6 1-3.5.2.X	
		2		3-3. 3-9.	
N	VARIABLE	1		1-3.5.2.6 3-3.	
NACQ	VARIABLE	1		1-3.5.2.6	
NBIT	VARIABLE	1		1-3.5.2.3	
NC	VARIABLE	1		1-3.5.2.3	
NCAR	VARIABLE	1		1-3.5.2.3	
NCMAX	VARIABLE	1		1-3.5.2.3	
NCNTRL	PRIVT SUBROUTINE	1		1-3.5.1.2 2-2.20	L2-2.20
NCPR	VARIABLE	1		1-3.5.2.3	
NCR	VARIABLE	1		1-3.5.2.3	
ND	VARIABLE	1		1-3.4.6	
NDOPIX	VARIABLE	1		1-3.5.2.8	
NDOT	VARIABLE	1		1-3.5.1.5	
NDOTS	VARIABLE	1		1-3.4.4 1-3.4.9 1-3.5.1.3	
NDSPR	VARIABLE	1		1-3.5.2.3	
NDUPIX	VARIABLE	1		1-3.5.2.8	

ORIGINAL PAGE IS
OF POOR QUALITY

NAME	DESCRIPTION		LINE	COMMENT	OCCURRENCES					LOCATION OF LISTING
NBLAB	COM3	VARIABLE	1		1-3.4.9	1-3.5.2.6	1-3.5.2.7			
NF		VARIABLE	1		1-3.5.2.3					
NFIELD	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.21				L2-2.21
NFL		VARIABLE	1		1-3.5.2.4					
NFLD		VARIABLE	1		1-3.5.2.4					
NFLDST	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.19				L2-2.19
NLIN		VARIABLE	1		1-3.5.1.3	1-3.5.1.5	1-3.5.2.8			
NLP		VARIABLE	1		1-3.5.2.8					
NN.TMP		FILE	1		1-CONTNTS	1-3.4.6	1-3.5.2.6	1-3.5.2.7		
NNNYYDDD.DAT		FILE	1		1-3.5.1.1					
NOACQ	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.9	1-3.5.2.X			
NOCAT	COM2	VARIABLE	1		1-3.4.9	1-3.5.1.4	1-3.5.2.5	1-3.5.2.9	1-3.5.2.X	
NOCHAN	COM1	VARIABLE	1		1-3.4.9	1-3.5.1.4				
NODO	COM1	VARIABLE	1		1-3.4.9	1-3.5.1.4	1-3.5.2.9			
NODOT		VARIABLE	1		3-4.4					
NODTMD		VARIABLE	1		1-3.4.9					
NODTWD		VARIABLE	1		1-3.4.9					
NODU	COM1	VARIABLE	1		1-3.4.9	1-3.5.1.4	1-3.5.2.9			
NOFLD		VARIABLE	1		3-11.1					
NOLIN		VARIABLE	1		1-3.5.2.8					
NOSPWD		VARIABLE	1		1-3.4.8	1-3.4.9				
NOSUB	COM1	VARIABLE	1		1-3.4.5	1-3.4.6	1-3.4.9	1-3.5.1.4	1-3.5.2.7	
			2		1-3.5.2.9					
NOTH	COM1	VARIABLE	1		1-3.4.9	1-3.5.1.4	1-3.5.2.9			
NPIX		VARIABLE	1		1-3.5.1.3	1-3.5.1.5	1-3.5.2.8			
NPIX4		VARIABLE	1		1-3.5.1.5					
NPTS		VARIABLE	1		3-11.2					
NR		VARIABLE	1		1-3.5.2.2	3-19.				
NRPOS		VARIABLE	1		1-3.5.2.3					
NS		VARIABLE	1		1-3.5.2.3					
NSAMP		VARIABLE	1		1-3.5.2.8	3-11.2				
NSEND	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.22				L2-2.22
NSTART	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.X				
NT		VARIABLE	1		1-3.5.2.5	3-2.				
