80-10169
JSC- 13972
NASA CR-
160650

"AS-BUILT" DESIGN SPECIFICATION
FOR
LACIE FORMATTED DOT CARDS IN EOD-LARSYS

Job Order 71-593

TIRF (77-0070)

(E80-10169) AS-BUILT DESIGN SPECIFICATION FOR LACIE FORMATTED DOT CARDS IN EOD-LARSYS (Lockheed Electronics Co.) 26 p HC A03/MF A01 CSCL 05B
N80-28770
Unclas
G3/43 00169

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

April 1978 LEC- 12154

JSC- 13972

"AS-BUILT" DESIGN SPECIFICATION
FOR
LACIE FORMATTED DOT CARDS IN EOD-LARSYS

Job Order 71-593

TIRF (77-0070)

Prepared by

P. J. Aucoin Jr.
Jeannie Gor

APPROVED BY

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

Prepared By
Lockheed Electronics Company, Inc.
For
Earth Observations Division
Science and Applications Directorate

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

April 1978 LEC- 12154

CONTENTS

1. SCOPE

This document contains the final design specification for an augumentation to the Procedure 1 EOD-LARSYS system. This addition involves reading starting and/or bias dots in the LACIE format from the DOTDATA processor.

It is implemented on the version of EOD-LARSYS on the Purdue-LARS 370/148 currently under conversion.

2. APPLICABLE DOCUMENTS

- As-Built Design Specification for EOD-LARSYS Procedure 1, JSC 13143.
- TIRF: 77-0070

3. SYSTEM DESCRIPTION

3.1 HARDWARE DESCRIPTION

N/A

3.2 SOFTWARE DESCRIPTION

The DOTDATA processor of the EOD-LARSYS system has been expanded to accept dot (field) cards in the LACIE Procedure 1 format. These cards have the general form:

DOT (TYPE) (CATEGORY) {(LACIE NUMBERS)}

where

DOT: starts in col. 1
TYPE: = 1 or 2
CATEGORY: 1 character category identifier
LACIE NUMBER: integer value from 1 to 209.

This addition requires an additional option read by subprogram SET13 and flagged by an additional variable in the DOTVEC labeled common block. A new routine, FLDLAC, reads and decodes the dot cards.

In order to specify dots not covered by LACIE numbers, a special code is implemented. This code consists of line and sample incrementors added to the LACIE number.

3.2.1 SOFTWARE COMPONENT NO. 1 (SET13)

Subprogram SET13 reads the control cards needed for dot processing.

3.2.1.1 Linkages

SET13 is called by routine DOTDAT and in turn calls utilities NUMBER, NXTCHR, FIND, and ORDER.

3.2.1.2 Interfaces

SET13 interfaces with other routines by means of common blocks DOTVEC, INFORM, and GLOBAL.

3.2.1.3 Inputs

- New/Revised Control Cards

OPTION	LACIE	Turn on LACIE dot format option

3.2.1.4 Outputs

N/A

3.2.1.5 Storage Requirements

TBD

3.2.1.6 Description

The option LACIE is added to the OPTION control card. The variable named LACIE is added to the DOTVEC labeled common block. LACIE is initialized to the value 0. Upon encountering an OPTION LACIE control card, LACIE is reset to the value 1, indicating LACIE type dot (field) cards will follow the *END* card.

3.2.1.7 Flowchart

N/A

3.2.1.8 Listing

See Appendix A.

3.2.2 SOFTWARE COMPONENT NO. 2 (FLDLAC)

The new subprogram FLDLAC reads and decodes the LACIE formatted field (dot) cards.

3.2.2.1 Linkages

FLDLAC will be called by subprogram DOTS if LACIE ≠0   Each call to FLDLAC provides, upon return.

1. a dot (field) description (first return)
2. transfer to dot file writing (second return)
3. transfer to dot file writing (third return)

3.2.2.2 Interfaces

FLDLAC interfaces with other routines through a calling sequence and common blocks DOTVEC and INFORM.

