

## N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM  
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT  
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED  
IN THE INTEREST OF MAKING AVAILABLE AS MUCH  
INFORMATION AS POSSIBLE

"Made available under NASA sponsorship  
in the interest of early and wide dis-  
semination of Earth Resources Survey  
Program information and without liability  
for any use made thereof."

8.0-10252  
JSC-14487

"AS-BUILT" DESIGN SPECIFICATION  
FOR  
FIELD STATISTICS (FIELDSTAT)

160685

Job Order 71-475

(TIRE 78-0011)

(E80-10252) AS-BUILT DESIGN SPECIFICATION  
FOR FIELD STATISTICS (FIELDSTAT) (Lockheed  
Electronics Co.) 99 p HC A05/ME A01

N80-30835

CSCL 05B

63/43

Unclas  
00252

Prepared By

Lockheed Electronics Company, Inc.

Systems and Services Division

Houston, Texas

Contract NAS 9-15200

For

EARTH OBSERVATIONS DIVISION

SPACE AND LIFE SCIENCES DIRECTORATE



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

September 1978

LEC-12804

JSC-14487

"AS-BUILT" DESIGN SPECIFICATION  
FOR  
FIELD STATISTICS (FIELDSTAT)

Job Order 71-475  
(TIRF 78-0011)

Prepared By  
D. K. McCarley

APPROVED BY

*for* James A. Wilkerson  
Philip L. Krumm, Acting Supervisor  
Scientific Applications Section

Prepared By  
Lockheed Electronics Company, Inc.  
For  
Earth Observations Division

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

September 1978

LEC-12804

## CONTENTS

Section	Page
1. INTRODUCTION. . . . .	1-1
2. APPLICABLE DOCUMENTS . . . . .	2-1
3. SYSTEM DESCRIPTION. . . . .	3-1
3.1 <u>GENERAL</u> . . . . .	3-1
3.2 <u>DATA FLOW DIAGRAMS</u> . . . . .	3-2
3.2.1 IMUNLD DATA FLOW . . . . .	3-2
3.2.2 FLDSTT DATA FLOW . . . . .	3-2
3.3 <u>FILE DESCRIPTIONS</u> . . . . .	3-2
3.3.1 IMAGE UNLOAD TAPE FILE (Input only). . . . .	3-2
3.3.2 SCREENED IMAGE DISK FILE - SGNMACQDT.IMØ (Output/Input) . . . . .	3-6
3.3.3 IMUNLD.DAT DISK FILE (Input only) . . . . .	3-6
3.3.4 HEDREC.SIT DISK FILE (Input only) . . . . .	3-7
3.3.5 FLDSTT.DAT DISK FILE (Input only) . . . . .	3-7
3.3.6 FSTLBL.DAT DISK FILE (Input only) . . . . .	3-9
3.3.7 GROUND TRUTH DISK FILE - SGNMGENDT.GTØ (Input only) . . . . .	3-9
3.3.8 DOT PIXEL STATISTICS DISK FILE - SGNMACQDT.DPØ (Output only). . . . .	3-11
3.3.9 PURE PIXEL STATISTICS DISK FILE - SGNMACQDT.PPØ (Output only). . . . .	3-14
3.3.10 MIXED PIXEL STATISTICS DISK FILE - SGNMACQDT.MPØ (Output only). . . . .	3-15
3.3.11 SUBPIXEL STATISTICS DISK FILE - SGNMACQDT.SPØ (Output only). . . . .	3-15
3.3.12 SCRATCH FILE - TMPSTT.TMP (Input/Output). . . . .	3-17

Section	Page
3.4 DETAILED DESIGN. . . . .	3-19
3.4.1 IMAGE UNLOAD MAIN PROGRAM (IMUNLD). . . . .	3-19
3.4.2 IMUNLD SUBROUTINES . . . . .	3-24
3.4.3 FIELDSTATS MAIN PROGRAM (FLDSTT) . . . . .	3-45
3.4.4 FLDSTT SUBROUTINES . . . . .	3-71
4. OPERATIONS. . . . .	4-1
4.1 <u>OPERATOR'S GUIDE</u> . . . . .	4-1
4.1.1 IMUNLD PROGRAM . . . . .	4-1
4.1.2 FLDSTT PROGRAM . . . . .	4-1
4.2 <u>USER'S GUIDE</u> . . . . .	4-2
4.2.1 BATCH PROCEDURES . . . . .	4-2
4.2.2 ONLINE PROCEDURES . . . . .	4-5

## 1. INTRODUCTION

This document describes the implementation details of the Field Statistics (FIELDSTAT) Program which is designed to provide statistics describing the signatures of the crop classes (as determined by ground truth) in a Large Area Crop Inventory Experiment (LACIE) segment. Included are details on the Image Unload Program which transfers raw image data from magnetic tapes to a disk data base during a screening and tagging process. The screened image data and existing ground truth data are correlated by the FIELDSTAT program and ground truth spectral value statistics files are produced.

## 2. APPLICABLE DOCUMENTS

The following documents, of the exact issue noted, formed the basis of the software described in this document.

- o TIRF 78-0011 dated February 3, 1978.
- o Field Statistics (FIELDSTAT) Implementation Specification Document for Accuracy Assessment (LEC 12172) dated May, 1978.

### 3. SYSTEM DESCRIPTION

#### 3.1 GENERAL

The first program run is the Image Unload program, IMUNLD. It reads from a disk file named IMUNLD.DAT three items of information: the auxiliary disk device number, the magnetic tape device ID, and the magnetic tape number of the image unload tape. It then reads the blind site segment numbers from a disk file named HEDREC.SIT. Then it begins scanning the input tape and, when a desired segment is encountered, a disk file is created whose type is IMØ and whose name is the nine-character segment number and acquisition date combination. Next, each of the 117 scan line records is transferred from the tape to the disk file and, during the transfer, each of the 196 pixels in a scan-line is passed through a screening process. If the pixel is not classified as good, it is tagged before being stored on the output disk file. When all of that file has been transferred, the disk file is closed. A disk file is created for each acquisition on the tape from a blind site segment. A report is produced which identifies each disk file created, the portion of the three-part data base where it was stored, and the results of the screening process.

The second program that is run is the Fieldstats program, FLDSTT. The FLDSTT and IMUNLD programs are independent except that FLDSTT uses as input the files that are output from IMUNLD. FLDSTT reads from a disk file named FLDSTT.DAT several items of information: the crop code translations file ID, the ground truth data file ID, the LACIE segment number, an output report option flag, and from one to fifteen acquisition dates. The program reads crop code translations and brightness and greenness coefficients from the crop code translations file. It assigns the specified ground truth data file, and then,



for each acquisition date input, that date is combined with the segment number to create the file names used for one input disk file (image unload - type. IMØ) and four output disk files. The four files output are the dot pixel statistics (type.DPØ), the pure pixel statistics (type.PPØ), the mixed pixel statistics (type.MPØ), and the subpixel statistics (type.SPØ). One scan line of data is read from the image unload file and from the ground truth file. For each of the 196 pixels in the scan line, counts and accumulations are made (on a per crop code basis) and kept on an interim scratch disk file. When all 117 scan lines have been processed, the data for the four output files are calculated and stored, and the desired output reports are generated. Then the process is repeated for the next acquisition date.

### 3.2 DATA FLOW DIAGRAMS

#### 3.2.1 IMUNLD DATA FLOW

Figure 3-1 illustrates the data flow for the IMUNLD program.

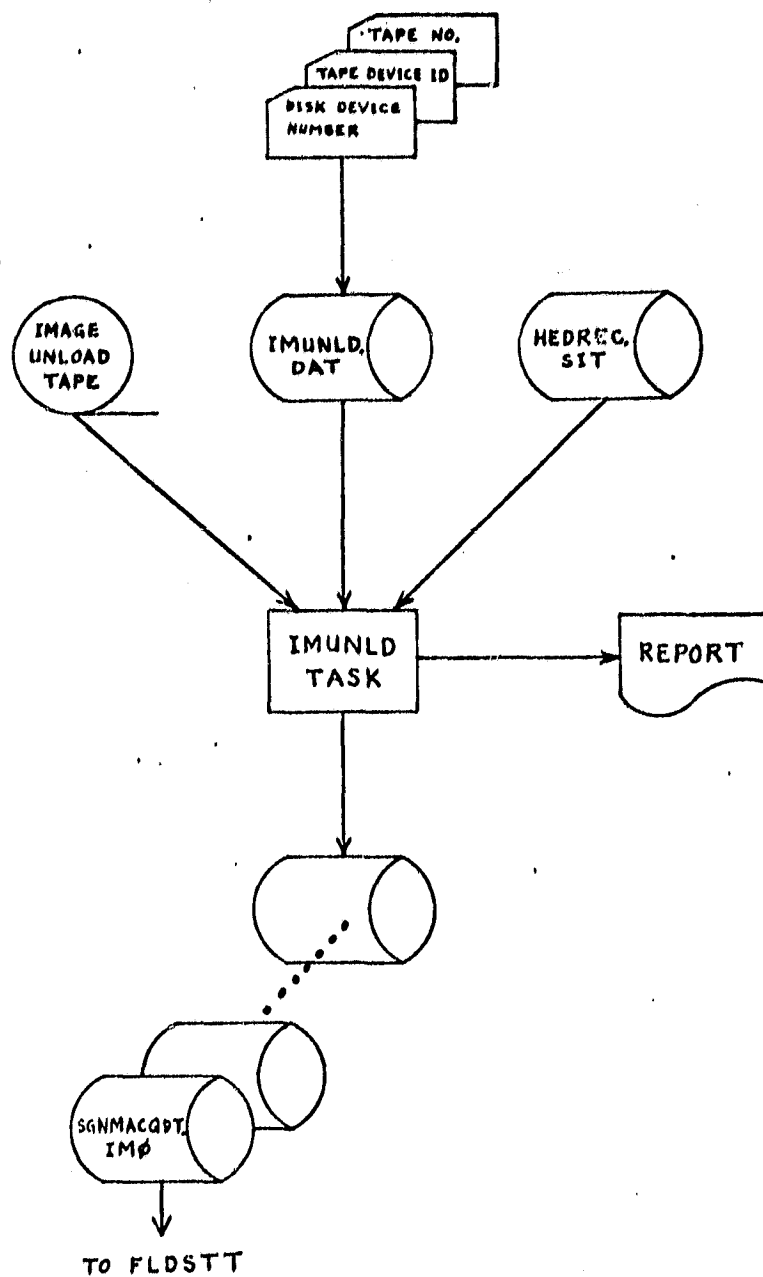
#### 3.2.2 FLDSTT DATA FLOW

Figure 3-2 illustrates the data flow for the FLDSTT program.

### 3.3 FILE DESCRIPTIONS

#### 3.3.1 IMAGE UNLOAD TAPE FILE (Input only)

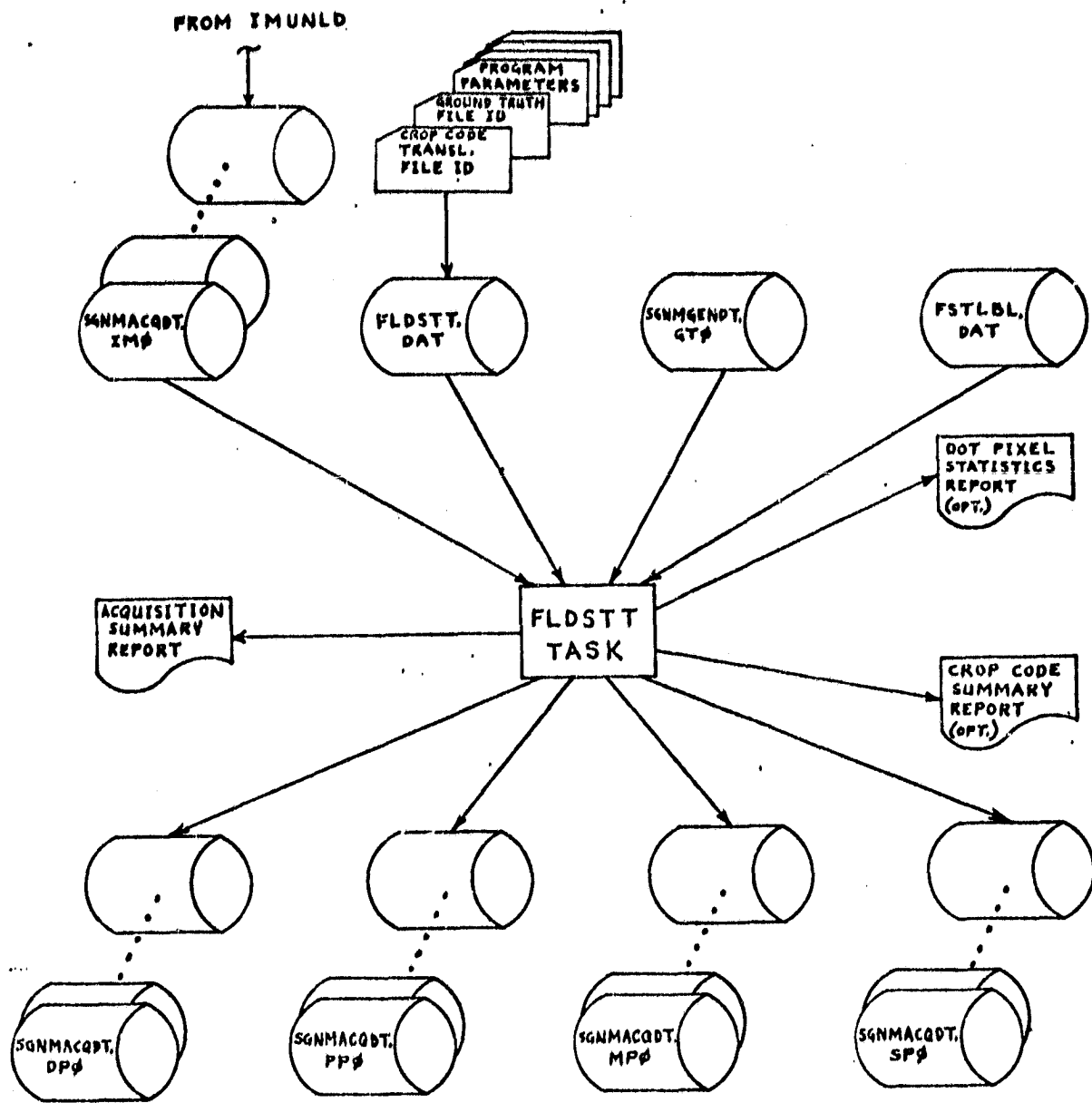
This tape contains the four channels of image data for an acquisition in one file. There may be as many as 100 files on the tape. Each file is in the Universal Format, consisting of one 3060-byte header record, 117 900-byte data records, and one end-of-file record. The end of tape is denoted by two consecutive end-of-file records. The information of interest in the header is described below.



IMUNLD DATA FLOW

FIGURE 3-1

~~3-3~~  
5



FLDSTT DATA FLOW

FIGURE 3-2

<u>BYTE</u>	<u>CONTENTS</u>
1-32	Tape title in EBCDIC: LACIE PFC IMAGE UNLOAD TAPE
33-38	Tape number in EBCDIC
39-60	N/A
61-63	Date of this tape generation
61	Day of month in binary
62	Month number in binary
63	Last two digits of year in binary
64-66	N/A
67-68	Segment number in binary (bytes inverted)
69-2200	N/A
2201-2202	Sun angle in degrees in binary (bytes inverted)
2203-2248	N/A
2249-2253	Acquisition date in EBCDIC in YYDDD format
2254-3060	N/A

The information of interest in each data record (which represents one scan line) is described below.

<u>BYTE</u>	<u>CONTENTS</u>
1-72	N/A
73-268	Channel one values for pixels 1-196 in binary
269-464	Channel two values
465-660	Channel three values
661-856	Channel four values
857-900	N/A

### 3.3 : SCREENED IMAGE DISK FILE - SGNMACQDT.IMØ (Output/Input)

The screened image file is a sequential, unformatted, FILES 11 disk file in Universal Format and is essentially a copy of the corresponding (segment number and acquisition date) file on the image unload tape. The distinction in the data records is that the channel four value for a pixel (which normally ranges from 0-127) is increased by 128 if the screening process determines that the pixel is not good. The header record is copied exactly from the tape with the following exceptions:

- a. The new EBCDIC file title (bytes 1-32) is FIELDSTAT SCREENED IMAGE FILE.
- b. The segment number bytes (67 and 68) are in their proper order.
- c. The current date is stored in bytes 61-63.

#### 3.3.3 IMUNLD.DAT DISK FILE (Input only)

This sequential, formatted disk file is a card-image file used to supply general information for any particular run of the IMUNLD program. Its format is as follows:

<u>RECORD</u>	<u>CHARACTER</u>	<u>CONTENTS</u>
1	1	Device number of the disk onto which the IMUNLD is to store the screened image files. For example, if DB2: is the disk device used, then this record would contain a 2.
2	1-3	Magnetic tape device ID for input Image Unload tape, e.g., MT1.
3	1-6	The Image Unload tape number, e.g., 770080.

### 3.3.4 HEDREC.SIT DISK FILE (Input only)

This sequential, unformatted disk file contains all the valid blind site segment numbers in one record as binary integers. The first 64 words are the 64 segments pertaining to Part 1 of the Accuracy Assessment (AA) data base. The second 64 words are the 64 segments pertaining to Part 2 of the AA data base, and the third 64 words are the 64 segments pertaining to Part 3 of the AA data base.

### 3.3.5 FLDSTT.DAT DISK FILE (Input only)

This sequential, formatted disk file is a card-image file used to supply general information for any particular run of the FLDSTT program. Its format is as follows:

<u>RECORD</u>	<u>CHARACTER</u>	<u>CONTENTS</u>
1	1-30	File ID where the crop code translations and brightness and greenness coefficients are stored. The file name is FSTLBL.DAT, but this field should include the specific device and User File Directory (UFD) to enable location of the desired file, e.g., DB2:[131,001]FSTLBL.DAT.
2	1-30	File ID where the ground truth data are stored. After this file is assigned, then this character string is used to create output files, therefore it must be very exact in format. For example, the file ID DB2:[131,001]161177353.GT0 is acceptable (the acquisition date must be characters 18-22).
3	1-4	Segment number to be processed.

<u>RECORD</u>	<u>CHARACTER</u>	<u>CONTENTS</u>
3	6-7	Output report option flag. A two-digit decimal number specifying which output reports are desired. The number is the sum of the numbers assigned to the various reports. The Acquisition Summary report is always produced, whereas the Dot Pixel Statistics report (assigned number 1) and the Crop Code Summary report (assigned number 2) are optional.
3	9-10	A two-digit decimal number, greater than zero and less than sixteen, which specifies how many acquisition dates are to be processed during this run of the FLDSTT program.
4	1-5	First acquisition date to be processed. Represented in standard YYDDD format. As many acquisition dates as are indicated on record three should be specified in successive fields on records four, five, and six.
4	11-15	Second acquisition date.
4	21-25	Third acquisition date.
4	31-35	Fourth acquisition date.
4	41-45	Fifth acquisition date.
5	1-5	Sixth acquisition date.
5	11-15	Seventh acquisition date.
5	21-25	Eighth acquisition date.
5	31-35	Ninth acquisition date.

<u>RECORD</u>	<u>CHARACTER</u>	<u>CONTENTS</u>
5	41-45	Tenth acquisition date.
6	1-5	Eleventh acquisition date.
6	11-15	Twelfth acquisition date.
6	21-25	Thirteenth acquisition date.
6	31-35	Fourteenth acquisition date.
6	41-45	Fifteenth acquisition date.

### 3.3.6 FSTLBL.DAT DISK FILE (Input only)

This sequential, formatted disk file is a card image file used to supply crop code translations and brightness and greenness coefficients to the FLDSTT program. The first sixteen cards each contain sixteen five-character fields. The first three characters of each field are the numeric crop code, and the last two characters are the alphanumeric translations. These fields must be in ascending numerical sequence, 000-255. The seventeenth card contains the four brightness coefficients beginning at character one and separated by commas. The eighteenth card contains the four greenness coefficients beginning at character one and separated by commas.

### 3.3.7 GROUND TRUTH DISK FILE - SGNMGENDT.GTØ (Input only)

The ground truth file is a sequential, unformatted, FILES 11 disk file in Universal format, with one 3060-byte header record and 351 540-byte data records. The only information in the header record that is of interest to the FLDSTT program is the segment number, stored in word thirty-four, which is checked against the specified segment number from the FLDSTT.DAT file.

In ground truth data, which records actual crop codes, each pixel is divided into six subpixels (three rows by two columns), so that one scan line across the 196 pixels actually records values for 2\*196 subpixels, and three scan lines are needed to record values



for the complete 196 pixels in each of the 117 pixel rows. Consequently, there are  $3 \times 117 = 351$  data records, each of which contains crop code values for two subpixels out of each of the 196 pixels in the scan line. The information in the data records is laid out as follows:

<u>RECORD</u>	<u>BYTE</u>	<u>CONTENTS</u>
1	1-72	N/A
1	73	Crop code for upper left subpixel of pixel one in row one.
1	74	Crop code for upper right subpixel of pixel one in row one.
1	75	Crop code for upper left subpixel of pixel two in row one.
.	.	.
.	.	.
.	.	.
1	464	Crop code for upper right subpixel of pixel 196 in row one.
1	465-540	N/A
2	1-72	N/A
2	73	Crop code for middle left subpixel of pixel one in row one.
.	.	.
.	.	.
.	.	.
2	464	Crop code for middle right subpixel of pixel 196 in row one.
2	465-540	N/A
3	1-72	N/A

<u>RECORD</u>	<u>BYTE</u>	<u>CONTENTS</u>
3	73	Crop code for lower left subpixel of pixel one in row one.
.	.	.
.	.	.
.	.	.
3	464	Crop code for lower right subpixel of pixel 196 in row one.
3	465-540	N/A
4	1-72	N/A
4	73	Crop code for upper left subpixel of pixel one in row two.
.	.	.
.	.	.
.	.	.
351	464	Crop code for lower right subpixel of pixel 196 in row 117.
351	465-540	N/A

It should be noted that the value of each crop code is biased by -128 if looked at as an eight-bit computer word.

### 3.3.8 DOT PIXEL STATISTICS DISK FILE - SGNMACQDT.DPØ (Output only)

The dot pixel statistics file is a sequential, unformatted, FILES 11 disk file with one 360-word header (two bytes per word) and eleven 270-word data records. The format of the header record is as follows:

<u>WORD</u>	<u>CONTENTS</u> (binary single word integer except as noted)
1-5	FIELDSTATS (ASCII characters)

<u>WORD</u>	<u>CONTENTS</u>
6-8	Date of file generation: Word 6 = day of the month Word 7 = month number Word 8 = value of the last two digits of the year
9	Segment number
10-11	Acquisition date (double word binary integer)
12	Sun angle in degrees
13	Number of good pixels
14	Number of garbled pixels
15	Number of cloud pixels
16	Number of shadow pixels
17	Number of water pixels
18-19	Channel one means (real)
20-21	Channel two means (real)
22-23	Channel three means (real)
24-25	Channel four means (real)
26-27	Gamma (real)
28	Number of one-class pixels
29	Number of two-class pixels
30	Number of three-class pixels
31	Number of four-class pixels
32	Number of five-class pixels
33	Number of six-class pixels
34	Number of different crop codes encountered in good pixels.

