JSC-13666-REV-A
8.0-10.200
NASA CR-160632

"AS-BUILT" DESIGN SPECIFICATION FOR PDP 11/45 ACCURACY ASSESSMENT SYSTEM

Job Order 71-695

(TIRFS 77-0060 & 77-0063)

(E80-10200) AS-BUILT DESIGN SPECIFICATION FOR PDP 11/45 ACCURACY ASSESSMENT SYSTEM (Lockheed Electronics Co.) 190 p HC A09/MF A01 CSCL 05B G3/43

N80-28797

Unclas 00200

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

December 1977

LEC-11358
Revision A

JSC- 13666

"AS-BUILT" DESIGN SPECIFICATION
FOR
PDP 11/45 ACCURACY ASSESSMENT SYSTEM

Job Order 71-695

(TIRFS 77-0060 & 77-0063)

Prepared By

C. W. Ahlers

APPROVED BY

for James A Wilkinson

Philip L. Krumm, Supervisor
Applications Software Section

Prepared By

Lockheed Electronics Company, Inc.

For

Earth Observations Division

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

December 1977

LEC-11358
REVISION A

CONTENTS

1. SCOPE

1.1 GENERAL

This specifies the detailed design for CAMS classification and proportion estimation accuracy assessment software implemented for background operation on the PDP 11/45. Included are:

a. Two input data preprocessors. The first is a stand-alone program which scans a DTRM tape and produces a printed file directory for that tape. The second is a stand-alone program for the generation of Accuracy Assessment Phase I input data tapes (B Tapes) from Bendix output field vertices data tapes.

b. A two unit software module for preparation of a proper ground truth data tape in LACIE Universal format (see Reference 1) and a utility program for printout of those data for monitoring puproses.

c. A three unit functional software module for development of accuracy assessment parameters from input ground truth data, image data and analyst labeled dot data. The accuracy assessment parameters are (see Appendix B):

   1. True wheat proportion
   2. Maximum likelihood proportion estimate
   3. Classification and pixel counting proportion estimate
   4. Probability of misclassification
   5. Variance of Procedure 1 proportion estimate
   6. Proportion of wheat pixels on field boundaried
   7. Dots labeled by ground truth
   8. Probability of misclassification of analyst labeled dots

2. APPLICABLE DOCUMENTS

The following documents, of exact issue shown, form parts of the specification to the extent specified herein.

2.1 REFERENCE 1

Earth Resources data Formate Control Book PHO-TR543, Volume 1, Revision A, Change 1, pages 7.1-9 through 7.1-25 provides complete definition of the "Universal" format of the DTRM and Bendix-100 output tapes. This is the Reference 1 (one) of the text.

TIRF 77-0060, Modify the BTREAD Program

TIRF 77-0063, Nov 77 Implementation of SIGMAP, SPECTL and ALLCRP

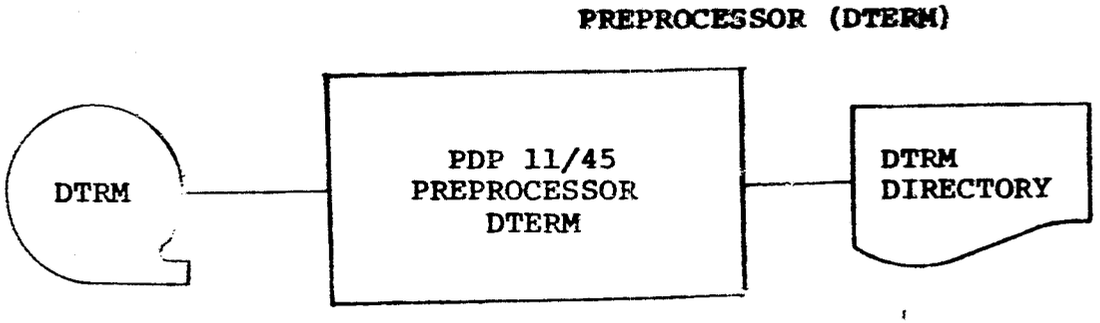
"As Built" Design Specification for PDP 11/45 Accuracy Assessent System (TIRF 77-0030), Oct 77 (LEC 11358, JSC 13666)

3. SYSTEM DESCRIPTION

The software system implemented on the PDP 11/45 for background generation of CAMS accuracy assessment indices, consist of two input data preprocessors (Figure 1), an intermediate optional unit for monitoring of ground truth data, and three two-unit functional modules (Figure 2). One preprocessor scans the DTRM tape to be input and reports the file directory of that tape. The second preprocessor, converts the Bendix 100 output data tape (field vertices in NOVA floating point) to a proper "B" tape equivalent (field vertices in PDP/45 integer form). The first functional module constructs a "ground truth" data file from field vertices data input on the accuracy assessment (Phase 1) input data tape ("B" tape), along with corresponding crop identifications input in punched cards. At this point, the user may apply the utility program to produce maps of the ground truth data for their examination before proceeding. The second functional module compares operational ERIPS and analysts classification results with the ground truth data to produce indices for the assessment of the accuracy of those classifications.

The initial step of the process is preparation of a proper "B" tape. This is application of that preprocessor which converts a Bendix 100 output data tape to a proper "B" tape data file equivalent.

The second processing step is application of the first software module to the field vertices data input on the "B" tape (field vertices file), along with corresponding field crop identifications input in punched cards to construct and produce a "ground truth" data tape in Universal format. This step may include printout of ground truth data maps if the user so desires, but only if the special utility (mapping) program is applied.


PREPROCESSOR (DTERM)
DTRM
PDP 11/45
PREPROCESSOR
DTERM
DTRM
DIRECTORY

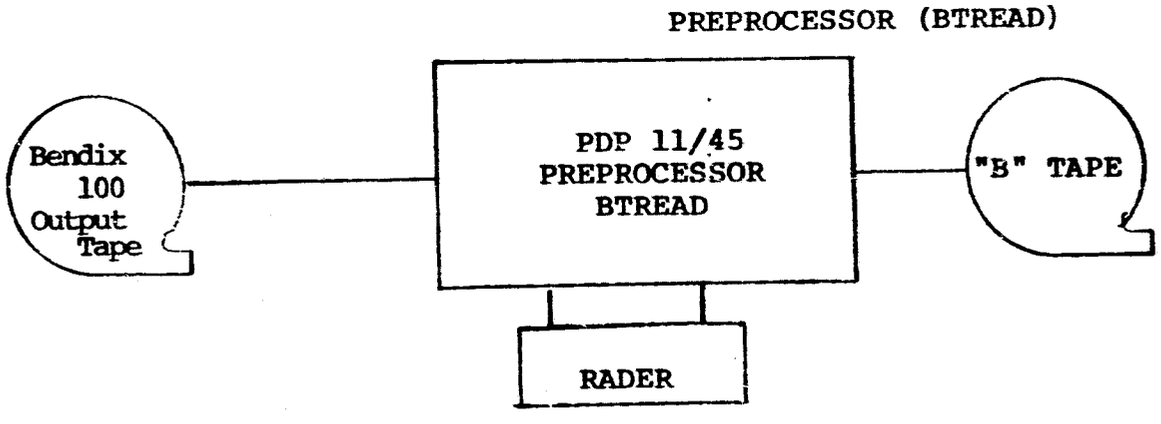
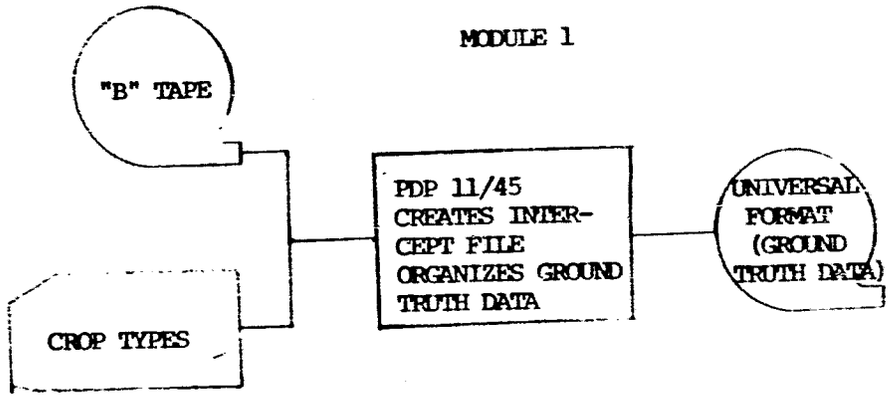
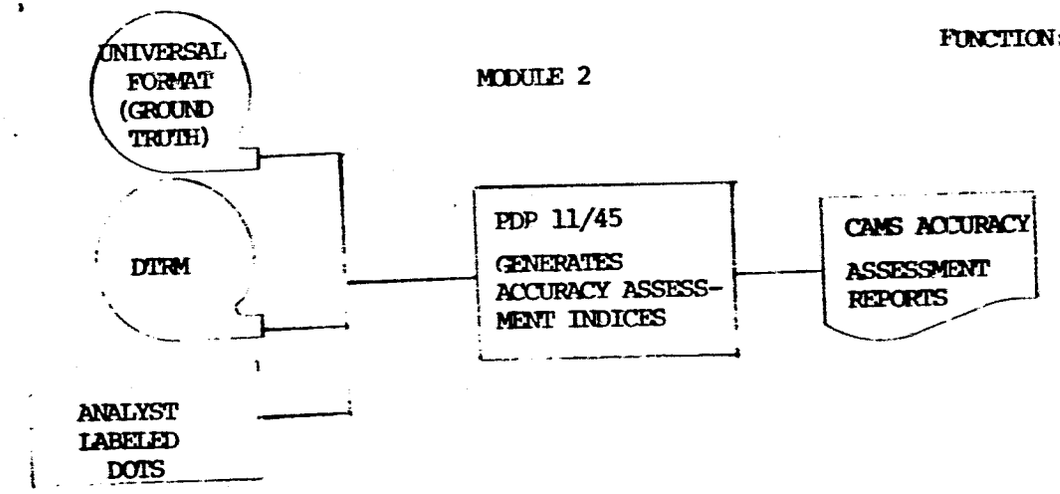

PREPROCESSOR (BTREAD)
Bendix
100
Output
Tape
PDP 11/45
PREPROCESSOR
BTREAD
"B" TAPE
RADER

Figure 1

PHASE 1 CAMS ACCURACY ASSESSMENT SOFTWARE SYSTEM



FUNCTION: Operates on field vertices and crop type data to produce ground truth registered to Landsat data on a sub-pixel level. Output is ground truth data tape in Universal format.



FUNCTION: Operates on input data to produce:

1) True wheat proportion
2) Maximum likelihood proportion estimate
3) Classification and pixel counting proportion estimate
4) Probability of misclassification
5) Variance of Procedure 1 proportion estimate
6) Proportion of wheat pixels on field boundaries
7) Dots labeled by ground truth
8) Probability of misclassification of analyst labeled dots

Figure 2

The final processing step is application of the final functional module to compare ground truth data with corresponding classification data. In the application, each unit of the module is executed separately. The first unit restricts the comparison of ground truth data to analyst "dot" labeling data to produce certain of the accuracy assessment indices. The second and third unit compares ground truth data to both analyst dot labeling data and ERIPS classification results (DTRM magnetic tape). Prior to execution of this second unit, a preprocessor is applied to the DTRM tape to define its directory.

3.1 HARDWARE DESCRIPTION

PDP 11/45 with the following peripheral units:

1. Card reader
2. Line printer
3. Tape unit (2)
4. Disk unit

3.2 SOFTWARE DESCRIPTION

This section presents brief functional descriptions of the two preprocessing units, the utility (mapping) program and the two functional modules of the CAMS accuracy assessment system, all of which were designed to be compatible with the PDP RSX-11D operating system. Included in the descriptions are all subroutines and subroutine interrelationships.

All preprocessing units are stand-alone operating programs. The preprocessor DTERM is used to construct and report a DTRM tape file directory for use as a source of a program control input for the functional modules ALLCRP and MILTCRP. The other preprocessor, BTREAD, is used to construct an accuracy assessment (Phase 1) ("B" tape) file for use as input to the first functional module.

The first unit of the first functional module edits the field vertices entered from the "B" tape to insure that they are proper for field boundary definitions. It then defines, for each field, the field boundary and associates the proper crop type (derived from card inputs) with it. It next defines the points at which that boundary intercepts the field dot lines. Finally, it constructs an internal file of those intercepts for use as input to the second unit of the module. For its operations, this first unit (PHASE 1) calls the subroutines S-01, S-12, S-23, S-34, S-45, S-55 and S-56. The second unit of this module (PHASE 2) employs only standard system utility routines to manipulate and restructure data from the "intercept" file produced by PHASE 1, for production and output of a proper "ground truth" tape (magnetic tape in Universal format). Subsequently, at the users option, maps of that ground truth can be generated and printed out through application of the utility (mapping) program SGMAP.

The second functional software unit consists of three stand-alone units SPATL, ALLCRP and MLTCRP. All of these units employ standard system utility routines and certain special subroutines to accomplish their comparison of ground truth data with classification data counterparts and subsequent calculations. Both units compute a specific set of accuracy assessment parameters. The first unit SPATL restricts its comparison to analyst labeled dots, while the second two units, ALLCRP and MLTCRP include both analyst labeled dots and ERIPS classification data in its comparison.

3.2.1 PREPROCESSOR (DTERM)

3.2.1.1 Linkage

This is a stand alone program which calls only standard system utility routines.

3.2.1.2 Interface

None.

3.2.1.3 Input

DTRM tape (see reference 1 for format description).

3.2.1.4 Output

Labeled printout of tape file directory including tape identification and for each tape file the following: file number, site number and acquisition date (day, month and year)

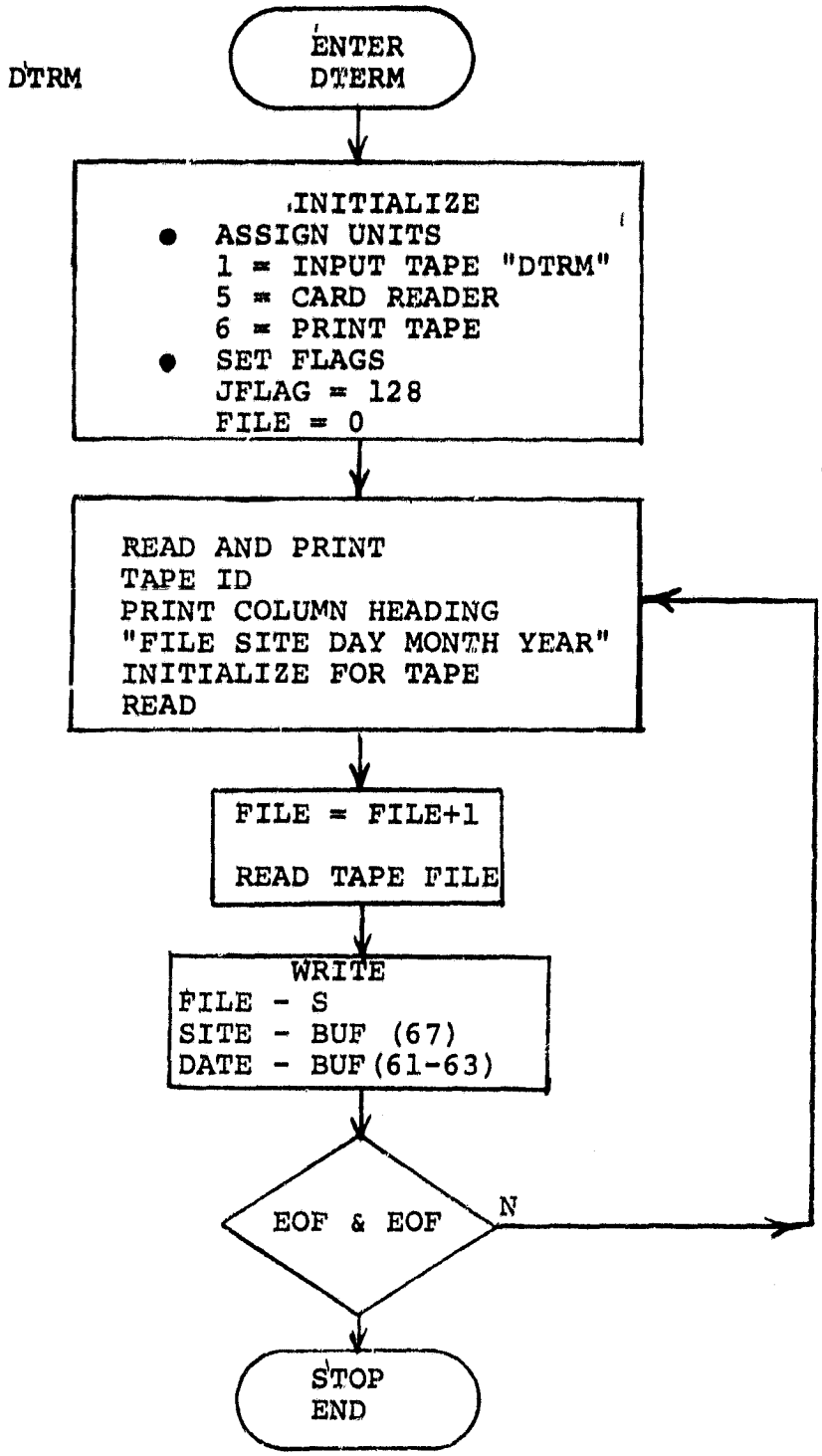
3.2.1.5 Storage

TBD

3.2.1.6 Description

DTERM reads the DTRM tape, extracts and reports the tape identification, locates and numbers each tape file, then constructs and reports a tape file directory.

3.2.1.7 Flowchart



3.2.1.8 Listing

```
      IMPLICIT INTEGER(A-Q),(S-Z)
      BYTE BUF(3060),T(8),D(9),A(80)
      EQUIVALENCE (S,BUF(67))
      COMMON /STATUS/W1,W2
        CALL TIME(T)
        CALL DATE(D)
        NPRT=6
        WRITE(NPRT,703) D,T
  703   FORMAT(1H1,'   JOB INITIATED ON ',9A1,' AT ',8A1,//,10X,
     1'PROGRAM DTERM,FTN')
      NRDR=5
      JFLAG=128
      FILE=0
        OPEN(UNIT=NRDR,NAME='DTERM.DAT',TYPE='OLD',ACCESS='SEQUENTIAL',
     1FORM='FORMATTED',CARRIAGE CONTROL='NONE')
        READ(NRDR,301) SDEV,NDEV
  301   FORMAT(A1,1X,I2)
        WRITE(NPRT,302) SDEV,NDEV
  302   FORMAT(//,10X,A1,'T',10X,'DEVICE NO.=',I5)
        IDEV=0
        IF(SDEV.EQ.'X') IDEV=1
        IF(NDEV.NE.0.AND.NDEV.NE.1) GO TO 500
        CALL TINIT(1,IDEV,NDEV)
      CALL  TATCH(1)
      CALL  TRWD(1)
      WRITE(NPRT,101)
  101 FORMAT('        FILE     SITE      DAY       MONTH     YEAR')
    1 CONTINUE
      CALL  TREAD(1,BUF,1530)
      CALL  TWAIT(1)
      JEOF=JAND(JFLAG,W1)
      IF(JEOF.EQ.128) GO TO 2
C  ASSUMES THAT END OF DATA IS SIGNALED BY 2 CONSECUTIVE EOFS
      CALL SWAB(S)
      FILE=FILE+1
      WRITE(NPRT,102) FILE,S,(BUF(I),I=61,63)
  102 FORMAT(1H ,5I10)
      CALL  TFILE(1,1)
      CALL  TWAIT(1)
      GO TO 1
    2 CONTINUE
      CALL  TRWD(1)
      CALL  TWAIT(1)
  500   CONTINUE
      CALL DATE(D)
      CALL TIME(T)
      WRITE(NPRT,103) D,T
      WRITE(NPRT,103) D,T
  103 FORMAT('  JOB COMPLETED ON  ',9A1,'  AT  ',8A1)
      STOP
      END
```

3-8 10

3.2.2 PREPROCESSOR (BTREAD)

3.2.2.1 Linkage

This program calls the special subroutines RADER, LABEL and FSORT, in addition to the standard system tape manipulation toutines.

3.2.2.2 Interface

RADER, LABEL, FSORT.

3.2.2.3 Input

Magnetic tape output from Bendix 100 system (Appendix A).

3.2.2.4 Output

"B" tape formatted file for direct input to Accuracy Assessment Phase 1 program (Appendix A)and a file of crop labels which is used by the program PHASE 2.

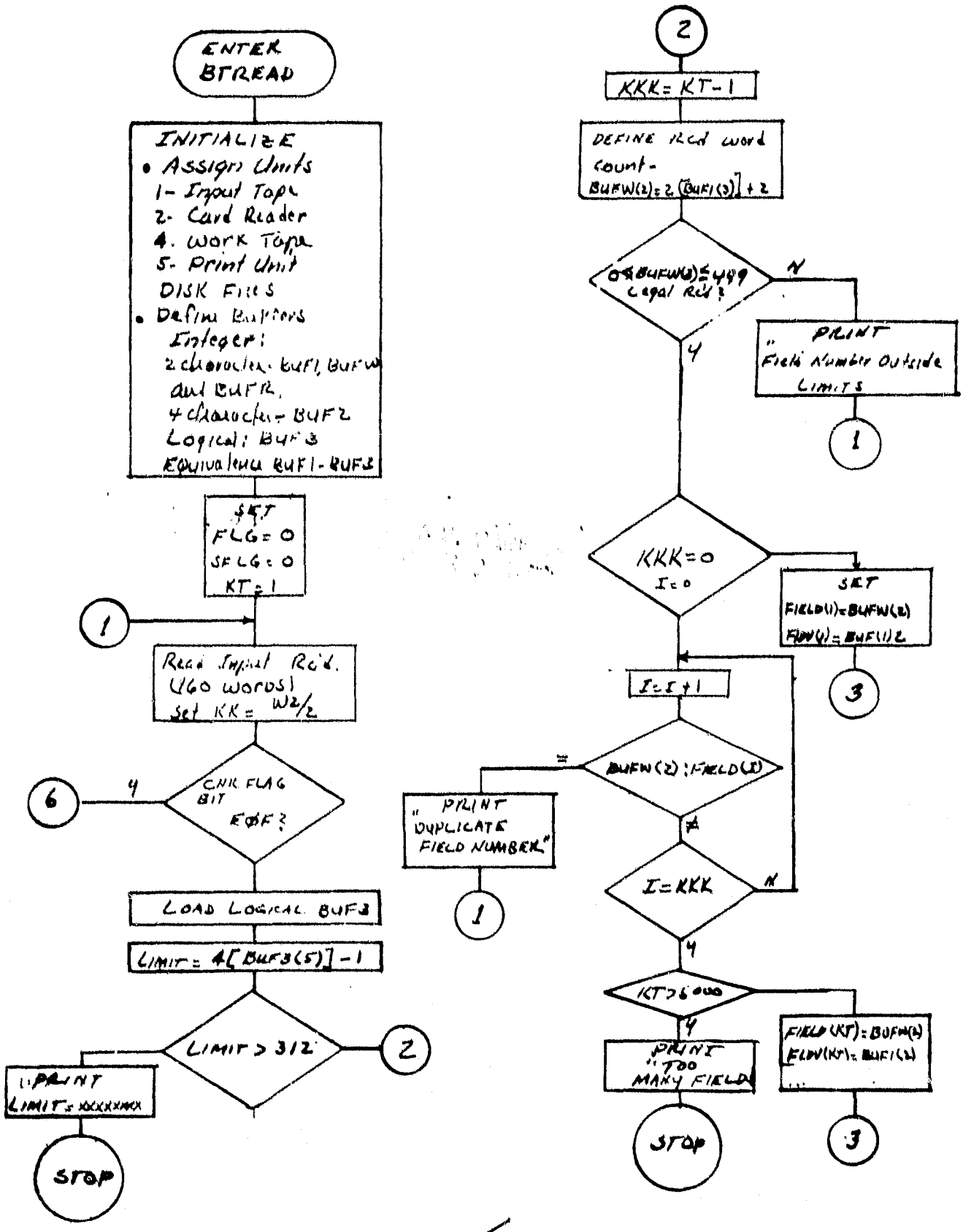
3.2.2.5 Storage

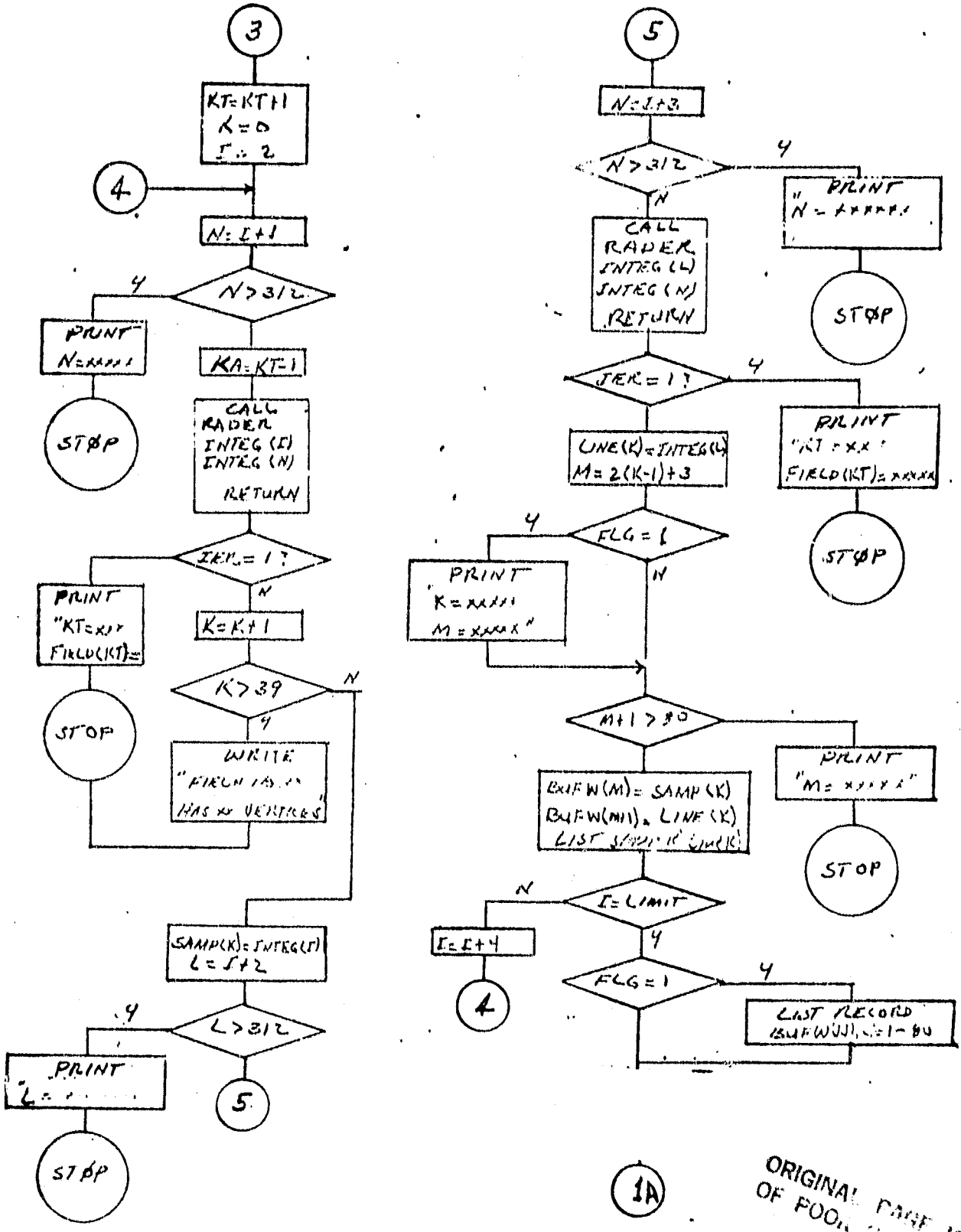
TBD

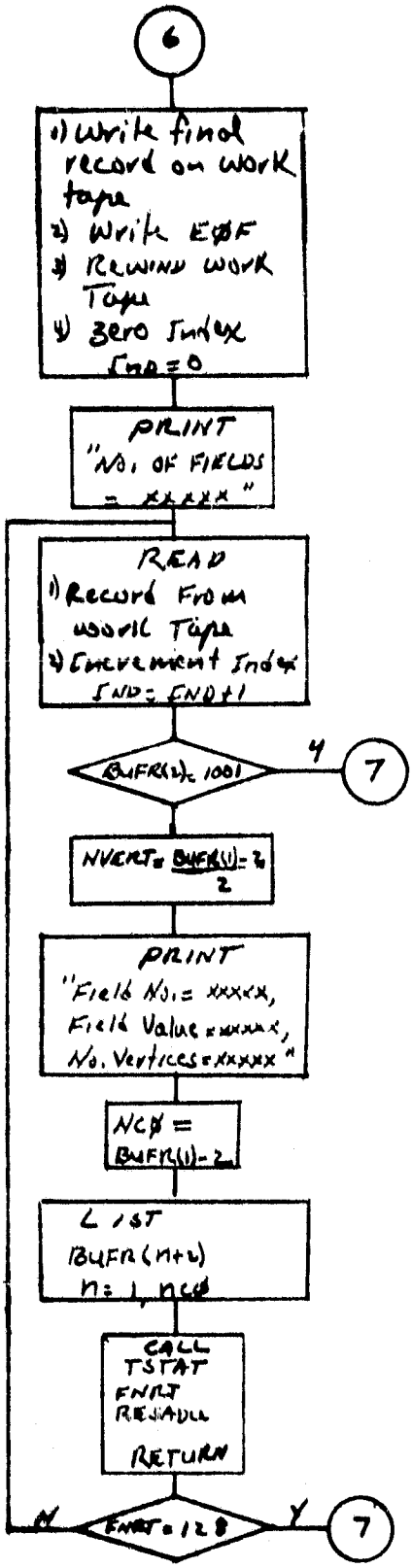
3.2.2.6 Description

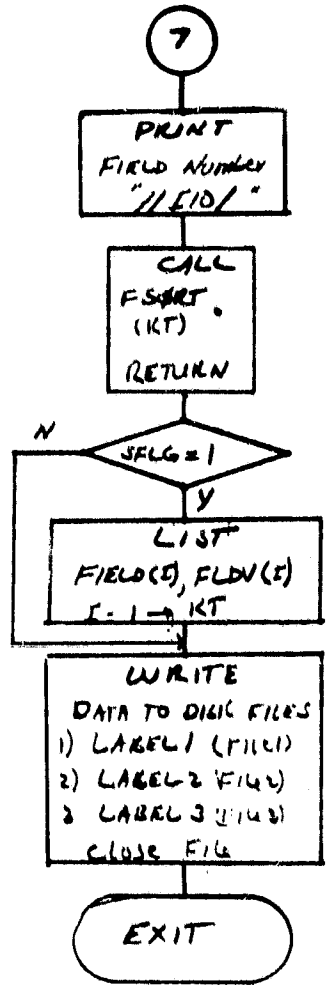
This program edits a Bendix 100 output data tape, extracts and translates acceptable field vertices data closes fields if possible and constructs a proper "B" tape file from them.

3.2.2.7 Flow Chart






6
1) Write final record on work tape
2) Write EOF
3) Rewind work Tape
4) Zero Index END=0
PRINT
"No. OF FIELDS = XXXXX"
READ
1) Record From work Tape
2) Increment Index IND=END+1
BUFR(2)=1001
4
7
NVERT=BUFR(1)-2/2
PRINT
"Field No.= XXXXX, Field Value=XXXXX, No. Vertices=XXXXX"
NCØ=BUFR(1)-2
LIST
BUFR(n+2)
n=1, nCØ
CALL TSTAT ENRT REJADD
RETURN
N
ENRT=128
Y
7


7
PRINT
FIELD NUMBER "//E10/"
CALL FSART. (IKT)
RETURN
N
SFLG=1
Y
LIST
FIELD(I), FLDV(I)
I=1→IKT
WRITE
DATA TO DIGI FILES
1) LABEL1 (FILE1)
2) LABEL2 FILE2)
3 LABEL 3 FILE3)
CLOSE FILE
EXIT

3.2.8 Listing

```
      IMPLICIT INTEGER (A-Z)
      INTEGER*2 BUF1(80),BUFW(80),BUFR(80)
      INTEGER*2 INTEG(312),SAMP(100),LINE(100)
      INTEGER*4 BUF2(40)
     LOGICAL*1 BUF3(160)
      EQUIVALENCE (BUF1,BUF3)
      EQUIVALENCE (BUF3(5),INTEG(1))
      EQUIVALENCE (BUF3(7),BUF2)
      BYTE T(8),D(9)
     COMMON /STATUS/W1,W2
      COMMON /ERROR/IER,NPNT
      COMMON /BT/NRDR,NPRT,FIELD(500),FLDV(500)
      CALL TIME(T)
      CALL DATE(D)
      NPRT=5
      NPNT=NPRT
      FLG=0
      SFLG=1
      WRITE(NPRT,703) D,T
  703 FORMAT(1H1,'  JOB INITIATED ON  ',9A1,'  AT  ',8A1,//,10X,
     1'PROGRAM BTREAD,FTN')
      NRDR=2
      CALL ASSIGN(NRDR,'BTREAD,DAT')
      READ(NRDR,704) RSDEV,RNDEV,RFILE
  704 FORMAT(A1,1X,2I2)
      WRITE(NPRT,705) RSDEV,RNDEV,RFILE
  705 FORMAT(1H0,10X,'B100 TAPE',//,10X,A1,'T',10X,
     1'DEVICE NO.=',I5,10X,'FILE NO.=',I5)
      RIDEV=0
      IF(RSDEV.EQ.'X') RIDEV=1
      READ(NRDR,704) WSDEV,WNDEV,WFILE
      WRITE(NPRT,706) WSDEV,WNDEV,WFILE
  706 FORMAT(//,10X,'SIMULATED B100 TAPE',//,10X,A1,'T',
     110X,'DEVICE NO.=',I5,10X,'FILE NO.=',I5)
      WIDEV=0
      IF(WSDEV.EQ.'X') WIDEV=1
      READ(NRDR,501) SGNO,DAY,MO,YR
  501 FORMAT(4I5)
      WRITE(NPRT,502) SGNO,DAY,MO,YR
  502 FORMAT(1H0,10X,'SEGMENT NUMBER=',I5,5X,'DAY=',I5,5X,
     *  'MONTH=',I5,5X,'YEAR=',I5)
      CALL TINIT(1,RIDEV,RNDEV)
      CALL TATCH(1)
      CALL TRWD(1)
      RFILE=RFILE+1
      CALL TFILE(1,RFILE)
      CALL TWAIT(1)
      CALL TINIT(4,WIDEV,WNDEV)
      CALL TATCH(4)
      CALL TRWD(4)
      WFILE=WFILE+1
      CALL TFILE(4,WFILE)
      CALL TWAIT(4)
      DO 12 I=1,500
      FLDV(I)=0
   12 FIELD(I)=0
      KT=1
   20 CALL TREAD(1,BUF3,160)
      CALL TWAIT(1)
      KK=W2/2
      CALL TSTAT(1,FNCT,RESIDU)
      BIT7=IAND(FNCT,128)
```

```
      IF(BIT7,EQ,128) GØ TØ 999
      INTERM=BUF3(1)
      BUF3(1)=BUF3(2)
      BUF3(2)=INTERM
      INTERM=BUF3(3)
      BUF3(3)=BUF3(4)
      BUF3(4)=INTERM
      INTERM=BUF3(5)
      BUF3(5)=BUF3(6)
      BUF3(6)=INTERM
      LIMIT=INTEG(1)*4-1
      IF(LIMIT,GT,312) WRITE (NPRT,401) LIMIT
  401 FØRMAT(1HØ,10X,'LIMIT =',I10)
      IF(LIMIT,GT,312) STØP
      KKK=KT-1
      BUFW(1)=BUF1(3)*2+2
      BUFW(2)=BUF1(1)
      IF(BUFW(2),LT,0,OR,BUFW(2),GT,499) GØ TØ 13
      IF(KKK,EQ,0) FIELD(1)=BUFW(2)
      IF(KKK,EQ,0) FLDV(1)=BUF1(2)
      IF(KKK,EQ,0) GØ TØ 15
      DØ 14 I=1,KKK
      IF(BUFW(2),EQ,FIELD(I)) GØ TØ 16
   14 CØNTINUE
      IF(KT,GT,500) WRITE(NPRT,301) KT
      IF(KT,GT,500) STØP
  301 FØRMAT(1HØ,10X,'TØØ MANY FIELDS      KT=',I5)
      FIELD(KT)=BUFW(2)
      FLDV(KT)=BUF1(2)
   15 KT=KT+1
      GØ TØ 21
   16 WRITE(NPRT,17) BUFW(2)
   17 FØRMAT(1X,'DUPLICATE FIELD NUMBER',I10/)
      GØ TØ 20
   13 WRITE(NPRT,18) BUFW(2)
   18 FØRMAT(1HØ,10X,'FIELD NUMBER ØUTSIDE LIMITS',I10)
      GØ TØ 20
   21 K=0
      DØ 850 I=2,LIMIT,4
      N=I+1
      IF(N,GT,312)WRITE(NPRT,402) N
  402 FØRMAT (1HØ,10X,'N = ', I10)
      IF(N,GT,312) STØP
      KA=KT+1
      CALL RADER(INTEG(I),INTEG(N))
      IF (IER,EQ,1) WRITE(NPRT,302) KA,FIELD(KA)
      IF(IER,EQ,1) STØP
  302 FØRMAT(1HØ,10X,'KT=',I5,5X,'FIELD(KT)=',I5)
      K=K+1
      IF(K,GT,39)WRITE(NPRT,403) FIELD(KA),K
  403 FØRMAT(1HØ,10X,'FIELD NØ.',I5,'  HAS ',I5,' VERTICES')
      IF(K,GT,39) STØP
      SAMP(K)=INTEG(I)
      L=I+2
      IF(L,GT,312) WRITE (NPRT,404) L
  404 FØRMAT(1HØ,10X,'L = ',I10)
      IF(L,GT,312) STØP
      N=I+3
      IF(N,GT,312) WRITE(NPRT,402) N
      IF(N,GT,312) STØP
      CALL RADER(INTEG(L),INTEG(N))
      IF(IER,EQ,1) WRITE(NPRT,302) KA,FIELD(KA)
      IF(IER,EQ,1) STØP
      LINE(K)=INTEG(L)
      M=2*(K-1)+3
      IF(FLG,EQ,1) WRITE(NPRT,802) K,M
```

```
  802   FORMAT(1H ,10X,'K=',I5,5X,'M=',I5)
        IF(M+1.GT.80) WRITE(NPRT,405) M
  405   FORMAT(1H0,10X,'M=',I5)
        IF(M+1.GT.80) STOP
        BUFW(M)=SAMP(K)
        BUFW(M+1)=LINE(K)
C       PRINT 11,SAMP(K),LINE(K)
  850   CONTINUE
        IF(FLG.EQ.1) WRITE(NPRT,801) (BUFW(JJ),JJ=1,80)
  801   FORMAT(1H ,10X,2015)
        SHUT=0
        II=BUFW(1)-1
        JJ=II+1
        SL=BUFW(II)
        LL=BUFW(JJ)
        DELS=BUFW(3)-BUFW(II)
        DELL=BUFW(4)-BUFW(JJ)
        DELS=IABS(DELS)
        DELL=IABS(DELL)
        IF(DELS.EQ.0.AND.DELL.EQ.0) GO TO 444
        SHUT=1
        IF(DELS.LE.2) BUFW(II)=BUFW(3)
        IF(DELL.LE.3) BUFW(JJ)=BUFW(4)
        IF(DELS.GT.2.OR.DELL.GT.3) SHUT=2
        IF(SHUT.EQ.1) WRITE(NPRT,445) BUFW(2),SL,LL
  445   FORMAT(1H0,10X,'FIELD NO.  ',I5,5X,'WAS CLOSED.  OLD VERT=',
     1215)
        IF(SHUT.EQ.2.AND.BUFW(1)+2.GT.80) WRITE(NPRT,664) BUFW(2)
  664   FORMAT(1H0,10X,'FIELD NO.  ',I5,5X,'CAN NOT BE CLOSED')
        IF(SHUT.EQ.2.AND.BUFW(1)+2.GT.80) GO TO 444
        IF(SHUT.EQ.2) WRITE(NPRT,446) BUFW(2),BUFW(3),BUFW(4),
     1BUFW(II),BUFW(JJ)
  446   FORMAT(1H0,10X,'FIELD NO. ',I5,5X,'HAS A VERTEX ADDED, ',
     1'OLD VERTICES=',2I5,5X,2I5)
        IF(SHUT.EQ.2) BUFW(JJ+1)=BUFW(3)
        IF(SHUT.EQ.2) BUFW(JJ+2)=BUFW(4)
        IF(SHUT.EQ.2) BUFW(1)=BUFW(1)+2
  444   CONTINUE
        CALL TWRIT(4,BUFW,80)
        CALL TWAIT(4)
   11   FORMAT(1X,2I15)
        GO TO 20
  999   BUFW(2)=1001
        CALL TWRIT(4,BUFW,80)
        CALL TWAIT(4)
        CALL TEOF(4)
        CALL TWAIT(4)
        CALL TRWD(4)
        CALL TFILE(4,WFILE)
        CALL TWAIT(4)
        IND=0
        WRITE(NPRT,120) KA
  120   FORMAT(1H0,10X,'NO. OF FIELDS=',I5)
 9991   CALL TREAD(4,BUFR,80)
        CALL TWAIT(4)
        IND=IND+1
        IF(BUFR(2).EQ.1001) GO TO 9992
        NVERT=(BUFR(1)-2)/2
        WRITE(NPRT,100) BUFR(2),FLDV(IND),NVERT
  100   FORMAT(1H ,10X,'FIELD NO.=',I5,5X,'FIELD VALUE=',I5,5X,
     1'NO. VERTICES=',I5)
        NCO=BUFR(1)-2
        WRITE(NPRT,110) (BUFR(II+2),II=1,NCO)
  110   FORMAT(1H , 8X,2015)
        CALL TSTAT(4,FNRT,RESAND)
        IF(IAND(FNRT,128).NE.128) GO TO 9991      3-16
```

```
 9992     WRITE(NPRT,8) BUFR(2)
 8        FØRMAT(//I10/)
          CALL FSØRT(KT)
          IF(SFLG,NE.1) GØ TØ 631
          DØ 631 I=1,KT
          WRITE (NPRT,632)I, FIELD(I),FLØV(I)
 632      FØRMAT(1H ,10X,315)
 631      CØNTINUE
          CALL CLØSE (NRDR)
          ØPEN(UNIT=NRDR,NAME='LABEL1,DAT',TYPE='ØLD',
         1ACCESS='SEQUENTIAL',FØRM='FØRMATTED',
         2CARRIAGE CØNTRØL='NØNE')
          WRITE(NRDR,505) SGNØ,DAY,MØ,YR
  505     FØRMAT(415)
          DØ 666 FLG=1,2
          CALL LABEL (KT,FLG)
          IF(FLG,EQ,2) GØ TØ 666
          CALL CLØSE(NRDR)
          ØPEN(UNIT=NRDR,NAME='LABEL2,DAT',TYPE='ØLD',
         1ACCESS='SEQUENTIAL',FØRM='FØRMATTED',
         2CARRIAGE CØNTRØL='NØNE')
          WRITE(NRDR,505) SGNØ,DAY,MØ,YR
  666     CØNTINUE
          CALL CLØSE(NRDR)
          ØPEN(UNIT=NRDR,NAME='LABEL3,DAT',TYPE='ØLD',
         1ACCESS='SEQUENTIAL',FØRM='FØRMATTED',
         2CARRIAGE CØNTRØL='NØNE')
          IZ=0
          IM1= -1
          WRITE(NRDR,505)SGNØ,DAY,MØ,YR
          WRITE(NRDR,565) IZ,IZ,IM1
  565     FØRMAT(315)
          CALL CLØSE(NRDR)
          CALL DATE(D)
          CALL TIME(T)
          WRITE(NPRT,333) D,T
  333     FØRMAT(1HØ,10X,'JØB CØMPLETED ØN  ',9A1,'  AT  ',8A1)
          STØP
          END
```

3.2.3 BTREAD SUBROUTINES

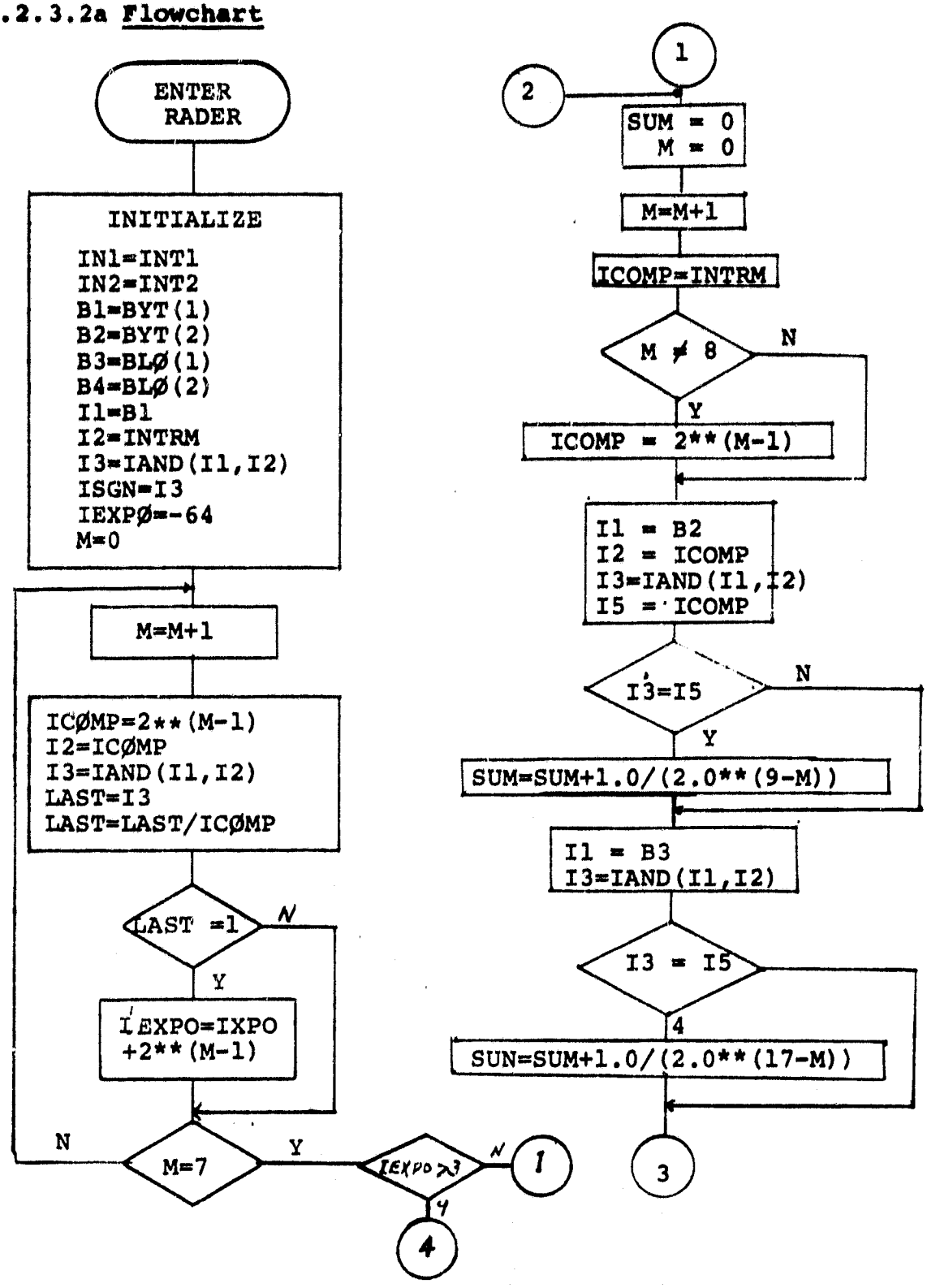
3.2.3.1 General

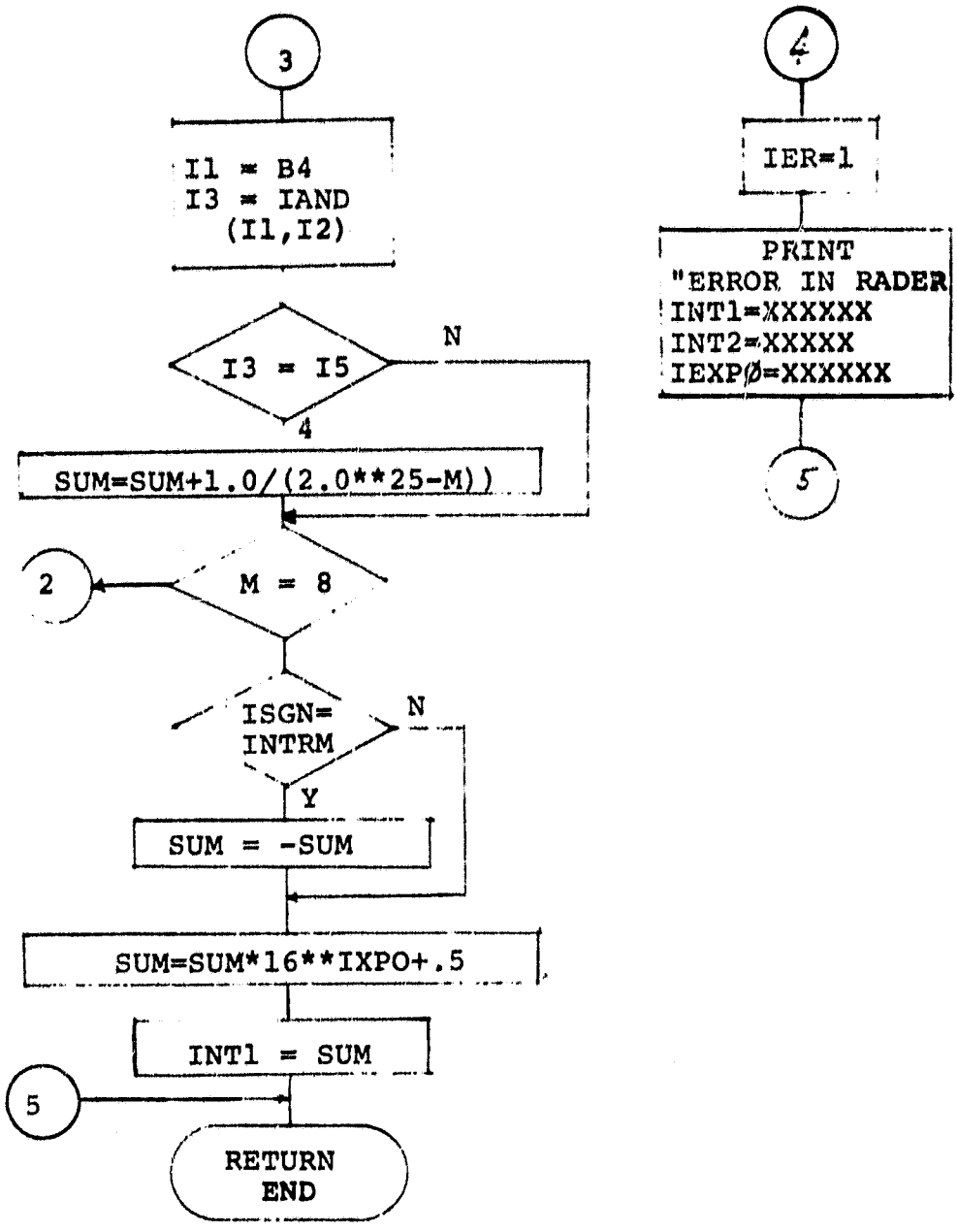
The special subroutines called by BTREAD are RADER, FSORT and LABEL. RADER and FSORT interface only with BTREAD. LABEL interfaces with BTREAD and a subordinate subroutine, COMP. Communication between BTREAD and its subroutines is totally through common.

3.2.3.2 Subroutine RADER

This special subroutine converts NOVA floating point numbers to equivalent PDP 11/45 integers.