NTH		VARIABLE	1		1-3.5.2.8					
NTYPE1	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.5	1-3.5.2.6			
NUCAT		VARIABLE	1		1-3.5.2.7					
NUMDOT	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.5				
NV		VARIABLE	1		1-3.5.2.4	1-3.5.2.8	3-11.1			
NX		VARIABLE	1		1-3.5.2.3					
OPEN	F4PLIB	SUBROUTINE	1		1-3.5.1.4	1-3.5.2.9				
OPMISS	PRIVT	SUBROUTINE	1		1-3.5.1.3	1-3.5.1.5				L2-3.13
			2							L2-5.7
OUTFILE.DAT		FILE	1		1-3.5.1.2					
OUTPUT	INALIB	SUBROUTINE	1		1-3.5.2.5	1-3.5.2.6	1-3.5.2.8	1-3.5.2.9		
P		VARIABLE	1		3-3.	3-19.				
PAINT	PRIVT	SUBROUTINE	1							L2-10.
PCTCT	COM2	VARIABLE	1		1-3.4.9					
PCTCTO	COM2	VARIABLE	1		1-3.4.9					
PDATE1	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.X				
PDATE2	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.X				
PDATE3	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.X				

NAME	DESCRIPTION		LINE	COMMENT	OCCURRENCES					LOCATION OF LISTING
PERDO		VARIABLE	1		1-[illegible].8					
PERDU		VARIABLE	1		1[illegible].8					
PERTH		VARIABLE	1		1-[illegible].0					
PERUND		VARIABLE	1		1-[illegible].8					
PFLAB	COM3	VARIABLE	1		1-[illegible]4.3	1-3.4.5	1-3.4.9	1-3.5.2.X		
PLOT		SUBROUTINE	1		1-3.5.2.6					L2-12.3
POLYCR		PROGRAM	1							L2-10.
POLYCR		SUBROUTINE	1							L2-10.3
PRESET	PRIVT	SUBROUTINE	1							L2-1.
PRMUPD	INTRAC	PROGRAM	1	TASK	1-CONTNTS	1-3.3.1	1-3.3.3	1-.3.34	1-3.3.6	L2-21.
			2		1-3.3.2.1	1-3.5.2.X				
PROCED	PRIVT	SUBROUTINE	1							L2-13.3
PTR		VARIABLE	1		1-3.5.1.3	1-3.5.1.5	3-6.			
Q10		SUBROUTINE	1		1-3.5.1.4					L2-1.
R		VARIABLE	1		1-3.5.1.4					
RANDOM	COM6	VARIABLE	1		1-3.4.9	1-3.5.2.6				
RANDOT		VARIABLE	1							L2-3.5
RDCARD	PRIVT	FUNCTION	1		1-3.5.1.3					L2-3.8
RDCLMN	PRIVT	SUBROUTINE	1		1-3.5.2.6					L2-14.2
RDDIR	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.9				L2-2.9.
RDDISK	PRIVT	SUBROUTINE	1		1-3.5.1.1					L2-1.
RDDODU	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.11				L2-2.10
RDDOT	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.10				L2-2.11
RDFLD	PRIVT	SUBROUTINE	1							L2-21.9
RDHEAD	PRIVT	SUBROUTINE	1		1-3.5.1.1					L2-1.
RGODAT		SUBROUTINE	1		1-3.5.2.6					L2-14.1
RDKYD3	PRIVT	SUBROUTINE	1							L2-10.6
RECPCT		VARIABLE	1		1-3.5.2.8					
RECPRN		SUBROUTINE	1		1-3.5.2.8					L2-16.2
RECPRO		PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.8	2-16.			L2-16.
REPORT	PRIVT	SUBROUTINE	1		1-3.5.1.1	1-3.5.1.5	2-15.10			L2-1 +
			2							L2-6.10
			3							L2-14.7
			4							L215.10
REPROP	PRIVT	SUBROUTINE	1	TASK	1-CONTNTS	1-3.5.2.8				L2-16.
REPRTN	PRIVT	SUBROUTINE	1							L2-14.9
REQUES	F4PLIB	SUBROUTINE	1		1-3.5.2.1					
RFAC		VARIABLE	1		1-3.5.2.3					
RIBLD		VARIABLE	1		1-3.5.2.8					
RNUM		VARIABLE	1		1-3.5.2.9					
ROFF	PRIVT	SUBROUTINE	1		1-3.5.2.9					L2-18.1
ROFNO		VARIABLE	1		1-3.5.2.9					
RPTGEN	PRIVT	SUBROUTINE	1		1-3.5.2.X					L2-21.7
RREAD	SHARE	SUBROUTINE	1		1-3.5.1.3	1-3.5.1.5	3-18.			L3-18.