<u>WORD</u>	<u>CONTENTS</u>
35-290	Crop code occurrence array. Applicable only to the pure pixel, mixed pixel, and subpixel statistics files. Each word corresponds to crop codes 0-255, in that order. A zero in the word means that crop code did not occur. A value, I, in the Jth word means that crop code J-1 did occur, and the statistics for that crop code are recorded in record I (header record excluded) of the file.
291-360	Zero fill.

Each data record contains nineteen twelve-word blocks of information in the first 228 words of the record and zero fill in the last forty-two words. Each block of information pertains to one of the designated 209 dot pixels and is laid out as follows:

<u>WORD</u>	<u>CONTENTS</u> (all are single-word binary integers)
1	Dot ID, range 1-209
2	Channel one spectral value, range 0-255
3	Channel two spectral value, range 0-255
4	Channel three spectral value, range 0-255
5	Channel four spectral value, range 0-127
6	Subpixel one crop code, range 0-255
7	Subpixel two crop code, range 0-255
8	Subpixel three crop code, range 0-255
9	Subpixel four crop code, range 0-255
10	Subpixel five crop code, range 0-255
11	Subpixel six crop code, range 0-255
12	Screening label, range 1-47

### 3.3.9 PURE PIXEL STATISTICS DISK FILE - SGNMACQDT.PPØ (Output only)

The pure pixel statistics file is a sequential, unformatted, FILES 11 disk file with one 360-word header record and from zero to 256 630-word data records. The header record is identical to the dot pixel statistics file header record. There will be one data record for each different crop code found in the good pixels of the segment. A pure pixel is a good pixel whose subpixels all have the same crop code. For a crop code detected in the segment, but having no pure pixels with that crop code, the data record will contain all zeroes except the first word, which will contain the crop code value. For crop codes that did have pure pixels, the data records have the following format:

<u>WORD</u>	<u>CONTENT</u> (single word binary integer except as noted)
1	Crop code
2-3	Channel one means (real)
4-5	Channel two means (real)
6-7	Channel three means (real)
8-9	Channel four means (real)
10-11	Channel one standard deviation (real)
12-13	Channel two standard deviation (real)
14-15	Channel three standard deviation (real)
16-17	Channel four standard deviation (real)
18-19	Channels 1x2 correlation (real)
20-21	Channels 1x3 correlation (real)
22-23	Channels 1x4 correlation (real)
24-25	Channels 2x3 correlation (real)
26-27	Channels 2x4 correlation (real)

<u>WORD</u>	<u>CONTENT</u>
28-29	Channels 3x4 correlation (real)
30	Pixel count
31-181	Brightness histogram, 151 words representing counts of values 0-150
182-297	Greenness histogram, 116 words representing counts of values (-30) -85
298-630	Zero fill.

### 3.3.10 MIXED PIXEL STATISTICS DISK FILE - SGNMACQDT.MPØ (Output only)

The mixed pixel statistics file is a sequential, unformatted, FILES 11 disk file with one 360-word header record and from zero to 256 630-word data records. The header record is identical to the dot pixel statistics file header record. There will be one data record for each different crop code found in the good pixels of the segment. A mixed pixel for a certain crop code is a good pixel with at least one subpixel having that crop code. Note that this definition includes pure pixels as a subset of mixed pixels. Each data record has exactly the same format as a pure pixel statistics file data record.

### 3.3.11 SUBPIXEL STATISTICS DISK FILE - SGNMACQDT.SPØ (Output only)

The subpixel statistics file is a sequential, unformatted, FILES 11 disk file with one 360-word header record and from zero to 256 630-word data records. The header record is identical to the dot pixel statistics file header record. There will be one data record for each different crop code found in the good pixels of the segment. Each data record will have the following format:

<u>WORD</u>	<u>CONTENTS</u> (double word binary integer except as noted)
1	Crop code (single word binary integer)
2-3	Channel one means (real)
4-5	Channel two means (real)
6-7	Channel three means (real)
8-9	Channel four means (real)
10-11	Channel one standard deviation (real)
12-13	Channel two standard deviation (real)
14-15	Channel three standard deviation (real)
16-17	Channel four standard deviation (real)
18-19	Channels 1x2 correlation (real)
20-21	Channels 1x3 correlation (real)
22-23	Channels 1x4 correlation (real)
24-25	Channels 2x3 correlation (real)
26-27	Channels 2x4 correlation (real)
28-29	Channels 3x4 correlation (real)
30-31	Subpixel count
32-333	Brightness histogram, 151 integers representing counts of values 0-150
334-565	Greenness histogram, 116 integers representing counts of values (-30) -85
566-630	Zero fill

### 3.3.12 SCRATCH FILE - TMPSTT.TMP (Input/Output)

The program FLDSTT uses a direct access, unformatted, scratch disk file, that has 1280 words per record. The first 256 records are reserved for interim storage of counts and accumulations on a per crop code basis. Records 257 through 267 are used for interim storage of the dot pixel statistics file data records and are formatted exactly like those records. The format of any one of the 256 crop code statistics records is as follows:

<u>WORD</u>	<u>CONTENTS</u> (single word binary integers except as noted)
1	Crop code
2,3	Sum of pure pixel channel one values (double word binary integer (DWBI))
4,5	Sum of pure pixel channel two values (DWBI)
6,7	Sum of pure pixel channel three values (DWBI)
8,9	Sum of pure pixel channel four values (DWBI)
10,11	Sum of channels 1x1 products (DWBI) - pure pixels
12,13	Sum of channels 1x2 products (DWBI) - pure pixels
14,15	Sum of channels 1x3 products (DWBI) - pure pixels
16,17	Sum of channels 1x4 products (DWBI) - pure pixels
18,19	Sum of channels 2x2 products (DWBI) - pure pixels
20,21	Sum of channels 2x3 products (DWBI) - pure pixels
22,23	Sum of channels 2x4 products (DWBI) - pure pixels
24,25	Sum of channels 3x3 products (DWBI) - pure pixels
26,27	Sum of channels 3x4 products (DWBI) - pure pixels
28,29	Sum of channels 4x4 products (DWBI) - pure pixels
30	Pure pixel count



<u>WORD</u>	<u>CONTENTS</u>
31-181	Pure pixel brightness counters, 151 integers representing values 0-150
182-297	Pure pixel greenness counters, 116 integers representing values (-30) -85
298-593	Repeat of words 2-297 except for mixed pixels
594-621	Repeat of words 2-29 except for subpixels
622-1157	Repeat of words 30-297 except DWBI for subpixels.

### 3.4 DETAILED DESIGN

#### 3.4.1 IMAGE UNLOAD MAIN PROGRAM (IMUNLD)

##### 3.4.1.1 Linkage

This program calls the special subroutines BEGRPT, FBLD, and FSTART in addition to system subroutines.

##### 3.4.1.2 Interface

BEGRPT, FBLD, FSTART

##### 3.4.1.3 Input

An Image Unload tape, a blind site segment numbers disk file, and a program parameters disk file.

##### 3.4.1.4 Output

Screened image disk files for use by at least the FLDSTT program, and a report showing, for each screened image disk file produced, the file name, the part of the AA data base into which it was stored, and the count of good, garbled, cloud, shadow, and water pixels.

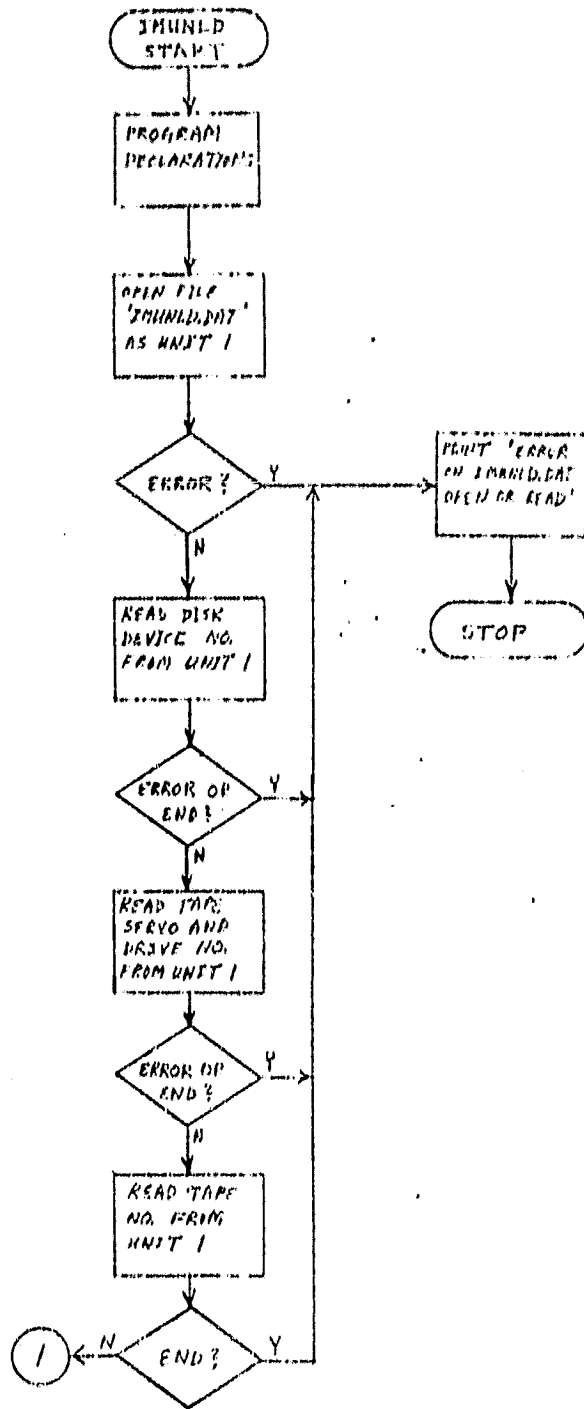
##### 3.4.1.5 Storage

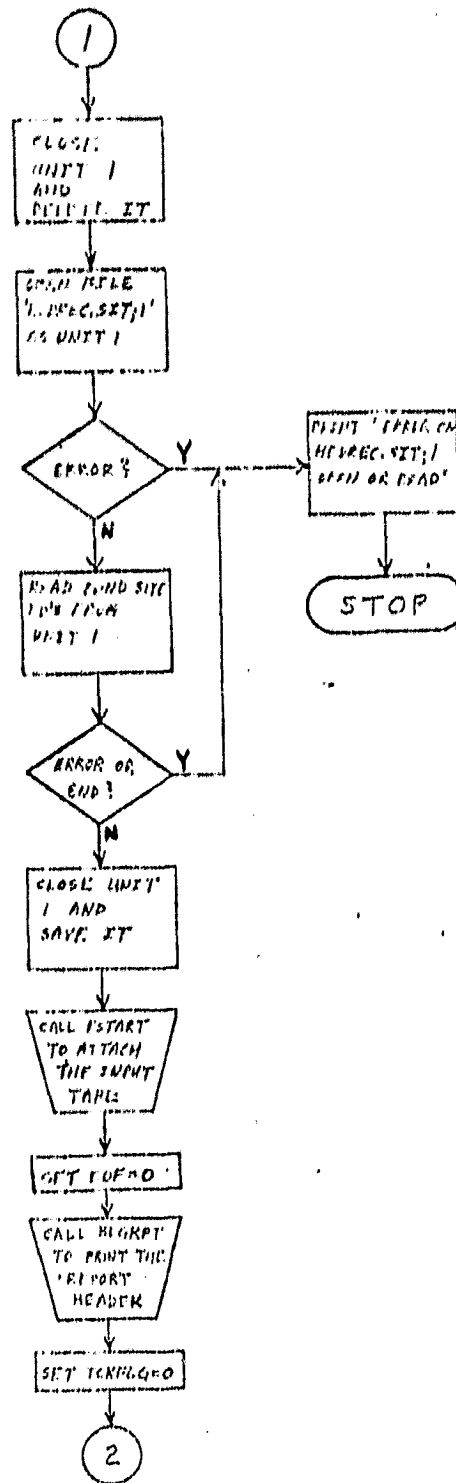
15,334 words.

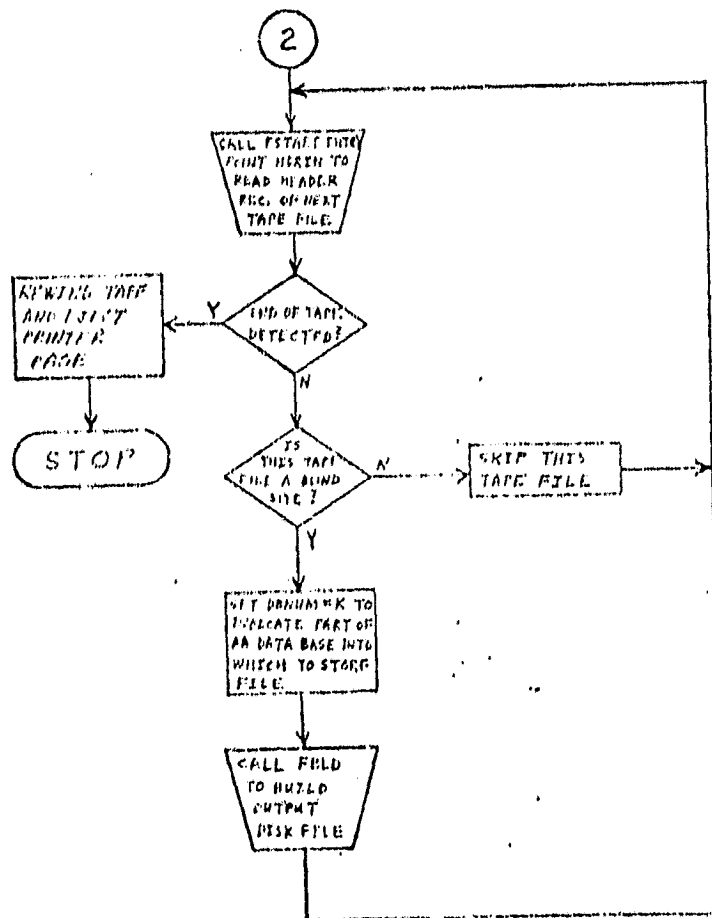
##### 3.4.1.6 Description

This program scans files on an Image Unload tape and, for each blind site segment it finds, it creates a disk file of the same data after screening each pixel and flagging those which are not good.

### 3.4.1.7 Flowchart







### 3.4.1.8 Listing

```

PROGRAM IMUNLD
C
C THIS PROGRAM TRANSFERS IMAGE FILED BRIND SITE DATA FROM
C TAPE TO DISK, SCREENING EACH PIXEL AND PARKING THE 22N-G-2D
C DATA BY ADDRESS 128 TO THE CHANNEL FROM VALUE.
C
      IMPLICIT INTEGER(A-Z)
      PARAMETER (SIZE=64, MAXDSK=3
      LOGICAL L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19, L20, L21, L22, L23, L24, L25, L26, L27, L28, L29, L30, L31, L32, L33, L34, L35, L36, L37, L38, L39, L40, L41, L42, L43, L44, L45, L46, L47, L48, L49, L50, L51, L52, L53, L54, L55, L56, L57, L58, L59, L60, L61, L62, L63, L64, L65, L66, L67, L68, L69, L70, L71, L72, L73, L74, L75, L76, L77, L78, L79, L80, L81, L82, L83, L84, L85, L86, L87, L88, L89, L90, L91, L92, L93, L94, L95, L96, L97, L98, L99, L100)
      COMMON /SERV/J, ACCTE, TAPN, (3), DEVICE
      DIMENSION SITE(SIZE, MAXDSK), SERVAR(2)
C
C
      OPEN(UNIT=1, NAME='IMUNLD.DAT', TYPE='OLD', ERR=3)
      GO TO 5
      CONTINUE
      PRINT 4000
4000  FORMAT(' ERROR ON IMUNLD.DAT OPEN OR READ'/1H1)
      STOP 'IMUNLD.DAT ERR'
      CONTINUE
      READ(1,7,ERR=3,ERR=3)DEVICE
      FORMAT(A1)
      READ(1,10,ERR=3,ERR=3)SERVAR, DRIVE
      FORMAT(A2,11)
      READ(1,20,ERR=3)TAPN
      FORMAT(3A2)
C
      CLOSE(UNIT=1, DISPOSE='DELETE')
      OPEN(UNIT=1, NAME='HEADREC.SIT', TYPE='OLD',
      FORM='UNFORMATTED', READONLY, ERR=25)
      GO TO 30
      CONTINUE
      PRINT 4001
4001  FORMAT(' ERROR ON HEADREC.SIT OPEN OR READ'/1H1)
      STOP 'ERR ON HEADREC.SIT'
      CONTINUE
      READ(1,ERR=25,ERR=25) ((SITE(J,K), J=1, SIZE), K=1, MAXDSK)
      CLOSE(UNIT=1, DISPOSE='SAVE')
      CALL FSTART(SERVAR, DRIVE)
      TYPE=0
      CALL BEGNPT(TAPN)
      TIME=0
      CALL JPRIN(ERR, TIME)
      IF (TIME, EQ, 1) GO TO 200
      DO 50 K=1, MAXDSK
      DO 50 J=1, SIZE
      IF (SITE(J,K), EQ, SERVAR) GO TO 60
      CONTINUE
      CALL TFILE(1,1)
      GO TO 40
      PRINT 400
      CALL FOLD(FOLD)
      GO TO 40
      CALL TEND(1)
      PRINT 400
      FORMAT(1H1)
      STOP
      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

### 3.4.2 IMUNLD SUBROUTINES

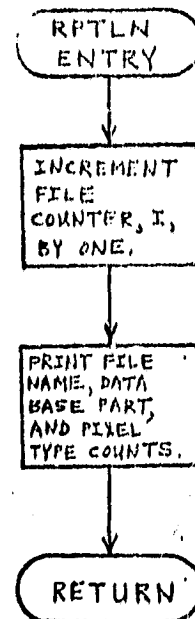
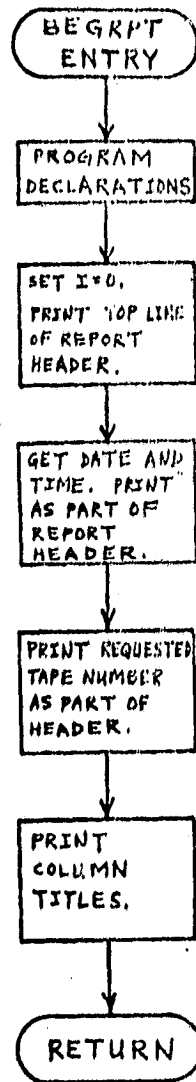
#### 3.4.2.1 General

The special subroutines called by IMUNLD are BEGRPT, FBLD, and FSTART. FSTART calls special subroutines ETOA and SCREEN. BEGRPT has an additional entry point RPTLN, referenced by FBLD. FSTART has additional entry points HDRIN, referenced by IMUNLD, and DATOUT, referenced by FBLD. SCREEN has an additional entry point SNAG, referenced by FSTART.

#### 3.4.2.2 Subroutine BEGRPT

This subroutine prints the header and each line of the output report.

### 3.4.2.2a Flowchart





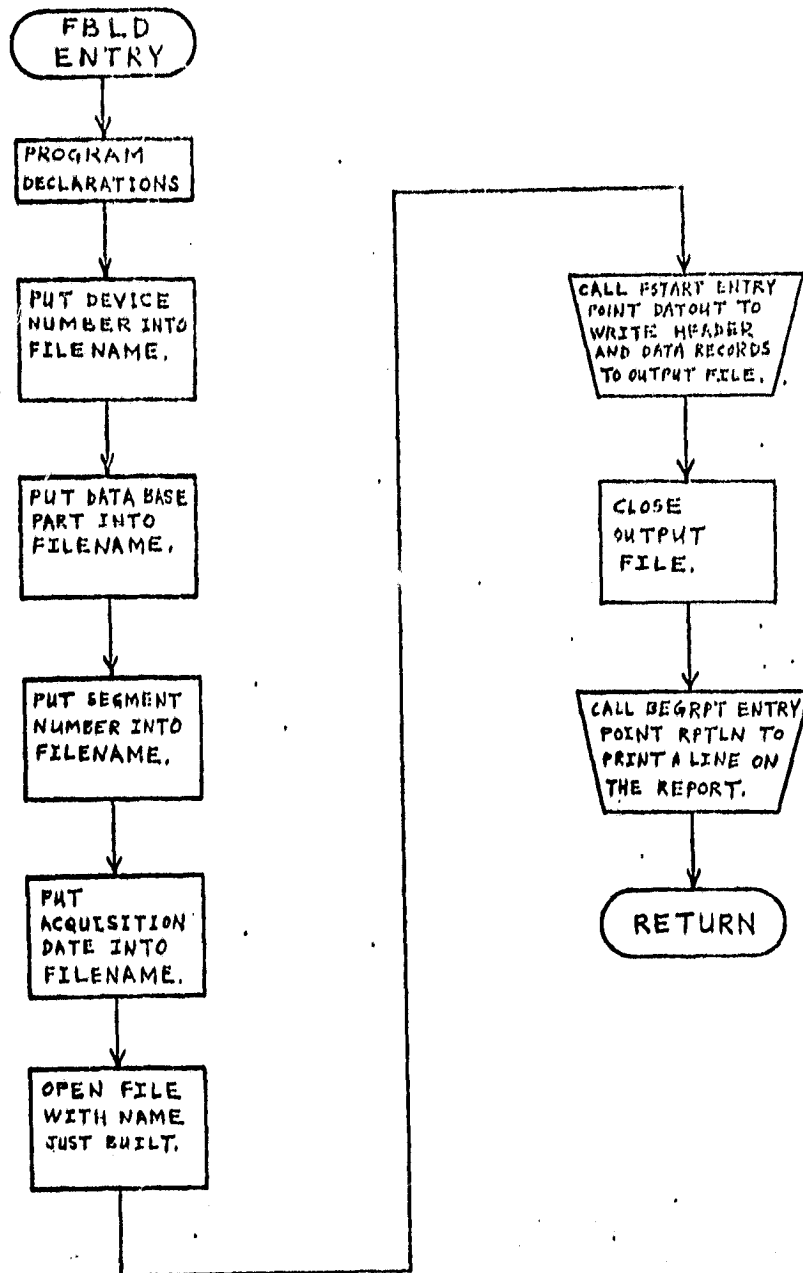
3.4.2.2b Listing