3.2.3.2a Flowchart




3
I1 = B4
I3 = IAND
(I1,I2)
I3 = I5
N
4
SUM=SUM+1.0/(2.0**25-M))
2
M = 8
ISGN=
INTRM
N
Y
SUM = -SUM
SUM=SUM*16**IXPO+.5
INT1 = SUM
5
RETURN
END
4
IER=1
PRINT
"ERROR IN RADER
INT1=XXXXXX
INT2=XXXXX
IEXPØ=XXXXXX
5

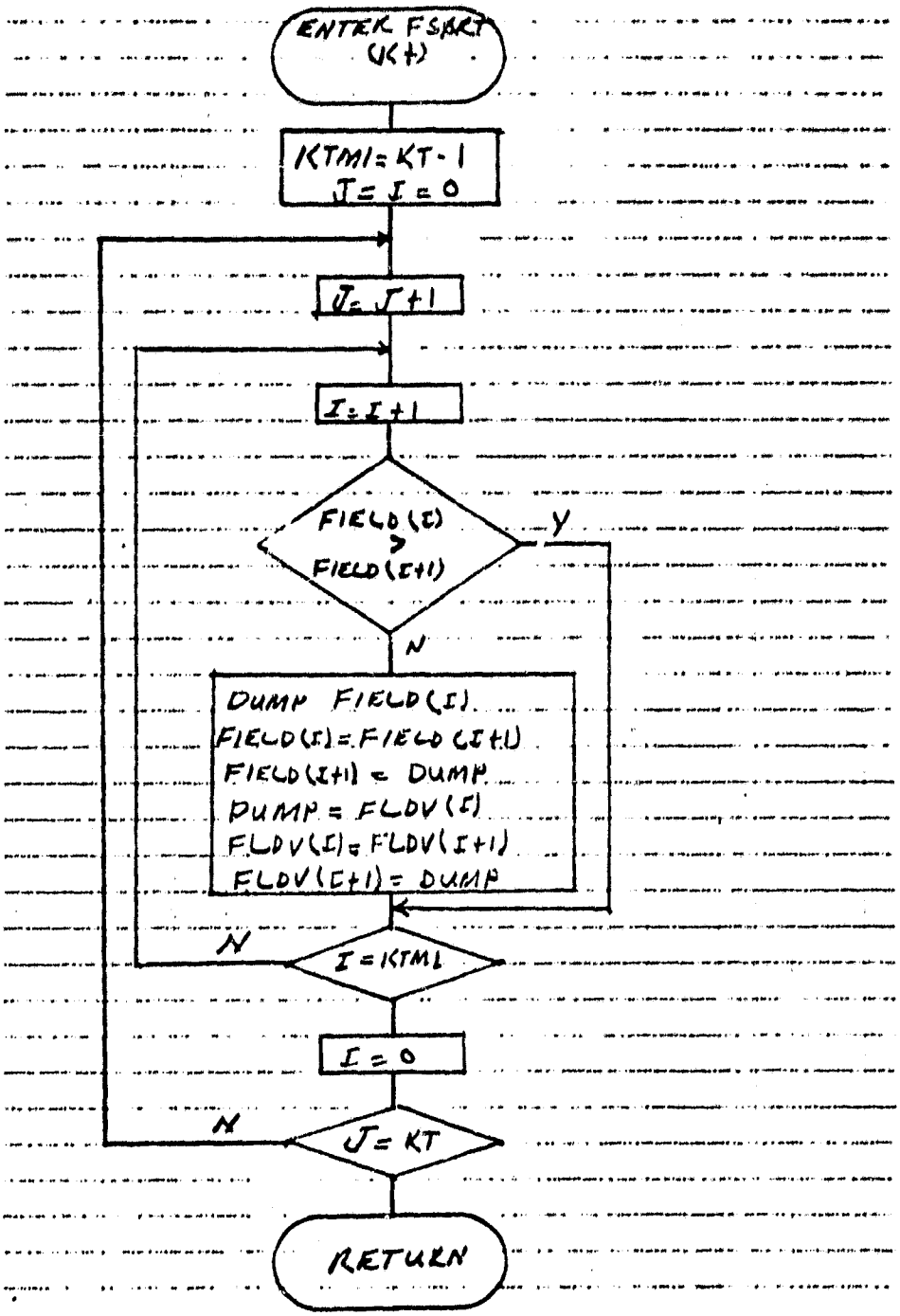
3.2.3.2b Listing

```
      SUBROUTINE RADER(INT1,INT2)
C  THIS FUNCTION CONVERTS A FLOATING POINT WORD IN NOVA EXCESS 64
C     TO A PDP 11/45 INTEGER
C THE 4 COMMON BYTES IN THIS FUNCTION CORRESPOND TO A SINGLE NOVA FLOATING POINT
C VARIABLE,     B1 IS THE LEFTMOST BYTE AND B4 IS THE RIGHTMOST BYTE,
C THE FUNCTION RETURNS THE INTEGER VALUE OF THE ROUNDED NOVA
C FLOATING POINT VARIABLE AS REPRESENTED BY THE 4 BYTES.
      BYTE BYT(2),BLO(2)
      BYTE B1,B2,B3,B4,ICOMP,INTRM,LAST,ISGN
      EQUIVALENCE (IN1,BYT(1)),(IN2,BLO(1))
        COMMON /ERROR/IER,NPRT
      DATA INTRM/0200/
        IER=0
      IN1=INT1
      IN2=INT2
      B1=BYT(1)
      B2=BYT(2)
      B3=BLO(1)
      B4=BLO(2)
      I1=B1
      I2=INTRM
      I3=IAND(I1,I2)
      ISGN=I3
      IEXPO=-64
      DO 20 M=1,7
      ICOMP=2**(M-1)
      I2=ICOMP
      I3=IAND(I1,I2)
      LAST=I3
      LAST=LAST/ICOMP
 20   IF(LAST.EQ.1)IEXPO=IEXPO+2**(M-1)
        IF(IEXPO.GT.3) IER=1
        IF(IEXPO.GT.3) WRITE(NPRT,100) INT1,INT2,IEXPO
 100    FORMAT(1HO,10X,'ERROR IN RADER',10X,'INT1=',I10,
     15X,'INT2=',I10,5X,'IEXPO=',I10)
        IF(IEXPO.GT.3) GO TO 22
      SUM=0
      DO 21 M=1,8
      ICOMP=INTRM
      IF(M.NE.8)ICOMP=2**(M-1)
      I1=B2
      I2=ICOMP
      I3=IAND(I1,I2)
      I5=ICOMP
      IF(I3.EQ.I5)SUM=SUM+1.0/(2.0**(9-M))
      I1=B3
      I3=IAND(I1,I2)
      IF(I3.EQ.I5)SUM=SUM+1.0/(2.0**(17-M))
      I1=B4
      I3=IAND(I1,I2)
 21   IF(I3.EQ.I5)SUM=SUM+1.0/(2.0**(25-M))
      IF(ISGN.EQV.INTRM)SUM=-SUM
      SUM=SUM*16**IEXPO+.5
      INT1=SUM
 22     CONTINUE
      RETURN
      END
```

3.2.3.3 Subroutine FSØRT

Subroutine FSØRT arranges the field entries in numerical order.

3.2.3.3a Flowchart



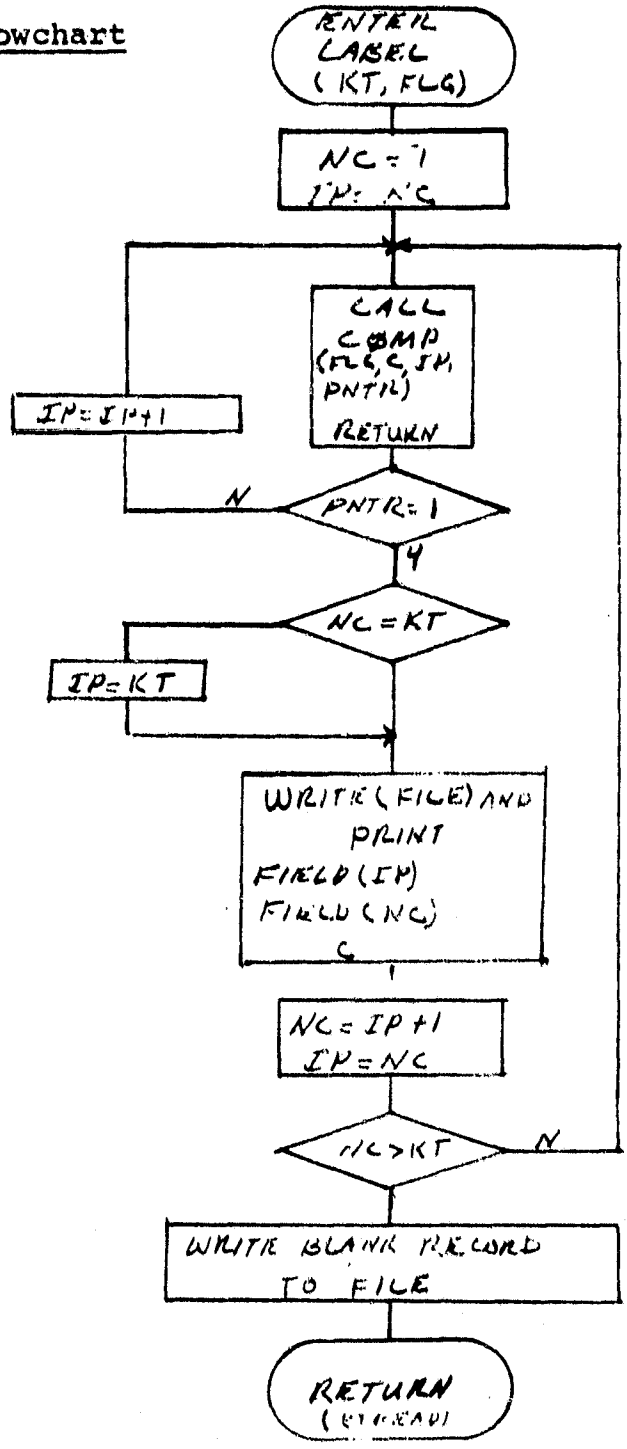
3.2.3.3b Listing

```
      SUBRØUTINE FSØRT(KT)
         IMPLICIT INTEGER (A-Z)
         CØMMØN /BT/NRDR, NPRT,FIELD(500),FLDV(500)
      KTM1=KT-1
      DØ 10 J=1,KT
      DØ 20 I=1,KTM1
      IF(FIELD(I).GT.FIELD(I+1)) GØ TØ 20
      DUMP=FIELD(I)
      FIELD(I)=FIELD(I+1)
      FIELD(I+1)=DUMP
      DUMP=FLDV(I)
      FLDV(I)=FLDV(I+1)
      FLDV(I+1)=DUMP
20    CØNTINUE
10    CØNTINUE
      RETURN
      END
```

3.2.3.4 Subroutine LABEL

Subroutine LABEL writes to file and prints out the field codes numbers, including operations flags, that are assigned by its subordinate routine COMP.

3.2.3.4a Flowchart



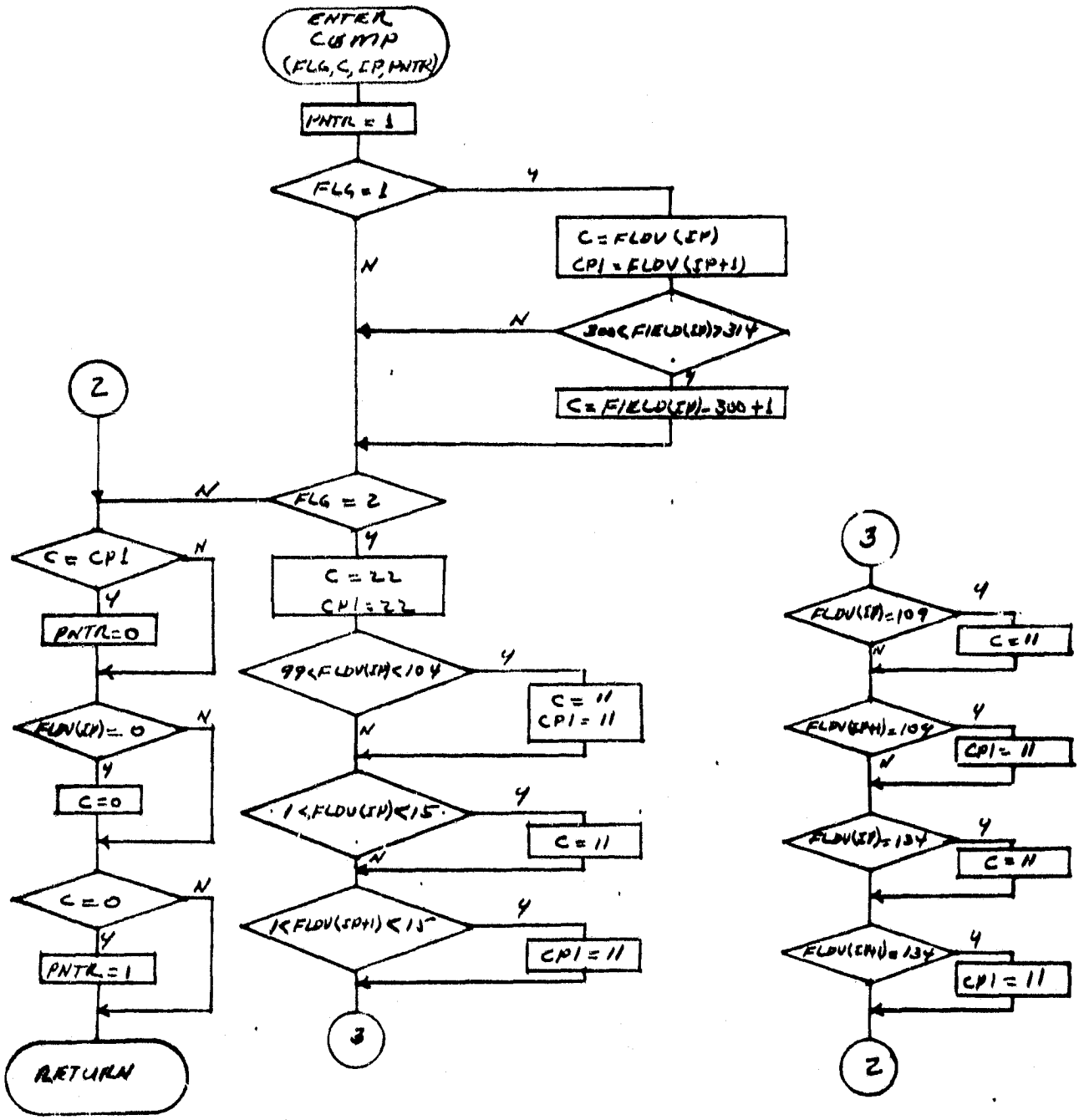
3.2.3.4b Listing

```
      SUBROUTINE LABEL(KT,FLG)
      IMPLICIT INTEGER (A-Z)
      COMMON /RT/NRDR, NPRT,FIELD(500),FLDV(500)
      WRITE(NPRT,507)
507   FORMAT (///,10X,'FIELD TO CODE TRANSFORMATION',//,7X,'FIELD',2X,
     *'TO',2X,'FIELD',6X,'CODE')
      NC=1
500   CONTINUE
      IP=NC
501   CONTINUE
      CALL COMP(FLG,C,IP,PNTR)
      IF(PNTR.EQ.1) GO TO 502
      IP=IP+1
      GO TO 501
502   CONTINUE
      IF(NC.EQ.KT) IP=KT
      WRITE(NRDR,503) FIELD(IP),FIELD(NC),C
      WRITE (NPRT,506) FIELD(IP),FIELD(NC),C
506   FORMAT (10X,3I10)
503   FORMAT(3I5)
      NC=IP+1
      IP=NC
      IF(NC.GT.KT) WRITE(NRDR,504)
504   FORMAT(/)
      IF(NC.GT.KT) GO TO 505
      GO TO 501
505   CONTINUE
      RETURN
      END
```

3.2.3.5 Subroutine COMP

Subroutine COMP is called by subroutine LABEL to assign appropriate codes to the fields.

3.2.3.5a Flowchart



3.2.3.5b Listing

```
      SUBROUTINE COMP  FLG,C,IP,PNTR)
      IMPLICIT INTEGER (A-Z)
      COMMON/BT/NRDR,NPRT,FIELD(500),FLDV(500)
      PNTR =1
      IF(FLG.EQ.1) C=FLDV(IP)
      IF(FLG.EQ.1) CP1 = FLDV(IP+1)
      IF(FLG.EQ.1.AND. FIELD(IP).GE.300.AND.FIELD(IP).LE.314)
     * C=FIELD(IP)-300+1
      IF(FLG.EQ.2) C=22
      IF(FLG.EQ.2) CP1=22
      IF(FLG.EQ.2.AND.FLDV(IP).LE.104.AND.
     * FLDV(IP).GE.99) C=11
      IF(FLG.EQ.2.AND.                FLDV (IP+1).LE.104.AND.
     * FLDV(IP+1).GE.99)  CP1=11
      IF(FLG.EQ.2.AND.FLDV(IP).GE.124.AND.FLDV(IP).LE.129) C=11
      IF(FLG.EQ.2.AND.FLDV(IP+1).GE.124. AND.FLDV(IP+1).LE.129)
     * CP1=11
      IF(FLG.EQ.2.AND.FLDV(IP).GE.1.AND.FLDV(IP).LE.15) C=11
      IF(FLG.EQ.2.AND.                FLDV(IP+1).GE.1.AND.FLDV(IP+1).
     * LE.15) CP1=11
      IF(FLG.EQ.2.AND.FLDV(IP).EQ.109) C=11
      IF(FLG.EQ.2.AND.FLDV(IP+1).EQ.109) CP1=11
      IF(FLG.EQ.2.AND.FLDV(IP).EQ.134) C=11
      IF(FLG.EQ.2.AND.FLDV(IP+1).EQ.134) CP1=11
      IF(C.EQ.CP1) PNTR=0
      IF(FLDV(IP).EQ.0) C=0
      IF(C.EQ.0) PNTR=1
      RETURN
      END
```

3.2.4 UTILITY UNIT SGMAP

3.2.4.1 Linkage

This, a routine for optional printout of the ground truth data entered for processing, calls the special subroutines DTMAP SPMAP and GTMAP directly, the special subroutine CRØPP through GTMAP, and standard system routines. It also employs, in the subroutine GTMAP, the special function MPCD (CRØP).

3.2.4.2 Interface

SGMAP interfaces directly only with its subroutines DTMAP and GTMAP. Communication with those subroutines is through calling arguments and the common blocks MAP and CH.

3.2.4.3 Input

All inputs are derived from the disk file output from Phase 1.

3.2.4.4 Output

Hard copy printout map of the ground truth data to be input to the Phase 2 processors.

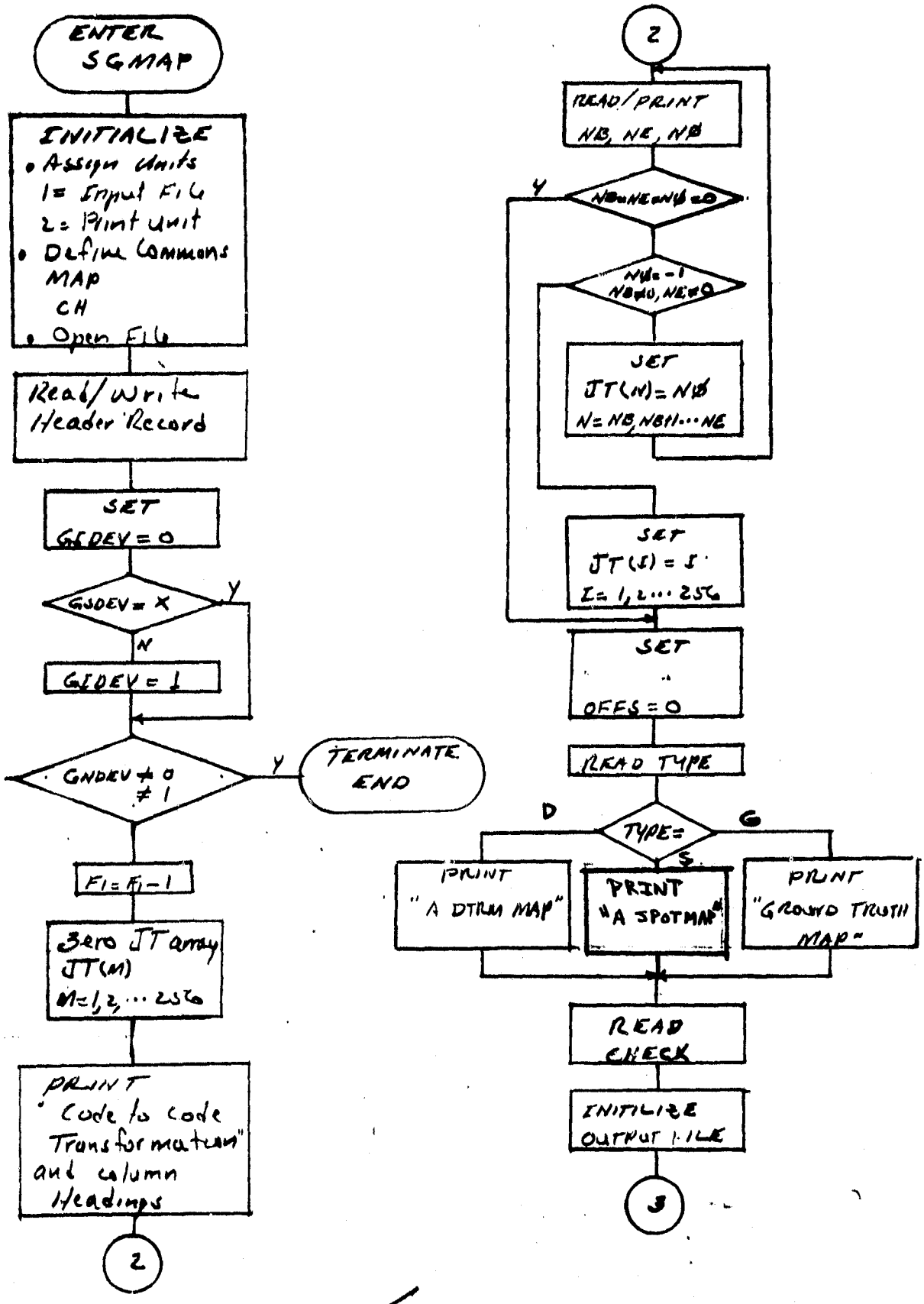
3.2.4.5 Storage

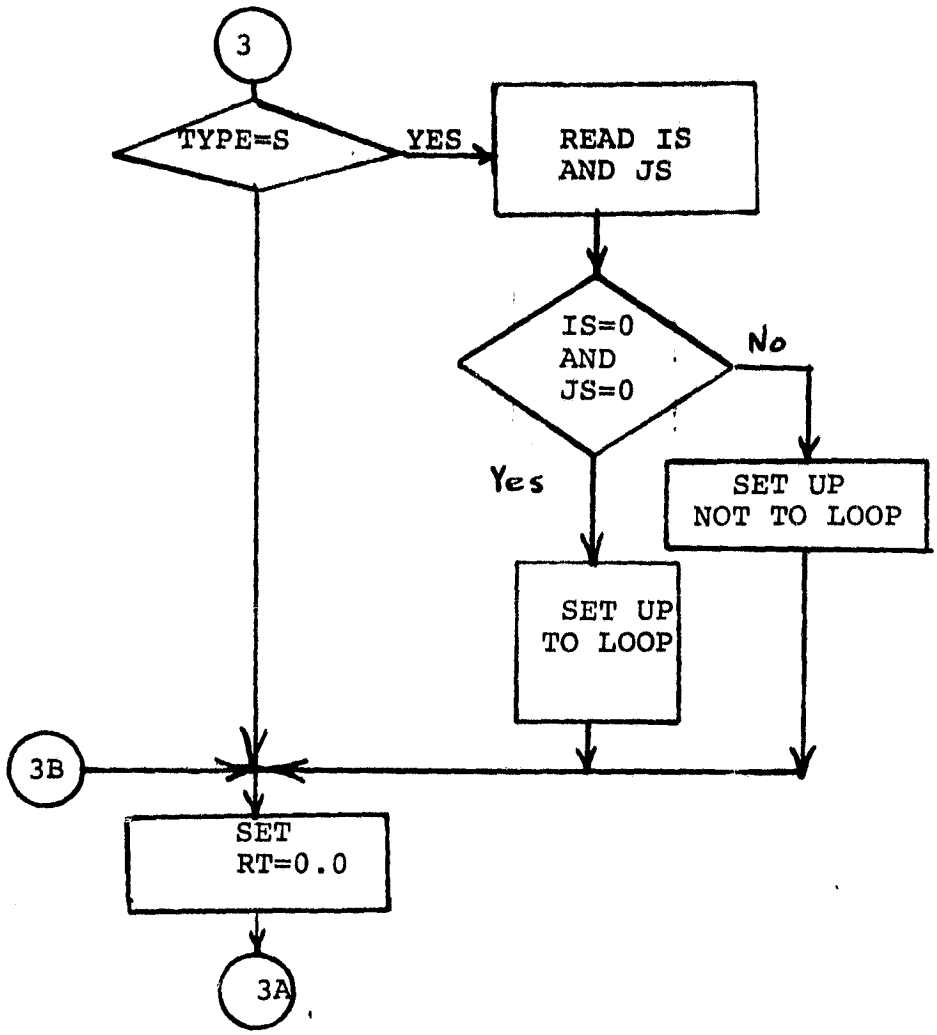
TBD

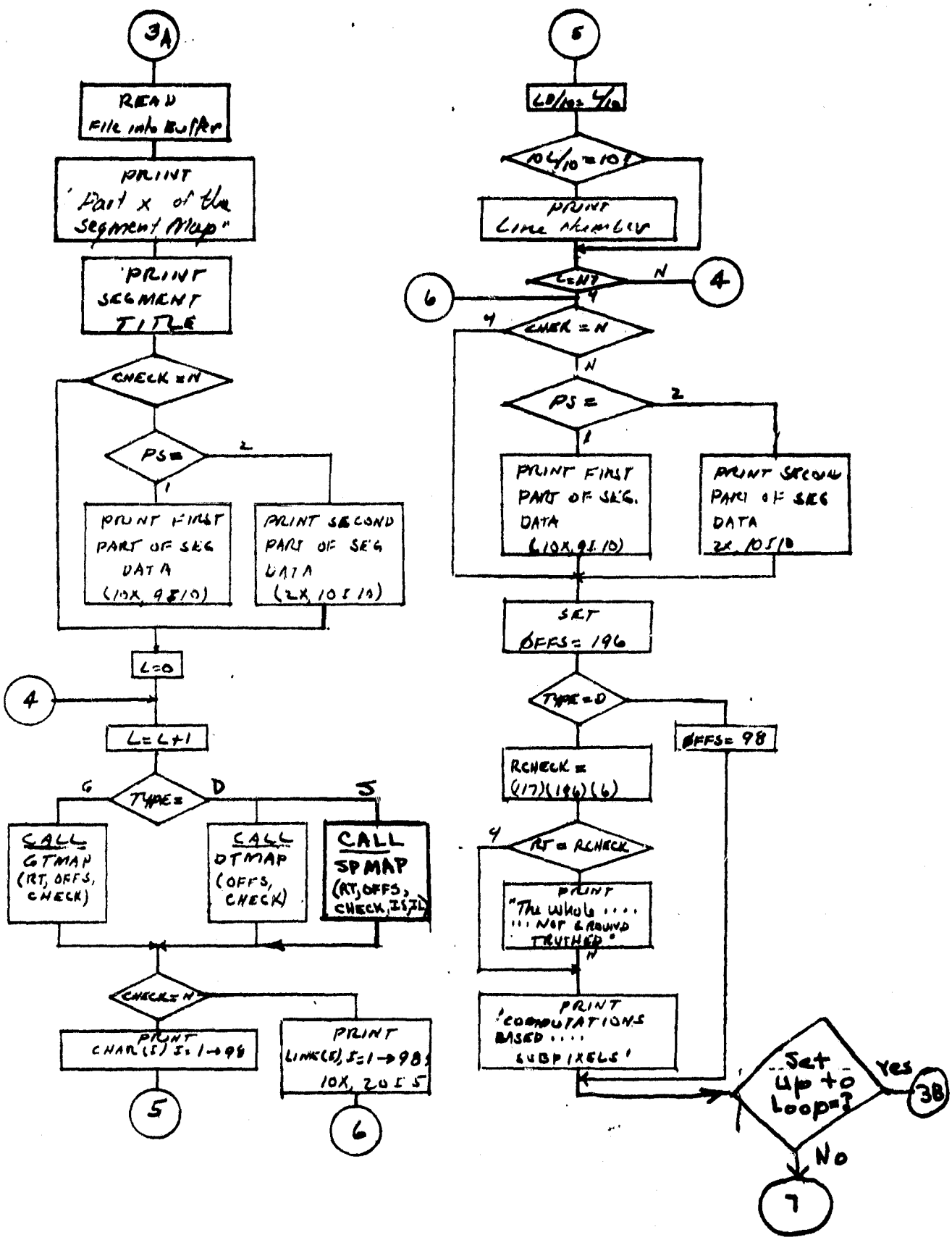
3.2.4.6 Description

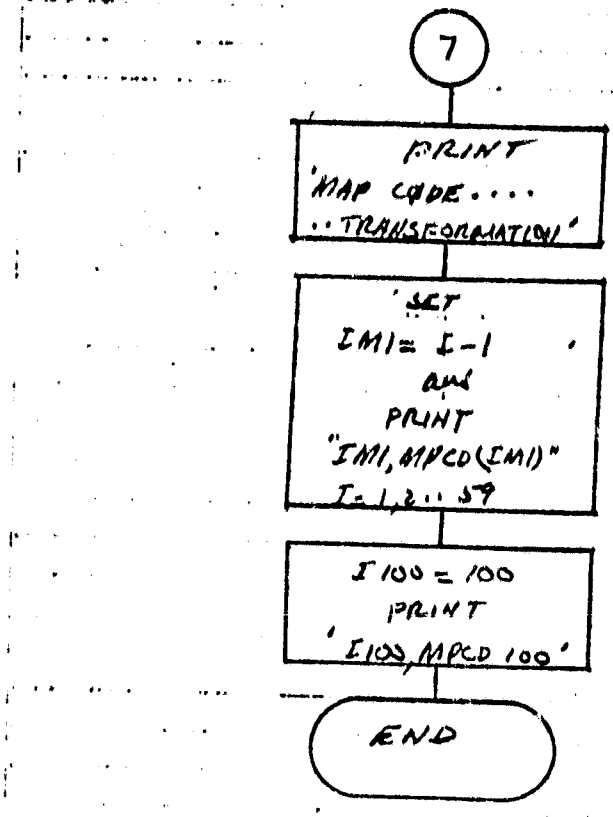
SGMAP extracts ground truth data from the Ground truth file previously made by a Phase 1 operation. It then manipulates and organizes those data into an output format for convenient user evaluation.

3.2.4.7 Flowchart




3
TYPE=S
YES
READ IS
AND JS
IS=0
AND
JS=0
No
Yes
SET UP
NOT TO LOOP
SET UP
TO LOOP
3B
SET
RT=0.0
3A


3A
READ File into Buffer
PRINT "Part x of the segment Map"
PRINT SEGMENT TITLE
CHECK = N
PS =
1
2
PRINT FIRST PART OF SEG DATA (10X, 9I5/)
PRINT SECOND PART OF SEG DATA (2X, 10I5/)
L=0
4
L=L+1
TYPE =
G
D
S
CALL GTMAP (RT, OFFS, CHECK)
CALL DTMAP (OFFS, CHECK)
CALL SPMAP (RT, OFFS, CHECK, I5,I)
CHECK = N
PRINT CHAR(I) I=1→98
PRINT LINE(I), I=1→98 10X, 20I5
5
6
6
LO/10 = 4/10
104/10 = 10?
PRINT Line Number
L=N?
N
4
4
CHEK = N
N
PS =
1
2
PRINT FIRST PART OF SEG. DATA (10X, 9I5/)
PRINT SECOND PART OF SEG DATA 2X, 10I5/)
SET OFFS = 196
TYPE = D
OFFS = 98
RCHECK = (17)(196)(6)
4
RT = RCHECK
PRINT "The Whole .... Not Found TRY HER"
N
PRINT 'COORDINATES BASED .... SUBPIXELS'
Set up to Loop=?
Yes
3B
No
7



3.2.4.8 Listing

```
      IMPLICIT INTEGER (A-Q),(S-Z)
      BYTE BUF
      BYTE MPCD
      BYTE           T(8),D(9)
      BYTE CHAR
      COMMON/MAP/BUF(3060), MT(6), LINE(98), JT(256)
      COMMON/CH/CHAR(98)
      EQUIVALENCE (S,BUF(67))
      CALL TIME(T)
      CALL DATE(D)
      NRDR=1
      NPRT=6
      WRITE(NPRT,703) D,T
      OPEN (UNIT=NRDR,NAME='SGMAP,DAT',TYPE='OLD',
     1ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
  703 FORMAT(1H1,'  JOB INITIATED ON  ',9A1,'  AT  ',8A1,//,10X,
     1'PROGRAM  SGMAP,FTN')
      READ(NRDR,704) GSDEV,GNDEV,F1
  704 FORMAT(A1,1X,2I2)
      WRITE(NPRT,705) GSDEV,GNDEV,F1
  705 FORMAT(//,10X,'          INPUT TAPE',///,10X,A1,'T',10X,'DEVICE NO.=',
     115,10X,'FILE NO.=',I5)
      GIDEV=0
      IF(GSDEV.EQ.'X') GIDEV=1
      IF(GNDEV.NE.0.AND.GNDEV.NE.1) GO TO 400
      F1=F1-1
      CALL CLOSE (NRDR)
      OPEN (UNIT=NRDR,NAME='MAP,DAT',TYPE='OLD',
     1ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
      DO 20 M=1,256
      JT(M)=0
   20 CONTINUE
      WRITE (NPRT,905)
  905 FORMAT(//,10X,'CODE TO CODE TRANSFORMATION',//,8X,'BEGIN',7X,
     1'END' ,7X,'CODE',7X,'SYMBOL')
  12  CONTINUE
      READ(NRDR,118) NB,NE,N0
  118 FORMAT(3I5)
      WRITE(NPRT,117) NB,NE,N0,MPCD(N0)
  117 FORMAT(1H ,3I10,9X,A1)
      IF((NB.EQ.0).AND.(NE.EQ.0).AND.(N0.EQ.0)) GO TO 122
      IF((NB.EQ.0).AND.(NE.EQ.0).AND.(N0.EQ.-1)) GO TO 124
      DO 119 N=NB,NE
      JT(N)=N0
  119 CONTINUE
      GO TO 121
  124 CONTINUE
      DO 123 I=1,256
  123 JT(I)=I
  122 CONTINUE
      OFFS=0
      CALL  TINIT(3,GIDEV,GNDEV)
      CALL  TATCH(3)
      READ (NRDR, 555) TYPE
  555 FORMAT (A1)
      IF (TYPE.EQ. 'G') WRITE (NPRT,556)
  556 FORMAT (1HO,10X, 'A GROUND TRUTH MAP')
      IF (TYPE.EQ.'D') WRITE (NPRT,557)
  557 FORMAT (1HO,10X,'A DTRM MAP')
      IF(TYPE.EQ.'S')  WRITE(NPRT,558)
  558 FORMAT(1HO,10X,'A SPOT MAP')
      READ(NRDR,555) CHECK
```

```
      IF(TYPE,EQ,'S')  READ(NRDR,559)  IS,JS
  559 FORMAT(2I5)
      IF(TYPE,EQ,'S',AND,IS,GT,79 )  IS=79
      IF(TYPE,EQ,'S',AND,JS,GT,99 )  JS=99
      IF(TYPE,EQ,'S') IS1=IS
      IF(TYPE,EQ,'S') IS2=IS
      IF(TYPE,EQ,'S') JS1=JS
      IF(TYPE,EQ,'S') JS2=JS
      IF(TYPE,NE,'S') JS2=1
      IF(TYPE,NE,'S') JS1=1
      IF(TYPE,NE,'S') IS1=1
      IF(TYPE,NE,'S') IS2=1
      IF(TYPE,EQ,'S',AND,IS,EQ,0,AND,JS,EQ,0)  IS1=1
      IF(TYPE,EQ,'S',AND,IS,EQ,0,AND,JS,EQ,0)  IS2=79
      IF(TYPE,EQ,'S',AND,IS,EQ,0,AND,JS,EQ,0)  JS1=1
      IF(TYPE,EQ,'S',AND,IS,EQ,0,AND,JS,EQ,0)  JS2=99
      DO 334 II=IS1,IS2,39
      DO 334 JJ=JS1,JS2,98
      IS=II
      JS=JJ
      RT=0,0
      IF(TYPE,EQ,'S')  OFFS=2*(JS-1)
      DO 333 PS=1,2
      CALL  TRWD(3)
      CALL  TWAIT(3)
      CALL  TFILE(3,F1)
      CALL  TWAIT(3)
      CALL  TREAD(3,BUF,1530)
      CALL  TWAIT(3)
      WRITE(NPRT,501) PS
  501 FORMAT(1H1,10X,'PART ',I1,' OF THE SEGMENT MAP')
      CALL SWAB(S)
      WRITE(NPRT,306) S,(BUF(I8),I8=61,65)
  306 FORMAT('  SITE=  ',I15,5X,'DAY=',I5,5X,'MON=',I5,5X,'YEAR=',I5)
      IF(TYPE,EQ,'S') WRITE(NPRT,560)  IS,JS
  560 FORMAT(1H+,90X,'IS=',I5,5X,'JS=',I5)
      IF (CHECK,EQ,'N') GO TO 113
      IF(PS,EQ,1) WRITE(NPRT,505)(M,M=1,9)
  505 FORMAT(1H , 10X,9I10)
      IF(PS,EQ,2) WRITE (NPRT,506) (M=1,M=1,10)
  506 FORMAT(1H ,2X, 10I10)
  113 CONTINUE
      DO 1 L=1,117
      IF(TYPE,EQ,'G') CALL GTMAP (RT,OFFS,CHECK)
      IF(TYPE,EQ,'D') CALL DTMAP(OFFS,CHECK)
      IF(TYPE,EQ,'S')  CALL SPMAP(RT,OFFS,CHECK,IS,L)
      IF(CHECK,EQ,'N') GO TO 111
      WRITE(NPRT,500) (CHAR(I),I=1,98)
  500 FORMAT(1H ,10X,98A1)
      LD10=L/10
      IF(LD10*10,EQ,L) WRITE (NPRT,507) LD10,LD10
  507 FORMAT (1H+,5X,I3,2X,98X,I4)
      GO TO 1
  111 CONTINUE
      WRITE(NPRT,515) (LINE(I), I = 1,98)
  515 FORMAT(1H ,10X,2015)
      WRITE(NPRT,615) L
  615 FORMAT(1H+,I5)
      WRITE(NPRT,516)
  516 FORMAT(1H0)
    1 CONTINUE
      IF(CHECK,EQ,'N') GO TO 112
      IF(PS,EQ,1) WRITE(NPRT,505)(M,M=1,9)
      IF(PS,EQ,2) WRITE (NPRT,506) (M=1,M=1,10)
  112 CONTINUE
      OFFS=196
```

```
      IF (TYPE.EQ.'D')  OFFS = 98
      IF(TYPE.EQ.'S')  OFFS=2*(JS-1)  *98
  333 CONTINUE
      IF (TYPE.EQ.'D')  GO TO 456
      RCHECK=117.0*196.0*6.0
      IF(TYPE.EQ.'S') GO TO 250
      IF(RT.NE.RCHECK) WRITE(NPRT,223)
  223 FORMAT(//,10X,'THE WHOLE SEGMENT WAS NOT GROUND TRUTHED')
  250 CONTINUE
      WRITE(NPRT,222) HT
  222 FORMAT(//,10X,'COMPUTATIONS BASED ON ',F10.2,' SUBPIXELS',/)
  456     CONTINUE
  334 CONTINUE
      WRITE(NPRT,601)
  601 FORMAT(//,10X,'MAP CODE TO SYMBOL TRANSFORMATION')
      DO 602 I=1,59
      IM1=I-1
      WRITE(NPRT,610) IM1,MPCD(IM1)
  602 CONTINUE
  610 FORMAT(1H ,10X,I3,5X,A1)
      I100=100
      WRITE(NPRT,610) I100,MPCD(I100)
  400 CONTINUE
      WRITE(NPRT,999)
      CALL DATE(D)
      CALL TIME(T)
      WRITE(NPRT,104) D,T
      WRITE(NPRT,104) D,T
  104 FORMAT('  JOB COMPLETED ON  ',9A1,'  AT  ',8A1)
  999 FORMAT(///)
      STOP
      END
```

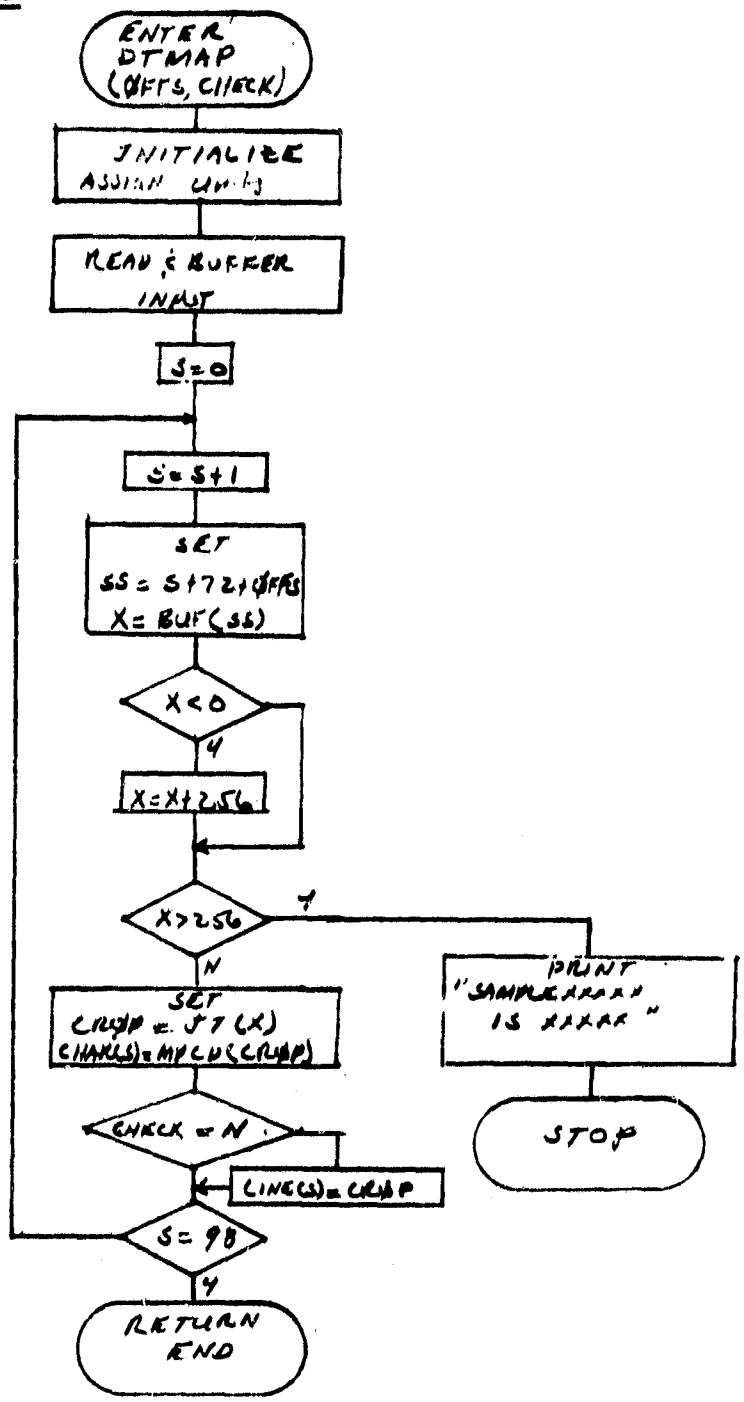
```
SGMAP,LP1/SH=SGMAP
[100,4]SWAB
[100,4]LECTAP
[1,1]F4POTS/LB
/
ASG=SY11
ASG=LP16
MAXBUF=3060
PRI=50
//
```

3.2.5 SGMAP SUBROUTINES

Three special subroutines DTMAP, SPMAP and GTMAP are called directly by SGMAP and a third subroutine CRØPP is called indirectly through GTMAP.

3.2.5.1 Subroutine DTMAP

3.2.5.1a Flowchart

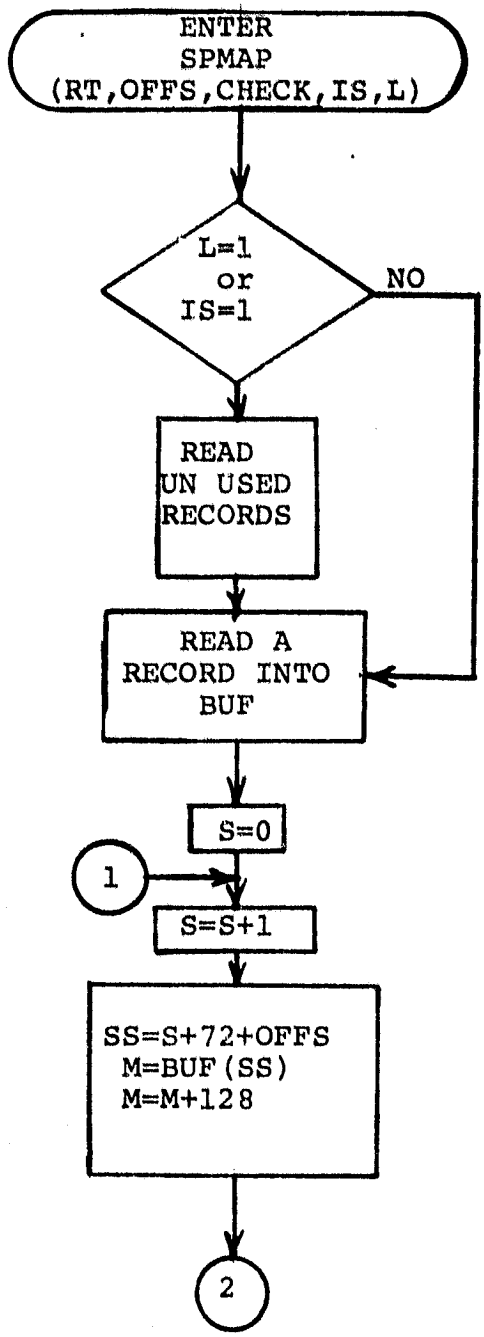


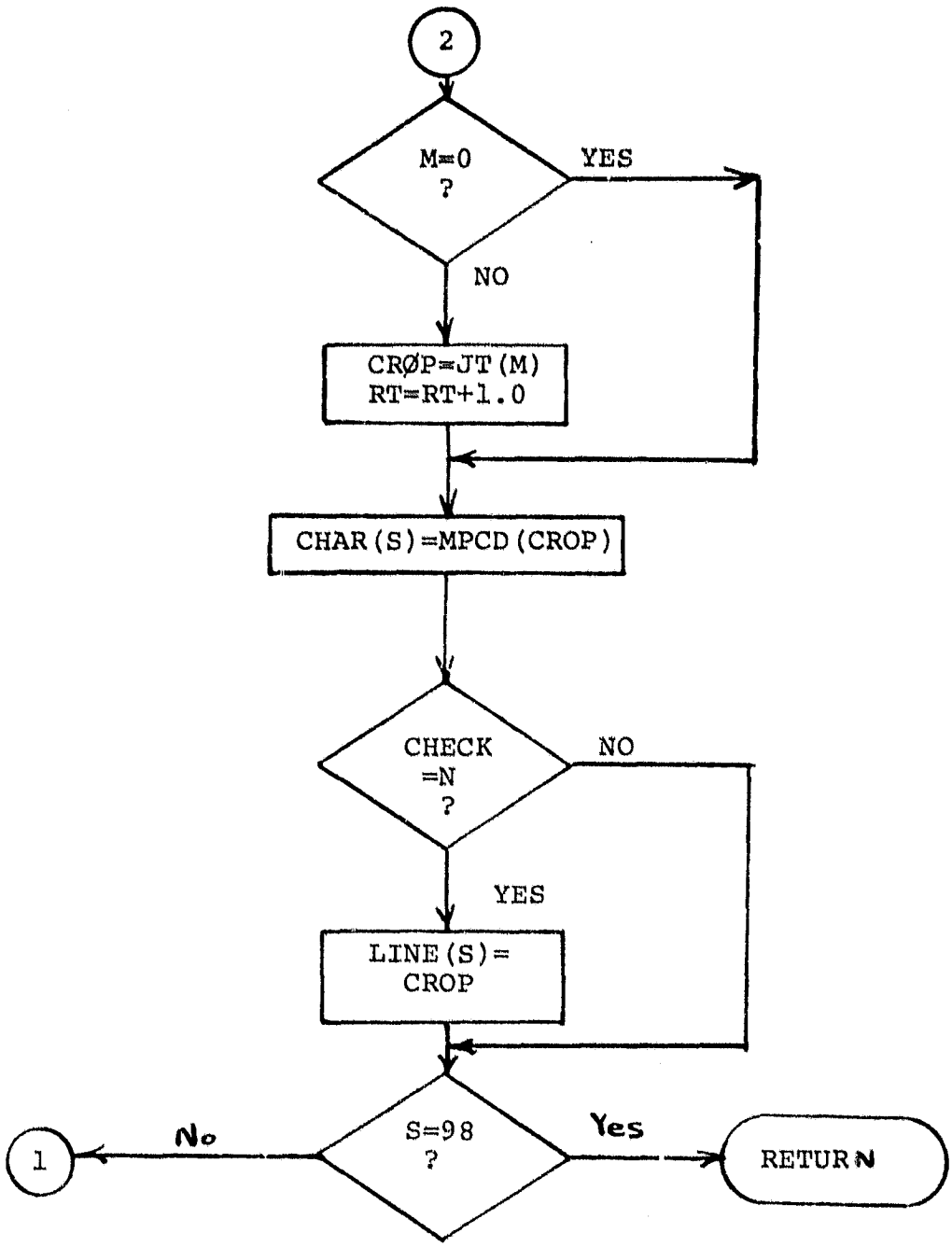
3.2.5.1b Listing

```
      SUBROUTINE DTMAP (OFFS,CHECK)
      IMPLICIT INTEGER (A-Q),(S-Z)
      BYTE MPCD
      BYTE CHAR
      BYTE BUF
      COMMON/MAP/BUF(3060), MT(6), LINE(98),JT(256)
      COMMON/CH/CHAR(98)
      CALL  TREAD (3,BUF,180)
      CALL  TWAIT(3)
      NPRT=6
C
      DO 13 S=1,98
      SS = S + 72 + OFFS
      X = BUF(SS)
      IF (X.LE.0) X = X + 256
      IF(X.GT.256) WRITE (NPRT,500) S,X
  500 FORMAT(1HO,10X,'SAMPLE ',I5,'  IS',I3)
      IF(X.GT.256) STOP
      CROP = JT(X)
      CHAR(S) = MPCD (CROP)
      IF(CHECK.EQ.'N') LINE(S)=CROP
   13 CONTINUE
C
C
      RETURN
      END
```

3.2.5.4 Subroutine SPMAP

3.2.5.4 Flowchart



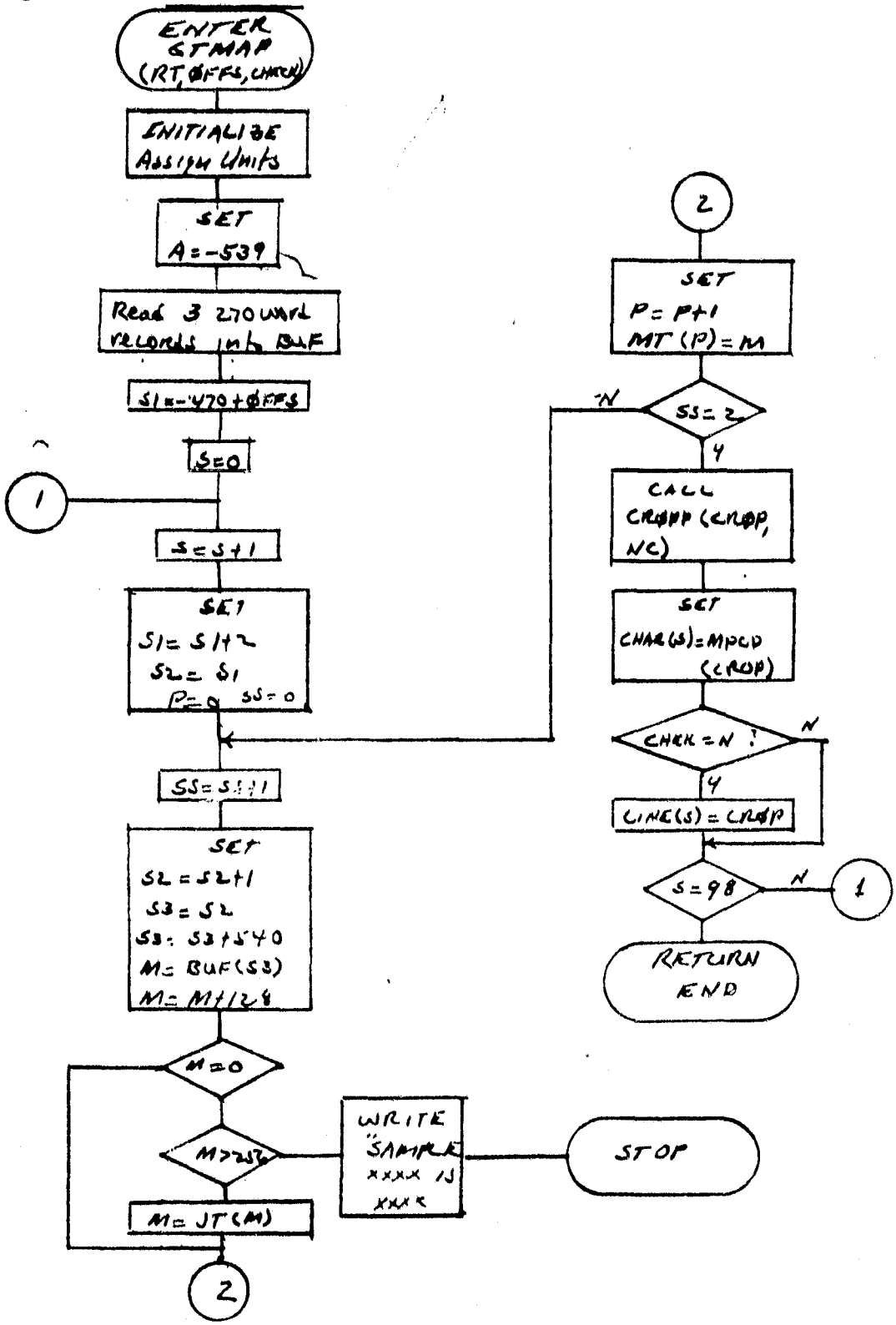

2
M=0
?
YES
NO
CRØP=JT(M)
RT=RT+1.0
CHAR(S)=MPCD(CROP)
CHECK
=N
?
NO
YES
LINE(S)=
CROP
S=98
?
No
Yes
1
RETURN

3.2.5.4b

```
      SUBROUTINE SPMAP(RT,OFFS,CHECK,IS,L)
      IMPLICIT INTEGER (A-O),(S-Z)
        BYTE MPCD
        BYTE CHAR
      BYTE BUF
        COMMON/MAP/BUF(3060), MT(6), LINE(98),CT(256)
        COMMON/CH/CHAR(98)
        NPRT=6
      IF(L.NE.1.OR.IS.EQ.1) GO TO 101
      ISM1=IS-1
      ISS=3*ISM1
      DO 100 LL=1,ISS
      CALL TREAD(3,BUF,270)
      CALL TWAIT(3)
  100 CONTINUE
  101 CONTINUE
      CALL TREAD(3,BUF,270)
      CALL TWAIT(3)
      DO 10 S=1,98
      SS=S*72+OFFS
      M=BUF(SS)
      M=M*128
      IF(M.EQ.0) GO TO 20
      CROP=CT(M)
      RT=RT+1.0
   20 CONTINUE
      CHAR(S)=MPCD(CROP)
      IF(CHECK.EQ.'N') LINE(S)=CROP
   10 CONTINUE
        RETURN
        END
```

3.2.5.2 Subroutine GTMAP(CRT, OFFS, CHECK)

3.2.5.2a Flowchart

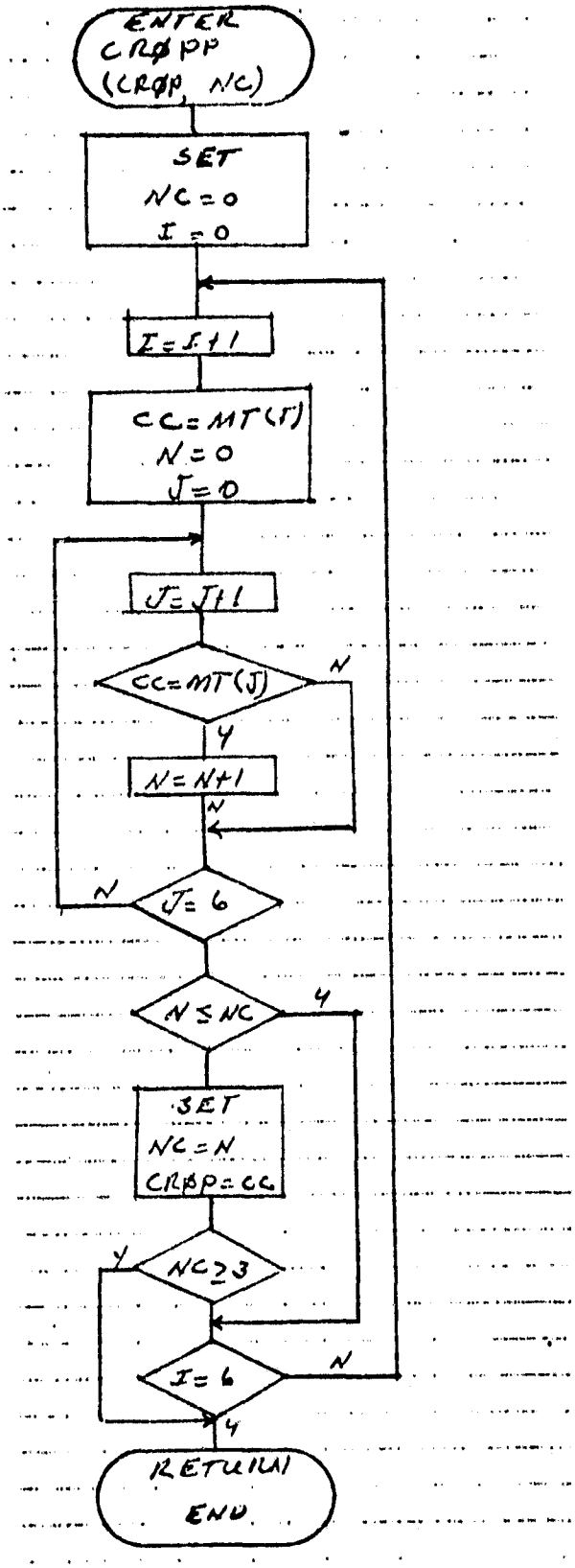


3.2.5.2b Listing

```
      SUBROUTINE GTMAP (RT,OFFS,CHECK)
      IMPLICIT INTEGER (A-Q),(S-Z)
        BYTE MPCD
        BYTE CHAR
      BYTE BUF
        COMMON/MAP/BUF(3060), MT(6), LINE(98),JT(256)
        COMMON/CH/CHAR(98)
        NPRT=6
      A=-539
      DO 2 SL=1,3
      A=A+540
      CALL  TREAD(3,BUF(A),270)
      CALL  TWAIT(3)
    2 CONTINUE
      S1=-470+OFFS
      DO 3 S=1,98
      S1=S1+2
      S2=S1
      P=0
      DO 4 SS=1,2
      S2=S2+1
      S3=S2
      S3=S3+540
      M=BUF(S3)
      M=M+128
      IF(M.NE.0) RT=RT+1.0
      IF(M.EQ.0) GO TO 700
        IF(M.GT.256) WRITE (NPRT,500) S,M
  500   FORMAT(1HO,10X,'SAMPLE ',I5,'  IS',I5)
        IF(M.GT.256) STOP
      M=JT(M)
  700 CONTINUE
      P=P+1
      MT(P)=M
    5 CONTINUE
    4 CONTINUE
      CALL CROPP(CROP,   NC)
      CHAR(S)=MPCD(CROP)
      IF(CHECK.EQ.'N') LINE(S)=CROP
    3 CONTINUE
        RETURN
        END
```

3.2.5.3 Subroutine CRØPP

3.2.5.3a Flowchart



3.2.5.3b Listing

```
      SUBROUTINE CROPP (CROP,   NC)
      IMPLICIT INTEGER (A-O),(S-Z)
      BYTE BUF
         COMMON/MAP/BUF(3060), MT(6), LINE(98), JT(256)
      NC=0
      DO 10 I=1,6
      CC=MT(I)
      N=0
      DO 20 J=1,6
      IF(CC.EQ.MT(J))N=N+1
   20 CONTINUE
      IF(N.LE.NC) GO TO 10
      NC=N
      CROP=CC
      IF(NC.GE.3) RETURN
   10 CONTINUE
      RETURN
      END
```

3.2.6 FIRST UNIT OF FIRST MODULE (PHASE 1)

3.2.6.1 Linkage

This, the executive routine of the first unit of the first functional module, calls standard system utility routines and the following special subroutines: S-01, S-12, S-23, S-34, S-45, S-55 and S-56.