3.2.2.3 Inputs

Calling Sequence:

SUBROUTINE FLDLAC (FIELDS, STAMNT, $, $, $, IPT, VERTEX)

Parameter	Dimension	In/Out	Description
FIELDS	(4,250)	Out	Category name and dot type for dot I stored in FIELDS (1,I) and FIELDS (4,I).
STAMNT	1	In/out	Initially set equal to 1, switch to indicate dots being taken from currently read card.
$			Returns to Dots
$			
$			

Parameter	Dimension	In/Out	Description
IPT	1	In/out	Initially set equal to 0, index number for field vertex information.
VERTEX	1000	Out	Vertex information for each dot.

In addition, FLDLAC stores the FLDINF vector in common block DOTVEC with rectangular co-ordinates of field enclosing each dot field.

3.2.2.4 Outputs

N/A

3.2.2.5 Storage Requirements

TBD

3.2.2.6 Description

The DOTDATA processor has been modified to permit reading and processsing of dot cards of the form

DOT (TYPE) (CATEGORY NAME) ({LACIE NUMBERS}), i.e.,

DOT 1 W 2 5 10 29 32 54 110

The present processing also uses input training field formats. "TYPE" cards are used to prefix a set of dots. This remains as the default option. The association between LACIE numbers and training field coordinates is as follows.

		SAMPLE NUMBER						
		10	20	30	•	•	•	190
	10	1	2	3				19
LINE NUMBER	20	20						38
	30	39						57
	⋮			LACIE NUMBERS				
	110	191						209

Two expansions of the LACIE card format are incorporated. These are

1. Free-field locations of all information cards, cols 1-80, data items separated by at least one blank, with the restriction that DOT identifiers start in col 1, and the dot type appear in column 5.

2. In order to cover pixels not included in the LACIE numbering scheme, input dot numbers will be represented as the numerical equivalents of

   $N = LI*10^8 + SI*10^4 + LACIE$

   where

   LI = #lines to be incremented (up or down) from the line number mapped from the LACIE number. The convention is

LI	negative	to increment up
LI	positive	to increment down
LI	zero	to avoid incrementation

   SI = #samples to be incremented (right or left)

SI	negative	to increment left
SI	positive	to increment right
SI	zero	to avoid incrementation.

For example, LI=2, SI=-3, LACIL = 38 yeilds
$N = 2*10^8 -3*10^4 + 38 = 199970038$
would correspond to the pixel at (187,22), ie, the pixel at sample number 187 and line number 22.
Letting LI= SI=0, LACIE = 38, obtain
N = 38, correspond to the pixel at (190,20).

Reduction of the value of N to sample and line coordinates will proceed as follows.

$N_1 = |N|/10^8$ (truncated to integer)

if $|N|-N1*10^8 \geq 10^7$ set $N_1 = N_1+1$

$LI = N_1 * \text{sign} (N)$

$N_2 = N-LI*10^8$

$N_3 = |N_2|/10^4$ (truncated to integer)

if $|N_2| - N_3*10^4 \geq 10^3$ set $N_3 = N_3+1$

$SI = N_3 * \text{sign} (N_2)$

$LACIE = N_2 - SI*10^4$

LR = LACIE ROW# = ((LACIE-1)/19 + 1) *10

LS = LACIE COL# = 10* (LACIE - ((LR-1)/10)*19)
where truncated divides are specified.

Finally,

L = LR + LI	line number corresponding to N
S = LS + SI	sample (column) number corresponding to N.

In the scheme to follow, each dot is considered to be a field. All type 1 dots will occur prior to type 2 dots, ie, the input cards cannot be scrambled with respect to dot type. Otherwise, arbitrary order to cards and LACIE numbers on each card are permitted.

At present, subprogram FLDTYP, called by DOTS, processes dot cards. This routine is not easily modified to accept the LACIE format. Consequently a new subprogram, FLDLAC, was written and called instead of FLDTYP from DOTS if LACIE ≠0. It is called from DOTS as

```
CALL FLDLAC (FIELDS, STAMNT, $100, $510, $520, IPT, VERTEX)
```

Initialization, at the start of the DOTS routine, invokes

```
                   IPT = 1
If  (LACIE.EQ.1)IPT = 0
                STAMNT = 1
                NOFLD2 = 0
                  TYPE = 1
                 NOCAT = 0
```

Subprogram FLDLAC has the following structure.

```
SUBROUTINE FLDLAC (FIELDS, STAMNT, *, *, *, IPT, VERTEX)
IMPLICIT INTEGER (A-Z)
DIMENSION FIELDS(4,1), VERTEX(1), CARD(62), LDØTS(30)
LØGICAL SWITCH
DATA    SWITCH/.TRUE./,      SWCHG/0/,ENDBCD/$EN/
INCLUDE    CØMBK1                 ( / INFØRM/)
INCLUDE    CMBK14                 ( / DØTVEC/)
```

The function of the various parameters is as follows.