```
SUBROUTINE BEGRPT(TAPEM).  
  IMPLICIT INTEGER(A-Z)  
  DIMENSION TAPEM(3),PTC(5)  
  LOGICAL*1 IDATE(9),ITIME(6),FILNAM(28),DBNUM  
C  
C  
  I=0  
  PRINT 20  
20  FORMAT(1H1,34X,'ACCURACY ASSESSMENT DATA TRANSACTION REPORT')  
  CALL DATE(IDATE)  
  CALL TIME(ITIME)  
  PRINT 30, IDATE, ITIME  
30  FORMAT(105X, 'DATE ',9A1,/,105X, 'TIME ',6A1)  
  PRINT 40, TAPEM  
40  FORMAT(47X, 'IMAGE DATA BASE UPDATE',/,47X,  
  * 'MAG TAPE NUMBER ',3A2)  
  PRINT 42  
42  *  FORMAT(1H0,13X, 'FILE NAME',7X, 'DATA      GOOD      GARBLED',  
  *      6X, 'CLOUD      SHADOW      WATER')  
  PRINT 50  
50  FORMAT(30X, 'BASE',5(5X, 'PIXELS')/)  
  RETURN  
  ENTRY RPTLN(FILNAM,DBNUM,PTC)  
  I=I+1  
  PRINT 60, I, (FILNAM(J), J=12,24), DBNUM, PTC  
60  FORMAT(5X,13.2H,/,2X,13A1, /X,11,5I11)  
  RETURN  
  END
```

### 3.4.2.3 Subroutine FBLD

This subroutine creates the proper file name, opens that file, transfers the data to it, closes it, and reports on it.

### 3.4.2.3a Flowchart



### 3.4.2.3b Listing

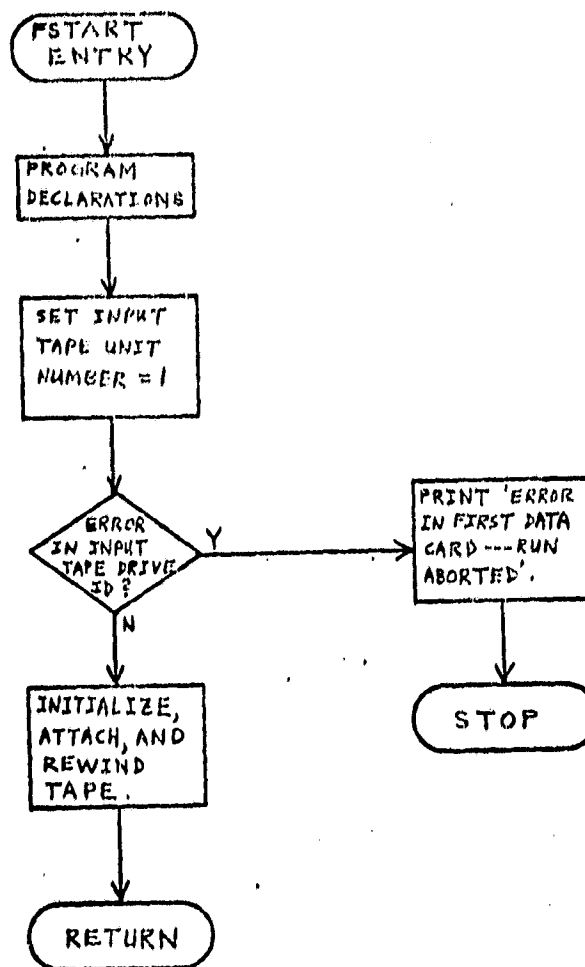
```
SUBROUTINE FOLD(DBNUM)
  IMPLICIT INTEGER(A-Z)
  LOGICAL*1 FILNAM(24), DBNUM, ACCOTE(6), DEVICE
  COMMON SEQNO, ACCOTE, TAPNO(3), DEVICE
  DIMENSION PTC(2), FIL(14)
  EQUIVALENCE (FIL, FILNAM)
  DATA FIL/2H00, 2H21, 2H11, 2H31, 2H, 1, 2HDX, 2HXX, 2HXX, 2HXX,
  2HXX, 2H, 1, 2HNO, 2H11, 0/

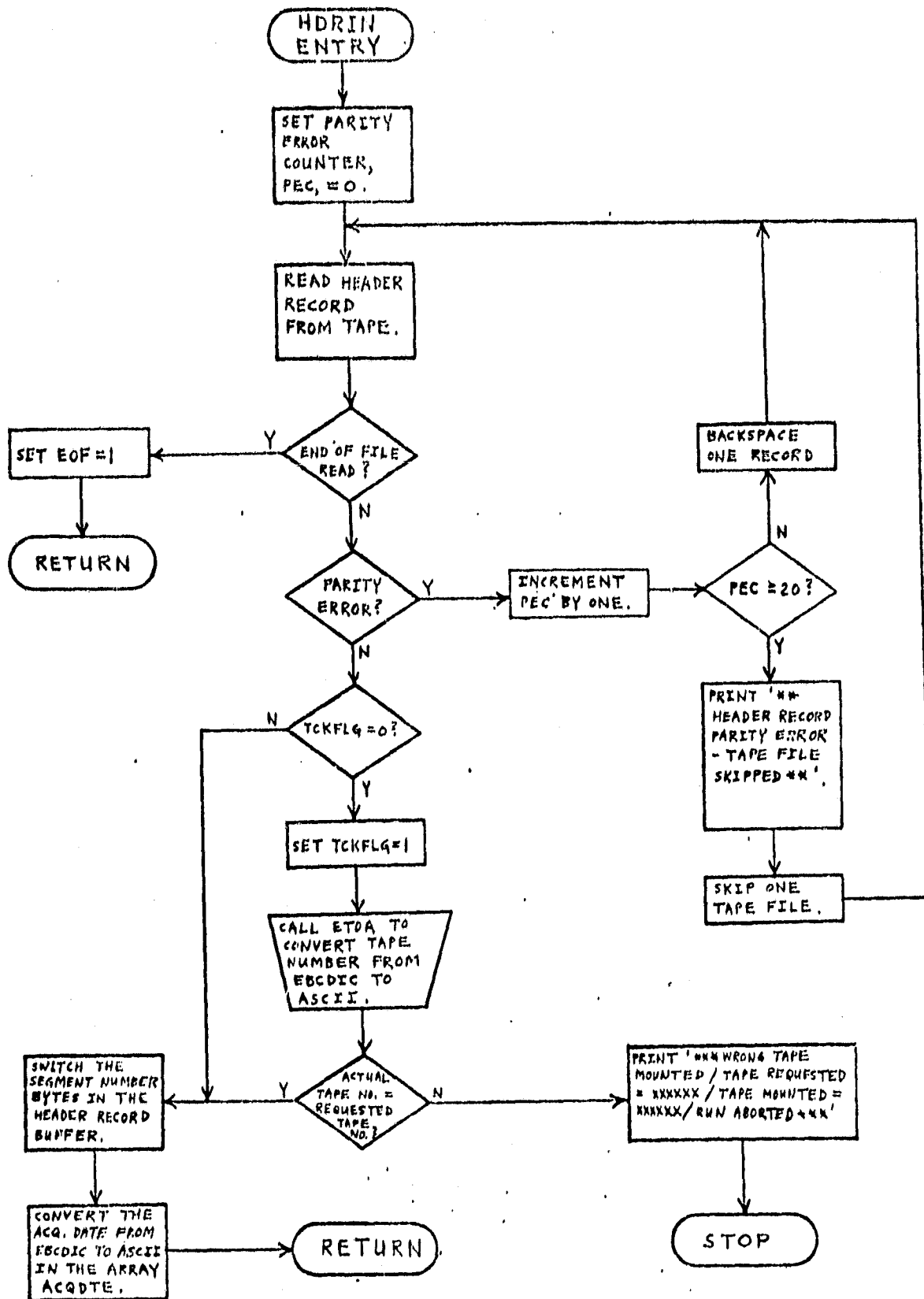
C
C
  FILNAM(3)=DEVICE
  ENCODE(1,50,FILNAM(10)) DBNUM
50  FORMAT(11)
  ENCODE(4,51,FILNAM(12)) SEQNO
51  FORMAT(14)
  DO 60 J=1,5
  FILNAM(J+15)=ACCOTE(J)
60  CONTINUE
  DO 71 J=12,15
  IF(FILNAM(J) .EQ. ' ') FILNAM(J)='0'
71  CONTINUE
  OPEN(UNIT=2, NAME=FILNAM, TYPE='UNKNOWN', FORM='UNFORMATTED')
  CALL DATOUT(PTC)
  CALL CLOSE(2)
  CALL RPTLN(FILNAM, DBNUM, PTC)
  RETURN
  END
```

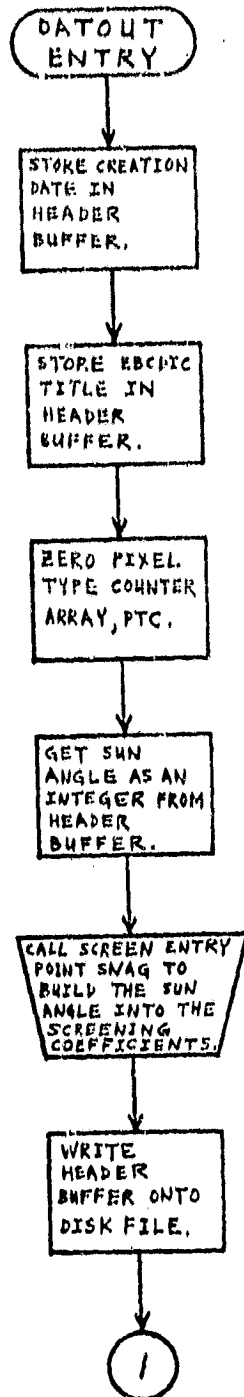
#### 3.4.2.4 Subroutine FSTART

This subroutine sets up the tape drive, reads each file's header record, reads and screens the data from a particular file, and writes the output disk file.

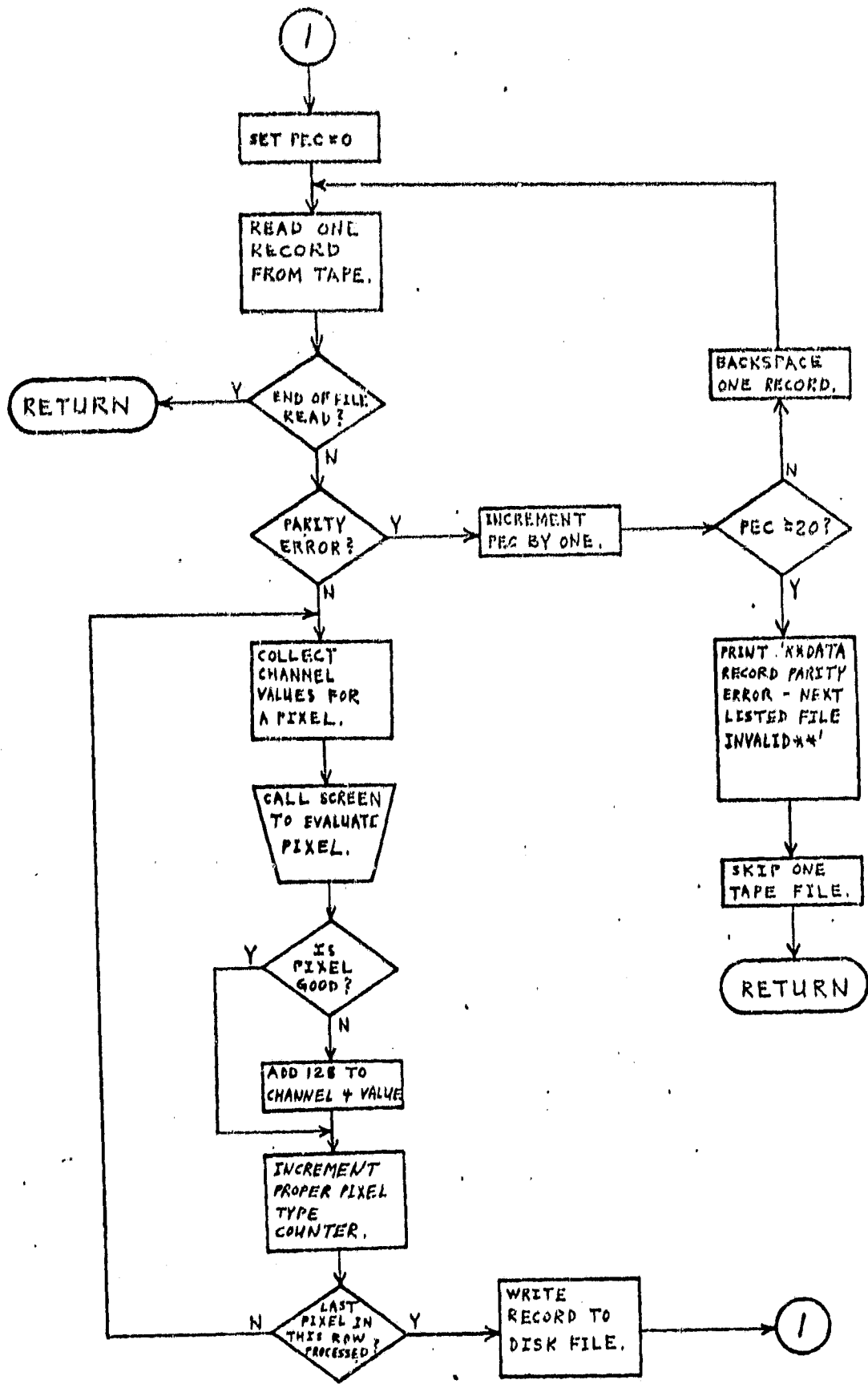
### 3.4.2.4a Flowchart











### 3.4.2.4b Listing

```

SUBROUTINE FSTART(SRVC,DRIVE)
  IMPLICIT INTEGER(A-Z)
  PARAMETER (NPAR=40)
  COMMON/STATE/ S(2)
  LOGICAL MREADY(3266), AGET(4), TAPEN(4), TITL(32), SCRVB(4)
  LOGICAL TACR(2), IT(4), TVICL, ETRAC
  CHARACTER*8 SGT, ACUTE, TAPEN(4), DEVICE
  DIMENSION HEADER(1530), RTN(1530), PIC(4), SCRVA(4)
  LOGICAL M(READY,READY,GET), (TAPEN, TAPEN), (SCRVA, SCRVA),
    (SUMARY, READY)
  DATA RTN/RT1/,X1/XY1/,SCRVA/40/,ETRAC/
  DATA TITL/"366","311","305","323","304","342","3","301,
    "343","100","302","306","311","312","3","325,
    "305","101","100","311","324","311","317","305,
    "100","306","311," 23,"305","110","100","107/
C
C
C      UNIT=1      LOGICAL UNIT
C      MTRKXT=0    ! UNIT, 1=XT
C      DRIVE=1     ! PHYSICAL UNIT
C
C
C      IF(SRVA=MT) 8,5,8
C      MTRKXT=0
C      GO TO 15
C      IF(SRVA=XT) 12,10,12
C      MTRKXT=1
C      GO TO 15
C      PRINT 14
C      FORMAT(140,'ERROR IN FIRST DATA CARL---RUN ABORTED!')
C      STOP
C      IF(DRIVE .LT. 0 .OR. DRIVE .GT. 1)GO TO 12
C
C
C      CALL TINIT(UNIT,MTRKXT,DRIVE)
C      CALL TATCH(UNIT)
C      CALL TRND(1)
C      RETURN
C
C
C      ENTRY HDRIN(EOT,TCKFLG)
C
C      REC=0
C      CALL TREAD(UNIT,HEADER,1530)
C      IF(S(1).EQ."366")GO TO 90
C      IF(S(1).NE."374")GO TO 205
C      REC=REC+1
C      IF(REC.GE.20)GO TO 202
C      CALL TSKIP(UNIT,-1)
C      GO TO 20
C      CONTINUE
C      PRINT 4001
C      FORMAT(100,'**HEADER RECORD PARITY ERROR - TAPE FILE SKIPPED**')
C      CALL TFILE(UNIT,1)
C      GO TO 20
C      CONTINUE
C      IF(TCKFLG.NE.0)GO TO 23
C      TCKFLG=1
C      CALL ETRAC(READY(33),RTN(6))
C      DO 24 I=1,6
C      IF(TAPEN(I).NE.0T(I))GO TO 23

```

```

21      CONTINUE
        GO TO 23
22      CONTINUE
        PRINT 4000,TEMPD,ITM
4000    FORMAT(1000, 'READING TAPE REQUESTED = ',
        * 5A12, ' TAPE COUNTED = ', A3, ' VOL. ABORTED ***', I4)
        STOP 'READING TAPE ABORTED'
23      CONTINUE
        TEMP=HEADBY(67)
        HEADBY(67)=HEADBY(68)
        HEADBY(68)=TEMP
        SER=HEADBY(54)
        DO 30 I=1,5
        ACODE(I)=HEADBY(I+2248)+ETAC  ; ECCIC TO ASCII CONV.
30      CONTINUE
        RETURN
        ENTRY DATOUT(PTC)
        CALL IDATE(MA, DAY, YR)
        HEADBY(61)=DAY
        HEADBY(62)=MA
        HEADBY(63)=YR
        DO 35 I=1,32
        HEADBY(I)=TITL(I)
35      CONTINUE
        DO 36 I=1,5
        PTC(I)=0
36      CONTINUE
        SHANB(1)=HEADBY(2202)
        SHANB(2)=HEADBY(2201)
        CALL SHAB(SHANG) ; PUT SUN. ANGLE INTO SCREENING ROUTINE.
        WRITE(2) HEADER
39      PAPER
40      CALL TNEY(UNIT, RECD, NLEN)
        IF(S(1), EQ. #466) RETURN
        IF(S(1), NE. #374) GO TO 45
        REC=REC+1
        IF(PTC, GE. 20) GO TO 42
        CALL TSKIP(UNIT, #1)
        GO TO 40
42      CONTINUE
        PRINT 4002
4002    FORMAT(' **DATA RECORD PARITY ERROR - NEXT LISTED FILE INVALID**')
        CALL TFILE(UNIT, 1)
        RETURN
45      CONTINUE
        DO 70 I=1,196
        SCRVR(1)=HEADBY(I+72)
        SCRVR(3)=HEADBY(I+268)
        SCRVR(5)=HEADBY(I+464)
        SCRVR(7)=HEADBY(I+660)
        CALL SCREEN(SCRV, LFL)
        IF(LFL, EQ. 1) GO TO 65
        SCRVR(4)=SCRVR(4)+128
        HEADBY(I+660)=SCRVR(7)
        IF(LFL, GT. 10, AND. LFL, LT. 14) GO TO 50
        IF(LFL, GT. 20, AND. LFL, LT. 27) GO TO 57
        IF(LFL, GT. 30, AND. LFL, LT. 35) GO TO 60
        PTC(2)=PTC(2)+1 ; GARDLED
        GO TO 70
50      CONTINUE
        PTC(4)=PTC(4)+1 ; SHADWN
        GO TO 70

```

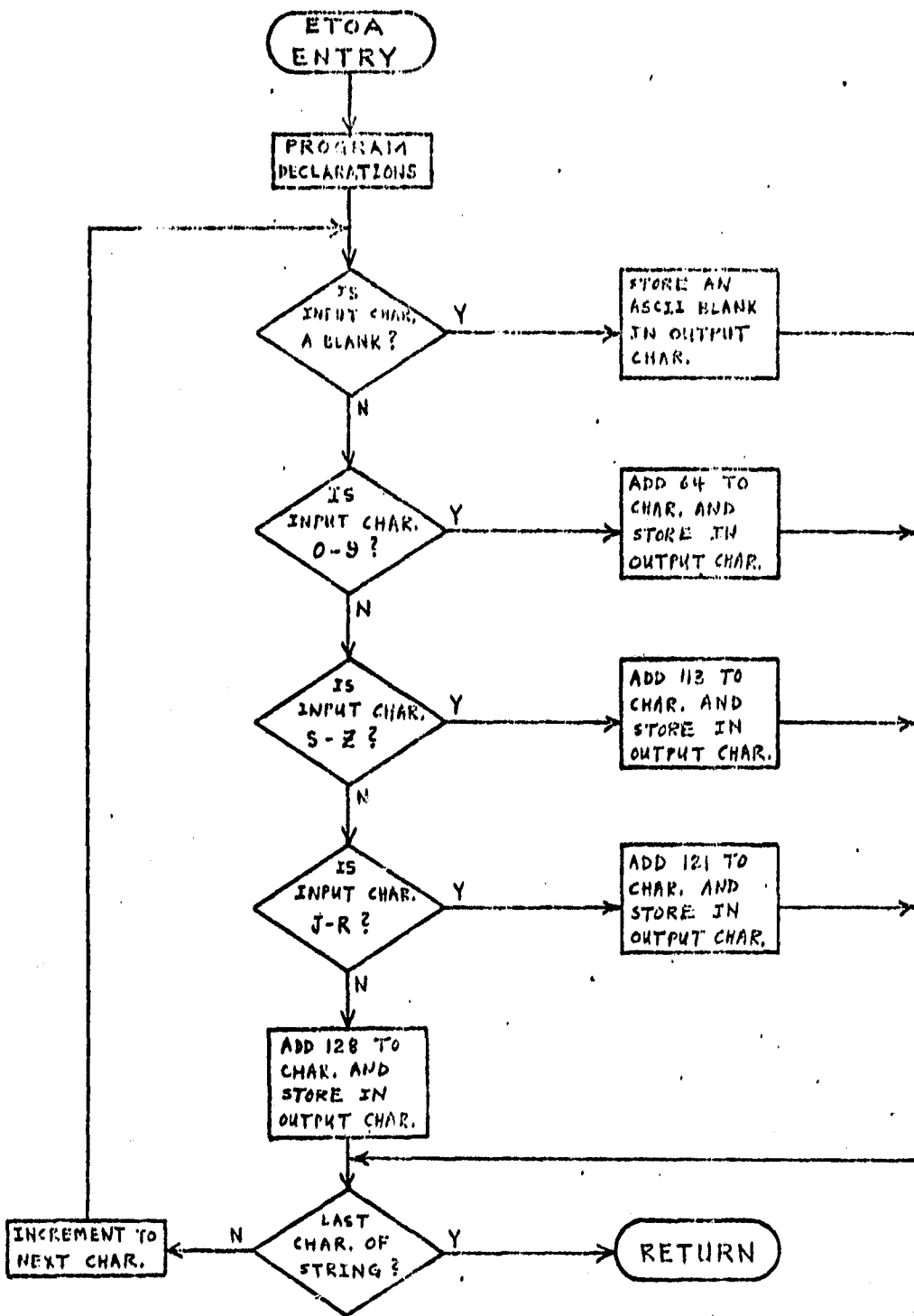
```
57 CONTINUE  
PTC(5)=PTC(5)+1  ; WATER  
GO TO 70  
60 CONTINUE  
PTC(3)=PTC(3)+1  ; CLOUD  
GO TO 70  
65 CONTINUE  
PTC(1)=PTC(1)+1  ; GRND  
70 CONTINUE  
WRITE(2) (RECORD(I),I=1,PLEN)  
GO TO 39  
C  
90 STOP  
RETURN  
END
```

05/10/71 11:18 AM

3.4.2.5 Subroutine ETOA

This subroutine converts an alphanumeric tape number from EBCDIC to ASCII characters.

### 3.4.2.5a Flowchart



### 3.4.2.5b Listing

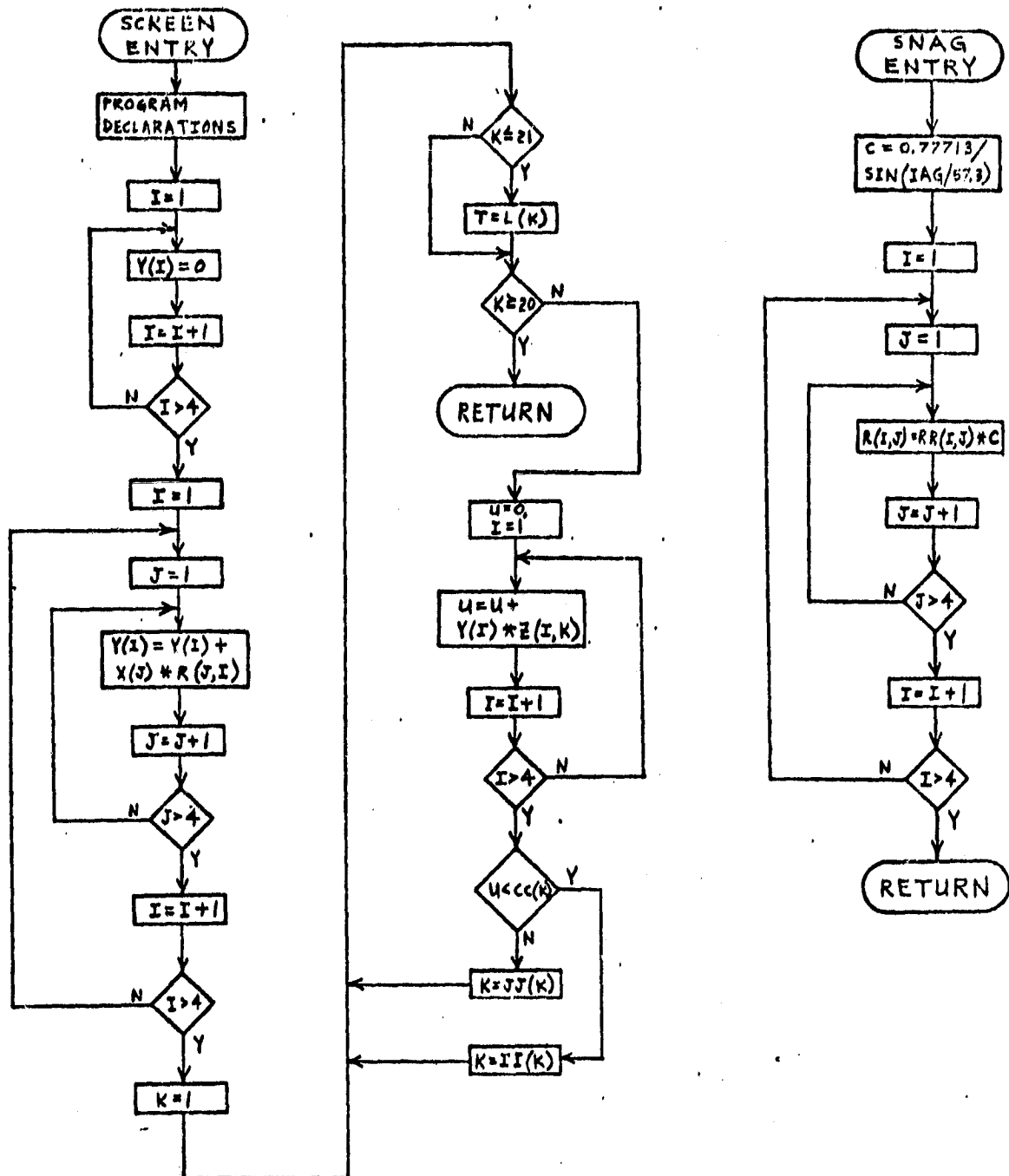
```
CONTINUE (1),A(F,T,0)
C
C      THIS IS A SPECIAL VERSION OF ASCII EQUIVALENT THAT ONLY
C      DEALS WITH DIGITS, AND UPPER CASE LETTERS. IT IS
C      THE SAME AS THE INTERNAL TAPE NUMBER TO ASCII.
C
      LOGICAL F(1),T(1),C1,C2,C3
      DATA C1,C2,C3/04,143,121/
      T(1)=.F.
      IF(F(1),F,0)GO TO 10
      T(1)=.T.
      C(1)=F
10     CONTINUE
      IF(F(1)+16)30,20,20
      C(1)=T
20     CONTINUE
      T(1)=F(1)+C1
      C(1)=T
30     CONTINUE
      IF(F(1)+50)40,40,40
      C(1)=T
40     CONTINUE
      T(1)=F(1)+C2
      C(1)=T
50     CONTINUE
      IF(F(1)+47)70,50,60
      C(1)=T
60     CONTINUE
      T(1)=F(1)+C3
      C(1)=T
70     CONTINUE
      T(1)=F(1)+C1+C1
80     CONTINUE 1 LOOP END
      RETURN
      END
```

#### 3.4.2.6 Subroutine SCREEN

This subroutine, using the sun angle and the four channel values for a pixel, makes a series of tests and decides whether the pixel is good, garbled, cloud, shadow, or water.



### 3.4.2.6a Flowchart



### 3.4.2.6b Listing