3.2.6.2 Interface

Phase 1 constructs and loads input data into the common block

3.2.6.1 Linkage

Phase 1 constructs and loads input data into the common block "stuff" (see listing) for communication with its subroutines. It constructs the "intercept" and "Header" files which are output to and stored on system files for use as input for the subsequent exection of the companion software module (Phase 2).

3.2.6.3 Input

Magnetic tape output from BTREAD, a translated Bendix-100 output data tape (see appendix A).

3.2.6.4 Output

"Intercept" file is output to system files for storage. Print option is provided for listing of buffered inputs.

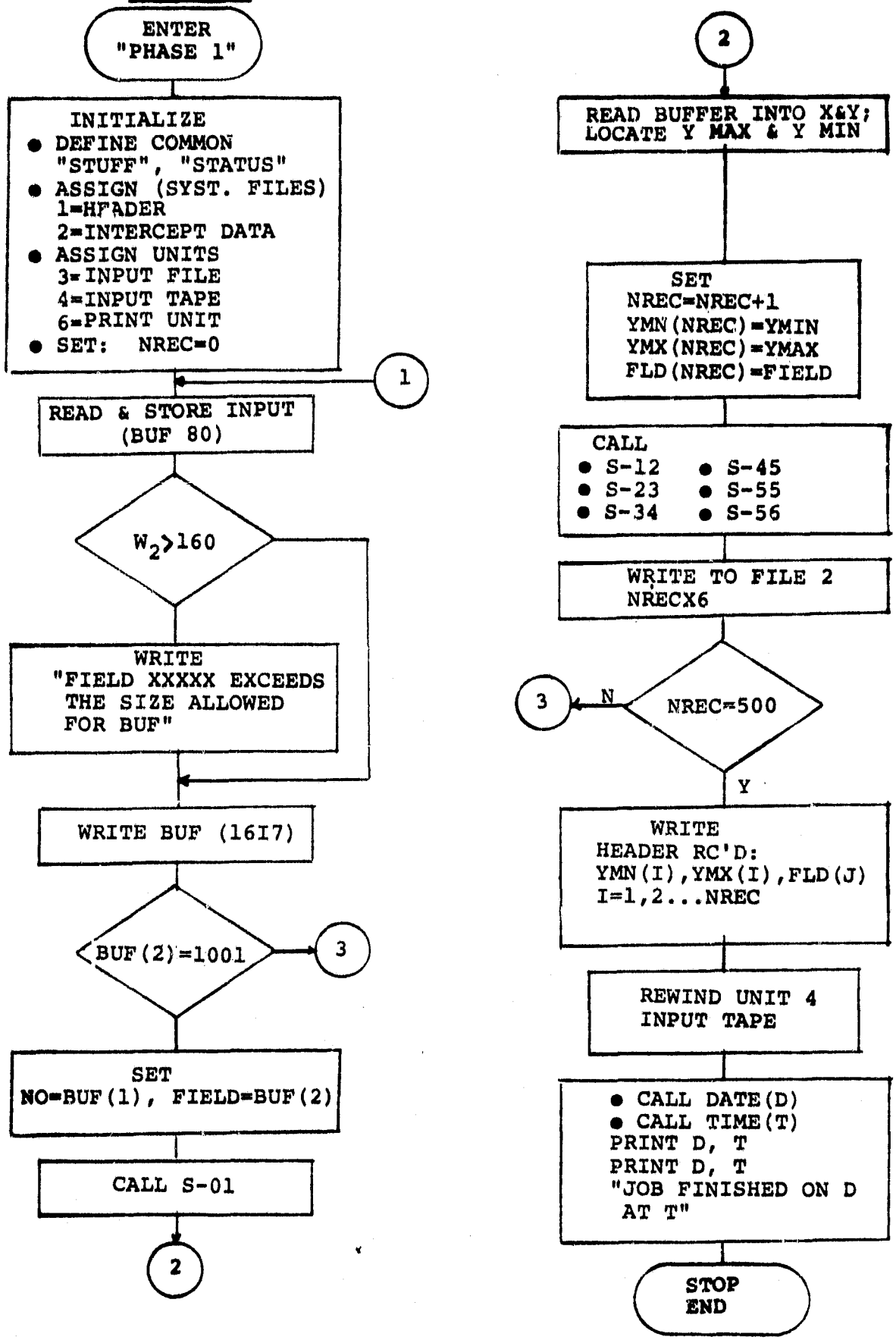
3.2.6.5 Storage

TBD

3.2.6.6 Description

"Phase 1" is the executive routine for the first unit of the first functional software module of the system. As such, it managers unit input/output and establishes the subroutine calling sequence for construction of the intercept file from the input field vertices data.

3.2.6.7 Flowchart



3.2.6.8 Listing

```
C  PROCESSES BENDIX TAPE TO PRODUCE INTERCEPT FILES
      IMPLICIT INTEGER (A-O),(S-Z)
      COMMON /STUFF/X6(512),NPRT,BUF(80),N0,X1(50),Y1(50),N1,YMIN,YMAX,
     *X2(55),Y2(55),N2,X3(70),Y3(70),N3,X4(512),Y4(512),N4,X5(200,11),J
      COMMON /STATUS/W1,W2
      DIMENSION YMN(500),YMX(500),FLD(500)
      BYTE D(9),T(8)
      BYTE SDEV
      CALL TIME(T)
      CALL DATE(D)
      NRDR=3
         OPEN (UNIT=NRDR,NAME='PHASE1,DAT',TYPE='OLD',
     1ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
      NPRT=6
      WRITE(NPRT,703) D,T
  703 FORMAT(1H1,'  JOB INITIATED ON  ',9A1,'  AT  ',8A1,//,10X,
     1'PROGRAM  PHASE1,FTN')
      READ (NRDR,301) SDEV,NDEV,FILE
  301 FORMAT(A1,1X,2I2)
      WRITE (NPRT,302) SDEV,NDEV,FILE
  302 FORMAT(//,10X,A1,'T',10X,'DEVICE NO.=',I5,10X,'FILE NO.=',I5)
         CALL CLOSE(NRDR)
         OPEN (UNIT=NRDR,NAME='LABEL1,DAT',TYPE='OLD',
     1ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
      READ(NRDR,305) S,DAY,MON,YR
  305 FORMAT(4I5)
         WRITE(NPRT,555) S,DAY,MON,YR
  555    FORMAT(//,10X,'SEG. NO.=',I5,5X,'DAY=',I5,5X,'MONTH=',
     *I5,5X,'YEAR=',I5)
         OPEN(UNIT=1,NAME='HEAD,DAT',TYPE='OLD',
     1ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
      CALL ASSIGN(2,'SY:INTCPT,DAT')
      DEFINE FILE 2 (500,512,U,AV)
      IDEV=0
      IF(SDEV.EQ.88)   IDEV=1
      IF(NDEV.NE.0.AND.NDEV.NE.1) GO TO 5
      CALL  TINIT(4,IDEV,NDEV)
      CALL  TATCH(4)
      CALL  TRWD(4)
      CALL  TWAIT(4)
      CALL  TFILE(4,(FILE-1))
      CALL  TWAIT(4)
      NREC=0
    1 CONTINUE
      CALL  TREAD(4,BUF,80)
      CALL  TWAIT(4)
      IF(W2.GT.160) WRITE(NPRT,101) BUF(2)
  101 FORMAT('  FIELD  ',I5,'  EXCEEDS THE SIZE ALLOWED FOR BUF')
```

```
D       WRITE(NPRT,102) BUF
  102 FORMAT(1H0,16I7)
      IF(BUF(2),EQ,1001) GO TO 2
      NO=BUF(1)
      FIELD=BUF(2)
      CALL S01
      NREC=NREC+1
      YMN(NREC)=YMIN
      YMX(NREC)=YMAX
      FLD(NREC)=FIELD.
      CALL S12
      CALL S23
      CALL S45
      CALL S55
      CALL S56
      WRITE(2'NREC) X6
      IF(NREC,EQ,500) GO TO 2
      GO TO 1
    2 CONTINUE
      WRITE(1,201) NREC
  201 FORMAT(1I5)
      WRITE(1,202) (YMN(I),I=1,NREC)
      WRITE(1,202) (YMX(I),I=1,NREC)
      WRITE(1,202) (FLD(I),I=1,NREC)
  202 FORMAT(50I5)
    5 CONTINUE
      CALL  TRWD(4)
      CALL  TWAIT(4)
      CALL DATE(D)
      CALL TIME(T)
      WRITE(NPRT,104) D,T
      WRITE(NPRT,104) D,T
  104 FORMAT('  JOB FINISHED ON  ',9A1,'     AT    ',8A1)
        CALL CLOSE(6)
      STOP
      END
```

3.2.7 FIRST UNIT SUBROUTINE (S-01)

3.2.7.1 Linkage

Called by "Phase 1" with simple return.

3.2.7.2 Interface

Communicates with calling routine through the common block "stuff" (see listing).

3.2.7.3 Input

All inputs are derived from the common block "stuff".

3.2.7.4 Output

Option is provided for trouble shooting printout (listing of working buffer contents).

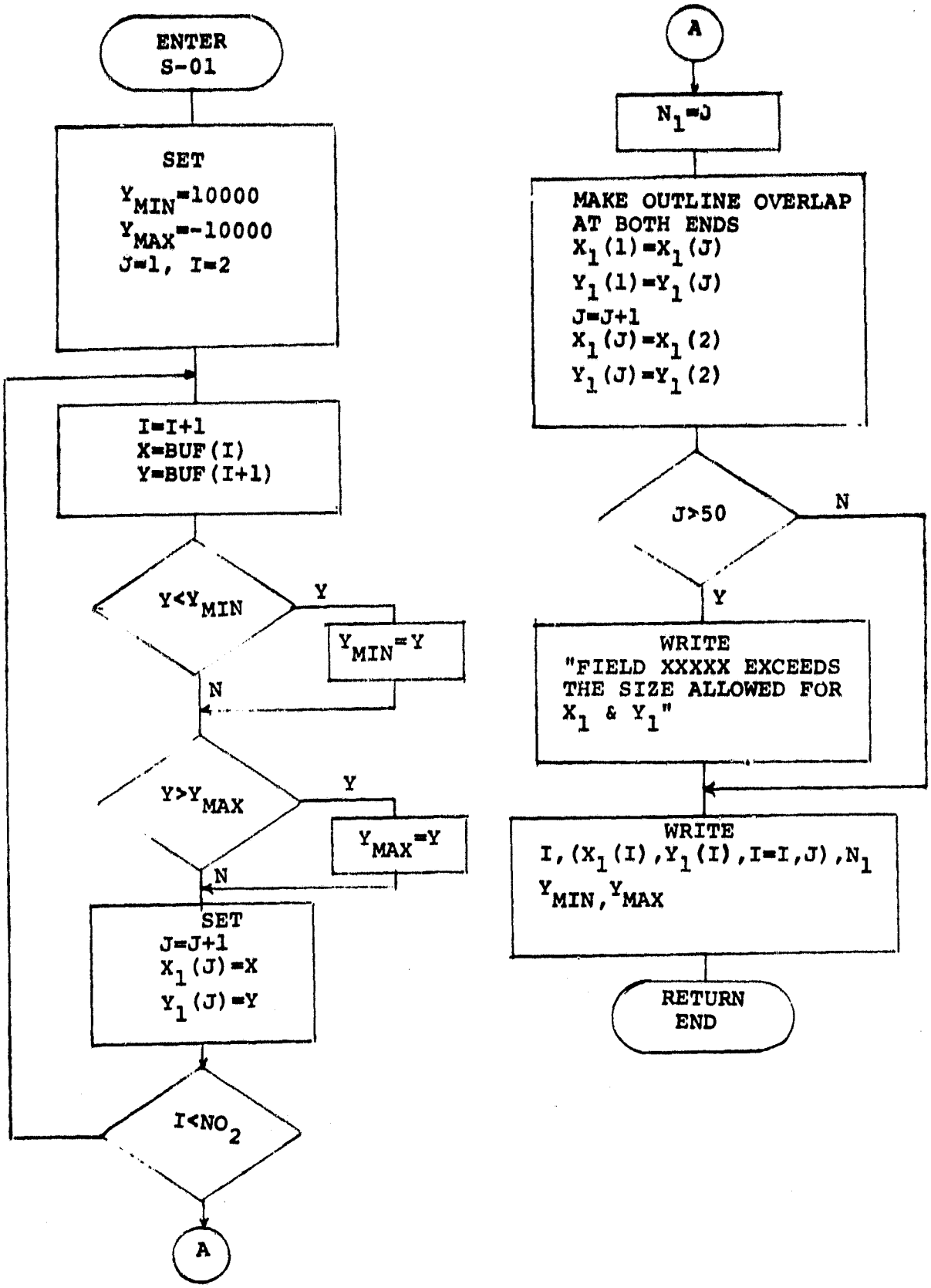
3.2.7.5 Storage

TBD

3.2.7.6 Description

S-01 is called by Phase 1 to load field vertices into X and Y arrays; and finds the maximum and minimum Y coordinates for each field.

3.2.7.7 Flowchart



3.2.7.8 Listing

```
HFORTRAN IV-PLUS V02-04                    17127137    22-APR-77          PAGE 1
SC1,FTN          /TR:BLOCKS/WR
0001                SUBROUTINE SC1
                C  READS BUFFER N1 & Y1 AND FINDS YMIN & YMAX
0002                IMPLICIT INTEGER (A-C),(S-Y)
0003                COMMON /STUFF/X6(512),NPRT,BUF(80),N0,X1(50),Y1(50),N1,YMIN,YMAX
                   *X2(55),Y2(55),N2,X3(70),Y3(70),N3,X4(512),Y4(512),N4,X5(200,11),
0004                YMIN=100000
0005                YMAX=-10000
0006                J=1
0007                DO 1 I=3,N0,2
0008                X=BUF(I)
0009                Y=BUF(I+1)
0010                IF(Y.LT.YMIN) YMIN=Y
0011                IF(Y.GT.YMAX) YMAX=Y
0012                J=J+1
0013                X1(J)=X
0014                Y1(J)=Y
0015              1 CONTINUE
0016                N1=J
                C  MAKE OUTLINE OVERLAP AT BOTH ENDS
0017                X1(1)=X1(J)
0018                Y1(1)=Y1(J)
0019                J=J+1
0020                X1(J)=Y1(2)
0021                Y1(J)=Y1(2)
0022                IF(J.GT.50) WRITE(NPRT,102) BUF(2)
0023            102 FORMAT(' FIELD ',I5,' EXCEEDS THE SIZE ALLOWED FOR X1 & Y1')
                D   WRITE(NPRT,101) (I,X1(I),Y1(I),I=1,J),N1,YMIN,YMAX
0024            101 FORMAT(1H ,3I10)
0025                RETURN
0026                END
```

3.2.8 FIRST UNIT SUBROUTINE (S-12)

3.2.8.1 Linkage

Called by "Phase 1" with simple return.

3.2.8.2 Interface

Communicates with calling routine through the common block "stuff" (see listing).

3.2.8.3 Input

All inputs are derived from the common block "stuff".

3.2.8.4 Output

Option is provided for trouble shooting printout (listing of buffer contents)

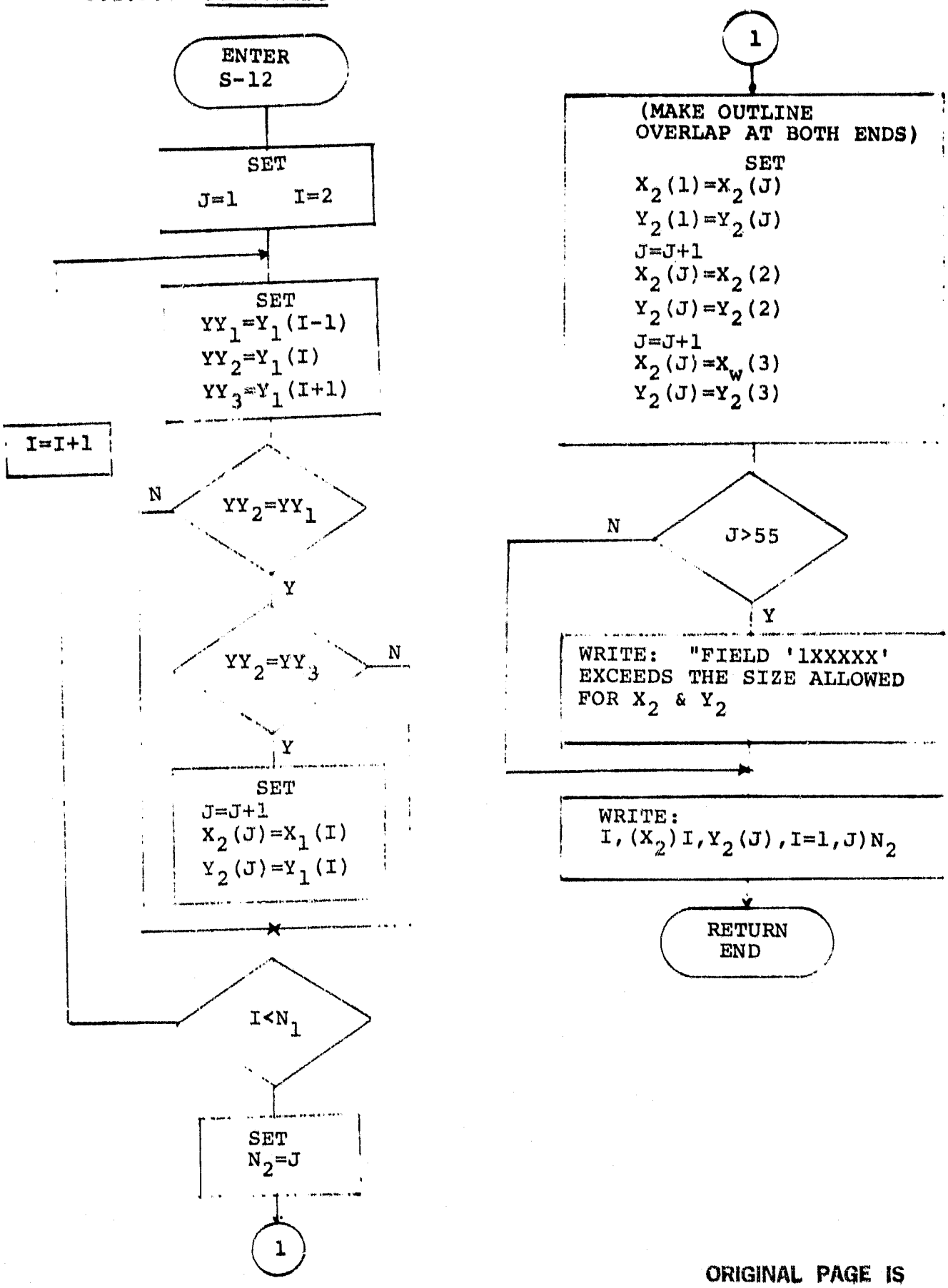
3.2.8.5 Storage

TBD

3.2.8.6 Description

S-12 is the second subroutine called by "Phase 1". It removes redundant points from the field vertices data and the returns control to "Phase 1".

3.2.8.7 Flowchart



3.2.8.8 Listing

```
HFORTRAN IV-PLUS V02-04          17127150   22-APR-77          PAGE
S12.FTN          /TR:RL:C4S/WR
0001              SUBROUTINE S12
                C  REMOVES REDUNDANT POINTS FROM X1 & Y1 SO THAT
                C  THERE ARE AT MOST TWO (CONTIGUOUS) POINTS ON A LINE
0002              IMPLICIT INTEGER (I-O),(S-Z)
0003              COMMON /STUFF/X6(512),LPRT,BUF(80),N0,X1(50),Y1(50),N1,YMIN,YMA
                 *X2(55),Y2(55),N2,X3(70),Y3(70),N3,X4(512),Y4(512),N4,X5(200,11)
0004              J=1
0005              DO 1 I=2,N1
0006              YY1=Y1(I-1)
0007              YY2=Y1(I)
0008              YY3=Y1(I+1)
0009              IF(YY2.NE.YY1) GO TO 2
0010              IF(YY2.NE.YY3) GO TO 2
                C  POINT I IS A REDUNDANT POINT
0011              GO TO 1
0012            2 CONTINUE
                C  POINT I IS NOT A REDUNDANT POINT
0013              J=J+1
0014              X2(J)=X1(I)
0015              Y2(J)=Y1(I)
0016            1 CONTINUE
0017              N2=J
                C  MAKE OUTLINE OVERLAP AT BOTH ENDS
0018              X2(1)=X2(J)
0019              Y2(1)=Y2(J)
0020              J=J+1
0021              X2(J)=Y2(2)
0022              Y2(J)=Y2(2)
0023              J=J+1
0024              X2(J)=Y2(3)
0025              Y2(J)=Y2(3)
0026              IF(J.GT.55) WRITE(LPRT,102) BUF(2)
0027          102 FORMAT(' FIELD ',I15,' EXCEEDS THE SIZE ALLOWED FOR X2 & Y2')
                D     WRITE(LPRT,101) (I,X2(I),Y2(I),I=1,J),N2
0028          101 FORMAT(1H ,3I10)
0029              RETURN
0030              END
```

3.2.9 FIRST UNIT SUBROUTINE (S-23)

3.2.9.1 Linkage

Called by "Phase 1" with simple return.

3.2.9.2 Interface

Communicates with calling routine through the common block "stuff" (see listing).

3.2.9.3 Input

All inputs are derived from the common block "stuff".

3.2.9.4 Output

Option is provided for trouble shooting printout (listing of working buffer contents).

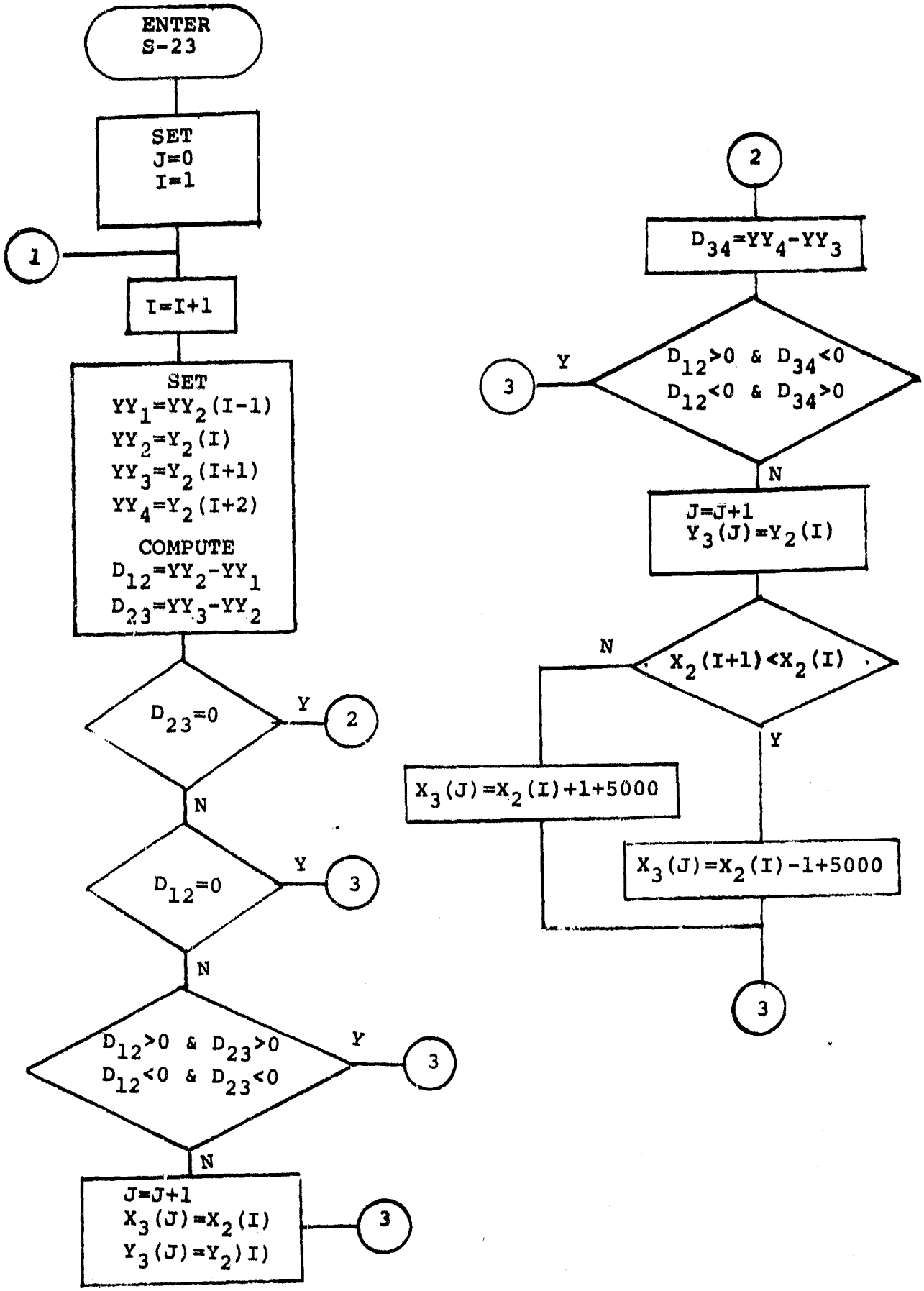
3.2.9.5 Storage

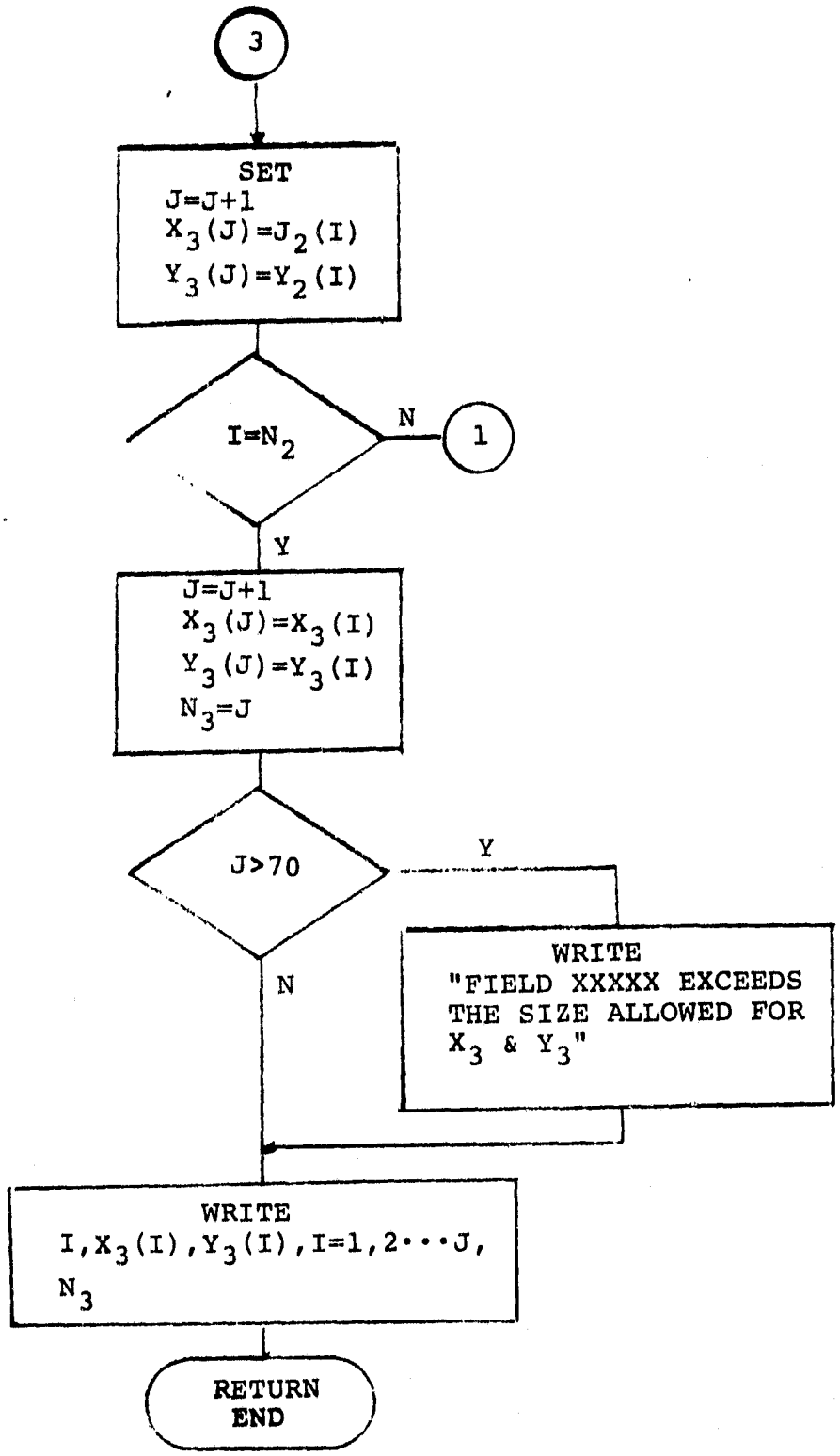
TBD

3.2.9.6 Description

S-23 examines the field boundary points (vertices), selects out those which are critical points (points of inflection, maxima or minima) then inserts redundant points to properly account for such critical points along field boundaries in the following field dot mapping process.

3.2.9.7 Flowchart




3
SET
J=J+1
X3(J)=J2(I)
Y3(J)=Y2(I)
I=N2
N
1
Y
J=J+1
X3(J)=X3(I)
Y3(J)=Y3(I)
N3=J
J>70
Y
N
WRITE
"FIELD XXXXX EXCEEDS
THE SIZE ALLOWED FOR
X3 & Y3"
WRITE
I,X3(I),Y3(I),I=1,2•••J,
N3
RETURN
END

```
HP FORTRAN IV-PLUS /02-04          17120105    22-APR-77          PAGE 1
S23,FTN          /TRACE/CHKS/KR
0001          SUBROUTINE S23
              C  INSERTS REDUNDANT POINTS AT MAXIMA, MINIMA, AND INFLECTIONS
0002          IMPLICIT INTEGER (A-Q),(S-Z)
0003          COMMON /STUFF/X5(512),APRT,BLF(80),NC,X1(50),Y1(50),N1,YMIN,YMAX,
             *X2(55),Y2(55),N2,X3(70),Y3(70),N3,X4(512),Y4(512),N4,X5(200,11),
0004          J=0
0005          DO 1 I=2,N2
0006          YY1=Y2(I-1)
0007          YY2=Y2(I)
0008          YY3=Y2(I+1)
0009          YY4=Y2(I+2)
0010          D12=YY2-YY1
0011          D23=YY3-YY2
              C  CHECK TO SEE IF POINTS I & (I+1) ARE POINTS OF INFLECTION
0012          IF(D23,EQ,0) GO TO 2
              C  CHECK TO SEE IF POINTS I & (I-1) ARE A TWO-POINT MAXIMUM OR MINIMUM
0013          IF(D12,EQ,0) GO TO 3
              C  CHECK TO SEE IF POINTS I & (I-1) ARE A ONE-POINT MAXIMUM OR MINIMUM
0014          IF((D12,GT,0),AND,(D23,GT,0)) GO TO 3
0015          IF((D12,LT,0),AND,(D23,LT,0)) GO TO 3
              C  POINT I IS A MAXIMUM OR MINIMUM
0016          J=J+1
0017          X3(J)=X2(I)
0018          Y3(J)=Y2(I)
0019          GO TO 1
0020        2 CONTINUE
              C  POINTS I & (I+1) MIGHT BE POINTS OF INFLECTION
0021          D34=YY4-YY3
0022          IF((D12,GT,0),AND,(D34,LT,0)) GO TO 3
0023          IF((D12,LT,0),AND,(D34,GT,0)) GO TO 3
              C  POINTS I & (I+1) ARE POINTS OF INFLECTION
0024          J=J+1
0025          Y3(J)=Y2(I)
0026          IF(X2(I+1),LT,X2(I)) GO TO 4
              C  PUT A REDUNDANT POINT TO RIGHT OF POINT I AND TAG BY ADDING 5000
0027          X3(J)=X2(I)+1+5000
0028          GO TO 3
0029        4 CONTINUE
              C  PUT A REDUNDANT POINT TO LEFT  OF POINT I AND TAG BY ADDING 5000
0030          X3(J)=X2(I)-1+5000
0031        3 CONTINUE
0032          J=J+1
0033          X3(J)=X2(I)
0034          Y3(J)=Y2(I)
0035        1 CONTINUE
0036          J=J+1
0037          X3(J)=X3(1)
0038          Y3(J)=Y3(1)
0039          N3=J
0040          IF(J,GT,70) WRITE(APRT,102) BLF(2)
0041      102 FORMAT(' FIELD ',I10,' EXCEEDS THE SIZE ALLOWED FOR X3 & Y3')
              C     WRITE(APRT,101) (I,X3(I),Y3(I),I=1,J),N3
0042      101 FORMAT(1H ,5I10)
0043          RETURN
0044          END
```

32.10 FIRST UNIT SUBROUTINE (S-34)

3.2.10.1 Linkage

Called by "Phase 1" with simple return.

3.2.10.2 Interface

Communicates with calling routine through the common block "stuff" (see listing).

3.2.10.3 Input

All inputs are derived from the common block "stuff".

3.2.10.4 Output

Option is provided for trouble-shooting printout (listing of working buffer contents).

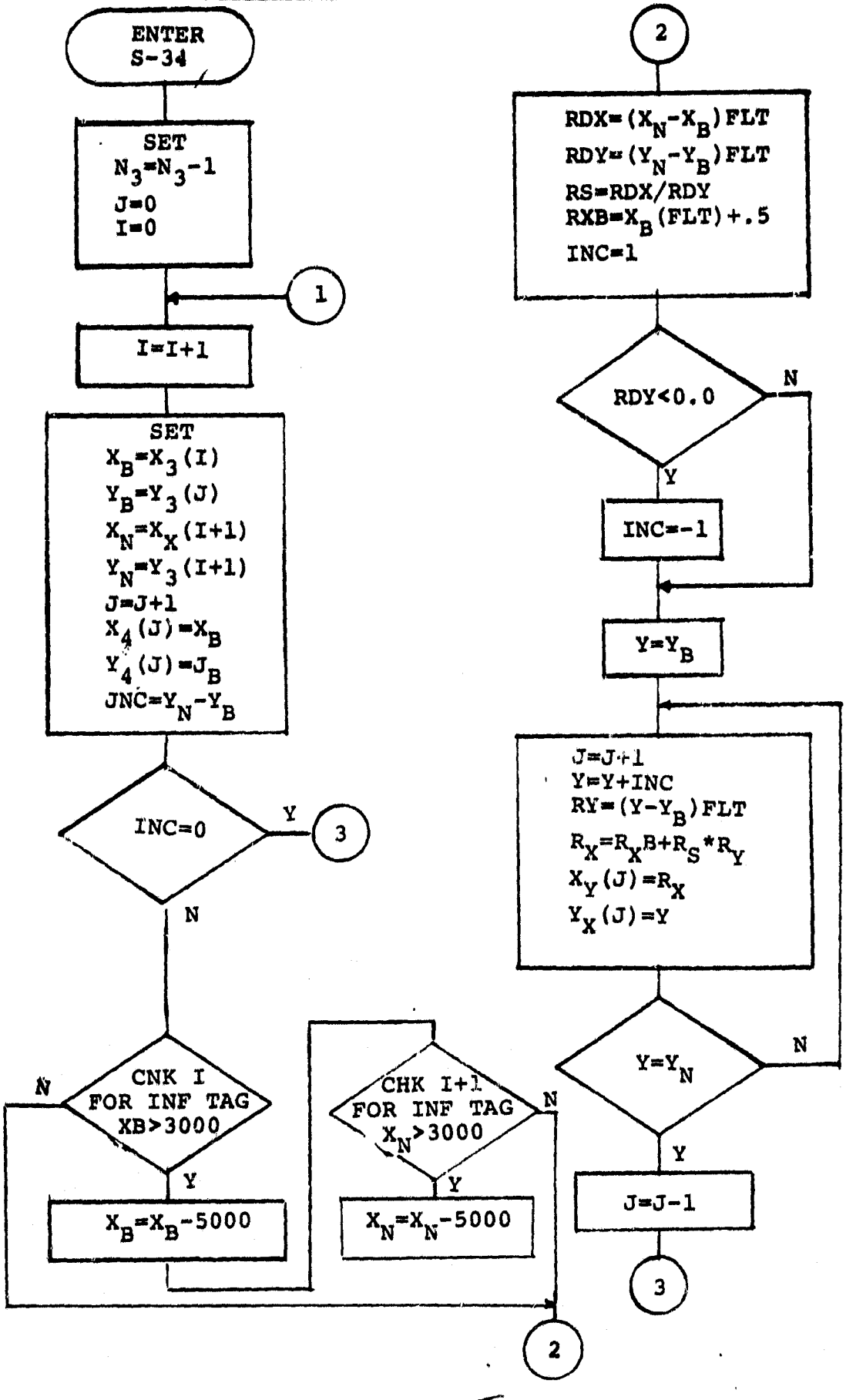
3.2.10.5 Storage

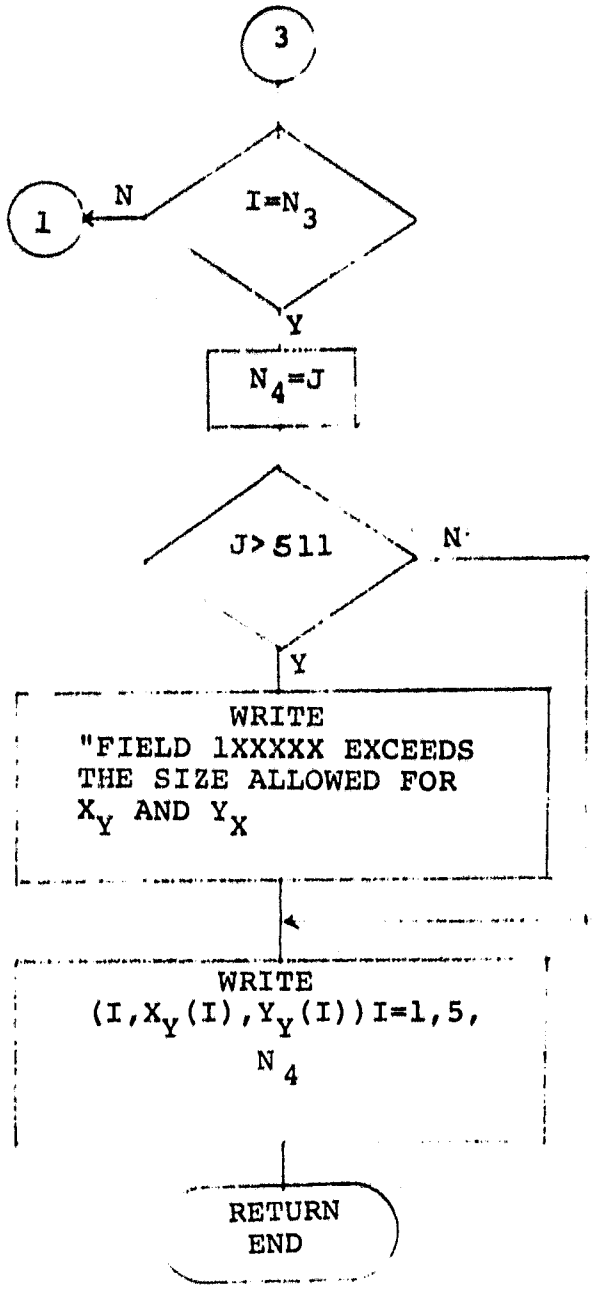
TBD

3.2.10.6 Description

S-34 defines field boundary intersections with scene lines which are intermediate to those of the input specified field boundary vertices.

3.2.10.7 Flowchart




3
I=N3
N
1
Y
N4=J
J>511
N
Y
WRITE
"FIELD 1XXXXX EXCEEDS
THE SIZE ALLOWED FOR
XY AND YX
WRITE
(I,XY(I),YY(I))I=1,5,
N4
RETURN
END

3.2.10.8 Listing

```
FORTRAN IV-PLUS V02-04                          17:28:23    22-APR-77          PAGE
S34.FTN       /TR:BLOCKS/WR
      SUBROUTINE S34
C  FILLS IN MISSING LINES
      IMPLICIT INTEGER (A-G),(S-Z)
      COMMON /STUFF/X6(512),NPRT,BUF(80),N0,X1(50),Y1(50),N1,YMIN,YMA
     *X2(55),Y2(55),N2,X3(70),Y3(70),N3,X4(512),Y4(512),N4,X5(200,11)
      N3=N3-1
      J=0
      DO 1 I=1,N3
      XB=X3(I)
      YB=Y3(I)
      XN=X3(I+1)
      YN=Y3(I+1)
      J=J+1
      X4(J)=XB
      Y4(J)=YB
      INC=YN-YB
      IF(INC.EQ.0) GO TO 1
C  MISSING LINES MUST BE FILLED IN
C  CHECK TO SEE IF EITHER I OR (I+1) HAS BEEN TAGGED AS POINT OF INFL
      IF(XB.GT.3000) XB=XB-5000
      IF(XN.GT.3000) XN=XN-5000
      RDX=FLOAT(XN-XB)
      RDY=FLOAT(YN-YB)
      RS=RDX/RDY
      RXB=FLOAT(XB)+0.5
      INC=1
      IF(RDY.LT.0.0) INC=-1
      Y=YB
C  FILL IN LINES BETWEEN (BUT NOT INCLUDING) POINTS I AND (I+1)
    3 CONTINUE
      J=J+1
      Y=Y+INC
      RY=FLOAT(Y-YB)
      RX=RXB+RS*RY
      X4(J)=RX
      Y4(J)=Y
      IF(Y.NE.YN) GO TO 3
      J=J-1
    1 CONTINUE
      N4=J
      IF(J.GT.511) WRITE(NPRT,102) BUF(2)
  102 FORMAT('  FIELD ',I15,' EXCEEDS THE SIZE ALLOWED FOR X4 & Y4')
D     WRITE(NPRT,101) (I,X4(I),Y4(I),I=1,J),N4
  101 FORMAT(1H ,3I10)
      RETURN
      END
```

3.2.11 FIRST UNIT SUBROUTINE (S-45)

3.2.11.1 Linkage

Called by "Phase 1" with simple return.

3.2.11.2 Interface

Communicates with calling routine through the common block "stuff" (see listing).

3.2.11.3 Input

All inputs are derived from the common block "stuff".

3.2.11.4 Output

Option is provided for trouble-shooting printout (listing of working buffer contents).

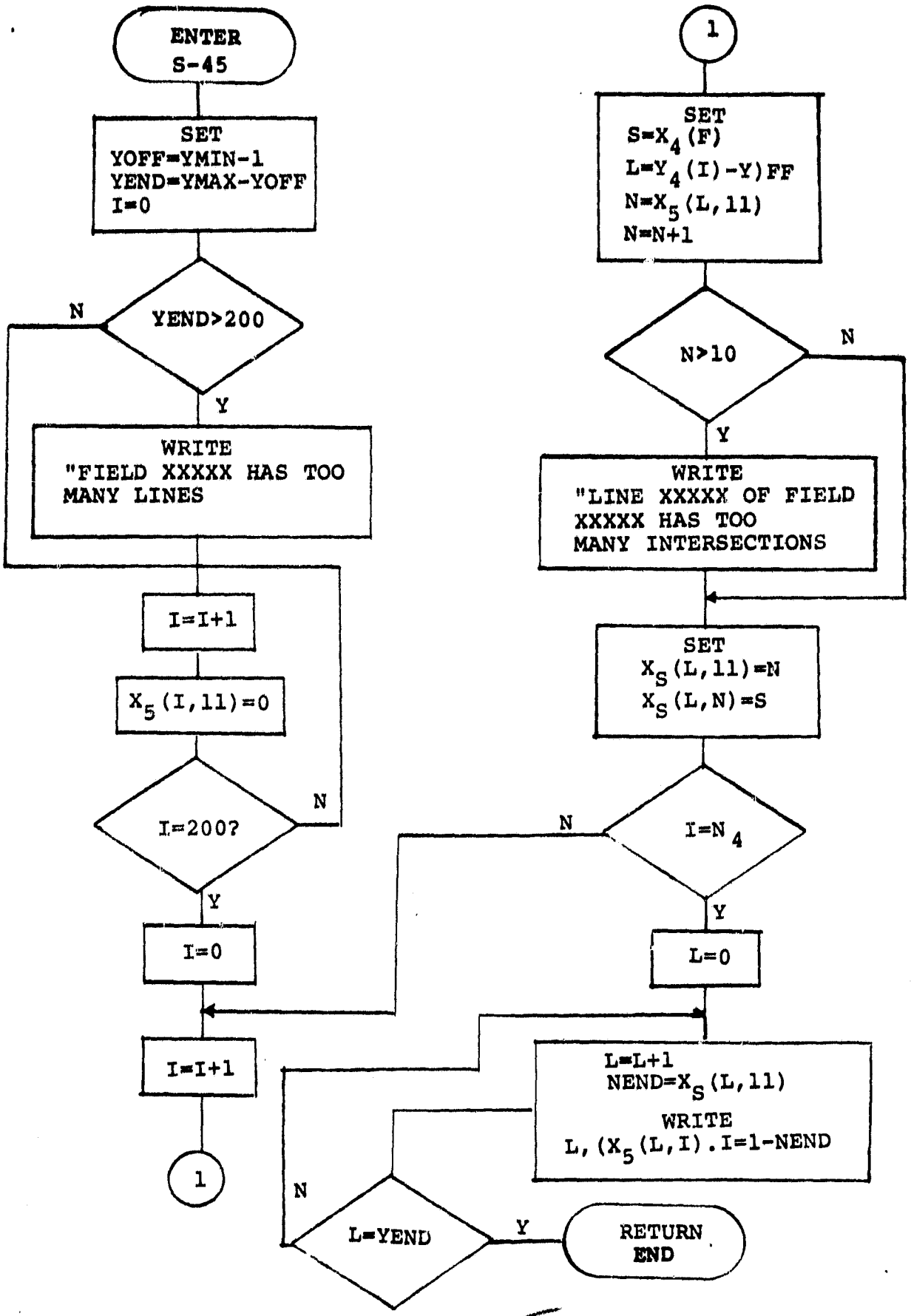
3.2.11.5 Storage

TBD

3.2.11.6 Description

Collects all field boundary intersections with given scene lines (intercepts).

3.2.11.7 Flowchart



3-64-65

3.2.12 FIRST UNIT SUBROUTINE (S-55)

3.2.12.1 Linkage

Called by "Phase 1" with simple return.

3.2.12.2 Interface

Communicates with calling routine through the common block "stuff" (see listing).

3.2.12.3 Input

All inputs are derived from the common block "stuff".

3.2.12.4 Output

Option is provided for trouble-shooting printout (listing of working buffer contents).