IPT index number for dot (field) vertex information

NOFLD2 number of fields (dots) for dots of current type (common block INFORM)

SWCHG number of times dot type has changed. This must be no greater than 1 or an input error will have occurred.

SWITCH flags a dot type change. The second return will be taken for subsequent writing of a dot field. (internal)

STAMNT if = 1, a new dot card has been read if =2, dots are being processed from a previously read card.

TYPE dot type being processed (common block DOTVEC)

The calling sequence of FLDLAC is the same as that for FLDTYP, and the meaning of FIELDS and VERTEX remains the same.

```
     IF  (STAMNT.EQ.2)  GØ  TØ  30
     IF  (.NOT.SWITCH)  GØ  TØ  20

10   READ A CARD, extract TYPES from column 5
     If (TYPE.EQ.TYPES)  GØ  TØ  20
     If (SWCHG.NE.0)   error exit
     TYPE=TYPES

20   RE-READ CARD, extract
          CATNM      category name
          NDCARD     #dots on this card
          NDØTS(I), I=1,NDCARD     dots on this card
     If (NDCARD.EQ.0)  GØ TØ 10
     ICNT = 0
     STAMNT = 2
     SWITCH = .TRUE.
     GØ TØ 100

30   If (ICNT.LT.NDCARD)    GO TO 100
     STAMNT = 1
     ICNT   = 0
     READ A CARD, extract first 3 characters and store as ID, extract
     TYPES
     IF (ID,EQ.ENDBCD) RETURN 3
     IF (TYPE.EQ.TYPES) Gϕ Tϕ 20
     SWITCH. = .FALSE.
     SWCHG   =  SWCHG+1
     IF (SWCHG.GT.1) error exit
     NϕCAT = 0; TYPE = TYPES
     IPT    = 0
     RETURN 2

100  ICNT = ICNT+1
     NϕFLD2 = NϕFLD2+1
     find sample and line numbers S and L from NDϕTS (ICNT) as
     described previously.
     Store
     FIELDS (1,NϕFLD2) = CATNM
     FIELDS (4,NϕFLD2) = 2
     FLDINF (1) = L  )
     FLDINF (2) = L  )
     FLDINF (3) = 1  }  rectangular bordering field (dot)
     FLDINF (4) = S  )
     FLDINF (5) = S  )
     FLDINF (6) = 1
     IF(IPT.NE.0) GO TO 35
     IPT = -3
 35  IPT = IPT+4
     VERTEX (IPT) = S
     VERTEX (IPT+1) = L
     VERTEX (IPT+2) = S
     VERTEX (IPT+3) = L
     RETURN 1
     END
```

Regarding the extraction of dot numbers NDOTS(I), I=1, NDCARD, a new routine, NUMBR, similar to existing function NUMBER is provided. (See Section 3.2.3.)

3.2.2.7 Flowchart

N/A

3.2.2.8 Listing

See Appendix A.

3.2.3 SOFTWARE COMPONENT NO. 3 (NUMBR)

The new subroutine NUMBR process one input card of information at a time. It recognizes blanks as delimitors and store all numbers in array NDOT(NDCARD). NDCARD will be the total number of dots on that particular CARD.

3.2.3.1 Linkages

NUMBR is called by FLDLAC routine and reference only routine I4A1BN.

3.2.3.2 Interfaces

Interface between NUMBR and FLDLAC and I4A1BN is via the calling arguments.

3.2.3.3 Inputs

Calling Sequence:

Subroutine NUMBR(NDOTS,NDCARD,CARD,COL).

Parameter	Dimension	In/Out	Description
NDOTS	(30)	In/out	Contains all the dots read from one LACIE formatted dot card.
NDCARD	1	In/out	Index to NDOTS of total number of dots read from one CARD.
CARD	(75)	In	Card to read
COL	1	In	Starting col. number of CARD.