```

C      SUBROUTINE SCREEN(X,T) SCREENS DATA X AND RETURNS LABEL T      SCR00010
C
C      COMPILER WITH THE CRISO SWITCH.
C
      SUBROUTINE SCREEN(X,T)
      REAL R(4,4),Z(4,19),RR(4,4),Y(4),CC(19)
      INTEGER II(19),JJ(19),L(21), (4),T
      DATA RR/.5323, .60316, .675 1., 26278,
C          -.24317, -.66006, .57700, .30833,
C          -.89950, .42530, .175, 2, -.04080,
C          -.01594, .13168, -.45157, .87232/
      DATA Z/0,0,0,0,0,0,1,0,
C          0,0,0,0,0,0,-1,0,
C          -.29375,0,0,1,0,0,0,
C          -.1875,0,0,-1,0,0,0,
C          -0,1,-1,0,0,0,0,0,
C          0,55555,1,0,0,0,0,0,
C          -0,83333333,1,0,0,0,0,0,
C          1,0,0,0,0,0,0,0,
C          -1,0,0,-1,0,0,0,
C          1,0,0,0,0,0,0,0,
C          -0,142857,0,0,-1,0,0,0,
C          -1,0,0,0,0,0,0,0,
C          -.0675,-1,0,0,0,0,0,
C          0,0,0,0,0,0,-1,0,
C          0,0,-1,0,0,0,-1,0,
C          -.5,-1,0,-1,0,-.5,
C          -.4,1,0,0,0,0,0,0,
C          -0,4,1,0,-0,6,-0,6,
C          -1,0,0,4,0,0,0,0/
      DATA CC/15, .12, -.4, .14, .20, .156, -.8, .100, .7, 5, 69, .3, 25,
C      -75, .0, 5, -1, 5, 4, 5, -10, -.12, ., -9, ., -37, 75/
      DATA II/2,3,4,5,6,7,8,10,10,12,12,18,18,19,18,18,60,20,20/
      DATA JJ/60,60,60,60,60,60,60,60,60,11,60,13,14,15,16,17,21,
C      19,60/
      DATA L/47,46,45,44,43,42,41,34,33,32,31,26,25,24,23,22,21,
      * 13,12,1,11/
      * 4* GARBLED; 34,33 CLUMP; 32,31 HAZE
      * 2* WATER; 1* SHADOW, 11 AVER WATER; 1 GARD.
C
      DO 5 I=1,4
5      Y(I)=0
      DO 6 I=1,4
      DO 6 J=1,4
6      Y(I)=Y(I)+X(J)*R(J,I)
      K=1
1      CONTINUE
      IF(K.LE.21)T=L(K)
      IF(K.GE. 20) RETURN
      U=0.
      DO 7 I=1,4
7      (U+Y(I))*2(I,K)
      IF(U.LT. CC(K)) GO TO 3
      K=JJ(K)
      GO TO 1
3      CONTINUE
      K=II(K)
      GO TO 1
      ENTRY SNAG(IAG)
      C=0,77713/SIN(IAG/57,3)
      DO 8 I=1,4
      SCR00020
      SCR00030
      SCR00040
      SCR00050
      SCR00060
      SCR00070
      SCR00080
      SCR00090
      SCR00100
      SCR00110
      SCR00120
      SCR00130
      SCR00140
      SCR00150
      SCR00160
      SCR00170
      SCR00180
      SCR00190
      SCR00200
      SCR00210
      SCR00220
      00230
      SCR00240
      SCR00250
      SCR00270
      SCR00280
      SCR00290
      00300
      SCR00310
      SCR00350
      SCR00360
      SCR00370
      SCR00380
      SCR00390
      SCR00400
      SCR00420
      SCR00430
      SCR00440
      SCR00450
      SCR00460
      SCR00470
      SCR00480
      SCR00490
      SCR00500
      SCR00510
      SCR00530
      SCR00540
      SCR00550
      SCR00560

```

ORIGINAL PAGE IS  
OF POOR QUALITY

8 DO N J=1,4  
R(I,J)RRR(I,J)\*C  
RETURN  
END

SCR00570  
SCR00580  
SCR00590  
SCR006

### 3.4.3 FIELDSTATS MAIN PROGRAM (FLDSTT)

#### 3.4.3.1 Linkage

This program calls the special subroutines GAMGEN, SCREEN, GETCCR, FASRPT, FGPRPT, and FCCSRP in addition to system subroutines.

#### 3.4.3.2 Interface

GAMGEN, SCREEN, GETCCR, FASRPT, FGPRPT, FCCSRP.

#### 3.4.3.3 Input

A screened image disk file, a ground truth disk file, a crop code translations disk file, and a program parameters disk file.

#### 3.4.3.4 Output

A dot pixel statistics disk file, a pure pixel statistics disk file, a mixed pixel statistics disk file, a subpixel statistics disk file, an acquisition summary report, an optional dot pixel statistics report, and an optional crop code summary report.

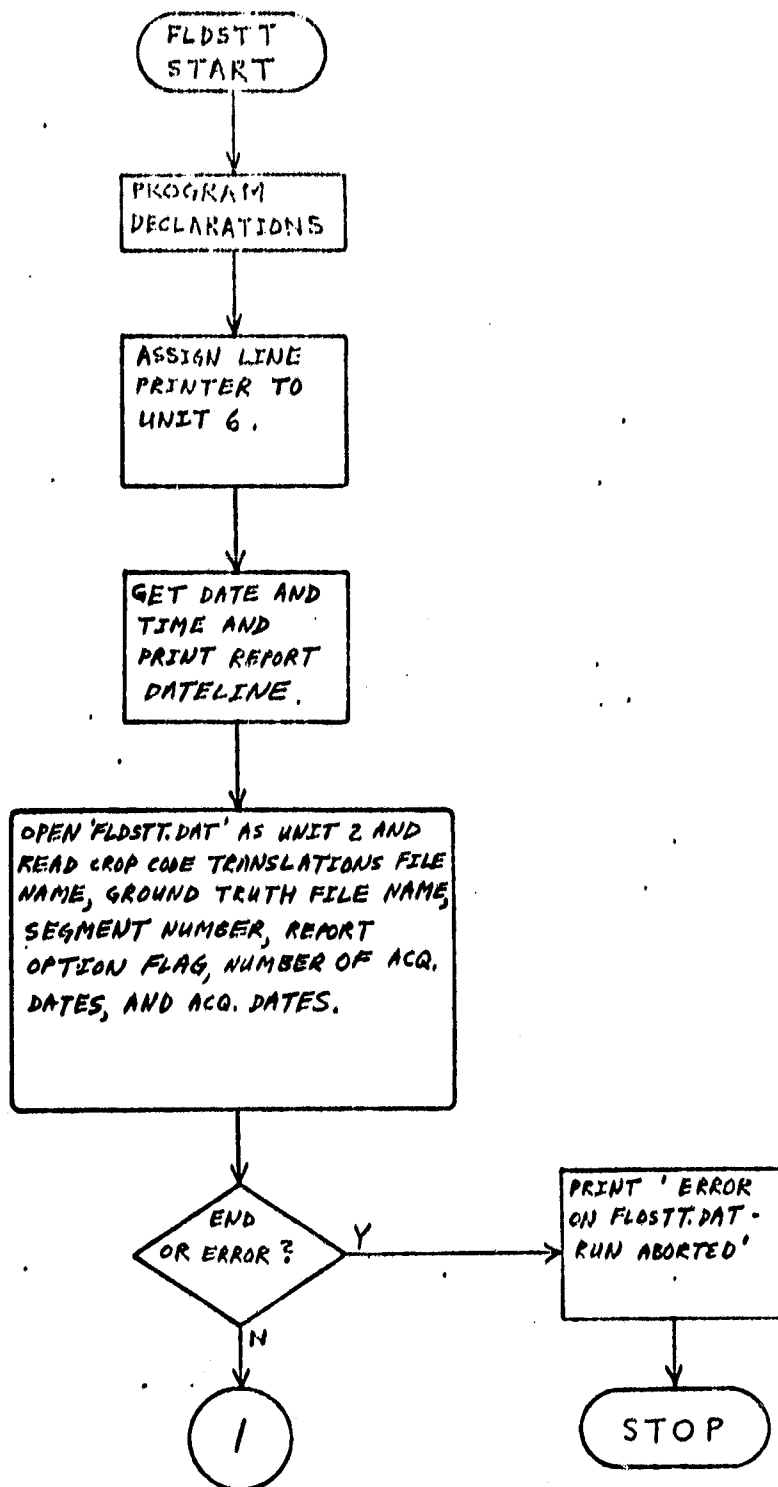