3.2.12.5 Storage

TBD

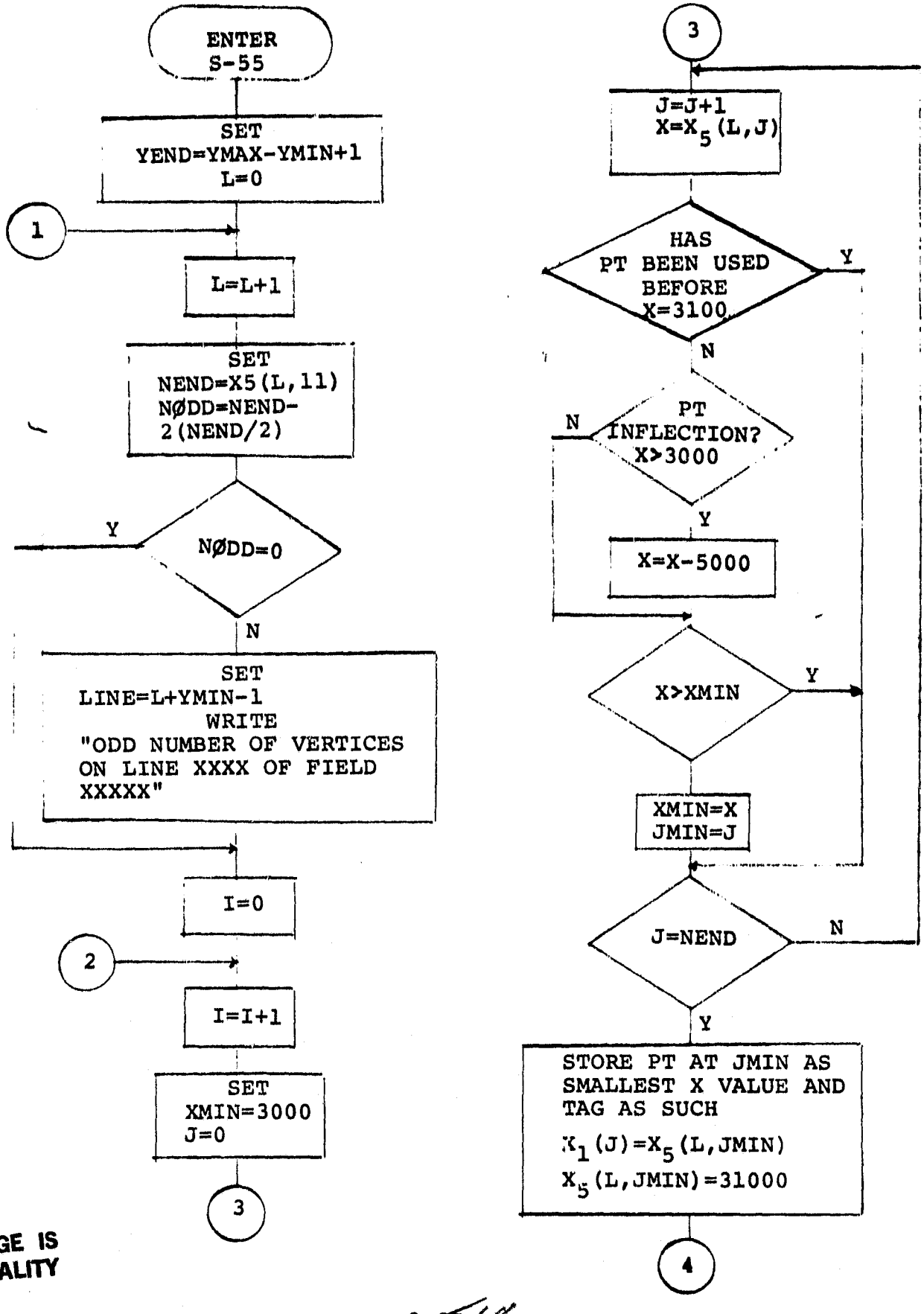
3.2.12.6 Description

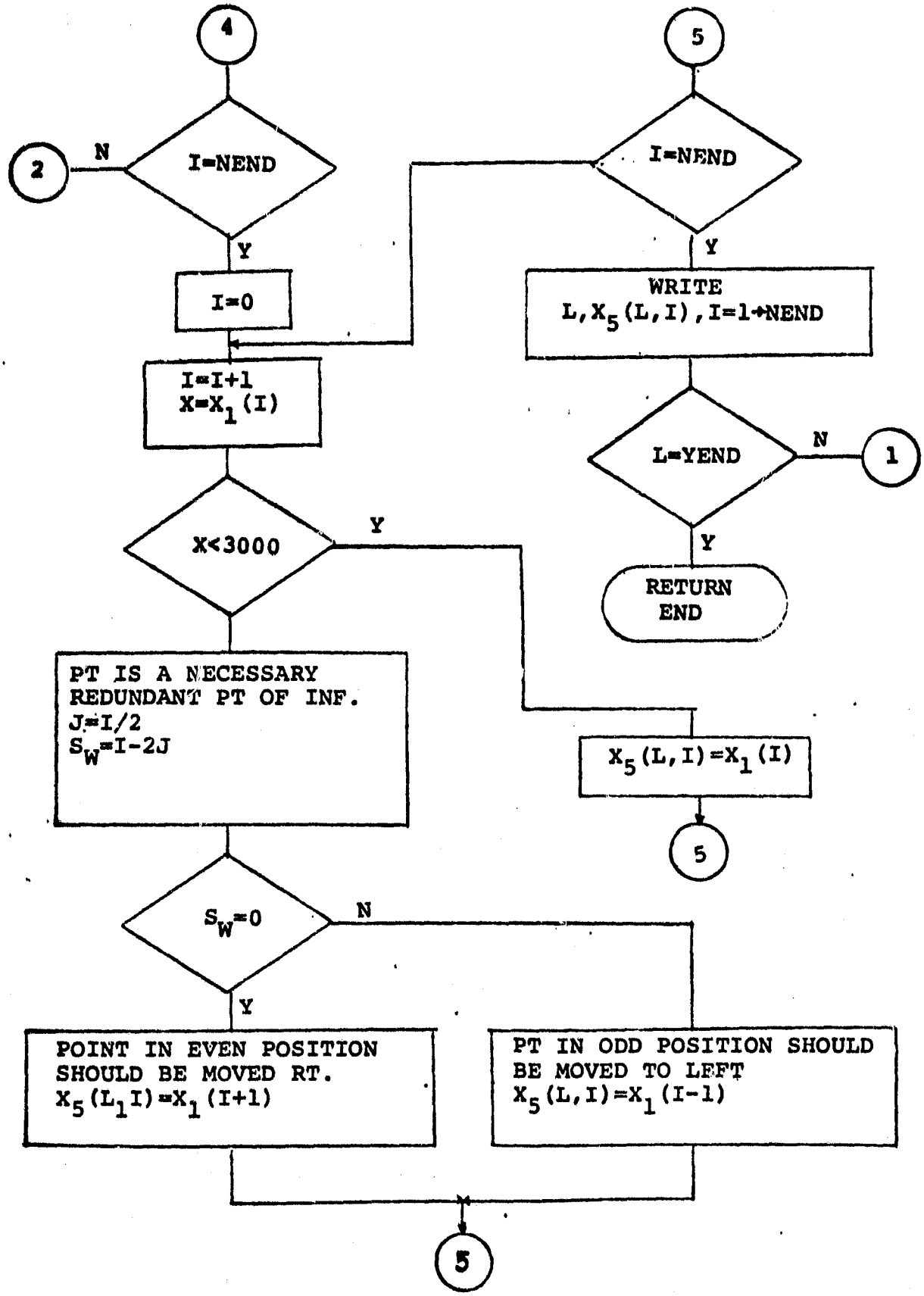
S-55 examines all intercepts, identifies all which are of special character or purpose, and places them in proper order.

3.2.11.8 Listing

```
HFORTRAN IV-PLUS V02-04                    17:28:37   22-APR-77          PAGE 1
S45.FTN                 /TRIPLACKS/WR
0001                    SUBROUTINE S45
                C  COLLECTS ALL INTERCEPTS WITHIN GIVEN LINES
0002                    IMPLICIT INTEGER (A-C),(S-Z)
0003                    COMMON /STUFF/X6(512),NPRT,BUF(60),N0,X1(50),Y1(50),N1,YMIN,YMAX
                       *X2(55),Y2(55),N2,X3(70),Y3(70),N3,X4(512),Y4(512),N4,X5(200,11),
0004                    YOFF=YMIN-1
0005                    YEND=YMAX-YOFF
0006                    IF(YEND.GT.200) WRITE(NPRT,102) BUF(2)
0007                102 FORMAT(' FIELD ',I15,' HAS TOO MANY LINES!')
0008                    DO 1 I=1,200
0009                    X5(I,11)=0
0010                  1 CONTINUE
0011                    DO 2 I=1,N4
0012                    S=X4(I)
0013                    L=Y4(I)-YOFF
0014                    N=X5(L,11)
0015                    N=N+1
0016                    IF(N.GT.10) WRITE(NPRT,103) L,BUF(2)
0017                103 FORMAT(' LINE',I15,' OF FIELD',I15,' HAS TOO MANY INTRSCTIONS
0018                    X5(L,11)=N
0019                    X5(L,N)=S
0020                  2 CONTINUE
0021                    DO 3 L=1,YEND
0022                    NEND=X5(L,11)
                D       WRITE(NPRT,101) L,(X5(L,I),I=1,NEND)
0023                101 FORMAT(1H ,11I7)
0024                  3 CONTINUE
0025                    RETURN
0026                    END
```

3.2.12.7 Flowchart




4
2
N
I=NEND
Y
I=0
I=I+1
X=X1(I)
X<3000
Y
PT IS A NECESSARY REDUNDANT PT OF INF.
J=I/2
SW=I-2J
SW=0
N
Y
POINT IN EVEN POSITION SHOULD BE MOVED RT.
X5(L,I)=X1(I+1)
PT IN ODD POSITION SHOULD BE MOVED TO LEFT
X5(L,I)=X1(I-1)
5
5
I=NEND
Y
WRITE
L,X5(L,I),I=1→NEND
L=YEND
N
1
Y
RETURN
END
X5(L,I)=X1(I)
5

3.2.12.8 Listing

```
HFORTRAN IV-PLUS V02-04          1712P146    22-APR-77          PAGE 1
S55.FTN          /TRIPLOCKS/WR
0001            SUBROUTINE S55
          C  PUTS INTERCEPTS IN ASCENDING ORDER
0002            IMPLICIT INTEGER (A-O),(S-Z)
0003            COMMON /STUFF/XA(512),NPRT,BUF(80),N0,X1(50),Y1(50),N1,YMIN,YMAX
               *X2(55),Y2(55),N2,X3(70),Y3(70),N3,X4(512),Y4(512),N4,X5(200,11),
0004            YEND=YMAX-YMIN+1
0005            DO 1 L=1,YEND
0006            NEND=X5(L,11)
0007            NODD=NEND-2*(NEND/2)
0008            IF(NODD.EQ.0) GO TO 6
          C  AN ODD NUMBER OF INTERSECTIONS IS NOT PERMITTED
0009            LINE=L+YMIN-1
0010            WRITE(NPRT,102) LINE,BUF(2)
0011          102 FORMAT(' ODD NUMBER OF VERTICES ON LINE ',I5,'  OF FIELD ',I5)
0012            6 CONTINUE
0013            DO 2 I=1,NEND
0014            XMIN=30000
0015            DO 3 J=1,NEND
0016            X=X5(L,J)
          C  IF THE POINT HAS BEEN USED BEFORE (AND TAGGED AS 31000) JUMP OVER I
0017            IF(X.EQ.31000) GO TO 3
          C  IF THE POINT IS TAGGED AS POINT OF INFLECTION SUBTRACT 5000
0018            IF(X.GT.30000) X=X-5000
0019            IF(X.GT.XMIN) GO TO 3
0020            XMIN=X
0021            JMIN=J
0022            3 CONTINUE
          C  POINT STORED AT JMIN HAS THE SMALLEST REMAINING X-VALUE
0023            X1(I)=X5(L,JMIN)
          C  TAG POINT AT JMIN AS HAVING BEEN USED
0024            X5(L,JMIN)=31000
0025            2 CONTINUE
0026            DO 4 I=1,NEND
0027            X=X1(I)
0028            IF(X.LT.30000) GO TO 5
          C  THIS POINT IS A NECESSARY REDUNDANT POINT OF INFLECTION
0029            J=I/2
0030            S=I-2*J
          C  POINT IN EVEN POSITION SHOULD BE MOVED TO RIGHT
0031            IF(S.EQ.0) X5(L,I)=X1(I+1)
          C  POINT IN ODD  POSITION SHOULD BE MOVED TO LEFT
0032            IF(S.NE.0) X5(L,I)=X1(I-1)
0033            GO TO 4
0034            5 CONTINUE
0035            X5(L,I)=X1(I)
0036            4 CONTINUE
          D     WRITE(NPRT,101) L,(X5(L,I),I=1,NEND)
0037          101 FORMAT(14 ,1117)
0038            1 CONTINUE
0039            RETURN
0040            END
```

3.2.13 FIRST UNIT SUBROUTINE (S-56)

3.2.13.1 Linkage

Called by "Phase 1" with simple return.

3.2.13.2 Interface

Communicates with the calling routine through the common block "stuff" (see listing).

3.2.13.3 Input

All inputs are derived from the common block "stuff".

3.2.13.4 Output

Option is provided for trouble-shooting printout (listing of working buffer contents).

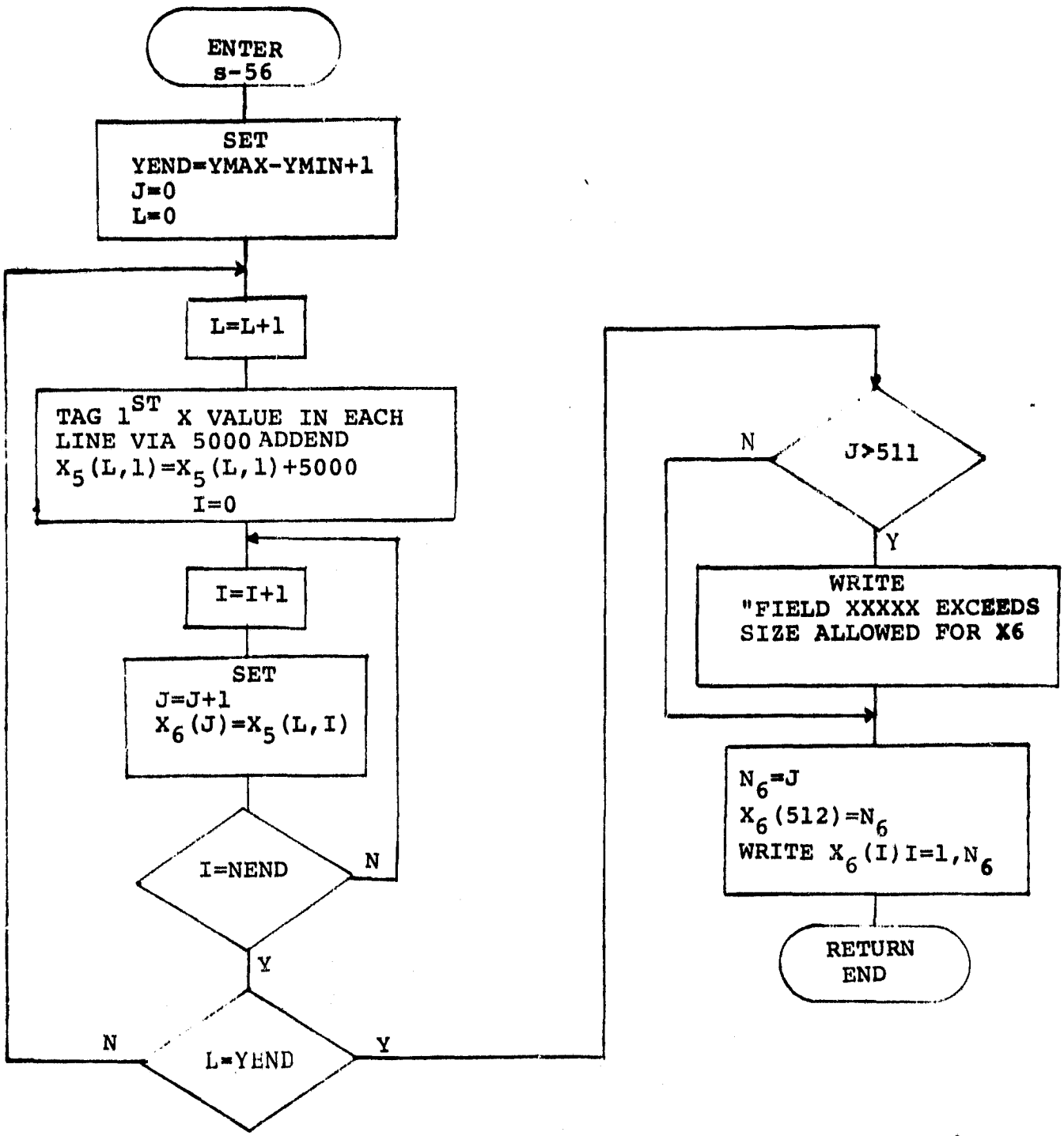
3.2.13.5 Storage

TBD

3.2.13.6 Description

S-56 packs ordered intercepts into a one-dimensional buffer.

3.2.13.7 Flowchart



1.2.13.8 Listing

```
HFORTRAN IV-PLUS V02-04                    17:20:56    22-APR-77          PAG
S56.FTN          /TR:BLOCKS/WR
      SUBROUTINE S56
C  PACKS INTERCEPTS INTO A ONE-DIMENSIONAL BUFFER
      IMPLICIT INTEGER (A-Q),(S-Z)
      COMMON /STUFF/X6(512),NPRT,BUF(80),N0,X1(50),Y1(50),N1,YMIN,Y
     *X2(55),Y2(55),N2,X3(72),Y3(72),N3,X4(512),Y4(512),N4,X5(200,1
      YEND=YMAX-YMIN+1
      J=0
      DO 1 L=1,YEND
C  TAG THE FIRST X-VALUE IN EACH LINE BY ADDING 5000
      X5(L,1)=X5(L,1)+5000
      NEND=X5(L,11)
      DO 2 I=1,NEND
      J=J+1
      X6(J)=X5(L,I)
    2 CONTINUE
    1 CONTINUE
      IF(J.GT.511) WRITE(NPRT,102) BUF(2)
  102 FORMAT('  FIELD ',I15,'  EXCEEDS THE SIZE ALLOWED FOR X6')
      N6=J
      X6(512)=N6
D     WRITE(NPRT,101) (X6(I),I=1,N6)
  101 FORMAT(1H ,20I6)
      RETURN
      END
```

3.2.14 SECOND UNIT OF FIRST MODULE (PHASE 2)

3.2.14.1 Linkage

This is a stand-alone program which calls only standard system utility routines.

3.2.14.2 Interface

Two pre-loaded system files are used to interface with the companion program "Phase 1".

3.2.14.3 Input

All inputs are drawn from two system files loaded by a previous execution of the companion unit "Phase 1" and one system file loaded by a previous execution of the companion unit BTREAD.

3.2.14.14 Output

The main output product is a magnetic tape containing ground truth data in Universal format (see reference 1). This is accompanied by a per-line print out of field start and end positions and of the crop type. Provisions for optional trouble shooting print out are included in the program.

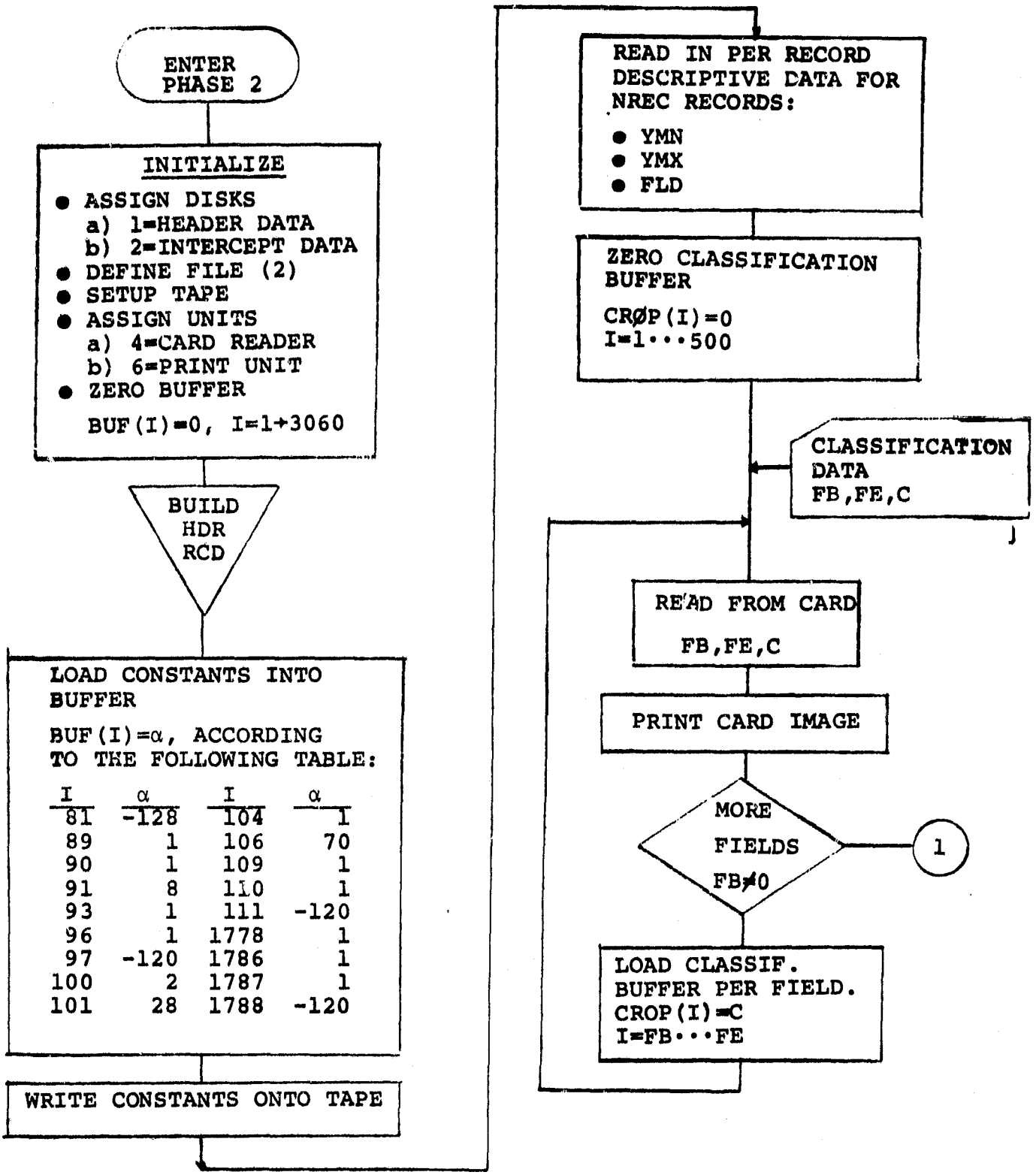
3.2.14.5 Storage

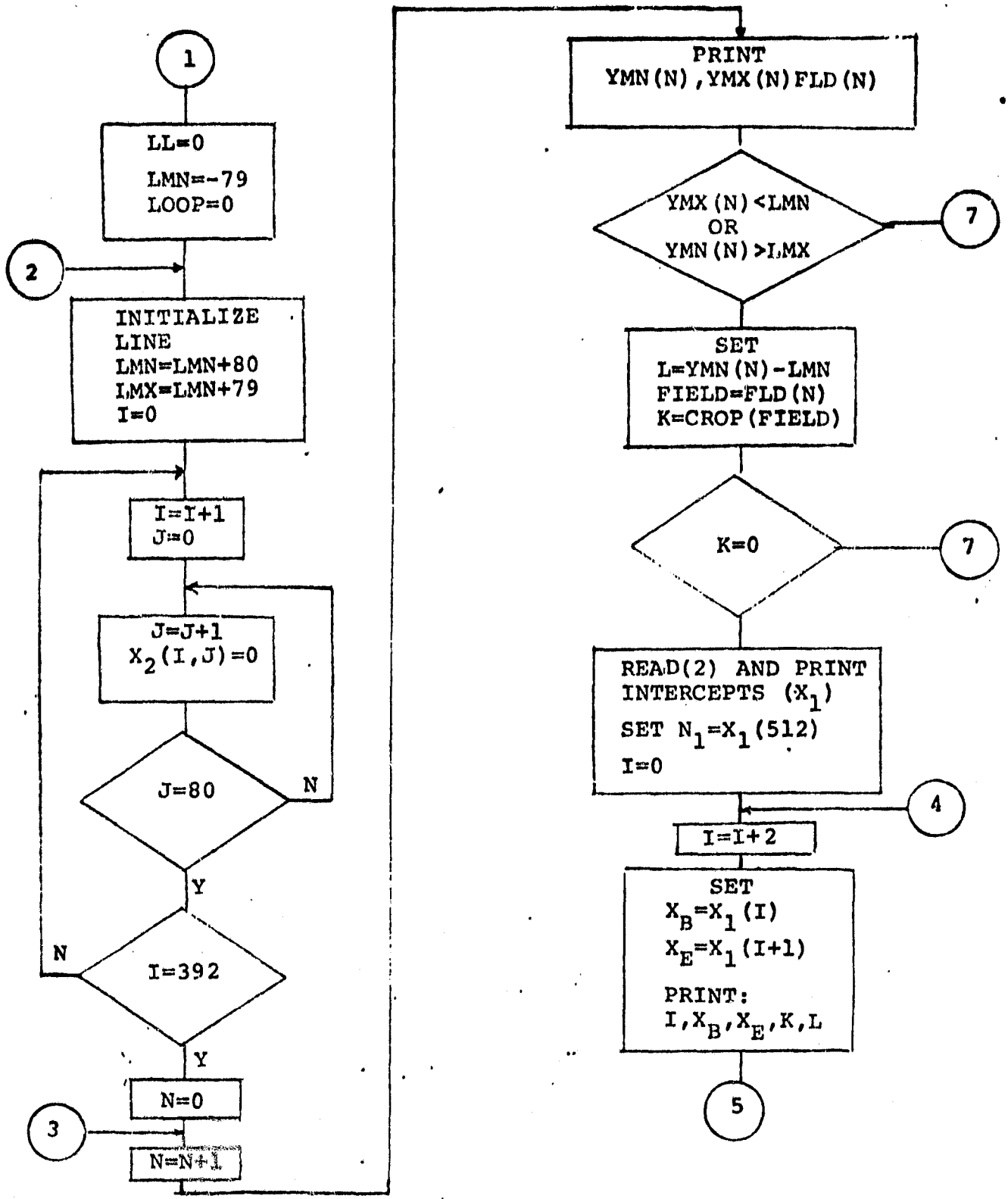
TBD

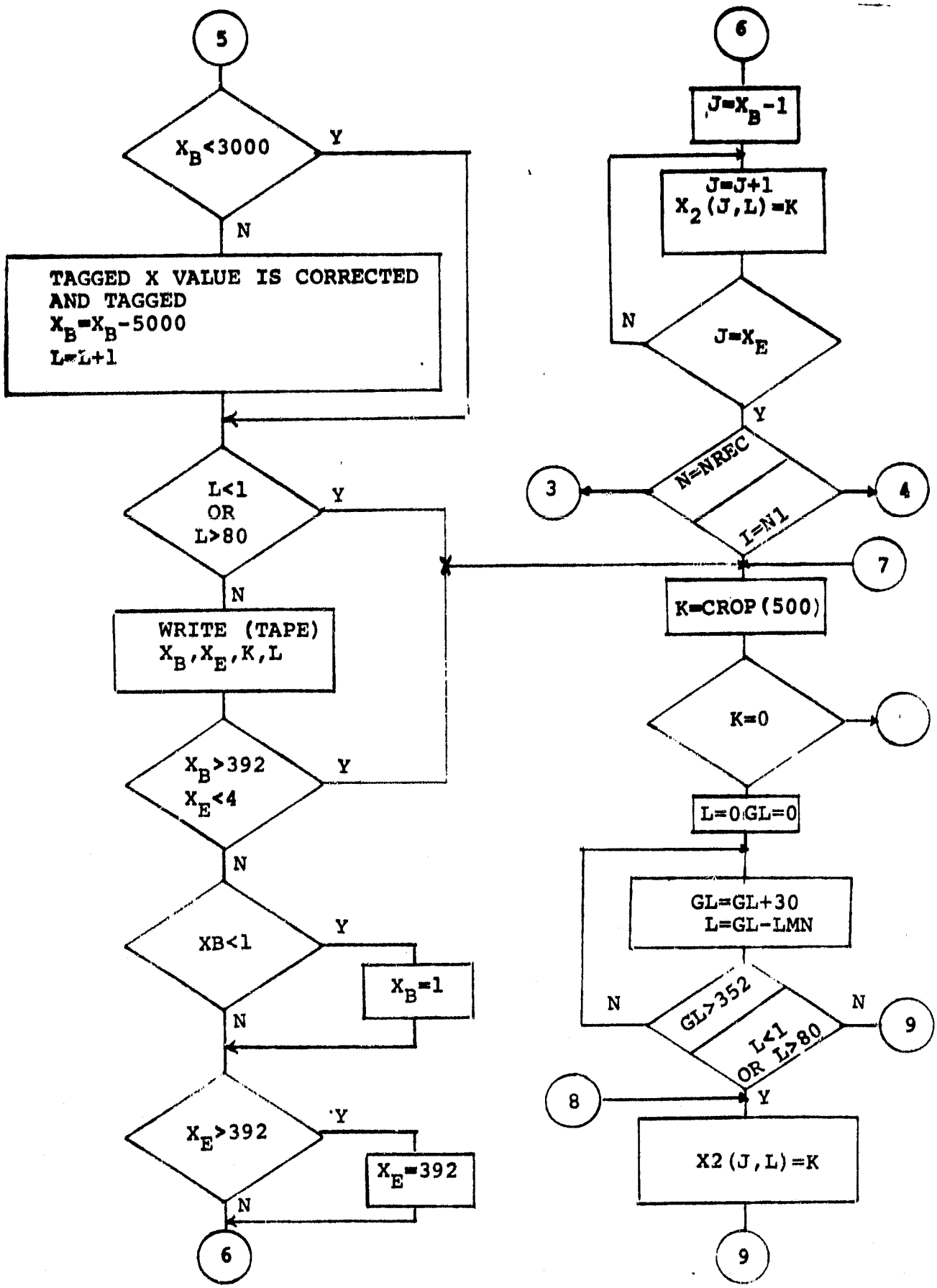
3.2.14.6 Description

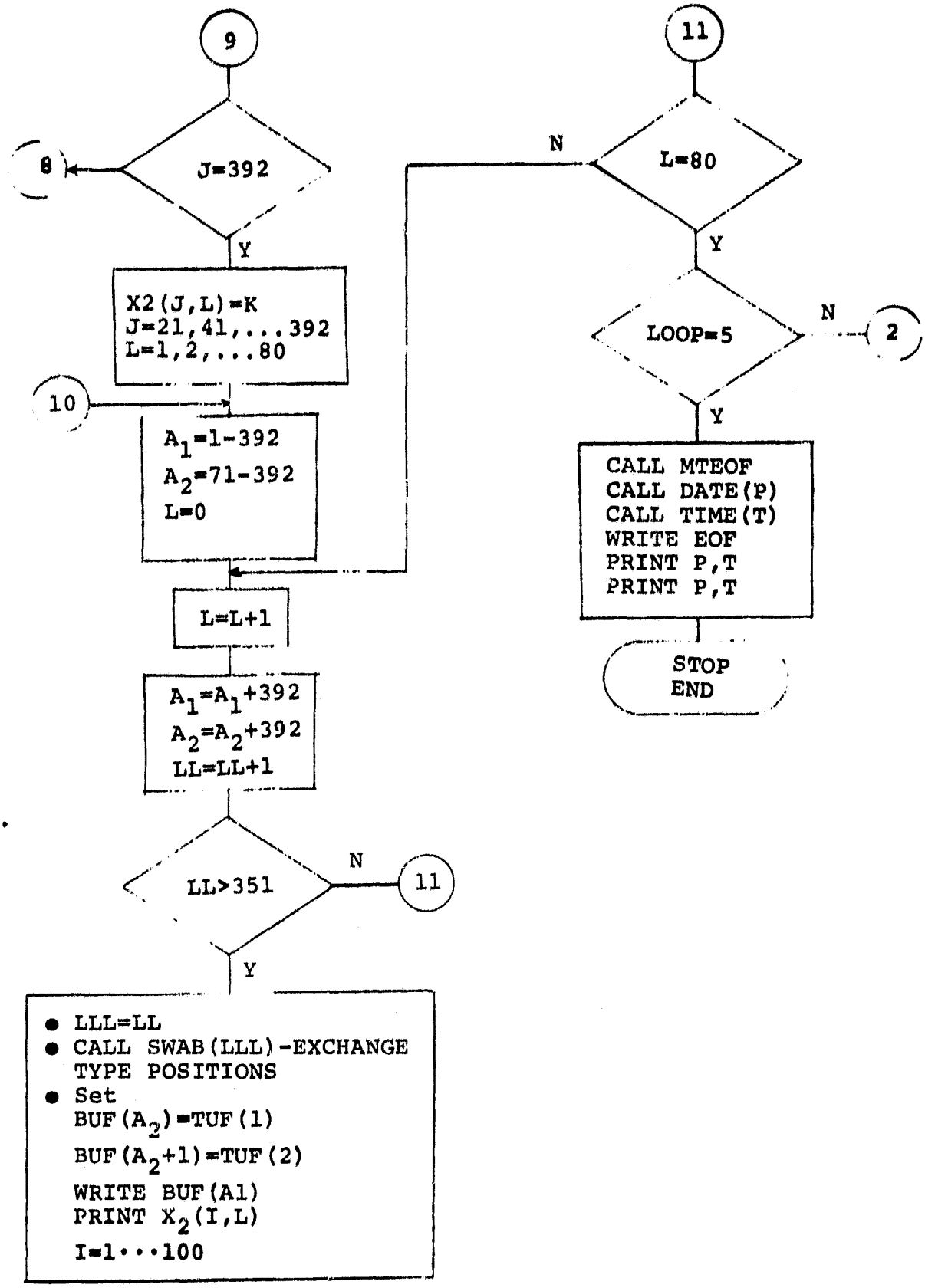
Phase 2 operates on the "Header" and "Intercept" files constructed by a previous execution of the companion Phase 1 unit. Phase 2 also uses crop label files w-ich are constructed by the program BTREAD. It organizes those data for output as a properly structured "ground truth" data tape (magnetic tape in Universal format).

3.2.14.7 Flowchart




1
LL=0
LMN=-79
LOOP=0
2
INITIALIZE
LINE
LMN=LMN+80
LMX=LMN+79
I=0
I=I+1
J=0
J=J+1
X2(I,J)=0
J=80
N
Y
I=392
N
Y
N=0
3
N=N+1
PRINT
YMN(N),YMX(N)FLD(N)
YMX(N)<LMN
OR
YMN(N)>LMX
7
SET
L=YMN(N)-LMN
FIELD=FLD(N)
K=CROP(FIELD)
K=0
7
READ(2) AND PRINT
INTERCEPTS (X1)
SET N1=X1(512)
I=0
4
I=I+2
SET
XB=X1(I)
XE=X1(I+1)
PRINT:
I,XB,XE,K,L
5


5
X_B<3000
Y
N
TAGGED X VALUE IS CORRECTED
AND TAGGED
$X_B=X_B-5000$
L=L+1
L<1
OR
L>80
Y
N
WRITE (TAPE)
X_B, X_E, K, L
X_B>392
X_E<4
Y
N
XB<1
Y
$X_B=1$
N
X_E>392
Y
$X_E=392$
N
6
6
$J=X_B-1$
J=J+1
$X_2(J,L)=K$
N
$J=X_E$
Y
N=NREC
I=N1
3
4
7
K=CROP(500)
K=0
L=0 GL=0
GL=GL+30
L=GL-LMN
N
GL>352
N
L<1
OR L>80
9
8
Y
X2(J,L)=K
9


9
J=392
8
Y
X2(J,L)=K
J=21,41,...392
L=1,2,...80
10
A1=1-392
A2=71-392
L=0
L=L+1
A1=A1+392
A2=A2+392
LL=LL+1
LL>351
N
11
Y
LLL=LL
CALL SWAB(LLL)-EXCHANGE TYPE POSITIONS
Set
BUF(A2)=TUF(1)
BUF(A2+1)=TUF(2)
WRITE BUF(A1)
PRINT X2(I,L)
I=1...100
11
L=80
N
Y
LOOP=5
N
2
Y
CALL MTEOF
CALL DATE(P)
CALL TIME(T)
WRITE EOF
PRINT P,T
PRINT P,T
STOP
END

3.2.14.8 Listing

```
C  PROCESSES INTERCEPT FILE TO PRODUCE UNIVERSAL TAPE
      IMPLICIT INTEGER (A-Q),(S-Z)
      DIMENSION X1(512),YMN(500),YMX(500),FLD(500),CROP(500)
      BYTE X2(392,80),BUF(3060),TUF(2),T(8),D(9),SDEV
      EQUIVALENCE (BUF(73),X2(1,1)),(TUF(1),LLL)
      EQUIVALENCE (S,BUF(67))
      COMMON /STATUS/W1,W2
      CALL TIME(T)
      CALL DATE(D)
         NRDD=1
      NRDR=4
      NPRT=6
      WRITE(NPRT,703) D,T
  703 FORMAT(1H1,'  JOB INITIATED ON  ',9A1,' AT  ',8A1,//,10X,
     1'PROGRAM  PHASE2,FTN')
         OPEN (UNIT=NRDR,NAME='PHASE2,DAT',TYPE='OLD',
     1ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
      READ(NRDR,301)  SDEV,NDEV,FILE
  301 FORMAT(A1,1X,2I2)
      WRITE (NPRT,302) SDEV,NDEV,FILE
  302 FORMAT(//,10X,A1,'T',10X,'DEVICE NO.=',I5,10X,'FILE NO.=',I5)
         CALL CLOSE(NRDR)
                        OPEN(UNIT=NRDR,NAME='LABEL,DAT',TYPE='OLD',
     1ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
         READ(NRDR,305) SS,DAY,MON,YR
  305 FORMAT(4I5)
         OPEN(UNIT=1,NAME='HEAD,DAT',TYPE='OLD',
     1ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
         OPEN(UNIT=2,NAME='INTCPT,DAT',TYPE='OLD',
     1ACCESS='DIRECT',FORM='UNFORMATTED',CARRIAGE CONTROL='NONE',
     2RECORDSIZE=256,MAXREC=500,ASSOCIATEVARIABLE=AV)
      IDEV=0
      IF(SDEV.EQ.88)    IDEV=1
      IF(NDEV.NE.0.AND.NDEV.NE.1) GO TO 14
      CALL  TINIT(3,IDEV,NDEV)
      CALL  TATCH(3)
      CALL  TRWD(3)
      CALL  TWAIT(3)
      CALL  TFILE(3,(FILE-1))
      CALL  TWAIT(3)
      DO 12 I=1,3060
      BUF(I)=0
   12 CONTINUE
         S=SS
      BUF(61)=DAY
      BUF(62)=MON
      BUF(63)=YR
      WRITE(NPRT,306) S,(BUF(IB),IB=61,63)
  306 FORMAT(' SITE=  ',I15,5X,'DAY=',I5,5X,'MON=',I5,5X,'YEAR=',I5)
      CALL SWAB(S)
      BUF(81)=-128
      BUF(89)=1
      BUF(90)=1
      BUF(91)=8
      BUF(93)=1
      BUF(96)=1
      BUF(97)=-120
      BUF(100)=2
      BUF(101)=28
      BUF(104)=1
      BUF(104)=70
```

```
      BUF(109)=1
      BUF(110)=1
      BUF(111)=-120
      BUF(1778)=1
      BUF(1786)=1
      BUF(1787)=1
      BUF(1788)=-120
      CALL  TWRIT(3,BUF,1530)
      CALL  TWAIT(3)
      REWIND 1
      READ(1,201) NREC
  201 FORMAT(115)
      READ(1,202) (YMN(I),I=1,NREC)
      READ(1,202) (YMX(I),I=1,NREC)
      READ(1,202) (FLD(I),I=1,NREC)
  202 FORMAT(5015)
      DO 1 I=1,500
      CROP(I)=0
    1 CONTINUE
      WRITE(NPRT,500)
  500 FORMAT(//,10X,'FIELD TO CODE TRANSFORMATION',//,7X,'FIELD',2X,
     1'TO',2X,'FIELD',6X,'CODE')
    2 CONTINUE
      READ(NRDR,203) FB,FE,C
  203 FORMAT(3I5)
      WRITE(NPRT,206) FB,FE,C
  206 FORMAT(1H ,3I10)
      IF(C.EQ.-1) GO TO 555
      IF(FB.EQ.0) GO TO 4
      DO 3 I=FB,FE
      CROP(I)=C
    3 CONTINUE
      GO TO 2
  555 CONTINUE
          DO 501 I=1,250
          II=I+250
          CROP(II)=I
  501     CROP(I) = I
    4 CONTINUE
      LL=0
      LMN=-79
      DO 13 LOOP=1,5
      LMN=LMN+80
      LMX=LMN+79
      DO 10 I=1,392
      DO 11 J=1,80
      X2(I,J)=0
   11 CONTINUE
   10 CONTINUE
      DO 5 N=1,NREC
D     WRITE(NPRT,208) YMN(N),YMX(N),FLD(N)
  208 FORMAT(1H ,3I10)
      IF((YMX(N).LT.LMN).OR.(YMN(N).GT.LMX)) GO TO 5
      L=YMN(N)-LMN
      FIELD=FLD(N)
      K=CROP(FIELD)
      IF(K.EQ.0) GO TO 5
         K=K-128
      READ(2'N) X1
      N1=X1(512)
D     WRITE(NPRT,888) (X1(I),I=1,N1)
  888 FORMAT(1H ,2016)
      DO 6 I=1,N1,2
      XB=X1(I)
      XE=X1(I+1)
D     WRITE(NPRT,888) I,XB,XE,K,L
```

3-79 SG

```
      IF(XB.LT.3000) GØ TØ 7
C  TAGGED X-VALUE IS CORRECTED BY SUBTRACTING 5000
      XB=XB-5000
C  TAGGED X-VALUE SIGNALS START ØF A NEW LINE
      L=L+1
    7 CØNTINUE
      IF(L.LT.1) GØ TØ 6
      IF(L.GT.80) GØ TØ 5
D     WRITE(NPRT,207) I,XB,XE,K,L
  207 FØRMAT(1H ,5I2Ø)
      IF(XB.GT.392) GØ TØ 6
      IF(XE.LT.1) GØ TØ 6
      IF(XB.LT.1) XB=1
      IF(XE.GT.392) XE=392
      DØ 8 J=XB,XE
      X2(J,L)=K
    8 CØNTINUE
    6 CØNTINUE
    5 CØNTINUE
      K=CRØP(500)
      IF(K.EQ.Ø) GØ TØ 19
         K=K-128
      DØ 15 GL=32,352,30
      L=GL-LMN
      IF((L.LT.1).ØR.(L.GT.80)) GØ TØ 15
      DØ 16 J=1,392
      X2(J,L)=K
   16 CØNTINUE
   15 CØNTINUE
      DØ 17 L=1,8Ø
      DØ 18 J=21,392,20
      X2(J,L)=K
   18 CØNTINUE
   17 CØNTINUE
   19 CØNTINUE
      A1=1-392
      A2=71-392
      DØ 9 L=1,80
      A1=A1+392
      A2=A2+392
      LL=LL+1
      IF(LL.GT.351) GØ TØ 14
      LLL=LL
      CALL SWAB(LLL)
      BUF(A2)=TUF(1)
      BUF(A2+1)=TUF(2)
      CALL  TWRIT(3,BUF(A1),270)
      CALL  TWAIT(3)
D     WRITE(NPRT,204) (X2(I,L),I=1,100)
  204 FØRMAT(1H ,100I1)
    9 CØNTINUE
   13 CØNTINUE
   14 CØNTINUE
      CALL  TEØF(3)
      CALL  TWAIT(3)
         CALL TEØF(3)
      CALL DATE(D)
      CALL TIME(T)
      WRITE(NPRT,205) D,T
      WRITE(NPRT,205) D,T
  205 FØRMAT('  JØB CØMPLETED ØN  ',9A1,'  AT  ',8A1)
      STØP
      END
```

```
PHASE2,LP1/SH=PHASE2
[100,4]SWAB
[100,4]LECTAP
[1,1]F4P0TS/LB
/
ASG=SY11
ASG=SY12
ASG=SY14
ASG=LP16
ACTFIL=8
MAXBUF=3060
PRI=100
//
```

3.2.15 SECOND MODULE - FIRST UNIT (SPATL)

3.2.15.1 Linkage

SPATL is a stand-alone program which calls only standard system utility routines and the special subroutines TAB and ZAP.

3.2.15.2 Interface

Communication with subroutines is through the calling arguments and the common BK1.

3.2.15.3 Input

SPATL requires input of a "ground truth" magnetic disk file product of an earlier execution of the first module of this system and of card entries of corresponding analyst "dot" labeling data (see Appendix A).

3.2.15.4 Output

Printout of accuracy assessment parameters (see Appendix B) including:

1. True wheat proportion
2. Proportion of wheat pixels on field boundaries
3. "Dots" labeled by ground truth
4. Probability of misclassification of analyst labeled dots.

A complete description of the SPATL printout is included in Appendix B.

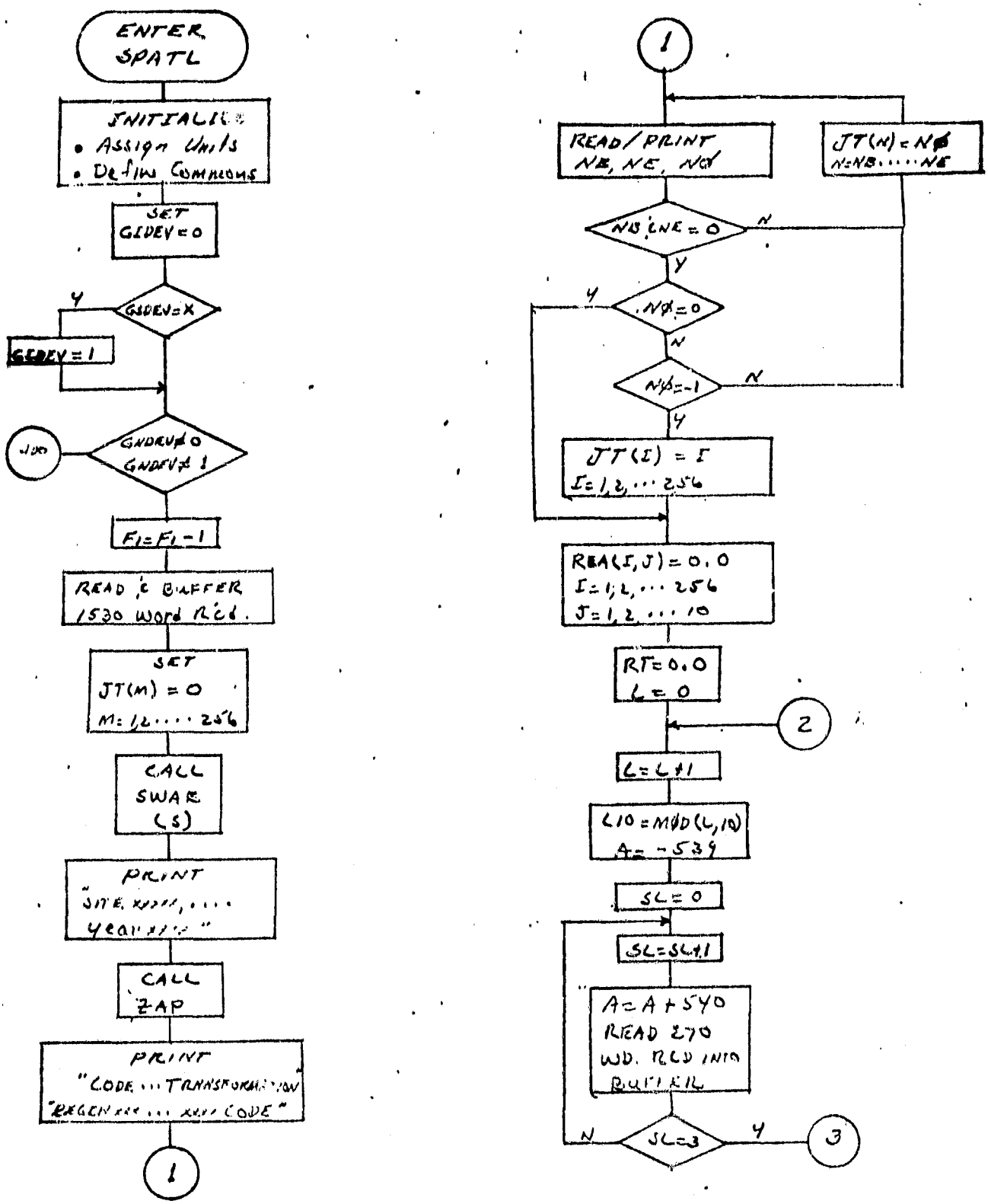
3.2.15.5 Storage

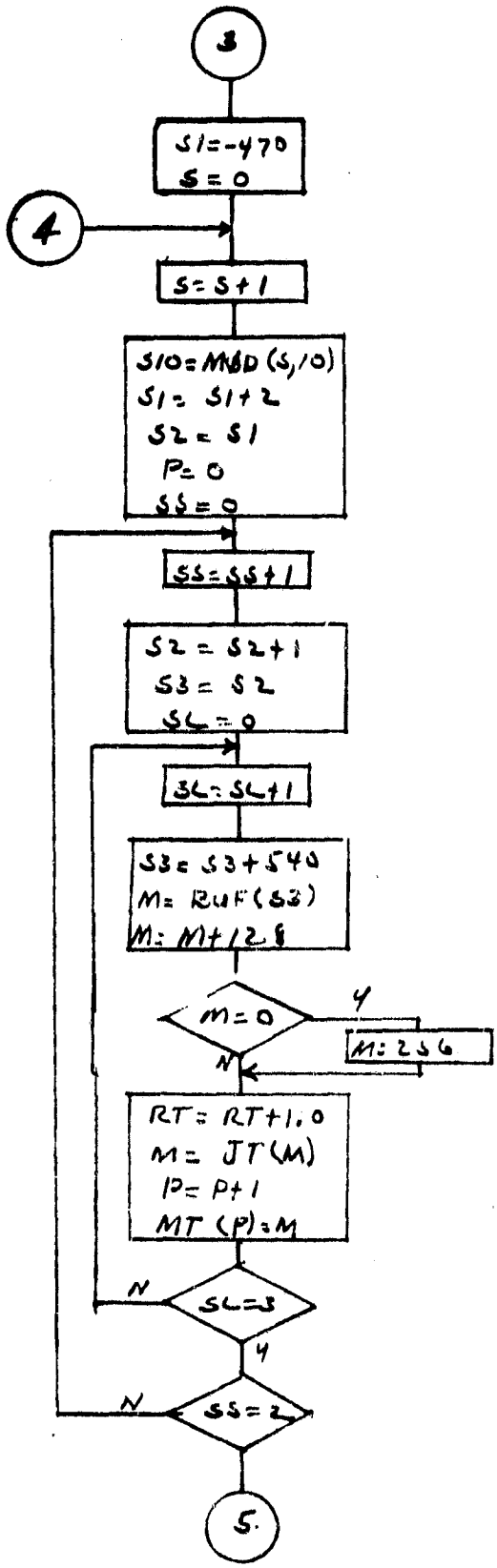
TBD

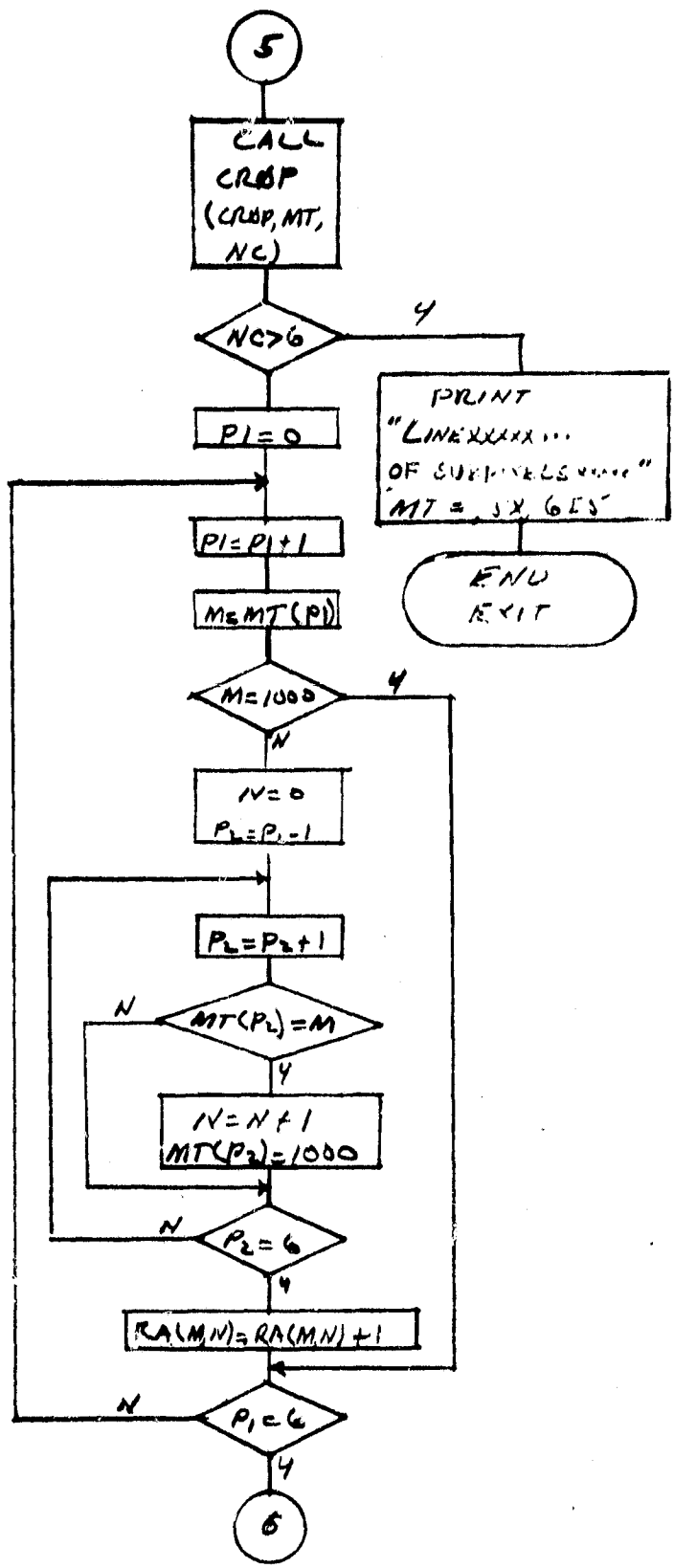
3.2.15.6 Description

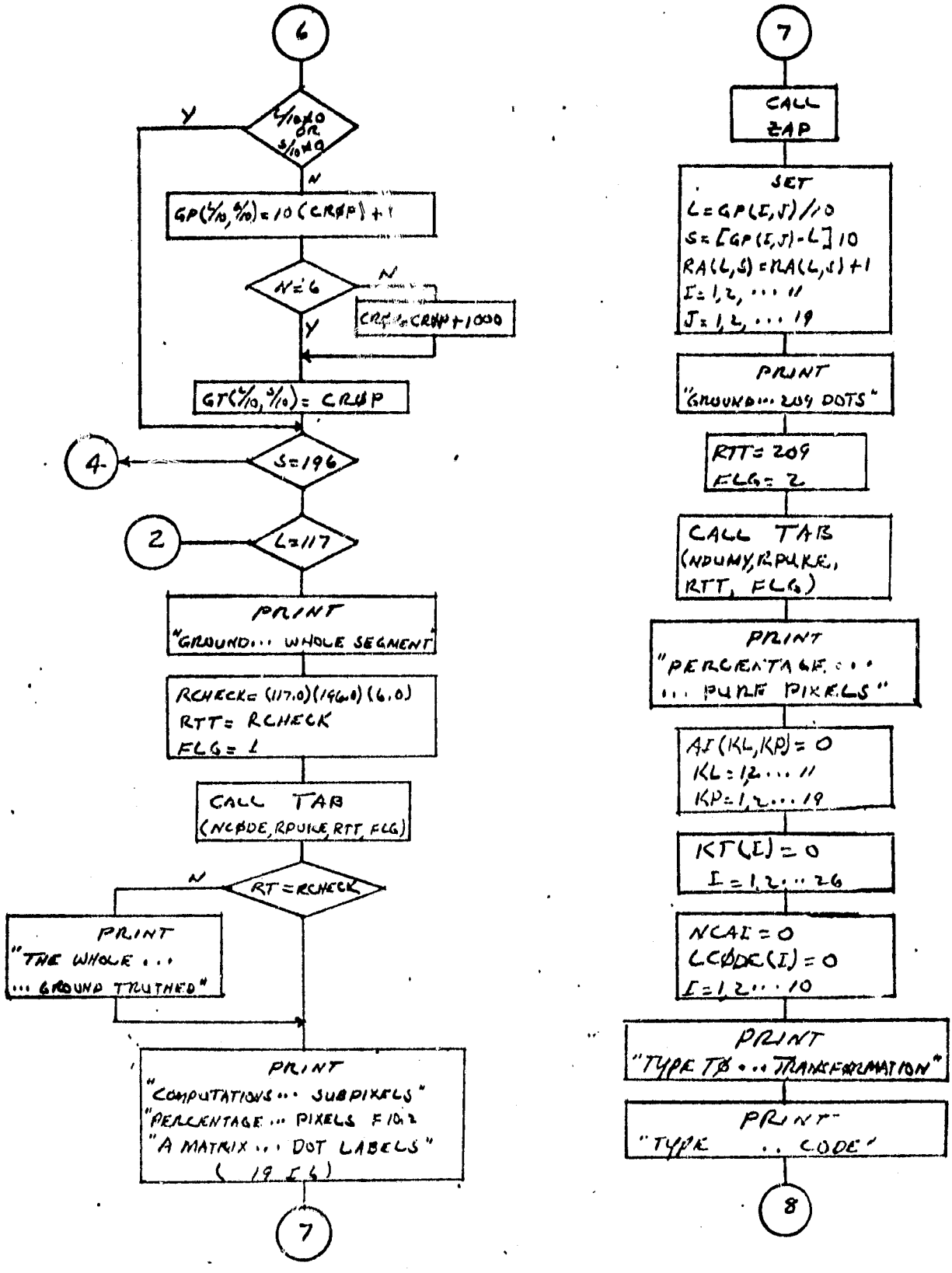
SPATL compares ground truth data with analyst dot labeling data to determine the accuracy of that analyst labeling.