3.2.2.4 Outputs

Array NDOTS(NDCARD) of dots read from a card.

3.2.2.5 Storage Requirements

TBD

3.2.3.6 Description

Subroutine NUMBR takes as input array CARD and uses the input COL + 1 as the starting element of CARD. Each element is tested for a blank. If not a blank the element is changed to integer representation by a call to routine I4A1BN. The entire number is collected until a blank is encountered. NDCARD is incremented by one and the number is stored in NDOTS(NDCARD). When the end of the CARD is reached, NUMBR returns to FLDLAC all the dots on the card in array NDOTS with NDCARD as the number of dots it processed.

3.2.3.7 Flowchart

N/A

3.2.3.8 Listing

See Appendix A.

APPENDIX A

PROGRAM LISTINGS

```
C      FIELDS - CATEGORY NAME AND DOT TYPE FOR DOT I STORED IN
C                FIELD(1,I) AND FIELD(4,I)
C      STAMNT - INITIALLY SET TO 1, SWITHCED TO INDICATE DOTS BEING
C                 TAKEN FROM CURRENTLY READ CARD.
C      IPT - INITIALLY SET TO 1,INDEX NUMBER FOR FIELD VERTEX INFORMATION
C      VERTEX - VERTEX INFORMATION FOR EACH DOT.
C
       SUBROUTINE FLDLHC(FIELDS,STAMNT,*,*,*,IPT,VERTEX)
       IMPLICIT INTEGER (A-Z)
       REAL DUM
       DIMENSION FIELDS(4,1),VERTEX(1),CARD(75),NDOTS(30)
       D'MENSION ACARD(80)
       LOGICAL SWITCH
       DATA SWITCH/.TRUE./,SWCHG/0/,ENDBCD/'$EN'/,
      *CATNM1/' '/
C      INCLUDE CMBK14
C      INCLUDE COMBK1
       COMMON/INFORM/NOCLS2,NUSUB2,NOFET2,VARS22,TOTVT2,NOFLD2,
      *               AVAR2,COVAR2,CLSID2,SUBNO2,SUBDS2,FLDSV2,VERTM2,
      *               FETVC2(30),SUBVC2(75),SUBPTR(75),CLSVC2(60),
      *               KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
      *               GRPCHK(61),GROUPS(124)
       COMMON /DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY
      *                ,SIZE ,LACIE
C$END
       IF(STAMNT.EQ.2)GO TO 30
       IF(.NOT.SWITCH)GO TO 20
       CALL REREAD(30,80)
 10    READ(21,103)(ACARD(I),I=1,80)
 103   FORMAT(80A1)
       WRITE(30,103)(ACARD(I),I=1,80)
       REWIND 30
       READ(30,1000)ID,TYPES,CARD
       REWIND 30
 1000  FORMAT(A3,1X,I1,75A1)
       IF(TYPE.EQ.TYPES )GO TO 20
       IF(SWCHG.NE.0)GO TO 40
       TYPE = TYPES
C
C       READ CARD
C
 20    COL = 0
       CATNM = NXTCHR(CARD,COL)
C*     IF NEXT CHAR IS NOT A CAT. NAME, CORRECT COL COUNT TO READ NUM
       IF(CATNM.GT.0)GO TO 21
       IF(CATNM.EQ.CATNM1)GO TO 23
       NOCAT=NOCAT + 1
       CATNAM(NOCAT)=CATNM
       CATNM1 = CATNM
       GO TO 23
 21    COL=COL - 1
 23    NDCARD=0
       CALL NUMBR(NDOTS,NDCARD,CARD,COL)
       IF(NDCARD.EQ.0)GO TO 10
       ICNT = 0
       STAMNT = 2
       SWITCH = .TRUE.
       GO TO 100
C
C      TEST FOR END OF DOTS TO BE PROCESSED ON CARD
C
 30    IF(ICNT.LT.NDCARD)GO TO 100
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