#### 3.4.3.5 Storage

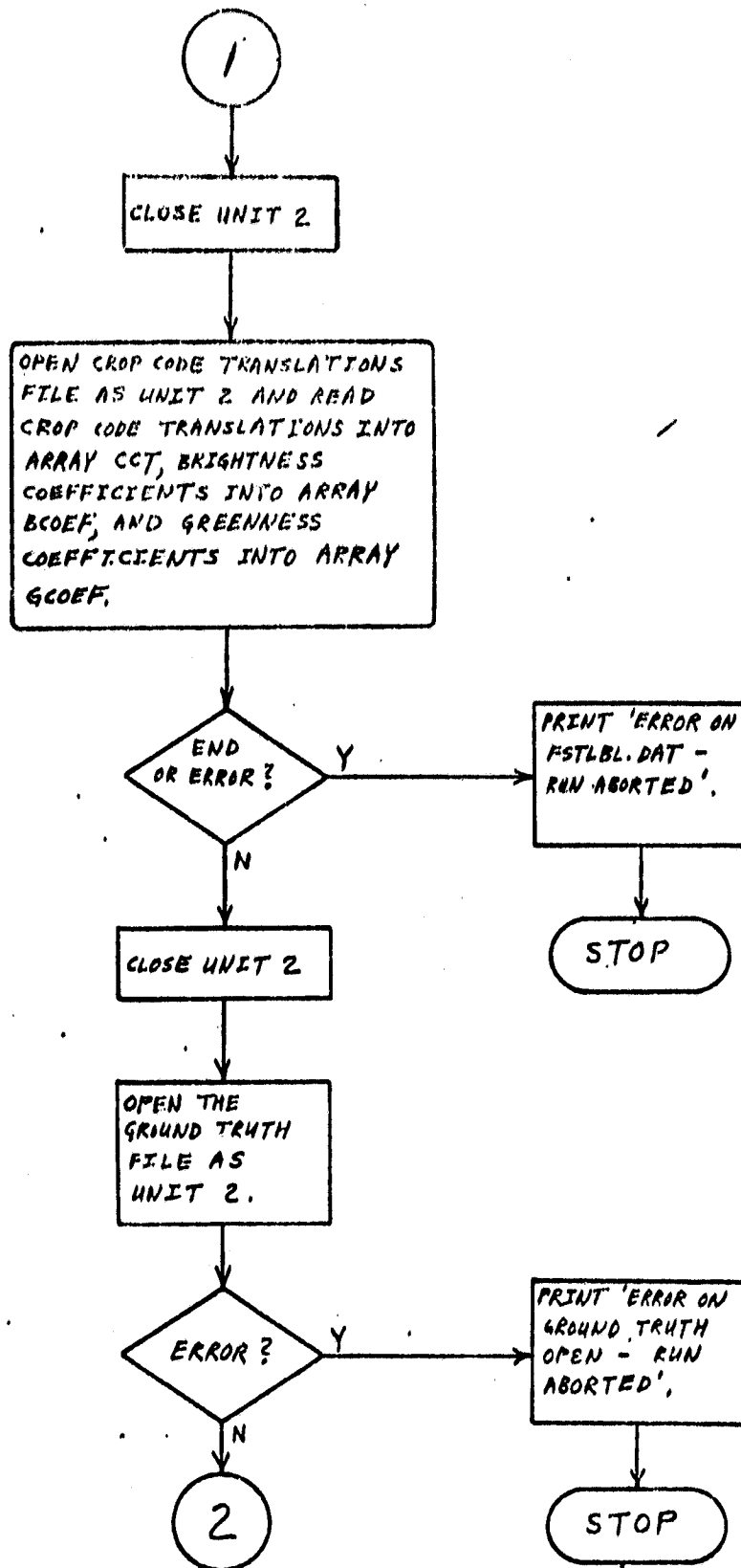
25,360 words.

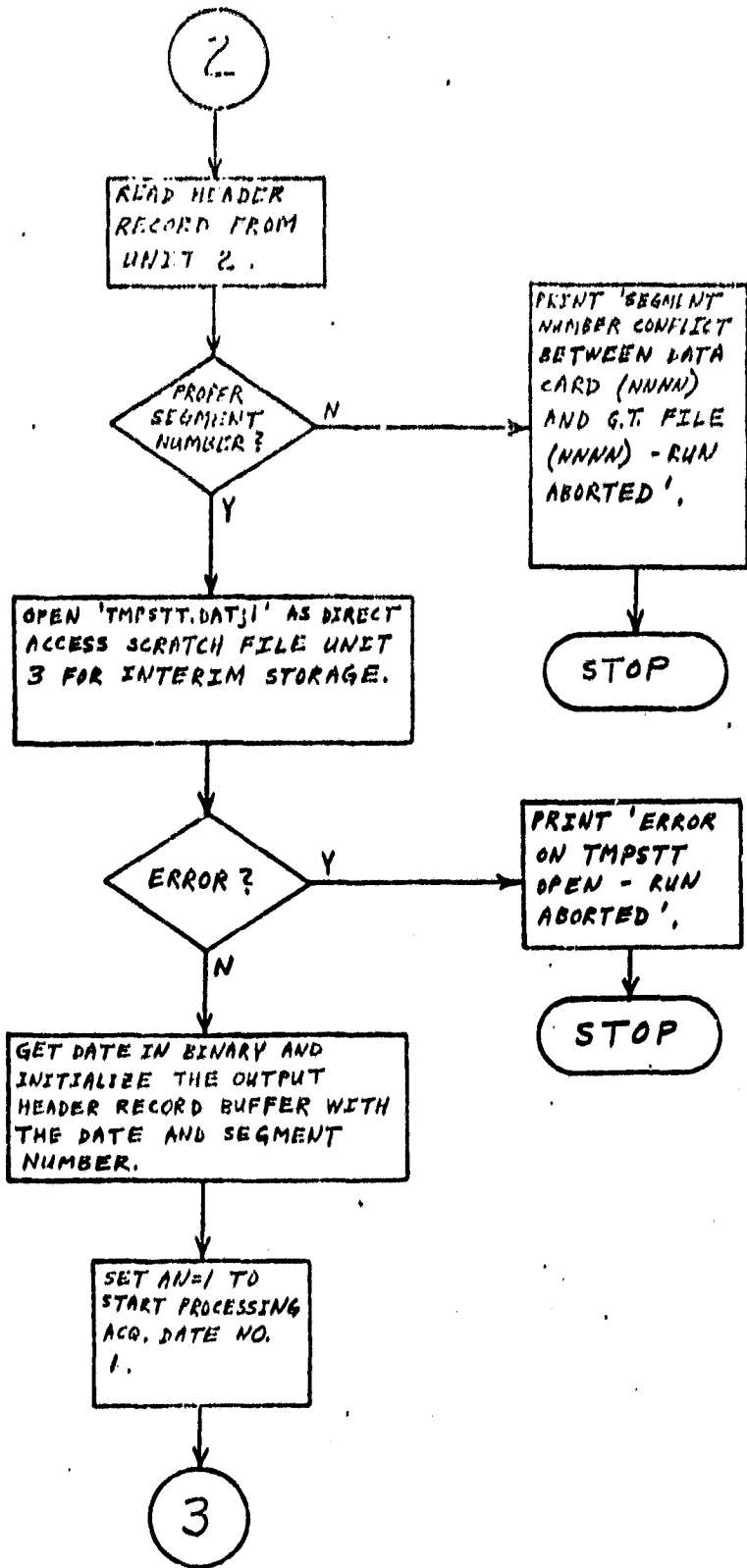
#### 3.4.3.6 Description

The FLDSTT program matches image unload data and ground truth data, calculates means, standard deviations, cross-channel correlations, and brightness and greenness histograms, stores these statistics on disk files, and produces reports on them.

### 3.4.3.7 Flowchart







3

CONVERT THE ORIGINAL TRAIL FILE NAME INTO THE SCREENED IMAGE FILE NAME BY OVERLAYING THE ACQUISITION DATE AND CHANGING THE TYPE TO IMP.

OPEN THE SCREENED IMAGE FILE AS UNIT 4.

ERROR?

PRINT 'ERROR ON SCREENED IMAGE FILE OPEN.'  
FILE = XXXXXXXX  
XXXXXXXXXX XXXXXXXX

17

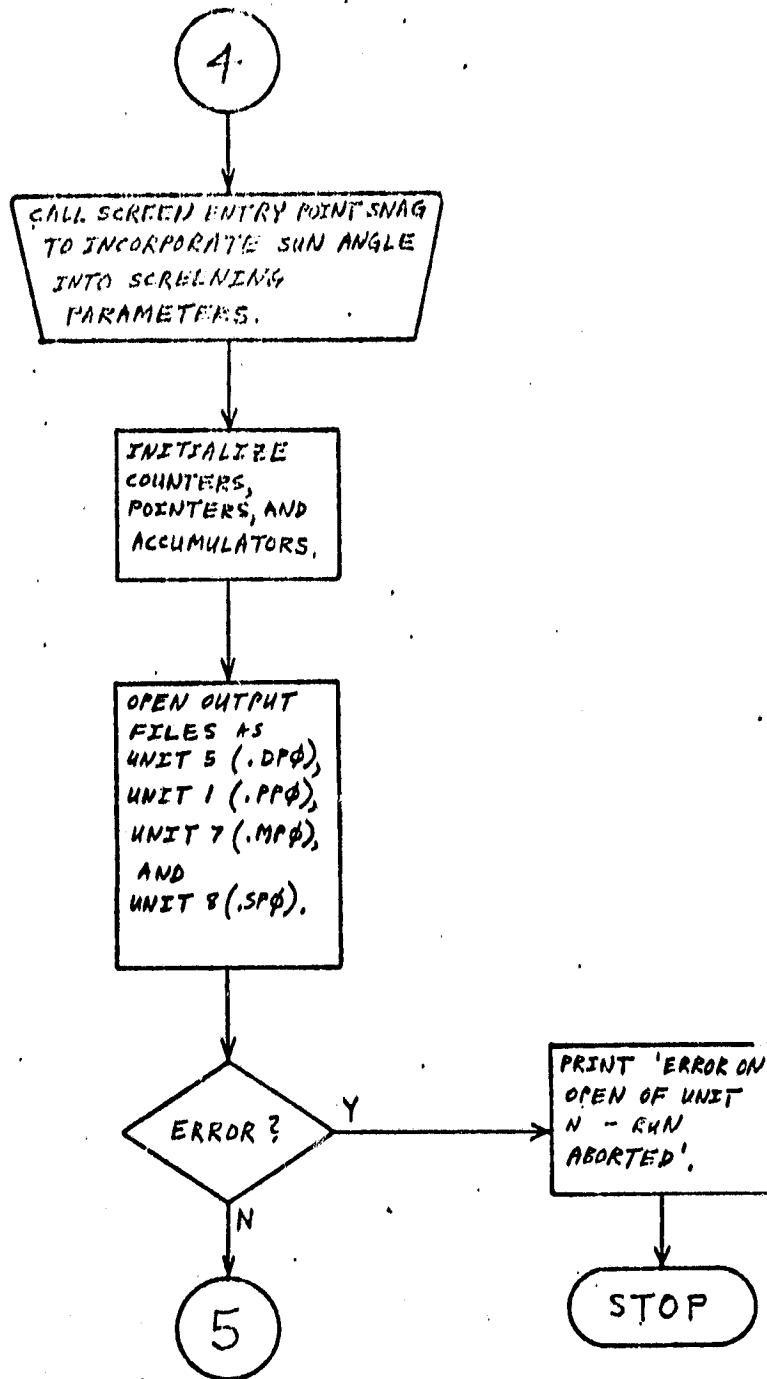
READ THE HEADER OF UNIT 4.

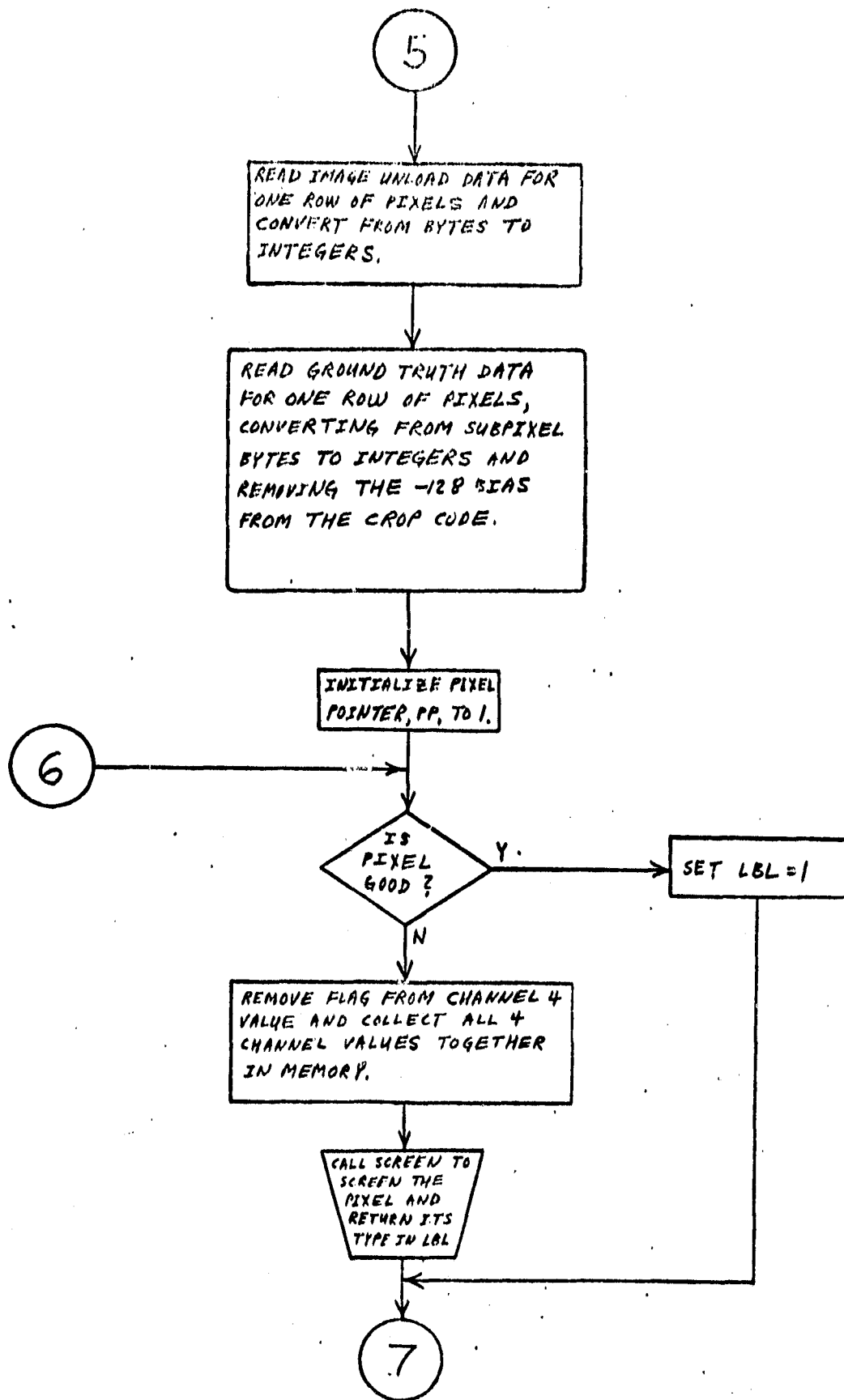
EXTRACT THE SUN ANGLE FROM THE HEADER.

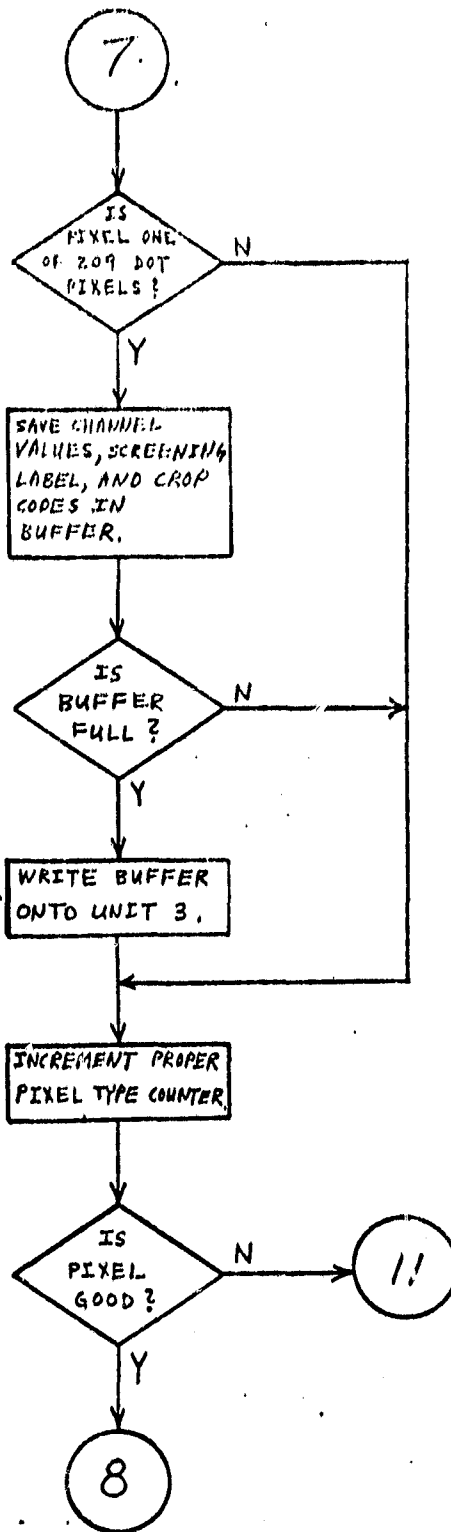
4

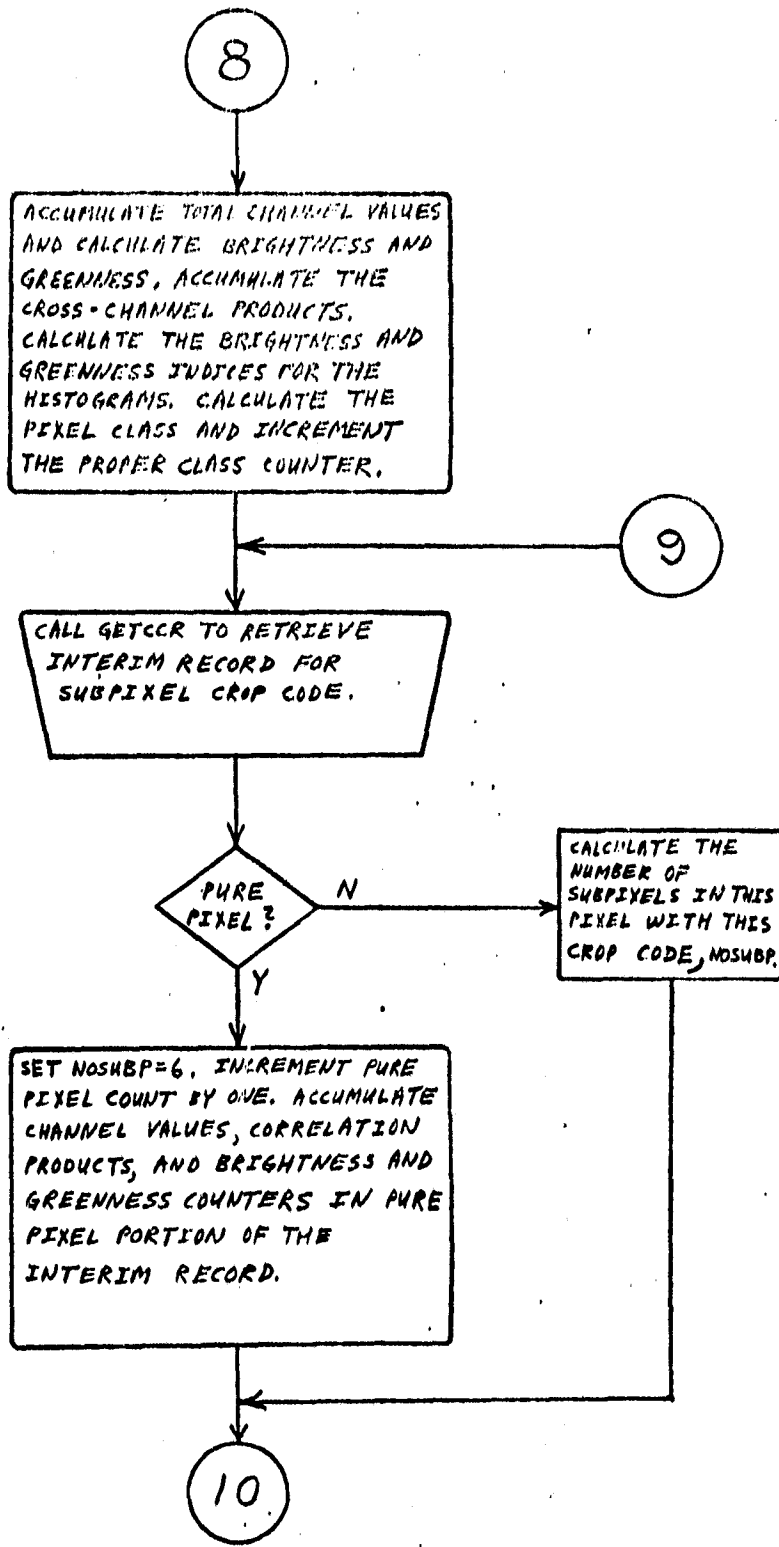
ORIGINAL COPY IS OF POOR QUALITY











10

INCREMENT MIXED PIXEL COUNTER BY 1. ACCUMULATE CHANNEL VALUES, CORRELATION PRODUCTS, AND BRIGHTNESS AND GREENNESS COUNTERS IN MIXED PIXEL PORTION OF INTERIM RECORD.

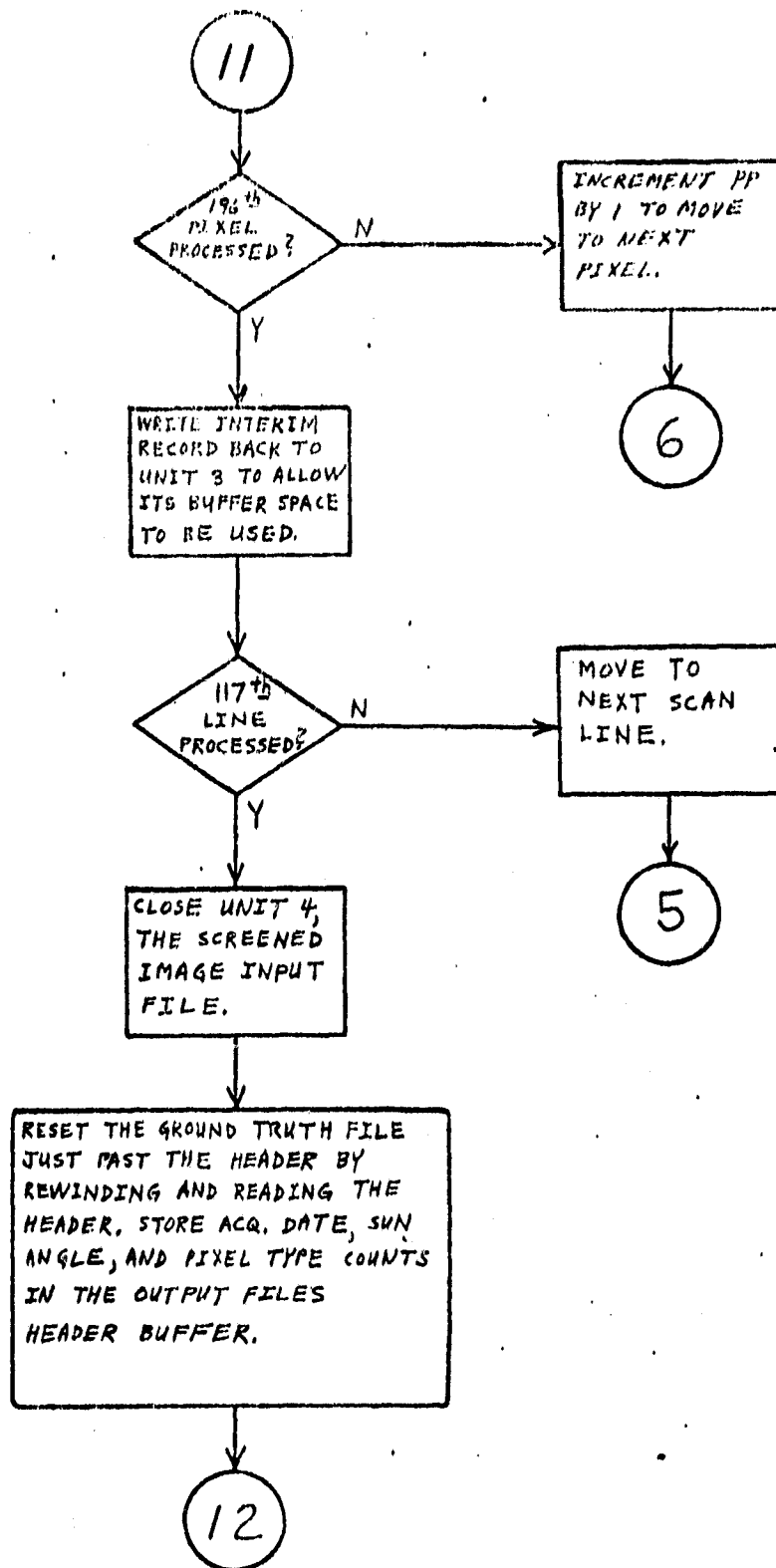
INCREMENT SUBPIXEL COUNT BY NOSUBP. ACCUMULATE NOSUBP \* (CHANNEL VALUES, CORRELATION PRODUCTS, AND BRIGHTNESS AND GREENNESS COUNTERS) IN SUBPIXEL PORTION OF INTERIM RECORD.

ALL CROP CODES IN THIS PIXEL PROCESSED?

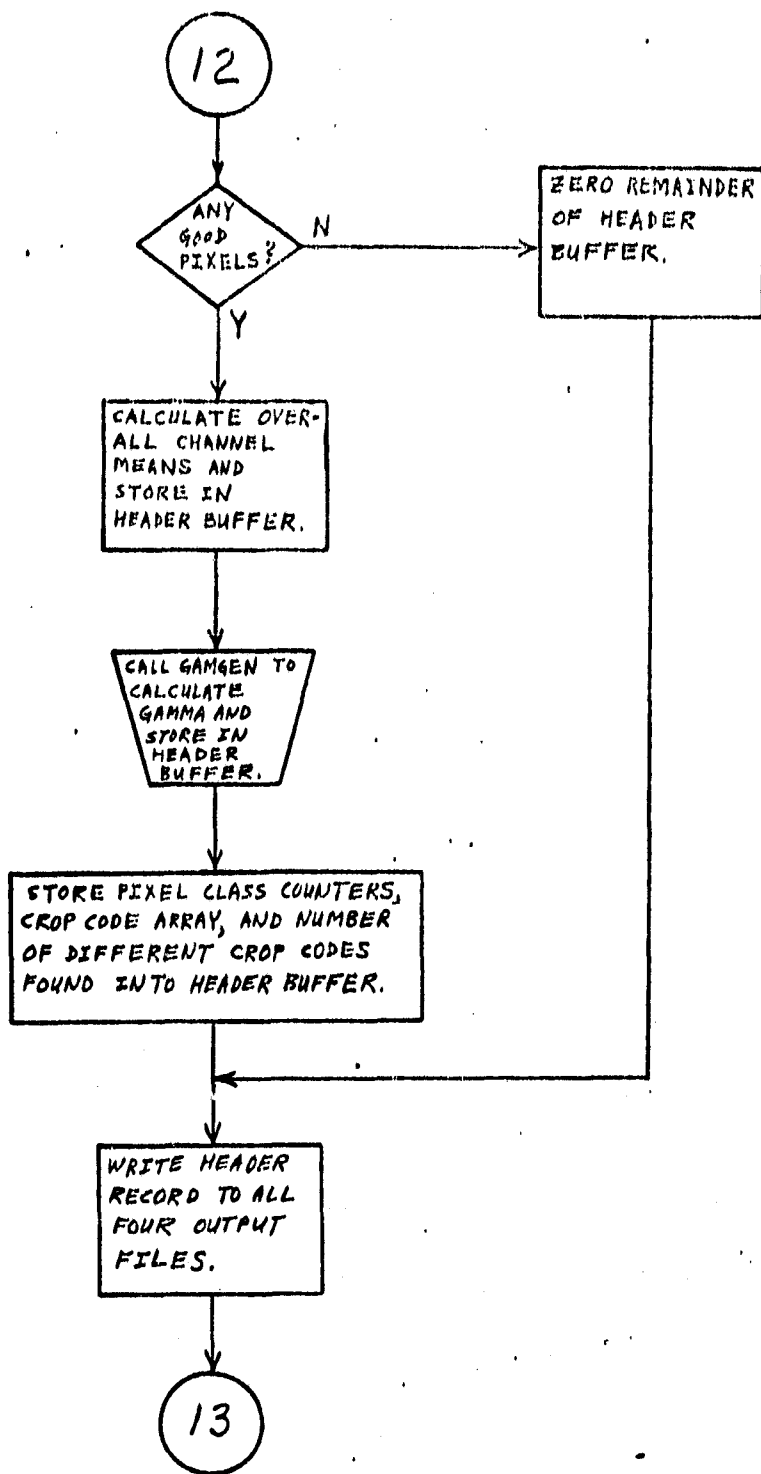
MOVE TO NEXT DIFFERENT CROP CODE IN THIS PIXEL.

11

9



ORIGINAL PAGE IS  
OF POOR QUALITY



13

TRANSFER THE II DOT PIXEL STATISTICS RECORDS FROM THE INTERIM FILE TO THE FINAL DOT PIXEL STATISTICS FILE ON UNIT 5. REWIND UNIT 5 AND READ THE HEADER RECORD TO BE PREPARED IF THE DOT PIXEL STATISTICS REPORT IS DESIRED.

CALL FASRPT TO PRINT THE ACQ. SUM. REPORT.

DOES PRINT OPTION FLAG HAVE BIT NO. 1 TURNED ON?

CALL FGPRPT TO PRINT THE DOT PIXEL STATISTICS REPORT.

CLOSE UNIT 5, THE DOT PIXEL STATISTICS FILE.

WERE THERE ANY GOOD PIXELS?

14

16



13

TRANSFER THE II LOT PLAL. STATISTICS RECORDS FROM THE INTERIM FILE TO THE FINAL DOT PIXEL STATISTICS FILE ON UNIT 5. REWIND UNIT 5 AND READ THE HEADER RECORD TO BE PREPARED IF THE DOT PIXEL STATISTICS REPORT IS DESIRED.

CALL FASRPT TO PRINT THE ACQ. SUM. REPORT.

DOES PRINT OPTION FLAG HAVE BIT NO. 1 TURNED ON?

CALL FGPRPT TO PRINT THE DOT PIXEL STATISTICS REPORT.

CLOSE UNIT 5, THE DOT PIXEL STATISTICS FILE.

WERE THERE ANY GOOD PIXELS?

14

16

14

RETRIEVE AN INITIAL STATISTICS RECORD FOR AN OCCURRING CROP CODE. STORE CROP CODE IN OUTPUT BUFFER. SET J=0. SET POINTER TO PURE PIXEL PORTION OF OUTPUT BUFFER.

GET PIXEL COUNT AS A REAL TYPE NUMBER.

PIXEL COUNT = 0.0?

ZERO PER-CHANNEL MEANS, STANDARD DEVIATIONS, CROSS-CHANNEL CORRELATIONS, AND HISTOGRAMS IN OUTPUT BUFFER.

CALCULATE PER-CHANNEL MEANS, STANDARD DEVIATIONS, AND CROSS-CHANNEL CORRELATIONS, AND STORE THEM IN THE OUTPUT BUFFER. TRANSFER THE BRIGHTNESS AND GREENNESS HISTOGRAMS TO THE OUTPUT BUFFER.

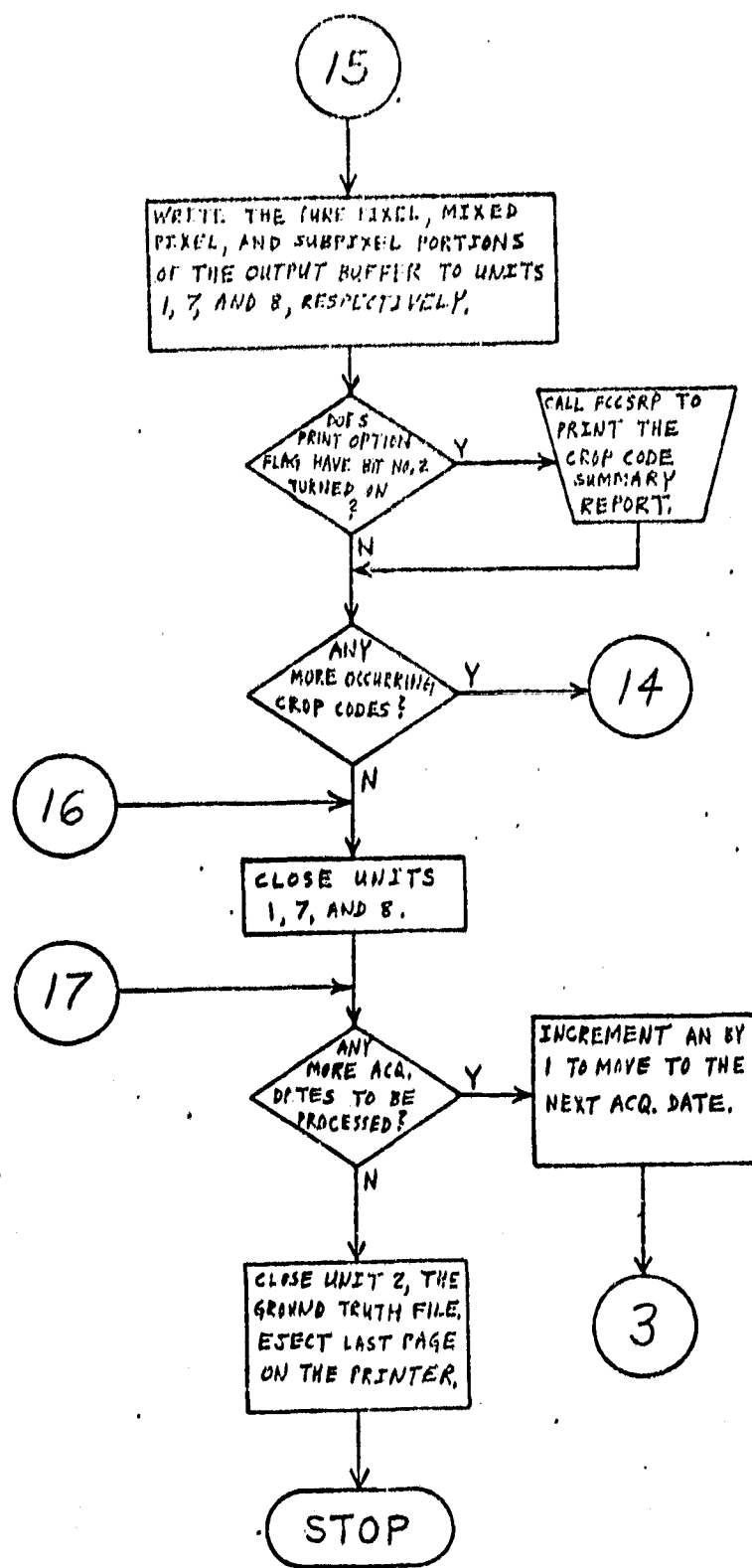
SET POINTER TO MIXED PIXEL PORTION OF OUTPUT BUFFER. SET J=148.

J=0?

SET POINTER TO SUBPIXEL PORTION OF OUTPUT BUFFER. SET J=296.

J=148?

15



UNITED STATES GOVERNMENT  
OFFICE OF PHOTOGRAPHY

3.4.3.8 Listing

DIMENSIONAL DECLARATIONS

THIS PROGRAM CALCULATES THE PERCENTAGE OF... THE PROGRAM... CALCULATES... A WEIGHT... A PART... A FILE... A SUPER... A FILE... STATISTICS... PURE, MIXED, AND SPLIT... FILE TYPES... PURE, MIXED, AND SPLIT.

PROGRAM DECLARATIONS.