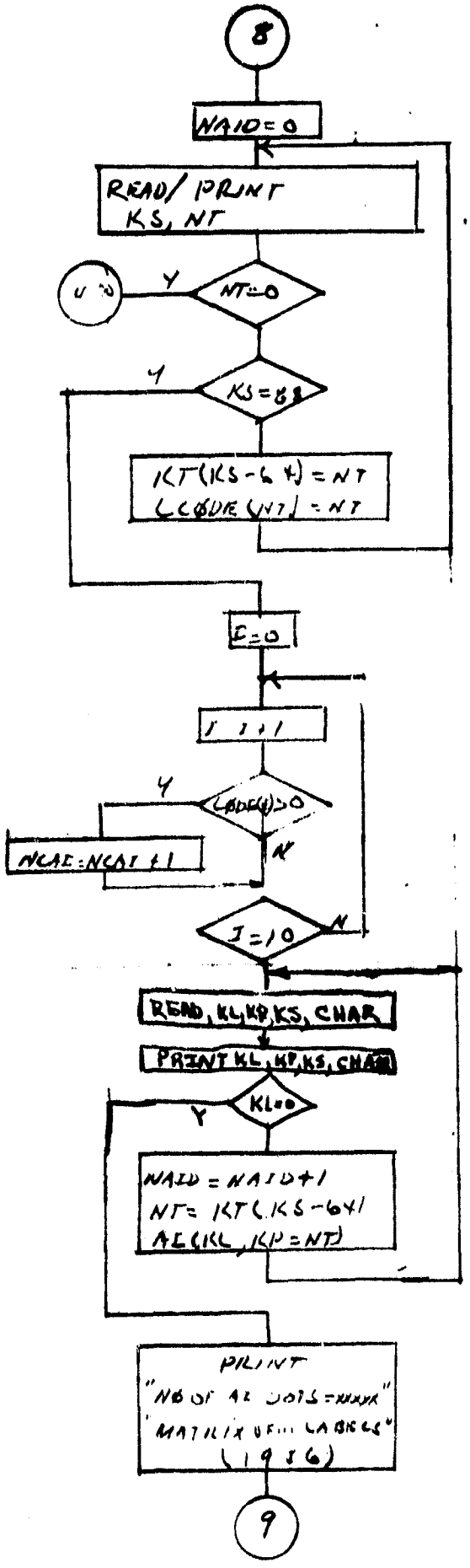
3.2.15.7 Flowchart

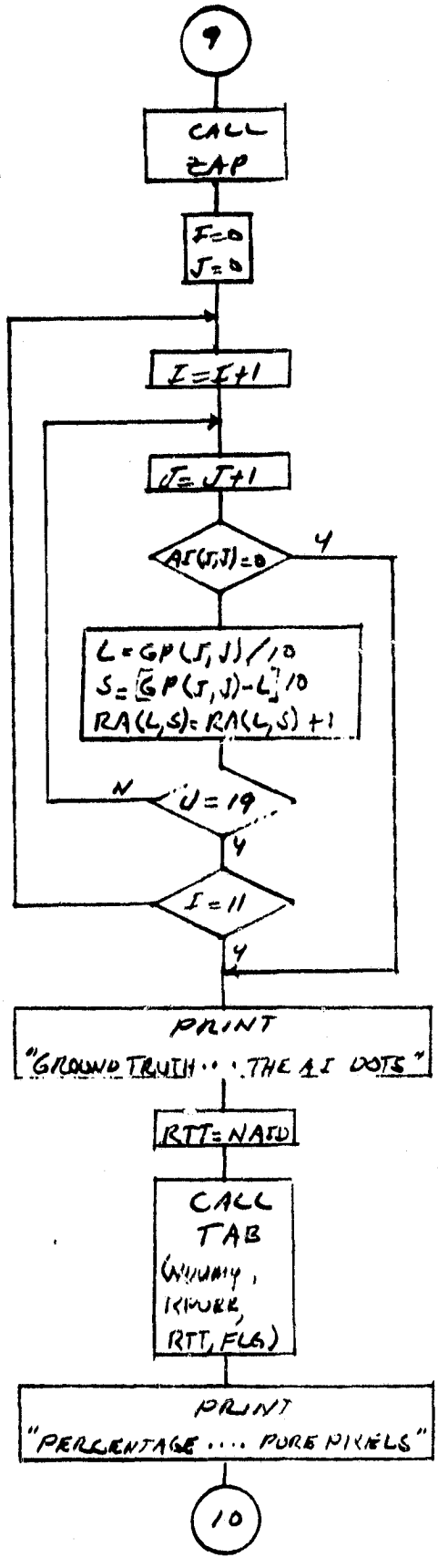


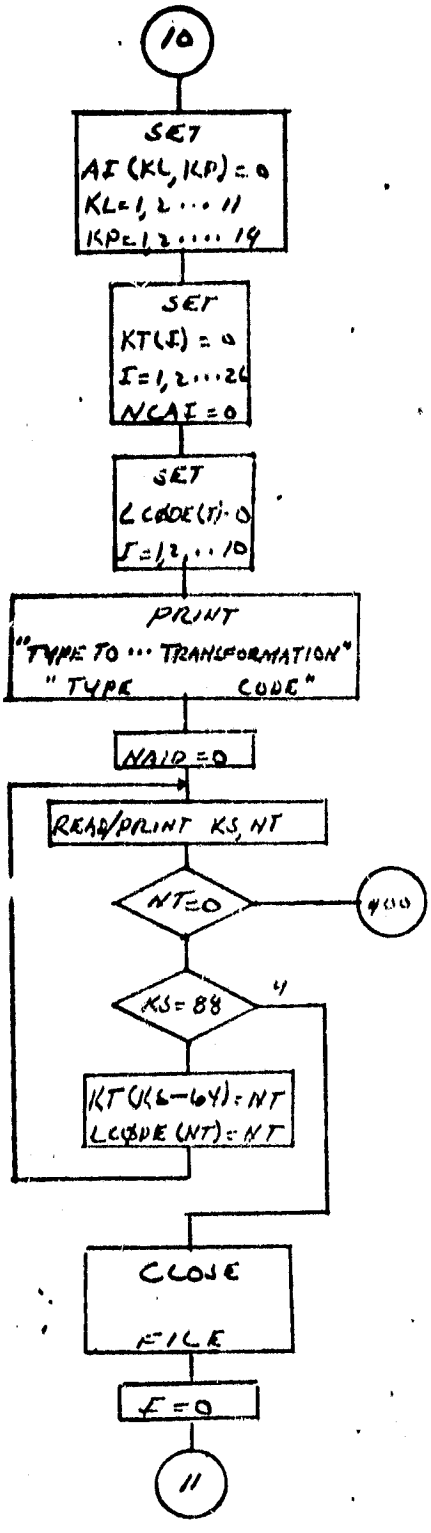

3
S1=-470
S=0
4
S=S+1
S10=MSD(S,10)
S1=S1+2
S2=S1
P=0
SS=0
SS=SS+1
S2=S2+1
S3=S2
SL=0
SL=SL+1
S3=S3+540
M=RUF(S3)
M=M+128
M=0
Y
M=256
N
RT=RT+1.0
M=JT(M)
P=P+1
MT(P)=M
N
SL=3
Y
N
SS=2
5

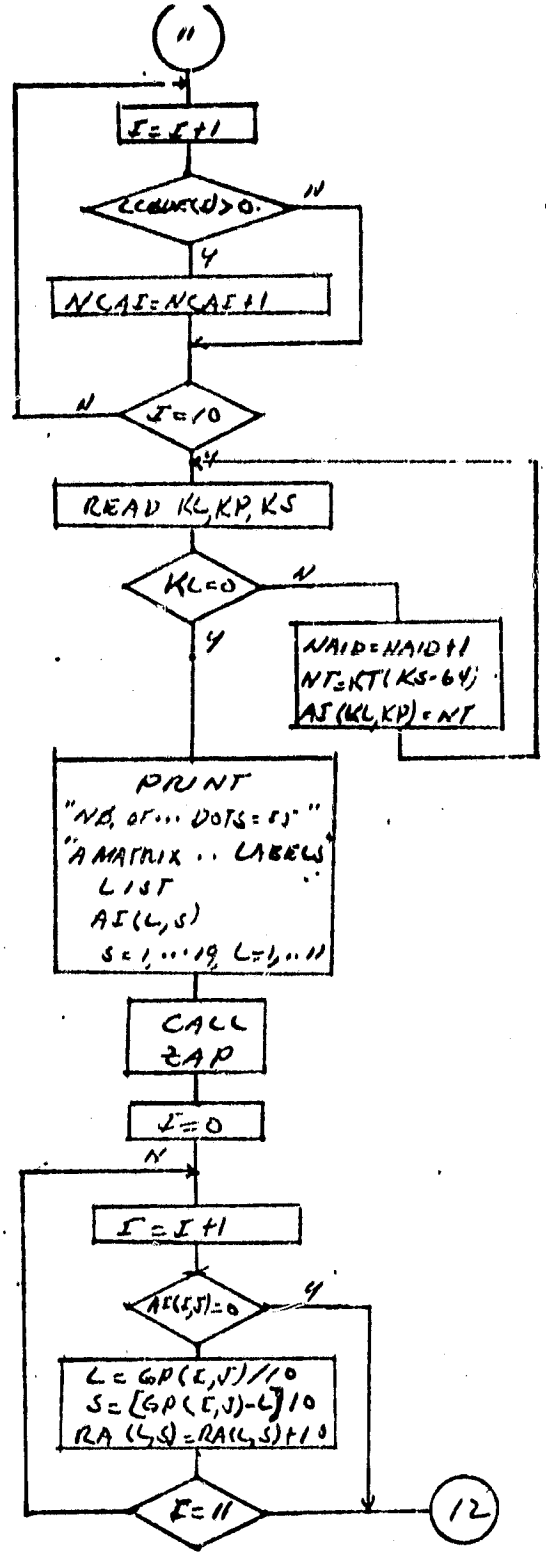

5
CALL CRDP (CRDP, MT, NC)
NC>6
Y
PRINT "LINEXXXXX... OF SUBROUTINE..."
MT=,5X,6ES
END
EXIT
P1=0
P1=P1+1
M=MT(P1)
M=1000
Y
N
IV=0
P2=P1-1
P2=P2+1
N
MT(P2)=M
Y
IV=N+1
MT(P2)=1000
N
P2=6
Y
RA(M,N)=RA(M,N)+1
N
P1=6
Y
6

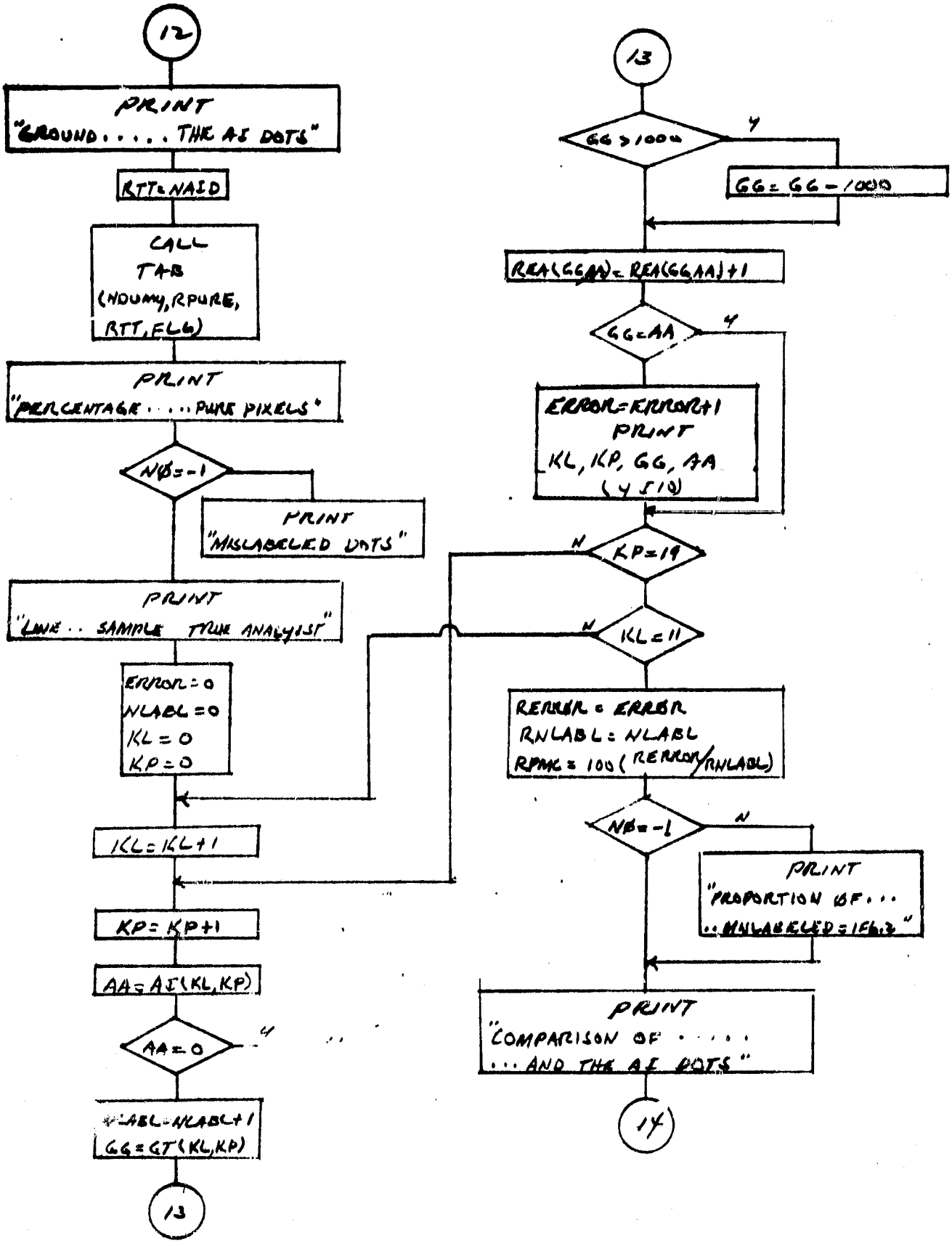


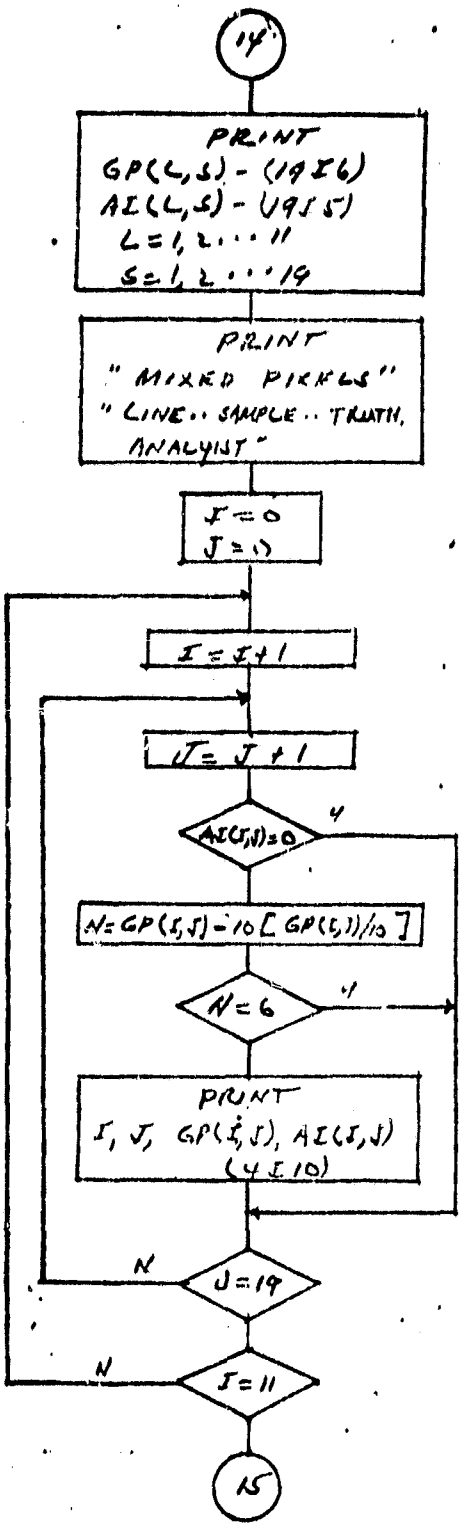

8
NAID=0
READ/PRINT
KS, NT
Y
NT=0
4
KS=64
ICT(KS-64)=NT
LCODE(NT)=NT
I=0
I=I+1
4
LADD(I)>0
N
NCAI=NCAI+1
N
I=10
READ, KL, KP, KS, CHAR
PRINT KL, KP, KS, CHAR
Y
KL=0
NAID=NAID+1
NT=ICT(KS-64)
AI(KL, KP)=NT
PRINT
"NO OF AI DOTS=NNNN"
"MATRIX OF... LABELS"
(19 I 6)
9

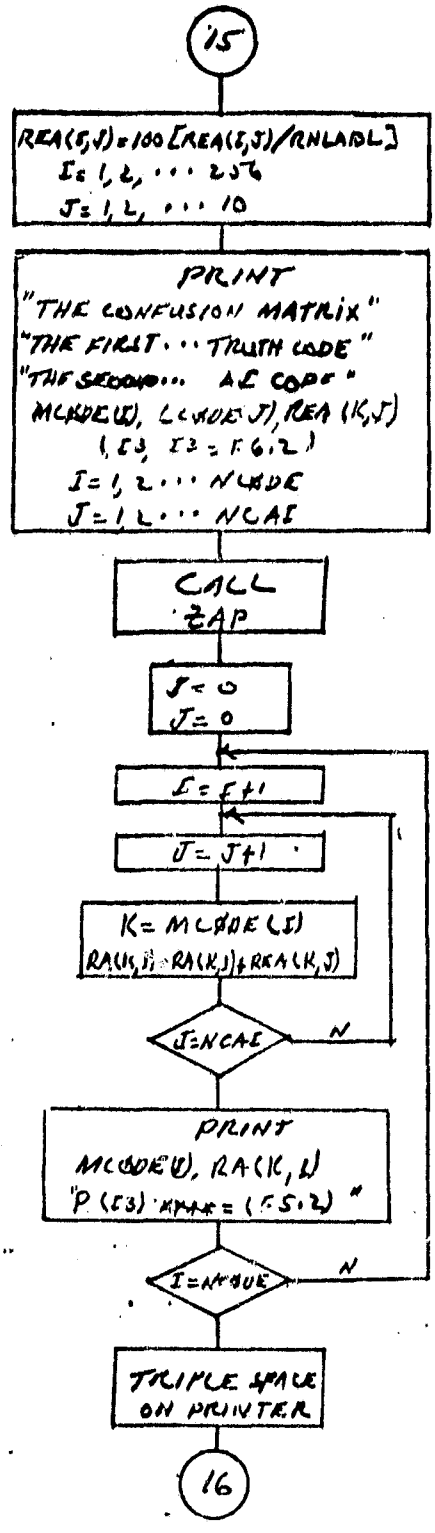

9
CALL
ZAP
I=0
J=0
I=I+1
J=J+1
AI(I,J)=0
4
L=GP(I,J)/10
S=GP(I,J)-L*10
RA(L,S)=RA(L,S)+1
N
J=19
4
I=11
4
PRINT
"GROUND TRUTH... THE AI DOTS"
RTT=NAID
CALL
TAB
(NNNNN,
IRNUM,
RTT, FLG)
PRINT
"PERCENTAGE .... PURE PIXELS"
10

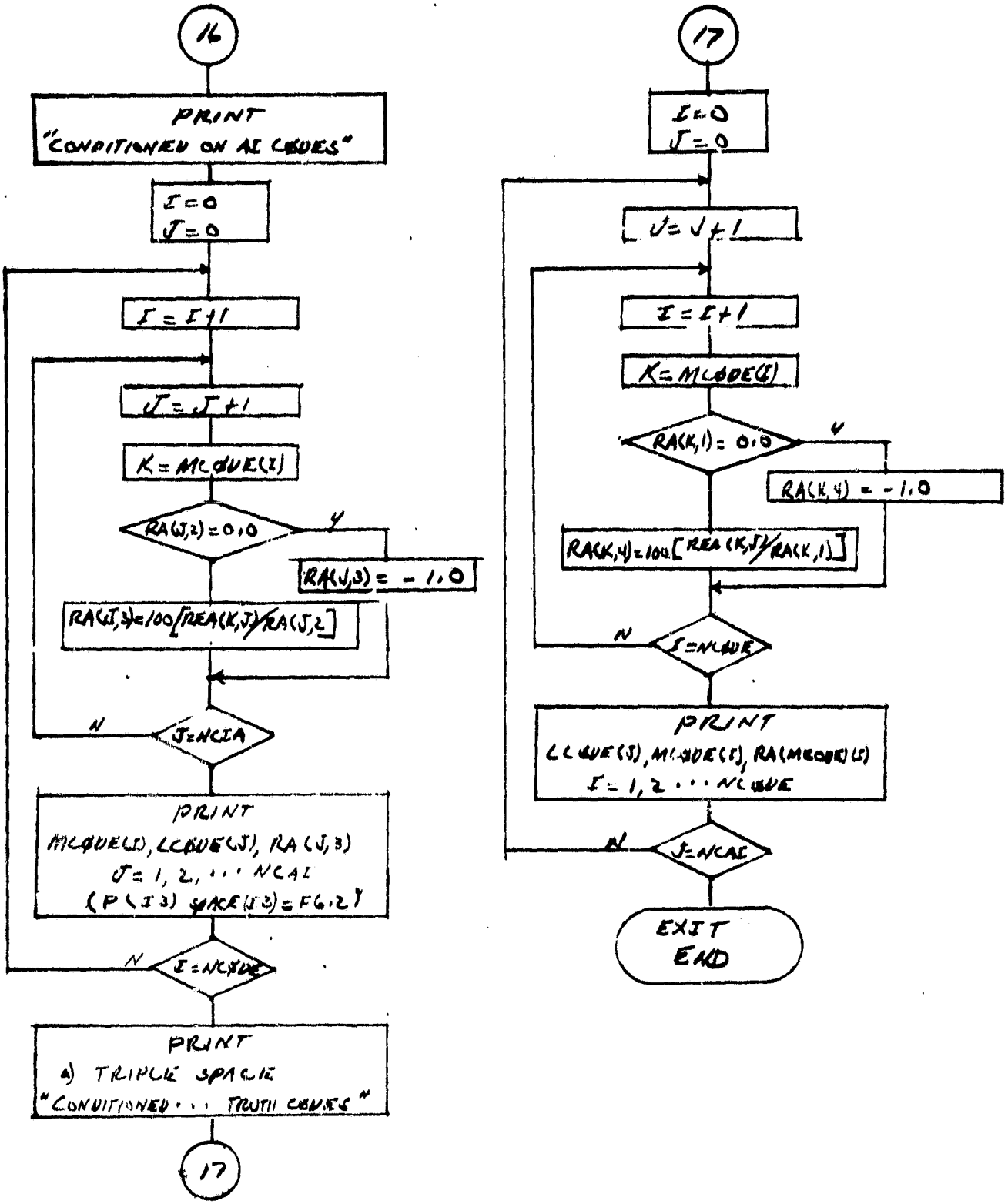





12
PRINT
"GROUND . . . . . THE AI DOTS"
RTT=NAID
CALL
TAB
(NDUMY, RPURE,
RTT, FLG)
PRINT
"PERCENTAGE . . . . PURE PIXELS"
NB=-1
PRINT
"MISLABELED DOTS"
PRINT
"LAKE . . SAMPLE TRUE ANALYSIS"
ERROR=0
NLABL=0
KL=0
KP=0
KL=KL+1
KP=KP+1
AA=AI(KL,KP)
AA=0
NLABL=NLABL+1
GG=GT(KL,KP)
13
13
GG>1000
Y
GG=GG-1000
REA(GG,AA)=REA(GG,AA)+1
GG=AA
Y
ERROR=ERROR+1
PRINT
KL, KP, GG, AA
(4 I 10)
N
KP=19
N
KL=11
RERROR=ERROR
RNLABL=NLABL
RPMC=100(RERROR/RNLABL)
NB=-1
N
PRINT
"PROPORTION OF . . .
. . MNLABELED=IF6.3"
PRINT
"COMPARISON OF . . . . .
. . . AND THE AI DOTS"
14






16
PRINT
"CONDITIONED ON AI CODES"
I=0
J=0
I=I+1
J=J+1
K=MCODE(I)
RA(J,2)=0.0
Y
RA(J,3)=-1.0
RA(J,3)=100[REA(K,J)/RA(J,2)]
N
J=NCIA
PRINT
MCODE(I), LCODE(J), RA(J,3)
J=1, 2, ... NCAI
(P(I3) SPACE(13)=F6.2)
N
I=NCODE
PRINT
4) TRIPLE SPACE
"CONDITIONED ... TRUTH CODES"
17
17
I=0
J=0
J=J+1
I=I+1
K=MCODE(I)
RA(K,1)=0.0
Y
RA(K,4)=-1.0
RA(K,4)=100[REA(K,J)/RA(K,1)]
N
I=NCODE
PRINT
LCODE(J), MCODE(I), RA(MCODE(I))
I=1, 2 ... NCODE
N
J=NCAI
EXIT
END

3.2.15.8 Listing

```
      IMPLICIT INTEGER (A-Q),(S-Z)
      BYTE BUF(3060),T(8),D(9),KS
      COMMON /BK1/RA(256,6),MCODE(256)
      DIMENSION          GT(11,19),      KT(26),AI(11,19),JT(256),
     *REA(256,10),MT(6)
      DIMENSION GP(11,19)
      DIMENSION LCODE(10)
      EQUIVALENCE (S,BUF(67))
      CALL TIME(T)
      CALL DATE(D)
      NRDD=1
         NRDR=2
      NPRT=6
      WRITE(NPRT,703) D,T
         OPEN(UNIT=NRDR,NAME='AI.DAT',TYPE='OLD',
     *   ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
         OPEN(UNIT=NRDD,NAME='SPATL.DAT',TYPE='OLD',
     *   ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
  703 FORMAT(1H1,'  JOB INITIATED ON  ',9A1,'  AT  ',8A1,//,10X,
     1'PROGRAM  SPATL.FTN')
      READ(NRDD,704) GSDEV,GNDEV,F1
  704 FORMAT(A1,1X,2I2)
      WRITE(NPRT,705) GSDEV,GNDEV,F1
  705 FORMAT(//,10X,'GROUND TRUTH TAPE',//,10X,A1,'T',10X,'DEVICE NO.=',
     1I5,10X,'FILE NO.=',I5)
      GIDEV=0
      IF(GSDEV.EQ.'X') GIDEV=1
      IF(GNDEV.NE.0.AND.GNDEV.NE.1) GO TO 400
      F1=F1-1
      CALL  TINIT(3,GIDEV,GNDEV)
      CALL  TATCH(3)
      CALL  TRWD(3)
      CALL  TWAIT(3)
      CALL  TFILE(3,F1)
      CALL  TWAIT(3)
      CALL  TREAD(3,BUF,1530)
      CALL  TWAIT(3)
      DO 20 M=1,256
      JT(M)=0
   20 CONTINUE
      CALL SWAB(S)
      WRITE(NPRT,306) S,(BUF(IB),IB=61,63)
  306 FORMAT('  SITE=  ',I5,5X,'DAY=',I5,5X,'MON=',I5,5X,'YEAR=',I5)
      CALL ZAP
      WRITE (NPRT,905)
  905 FORMAT(//,10X,'CODE TO CODE TRANSFORMATION',//,8X,'BEGIN',7X,
     1'END',7X,'CODE')
  121 CONTINUE
      READ(NRDD,118) NB,NE,NO
  118 FORMAT(3I5)
      WRITE(NPRT,117) NB,NE,NO
  117 FORMAT(1H ,3I10)
      IF((NB.EQ.0).AND.(NE.EQ.0).AND.(NO.EQ.0)) GO TO 122
      IF((NB.EQ.0).AND.(NE.EQ.0).AND.(NO.EQ.-1)) GO TO 224
      DO 119 N=NB,NE
      JT(N)=NO
  119 CONTINUE
      GO TO 121
  224 CONTINUE
      DO 225 I=1,256
```

```
  122 CONTINUE
      DO 123 I=1,256
      DO 124 J=1,10
      REA(I,J)=0.0
  124 CONTINUE
  123 CONTINUE
      RT=0.0
      DO 1 L=1,117
      L10=MOD(L,10)
      A=-539
      DO 2 SL=1,3
      A=A+540
      CALL  TREAD(3,BUF(A),270)
      CALL  TWAIT(3)
    2 CONTINUE
      S1=-470
      DO 3 S=1,196
      S10=MOD(S,10)
      S1=S1+2
      S2=S1
      P=0
      DO 4 SS=1,2
      S2=S2+1
      S3=S2
      DO 5 SL=1,3
      S3=S3+540
      M=BUF(S3)
        M = M+128
      IF(M.NE.0) RT=RT+1.0
      IF(M.EQ.0) M=256
      M=JT(M)
      P=P+1
      MT(P)=M
    5 CONTINUE
    4 CONTINUE
      CALL CROPL(CROP,MT,NC)
        IF(NC.GT.6) WRITE(NPRT,880) L,S,NC
  880   FORMAT (1HO,10X,'LINE=',I5,5X,'SAMPLE=',I5,5X,'NO. OF SUBPIXELS='
     *,I5)
        IF(NC.GT.6) WRITE (NPRT,881) (MT(I),I=1,6)
        IF(NC.GT.6) STOP
  881   FORMAT(1HO,10X,'MT=',5X,6I5)
      DO 6 P1=1,6
      M=MT(P1)
      IF(M.EQ.1000) GO TO 6
      N=0
      DO 7 P2=P1,6
      IF(MT(P2).NE.M) GO TO 7
      N=N+1
      MT(P2)=1000
    7 CONTINUE
      RA(M,N)=RA(M,N)+1
    6 CONTINUE
      N=NC
      IF((L10.NE.0).OR.(S10.NE.0)) GO TO 8
      GP(L/10,S/10)=CROP*10+N
      IF(N.NE.6) CROP=CROP+1000
      GT(L/10,S/10)=CROP
    8 CONTINUE
    3 CONTINUE
    1 CONTINUE
      WRITE(NPRT,850)
  850 FORMAT(//,10X,'GROUND TRUTH INFORMATION FOR THE WHOLE SEGMENT')
      RCHECK=117.0*196.0*6.0
```


3-27-93

```
      CALL TAB(NCODE,RPURE,RTT,FLG)
      IF(RT.NE.RCHECK) WRITE(NPRT,223)
  223 FORMAT(//,10X,'THE WHOLE SEGMENT WAS NOT GROUND TRUTHED!)
      WRITE(NPRT,222) RT
  222 FORMAT(//,10X,'COMPUTATIONS BASED ON ',F10.2,' SUBPIXELS',/)
      WRITE(NPRT,102) RPURE
  102 FORMAT('  PERCENTAGE OF SCENE IN PURE PIXELS',1F10.2)
      WRITE(NPRT,903)
  903 FORMAT(//,10X,'A MATRIX OF GROUND TRUTH DET LABELS')
      WRITE(NPRT,103)((GT(L,S),S=1,19),L=1,11)
  103 FORMAT(1H ,1916)
      CALL EAP
      DO 200 I=1,11
      DO 200 J=1,19
      L=GP(I,J)/10
      S=GP(I,J)-L*10
      RA(L,S)=RA(L,S)+1
  200 CONTINUE
      WRITE(NPRT,851)
  851 FORMAT(//,10X,'GROUND TRUTH INFORMATION FOR THE 209 DOTS')
      RTT=209
      FLG=2
      CALL TAB(NDUMY,RPURE,RTT,FLG)
      WRITE(NPRT,255) RPURE
  255 FORMAT(/,10X,'PERCENTAGE OF THE 209 DOTS WHICH ARE PURE PIXELS',
     1F10.2)
      CALL  TRWD(3)
      CALL  TWAIT(3)
      DO 16 KL=1,11
      DO 17 KP=1,19
      AI(KL,KP)=0
   17 CONTINUE
   16 CONTINUE
      DO 14 I=1,26
      KT(I)=0
   14 CONTINUE
      NCAI=0
      DO 666 I=1,10
  666 LCODE(I)=0
      WRITE(NPRT,702)
  702 FORMAT(/,10X,'TYPE TO CODE TRANSFORMATION')
      WRITE(NPRT,300)
  300 FORMAT(/,3X,'TYPE',6X,'CODE')
      NAID=0
   13 CONTINUE
      READ(NRDD,108) KS,NT
  108 FORMAT(1A1,4X,I15)
      WRITE(NPRT,105) KS,NT
  105 FORMAT(1H , 5X,A1,I10)
      IF(NT.EQ.0) GO TO 400
      IF(KS.EQ.88) GO TO 12
      KT(KS-64)=NT
      LCODE(NT)=NT
      GO TO 13
   12 CONTINUE
      CALL CLOSE(NRDD)
      DO 777 I=1,10
  777 IF(LCODE(I).GT.0) NCAI=NCAI+1
  212    CONTINUE
      READ(NRDR,106) KL,KP,KS
  106 FORMAT(10X,I12,1X,I12,1X,1A1)
      IF(KL.EQ.0) GO TO 15
      NAID=NAID+1
      NT=KT(KS-64)
```

```
 15 CONTINUE
    WRITE(NPRT,502) NAID
502 FORMAT(/,10X,'NO. OF AI DOTS=',I5)
    WRITE(NPRT,902)
902 FORMAT(//,10X,'A MATRIX OF AI DOT LABELS')
    WRITE(NPRT,103) ((AI(L,S),S=1,19),L=1,11)
    CALL ZAP
    DO 201 I=1, 1
    DO 201 J=1,19
    IF(AI(I,J).EQ.0) GO TO 201
    L=GP(I,J)/10
    S=GP(I,J)-L*10
    RA(L,S)=RA(L,S)+1
201 CONTINUE
    WRITE(NPRT,852)
852 FORMAT(//,10X,'GROUND TRUTH INFORMATION FOR THE AI DOTS')
    RTT=NAID
    CALL TAB(NDUMY,RPURE,RTT,FLG)
    WRITE(NPRT,256)  RPURE
256 FORMAT(//,10X,'PERCENTAGE OF THE AI DOTS WHICH ARE PURE PIXELS',
   1F10.2)
       IF(NO.NE.-1) WRITE(NPRT,901)
901 FORMAT(//,10X,'MISLABELED DOTS')
    WRITE(NPRT,102)
107 FORMAT('        LINE      SAMPLE      TRUE      ANALYIST')
    ERROR=0
    NLABL=0
    DO 18 KL=1,11
    DO 19 KP=1,19
    AA=AI(KL,KP)
    IF(AA.EQ.0) GO TO 19
    NLABL=NLABL+1
    GG=GT(KL,KP)
    IF(GG.GE.1000) GG=GG-1000
    REA(GG,AA)=REA(GG,AA)+1.0
    IF(GG.EQ.AA) GO TO 22
    ERROR=ERROR+1
    WRITE(NPRT,110) KL,KP,GG,AA
110 FORMAT(1H ,4I10)
 22 CONTINUE
 19 CONTINUE
 18 CONTINUE
    RERROR=ERROR
    RNLABL=NLABL
    RPMC=100.0*RERROR/RNLABL
       IF(NO.NE.-1) WRITE (NPRT,109) RPMC
109 FORMAT('  PROPORTION OF DOTS MISLABELED=',1F6.2)
    WRITE(NPRT,951)
951 FORMAT(//,10X,'COMPARISON OF THE GROUND TRUTH AND THE AI DOTS')
    DO  800 L=1,11
    WRITE(NPRT,103) (GP(L,S),S=1,19)
    WRITE(NPRT,801) (AI(L,S) ,S=1,19)
801 FORMAT(1H ,19(I5,1X),/)
800 CONTINUE
    WRITE(NPRT,301)
301 FORMAT(/,10X,'MIXED PIXELS')
    WRITE(NPRT,107)
    DO 302 I=1,11
    DO 302 J=1,19
    IF(AI(I,J).EQ.0) GO TO 302
    N=GP(I,J)-GP(I,J)/10*10
    IF(N.EQ.6) GO TO 302
    WRITE(NPRT,110) I,J,GP(I,J),AI(I,J)
302 CONTINUE
    DO 125 I=1,256
```

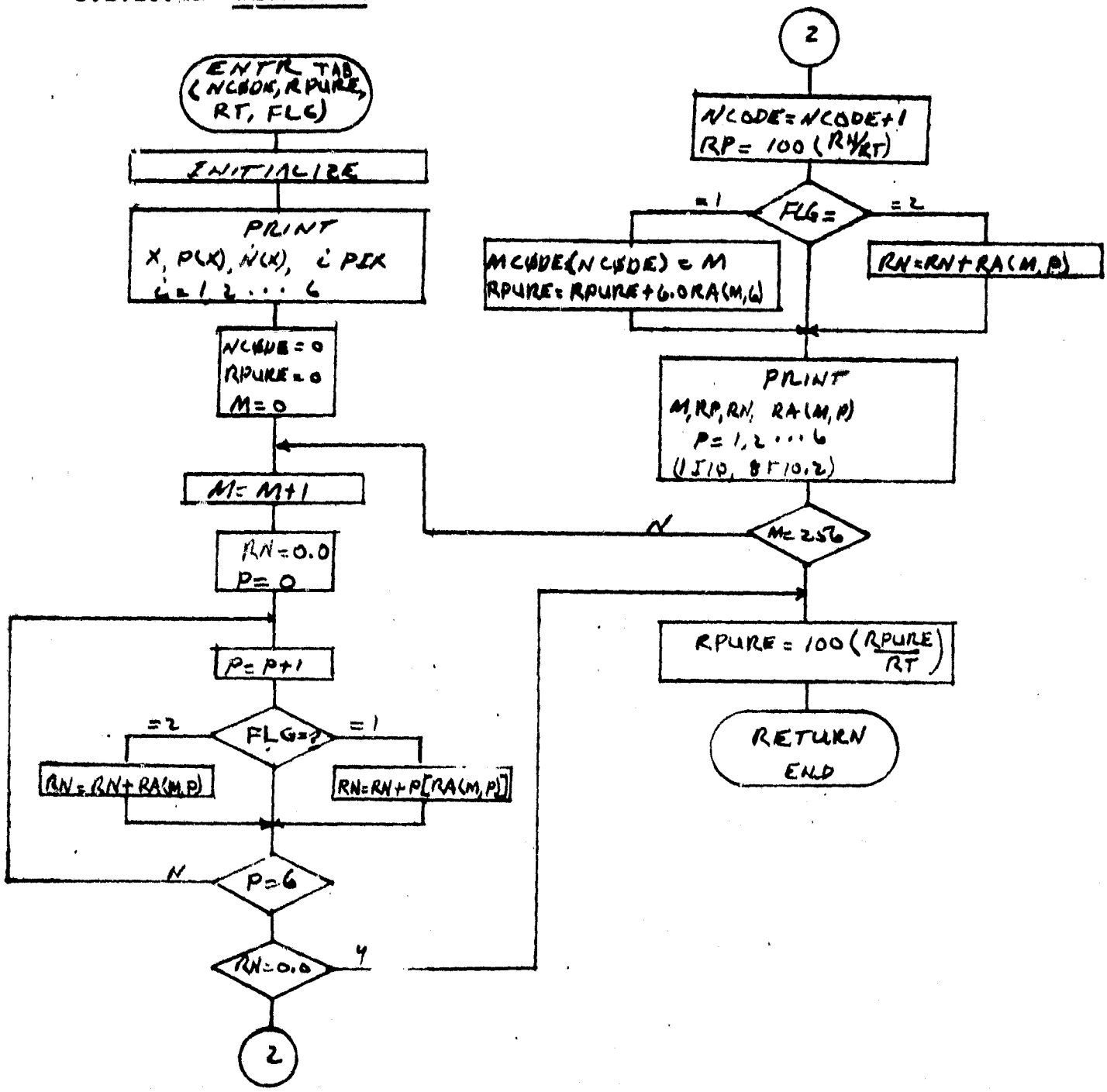
```
      REA(I,J)=100.0*REA(I,J)/RNLABL
  126 CONTINUE
  125 CONTINUE
      WRITE(NPRT,960)
  960 FORMAT(//,10X,'THE CONFUSION MATRIX')
      WRITE(NPRT,965)
  965 FORMAT(/,10X,'THE FIRST INDEX IS THE GROUND TRUTH CODE',
     1/,10X,'THE SECOND INDEX IS THE AI CODE')
      DO 910 I=1,NCODE
         K=MCODE(I)
      WRITE(NPRT,911) (MCODE(I),LCODE(J),REA(K,J),J=1,NCAI )
  911 FORMAT(/,5( 5X,'P(',I3,',',I3,')=',F6.2))
  910 CONTINUE
      WRITE(NPRT,999)
  999 FORMAT(//)
      CALL ZAP
      DO 988 I=1,NCODE
      DO 966 J=1,NCAI
         K=MCODE(I)
      RA(K,1)=RA(K,1)+REA(K,J)
  966 CONTINUE
      WRITE(NPRT,967) MCODE(I),RA(K,1)
  967 FORMAT(/,10X,'P(',I3,',***)=',F5.2)
  988 CONTINUE
      WRITE(NPRT,999)
      DO 968 J=1,NCAI
      DO 969 I=1,NCODE
         K=MCODE(I)
      RA(J,2)=RA(J,2)+REA(K,J)
  969 CONTINUE
      WRITE(NPRT,970) LCODE(J),RA(J,2)
  970 FORMAT(/,10X,'P(***,',I3,')=',F5.2)
  968 CONTINUE
      WRITE(NPRT,999)
      WRITE(NPRT,974)
  974 FORMAT(/,10X,'CONDITIONED ON AI CODES')
      DO 971 I=1,NCODE
      DO 972 J=1,NCAI
         K=MCODE(I)
         IF(RA(J,2).EQ.0.0) RA(J,3)=-1.0
         IF(RA(J,2).EQ.0.0) GO TO 972
      RA(J,3)=REA(K,J)/RA(J,2)*100.0
  972 CONTINUE
      WRITE(NPRT,973) (MCODE(I),LCODE(J),RA(J,3),J=1,NCAI )
  973 FORMAT(/,5( 5X,'P(',I3,'/',I3,')=',F6.2))
  971 CONTINUE
      WRITE(NPRT,999)
      WRITE(NPRT,975)
  975 FORMAT(/,10X,'CONDITIONED ON GROUND TRUTH CODES')
      DO 976 J=1,NCAI
      DO 977 I=1,NCODE
         K=MCODE(I)
         IF(RA(K,1).EQ.0.0) RA(K,4)=-1.0
         IF(RA(K,1).EQ.0.0) GO TO 977
      RA(K,4)=REA(K,J)/RA(K,1)*100.0
  977 CONTINUE
      WRITE(NPRT,973) (LCODE(J),MCODE(I),RA(MCODE(I),4),I=1,NCODE)
  976 CONTINUE
  400 CONTINUE
      WRITE(NPRT,999)
      CALL DATE(D)
      CALL TIME(T)
      CALL CLOSE(NRDD)
         CALL CLOSE(NRDR)
      WRITE(NPRT,104) D,T
```

3.2.16 SPATL SUBROUTINES

Two special subroutines TAB and ZAP are called directly by SPATL. Communication between these and SPATL is through the common BK1.

3.2.16.1 Subroutine TAB

3.2.16.1a Flowchart

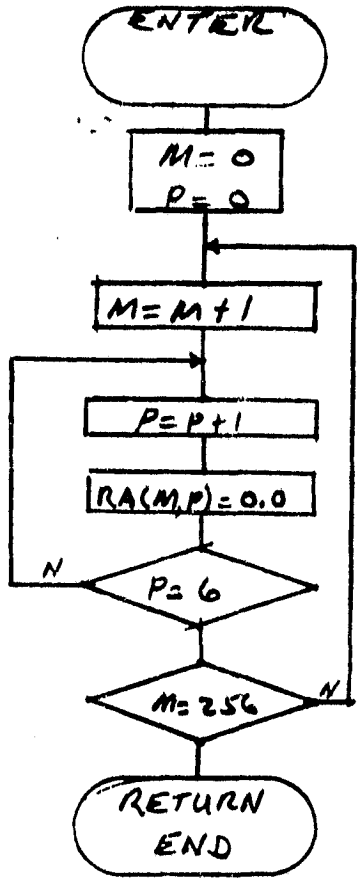


3.2.15.1b Listing

```
      SUBROUTINE TAB(NCODE,RPURE,RT,FLG)
      IMPLICIT INTEGER (A-Q),(S-Z)
      COMMON /BK1/RA(256,6),MCODE(256)
      NPRT=6
      WRITE(NPRT,950) (II,II=1,6)
  950 FORMAT(//,9X,'X',6X,'F(X)',6X,'N(X)',6(6X,'P(X',I1,'))
      NCODE=0
      RPURE=0.0
      DO 9 M=1,256
      RN=0.0
      DO 10 P=1,6
      IF(FLG.EQ.1) RN=RN+P*RA(M,P)
      IF(FLG.EQ.2) RN=RN+RA(M,P)
   10 CONTINUE
      IF(RN.EQ.0.0) GO TO 9
      NCODE=NCODE+1
      IF(FLG.EQ.1) MCODE(NCODE)=M
      RP=100.0*RN/RT
      IF(FLG.EQ.1) RPURE=RPURE+6.0*RA(M,6)
      IF(FLG.EQ.2) RPURE=RPURE+    RA(M,6)
      WRITE(NPRT,101) M,RP,RN,(RA(M,P),P=1,6)
  101 FORMAT(1H ,I10,8F10.2)
    9 CONTINUE
      RPURE=100.0*RPURE/RT
      RETURN
      END
```

3.2.16.2 Subroutine ZAP

3.2.16.2a Flowchart



3.2.16.2b Listing

```
      SUBROUTINE ZAP
      IMPLICIT INTEGER (A-Q),(S-Z)
      COMMON  /BK1/RA(256,6),MCODE(256)
      DO 20 M=1,256
      DO 21 P=1,6
      RA(M,P)=0.0
   21 CONTINUE
   20 CONTINUE
      RETURN
      END
```

3.2.17 SECOND MODULE - SECOND UNIT (ALLCRP)

3.2.17.1 Linkage

ALLCRP is a stand alone program which uses only standard system utility routines.

3.2.17.2 Interface

None

3.2.17.3 Input

ALLCRP requires input of a "ground truth" magnetic tape of file product of an earlier execution of the first module of this system. It also requires input of a companion DTRM tape (reference 1) and card entries of corresponding analyst "dot" labeling data and a crop to small grain transformation (see Appendix A).

3.2.17.4 Output

Printout of accuracy assessment parameters (see Appendix B) including:

1. Maximum likelihood proportion estimate
2. Classification and pixel counting proportion estimate
3. Probability of misclassification
4. Variance of Procedure 1 proportion estimate

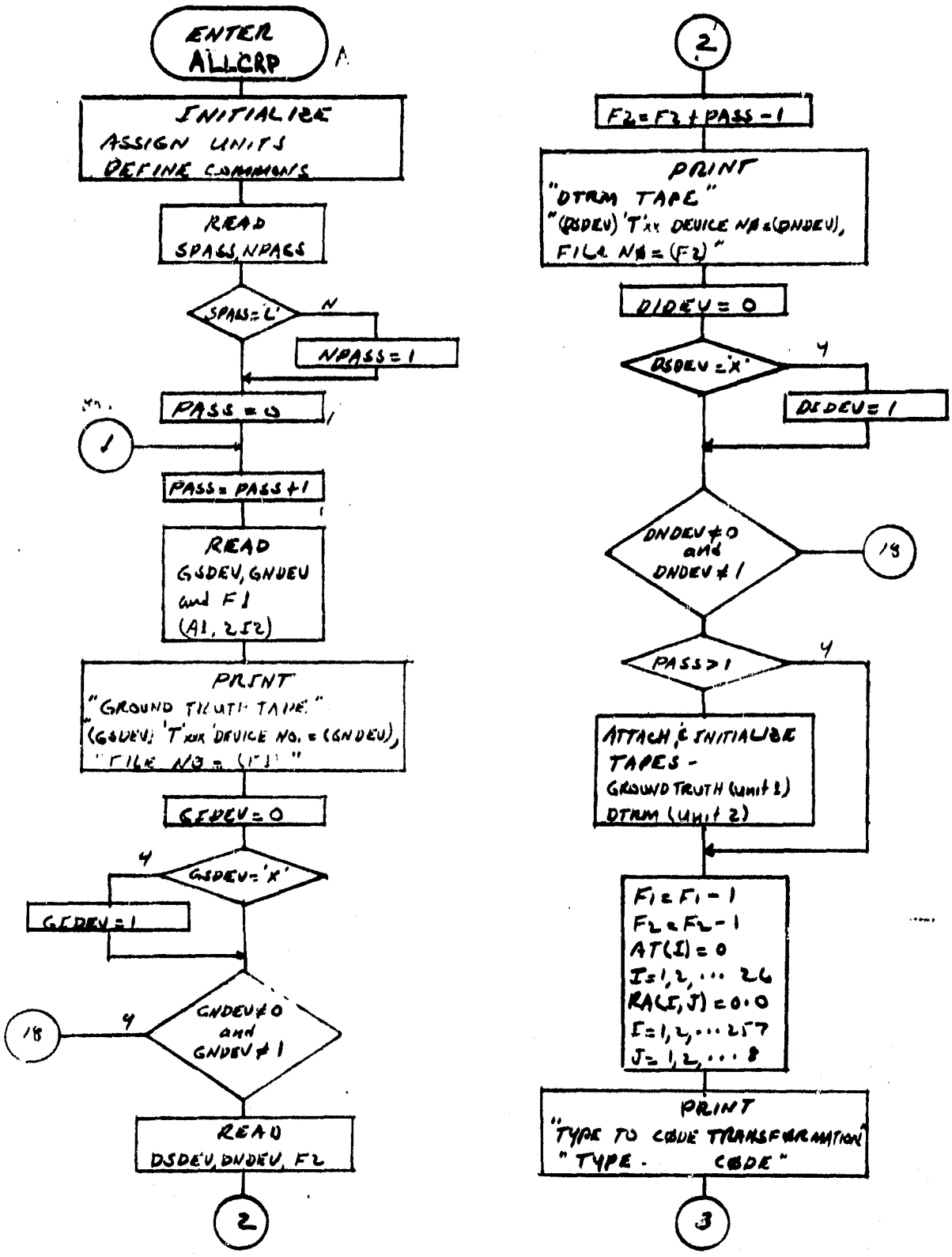
3.2.17.5 Storage

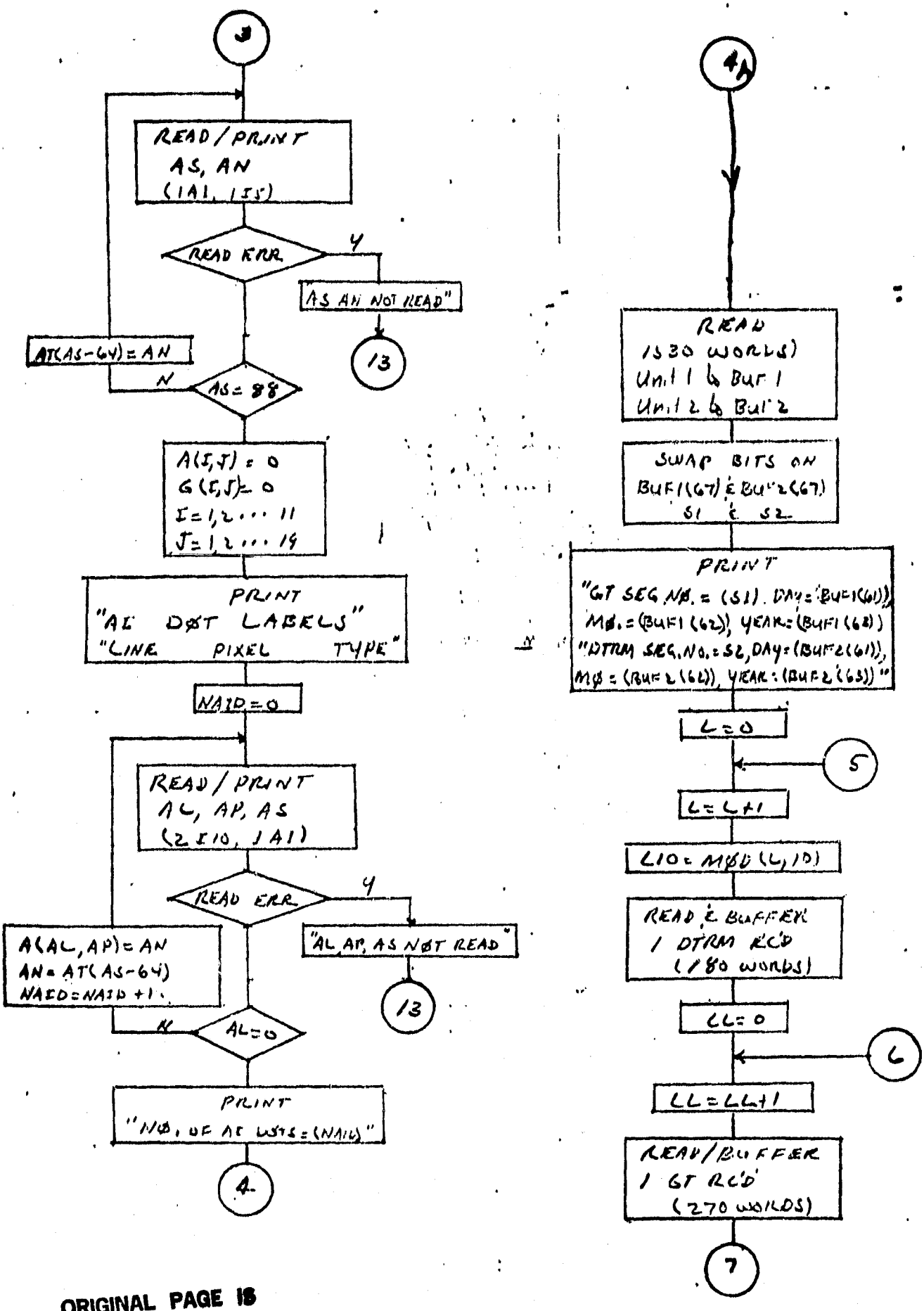
TBD

3.2.17.6 Description

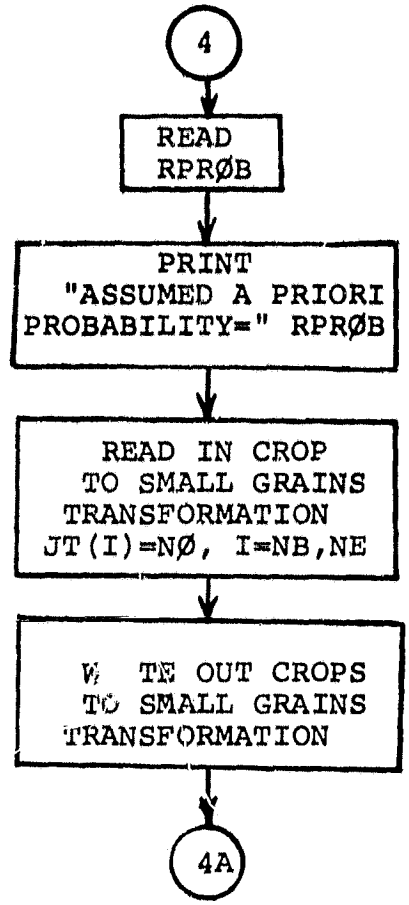
ALLCRP compares as small grains and other ground truth data with operational classification data, analyst dot labeling and ERIPS automatic labeled data (DTRM) to determine their accuracy.