RS		VARIABLE	1		3-19.					
RSIZ		VARIABLE	1		1-3.5.2.3					
RSKIP		VARIABLE	1		1-3.5.2.3					
RWD		VARIABLE	1		3-5.					
S		VARIABLE	1		3-3.					
SCATXY.TMP		FILE	1		1-3.4.8	1-3.5.2.[illegible]	1-3.5.2.5			
SCPLOT	DRIVER	PROGRAM	1	TASK	1-CONTNTS	1-3.5.2.1				L2-12.
SDLINE			1							L2-10.7
SDPNT	PRIVT	SUBROUTINE	1							L2-10.8
SE		VARIABLE	1		1-3.5.2.3					

NAME	DESCRIPTION		LINE	COMMENT	OCCURRENCES					LOCATION OF LISTING
SECNDS	F4PLIB	SUBROUTINE	1		[illegible].1.4	.-3.5.2.5				
SEGDEL	OFFICE	PROGRAM	1	TASK	[illegible] TNTS	1-3.3.3	1-3.[illegible]4	1-3.3.5	1-3.5.1.6	L2-6.
			2							
SEGEND	PRIVT	SUBROUTINE	1		[illegible].1.2	2-2.8				L2-2.8
SEGNO		VARIABLE	1		1-[illegible].2.X	3-6.				
SEGNUM		VARIABLE	1		1-3.5.1.5					
SELDOT	PRIVT	SUBROUTINE	1		1-3.5.2.9					L2-18.2
SETBIT	IMALIB	SUBROUTINE	1		1-3.5.2.5					
SETEF	F4PLIB	SUBROUTINE	1		1-3.5.2.5	1-3.5.2.9				
SETVID		SUBROUTINE	1		1-3.5.2.5					L2-12.4
SETWIN		SUBROUTINE	1		1-3.5.2.5					L2-12.5
SHELL	SHARE	SUBROUTINE	1		1-3.5.2.5	3-19.				L3-19.
SKIP	PRIVT	FUNCTION	1		1-3.5.1.3					L2-3.9
SOILGR	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.5				
SORT	PRIVT	SUBROUTINE	1							L2-10.X
SORTRC	PRIVT	SUBROUTINE	1							L2-5.6
SPWIND	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.5				
SRDISK	PRIVT	SUBROUTINE	1		1-3.5.1.1					L2-1.
SRDISK		VARIABLE	1		1-3.5.1.1					
SS		VARIABLE	1		1-3.5.2.3					
SSSSYYDDD.DAT		FILE	1		1-3.5.1.1					
START		VARIABLE	1		1-3.5.1.3					
STASAV	PRIVT	SUBROUTINE	1		1-3.5.2.X					L2-21.5
STATFIL.TMP		FILE	1		1-3.5.2.2					
STRAYS	PRIVT	SUBROUTINE	1		1-3.5.1.1					L2-1.
STYPE	PRIVT	SUBROUTINE	1		1-3.5.2.5					L2-13.1
SUBCAT	COM1	VARIABLE	1		1-3.4.9	1-3.5.1.4	-13.5.2.	9-		
SUBPOP	COM1	VARIABLE	1		1-3.4.9	1-3.5.1.4				
SUBSTR	SHARE	SUBROUTINE	1		1-3.5.1.3	1-3.5.1.5	1-3.5.1.6	1-3.5.2.X	3-20.	L2-3.12
			2							L2-6. +
			3							L2-13.+
			4							L2-21.6
			5							L3-20.
SUD		VARIABLE	1		1-3.5.2.3					
SUNAZ	COM2	VARIABLE	1		1-3.4.9					
SUNEL	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.6				
SWCLR	PRIVT	SUBROUTINE	1							L2-10.9
T2	PRIVT	SUBROUTINE	1	T2DRMOD						
T2DR	PRIVT	SUBROUTINE	1							L2-1.