```

IMPLICIT INTEGER*(A-Z)
DIMENSION BUF(1530), GTFLM(10), PRC(6), CHV(194,4), SPC(194,4),
*   DTENT(270), CSF(6), TSREC(216), PRPCNT(151), PRCNT(116),
*   MPRCT(151), MPRCT(116), FSR(1140), I1(6), I2(6), I3(6),
*   SCBY(4), TBUF(270), TBUF(450), UNITM(4), LADR(360),
*   PPST(294), MPST(296), SPST(504), ZERAPS(333), SUZ(65)
DIMENSION FLPR(14)
BYTE LADR(360), CHV(194,4), TBUF(450), TITL(32),
*   TSREC(232), IUSEG(2), INTJ(2), SNAVE(2), PDATE(9),
*   PTIME(8), TBUF(450), FTYL(2,4), GTELM(32)
INTEGER*4 ACC(15), TCHA(4), CPA(10), PRC(6), PPRS(10), MPCS(4),
*   MPRS(10), MPRS(4), PPS(10), PRCNT, MPRCNT(151), MPRCNT(116),
*   TSP(578), DFSREC(225), FHACC
REAL BRIGHT, CRCH, CCFE(4), GUFEE(4), RHDR(4), RESR(578), CNT, GAM, DNH,
*   RT
EQUIVALENCE (HDF, BUF, CHVF, TSREC, TSREC, CCFE), (CHV, CHV,
*   (FSREC, SPC), (PRCS, BUF(2)), (PPRS, BUF(10)),
*   (PPCNT, BUF(30)), (MPRCNT, BUF(31)), (MPRCNT, BUF(142)),
*   (MPCS, BUF(294)), (MPPS, BUF(306)), (MPRCNT, BUF(326)),
*   (MPRCNT, BUF(327)), (MPRCNT, BUF(479)), (MPCS, BUF(594))
EQUIVALENCE (PPRS, BUF(602)), (MPRCNT, BUF(622)), (MPRCNT, BUF(624)),
*   (MPRCNT, BUF(924)), (RHDR, FHDF(14)), (GAM, FHDF(26)),
*   (FSREC(2), FSR), (TSREC(2), TSP), (FSREC(622), DFSREC),
*   (IUSEG, IUSEGR), (INT, INTE), (SNAVE, SNAVGB),
*   (FHACC, FHDF(10)), (GTELM, GTFLM)
EQUIVALENCE (FSREC(2), PPST), (FSREC(296), MPST),
*   (FSREC(594), SPST), (TBUF, TBUF),
*   (ZERAPS, SUZ)
COMMON/CARRY/ACR, CCA(254), CCIC, /RCMN1/ENLR,
*   /RCMN2/FSREC, /RCMN3/CT(256)
DATA CHV/784*0/, DTTRUF/270*0/,
*   I1/6,7,8,10,11,13/,
*   I2/1,1,1,2,2,3/, I3/2,3,4,3,4,4/
DATA FHDR/'FI', 'EL', 'DS', 'TA', 'TS', 355*0/, ZERAPS/333*0/
DATA FTYPE/'D', 'P', 'M', 'M', 'M', 'M', 'S', 'P',
*   UNITM/5,1,7,3/
DATA TITL/'306','311','305','323','304','342','343','301',
*   '343','100','342','303','331','305','315','325',
*   '305','304','100','311','324','301','307','305',
*   '100','305','311','32','305','100','100','100/

```

START OF PROGRAM.

PRINT DATE AND TIME HEADIN.

CALL ASSIGN('LPI',3)

```

CALL UNIT (UNIT) = 1, NAME = 'G. T. FILE', TYPE = 'FORMATTED',
CALL UNIT (UNIT) = 1, NAME = 'G. T. FILE', TYPE = 'FORMATTED',
CALL UNIT (UNIT) = 1, NAME = 'G. T. FILE', TYPE = 'FORMATTED',
4040 FORMAT(15(15X))
C
C      RETRIEVE PROGRAM PARAMETERS FROM THE FILE 'G. T. FILE', DAT.
C
OPEN (UNIT=2, NAME='FLSTL', TYPE='FORMATTED', READONLY, ERR=32)
READ (2, 4025, END=10, ERR=10) (A(15), I(15), F(15), I(15))
4025 FORMAT(15(15X))
GOTO 30
10 CONTINUE
WRITE (4, 4025)
4025 FORMAT(' ERROR ON FLSTL, DAT - RUN ABORTED')
STOP      'ERR ON FLSTL, DAT'
20 CONTINUE
IF (GTFLEN(15).NE.'1') GOTO (13), (15), (17), (19) TO 10
IF (GTFLEN(15).EQ.'1') GOTO 10
IF (GTFLEN(15).EQ.'15') GOTO 15
READ (2, 4001, END=10, ERR=10) (A(15), I(15), F(15))
4001 FORMAT(5(15,5X))
CLOSE (UNIT=2)
C
C      RETRIEVE ORP CORR TRANSLATIONS, BOLD MASS COEFFICIENTS,
C      AND GREENNESS COEFFICIENTS.
C
FLFLN(16)=0
OPEN (UNIT=2, NAME='FLFLN', TYPE='FORMATTED', READONLY, ERR=32)
READ (2, 4028, END=32, ERR=32) (A(16))
4028 FORMAT(16(3X, A2))
READ (2, *, END=32, ERR=32) (A(16))
READ (2, *, END=32, ERR=32) (A(16))
CLOSE (UNIT=2)
GOTO 33
32 CONTINUE
WRITE (4, 4029)
4029 FORMAT(' ERROR ON FLSTL, DAT - RUN ABORTED')
STOP      'ERR ON FLSTL, DAT'
33 CONTINUE
C
C      OPEN THE GROUND TRUTH DISK FILE, READ ITS HEADER (FIRST
C      SEQUENTIAL RECORD), AND CHECK THE SEGMENT NUMBER.
C
GTFLN(16)=0
OPEN (UNIT=2, NAME='GTFLN', TYPE='FORMATTED', READONLY,
*      FORM='UNFORMATTED', ERR=100)
GOTO 110
100 CONTINUE
WRITE (4, 4003)
4003 FORMAT(' ERROR ON GROUND TRUTH OPEN - RUN ABORTED')
STOP      'G. T. OPEN ERROR'
110 CONTINUE
READ (2) HDR
IF (HDR(34).EQ.'SEG') GOTO 120
WRITE (4, 4004) HDR(34)
4004 FORMAT(' SEGMENT NUMBER CONFLICT BETWEEN DATA CARD (1,14,
*      1) AND G. T. FILE (1,14,1) - RUN ABORTED')
STOP      'G. T. SEG ERROR'
120 CONTINUE
C
C      OPEN SCRATCH FILE TPSTL, TYPE 11 FOR INTERMEDIATE STORAGE
C      OF ACCUMULATIONS AND CORR'S.

```

ORIGINAL PAGE IS  
OF POOR QUALITY



```

337 CONTINUE : LOOP END
    DO 338 I=1,3
      MCC(I)=C1*SCALE*CLASS(C(I),1)
340 CONTINUE : LOOP END
    DO 341 I=1,2
      MCC(I)=C1*SCALE*ARRAY(C(I),I,PRINT=5)
343 CONTINUE : LOOP END
    SPCG(I,J)=C1 : SURPIXEL COORDINATES
341 CONTINUE : LOOP END
C
C
C      FROM OUTPUT FILES,
      DO 344 I=1,4
        GTITLE(1:4)=FTYPE(1,I)
        GTITLE(5:8)=FTYPE(2,I)
        MPR(COIT=COIT,IT=I),MACHGET(ON,FORMAT=FORMATTED),END=346)
344 CONTINUE : LOOP END
        GO TO 350
346 CONTINUE
        WRITE(4,4030)UNITN(I)
4030 FORMAT('1=PROR OR OPEN OF UNIT',I3,' = NOW ADPTED')
        STOP : OUTPUT NICK FILE OPEN ERROR
350 CONTINUE
C
C
C      READ DATA FOR ONE LINE OF PIXELS.
      CCIC=1 : SCALE BUFFER CHARACT.
      READ(4)TBUF
      J=73
      DO 400 I=1,1567,2
        CHV(I)=TBUF(J) : TRANSFER CHANNEL VALUE BITS TO INTEGER WORDS
        J=J+1
400 CONTINUE : LOOP END
      DO 420 I=1,5,2
        READ(2)CBUF
        K=1
        DO 410 J=73,463,2
          L=CHV(I) : CONVERT SURPIXEL FROM BYTE TO INTEGER,
          SPCG(K,I)=L+128 : REMOVE -2^7 BIAS FROM VALUE.
          L=CHV(I+1)
          SPCG(K,I+1)=L+128
          K=K+1
410 CONTINUE : LOOP END
420 CONTINUE : LOOP END
      PP=1 : PIXEL POINTER
430 CONTINUE
C
C
C      PROCESS PIXEL PP.
C
C
C      SCREEN THE PIXEL IF NECESSARY.
      IF(CHV(PP,4).GT.127)GO TO 435
      LBL=1
      GO TO 445
435 CONTINUE
      CHV(PP,4)=CHV(PP,4)-128 : REMOVE NON-COLOR PIXEL FLAG
      DO 440 I=1,4

```

```

      SCRVI(I)=CVI(PP,I)  I COLLECT VALUES FOR SCREENING ROUTINE.
447 CONTINUE I LOOP END
      CALL SCRTM(SCRVI,I)  I SCREEN THE PIXEL,
448 CONTINUE

```

C  
C  
C

```

      CHECK FOR DOT PIXEL AND PROCESS IF NECESSARY.

      IF(DLN.EQ.0)GO TO 490
      IF(PP.EQ.1)GO TO 490
      DTSUF(DISP+1)=DID  I STORE DOT ID.
      DISP=DISP+1
      DO 450 I=1,4
      DTSUF(DISP+1)=CVI(PP,I)  I STORE CHANNEL VALUES.
      DTSUF(DISP+1)=SPCC(PP,I)  I STORE SUBPIXEL GRIP CODES.
450 CONTINUE I LOOP END
      DTSUF(DISP+1)=SPCC(PP,5)
      DTSUF(DISP+1)=SPCC(PP,6)
      DTSUF(DISP+12)=LBL  I STORE SCREENING LABEL
      IP=DISP+10
      DISP=DISP+12
      IF(DPW.EQ.200)GO TO 460
      GO TO 490
460 CONTINUE

```

C  
C  
C

```

      WRITE DOT PIXEL TMP. STAT. FILE RECORD.

```

```

      WRITE(3'DBREC)DTSUF
      DBREC=DBREC+1
      DSP=0
      DPL=10
      DLN=DLN+10
490 CONTINUE
      IF(LBL.EQ.1)GO TO 530
      IF(LBL.GT.10.AND.LPL.LT.14)GO TO 500
      IF(LBL.GT.20.AND.LPL.LT.27)GO TO 510
      IF(LBL.GT.30.AND.LPL.LT.35)GO TO 520
      XPC=XPC+1
      GO TO 760
500 CONTINUE
      SPC=SPC+1
      GO TO 760
510 CONTINUE
      WPC=WPC+1
      GO TO 760
520 CONTINUE
      CPC=CPC+1
      GO TO 760
530 CONTINUE

```

C  
C  
C

```

      PROCESS GOOD PIXEL.

```

```

      GPC=GPC+1

```

C  
C  
C

```

      CALCULATE BRIGHTNESS AND GREENNESS.

```

```

      BRIGHT=0.0
      GREEN=0.0
      DO 540 I=1,4
      TCHA(I)=TCHA(I)+CVI(PP,I)  I ACCUMULATE TOTAL CHANNEL VALUES.
      BRIGHT=BRIGHT+ACDEF(I)*CVI(PP,I)
      GREEN=GREEN+GDEF(I)*CVI(PP,I)
540 CONTINUE I LOOP END

```



C  
C  
C

ACCUMULATE THE GRASS PRODUCTS.

```
J=1
K=1
L=1
M=4
DO 550 I=1,17
  CCA(I)=CCA(P, J) + CCA(I, P, K)
  IF (I.EQ.1) GO TO 550
  K=K+1
  GO TO 560
550 CONTINUE
  L=L+1
  M=M+1
  J=J+1
  K=J
560 CONTINUE I LOOP END
```

C  
C  
C

CALCULATE BRIGHTNESS AND GREENNESS INDICES, RNDX AND GNDX.

```
IF (BRIGHT, LT. 0.5) GO TO 570
IF (BRIGHT, GT. 149.5) GO TO 580
RNDX=BRIGHT+1.5
GO TO 590
570 CONTINUE
RNDX=1
GO TO 590
580 CONTINUE
RNDX=151
590 CONTINUE
IF (GREEN, LT. -29.5) GO TO 600
IF (GREEN, GT. 84.5) GO TO 610
GNDX=GREEN+31.5
GO TO 620
600 CONTINUE
GNDX=1
GO TO 620
610 CONTINUE
GNDX=11.6
620 CONTINUE
```

C  
C  
C

CALCULATE PIXEL CLASS.

```
CLASS=1
J=2
CCSP(1)=SPCC(P, 1)
630 CONTINUE
DO 640 I=1, CLASS
  IF (SPCC(P, J), EQ. CCSP(I)) GO TO 650
640 CONTINUE I LOOP END
  CLASS=CLASS+1
  CCSP(CLASS)=SPCC(P, J)
650 CONTINUE
  J=J+1
  IF (J, LE. 6) GO TO 630
  PCC(CLASS)=PCC(CLASS)+1 I INCREMENT CLASS COUNTER
  CCR=1 I GRASS CODE NUMBER (WITHIN PIXEL)
660 CONTINUE
  CALL GETCCR(3, CCSP(CCR), CCF) I GET GRASS CODE RECORD
  IF (CLASS, EQ. 1) GO TO 700
```

ORIGINAL PAGE IS  
OF POOR QUALITY

C  
C

MIXED PIXEL CALCULATIONS.

```

N0SUBP=6 ; NO. SUBPIXELS WITH CRP CODE.
DO 670 I=1,6
  IF(SPOC(PP,I).EQ.COSP(CCN))N0SUBP=N0SUBP+1
670 CONTINUE ; LOOP END
675 CONTINUE
  PRCNT=PRCNT+1 ; INCREMENT MIXED PIXEL COUNT
  BRPCNT(BNDX)=BRPCNT(BNDX)+1 ; BRIGHTNESS HISTOGRAM
  GRPCNT(GNDX)=GRPCNT(GNDX)+1 ; GREENNESS HISTOGRAM
  DO 680 I=1,4
  PPCS(I)=PPCS(I)+CHV(PP,I) ; PER CHANNEL SUMS
680 CONTINUE ; LOOP END
  DO 690 I=1,10
  PPRS(I)=PPRS(I)+CPA(I) ; CROSS PRODUCT SUMS
690 CONTINUE ; LOOP END
GO TO 730
700 CONTINUE

```

C  
C  
C

PURE PIXEL CALCULATIONS.

```

N0SUBP=6 ; NO. OF SUBPIXELS WITH THIS CRP CODE.
PRCNT=PRCNT+1 ; INCREMENT PURE PIXEL COUNT.
BRPCNT(BNDX)=BRPCNT(BNDX)+1 ; BRIGHTNESS HISTOGRAM.
GRPCNT(GNDX)=GRPCNT(GNDX)+1 ; GREENNESS HISTOGRAM.
DO 710 I=1,4
  PPCS(I)=PPCS(I)+CHV(PP,I) ; PER CHANNEL SUMS
710 CONTINUE ; LOOP END
  DO 720 I=1,10
  PPRS(I)=PPRS(I)+CPA(I) ; CROSS PRODUCT SUMS
720 CONTINUE ; LOOP END
GO TO 775 ; MIXED PIXELS INCLUDE PURE PIXELS.
730 CONTINUE

```

C  
C  
C

WEIGHTED PIXEL (SUBPIXEL) CALCULATIONS.

```

WPCNT=WPCNT+N0SUBP ; INCREMENT WEIGHTED PIXEL COUNT
WBRPCNT(BNDX)=WBRPCNT(BNDX)+N0SUBP ; BRIGHTNESS HISTOGRAM
WGRPCNT(GNDX)=WGRPCNT(GNDX)+N0SUBP ; GREENNESS HISTOGRAM
DO 740 I=1,4
  WPCS(I)=WPCS(I)+N0SUBP*CHV(PP,I) ; PER CHANNEL SUMS
740 CONTINUE ; LOOP END
  DO 750 I=1,10
  WPRS(I)=WPRS(I)+N0SUBP*CPA(I) ; CROSS PRODUCT SUMS
750 CONTINUE ; LOOP END
IF(CCN.EQ.CLASS)GO TO 760
CCN=CCN+1
GO TO 660
760 CONTINUE

```

C  
C  
C

IF THE 196TH PIXEL HAS BEEN PROCESSED, GO TO THE NEXT LINE.

```

IF(PP.EQ.196)GO TO 770
PP=PP+1
GO TO 430
770 CONTINUE

```

C  
C  
C  
C  
C  
C

IF THE 117TH LINE HAS BEEN PROCESSED, GO TO THE OUTPUT FILE BUILDING SECTION.

WRITE TEMP. STATS. BUFFER TO ALLOW SHARING.