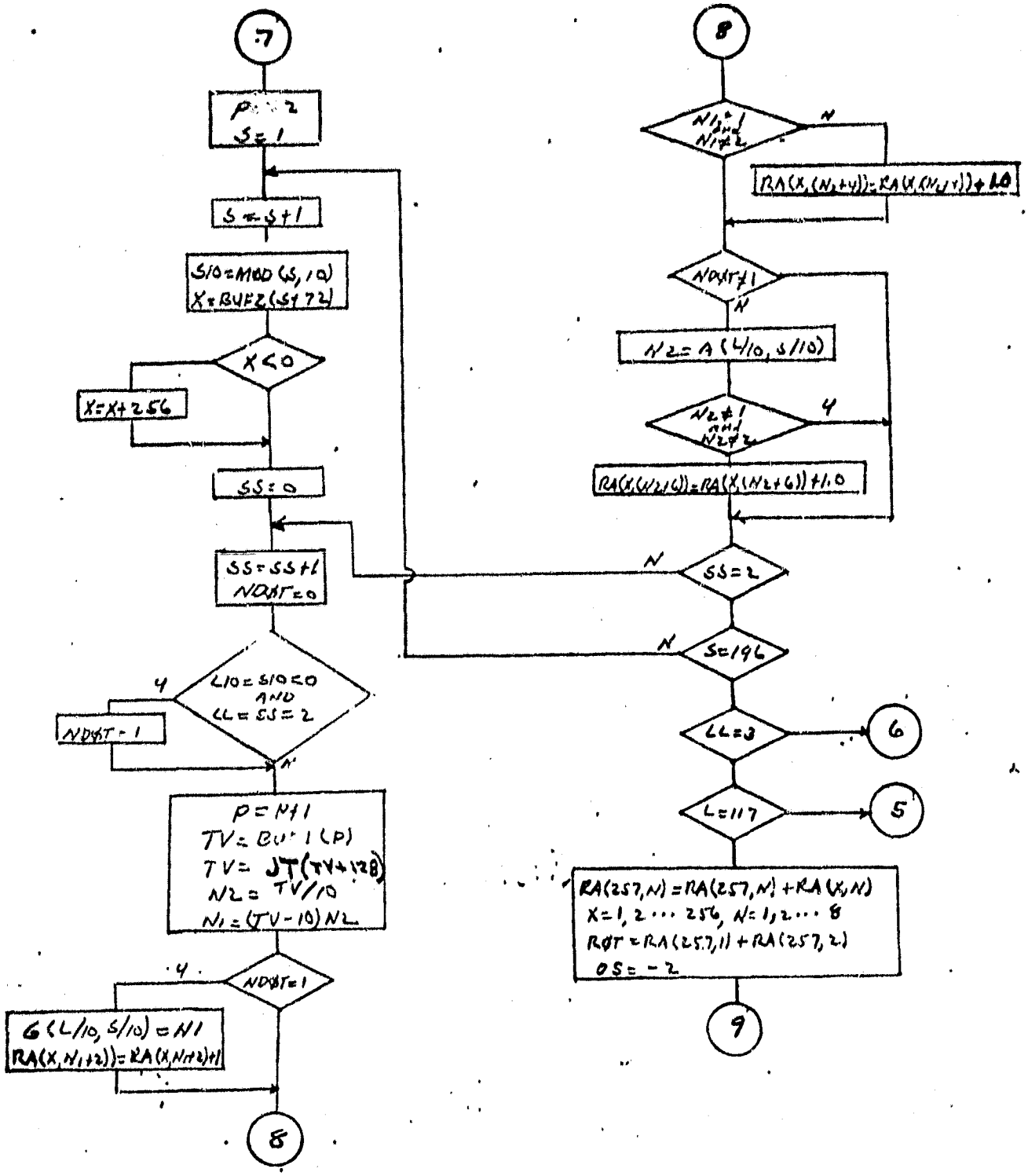
3.2.17.7 Flowchart

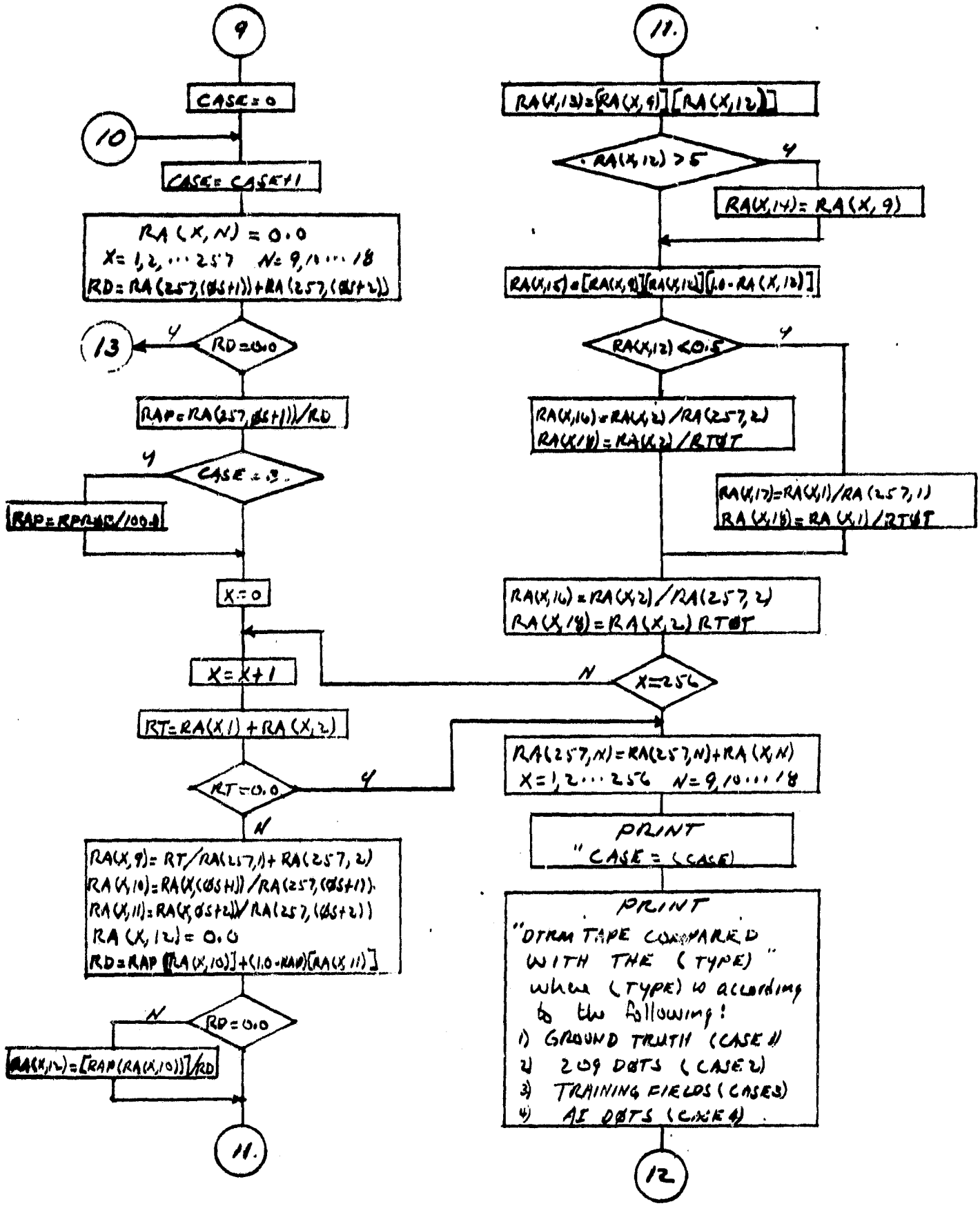




ORIGINAL PAGE IS
OF POOR QUALITY


4
READ
RPRØB
PRINT
"ASSUMED A PRIORI
PROBABILITY=" RPRØB
READ IN CROP
TO SMALL GRAINS
TRANSFORMATION
JT(I)=NØ, I=NB,NE
W TE OUT CROPS
TO SMALL GRAINS
TRANSFORMATION
4A


7
P = 2
S = 1
S = S + 1
S10 = MOD (S, 10)
X = BUFZ (S + 72)
X < 0
X = X + 256
SS = 0
SS = SS + 1
NDAT = 0
L10 = S10 = 0 AND LL = SS = 2
NDAT = 1
P = N1 + 1
TV = BUF1 (P)
TV = JT (TV + 128)
N2 = TV / 10
N1 = (TV - 10) N2
NDAT = 1
G (L/10, S/10) = N1
RA (X, N1 + 2) = RA (X, N1 + 2) + 1
8
N1 = 1 AND N1 = 2
RA (X, (N1 + 4)) = RA (X, (N1 + 1)) + 1.0
NDAT = 1
N2 = A (L/10, S/10)
N2 = 1 AND N2 = 2
RA (X, (N2 + 6)) = RA (X, (N2 + 6)) + 1.0
SS = 2
S = 196
LL = 3
6
L = 117
5
RA (257, N) = RA (257, N) + RA (X, N)
X = 1, 2 ... 256, N = 1, 2 ... 8
ROT = RA (257, 1) + RA (257, 2)
OS = - 2
9


9
CASE = 0
10
CASE = CASE + 1
RA(X,N) = 0.0
X = 1, 2, ··· 257 N = 9, 10 ··· 18
RD = RA(257,(BS+1)) + RA(257,(BS+2))
13
Y
RD = 0.0
RAP = RA(257,(BS+1))/RD
Y
CASE = 3
RAP = RAP(RAP/100.0)
X = 0
X = X + 1
RT = RA(X,1) + RA(X,2)
RT = 0.0
Y
N
RA(X,9) = RT/RA(257,1) + RA(257,2)
RA(X,10) = RA(X,(BS+1))/RA(257,(BS+1))
RA(X,11) = RA(X,(BS+2))/RA(257,(BS+2))
RA(X,12) = 0.0
RD = RAP[RA(X,10)] + (1.0 - RAP)[RA(X,11)]
N
RD = 0.0
RA(X,12) = [RAP(RA(X,10))]/RD
11.
11.
RA(X,13) = [RA(X,9)][RA(X,12)]
RA(X,12) > 5
Y
RA(X,14) = RA(X,9)
RA(X,15) = [RA(X,9)][RA(X,14)][1.0 - RA(X,12)]
RA(X,12) < 0.5
Y
RA(X,16) = RA(X,2)/RA(257,2)
RA(X,18) = RA(X,2)/RTOT
RA(X,17) = RA(X,1)/RA(257,1)
RA(X,18) = RA(X,1)/RTOT
RA(X,16) = RA(X,2)/RA(257,2)
RA(X,18) = RA(X,2) RTOT
N
X = 256
RA(257,N) = RA(257,N) + RA(X,N)
X = 1, 2 ··· 256 N = 9, 10 ··· 18
PRINT
" CASE = (CASE)
PRINT
"DTRM TAPE COMPARED
WITH THE (TYPE)"
WHERE (TYPE) IS ACCORDING
TO THE FOLLOWING:
1) GROUND TRUTH (CASE 1)
2) 209 DOTS (CASE 2)
3) TRAINING FIELDS (CASE 3)
4) AI DOTS (CASE 4)
12

3.2.17.8 - Listing

```
      IMPLICIT INTEGER (A-Q),(S-Z)
         DIMENSION JT(256)
      DIMENSION AT(26),A(11,19),          RA(257,18)
      BYTE BUF1(3060),BUF2(3060),T(8),D(9)
      BYTE AS
      BYTE CHAR(50)
      EQUIVALENCE (S1,BUF1(67)),(S2,BUF2(67))
      COMMON /GG/G(11,19)
      COMMON /DD/DDTR(3,38)
      COMMON /STATUS/W1,W2
      CALL TIME(T)
      CALL DATE(D)
         NRDR = 4
         NRDD=5
      NPRT=6
      WRITE(NPRT,703) D,T
  703 FORMAT(1H1,'  JOB INITIATED ON  ',9A1,'  AT  ',8A1,//,10X,
     1'PROGRAM ALLCRP,FTN')
         OPEN(UNIT=NRDD,NAME='LOOP,DAT',TYPE='OLD',
     *  ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
         READ(NRDD,801) SPASS,NPASS
  801    FORMAT(A1,4X,I5)
         CALL CLOSE(NRDD)
         IF(SPASS.NE.'L') NPASS=1
         DO 802 PASS=1,NPASS
         OPEN(UNIT=NRDD,NAME='ALLCRP,DAT',TYPE='OLD',
     *  ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
      READ(NRDD,704) GSDEV,GNDEV,F1
  704 FORMAT(A1,1X,2I2)
      WRITE(NPRT,705) GSDEV,GNDEV,F1
  705 FORMAT(//,10X,'GROUND TRUTH TAPE',//,10X,A1,'T',10X,'DEVICE NO.=',
     1I5,10X,'FILE NO.=',I5)
      GIDEV=0
      IF(GSDEV.EQ.'X') GIDEV=1
      IF(GNDEV.NE.0.AND.GNDEV.NE.1) GO TO 18
      READ(NRDD,704) DSDEV,DNDEV,F2
         F2=F2+PASS-1
      WRITE(NPRT,707) DSDEV,DNDEV,F2
  707 FORMAT(//,10X,' DTRM TAPE       ',//,10X,A1,'T',10X,'DEVICE NO.=',
     1I5,10X,'FILE NO.=',I5)
      DIDEV=0
      IF(DSDEV.EQ.'X') DIDEV=1
      IF(DNDEV.NE.0.AND.DNDEV.NE.1) GO TO 18
         OPEN(UNIT=NRDR,NAME='AI,DAT',TYPE='OLD',ACCESS='SEQUENTIAL',
     *  FORM='FORMATTED',CARRIAGE CONTROL='NONE')
         IF(PASS.GT.1) GO TO 888
      CALL  TINIT(1,GIDEV,GNDEV)
      CALL  TINIT(2,DIDEV,DNDEV)
      CALL  TATCH(1)
      CALL  TATCH(2)
  888    CONTINUE
      CALL  TRWD(1)
      CALL  TWAIT(1)
      CALL  TRWD(2)
      CALL  TWAIT(2)
      F1=F1-1
      F2=F2-1
C
      DO 1 I=1,26
      AT(I)=0
    1 CONTINUE
      DO 3 I=1,257
```

```
      DØ 4 J=1,8
      RA(I,J)=0.0
    4 CØNTINUE
    3 CØNTINUE
      WRITE(NPRT,702)
  702 FØRMAT(/,10X,'TYPE TØ CØDE TRANSFØRMATIØN')
      WRITE(NPRT,300)
  300 FØRMAT(/,3X,'TYPE',6X,'CØDE')
    5 CØNTINUE
      READ(NRDD,101,ERR=927) AS,AN
  101 FØRMAT(1A1,4X,I10)
      WRITE(NPRT,102) AS,AN
  102 FØRMAT(1H ,5X,A1,I10)
      IF(AS.EQ.88) GØ TØ 6
      AT(AS-64)=AN
      GØ TØ 5
    6 CØNTINUE
        DØ 36 M=1,256
        JT(M)=0
   36     CØNTINUE
      DØ 7 I=1,11
      DØ 8 J=1,19
      A(I,J)=0
      G(I,J)=0
    8 CØNTINUE
    7 CØNTINUE
      WRITE(NPRT,701)
  701 FØRMAT(/,10X,'AI DØT LABLES')
      WRITE(NPRT,301)
  301 FØRMAT(/,7X,'LINE',6X,'PIXEL',7X,'TYPE')
      NAID=0
    9 CØNTINUE
      READ(NRDR,103,ERR=929)  AL,AP,AS,(CHAR(I),I=1,50)
  103 FØRMAT(10X,I12,1X,I12,1X,1A1,13X,50A1)
      WRITE(NPRT,104) AL,AP,AS,(CHAR(I),I=1,50)
  104 FØRMAT(1H ,2I10,9X,1A1 ,13X,50A1)
      IF(AL.EQ.0) GØ TØ 10
      NAID=NAID+1
      AN=AT(AS-64)
      IF(A(AL,AP).NE.0) WRITE(NPRT,313) AL,AP
  313 FØRMAT(///,10X,'DUPLICATE DØT LABEL FØR DØT = ',2I5,///)
      A(AL,AP)=AN
      GØ TØ 9
   10 CØNTINUE
      WRITE(NPRT,502) NAID
  502 FØRMAT(/,10X,'NØ. ØF AI DØTS=',I5)
      READ(NRDD,107) RPRØB
  107 FØRMAT(1F10.2)
      WRITE(NPRT,108) RPRØB
  108 FØRMAT('  ASSUMED A PRIØRI PRØBABILITY=  ',1F10.2)
      WRITE (NPRT,905)
  905 FØRMAT(//,10X,'CØDE TØ CØDE TRANSFØRMATIØN',//,8X,'BEGIN',7X,
     1'END' ,7X,'CØDE')
  121 CØNTINUE
      READ(NRDD,118) NB,NE,NØ
  118 FØRMAT(3I5)
      WRITE(NPRT,117) NB,NE,NØ
  117 FØRMAT(1H ,3I10)
      IF((NB.EQ.0).AND.(NE.EQ.0).AND.(NØ.EQ.0)) GØ TØ 122
      IF((NB.EQ.0).AND.(NE.EQ.0).AND.(NØ.EQ.-1)) GØ TØ 224
      DØ 119 N=NB,NE
      JT(N)=NØ
  119 CØNTINUE
      GØ TØ 121
  224 CØNTINUE
      DØ 225 I=1,256
```

```
  225 JT(I)=I
  122 CONTINUE
      CALL  TFILE(1,F1)
      CALL  TWAIT(1)
      CALL  TFILE(2,F2)
      CALL  TWAIT(2)
      CALL  TREAD(1,BUF1,1530)
      CALL  TWAIT(1)
      CALL  TREAD(2,BUF2,1530)
      CALL  TWAIT(2)
      CALL SWAB(S1)
      CALL SWAB(S2)
      WRITE(NPRT,302) S1,(BUF1(IB),IB=61,63),S2,(BUF2(IB),IB=61,63)
  302 FORMAT(//,10X,'GT SEG, NO.=',I5,5X,'DAY=',I5,5X,'MONTH=',I5,5X,
     1'YEAR=',I5,//,10X,'DTRM SEG,NO.=',I5,5X,'DAY=',I5,5X,'MONTH=',I5,
     25X,'YEAR=',I5)
      DO 11 L=1,117
      L10=MOD(L,10)
      CALL  TREAD(2,BUF2,180)
      CALL  TWAIT(2)
      DO 12 LL=1,3
      CALL  TREAD(1,BUF1,270)
      CALL  TWAIT(1)
      P=72
      SAM=0
      GEO=0
      DO 13 S=1,196
      S10=MOD(S,10)
      X=BUF2(S+72)
      IF(X.LE.0) X=X+256
      DO 14 SS=1,2
      NDOT=0
      IF((L10.EQ.0).AND.(S10.EQ.0).AND.(LL.EQ.3).AND.(SS.EQ.2)) NDOT=1
      P=P+1
      TV=BUF1(P)
         TV=TV+128
         TV=JT(TV)
      N2=TV/10
      N1=TV-10*N2
      IF(L10.EQ.0.AND.S10.EQ.0) SAM=SAM+1
      IF(L10.EQ.0.AND.S10.EQ.0) DOTR(LL,SAM)=N1
      IF((N1.NE.1).AND.(N1.NE.2)) GO TO 15
      RA(X,N1)=RA(X,N1)+1.0
      IF(NDOT.NE.1) GO TO 15
      SD10=S/10
      CALL GDUM(L,GEO,N1,NPRT,SD10)
      RA(X,(N1+2))=RA(X,(N1+2))+1.0
   15 CONTINUE
      IF((N2.NE.1).AND.(N2.NE.2)) GO TO 31
      RA(X,(N2+4))=RA(X,(N2+4))+1.0
   31 CONTINUE
      IF(NDOT.NE.1) GO TO 14
      N2=A(L/10,S/10)
      IF((N2.NE.1).AND.(N2.NE.2)) GO TO 14
      RA(X,(N2+6))=RA(X,(N2+6))+1.0
   14 CONTINUE
   13 CONTINUE
   12 CONTINUE
D     IF(L10.EQ.0) CALL PDOTR
   11 CONTINUE
D     DO 451 I=1,11
D 451 WRITE(NPRT,452) (G(I,J),J=1,19)
D 452 FORMAT(1H ,10X,19I5)
      DO 16 X=1,256
      DO 17 N=1,8
      RA(257,N)=RA(257,N)+RA(X,N)
```

```
   17 CONTINUE
   16 CONTINUE
      RTOT=RA(257,1)+RA(257,2)
      OS=-2
      DO 18 CASE=1,4
      OS=OS+2
      DO 23 X=1,257
      DO 24 N=9,18
      RA(X,N)=0.0
   24 CONTINUE
   23 CONTINUE
      RD=RA(257,(OS+1))+RA(257,(OS+2))
      IF(RD.EQ.0.0) GO TO 18
      RAP=RA(257,(OS+1))/RD
      IF(CASE.EQ.3) RAP=RAP*0.100.0
      DO 19 X=1,256
      RT=RA(X,1)+RA(X,2)
      IF(RT.EQ.0.0) GO TO 19
      RA(X,9)=RT/(RA(257,1)+RA(257,2))
      RA(X,10)=RA(X,(OS+1))/RA(257,(OS+1))
      RA(X,11)=RA(X,(OS+2))/RA(257,(OS+2))
      RA(X,12)=0.0
      RD=RAP*RA(X,10)+(1.0-RAP)*RA(X,11)
      IF(RD.EQ.0.0) GO TO 20
      RA(X,12)=RAP*RA(X,10)/RD
   20 CONTINUE
      RA(X,13)=RA(X,9)*RA(X,12)
      IF(RA(X,12).GT.0.5) RA(X,14)=RA(X,9)
      RA(X,15)=RA(X,9)*RA(X,12)*(1.0-RA(X,12))
      IF(RA(X,12).LT.0.5) GO TO 21
      RA(X,16)=RA(X,2)/RA(257,2)
      RA(X,18)=RA(X,2)/RTOT
      GO TO 22
   21 CONTINUE
      RA(X,17)=RA(X,1)/RA(257,1)
      RA(X,18)=RA(X,1)/RTOT
   22 CONTINUE
   19 CONTINUE
      DO 25 X=1,255
      DO 26 N=9,18
      RA(257,N)=RA(257,N)+RA(X,N)
   26 CONTINUE
   25 CONTINUE
      WRITE(NPRT,111) CASE
  111 FORMAT(1H1,'  CASE=',I5)
      IF(CASE.EQ.1) WRITE(NPRT,601)
      IF(CASE.EQ.2) WRITE(NPRT,602)
      IF(CASE.EQ.3) WRITE(NPRT,603)
      IF(CASE.EQ.4) WRITE(NPRT,604)
  601 FORMAT(/10X,'DTRM TAPE COMPARED WITH THE GROUND TRUTH')
  602 FORMAT(/10X,'DTRM TAPE COMPARED WITH THE 209 DATS')
  603 FORMAT(/10X,'DTRM TAPE COMPARED WITH THE TRAINING FIELDS')
  604 FORMAT(/10X,'DTRM TAPE COMPARED WITH THE AI DATS')
      WRITE(NPRT,401)
  401 FORMAT(/,4X,'X',2X,'N(X,W)',2X,'N(X,0)',1X,'NT(X,W)',1X,'NT(X,0)',
     13X,'PH(X)',2X,'PT(X/W)',1X,'PT(X/0)',1X,'PH(W/X)',1X,'PH(X,W)',
     23X,'PT(X)',1X,'ND*V(X)',1X,'PH(X/0)',1X,'PH(X/W)',2X,'PMC(X)')
      DO 27 X=1,257
      RT=RA(X,1)+RA(X,2)
      IF(RT.EQ.0.0) GO TO 27
      IF(X.EQ.257) WRITE(NPRT,402)
  402 FORMAT(/,10X,'TOTALS'///9X,'N(W)',4X,'N(0)',3X,'NT(W)',3X,'NT(0)',
     335X,'PH(W)',4X,'PDOT',4X,'ND*V',21X,'PMC')
      WRITE(NPRT,109) X,RA(X,1),RA(X,2),RA(X,(OS+1)),RA(X,(OS+2)),
     *(RA(X,N),N=9,18)
  109 FORMAT(1H ,I5,4F8.0,10F8.5)
```

3-109 110

```
   27 CONTINUE
          GO TO 18
  927      WRITE (NPRT,928)
  928     FORMAT (1H , 20X, 'AS AN NOT READ!')
          GO TO 18
  929     WRITE (NPRT,930)
  930     FORMAT (1H ,20X, 'AL,AP,AS NOT READ!')
   18 CONTINUE
          IF(NPASS,GT,1) WRITE(NPRT,805)
  805     FORMAT (1H1)
          F2=F2+1
          CALL CLOSE(NRDR)
          CALL CLOSE(NRDD)
  802     CONTINUE
C     CALL COMPAR(A,G,NPRT)
      CALL DATE(D)
      CALL TIME(T)
      WRITE(NPRT,333)
  333 FORMAT(///)
      WRITE(NPRT,110) D,T
      WRITE(NPRT,110) D,T
  110 FORMAT('  JOB COMPLETED ON  ',9A1,'  AT  ',8A1)
      STOP
      END
```

```
ALLCRP,LPT/SH=ALLCRP
CROPL
GDUM
PDOTR
[100,4]SWAP
[100,4]LECTAP
[1,1]F4POTS/LB
/
ACTFIL=8
UNITS=8
ASG=SY:4
ASG=SY:5
ASG=LP:6
MAXBUF=3060
PRI=50
//
```

3.2.18 SECOND MODULE - THIRD UNIT MLTCRP

3.2.18.1 Linkage

MLTCRP is a stand-alone program which calls standard system utility routines and the companion subroutines MLTRDD and MLTANL. MLTRDD calls subroutines ZOT, INDDT, INDGT, IERR, DDUM, and PDOTR as well as functions IFCN and JFCN. PROBT calls subroutines INDDM, MSUM, SORT, MTXPT, PROB, and PROBC.

3.2.18.2 Interface

Communication with subroutines is through the calling arguments and the common blocks RD, DD, MPI, CK,MTX, and PS.

3.2.18.3 Input

MLTCRP requires input of a "ground truth" magnetic tape, a companion "DTRM" tape (reference 1), card entries of corresponding analyst "dot" labels and transformations for both tape inputs.

3.2.18.4 Output

Printout of accuracy assessment parameters (see Appendix B) including:

1. Count matrices
2. Joint probabilities
3. Conditional probabilities
4. Dot Labels

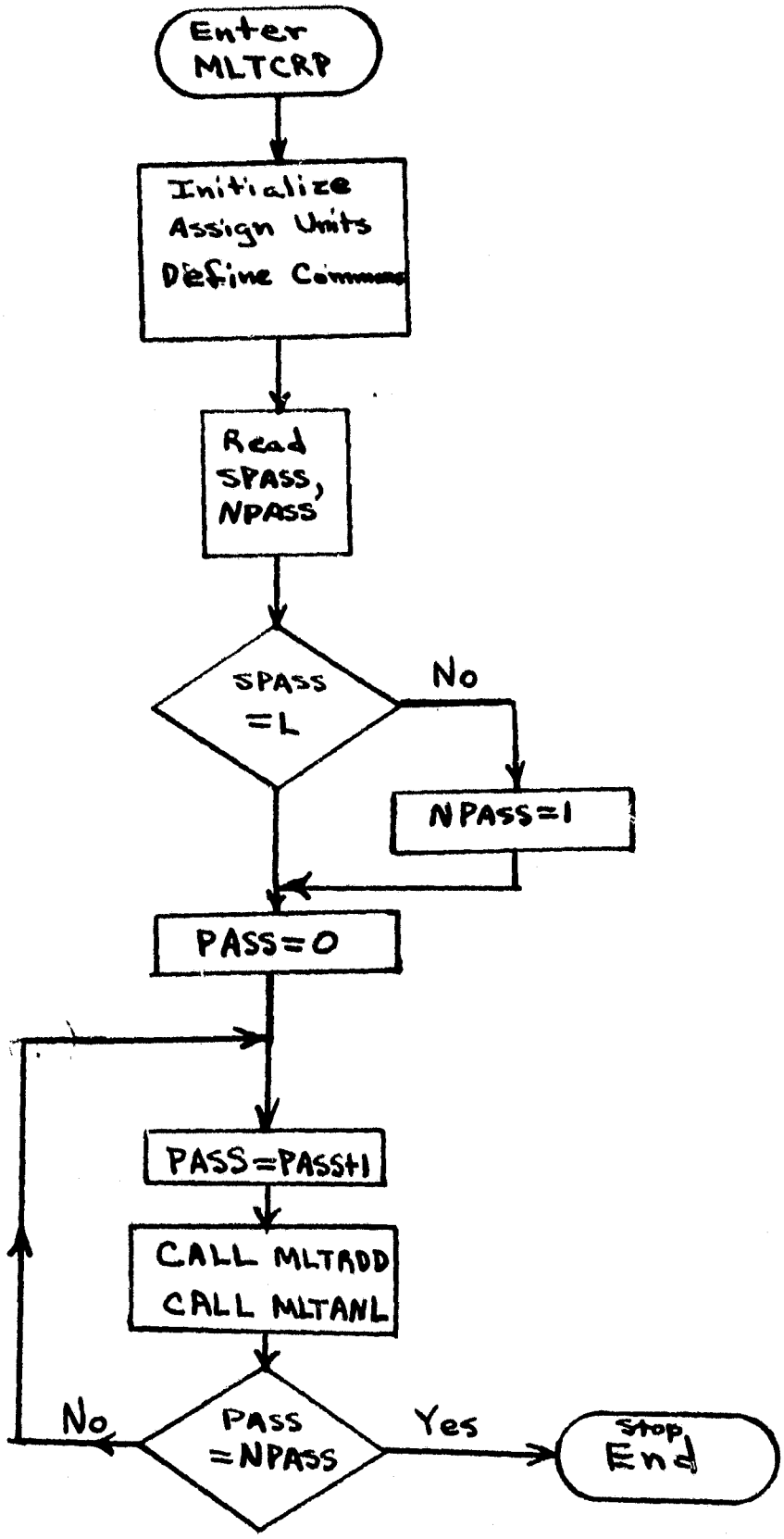
3.2.18.5 Storage

TBD

3.2.18.6 Description

MLTCRP compares using all crops ground truth data with operational classification data, analyst dot labeling and ERIPS automatic labeled data (DTRM) to determine their accuracy.

3.2.18.7a Flowchart



3.2.18.7b

```
      IMPLICIT INTEGER (A-Q),(S-Z)
      CØMMØN /RD/A(11,19),G(11,19),DT(11,19)
      CØMMØN /MPI/MIND ,RT,NPRT,TIND,NAID
      CØMMØN /CK/JG(256,4),JD(256, 4),ING,IND
      CØMMØN /MTX/RA( 50,50)
      CØMMØN /PS/BASS
      BYTE T(8),D(9)
      CALL TIME(T)
      CALL DATE(D)
         NRDR = 4
         NRDD=5
      NPRT=6
      WRITE(NPRT,703) D,T
  703 FØRMAT(1H1,'  JØB INITIATED ØN  ',9A1,'  AT  ',8A1,//,10X,
     1'PRØGRAM MLTCRP,FTN')
         ØPEN(UNIT=NRDD,NAME='L00P.DAT',TYPE='ØLD',
     *  ACCESS='SEQUENTIAL',FØRM='FØRMATTED',CARRIAGE CØNTRØL='NØNE')
         READ(NRDD,801) SPASS,NPASS
  801    FØRMAT(A1,4X,I5)
         CALL CLØSE(NRDD)
         IF(SPASS.NE.'L') NPASS=1
         DØ 802 PASS=1,NPASS
      BASS=PASS
      CALL MLTRDD
      CALL MLTANL
      WRITE(NPRT,334)
  334 FØRMAT(1H1)
  802    CØNTINUE
      CALL DATE(D)
      CALL TIME(T)
      WRITE(NPRT,333)
  333 FØRMAT(///)
      WRITE(NPRT,110) D,T
      WRITE(NPRT,110) D,T
  110 FØRMAT('  JØB CØMPLETED ØN  ',9A1,'  AT  ',8A1)
      STØP
      END
```

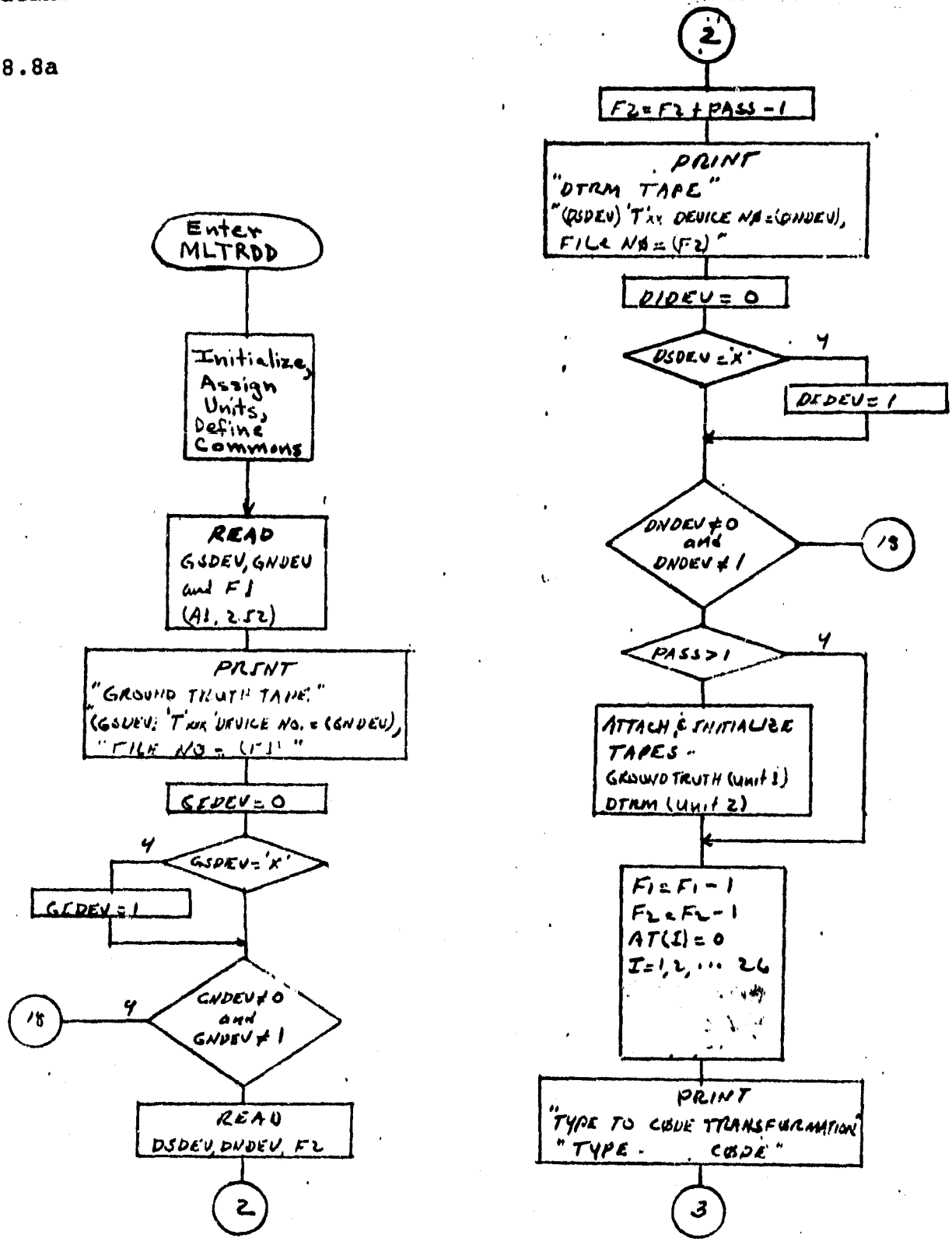
```
        ,PSECT RD,RW,GBL,REL,ØVR
        ,PSECT DD,RW,GBL,REL,ØVR
        ,PSECT MPI,RW,GBL,REL,ØVR
        ,PSECT CK,RW,GBL,REL,ØVR
        ,PSECT MTX,RW,GBL,REL,ØVR
        ,PSECT PS,RW,GBL,REL,ØVR
      ,RØØT MAIN=*(A,B)
MAINI ,FCTR MLTCRP=FLB
FLBI   ,FCTR [1,1]F4PØTS,ØLB/LB
AI      ,FCTR MLTRDD=*(LECTAP,SWAB,C)
BI      ,FCTR MLTANL
CI      ,FCTR PDØTR-DDUM-CRØPL
      ,END
```

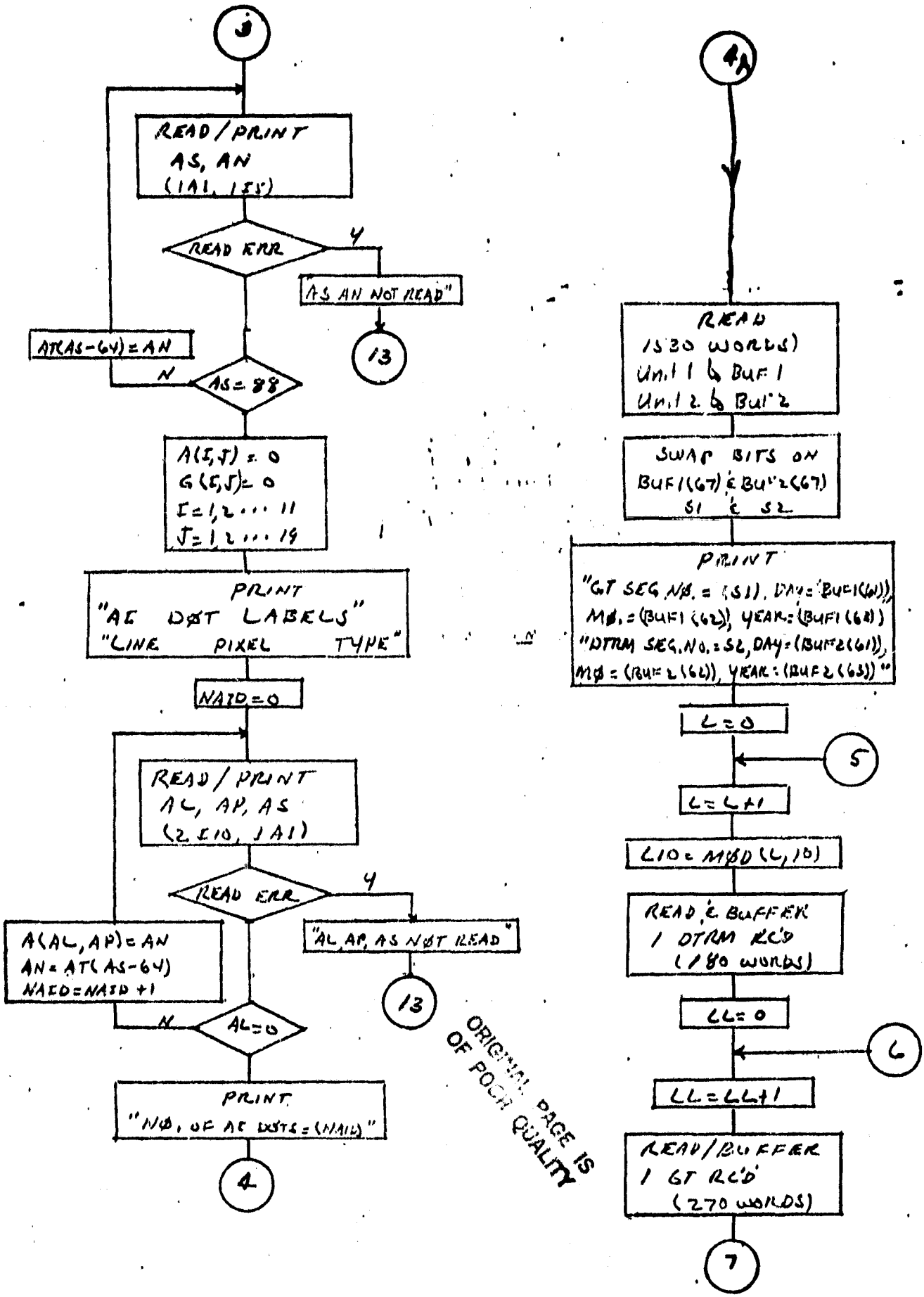
```
MLTCRP,LPI/SW=MLTCRP/MP
ACTFIL=8
UNITS=8
ASG=SY14
ASG=SY15
ASG=LP16
MAXBUF=3060
PRI=50
//
```

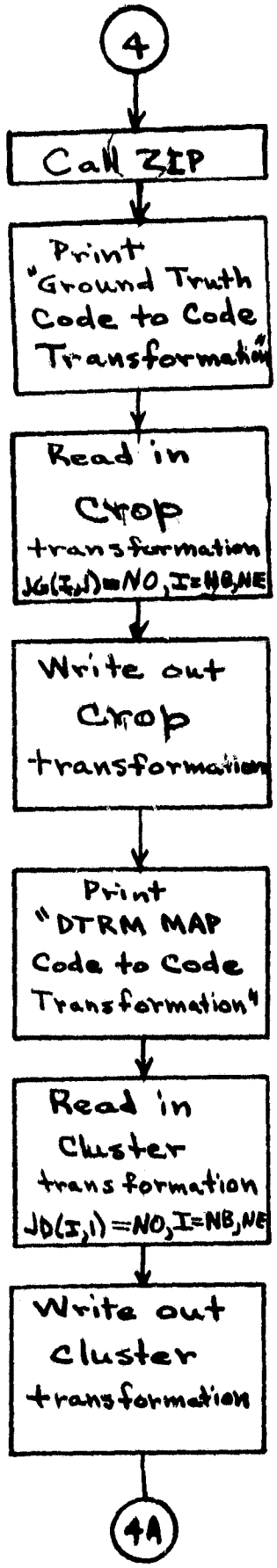
3.2.18.8 Subroutine MLTRDD

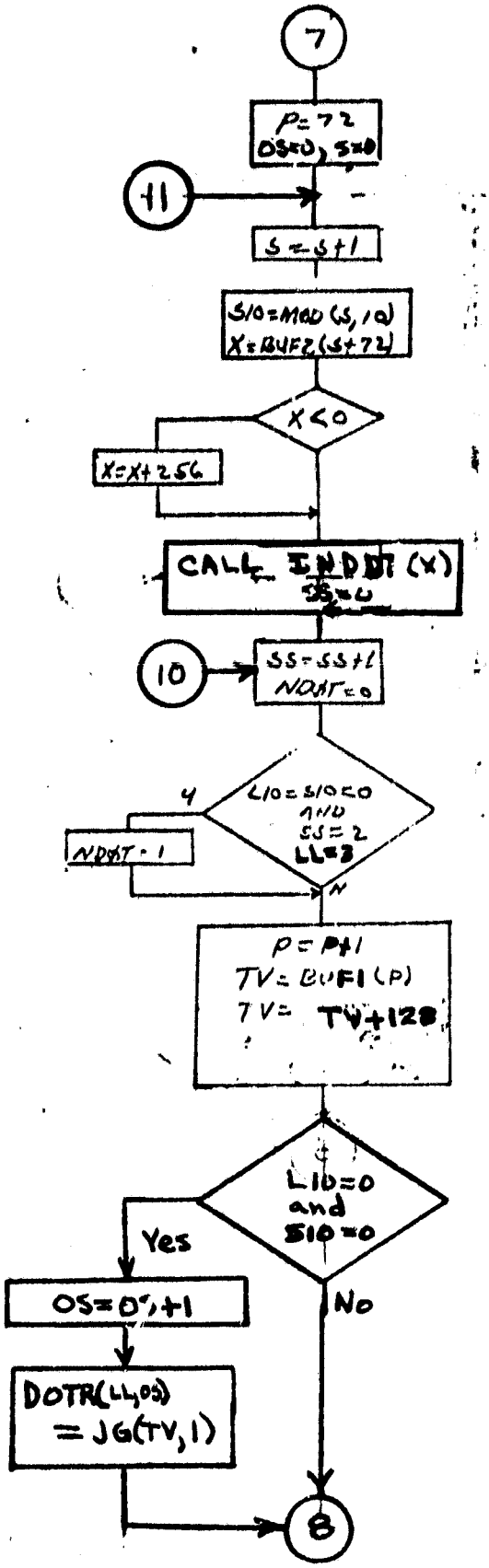
Subroutine MLTRDD reads in and analyses the input data.