TABL.:	SHARE	SUBROUTINE	1		3-4.					L3-4.
TAPSCN	PRIVT	SUBROUTINE	1		1-3.5.1.1					L2-1.
TC		VARIABLE	1		1-3.5.2.7	3-13.				
TCHLST		SUBROUTINE	1		1-3.5.2.3					L2-9.3
TCLANM.MAP		FILE	1		1-3.5.1.5					
TCLUNM.MAP		FILE	1		1-3.5.1.5					
TDATE1	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.X				
TIATE2	COM2	VARIABLE	1		1-3.4.9	1-3.5.2.X				
TDATE3	COM2	VARIABLE	1		1-3.4.9	1-3.5.1.3	1-3.5.1.4	1-3.5.2.X		
TDIS		VARIABLE	1		1-3.5.2.6					
THLOPM		SUBROUTINE	1	TASK	1-CONTNTS	1-3.5.2.5				L2-12.8

ORIGINAL PAGE IS
OF POOR QUALITY

NAME	DESCRIPTION		LINE	COMMENT	OCCURRENCES					LOCATION OF LISTING
THMLOP	SHARE	SUBROUTINE	1		1-[illegible]NTS	3-21.				L3-21.
TI		VARIABLE	1		[illegible]					
TIME	F4PLIB	SUBROUTINE	1		1-[illegible]1.4	1-3.5.2.5	1-3.5.2.6	1-3.5.2.8	1-3.5.2.9	
TOBDON		VARIABLE	1		[illegible]1.1					
TP		VARIABLE	1		1[illegible]5.2.3					
TP1XL		VARIABLE	1		1-3.5.2.9					
TPCT		VARIABLE	1		1-3.5.2.9					
TRAJPL	PRIVT	SUBROUTINE	1		1-3.5.2.5					L2-13.8
TSTAT.DAT		FILE	1		1-3.5.1.5					
TTX2		VARIABLE	1		1-3.5.2.3					
TTX1		VARIABLE	1		1-3.5.2.3					
TTY1		VARIABLE	1		1-3.5.2.3					
TTY2		VARIABLE	1		1-3.5.2.3					
TUN		VARIABLE	1		1-3.5.2.3					
TUNIT		VARIABLE	1		1-3.5.2.3					
TWRITE	SHARE	SUBROUTINE	1		1-3.5.2.7	2-15.6	3-22.			L2-15.6
			2							L3-22.
TX		VARIABLE	1		1-3.5.2.3	1-3.5.2.7				
TX1	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.5	1-3.5.2.7		
TX2	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.5	1-3.5.2.7		
TY		VARIABLE	1		1-3.5.2.3	1-3.5.2.7				
TY1	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.5	1-3.5.2.7		
TY2	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.3	1-3.5.2.5	1-3.5.2.7		
TYPE	COM5	VARIABLE	1		1-3.4.4	1-3.4.9	1-3.5.2.5	1-3.5.2.6	1-3.5.2.9	
			2		1-3.5.2.X					
TYPE	COM5	VARIABLE	1		1-3.5.2.6	1-3.5.2.9	1-3.5.2.X			
TYPE1		VARIABLE	1		1-3.5.1.3					
TYP1		VARIABLE	1		1-3.5.1.3					
UFLAG1	COM3	VARIABLE	1		1-3.4.9	1-3.5.2.X				
UFLAG2	COM3	VARIABLE	1		1-3.4.9	1-3.5.2.4	1-3.5.2.X			
UFLAG3	COM3	VARIABLE	1		1-3.4.4	1-3.4.9	1-3.5.2.X			
UFLAG4	COM4	VARIABLE	1		1-3.4.9	1-3.5.2.X				
UNBIAS	PRIVT	SUBROUTINE	1		1-3.5.2.9					L2-18.3
UNCDIS	PRIVT	SUBROUTINE	1		1-3.5.2.7	2-15.2				L2-15.2
UNID		VARIABLE	1		1-3.5.2.8					
UNIT		VARIABLE	1		3-5.					