```

C      WRITE(UNIT=UNIT+1)TSREC
      IF(LIMIT(17).GT.790)GO TO 790
      LI=PIXEL/50+1
      GO TO 750
790 CONTINUE

C      ALPHABET INPUT SCREENED IMAGE FILE.
C
C      CLASS(UNIT=4)
C
C      BUILD OUTPUT FILES.
C
      READ(2)
      READ(2)HDR  ! RESET GROUND TRUTH FILE BACK JUST PAST HEADER.
C
C      BUILD HEADERS FOR FILES.
C
      FHADR=ACQ(AN)  ! ACQ. DATE
      FHDR(12)=SUNANG  ! SUN ANGLE
      FHDR(13)=GPC  ! GOOD PIXEL COUNT
      FHDR(14)=XPC  ! GARBLED PIXEL COUNT
      FHDR(15)=CPC  ! CLOUD PIXEL COUNT
      FHDR(16)=SPC  ! SHADOW PIXEL COUNT
      FHDR(17)=WPC  ! WATER PIXEL COUNT
      IF(GPC.GT.0)GO TO 805
      DZ 790 I=19,290
      FHDR(I)=0
790 CONTINUE  ! LOOP END
      GO TO 835
805 CONTINUE
      DZ 810 I=1,4
      RHDR(I)=TCHAI
      RHDR(I)=RHDR(I)/GPC  ! OVERALL CHANNEL MEANS
810 CONTINUE  ! LOOP END
      CALL GAMGEN(SUNANG,RHDR,GAM)  ! GAMMA
      DZ 820 I=1,6
      FHDR(I+27)=PCC(I)  ! PIXEL CLASS COUNTERS
820 CONTINUE  ! LOOP END

C      STORE CROP CODE ARRAY AND NUMBER OF CROP CODES.
C
      NCCC=0
      DO 830 I=1,256
      IF(CCA(I).EQ.0)GO TO 830
      NCCC=NCCC+1
      FHDR(I+34)=NCCC
830 CONTINUE  ! LOOP END
      FHDR(34)=NCCC
835 CONTINUE

C      WRITE HEADER RECORD TO THE FILES.
C
      DZ 840 I=1,4
      WRITE(UNIT=UNIT)FHDR
840 CONTINUE  ! LOOP END

C      TRANSFER DWT PIXEL STATS FROM TMP. TO FINAL FILE.
C
      DZ 850 I=297,267
      READ(3)DWTBUF
      WRITE(5)DWTBUF

```

```

850 CONTINUE
    RE 1005
    READ(5)HDR
C
C     PRINT THE FIELDSTAT ACQUISITION SUMMARY REPORT.
C
C     CALL FSRPT(6,PRATE)
C
C     PRINT THE FIELDSTAT CORE PIXEL REPORT, IF DESIRED.
C
C     IF(1.A.D.FLG)GOTO 870,880,870
870 CONTINUE
    CALL FSRPT(6,PRATE,5)
880 CONTINUE
    CLOSE(UNIT=5)
C
C     RETRIEVE THE INTERMEDIATE STATS RECORD FOR EACH OCCURRING CROP
C     CODE.
C
C     IF(GPC.EQ,0)GOTO 1025
    DO 1020 I=1,255
    IF(CCA(I))1020,1020,900
900 CONTINUE
    READ(3,CCA(I))TSREC
C
C     STORE CROP CODE.
    FSREC(I)=TSREC(I)
C
C     PERFORM A LOOP 3 TIMES CALCULATING STATISTICS FOR PURE, MIXED,
C     AND WEIGHTED PIXELS.
C
C     DO 900 J=0,296,148
    FJ=2*J+1  ! WORD POINTER TO PROPER BUFFER SECTION.
C
C     GET PIXEL COUNT AS A REAL NUMBER, IF IT IS ZERO, THEN ZERO
C     PROPER PORTION OF RECORD BUFFER.
C
C     IF(J.EQ,296)GOTO 910
    L=148
    CNT=TSPEC(FJ+29)  ! SINGLE WORD INTEGER TO REAL CONV.
    GO TO 920
910 CONTINUE
    L=282
    CNT=PCNT  ! DOUBLE WORD INTEGER TO REAL CONV.
920 CONTINUE
    IF(CNT.GT,0.0)GOTO 940
    DO 930 K=J+1,J+L
    RFSR(K)=0.0
930 CONTINUE  ! LOOP END
    GO TO 890
940 CONTINUE
C
C     CALCULATE PER CHANNEL MEANS.
C
C     DO 950 K=J+1,J+L
    RFSR(K)=TSR(K)/CNT
950 CONTINUE  ! LOOP END
C
C     CALCULATE PER CHANNEL STD. DEV.
C
C     K=5
    L=4

```

```

DO 960 M=5,8
IF(CM1,GT,1,0)GO TO 955
RFSR(J+M)=0,0
GO TO 960
955 CONTINUE
RT=TSR(J+K)-CM1*RFSR(J+M-4)*RFSR(J+M-4)
IF(RT,GE,0)GO TO 958
RFSR(J+M)=J,0
GO TO 959
958 CONTINUE
RFSR(J+M)=SQRT(RT/(CM1-1,0))
959 CONTINUE
K=K+1
L=L-1
960 CONTINUE ! LOOP END

C
C
C      CALCULATE CROSS-CHANNEL CORRELATIONS.
DO 970 K=1,6
DNM=(CM1-1,0)*RFSR(J+I(J(K))+4)*RFSR(J+I(K(K))+4)
IF(DNM,GT,0)GO TO 965
RFSR(J+K+8)=0,0
GO TO 970
965 CONTINUE
RFSR(J+K+8)=(TSR(J+I(K(K)))-CM1*RFSR(J+I(J(K)))*RFSR(J+I(K(K))))/DNM
970 CONTINUE ! LOOP END

C
C
C      TRANSFER BRIGHTNESS AND GREENNESS HISTOGRAMS TO RECORD BUFFER.
L=296
IF(J,GE,296)I=564
DO 980 K=FJ+29,FJ+L
FSREC(K)=TSREC(K)
980 CONTINUE ! LOOP END
990 CONTINUE ! LOOP END

C
C
C      WRITE THE RECORDS TO THE OUTPUT FILES.
WRITE(1)FSREC(1),PPSTT,ZERRES
WRITE(7)FSREC(1),MPSTT,ZERRES
WRITE(8)FSREC(1),WPSTT,SUBZ

C
C
C      PRINT THE FIELDSTAT CROP CODE SUMMARY REPORT, IF DESIRED.
IF(2,AND,FLG)1010,1010,1000
1000 CONTINUE
CALL FDCSRP(6,PDATE)
1010 CONTINUE
1020 CONTINUE ! LOOP END
1025 CONTINUE

C
C
C      CLOSE OUTPUT FILES.
DO 1027 K=2,4
CLOSE(UNIT=UNIT+2(K))
1027 CONTINUE ! LOOP END
1030 CONTINUE

C
C
C      IF THE NUMBER OF ACQUISITIONS DESIRED HAVE BEEN FOUND, THEN
      GO TO END PROCESSING, OTHERWISE LOOP BACK FOR NEXT ACQ.
IF(CM1,GE,CMACC)GO TO 1040

```

```
AN=4000
GC T 120
1040 CONTINUE
C
C      END PROCESSING.
C
CLOSE (UNIT=2)
WRITE (6,4027) ' IMAGE PROCESSOR QUIRK - FORCE LAST PRINT BUFFER
4027 FORMAT(1H1)
STOP
END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

#### 3.4.4 FLDSTT SUBROUTINES

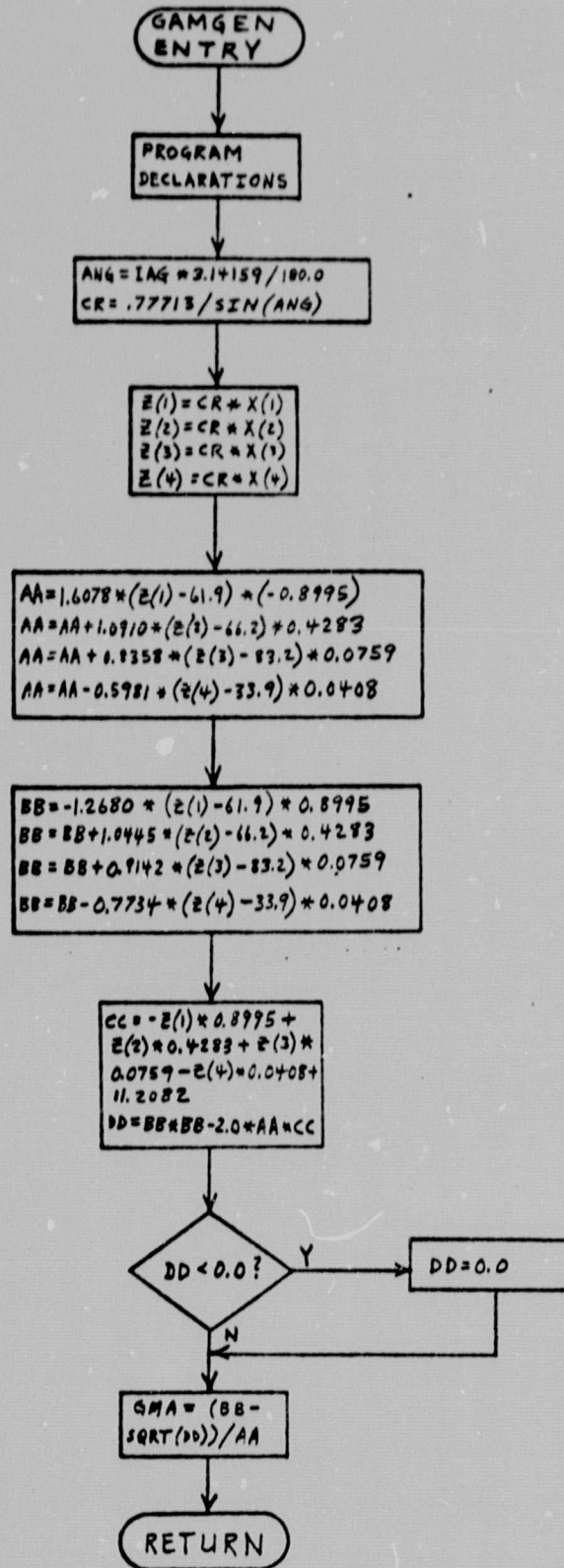
##### 3.4.4.1 General

The special subroutines called by FLDSTT are GAMGEN, SCREEN, GETCCR, FASRPT, FGPRPT, and FCCSRP. SCREEN has an additional entry point SNAG, referenced by FLDSTT.

##### 3.4.4.2 Subroutine GAMGEN

This subroutine uses the sun angle and the four-channel mean vector of good pixels to compute the optical depth parameter, gamma.

### 3.4.4.2a Flowchart





### 3.4.4.2b Listing

```

SUBROUTINE GAMGEN(IAG,X,GMA)
C   GAMGEN COMPUTES THE XSTAR OPTICAL DEPTH PARAMETER GAMMA
C   IAG IS SUN ELEVATION ANGLE IN DEGREES
C   X IS 4 CHANNEL MEAN VECTOR OF 'G2D' PIXELS
C   GMA IS THE HAZE PARAMETER OUTPUT
REAL X(4),GMA,Z(4)
INTEGER *2 IAG
C.....
C.....
ANG=IAG *3.14159 /180.0
CR=.77713/SIN(ANG)
C   IN CASE OF NON PHASE 2 DATA INSERT CALIBRATION HERE
Z(1)=CR*X(1)
Z(2)=CR*X(2)
Z(3)=CR*X(3)
Z(4)=CR*X(4)
AA=1.6078*(Z(1)-61.9)*(-0.8995)
AA=AA+1.0910*(Z(2)-66.2)*0.4233
AA=AA+0.8358*(Z(3)-83.2)*0.0159
AA=AA-0.5981*(Z(4)-33.9)*0.0408
BB=-1.2680*(Z(1)-61.9)*0.8995
BB=BB+1.0445*(Z(2)-66.2)*0.4233
BB=BB+0.9142*(Z(3)-83.2)*0.0159
BB=BB-0.7734*(Z(4)-33.9)*0.0408
CC=-Z(1)*0.8995+Z(2)*0.4233+Z(3)*0.0759-Z(4)*0.0408+11.2082
DD=BB*BB-2.0*AA*CC
IF(DD.LT.0.0) DD=0.0
GMA=(BB-SQRT(DD))/AA
C   WRITE(16,2020)GMA
2020  FORMAT(//,'   GAMMA =',F8.4,/)
RETURN
END

```

```

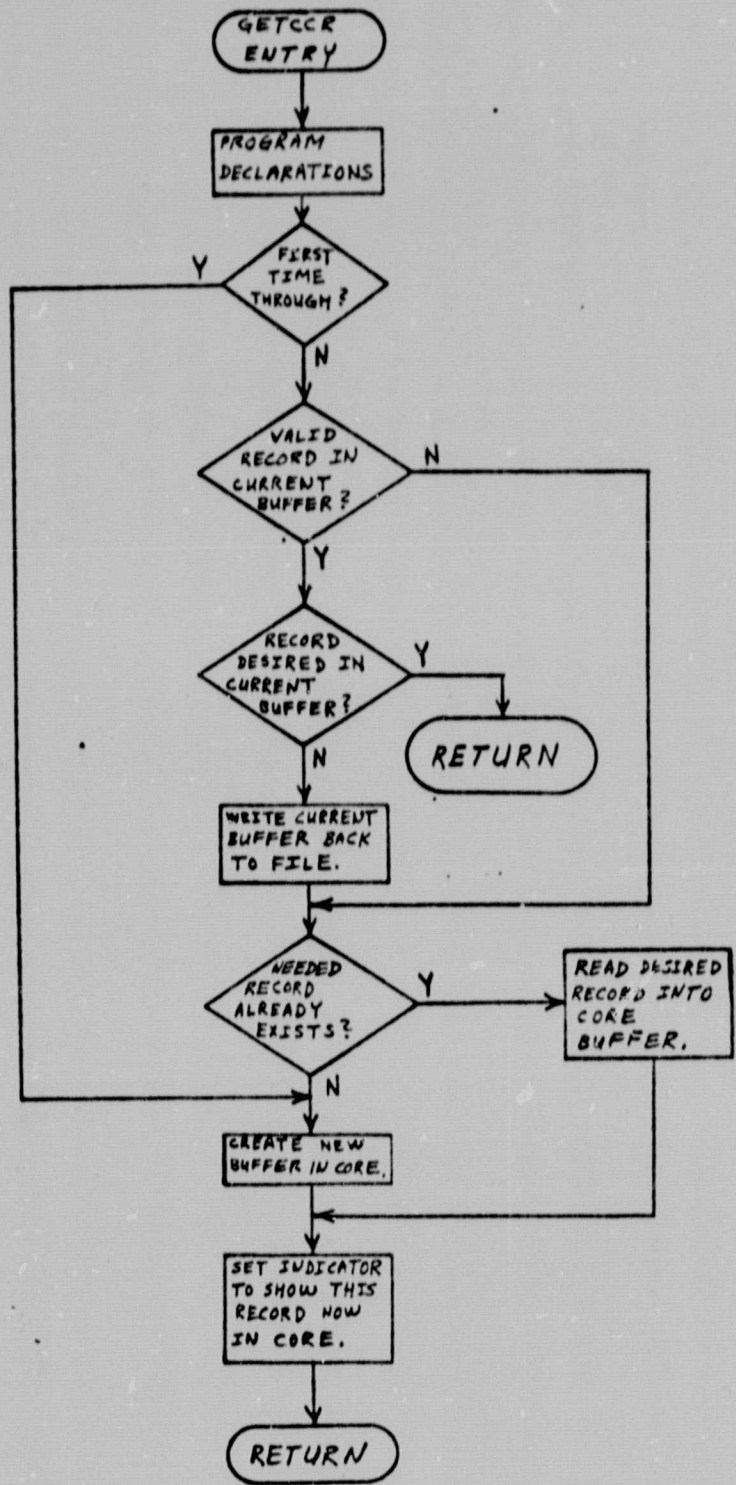
XXC00010
XXC00020
XXC00030
XXC00040
XXC00050
XXC00060
XXC00070
XXC00080
XXC00090
XXC00100
XXC00110
XXC00120
XXC00130
XXC00140
XXC00150
XXC00160
XXC00170
XXC00180
XXC00190
XXC00200
XXC00210
XXC00220
XXC00230
XXC00240
XXC00250
XXC00260
XXC00270
XXC00280
XXC00290
XXC00300
XXC00320

```

3.4.4.3 Subroutine GETCCR

This subroutine reads the direct access record corresponding to an input crop code and stores the record into a designated buffer area.

3.4.4.3a Flowchart



3.4.4.3b Listing

```

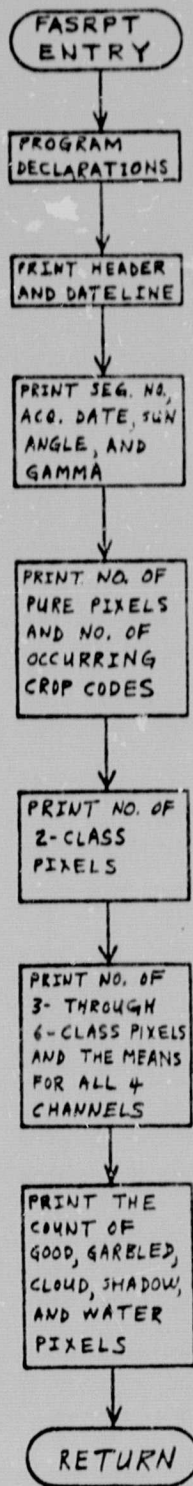
SUBROUTINE GETCOR(UNIT,CRPCD,BUFFER)
C
C   THIS SUBROUTINE READS FROM THE INPUT UNIT NO, THE DIRECT
C   ACCESS RECORD CORRESPONDING TO THE INPUT CRP CODE AND STORES
C   IT INTO THE DESIGNATED BUFFER,
C
  IMPLICIT INTEGER*2 (A-Z)
  DIMENSION BUFFER(1160)  ! RECORD BUFFER
  COMMON/CARRY/NCR,CCA(256),CCIC  ! CRP CODE ARRAY, REC. POINTERS
C
C   FIRST TIME THROUGH, SET UP BUFFER,
C
  IF(NCR.EQ.0)GO TO 7
C
C   WRITE CURRENT BUFFER IF ANOTHER CRP CODE RECORD NEEDED,
C
  IF(CCIC.EQ.-1)GO TO 5
  IF(CCIC.EQ.CRPCD)RETURN
  WRITE(UNIT'CCA(CCIC+1))BUFFER
5  CONTINUE
C
C   IF RECORD ALREADY EXISTS, GO TO 30.
C
  IF(CCA(CRPCD+1).NE.0)GO TO 30
7  CONTINUE
C
C   CREATE A NEW RECORD.
C
  BUFFER(1)=CRPCD
  DO 10 I=2,1160
    BUFFER(I)=0
10  CONTINUE
  NCR=NCR+1  ! INCREMENT NO. OF CRP CODE RECORDS.
  CCA(CRPCD+1)=NCR  ! STORE NEW POINTER INTO CRP CODE ARRAY.
20  CONTINUE
  CCIC=CRPCD  ! SET THIS CRP CODE AS CURRENT CODE IN CORE.
  RETURN
30  CONTINUE
C
C   READ RECORD FROM FILE.
C
  READ(UNIT'CCA(CRPCD+1))BUFFER
  GO TO 20
END

```

#### 3.4.4.4 Subroutine FASRPT

This subroutine uses information in common block RCMN1, which is the header record of any of the FLDSTT output files, to print the Fieldstats Acquisition Summary Report. The report gives the segment number, the acquisition date, the sun angle, gamma, the count of each class and type of pixel, the number of different crop codes found, and the overall means for each of the four channels of image unload data.

3.4.4.4a Flowchart



### 3.4.4.4b Listing

```

SUBROUTINE FASRPT(PRNTR, PDATE)
C
C THIS SUBROUTINE PRINTS THE FIELDSTATS ACQUISITION SUMMARY
C REPORT. INPUT IS
C
C PRNTR = INTEGER PRINTER UNIT NUMBER,
C
C PDATE = 9-BYTE ARRAY CONTAINING THE CURRENT DATE (ASCII),
C
C FHDR = 360-WORD ARRAY CONTAINING THE HEADER RECORD FROM
C THE FIELDSTATS FILE. (/RCHN1/)
C
C
C IMPLICIT INTEGER*2 (A-Z)
C INTEGER*4 ACCD
C BYTE PDATE(9)
C COMMON/RCHN1/FHDR(360)
C DIMENSION PCNT(5), PCLSS(6)
C REAL M(4), GAM
C EQUIVALENCE (PCNT, FHDR(13)), (M, FHDR(18)), (PCLSS, FHDR(28)),
C (SEG, FHDR(9)), (ACCD, FHDR(10)), (SA, FHDR(12)),
C (GAM, FHDR(26)), (MCC, FHDR(34))
C WRITE(PRNTR, 4000)PDATE
4000 FORMAT(1H1, 20X, 'FIELDSTAT ACQUISITION SUMMARY'/31X, 9A1)
C WRITE(PRNTR, 4001)SEG, ACCD, SA, GAM
4001 FORMAT(4X, 14, ' SEGMENT'/3X, 15, ' DATE'/6X, 12, ' SUN ANGLE'/1X,
C F7, 4, ' GAMMA')
C WRITE(PRNTR, 4002)PCLSS(1), MCC
4002 FORMAT(1H0, 8X, 15, ' PURE PIXELS', 16X, 13, ' CRAP CODES',
C ' PROCESSED')
C WRITE(PRNTR, 4003)PCLSS(2)
4003 FORMAT(9X, 15, ' 2-CLASS PIXELS')
C D2 10 I=1, 4
C WRITE(PRNTR, 4004)PCLSS(1+2), I+2, M(I), I
4004 FORMAT(9X, 15, 13, ' 1-CLASS PIXELS', 18X, F9, 4, ' CHANNEL', 12, ' MEAN')
C 10 CONTINUE
C WRITE(PRNTR, 4005)PCNT
4005 FORMAT(//9X, 15, ' GOOD PIXELS'//9X, 15, ' GARBLED PIXELS'/9X,
C 15, ' CLOUD PIXELS'/9X, 15, ' SHADOW PIXELS'/9X, 15,
C ' WATER PIXELS')
C RETURN
C END

```

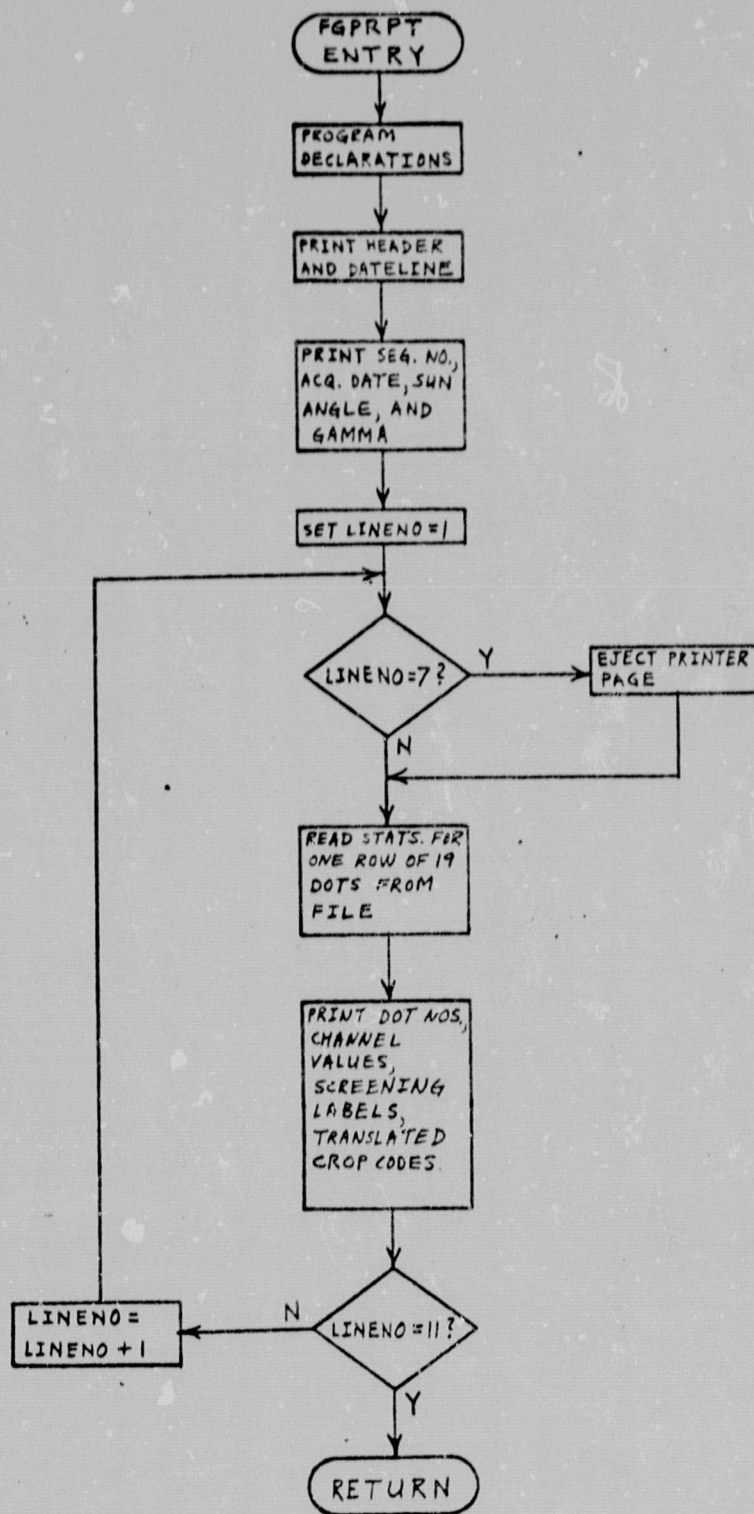
ORIGINAL PAGE IS  
OF POOR QUALITY

#### 3.4.4.5 Subroutine FGPRPT

This subroutine uses file header information in common block RCMN1, crop code translation information in common block RCMN3, and information read from the dot pixel statistics file to print the Fieldstats Dot Pixel Statistics Report. The report gives the segment number, the acquisition date, the sun angle, gamma, and, for each of the 209 dot pixels, the dot pixel number, the four image unload channel values, the screening label, and the six subpixel crop code translations.