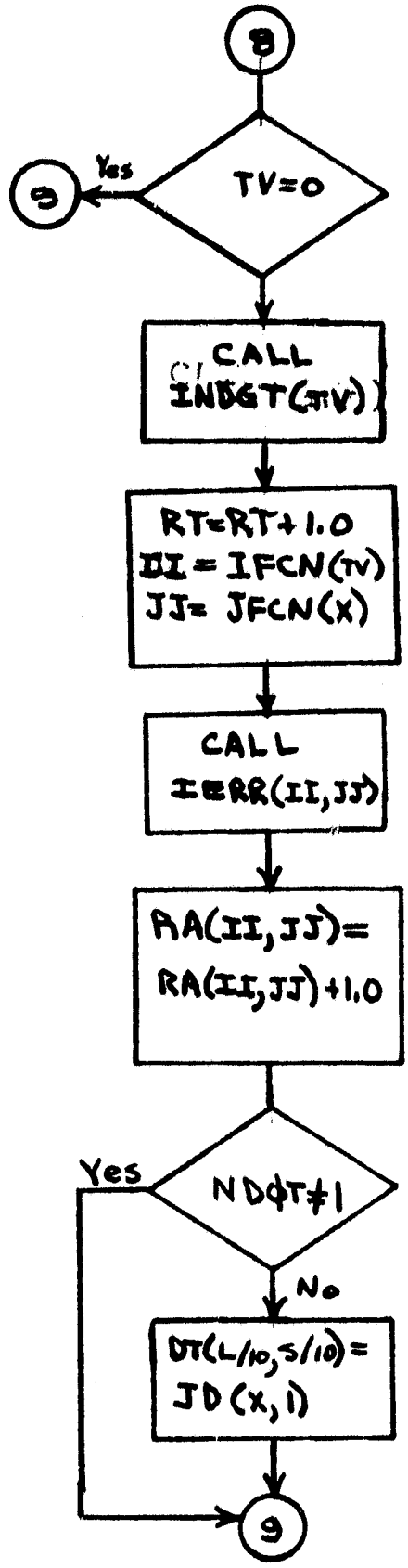
3.2.18.8a

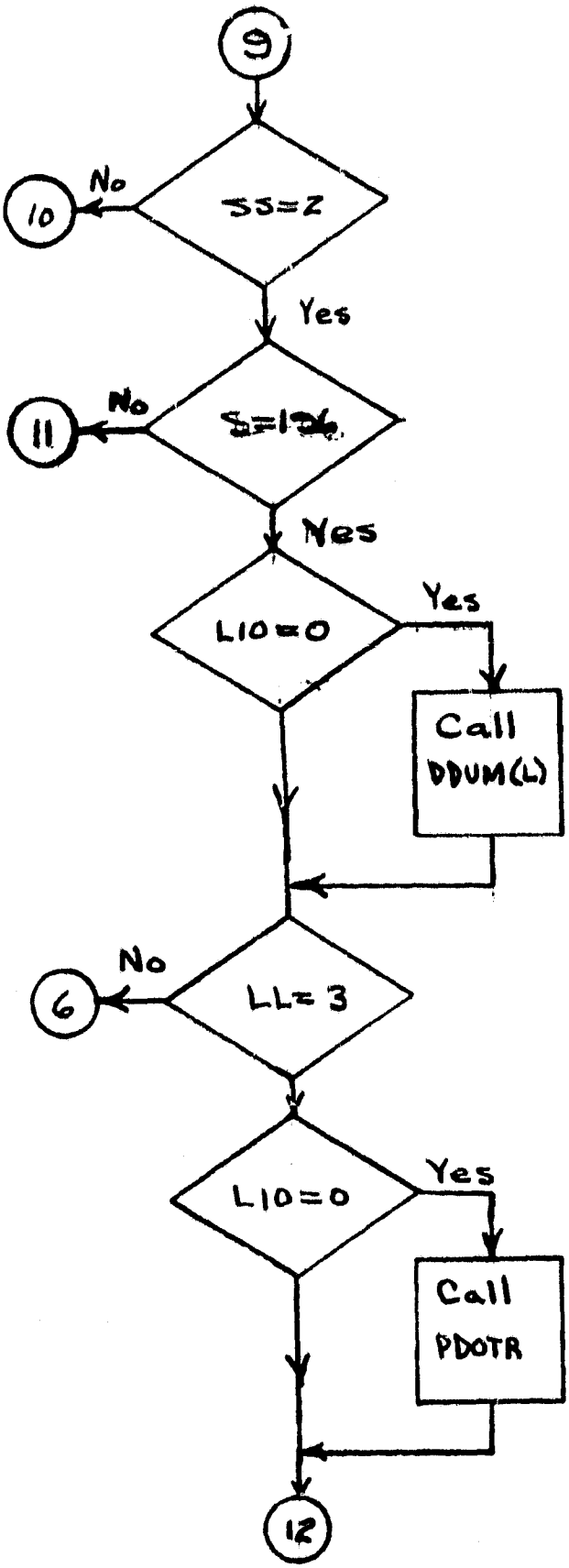


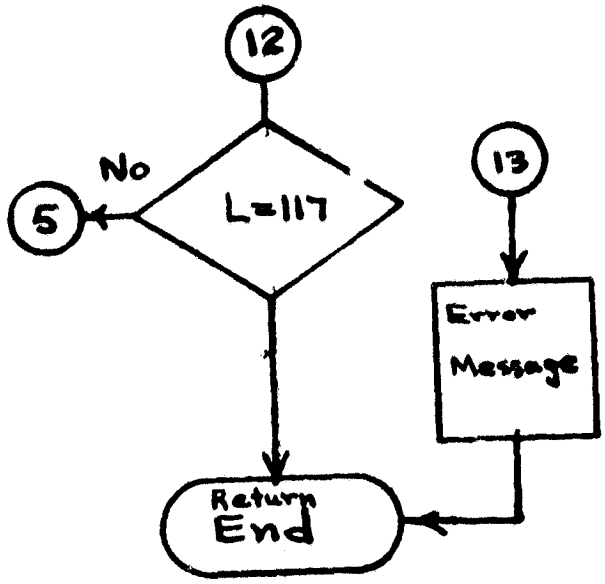



4
Call ZIP
Print "Ground Truth Code to Code Transformation"
Read in Crop transformation JG(I,1)=NO,I=NB,NE
Write out Crop transformation
Print "DTRM MAP Code to Code Transformation"
Read in Cluster transformation JD(I,1)=NO,I=NB,NE
Write out Cluster transformation
4A


7
P=72
OS=0, S=0
11
S=S+1
S/0=MOD(S,10)
X=BUFZ(S+72)
X<0
X=X+256
CALL INDGT(X)
10
SS=SS+1
NDGT=0
4
L/0=S/0=0 AND SS=2 LL=3
NDGT=1
N
P=P+1
TV=BUFI(P)
TV=TV+128
L/0=0 and S/0=0
Yes
No
OS=0, +1
DOTR(LL,OS) = JG(TV,1)
8


8
9
Yes
TV=0
CALL INDGT(TV)
RT=RT+1.0
II=IFCN(TV)
JJ=JFCN(X)
CALL IERR(II,JJ)
RA(II,JJ)= RA(II,JJ)+1.0
Yes
NDGT≠1
No
DT(L/10,S/10)= JD(X,1)
9


9
No
10
SS=Z
Yes
No
11
S=126
Yes
L10=0
Yes
Call
DDUM(L)
No
6
LL=3
L10=0
Yes
Call
PDOTR
12


12
No
5
L=117
13
Error
Message
Return
End

```
3.2.18.8b

      SUBROUTINE MLTRDD
      IMPLICIT INTEGER (A-Q),(S-Z)
      DIMENSION AT(26)
      COMMON /MPI/MIND ,RT,NPRT,TIND,NAID
      COMMON /RD/A(11,19),G(11,19),DT(11,19)
      COMMON /PS/PASS
      COMMON /DD/DDTR(3,38)
      BYTE BUF1(3060),BUF2(3060)
      BYTE AS
      BYTE CHAR(50)
      EQUIVALENCE (S1,BUF1(67)),(S2,BUF2(67))
      COMMON /STATUS/W1,W2
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      COMMON /MTX/RA( 50,50)
         NRDR = 4
         NRDD=5
         OPEN(UNIT=NRDD,NAME='MLTCRP.DAT',TYPE='OLD',
     *  ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGE CONTROL='NONE')
      READ(NRDD,704) GSDEV,GNDEV,F1
  704 FORMAT(A1,1X,2I2)
      WRITE(NPRT,705) GSDEV,GNDEV,F1
  705 FORMAT(//,10X,'GROUND TRUTH TAPE',//,10X,A1,'T',10X,'DEVICE NO.=',
     1I5,10X,'FILE NO.=',I5)
      GIDEV=0
      IF(GSDEV.EQ.'X') GIDEV=1
      IF(GNDEV.NE.0.AND.GNDEV.NE.1) GO TO 18
      READ(NRDD,704) DSDEV,DNDEV,F2
         F2=F2+PASS-1
      WRITE(NPRT,707) DSDEV,DNDEV,F2
  707 FORMAT(//,10X,' DTRM TAPE       ',//,10X,A1,'T',10X,'DEVICE NO.=',
     1I5,10X,'FILE NO.=',I5)
      DIDEV=0
      IF(DSDEV.EQ.'X') DIDEV=1
      IF(DNDEV.NE.0.AND.DNDEV.NE.1) GO TO 16
         OPEN(UNIT=NRDR,NAME='AI.DAT',TYPE='OLD',ACCESS='SEQUENTIAL',
     *  FORM='FORMATTED',CARRIAGE CONTROL='NONE')
         IF(PASS.GT.1) GO TO 888
      CALL  TINIT(1,GIDEV,GNDEV)
      CALL  TINIT(2,DIDEV,DNDEV)
      CALL  TATCH(1)
      CALL  TATCH(2)
  888    CONTINUE
      CALL  TRWD(1)
      CALL  TWAIT(1)
      CALL  TRWD(2)
      CALL  TWAIT(2)
      F1=F1-1
      F2=F2-1
C
      DO 1 I=1,26
      AT(I)=0
    1 CONTINUE
      WRITE(NPRT,702)
  702 FORMAT(/,10X,'TYPE TO CODE TRANSFORMATION')
      WRITE(NPRT,300)
  300 FORMAT(/,3X,'TYPE',6X,'CODE')
    5 CONTINUE
      READ(NRDD,101,ERR=927) AS,AN
  101 FORMAT(1A1,4X,I1)
      WRITE(NPRT,102) AS,AN
  102 FORMAT(1H ,5X,A1,I10)
      IF(AS.EQ.88) GO TO 6
```

3-120 121

```
      AI(AS-64)=AN
      GØ TØ 9
    6 CØNTINUE
      DØ 7 I=1,11
      DØ 8 J=1,19
      A(I,J)=0
      G(I,J)=0
      DT(I,J)=0
    8 CØNTINUE
    7 CØNTINUE
      WRITE(NPRT,701)
  701 FØRMAT(/,10X,'AI DØT LABLES')
      WRITE(NPRT,301)
  301 FØRMAT(/,7X,'LINE',6X,'PIXEL',7X,'TYPE')
      NAID=0
    9 CØNTINUE
      READ(NRDR,103,ERR=929)  AL,AP,AS,(CHAR(I),I=1,50)
  103 FØRMAT(10X,1I2,1X,1I2,1X,1A1,13X,50A1)
      WRITE(NPRT,104) AL,AP,AS,(CHAR(I),I=1,50)
  104 FØRMAT(1H ,2I10,9X,1A1,13X,50A1)
      IF(AL.EQ.0) GØ TØ 10
      NAID=NAID+1
      AN=AT(AS-64)
      IF(A(AL,AP).NE.0) WRITE(NPRT,313) AL,AP
  313 FØRMAT(///,10X,'DUPLICATE DØT LABEL !ØN DØT * ',2I5,///)
      A(AL,AP)=AN
      GØ TØ 9
   10 CØNTINUE
      WRITE(NPRT,502) NAID
  502 FØRMAT(/,10X,'NØ. ØF AI DØTS=',I5)
      DØ 50 I=1,256
      JG(I,1)=0
      JD(I,1)=0
   50 CØNTINUE
      CALL ZIP
         WRITE(NPRT,915)
  915    FØRMAT(1HØ,10X,'GRØUND TRUTH')
      WRITE (NPRT,905)
  905 FØRMAT(//,10X,'CØDE TØ CØDE TRANSFØRMATIØN',//,8X,'BEGIN',7X,
     1'END' ,7X,'CØDE')
  121 CØNTINUE
      READ(NRDD,118) NB,NE,NØ
  118 FØRMAT(3I5)
      WRITE(NPRT,117) NB,NE,NØ
  117 FØRMAT(1H ,3I10)
      IF((NB.EQ.0).AND.(NE.EQ.0).AND.(NØ.EQ.0)) GØ TØ 122
      IF((NB.EQ.0).AND.(NE.EQ.0).AND.(NØ.EQ.-1)) GØ TØ 224
      DØ 119 N=NB,NE
        JG(N,1)=NØ
  119 CØNTINUE
      GØ TØ 121
  224 CØNTINUE
      DØ 225 I=1,256
  225    JG(I,1)=I
  122 CØNTINUE
      WRITE(NPRT,916)
  916 FØRMAT(1HØ,10X,'DTRM MAP')
      WRITE (NPRT,905)
  321    CØNTINUE
      READ(NRDD,118) NB,NE,NØ
      WRITE(NPRT,117) NB,NE,NØ
      IF((NB.EQ.0).AND.(NE.EQ.0).AND.(NØ.EQ.0)) GØ TØ 322
      IF((NB.EQ.0).AND.(NE.EQ.0).AND.(NØ.EQ.-1)) GØ TØ 424
      DØ 319 N=NB,NE
         JD(N,1) = NØ
  319    CØNTINUE
```

~~3-121~~ 122

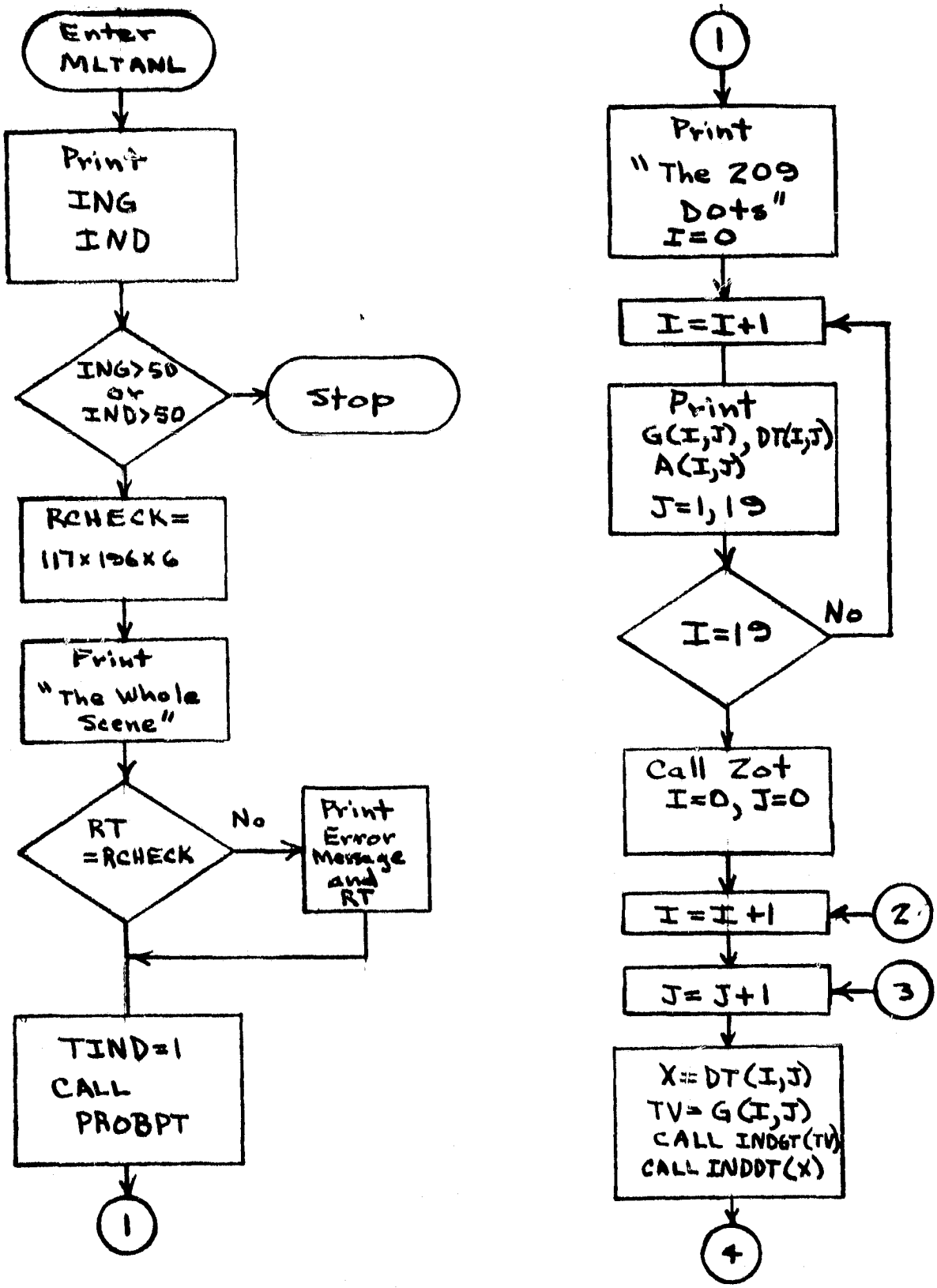
```
      GØ TØ 321
  424 CØNTINUE
      DØ 325 I=1,256
      JD(I,1)=I
  325 CØNTINUE
  322 CØNTINUE
      CALL TFILE(1,F1)
      CALL TWAIT(1)
      CALL TFILE(2,F2)
      CALL TWAIT(2)
      CALL TREAD(1,BUF1,1530)
      CALL TWAIT(1)
      CALL TREAD(2,BUF2,1530)
      CALL TWAIT(2)
      CALL SWAB(S1)
      CALL SWAB(S2)
      WRITE(NPRT,302) S1,(BUF1(IB),IB=61,63),S2,(BUF2(IB),IB=61,63)
  302 FØRMAT(//,10X,'GT SEG, NØ,=',I5,5X,'DAY=',I5,5X,'MØN=',I5,5X,
     1'YEAR=',I5//,10X,'DTRM SEG,NØ,=',I5,5X,'DAY=',I5,5X,'MØN=',I5,
     25X,'YEAR=',I5)
      WRITE(NPRT,337)
  337 FØRMAT(////,10X,'THE CØNFIGURATIØNS ØF THE 209 DØTS',////)
      DØ 11 L=1,117
      L10=MØD(L,10)
      CALL TREAD(2,BUF2,180)
      CALL TWAIT(2)
      DØ 12 LL=1,3
      CALL TREAD(1,BUF1,270)
      CALL TWAIT(1)
      P=72
      ØS=0
      DØ 13 S=1,196
      S10=MØD(S,10)
      X=BUF2(S+72)
      IF(X,LE,0) X=X+256
      CALL INDØT(X)
      DØ 14 SS=1,2
      NDØT=0
      IF((L10,EQ,0),AND,(S10,EQ,0),AND,(LL,EQ,0),AND,(SS,EQ,2)) NDØT=1
      P=P+1
      TV=BUF1(P)
         TV=TV+128
      IF(L10,EQ,0,AND,S10,EQ,0)  ØS=ØS+1
      IF(L10,EQ,0,AND,S10,EQ,0)  DØTR(LL,ØS)=BIG(TV,1)
      IF(TV,EQ,0) GØ TØ 14
      CALL INDGT(TV)
      RT=RT+1,0
      II=IFCN(IV)
      JJ=JFCN(X)
      CALL IERR(II,JJ)
      RA(II,JJ)=RA(II,JJ)+1,0
      IF(NDØT,NE,1) GØ TØ 15
      DT(L/10,S/10)=JD(X,1)
   15 CØNTINUE
   14 CØNTINUE
   13 CØNTINUE
      IF(L10,EQ,0)                    CALL DDUM(L)
   12 CØNTINUE
      IF(L10,EQ,0)                    CALL PDØTR
   11 CØNTINUE
         GØ TØ 16
  927   WRITE (NPRT,928)
  928   FØRMAT (1H , 20X, 'AS AN NØT READ')
        GØ TØ 18
  929   WRITE (NPRT,930)
  930   FØRMAT (1H ,20X, 'AL,AP,AS NØT READ')
```

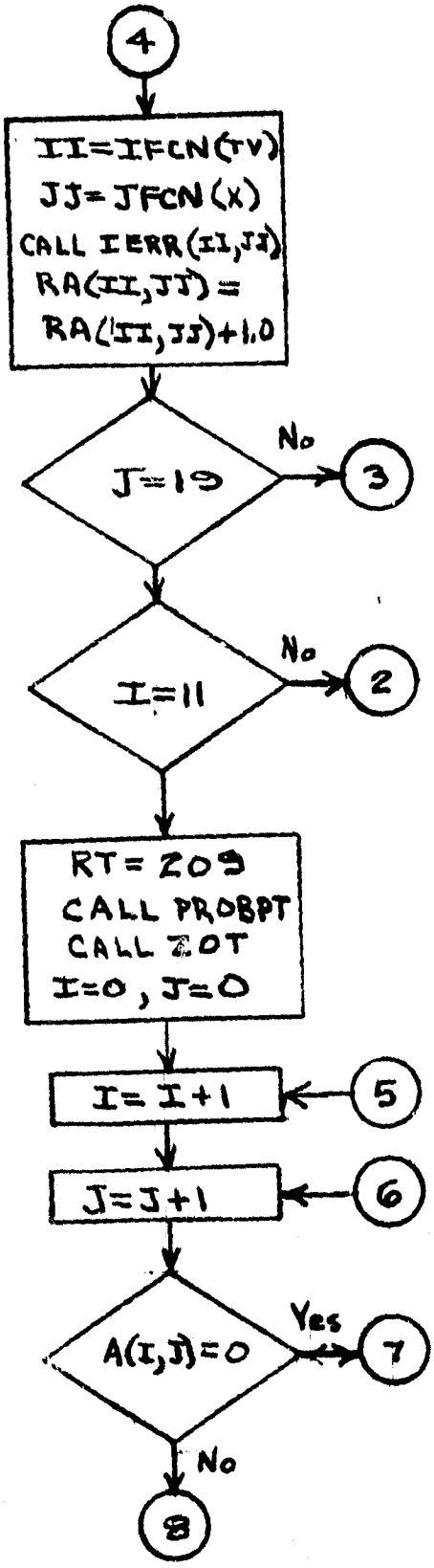
```
10 CONTINUE
   CALL CLOSE(NRDD)
   CALL CLOSE(NRDR)
   RETURN
   END
```

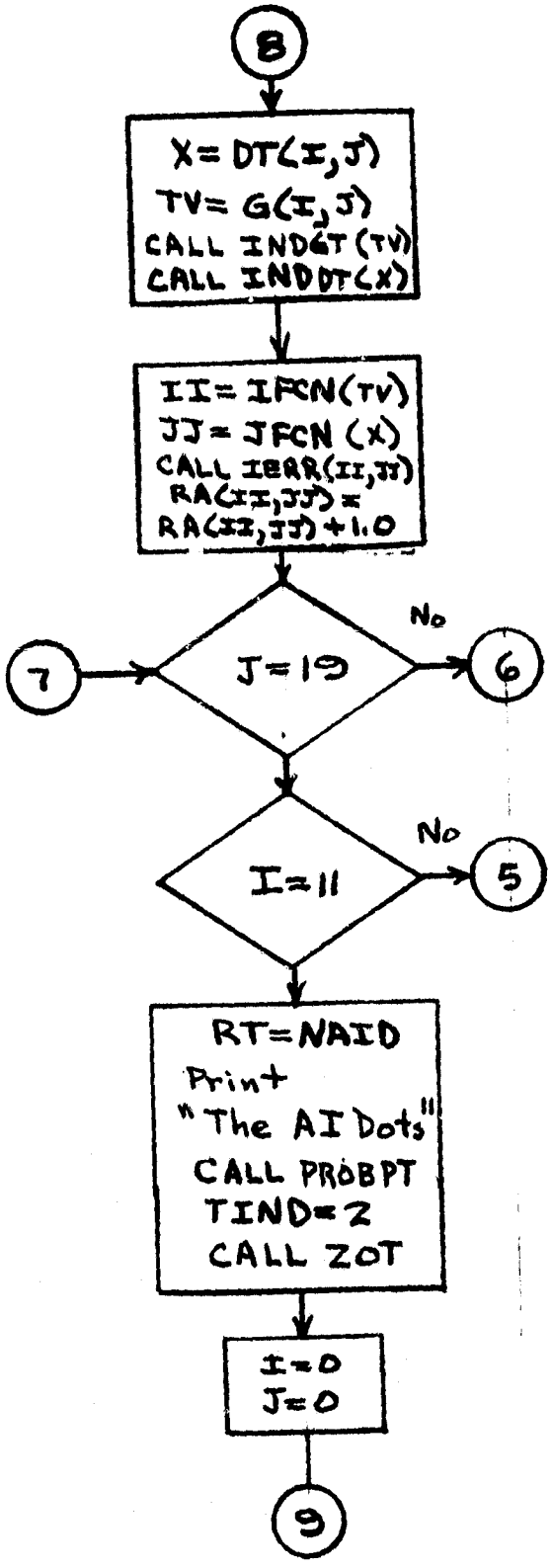
3.2.18.9 Subroutine MLTANL

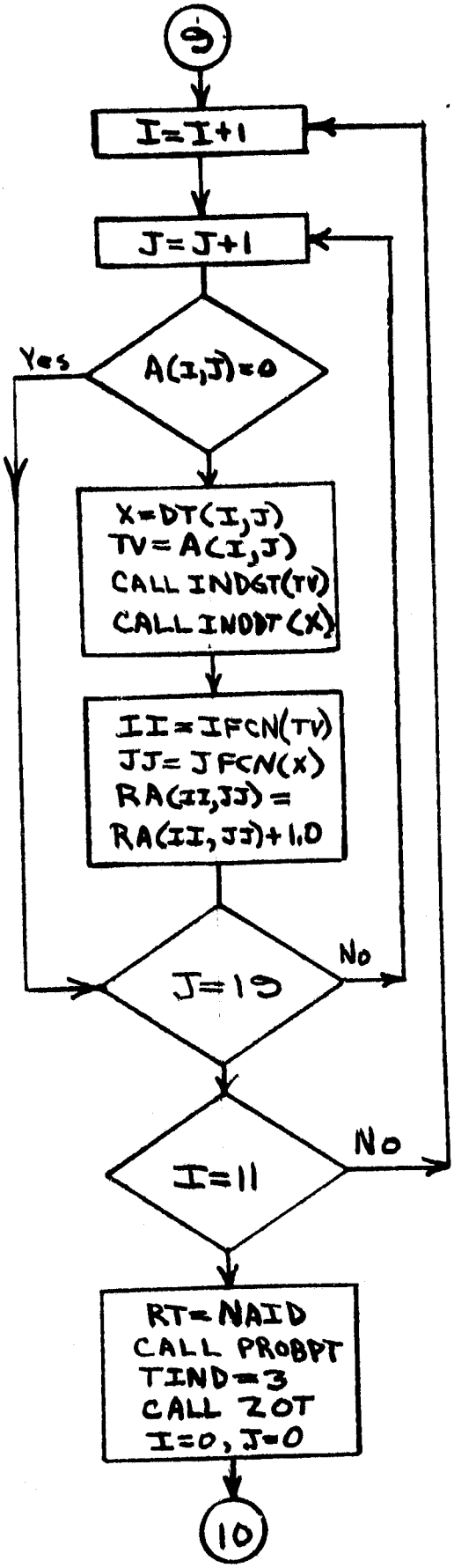
Subroutine MLTANL computes the analysis of the data

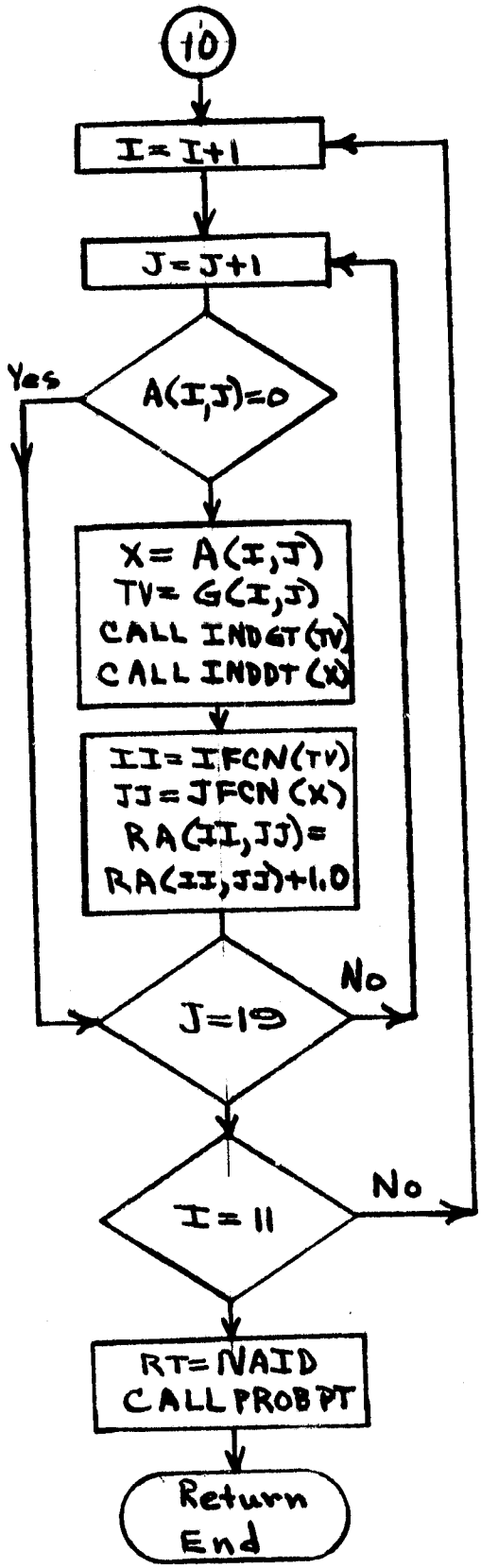
3.2.18.9a Flowchart




4
II=IFCN(TV)
JJ=JFCN(X)
CALL IERR(II,JJ)
RA(II,JJ)=
RA(II,JJ)+1.0
J=19
No
3
I=11
No
2
RT=209
CALL PROBPT
CALL ZOT
I=0, J=0
I=I+1
5
J=J+1
6
A(I,J)=0
Yes
7
No
8


8
X=DT(I,J)
TV=G(I,J)
CALL INDGT(TV)
CALL INDDT(X)
II=IFCN(TV)
JJ=JFCN(X)
CALL IERR(II,JJ)
RA(II,JJ)=
RA(II,JJ)+1.0
7
J=19
No
6
I=11
No
5
RT=NAID
Print
"The AI Dots"
CALL PROBPT
TIND=2
CALL ZOT
I=0
J=0
9


9
I=I+1
J=J+1
Yes
A(I,J)=0
X=DT(I,J)
TV=A(I,J)
CALL INDGT(TV)
CALL INDDT(X)
II=IFCN(TV)
JJ=JFCN(X)
RA(II,JJ)=
RA(II,JJ)+1.0
J=19
No
I=11
No
RT=NAID
CALL PROBPT
TIND=3
CALL ZOT
I=0, J=0
10


10
I=I+1
J=J+1
Yes
A(I,J)=0
X=A(I,J)
TV=G(I,J)
CALL INDGT(TV)
CALL INDDT(X)
II=IFCN(TV)
JJ=JFCN(X)
RA(II,JJ)=
RA(II,JJ)+1.0
J=19
No
I=11
No
RT=NAID
CALL PROBPT
Return
End

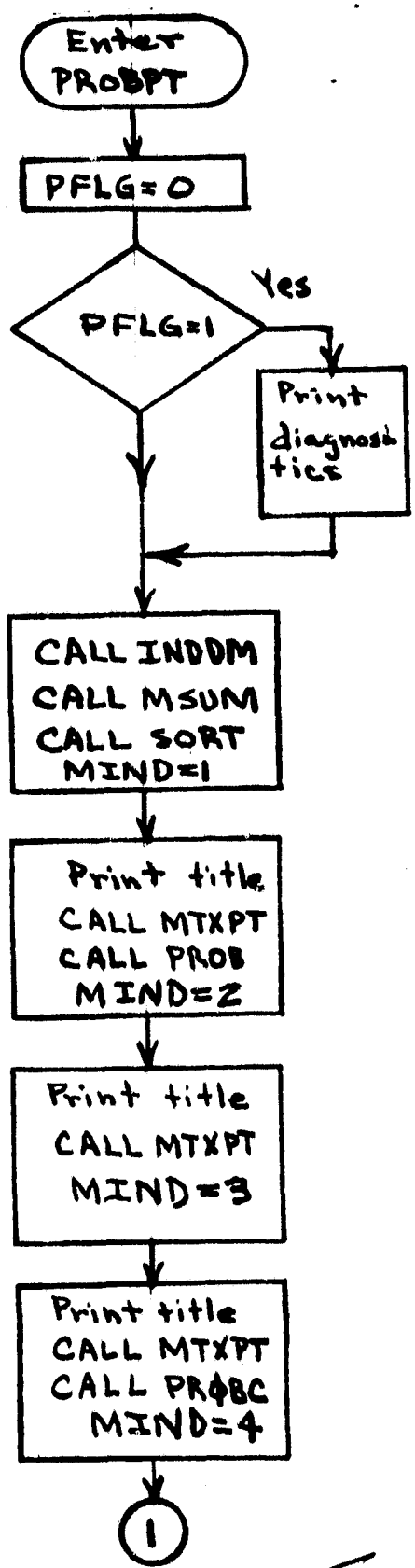
```
3.2.18.9b

      SUBROUTINE MLTANL
      IMPLICIT INTEGER (A-Q),(S-Z)
      COMMON /RD/A(11,19),G(11,19),DT(11,19)
      COMMON /MTX/RA( 50,50)
      COMMON /CK/JG(256, 4),JD(256, 4),ING,IND
      COMMON /MPI/MIND ,RT,NPRT,TIND,NAID
      WRITE(NPRT,776) ING
      WRITE(NPRT,775) IND
  775 FORMAT(1H0,10X,'IND=',I5)
  776 FORMAT(1H0,10X,'ING=',I5)
      IF(ING.GT. 50) STOP
      IF(IND.GT. 50) STOP
      RCHECK=117.0*196.0*6.0
         WRITE(NPRT,230)
  230    FORMAT (1H0,10X,'THE WHOLE SCENE')
      IF(RT.NE.RCHECK) WRITE(NPRT,223)
  223 FORMAT(//,10X,'THE WHOLE SEGMENT WAS NOT GROUND TRUTHED!')
      WRITE(NPRT,222) RT
  222 FORMAT(//,10X,'COMPUTATIONS BASED ON ',I10,2,' SUBPIXELS',/)
      TIND=1
      CALL PROBPT
      WRITE (NPRT,880)
  880    FORMAT (1H1, 10X, 'THE 209 DOTS')
         DO 881 I=1,11
         WRITE(NPRT,882) (G(I,J), J=1,19)
  882    FORMAT (1H0,10X,'GT', 19I5)
         WRITE (NPRT,883) (DT(I,J), J=1,19)
  883    FORMAT(1H , 10X,'DT',19I5)
         WRITE (NPRT,884) (A(I,J),J=1,19)
  884    FORMAT (1H , 10X,'AI',19I5)
  881    CONTINUE
         CALL ZOT
         DO 820 I=1,11
         DO 820 J=1,19
         X = DT(I,J)
         TV = G(I,J)
         CALL INDGT(TV)
         CALL INDDT(X)
         II=IFCN(TV)
         JJ=JFCN(X)
      CALL IERR(II,JJ)
         RA(II,JJ) = RA(II,JJ)+1.0
  820    CONTINUE
         RT=209.0
      CALL PROBPT
         CALL ZOT
         DO 920 I=1,11
         DO 920 J=1,19
      IF(A(I,J).EQ.0)  GO TO 920
         X = DT(I,J)
         TV = G(I,J)
         CALL INDGT(TV)
         CALL INDDT(X)
         II=IFCN(TV)
         JJ=JFCN(X)
      CALL IERR(II,JJ)
         RA(II,JJ) = RA(II,JJ)+1.0
  920 CONTINUE
      RT=NAID
      WRITE(NPRT,921)
  921 FORMAT(1H0,10X,'THE AI DOTS')
      CALL PROBPT
```

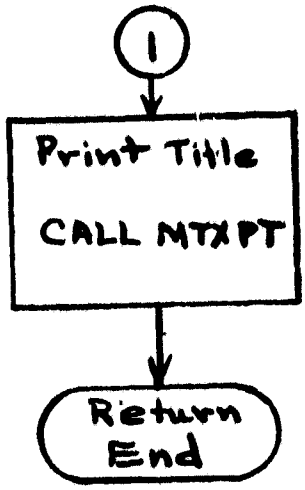
```
      TIND=2
      CALL ZØT
      DØ 950 I=1,11
      DØ 950 J=1,19
      IF(A(I,J),EQ.0) GØ TØ 950
      X=DT(I,J)
      TV=A(I,J)
      CALL INDGT(TV)
      CALL INDDT(X)
      II=IFCN(TV)
      JJ=JFCN(X)
      RA(II,JJ)=RA(II,JJ)+1,0
  950 CØNTINUE
      RT=NAID
      CALL PRØBPT
        TIND=3
        CALL ZØT
        DØ 960 I=1,11
        DØ 960 J=1,19
        IF(A(I,J),EQ.0) GØ TØ 960
        X=A(I,J)
        TV=G(I,J)
        CALL INDGT(TV)
        CALL INDDT(X)
        II=IFCN(TV)
        JJ=JFCN(X)
        RA(II,JJ)=RA(II,JJ)+1,0
  960   CØNTINUE
        RT=NAID
        CALL PRØBPT
      RETURN
      END
```

3.2.18.10 Subroutine PROBPT

Subroutine PROBPT calls subroutines which compute probabilities and counts and print them out.

3.2.18.10a Flowchart




1
Print Title
CALL MTXPT
Return
End

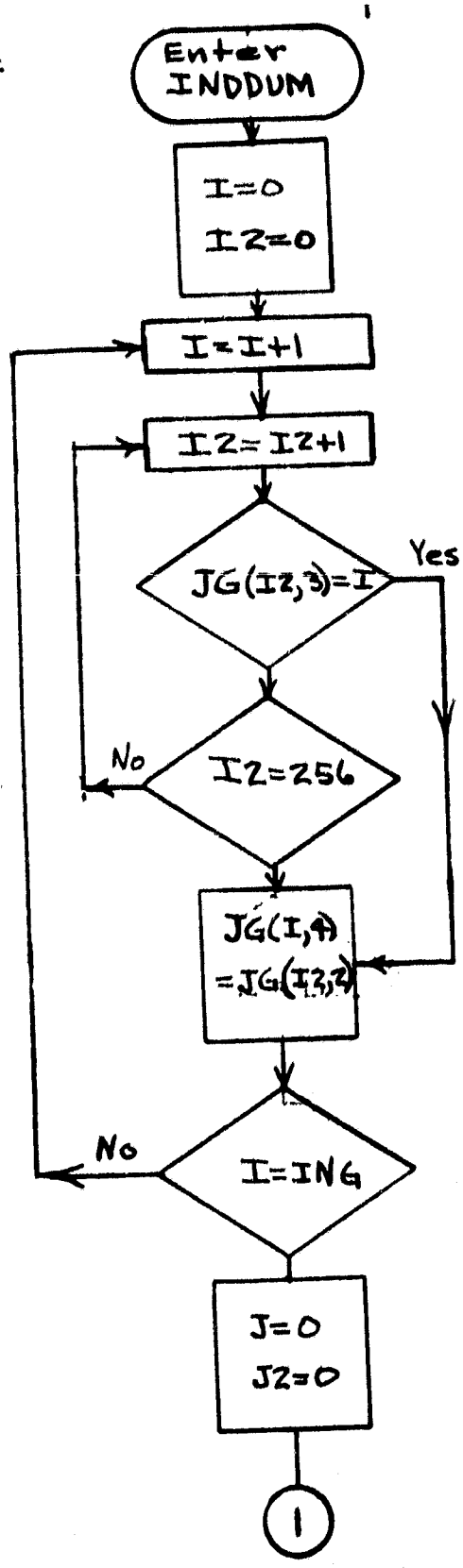
3.2.18.10b

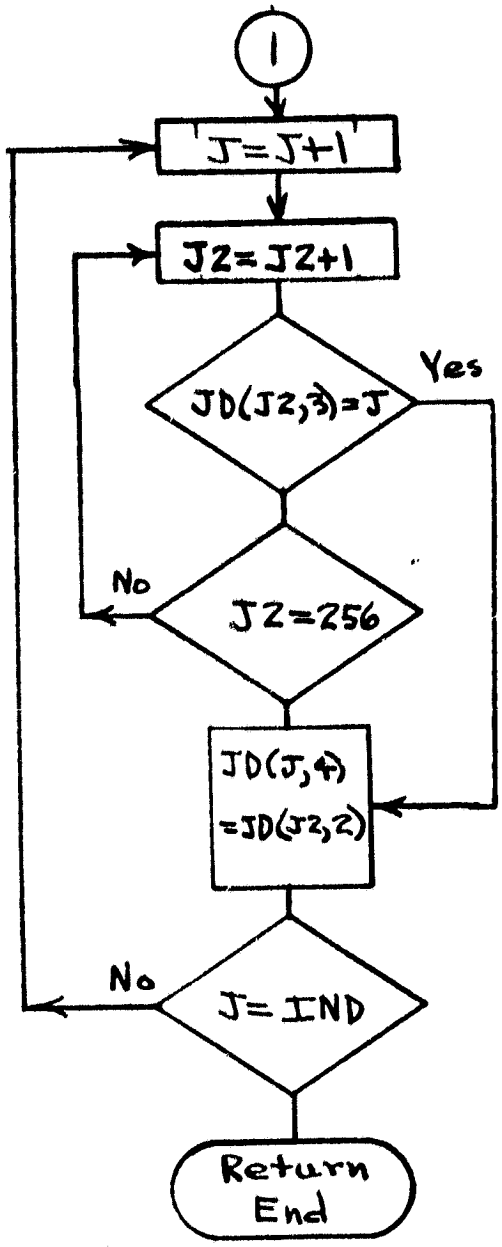
```
      SUBROUTINE PROBPT
      IMPLICIT INTEGER (A-Q),(S-Z)
      COMMON /MPI/MIND ,RT,NPRT,TIND,NAID
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      COMMON /MTX/RA( 50,50)
      PFLG=0
      IF(PFLG,NE,1) GO TO 261
         WRITE (NPRT,260) ((JD(I,J), J=1,4),I=1,256)
  260    FORMAT (1H , 10X,4I10)
         WRITE (NPRT,260) ((JG(I,J),J=1,4),I=1,256)
  261 CONTINUE
      IF(IND,LE,0) STOP
         CALL INDDM
      IF(PFLG,NE,1) GO TO 262
         WRITE (NPRT,260) ((JD(I,J), J=1,4),I=1,256)
         WRITE (NPRT,260) ((JG(I,J),J=1,4),I=1,256)
  262 CONTINUE
      CALL MSUM
      CALL SORT
      MIND=1
      IF(TIND,EQ,1)  WRITE(NPRT,750)
      IF(TIND,EQ,2)  WRITE(NPRT,850)
        IF(TIND,EQ,3) WRITE(NPRT,950)
  750 FORMAT(1H1,10X,'THE MATRIX N(G,D)')
  850 FORMAT(1H1,10X,'THE MATRIX N(AI,D)')
  950   FORMAT(1H1,10X,'THE MATRIX N(G,AI)')
      CALL MTXPT
      CALL PROB
      MIND=2
      IF(TIND,EQ,1) WRITE(NPRT,755)
      IF(TIND,EQ,2) WRITE(NPRT,855)
        IF(TIND,EQ,3) WRITE(NPRT,955)
  755 FORMAT(1H1,10X,'THE MATRIX P(G,D)')
  855 FORMAT(1H1,10X,'THE MATRIX P(AI,D)')
  955   FORMAT(1H1,10X,'THE MATRIX P(G,AI)')
        CALL MTXPT
          MIND = 3
        IF(TIND,EQ,1) WRITE(NPRT,756)
        IF(TIND,EQ,2) WRITE(NPRT,856)
          IF(TIND,EQ,3.) WRITE(NPRT,956)
  756     FORMAT (1H1,10X, 'THE MATRIX P(D/G)')
  856     FORMAT (1H1,10X, 'THE MATRIX P(D/AI)')
  956     FORMAT(1H1,10X,'THE MATRIX P(AI/G)')
          CALL MTXPT
          CALL PROBC
          MIND =4
        IF(TIND,EQ,1) WRITE(NPRT,757)
        IF(TIND,EQ,2) WRITE(NPRT,857)
          IF(TIND,EQ,3) WRITE(NPRT,957)
  757     FORMAT (1H1,10X, 'THE MATRIX P(G/D)')
  857     FORMAT (1H1,10X, 'THE MATRIX P(AI/D)')
  957     FORMAT(1H1,10X,'THE MATRIX P(G/AI)')
          CALL MTXPT
        RETURN
        END
```

3.2.18.11 Subroutine INDDUM

Subroutine INDDUM stores input codes for access.

3.2.18.11a <u>Flowchart</u>




1
J=J+1
JZ=JZ+1
JD(JZ,3)=J
Yes
No
JZ=256
JD(J,4)
=JD(JZ,2)
No
J=IND
Return
End

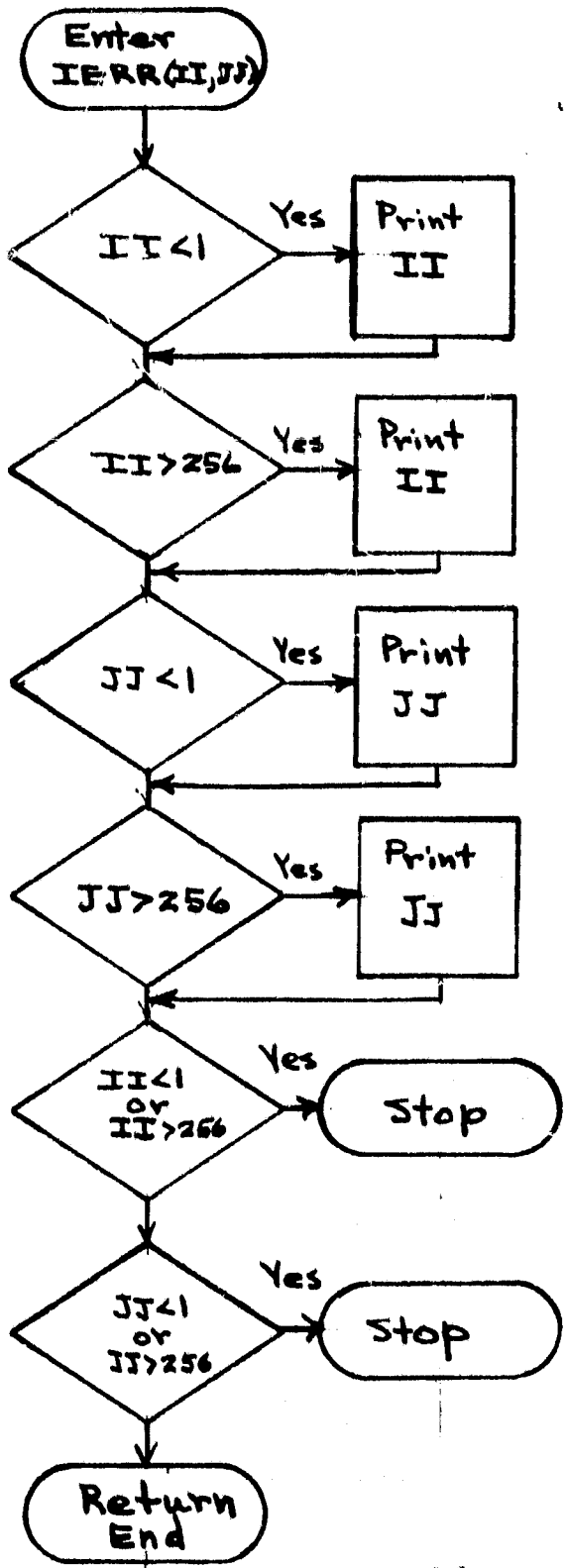
3.2.18.11b

```
      SUBROUTINE INDDM
      IMPLICIT INTEGER(A-Z)
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      DO 10 I=1,ING
      DO 20 I2=1,256
   20 IF(JG(I2,3).EQ.I) GO TO 30
   30 CONTINUE
      JG(I,4) = JG(I2,2)
   10 CONTINUE
      DO 11 J=1,IND
      DO 21 J2=1,256
   21 IF (JD(J2,3).EQ.J) GO TO 31
   31 CONTINUE
      JD(J,4) =JD(J2,2)
   11 CONTINUE
      RETURN
      END
```

3.2.18.12 Subroutine IERR

Subroutine IERR indicates errors in input codes.

3.2.18.12a Flowchart



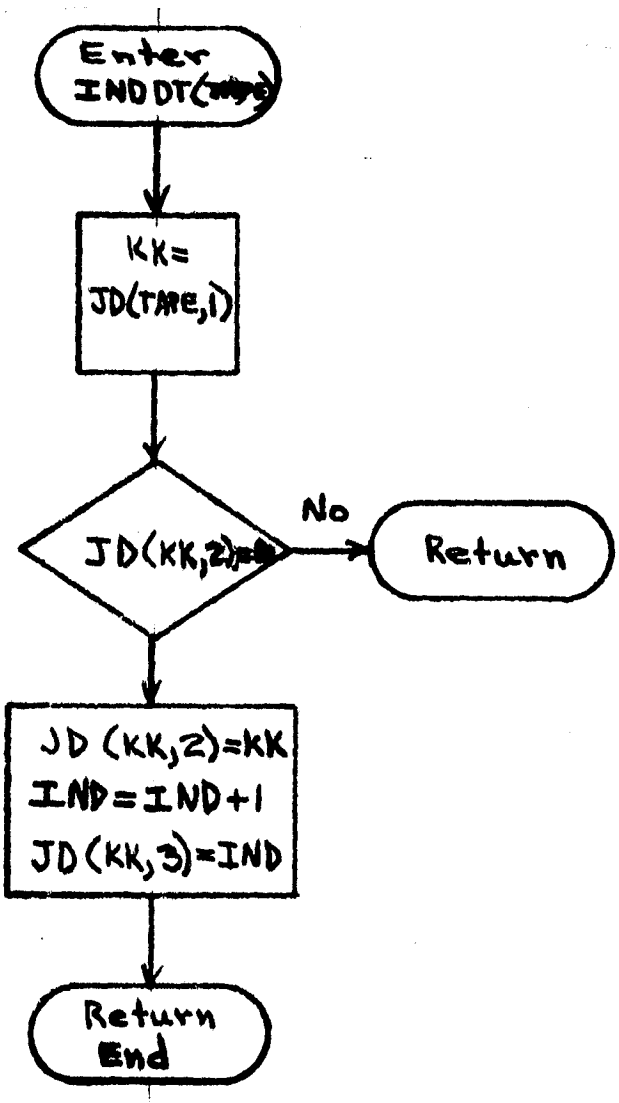
3.2.18.12b

```
      SUBRØUTINE IERR(II,JJ)
      CØMMØN /MPI/MIND ,RT,NPRT,TIND,NAID
      IF(II.LT.1)                    WRITE(NPRT,825)  II
      IF             (II.GT.256)     WRITE(NPRT,825)  II
      IF(JJ.LT.1)                    WRITE(NPRT,826)  JJ
      IF(JJ.GT.256)                  WRITE(NPRT,826)  JJ
      IF(II.LT.1.ØR.II.GT.256)       STØP
      IF(JJ.LT.1.ØR.JJ.GT.256)       STØP
  825 FØRMAT(1HØ,10X,'II=',I5)
  826 FØRMAT(1HØ,10X,'JJ=',I5)
      RETURN
      END
```

3.2.18.13 Subroutine INDDT

Subroutine INDDT records the occurance of cluster numbers.

3.2.18.13a Flowchart



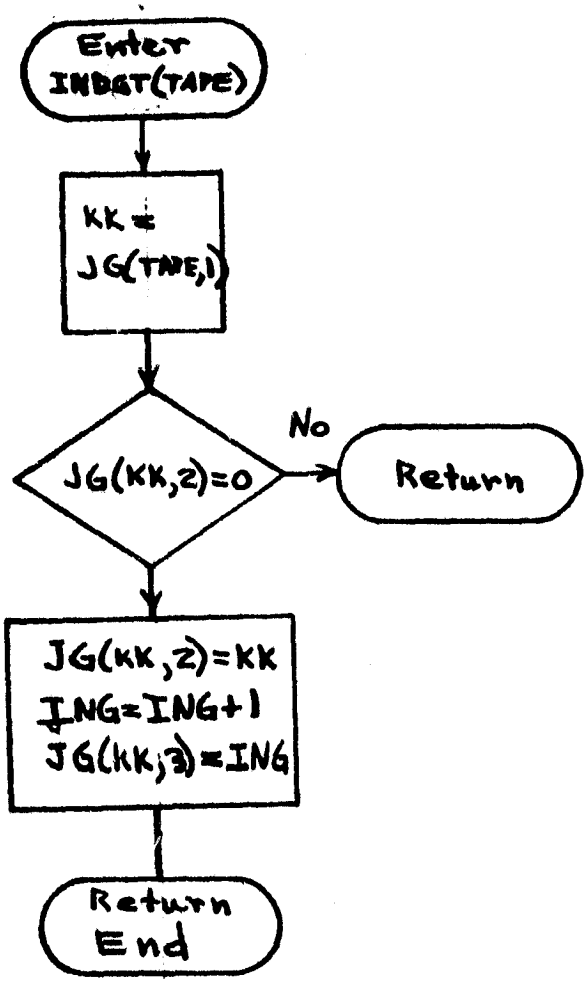
3.2.18.13b

```
      SUBROUTINE INDDT(TAPE)
          IMPLICIT INTEGER(A-Z)
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
          KK=JD(TAPE,1)
          IF(JD(KK,2).NE.0) RETURN
      IND=IND+1
          JD(KK,2) = KK
          JD(KK,3) = IND
          RETURN
          END
```

3.2.18.14 Subroutine INDGT

Subroutine INDGT records the occurance of ground truth numbers.

3.2.18.14a Flowchart



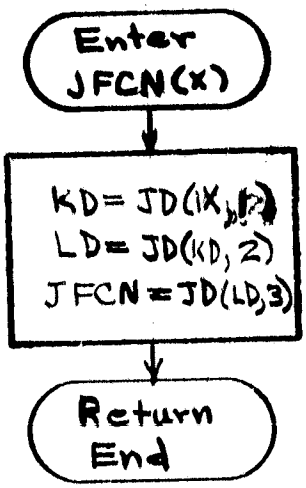
3.2.18.14b

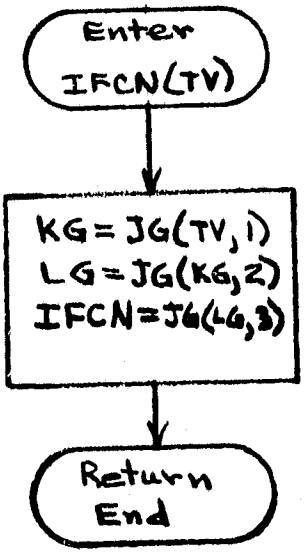
```
      SUBROUTINE INDGT(TAPE)
         IMPLICIT INTEGER(A-Z)
      COMMON /CK/JG(256,4),JD(256, 4),ING,INU
         KK=JG(TAPE,1)
         IF(JG(KK,2),NE,0) RETURN
      ING=ING+1
         JG(KK,2) = KK
         JG(KK,3) =ING
      RETURN
      END
```

3.2.18.15 Functions JFCN and IFCN

Functions JFCN and IFCN point to the correct location in the count matrix.

3.2.18.15a Flowcharts



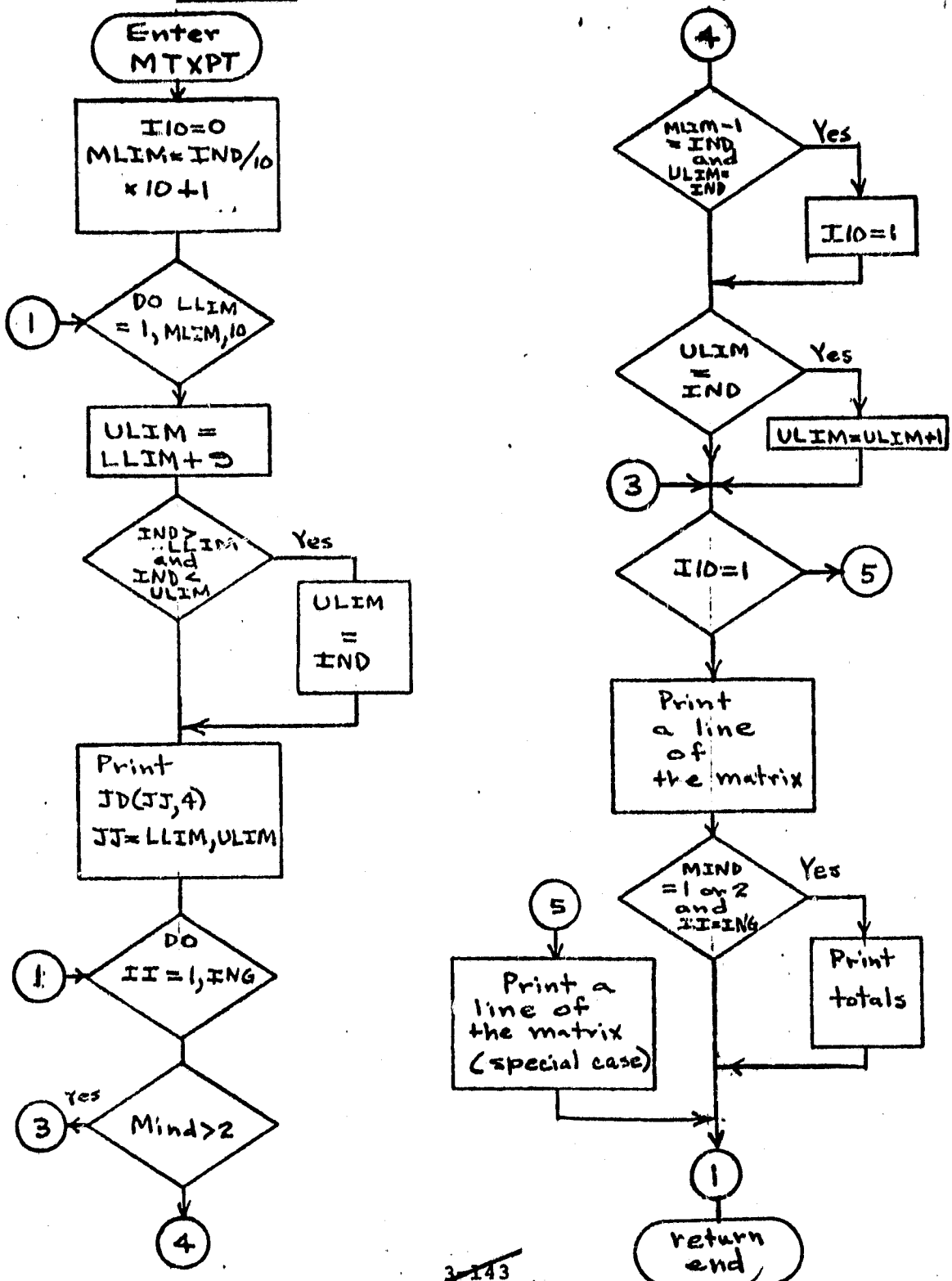


```
      FUNCTION JFCN(X)
      IMPLICIT INTEGER (A-Q),(S-Z)
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      KD=JD(X,1)
      LD=JD(KD,2)
      JFCN=JD(LD,3)
      RETURN
      END

      FUNCTION IFCN(TV)
      IMPLICIT INTEGER (A-Q),(S-Z)
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      KG=JG(TV,1)
      LG=JG(KG,2)
      IFCN=JG(LG,3)
      RETURN
      END
```

3.2.18.16 Subroutine MTXPT

Subroutine MTXPT prints probability and count matrices.