UNLDOT	PRIVT	SUBROUTINE	1		1-3.5.2.X					L2-3
UNPAK	PRIVT	FUNCTION	1		1-3.5.1.3					L2-3.1
UNPAK		VARIABLE	1		1-3.5.1.3					
UPCT		VARIABLE	1		1-3.5.2.9					
UPDATE	PRIVT	SUBROUTINE	1		1-3.5.1.3	1-3.5.2.6				L2-3.6
UPDOT	PRIVT	SUBROUTINE	1		1-3.5.1.2	2-2.15				L2-2.15
V		VARIABLE	1		1-3.5.2.3	3-9.				
VALCK	PRIVT	SUBROUTINE	1							L2-12.6
VAR	COM2	VARIABLE	1		1-3.4.9					
VAR0	COM2	VARIABLE	1		1-3.4.9					
VDALTR	SHARE	SUBROUTINE	1		1-3.5.2.5	3-23.				L3-23.
VERTEX		VARIABLE	1		1-3.5.2.8	3-11.1				
WAIT	IMALIB	SUBROUTINE	1		1-3.5.2.5					
WAITFR	F4PLIB	SUBROUTINE	1		1-3.5.1.4	1-3.5.2.1				
WC		VARIABLE	1		1-3.5.2.3					
WINDER	SHARE	SUBROUTINE	1		1-CONTNTS	3-24.				L3-24.

NAME	DESCRIPTION	LINE	COMMENT	OCCURRENCES					LOCATION OF LISTING
WINDRM	PROGRAM	1							L2-12.7
WINDRM	SUBROUTINE	1	TASK	1-CONTNTS	1-3.5.2.5				L2-12.
WORK	VARIABLE	1		1-3.5.1.3					
WRDIR	PRIVT SUBROUTINE	1		1-3.5.1.2	2-2.14				L2-2.14
WROODU	PRIVT SUBROUTINE	1		1-3.5.1.2	2-2.12				L2-2.12
WRDCT	PRIVT SUBROUTINE	1		1-3.5.1.2	2-2.13				L2-2.13
X	VARIABLE	1		1-3.5.2.3	1-3.5.2.8				
XPCT	VARIABLE	1		1-3.5.2.9					
XXXXDOOTS.DAT	FILE	1		1-3.5.2.X					
XXXXDOTS.DAT	FILE	1		1-3.3	1-3.3.3	1-3.4.4	1-3.5.1.3	1-3.5.2.2	
XXXXFIELD.DAT	FILE	1		1-3.3	1-3.3.4	1-3.5.1.2	1-3.5.2.2	1-3.5.2.X	
XXXXPCLAS.MAP	FILE	1		1-3.3	1-3.3.5	1-3.5.2.2	1-3.5.2.X		
XXXXPCLUS.MAP	FILE	1		1-3.3	1-3.3.5	1-3.4.3	1-3.5.2.2	1-3.5.2.X	
XXXXPSTAT.DAT	FILE	1		1-3.3	1-3.3.5	1-3.4.5	1-3.5.2.2	1-3.5.2.X	
XXXXTCLAS.MAP	FILE	1		1-3.3	1-3.3.5	1-3.5.2.2	1-3.5.2.X		
XXXXTCLUS.MAP	FILE	1		1-3.3	1-3.3.5	1-3.4.3	1-3.5.2.2	1-3.5.2.X	
XXXXTSTAT.DAT	FILE	1		1-3.3	1-3.3.5	1-3.4.5	1-3.5.1.4	1-3.5.2.2	
		2		1-3.5.2.X					
XXXXY DDD.DAT	FILE	1		1-3.3					
XZ	VARIABLE	1		1-3.5.2.3					
Y	VARIABLE	1		1-3.5.2.3	1-3.5.2.5				
YLINE	VARIABLE	1		3-11.2					
YR	VARIABLE	1		1-3.5.1.2	1-3.5.1.3	1-3.5.1.5	1-3.5.2.X		
YZ	VARIABLE	1		1-3.5.2.3					
ZNAME	PRIVT SUBROUTINE	1							L2-5.12
ZD	VARIABLE	1		1-3.5.2.7					
ZOOM	PRIVT SUBROUTINE	1		1-3.5.2.3	1-3.5.2.7	1-3.5.2.8	2-15.9	3-25.	L2-15.9
		2							+L3-25.