3.4.4.5a Flowchart



### 3.4.4.5b Listing

```

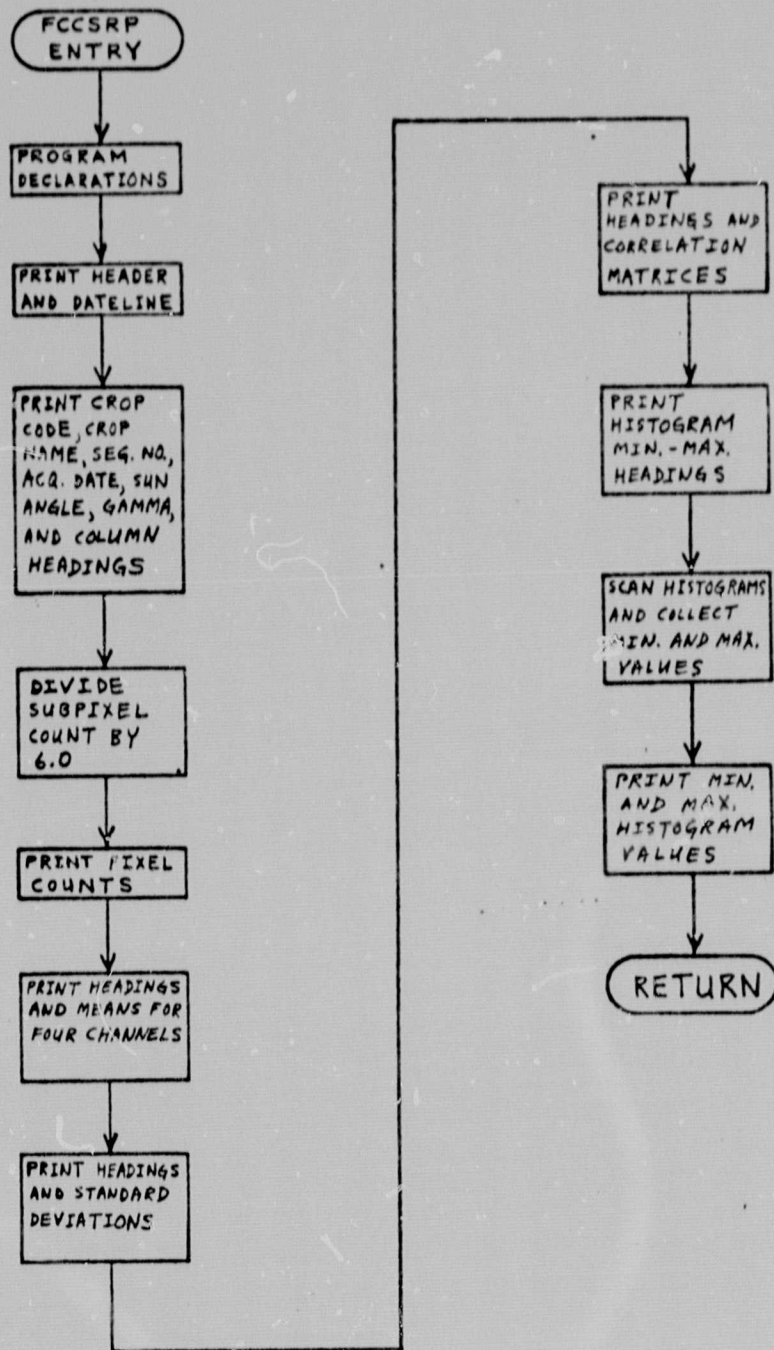
SUBROUTINE FCRPRT(PRNTR,PLATE,UNIT)
C
C   THIS SUBROUTINE PRINTS THE FIELDSTATS GRID PIXEL REPORT,
C   INPUT IS
C
C   PRNTR = INTEGER PRINTER UNIT NUMBER,
C   PDATE = 9-BYTE ARRAY CONTAINING THE CURRENT DATE (ASCII),
C   UNIT = INTEGER INTERMEDIATE INPUT UNIT NUMBER,
C
C   CCT = INTEGER*2 ARRAY CONTAINING CR2P CODE
C   TRANSFORMATIONS, (/RCMN3/)
C
C   FHDR = INTEGER*2 ARRAY CONTAINING THE FILE HEADER, (/RCMN1/)
C
C   IMPLICIT INTEGER*2 (A-Z)
C   DIMENSION BUFFER(270)
C   COMMON/RCMN1/FHDR(360),/RCMN3/CCT(256)
C   BYTE PDATE(9)
C   INTEGER*4 ACCDTE
C   REAL GAMMA
C   EQUIVALENCE (SEG,FHDR(9)),(ACCDTE,FHDR(10)),
C   (SUNANG,FHDR(12)),(GAMMA,FHDR(26))
C
C   PRINT HEADER,
C
C   WRITE(PRNTR,4000)PDATE
C   4000 FORMAT(1H1,42X,'FIELDSTAT GRID PIXEL REPORT'/52X,9A1)
C   WRITE(PRNTR,4001)SEG,ACCDTE,SUNANG,GAMMA
C   4001 FORMAT(4X,I4,' SEGMENT'/3X,I5,' DATE'/6X,I2,' SUN ANGLE'/1X,
C   * F7,4,' GAMMA'/)
C   LINENO=1
C   10 CONTINUE
C   IF(LINENO,EO.7)WRITE(PRNTR,4006)
C   4006 FORMAT(1H1)
C   READ(UNIT)BUFFER
C   WRITE(PRNTR,4002)(BUFFER(I),I=1,217,12)
C   4002 FORMAT(1X,19(13,3X))
C   DO 20 K=1,4
C   WRITE(PRNTR,4003)(BUFFER(I+K),I=1,217,12)
C   4003 FORMAT(19(3X,13))
C   20 CONTINUE
C   WRITE(PRNTR,4004)(BUFFER(I),I=12,228,12)
C   4004 FORMAT(2X,19(12,4X))
C   DO 30 K=5,10,2
C   WRITE(PRNTR,4005)(CCT(BUFFER(I+K)-1),CCT(BUFFER(I+K+1)+1),
C   * I=1,217,12)
C   4005 FORMAT(1X,39(A2,1X))
C   30 CONTINUE
C   IF(LINENO,EO.11)RETURN
C   LINENO=LINENO+1
C   GO TO 10
C   END

```

#### 3.4.4.6 Subroutine FCCSRP

This subroutine uses file header information in common block RCMN1, crop code translation information in common block RCMN3, and a buffer of per crop code information in common block RCMN2 to print the Fieldstats Crop Code Summary Report. The buffer in common block RCMN2 is a composite of records read from each of the three pixel type output statistics files from FLDSTT. The report gives, for each occurring crop code, the crop code numeric value, the crop code translation, the segment number, the acquisition date, the sun angle, gamma, and then for each pixel type (pure, mixed, and subpixels), the count of pixels, the four channel means, the four channel standard deviations, the cross-channel correlations, and the maximum and minimum values from the brightness and greenness histograms.

### 3.4.4.6a Flowchart



3.4.4.6b Listing

```

SUBROUTINE FCCSRP(PRNTR, PDATE)
C
C THIS SUBROUTINE PRINTS THE FIELDSTATS CR2P CODE SUMMARY REPORT.
C INPUT IS
C
C PRNTR = INTEGER PRINTER UNIT NUMBER.
C
C PDATE = 9-BYTE ARRAY CONTAINING THE CURRENT DATE (ASCII).
C
C BUFFER = INTEGER*2 ARRAY CONTAINING A CR2P CODE STATISTICS
C RECORD FROM THE FIELDSTATS FILE. (/RCMN2/)
C
C FHDR = INTEGER*2 ARRAY CONTAINING THE FILE HEADER. (/RCMN1/)
C
C CCT = INTEGER*2 ARRAY CONTAINING CR2P CODE
C TRANSFORMATIONS. (/RCMN3/)
C
C IMPLICIT INTEGER*2 (A-Z)
C COMMON /RCMN1/ FHDR(360), /RCMN2/ BUFFER(1160), /RCMN3/ CCT(256)
C BYTE PDATE(9)
C DIMENSION MM(12)
C INTEGER*4 ACCDTE, DIBUF(268)
C REAL GAMMA, WPCNT, RBUF(578)
C EQUIVALENCE (BUFFER(2), RBUF), (BUFFER(622), DIBUF),
C (SEG, FHDR(9)), (ACCDTE, FHDR(10)),
C (SUNANG, FHDR(12)), (GAMMA, FHDR(26))
C
C PRINT HEADER.
C
C WRITE(PRNTR, 4000) PDATE
4000 FORMAT(1H1, 21X, 'FIELDSTAT CR2P CODE SUMMARY'/31X, 9A1)
C WRITE(PRNTR, 4001) BUFFER(1), CCT(BUFFER(1)+1), SEG, ACCDTE, SUNANG,
C GAMMA
4001 FORMAT(5X, I3, ' CR2P CODE'/6X, A2, ' CR2P NAME'/4X, I4, ' SEGMENT'/3X,
C 15, ' DATE'/6X, I2,
C ' SUN ANGLE'/1X, F7.2, ' GAMMA'/7' -----PURE PIXELS----- ',
C ' -----MIXED PIXELS-----' -----SUBPIXELS-----'/)
C WPCNT=DIBUF(1)/6.0
C WRITE(PRNTR, 4002) BUFFER(30), BUFFER(326), WPCNT
4002 FORMAT(1X, I5, ' PIXELS', 12X, I5, ' PIXELS', 11X, F8.1, ' PIXELS'/)
C WRITE(PRNTR, 4003)
4003 FORMAT(3(6X, ' MEANS', 13X))
C
C PRINT MEANS FOR EACH CHANNEL.
C
C DO 10 I=1,4
C WRITE(PRNTR, 4004) I, RBUF(I), I, RBUF(148+I), I, RBUF(296+I)
4004 FORMAT(1X, 3(I1, 2X, F9.4, 12X))
C 10 CONTINUE
C WRITE(PRNTR, 4005)
4005 FORMAT(/3(4X, ' STANDARD', 12X)/3(3X, ' DEVIATIONS', 11X))
C
C PRINT STANDARD DEVIATIONS FOR EACH CHANNEL.
C
C DO 20 I=1,4
C WRITE(PRNTR, 4004) I, RBUF(4+I), I, RBUF(152+I), I, RBUF(300+I)
C 20 CONTINUE
C WRITE(PRNTR, 4006)
4006 FORMAT(/3(7X, ' CORRELATIONS', 5X))
C
C PRINT CORRELATION MATRIX (UPPER HALF).

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

C      WRITE(PRINTR,4007)RBUF(9),RBUF(10),RBUF(11),RBUF(157),RBUF(159),
      RBUF(159),RBUF(305),RBUF(306),RBUF(307)
4007  FORMAT(3(' 1.000',3F6.3))
      WRITE(PRINTR,4008)RBUF(12),RBUF(13),RBUF(160),RBUF(161),
      RBUF(308),RBUF(309)
4008  FORMAT(3(7X,'1.000',2F6.3))
      WRITE(PRINTR,4009)RBUF(14),RBUF(162),RBUF(310)
4009  FORMAT(3(13X,'1.000',F6.3)/3(19X,'1.000'))
      WRITE(PRINTR,4010)
4010  FORMAT(/3(5X,'MIN  MAX',11X))

C      CALCULATE AND PRINT HISTOGRAM MINIMUMS AND MAXIMUMS.
C
C      L=1  ; MIN/MAX ARRAY MIN POINTER,
C      J=31  ; POINTER TO HISTOGRAM ARRAY,
C      K=0  ; HISTOGRAM OFFSET BELOW ZERO,
C      N=150 ; NO. OF VALUES (=1) IN HISTOGRAM ARRAY,
30  CONTINUE

C      INITIALIZE MIN AND MAX VALUES AND LOOP INDEX I.
C
C      MM(L)=0  ; MIN.
C      MN(L+1)=0 ; MAX.
C      I=J
40  CONTINUE
      IF(BUFFER(I).GT.0)GO TO 50
      I=I+1
      IF(I.GT.J+N)GO TO 80
      GO TO 40
50  CONTINUE
      MN(L)=I-J-K
60  CONTINUE
      MM(L+1)=I-J-K
70  CONTINUE
      I=I+1
      IF(I.GT.J+N)GO TO 80
      IF(BUFFER(I).GT.0)GO TO 60
      GO TO 70
80  CONTINUE
      IF(L.NE.1)GO TO 90
      L=7
      J=182
      K=30
      N=115
      GO TO 30
90  CONTINUE
      IF(L.NE.7)GO TO 100
      L=3
      J=327
      K=0
      N=150
      GO TO 30
100 CONTINUE
      IF(L.NE.3)GO TO 110
      L=9
      J=478
      K=30
      N=115
      GO TO 30
110 CONTINUE

```

C REPEAT PROCEDURE EXCEPT DOUBLE WORD INTEGERS  
C ARE USED FOR THE WEIGHTED PIXEL HISTOGRAMS.  
C

```
L=5  
J=2  
K=0  
N=150  
120 CONTINUE  
MM(L)=0  
MM(L+1)=0  
I=J  
130 CONTINUE  
IF(DIBF(I).GT.0)GO TO 140  
I=I+1  
IF(I.GT.J+N)GO TO 170  
GO TO 130  
140 CONTINUE  
MM(L)=I-J-K  
150 CONTINUE  
MM(L+1)=I-J-K  
160 CONTINUE  
I=I+1  
IF(I.GT.J+N)GO TO 170  
IF(DIBF(I).GT.0)GO TO 150  
GO TO 160  
170 CONTINUE  
IF(L.NE.5)GO TO 180  
L=11  
J=153  
K=30  
N=115  
GO TO 120  
180 CONTINUE  
WRITE(PRINT,4011)MM  
4011 FORMAT(3(' BR ',I4,1X,I4,11X)/3(' GR ',I4,1X,I4,11X))  
RETURN  
END
```

## 4. OPERATIONS

### 4.1 OPERATOR'S GUIDE

#### 4.1.1 IMUNLD PROGRAM

When the operator receives an IMUNLD batch run, he should perform the following steps:

1. Log onto the system under UIC[5,5].
2. Check system time and date for correct entry.
3. Mount the Image Unload tape (specified on the Run Request Form, Figure 4-1) without a write ring as a foreign tape.
4. Mount the AA RP04 disk (specified on the Run Request Form) on DB2:.
5. Check the IMUNLD card deck to ensure that the second card after the \$CREATE IMUNLD.DAT card contains the tape drive ID that matches the drive on which the tape was mounted.
6. Load the card reader with the IMUNLD card deck and start the batch processor.
7. At the end of the run dismount the tape and disk, and return the card deck, Run Request Form (signed off), Image Unload tape, and printer listing to the user, and the disk to its locker.

#### 4.1.2 FLDSTT PROGRAM

When the operator receives a FLDSTT batch run, he should perform the following steps:

1. Log onto the system under UIC[5,5].
2. Check system time and date for correct entry.
3. Mount the AA RP04 disk (specified on the Run Request Form) on DB2:.



AEROJET ( )	NAME	REQUEST/DPAR	SEQ. #
BENDIX ( )			
G E 100 ( )	ORGANIZATION/PHONE	DATA	DATE
ANALOG ( )			
DIGITAL GRAPHICS ( )			

INPUT INFORMATION

TAPE IDENTIFICATION	MISSION	FLIGHT	SITE	TYPE
START	OTHER INFORMATION			
STOP				
START				
STOP				

OUTPUT INFORMATION

PRODUCT	FORMAT	FILM B&W/COLOR	TYPE
PROGRAM	VISICORDER	PRINTOUT	SPECIAL

REQUESTER REMARKS/JUSTIFICATION

---



---



---



---



---



---



---



---



---



---



---

OPERATOR REMARKS

---



---



---



---

ORIGINAL PAGE IS  
OF POOR QUALITY

4. Load the card reader with the FLDSTT card deck and start the batch processor.
5. At the end of the run, dismount the disk and return it to its locker, and return the card deck, Run Request Form (signed off), and printer listing to the user.

#### 4.2 USER'S GUIDE

##### 4.2.1 BATCH PROCEDURES

##### 4.2.1.1 IMUNLD Program

To run the IMUNLD program in batch, the user should perform the following steps:

1. Prepare the following deck of cards:  
\$JOB/NAME=IMUNLD/LIMIT=999/ACCOUNT=110 6/MCR  
\$CREATE IMUNLD.DAT  
    Data Set 1  
\$EOD  
\$MCR REM RSXBAT  
\$RUN IMUNLD.TSK  
\$EOJ

Where Data Set 1 contains the three cards described below:

Card 1 - contains a 2 in column one to indicate that DB2: is the drive for the AA disk.

Card 2 - contains the input tape drive device ID in the first three columns, for example, MT0 or MT1. This card will be adjusted by the operator if he needs to use a different drive than the one specified.

Card 3 - Contains the tape number in the first six columns. This number will be checked against the number stored internally on the tape

2. Complete the NAME, ORGANIZATION/PHONE, DATE, TAPE IDENTIFICATION, OTHER INFORMATION (specify what AA RP04 disk to use) boxes, and check the PRINTOUT box on the Run Request Form (Figure 4-1).

3. Submit the Image Unload tape, the card deck, and the Run Request Form to Operations for running on the Image Processor.
4. After the program has been run, retrieve the tape, card deck, and printout, and sign the bottom of the Run Request Form to indicate receipt of the output.

#### 4.2.1.2 FLDSTT Program

To run the FLDSTT program in batch, the user should perform the following steps:

1. Prepare the following deck of cards:
 

```

$JOB/NAME=FLDSTT/LIMIT=999/ACCOUNT=110 6/MCR
$MCR PIP FLDSTT.DAT;*/DE
$CREATE FLDSTT.DAT
      Data Set 2
$EOD
$MCR REM RSXBAT
$RUN FLDSTT.TSK
$EOJ
      
```

Where Data Set 2 contains the four to six cards described below:  
 Card 1 - contains, in the first thirty columns, the file ID where the crop code translations and brightness and greenness coefficients are stored. The file name is FSTLBL.DAT, but this card should include the specific device and user file directory to enable location of the desired file, for example, DB2:[131,ØØ1] FSTLBL.DAT.

Card 2 - contains, in the first thirty columns, the file ID where ground truth data are stored. After this file is assigned, this character string is modified and used to create the output files, therefore it must be very exact in format. For example, the file ID DB2:[131,ØØ1]161177353.GTØ is acceptable (the acquisition date must be in columns eighteen through twenty-two).

Card 3 - contains three fields:

Field 1 - columns one through four - contains the segment number to be processed. This is checked against the number stored internally on the ground truth file.

Field 2 - columns six and seven - contains the output report option flag. This is a two-digit decimal number specifying which output reports are desired. The number is the sum of the numbers assigned to the various reports. The Acquisition Summary report is always produced, whereas the Dot Pixel Statistics report (assigned number 1) and the Crop Code Summary report (assigned number 2) are optional.

Field 3 - columns nine and ten - contains a two-digit decimal number, greater than zero and less than sixteen, which specifies how many acquisition dates are to be processed during this run of the FLDSTT program. Note that each acquisition date processed will take one hour of elapsed time at a minimum.

Cards 4,5, and 6 - contain fifteen fields which are the acquisition dates to be processed, represented in standard YYDDD format. On each card there are five five-character fields, starting in column one and skipping five spaces between each field. Cards 5 and 6 do not need to be present if they are not needed.

2. Complete the NAME, ORGANIZATION/PHONE, DATE, OTHER INFORMATION (specify what AA RP04 disk to use) boxes, and check the PRINTOUT box on the Run Request Form.
3. Submit the card deck and Run Request Form to Operations for running on the Image Processor.
4. After the program has been run, retrieve the card deck and printout, and sign the bottom of the Run Request Form to indicate receipt of the output.

C-2

~~4-4~~

94

## 4.2.2 ONLINE PROCEDURES

### 4.2.2.1 IMUNLD Program

To run the IMUNLD program from an interactive terminal, the user should sign on to UIC[110,6], have the proper tape and disk mounted, create the file IMUNLD.DAT containing Data Set 1 (described in 4.2.1.1), and execute IMUNLD.TSK.

### 4.2.2.2 FLDSTT Program

To run the FLDSTT program from an interactive terminal, the user should sign on to UIC[110,6], have the proper disk mounted, edit the data in Data Set 2 (described in 4.2.1.2) in file FLDSTT.DAT and execute FLDSTT.TSK.