3.2.18.16a Flowchart



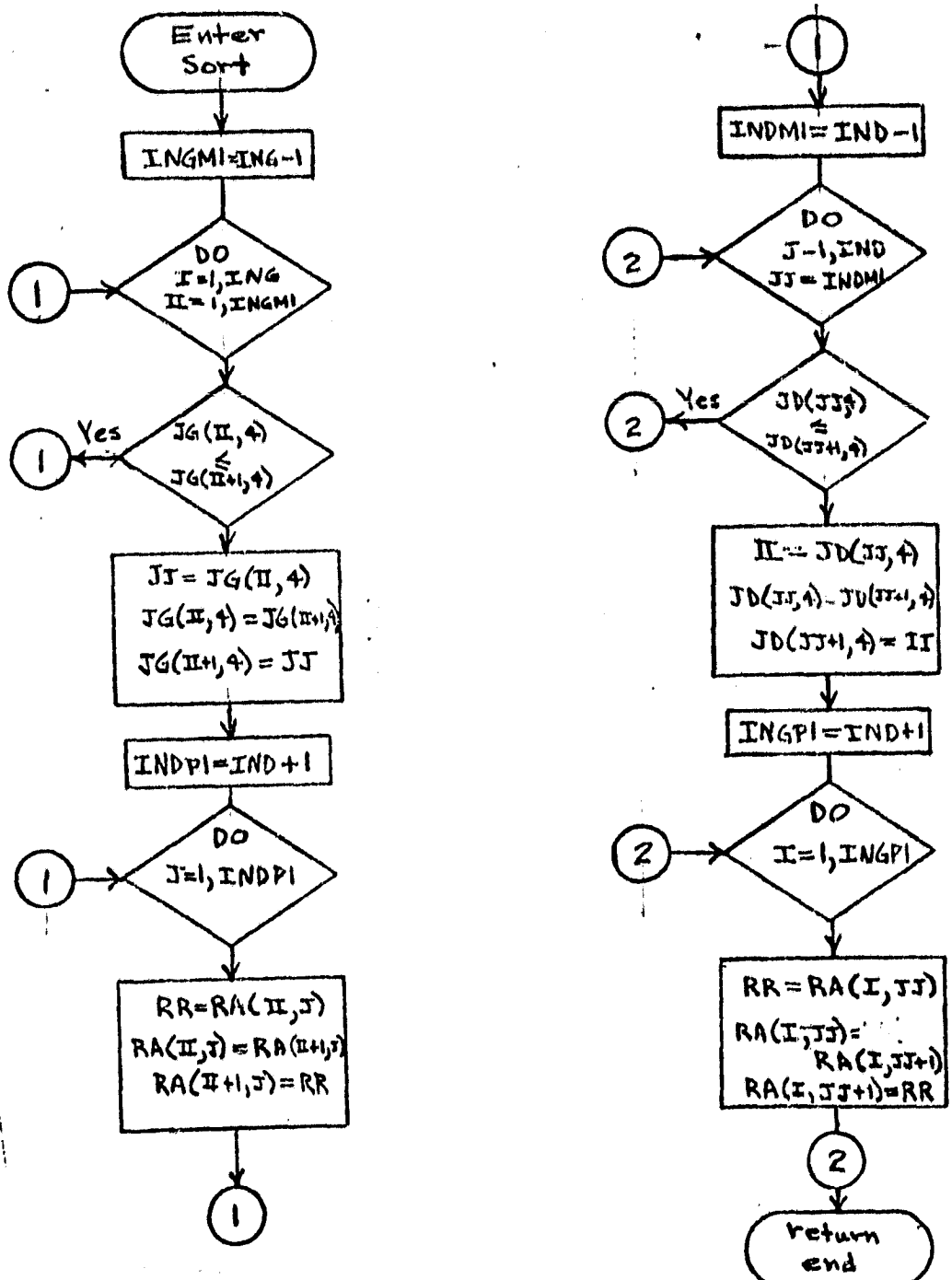
3.2.18.16b

```
      SUBROUTINE MTXPT
      IMPLICIT INTEGER (A-Q),(S-Z)
      COMMON /MPI/MIND ,RT,NPRT,TIND,NAID
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      COMMON /MTX/RA( 50,50)
      I10=0
      MLIM=(IND/10)*10+1
      DO 666 LLIM=1,MLIM,10
      ULIM=LLIM+9
      IF(IND.GE.LLIM.AND.IND.LT.ULIM)  ULIM=IND
      WRITE(NPRT,751) (JD(JJ,4),JJ=LLIM,ULIM)
  751 FORMAT(1H0,10X,10(3X,I5,2X))
      DO 652 II=1,ING
         IF (MIND.GT.2) GO TO 200
      IF(MLIM=1.EQ.IND.AND.ULIM.EQ.IND) I10=1
      IF(ULIM.EQ.IND) ULIM=ULIM+1
  200    CONTINUE
      IF(I10.EQ.1) GO TO 400
      IF(MIND.EQ.1) WRITE (NPRT,653) JG(II,4),(RA(II,JJ),JJ=LLIM,ULIM)
      IF(MIND.EQ.2) WRITE (NPRT,253) JG(II,4),(RA(II,JJ),JJ=LLIM,ULIM)
      IF(MIND.EQ.4) WRITE (NPRT,253) JG(II,4),(RA(II,JJ),JJ=LLIM,ULIM)
       IF (MIND.EQ.3) WRITE(NPRT,253) JG(II,4), (RA(II,JJ)/RA(II,IND+1),
     * JJ=LLIM,ULIM)
      IF(MIND.EQ.1.AND.II.EQ.ING)    WRITE(NPRT,356) (RA(ING+1,JJ),JJ=
     * LLIM,ULIM)
      IF(MIND.EQ.2.AND.II.EQ.ING)    WRITE(NPRT,256) (RA(ING+1,JJ),JJ=
     * LLIM,ULIM)
  356 FORMAT(1H0,10X,10F10.0)
  256 FORMAT(1H0,10X,10F10.5)
  653 FORMAT(1H0, 5X,I5,10F10.0)
  253 FORMAT(1H0, 5X,I5,10F10.5)
      GO TO 652
  400 CONTINUE
      IF(MIND.EQ.1) WRITE (NPRT,643) JG(II,4),(RA(II,JJ),JJ=LLIM,ULIM)
      IF(MIND.EQ.2) WRITE (NPRT,243) JG(II,4),(RA(II,JJ),JJ=LLIM,ULIM)
      IF(MIND.EQ.1.AND.II.EQ.ING)    WRITE(NPRT,346) (RA(ING+1,JJ),JJ=
     * LLIM,ULIM)
      IF(MIND.EQ.2.AND.II.EQ.ING)    WRITE(NPRT,246) (RA(ING+1,JJ),JJ=
     * LLIM,ULIM)
  346 FORMAT(1H0,10X,11F10.0)
  246 FORMAT(1H0,10X,11F10.5)
  643 FORMAT(1H0, 5X,I5,11F10.0)
  243 FORMAT(1H0, 5X,I5,11F10.5)
  652 CONTINUE
         WRITE (NPRT,100)
  100    FORMAT  (1H0)
         IF(I10.EQ.1) GO TO 500
         IF(MLIM=1.EQ.IND.AND.IND.EQ.ULIM) GO TO 500
  666 CONTINUE
  500    CONTINUE
      RETURN
      END
```

3.2.18.17 Subroutine SORT

Subroutine SORT orders the crop codes, the cluster numbers and their counts.

3.2.18.17a Flowchart



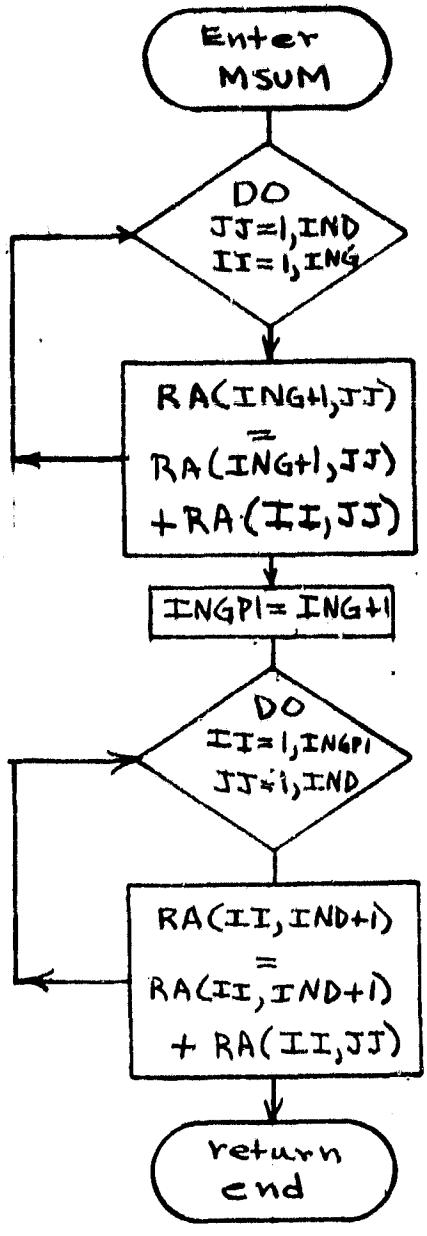
3.2.18.17b

```
      SUBROUTINE SORT
      IMPLICIT INTEGER (A-Q),(S-Z)
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      COMMON /MTX/RA( 50,50)
         INGM1=ING-1
         DO 10 I=1,ING
         DO 10 II=1,INGM1
         IF(JG(II,4).LE.JG(II+1,4)) GO TO 10
         JJ = JG(II,4)
         JG(II,4) = JG(II+1,4)
         JG(II+1,4) =JJ
         INDP1=IND+1
         DO 20 J=1,INDP1
         RR = RA (II,J)
         RA(II,J) = RA(II+1,J)
         RA(II+1,J)=RR
   20    CONTINUE
   10    CONTINUE
         INDM1=IND-1
         DO 30 J=1,IND
         DO 30 JJ=1,INDM1
         IF (JD(JJ,4).LE.JD(JJ+1,4))GO TO 30
         II=JD(JJ,4)
         JD(JJ,4)=JD(JJ+1,4)
         JD(JJ+1,4)=II
         INGP1=ING+1
         DO 40 I=1,INGP1
         RR=RA(I,JJ)
         RA(I,JJ)=RA(I,JJ+1)
         RA(I,JJ+1) =RR
   40    CONTINUE
   30    CONTINUE
         RETURN
         END
```

3.2.18.18 Subroutine MSUM

Subroutine MSUM computes column and row totals for the count and matrix.

3.2.18.18a Flowchart



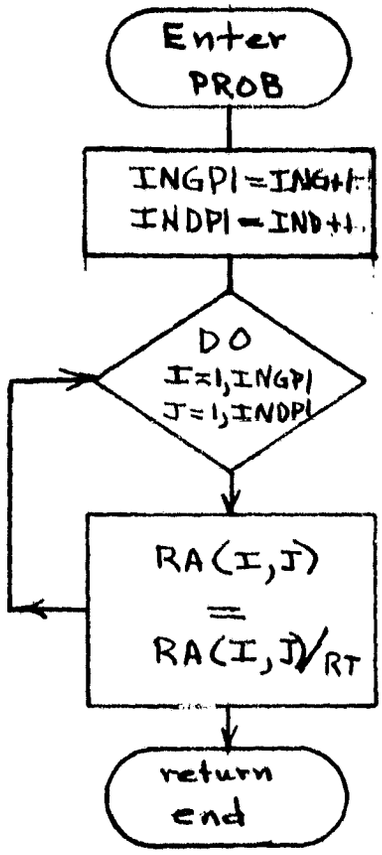
3.2.18.18b

```
      SUBROUTINE MSUM
      IMPLICIT INTEGER (A-O),(S-Z)
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      COMMON /MTX/RA( 50,50)
      DO 651 JJ=1,IND
      DO 651 II=1,ING
      RA(ING+1,JJ)=RA(ING+1,JJ)+RA(II,JJ)
  651 CONTINUE
      INGP1=ING+1
      DO 650 II=1,INGP1
      DO 650 JJ=1,IND
      RA(II,IND+1)=RA(II,IND+1)+RA(II,JJ)
  650 CONTINUE
      RETURN
      END
```

3.2.18.19 Subroutine PROB

Subroutine PROB changes the count matrix into joint probabilities

3.2.18.19a Flowchart



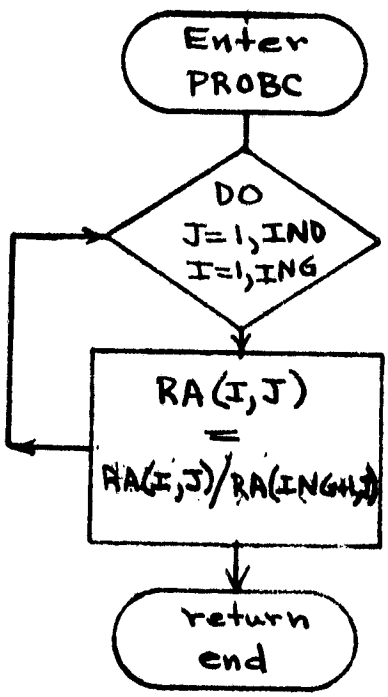
3.2.18.19b

```
      SUBROUTINE PR08
      IMPLICIT INTEGER (A-O),(S-Z)
      COMMON /MPI/MIND ,RT,NPRT,TIND,NAID
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      COMMON /MTX/RA( 50,50)
      INGP1=ING+1
      INDP1=IND+1
      DO 600 I=1,INGP1
      DO 600 J=1,INDP1
      RA(I,J) =RA(I,J)/RT
  600 CONTINUE
      RETURN
      END
```

3.2.18.20 Subrotuine PROBC

Subroutine PROBC computes conditional probabilities.

3.2.18.20a Flowchart



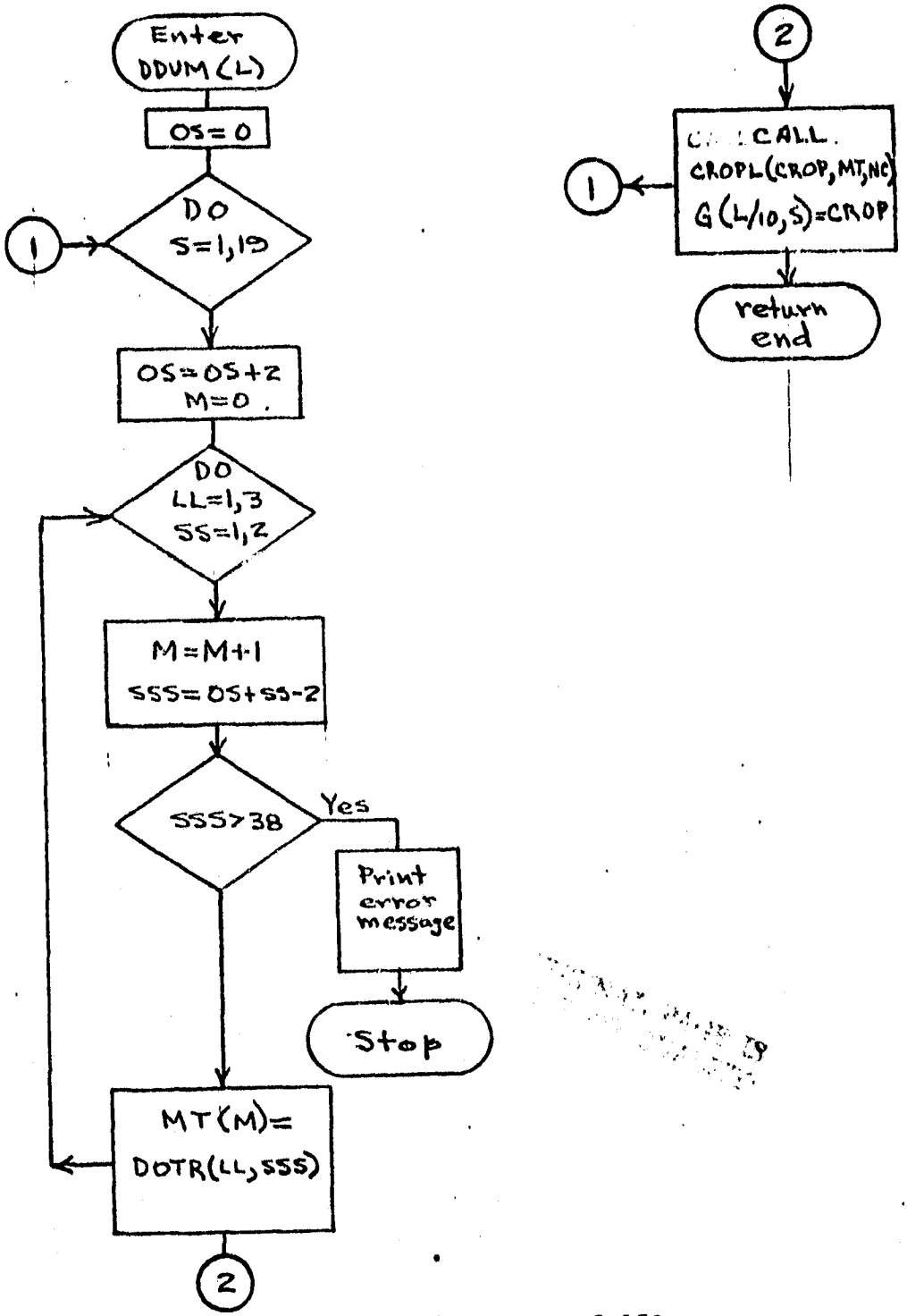
3.2.18.20b

```
      SUBROUTINE PROBC
      IMPLICIT INTEGER (A-Q),(S-Z)
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      COMMON /MTX/RA( 50,50)
      DO 10 J=1,IND
      DO 10 I=1,ING
      RA(I,J) = RA(I,J)/RA(ING+1,J)
10    CONTINUE
      RETURN
      END
```

3.2.18.21 Subroutine DDUM

Subroutine DDUM puts the dot labels into the matrix G.

3.2.18.21a Flowchart



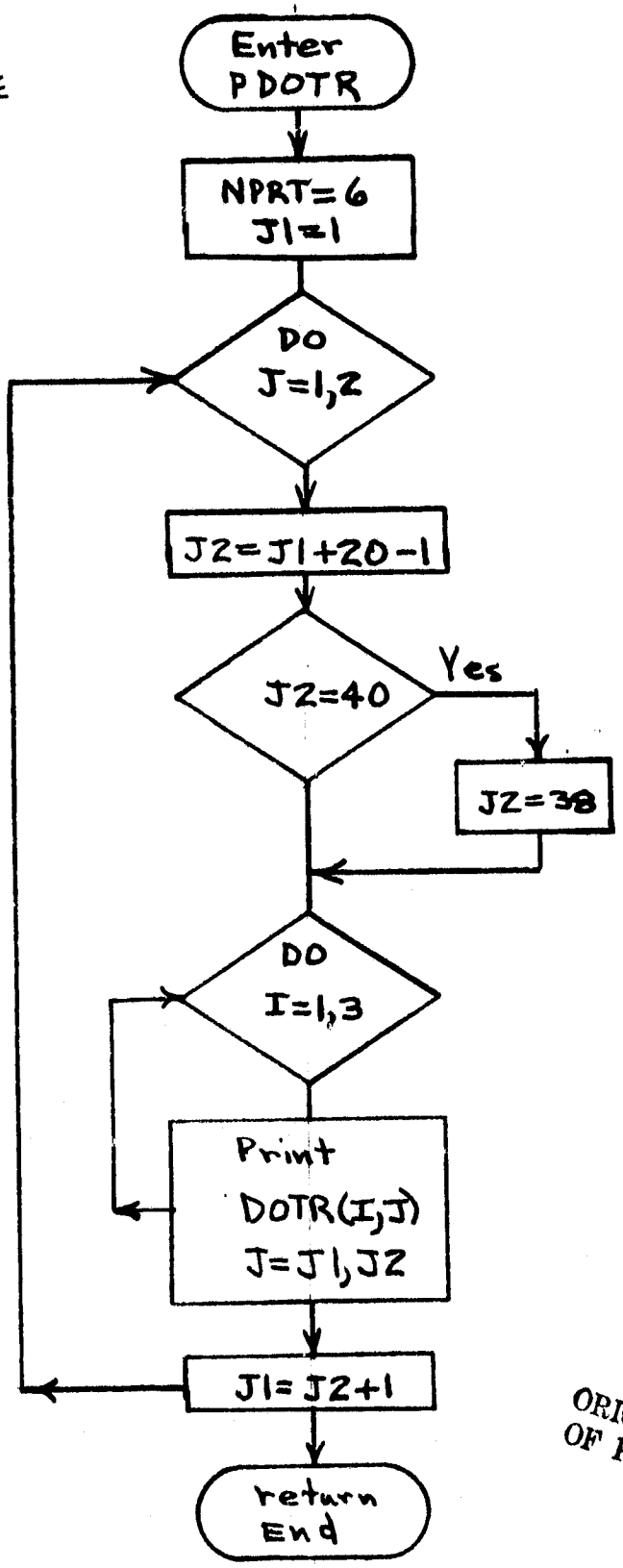
3.2.18.21a

```
      SUBROUTINE DDUM(L)
      IMPLICIT INTEGER (A-Z)
      DIMENSION MT(6)
      COMMON /DD/DOTR(3,38)
      COMMON /RD/A(11,19),G(11,19),DT(11,19)
      COMMON /MPI/MIND ,RT,NPRT,TIND,NAID
      OS=0
      DO 100 S=1,19
      OS=OS+2
      M=0
      DO 200 LL=1,3
      DO 200 SS=1,2
      M=M+1
      SSS=OS+SS-2
      IF(SSS.GT.38) WRITE(NPRT,300) SSS
      IF(SSS.GT.38) STOP
  300 FORMAT(1H0,10X,'SSS=',I5)
      MT(M)=DOTR(LL,SSS)
  200 CONTINUE
      CALL CROPL(CROP,MT,NC)
      G(L/10,S)=CROP
  100 CONTINUE
      RETURN
      END
```

3.2.18.22 Subroutine PDOTR

Subroutine PDOTR prints the labels for the subpixels that make up the dots.

3.2.18.22a Flowchart



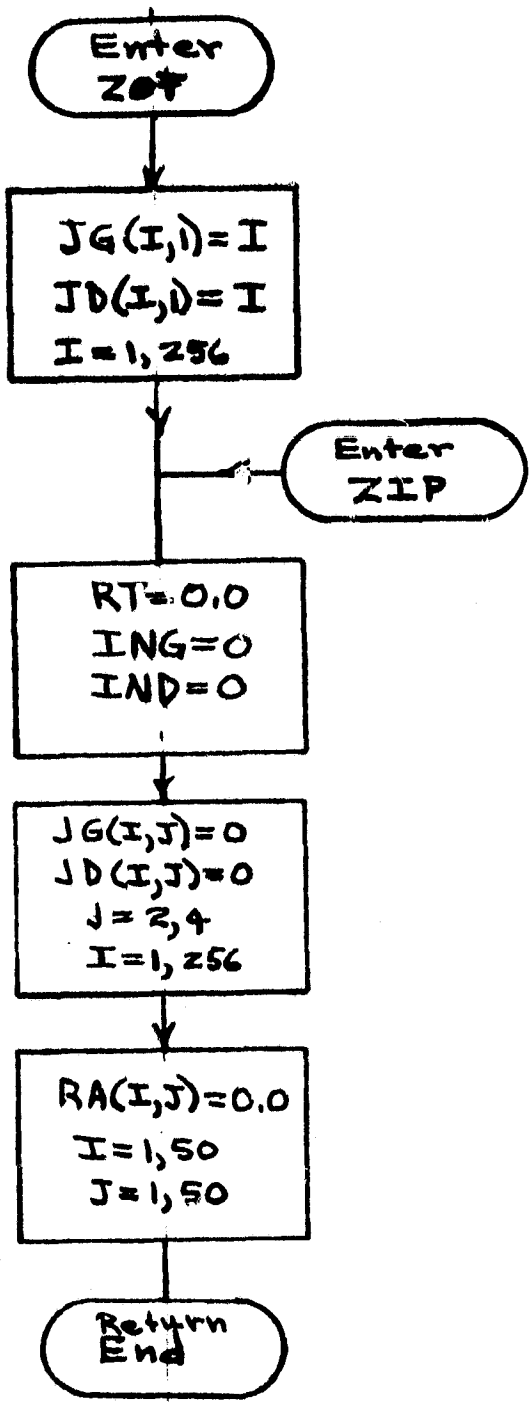
3.2.18,22b

```
      SUBROUTINE PDOTR
      IMPLICIT INTEGER (A-Z)
      COMMON /DD/DOTR(3,38)
      NPRT=6
      J1=1
      DO 200 JJ=1,2
      J2=J1+20-1
      IF(J2.EQ.40) J2=38
      DO 100 I=1,3
      WRITE(NPRT,101) (DOTR(I,J),J=J1,J2)
  101 FORMAT(1H ,10X,20(2I4,2X))
  100 CONTINUE
      WRITE(NPRT,102)
  102 FORMAT(1H0)
      J1=J2+1
  200 CONTINUE
      WRITE(NPRT,102)
      RETURN
      END
```

3.2.18.23 Subroutines ZOT and ZIP

Subroutines ZOT and ZIP initialize.

3.2.18.23a Flowchart



3.2.18.23b

```
      SUBROUTINE ZOT
      IMPLICIT INTEGER (A-G),(S-Z)
      COMMON /MPI/MIND ,RT,NPRT,TIND,NAID
      COMMON /CK/JG(256,4),JD(256, 4),ING,IND
      COMMON /MTX/RA( 50,50)
      DO 50 I=1,256
      JG(I,1)=I
      JD(I,1)=I
   50 CONTINUE
      ENTRY ZIP
      RT=0.0
      ING=0
      IND=0
      DO 850 I=1,256
      DO 850 J=2,4
      JG(I,J)=0
      JD(I,J)=0
  850 CONTINUE
      DO 851 I=1,50
      DO 851 J=1,50
  851 RA(I,J)=0.0
      RETURN
      END
```

4. OPERATIONS

This section presents all information necessary to insure proper execution of the various elements of the Accuracy Assessment software system.

4.1 OPERATORS GUIDE

This paragraph describes the required system configuration and operating procedures required for application of the various elements of the Accuracy Assessment software system. These are provided in sufficient detail to insure proper application of the system by the usual Accuracy Assessment system user.

4.1.1 EQUIPMENT SETUP

In general, the required hardware configuration is the PDP 11/45 computer with an RSX-11D operating system, along with appropriate peripherals. In general, these peripherals will be the standard tape drives (one or two, depending on the software element), disk drives for system software and data storage and an output print unit. Details of tape and disk utilization for each of the software elements are provided in the following paragraph describing "setups".

4.1.2 PROGRAM "SETUPS"

The required setup for proper execution of each of the major Accuracy Assessment software system elements are presented below.

4.1.2.1 Preprocessor DTERM Setup

- Mount a DTRM tape (foreign) without a write ring.
- Input a tape mount card into the DTERM·DAT data set. That card (formatted; A1, 1X, I2) should be as follows:

First entry = MT (Unit type designation)

Second entry = integer value (unit number)

The first entry should be followed by a blank.

Example: MT B 0b1

4.1.2.2 Preprocessor BTREAD Setup

- Mount a "Bendix 100" output tape (foreign) without a write ring.
- Mount a scratch tape with a write ring for output.
- Put two tape mount cards formatted (A1, IX, 212) into the BTREAD·DAT data set; the first corresponding to "Bendix 100" tape (input) and the second for the scratch tape to contain the BTREAD output "B100" tape simulation. The contents of these two cards is:

  First entry = MT (Unit type designation)

  Second entry = Integer value (unit number)

  Third entry = Integer value (tape file number)

  A blank must follow the first two entries:

  Example: MT0113

- An ID card (formatted 4I5) must be input behind and with the two tape mount cards. The contents of this card are:

  First entry = Four digit segment number
  Second entry = Two digit acquisition day of month
  Third entry = Two digit acquisition month
  Fourth entry = Two digit acquisition year - 1900
  Blanks must follow the first three entries.
  Example: 1011b05b07b77

4.1.2.3 Optional Utility SGMAP Setup

- Mount the Ground Truth tape output product of Phase 2 (foreign) without a write ring.
- Put a tape mount cart formatted (A1, 1X, 212) into the SIGMAP•DAT data set. The contents of that card should be:

  First entry = MT (unit type designation)

  Second entry = Integer value (unit number)

  Third entry = Integer value (tape file number)

  Example: MT0106

- A set of "codes to code" cards (formatted 315) should be entered as bulk of the MAP•DAT set. The contents of each card are:

  First entry = Integer (start value of code range)

  Second entry = Integer (end value of code range)

  Third entry = Integer (code assigned to all points within the range specified by the first and second entries in the card).

  A single card containing 0 0 -1 as the three entries should be input as the MAP•DAT card set if no code-to-code transformation is desired. In any event, the MAP•DAT data should be terminated with a blank card.

- Two additional card entries formatted A1 are entered with the MAP•DAT data set. These are:

  1. Data type card containing either:

     a. GT (Ground Truth data to be output)

     b. DTRM (DTERM data to be output)

     c. SPØT (Subset of ground truth to be output)

1. With this option another card is input with the coordinates of the upper left hand corner of this subset.
2. Output type card containing either:
   a. MAP (output is to be a map)
   b. NUM (output is to be a numerical dump)

4.2.1.4 First Unit, First Module, Phase 1 Setup

- Mount the tape product of a previous BTREAD execution.
- Put a tape mount card (formatted A1, 1X, 2I2) into the PHASE1·DAT data set. The card contents are:

  First entry = MT (unit type designation)

  Second entry = Integer (unit number)

  Third entry = Integer (number of tape file to be processed)

  The first and second entries should be followed by a blank.

4.2.1.5 Second Unit, First Module, Phase 2 Setup

- Mount a scratch tape (foreign) with write ring to contain output ground truth data.
- Put a corresponding tape mount card (formatted A1, 1X, 2I2) containing:

  First entry = MT (unit type designation)

  Second entry = Integer value (unit number)

  Third entry = Integer value (tape file number)
- Put an ID card (formatted 3I5) into the LABEL·DAT data set. The contents of this initial card of the data set are:

  First entry = Up to five digit segment number

  Second entry = Two digit acquisition day of month

  Third entry = Two digit acquisition month of year

Fourth entry = Two digit axquisition year - 1900

- A set of ground truth label cards (formatted 315) each containing:

  First entry = Integer (number of field starting sequence of fields with common crop code)

  Second entry = Integer (number of field ending sequence of fields with common crop code)

  Third entry = Crop code assigned to all fields in the card defined sequence to replace the analyst assigned codes

  = -1 if field numbers are to be used

- LABEL1·DAT made by BTREAD map be put into LABEL·DAT an alternative.

4.2.1.6 First Unit, Second Module SPATL Setup

- Mount the ground truth tape product of a previous execution of Phase 2 (foreign) without a write ring.
- Put a corresponding tape mount card (formatted A1, 1X, 212) into the SPATL·DAT data set containing:

  First entry = MT (unit type designation)

  Second entry = Integer value (unit number)

  Third entry = Integer value (number of tape file to be processed)

- Entry a set of codes-to-code cards (formatted 315) into the SPATL·DAT data set, each containing:

  First entry = Integer (start value of code range)

  Second entry = Integer (end value of code range)

  Third entry = Integer (code to be assigned to all points within the range specified by the first and second card entries)

= -1 if analyst assigned value (tape value) is to be used

- Enter a set of A1 transformation cards (formatted 1A1, 4X, 15) into the SPATL•DAT data set, each containing:

  First entry = Alpha classification symbol

  = X if end of file EØF

  Second entry = Integer code equivalent

- Enter a set of A1 label cards (formatted 10X, 12, iX, 12, 1X, A1) into the A1•DAT data set, each containing:

  First entry = Integer point line number

  Second entry = Integer point number in line

  Third entry = Alpha classification symbol

  End of file (EØF) is denoted by a blank card.

4.2.1.6 Second Unit, Second Module ALLCRP Setup

- Mount a ground truth tape (foreign) product of a previous execution of Phase 2 without a write ring.
- Mount a DTRM tape (foreigh) without a write ring.
- Put two tape mount cards (formatted A1, 1X, 212), the first corresponding to the ground truth tape and the second to the DTRM tape, into the ALLCRP•DAT data set and each containing:

  First entry = MT (unit type designation)

  Second entry = Integer (unit number)

  Third entry = Integer (number of file in tape to be processed)

- Enter a set of A1 transformation cards (formatted A1, 4X, 15) into the ALLCRP•DAT data set, each containing:

  First entry = Alpha (classification symbol)

  = X, if end of file (EOF)

Second entry = Integer code equivalent

= blank if end of file (EØF)

- Enter an a-priori probability card (formatted F10.2) in the ALLCRP·DAT set following the A1 transformation cards. This card will contain the numeric value of the assumed a-priori probability as the sole entry.

- Enter a set of codes-to-code cards (formatted 315) for the ground truth, each containing:

  First entry = Integer (start value of code range)

  Second entry = Integer (end value of code range)

  Third entry = Integer (code to be assinged to all points within the range specified by the first and second card entries)

  = -1 if ground truth assigned value (tape value) is to be used

- Enter a set of codes-to-code cards (formatted 315) for the DTRM file, each containing:

  First entry = Integer (start value of code range)

  Second entry = Integer (end value of code rnage)

  Third entry = Integer (code to be assigned to all points within the range specified by the first and second card entries)

  = -1 if cluster number (tape value) is to be used

- Enter a set of A1 label cards (formatted 10X, 12, 1X, 12, 1X, A1) into the A1·DAT data set, each containing:

  First entry = Integer (line number of dot)

Second entry = Alpha (analyst label symbol for dot)

Third entry = Alpha (analyst label symbol for dot)

A blank card entry denotes end of file (EOF).

- Put a loop card (formatted A1, 4X, 15) into the LOOP•DAT data set containing:

  First entry = LØØP

  Second entry = Integer denoting the number of files (arbitrary but nominally 3) to be processed in the sequence of MLTCRP executions.

- Enter a set of A1 label cards (formatted 10X, 12, 1X, 12, 1X, A1) into the A1•DAT data set, each containing:

  First entry = Integer (line number of dot)

  Second entry = Integer (number of dot in line)

  Third entry = Alpha (analyst label symbol for dot)

  A blank card entry denotes end of file (EOF).

- Put a loop card (formatted A1, 4X, 15) into the LOOP•DAT data set containing:

  First entry = LØØP

  Second entry = Integer denoting the number of files (arbitrary but nominally 3) to be processed in the sequence of ALLCRP executions.

4.2.1.7 Second Unit, Second Module MLTCRP Setup

- Mount a ground truth tape (foreign) product of a previous execution of Phase 2 without a write ring.
- Mount a DTRM tape (foreign) without a write ring.
- Put two tape mount cards (formatted A1, 1X, 212), the first corresponding to the ground truth tape and the second to the DTRM tape, into the MLTCRP•DAT data set and each containing:

First entry = Alpha (classification symbol)

= X, if end of file (EOF)

Second entry = Integer code equivalent

= blank if end of file (EØF)

4.1.3 START UP PROCESSING

Following proper program setup as specified in the previous paragraph, software activation is according to usual user procedures.

4.1.4 OPERATING INSTRUCTIONS

Not Applicable

4.1.5 TAKE DOWN INSTRUCTIONS

Not Applicable

4.2 USERS GUIDE

Use of the Accuracy Assessment Software System is restricted to a small, highly specialized group of people. Each member of this group is, or will be, thoroughly trained in the use of the system by way of an individualized, hands-on training program. In view of this fact, the requirement for formal user instruction documentation is minimal and is fully satisfied by the preceding paragraphs and the following functional description.

The normal functional flow of the Accuracy Assessment Software system is as depicted in Figure 4-1. The functions of the various elements of that system are presented below, in the order their occurance in that flow.

ACCURACY ASSESSMENT SOFTWARE SYSTEM FLOW (NOMINAL)

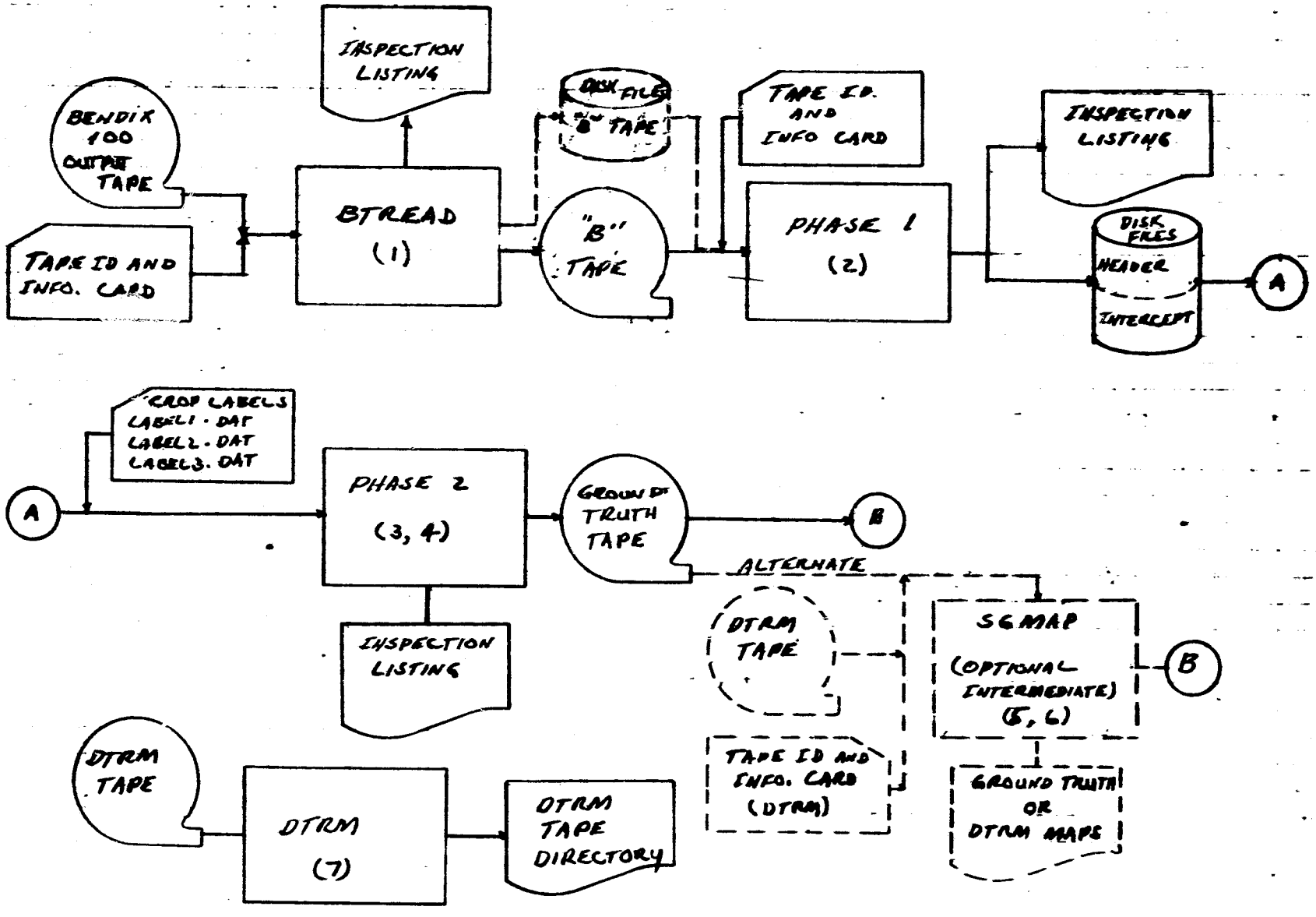
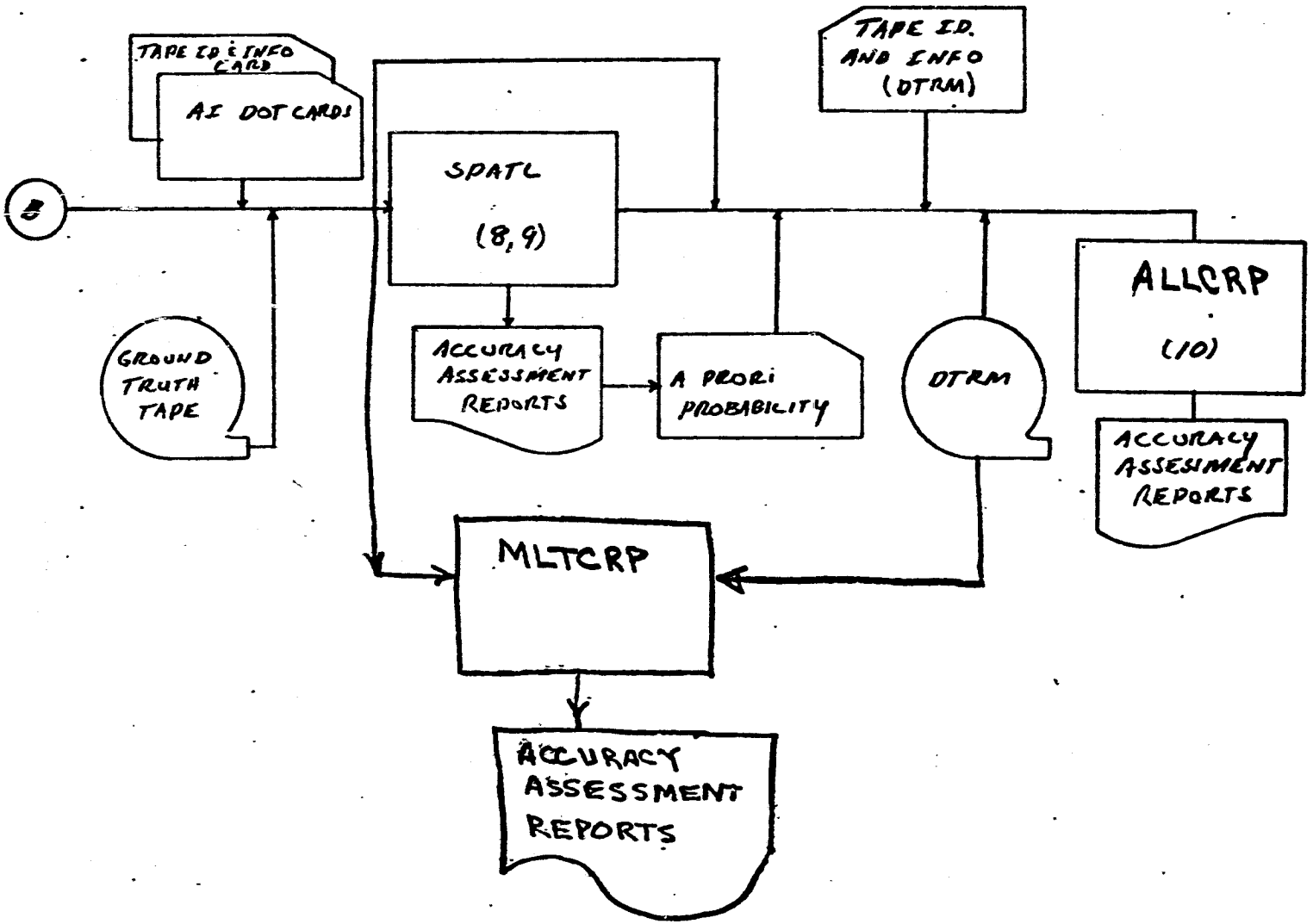


Figure 4-1

ACCURACY ASSESSMENT SOFTWARE SYSTEM FLOW (NOMINAL) CONT.



4.2.1 BTREAD FUNCTION

BTREAD operates on ground truth field vertices data produced and output to magnetic tape by the "Bendix 100" system. It converts those data from NOVA floating point to DEC integer equivalents, than organizes them into the appropriate files LABEL1·DAT, LABEL2·DAT and LABEL3·DAT, which are output to tape for input to the following Phase 1 element.

4.2.2 PHASE 1 FUNCTION

Phase 1 operates on the data files produced by BTREAD to define ground truth field boudary intercepts with dot lines. These intercepts are then structured into files for direct entry to the following Phase 2. Intercept definition is by way of a fixed sequence of operations, each accomplished in Phase 1 by a special subroutine. That sequence of subroutine operations is as follows.

4.2.2.1 S-01 FUNCTION

Buffers input coordinates of intercepts and locates maximum and minimum y coordinates.

4.2.2.2 S-12 FUNCTION

Removes redundant points from the x and y coordinate arrays to insure that there are no more than two (contiguous) points per line.

4.2.2.3 S-23 FUNCTION

Inserts redundant points at maxima, minima and inflections of the field boudaries.

4.2.2.4 S-34 FUNCTION

Fills in missing field boundary segments

4.2.2.5 S-45 FUNCTION

Connects all intercepts within given lines.

4.2.2.6 S-55 FUNCTION

Puts intercepts in ascending order.

4.2.2.7 S-56 FUNCTION

Packs intercepts into a one dimensional buffer (INTCPT·DAT) for direct entry to the Phase 2 element.

In addition to the above functions, Phase 1 organizes data identification information derived from card inputs into a header data file (HEAD·DAT) for direct input to Phase 2.

4.2.3 PHASE 2 FUNCTION

Phase 2 operates on the data file products of a previous Phase 1 operation; HEAD·DAT and INTCPT·DAT, along with the labeling data files (LABEL1·DAT, LABEL2·DAT, and LABEL3·DAT) produced by a corresponding execution of BTREAD. Any or all of LABEL(i)·DAT, i = 1, 2 or 3, may be put into the LABEL·DAT file through a PIP operation, eliminating the requirement for card input of labeling data. The resulting LABEL·DAT data sets are:

- All crop classes - from LABEL1·DAT
- Small grains and other - from LABEL2·DAT
- Field numbers - from LABEL3·DAT

In each case a file is PIPed into LABEL·DAT and a corresponding file number change must be made to the tape mount card in PHASE1·DAT.

The end product of PHASE 2 is a wall-to-wall mapping of ground truth data in Universal format, output as a "ground truth" tape.

4.2.4 SGMAP FUNCTION

At the option of the user printout maps of the "ground truth" data output of PHASE 2 may be generated by an application of the utility element SGMAP. SGMAP operates on the data sets MAP1·DAT, MAP2·DAT, and MAP3·DAT which correspond to the PHASE2 LABEL1·DAT, LABEL2·DAT and LABEL3·DAT, respectively. These files are placed into the MAP·DAT data set by use of the PIP facility, with the appropriate file so designated on the tape mount card in the SGMAP·DAT data set. SGMAP then formats the data for block print out as an alpha-numeric map. Each map is accompanied by a map symbol to crop type code table providing full definition of the symbology employed for a given output map. SGMAP also prints out maps of DTERM files as well as subpixel level maps of ground truth files.

4.2.5 SPATL FUNCTION

SPATL operates on the ground truth data product of PHASE 2 and card input A1 dot label data. SPATL compares the A1 dot data with the ground truth data for the production of certain accuracy assessment parameters (see paragraph 3.2.15.4) including the ground truth wheat production which is used for the a priori probability input to ALLCRP

SPATL may be executed either for consideration of all crop classes or for consideration of "small grains and other".

4.2.6 ALLCRP FUNCTION

ALLCRP operates on the "small grains and other" ground truth data, described by its input transformation the SPATL derived

wheat proportion (a prori probability) and corresponding DTRM data to produce certain accuracy assessment parameters. Its products are identified in paragraph 3.2.17.4.

4.2.7 MLTCRP FUNCTION

MLTCRP operates on the ground truth data, the corresponding DTRM data and the A.I. dot labels to produce certain accuracy assessment parameters. Its products are comparisons of DTRM data with the other forms of data.

4.3 MAINTENANCE DOCUMENTATION

Not Applicable

APPENDIX A

ACCURACY ASSESSMENT INPUTS

APPENDIX A

A.1 "BENDIX 100" OUTPUT DATA TAPE

Data are presented as 80 word records on a 9-track, 800 BPI magnetic tape with odd parity. All values are expressed in NOVA floating point and are presented in the following format.

RECORD WORD	WORD CONTENT
1	Field Number (1-499)
2	Number of Vertices +2
3 - 80	Field vertices coordinates (odd numbered words-x; even numbered words-y)

A.2 "B" TAPE

Data are presented as 80 word records on a 9-track 800 BPI magnetic tape with odd parity. All values are expressed in DEC integer form in a format identical with that of the "BENDIX 100" output tape described above.

A.3 DOT LABELING CARD INPUTS

Analyst labeling data are input in punched cards in the following format.

FIELD (Card Cols.)	CONTENT	CHARACTER Type
1-10	Blank	
11-12	Line number	Integer
13	Blank	
14-15	Dot number	Integer
16	Blank	
17-18	Crop class	Alpha
19-31	Blank	
32-33	Not used	Alpha
34	Blank	
35-38	Not Used	Integer
39	Blank	
40-43	Not used	Integer
44	Blank	
45-48	Not used	Integer
49-52	Blank	
53	Not used	Integer
54-57	Blank	
58-64	Not used	Integer
65-67	Blank	
68-72	Not used	Integer
73-80	Blank	

APPENDIX B

ACCURACY ASSESSMENT OUTPUTS

APPENDIX B

B.1 SPATL PRINTED OUTPUTS

SPATL output is in labeled blocks, preceded in the printout by listings of the punched card inputs (codes-to-code table, and A1 dot labeling data) and run identification data (ground truth file, site and acquisition date). The following are the SPATL output data blocks in order of their occurance in the printout.

- Code-to-code Transformation

Each line of the table provides the range of codes to be transformed (beginning and end values) and the code assigned to that range.

- Ground Truth Information for the Whole Segment

Each line of this table presents, for a given code, values for the parameters explained in the following table.

PARAMETER	EXPLANATION
PI(X)	Proportion of pixels in the feature space x of the scene
ND*V(X)	Product of the number of dots and the contribution of pixels in cell x to the sampling variance
PMC(X)	Contribution to the probability of misclassification due to cell x
N(W)	Number of ground truth wheat subpixels
N(Ø)	Number of other ground truth subpixels
NT(W)	Number of subpixel (pixels for cases 2 and 4) of wheat in the data set of comparison
NT(Ø)	Number of subpixels (pixels for cases 2 and 4) from all other classes in the data set of comparison
PH(W)	Probability estimate for the class w
PDØT	Proportion estimate based on classification and pixel counting
ND*V	Sampling variance
PMC	Probability of misclassification

B.2 ALLCRP PRINTED OUTPUTS

ALLCRP output is in labeled blocks preceded by an 80-80 listing of punched Card A1 dot label data described in Appendix A (unused fields are included in this listing). Printout of the output starts with a self explanatory block identifying the tape input and their unit assignments. This is followed by a presentation of the following output data blocks.

- Type to code Transformation

  This is a table presenting the ALLCRP output numeric equivalents for the various alpha crop types.

- A1 Dot Lables

  This is a three-column self explanatory listing of the input A1 dot labels. The block is terminated by a summary of the input data including:

  The number of A1 dots

  The value of the assumed a prori probability

  The identify of the ground truth (GT) and DTRM data (segment number and acquisition date).

PARAMETER	EXPLANATION
X	Crop label
P(X)	Probability of crop x in scene
N(X)	Number of class x subpixels with designated code
PIX n (n = 1,2, ... 6)	Number of pixels containing n subpixels of class x and the designated code

The block is terminated by two self explanatory lines providing data qualification.

- A matrix of Ground Truth Dot Labels

  Codes for 209 ground truth dots.

  Mixed dots denoted by values >1000.

- Ground Truth Information for the 209 Dots.

  Data block format is identical with that for "Ground Truth Information for the whole segment" above.

- Type-to-code Transformation

  Provides numeric value equivalents (SPATL output) to alpha character crop codes.

- A Matrix of AI Dots

  This matrix, preceded by a statement as to the number of input AI dots, is a display of the positions of those AI dots in the 209 dot array. Zero values in the display denote unclassified dots.

- Ground Truth For the AI Dots

  The format of this block is identical with the "Ground Truth Information for the Whole Segment" block explained above except that it is terminated by different self explanatory qualification data.

This final data block is self explanatory

- Data Comparison Data Blocks

ALLCRP computation results are presented in fully labeled data blocks. Each such block is preceded by a title line which identifies the comparison yielding the data in the block. The various data elements of the block are explained in the following table.

PARAMETER	EXPLANATION
X	Crop class
N(X,W)	Number of ground truth subpixels from the wheat class in the DTRM cell x
N(XØ)	Number of ground truth subpixels from other classes in the DTRM cell x
NT(X,W)	Number of subpixels (pixels for cases 2 and 4) form the data set of comparison which are wheat and are in the DTRM cell x
NT(X,Ø)	Number of pixels from the data set of comparison from all other classes which are in the DTRM cell x
PH(X)	Probability estimate for the class x
PT(X/α)	TBD
PH(C_1/C_2)	Estimate of the probability that the class C_1 will occur given that class C_2 has occured (C_1 and C_2 represent classes such as W,Ø).

B.3 MLTCRP PRINTED OUTPUTS

MLTCRP output is in labeled blocks preceded by listing of punched card AI dot labels (described in Appendix A). Following this there is a list of the subpixel labels that make up each dot, then there is a series of count and probability matrices.

- Type to code Transformation

  This is a table presenting the MLTCRP output numeric equivalents for the various alpha crop types.

- A1 Dot Labels

  This is a three-column self explanatory listing of the input A1 dot labels. The block is terminated by a summary of the input data including:

  The number of A1 dots

  The value of the assumed a prori probability

  The identity of the ground truth (GT) and DTRM data (segment number and acquisition date).

- Ground Truth - Code-to-code

  Each line of the table provides the range of codes to be transformed (beginning and end values) and the code assigned to that range (BEGIN=0,END=0, and CODE=-1 is no transformation).

- DTRM Map - Code-to-code Transformation

  Similar in nature to the ground truth code-to-code transformation.

- Configurations of the 209 dots

  This table indicates the subpixel ground truth labels that make each dot.

- Matrices

  Several labeled matrices follow having labels above and to the left (totals are not labeled). $N(X_1,X2)$ is a count matrix indicating the number of joint occurances of X_1 (label type to the left) and X_2 (label type above). $P(X_1,X_2)$ is a matrix of joint probabilities for X_1 and X_2.

  $P(X_1/X_2)$ is a matrix of conditional probabilities. The probability of X_1 given X_2 is true.

  X_1 and X_2 are assigned to be G, D, and AI.

  G - ground truth crop type

  D - class or cluster number

  AI - transformed Analyst dot label.

  These matrices are printed for several possible comparisons at different sampling rates (the whole scene, the 209 dots, and the AI labeled dots).

- The 209 dots

  This matrix indicates for each input type labels for the 209 dots.

- Matrices

  Several labeled matrices follow having labels above and to the left (totals are not labeled). $N(X_1,X2)$ is a count matrix indicating the number of joint occurances of X_1 (label type to the left) and X_2 (label type above). $P(X_1,X_2)$ is a matrix of joint probabilities for X_1 and X_2.

  $P(X_1/X_2)$ is a matrix of conditional probabilities. The probability of X_1 given X_2 is true.

  X_1 and X_2 are assigned to be G, D, and AI.

  G - ground truth crop type

  D - class or cluster number

  AI - transformed Analyst dot label.

  These matrices are printed for several possible comparisons at different sampling rates (the whole scene, the 209 dots, and the AI labeled dots).

- The 209 dots

  This matrix indicates for each input type labels for the 209 dots.