L
C      READ NEXT CARD
C
       STAMNT = 1
       ICNT = 0
       READ(21,103)(ACARD(I),I=1,80)
       WRITE(30,103)(ACARD(I),I=1,80)
       REWIND 30
       READ(30,1000 )ID,TYPES,CARD
       REWIND 30
       IF(ID.EQ.ENDBCD)RETURN 3
       IF(TYPE.EQ.TYPES)GO TO 20
       SWITCH = .FALSE.
       SWCHG = SWCHG + 1
       IF(SWCHG.GT.1)GO TO 40
       TYPE = TYPES
       NOCAT = 0
        IPT = 0
       RETURN 2
C
C
 100   ICNT = ICNT + 1
       NOFLD2 = NOFLD2 + 1
C
C      COMPUTE LINE INCREMENT
C
       NN = NDOTS(ICNT)
       N1 =IABS( NN) / 100000000
       LI = IABS(NN) - NI ◆ 100000000
       IF(L1.GE.100000000)N1 = N1 +1
C
C      COMPUTE SAMPLE INCREMENT
C
       KK=1
       IF(NN.LT.0)KK=-1
       LI = NI ◆ KK
       N2 = NN - LI ◆ 100000000
       N3 = IABS( N2)/10000
       SI = IABS(N2)-N3 ◆ 10000
       IF(SI.GE.1000)N3 = N3 + 1
       KK=1
       IF(N2.LT.0)KK=-1
       SI = N3 ◆KK
       LACI = N2 - SI ◆ 10000
       LR = (LACI-1)/19
       LR = (LR+1) ◆ 10
       LS = LR - 1
       LS = LS /10
       LS = 10 ◆ (LACI - (LS◆19))
       L = LR -LI
       S = LS + SI
C
C      STORE DOT INFO
C
       FIELDS(1,NOFLD2) = CATNM
       FIELDS(4,NOFLD2) = 2
       FLDINF(1) = L
       FLDINF(2) =
       FLDINF(3) = 1
       FLDINF(4) = S
       FLDINF(5) = S
       FLDINF(6) = 1
```

```
      IF(IPT.NE.0)GO TO 35
      IPT = -3
35    IPT = IPT + 4
      VERTEX(IPT) = S
      VERTEX(IPT+1)=L
      VERTEX(IPT+2)=S
      VERTEX(IPT+3)=L
      RETURN 1
40    WRITE(6,2000)
2000 FORMAT(//5X,'ERROR HAS OCCURRED IN READING LACIE FORMATTED DOT CAR
     *DS - SUBROUTINE FLDLAC - EXIT TAKEN')
      RETURN 3
      END
```

```
>TYPE NUMBR FORTRAN (COL 1-72

C*     SUBROUTINE NUMBR WILL PROCESS ONE CARD AT A TIME.
C*     IT READS AND STORES ALL NUMBERS IN ARRAY NDOTS,WITH
C*     NDCARD AS AN INDEX.  BLANKS ARE THE ONLY RECONIZED
C*      DELIMITERS.
       SUBROUTINE NUMBR(NDOTS,NDCARD,CARD,COL)
       IMPLICIT INTEGER (A-Z)
       DIMENSION NDOTS(1),CARD(1)
       DATA BLANK/' '/,CRDSIZ/75/
       NUM=0
       NC = COL + 1
  5    IF (NC.GT.CRDSIZ)GO TO 50
       DO 10 I=NC,CRDSIZ
       IF(CARD(I).EQ.BLANK)GO TO 7
       CALL I4A1BN(CARD(I),1,NWORD)
       NUM = NUM*10 + NWORD
       GO TO 30
   7   IF(NUM.LT.1)GO TO 30
       IF(NUM.GT.209)WRITE(6,500)NUM
       NDCARD=NDCARD + 1
       NDOTS(NDCARD)=NUM
       NUM = 0
 30    CONTINUE
 10    CONTINUE
 500   FORMAT(//5X,'LACIE DOT READ THAT IS GREATER THAN SIZE LIMIT
      :OF 209 - EXECUTION CONTINUED WITH VALUE READ OF ',I4)
 50    CONTINUE
       RETURN
       END

R; T=0.07/0.32 10:52:26

>
```

TYPE SET13 FORTRAN (COL 1-72

```
      SUBROUTINE SET13
      IMPLICIT INTEGER (A-Z)
      DIMENSION CODE(9),CARD(2),EQUCOM(3),ACARD(20)
      DIMENSION SLASH(2)
      DATA SLASH /1,'/'/
      DATA CODE/'CHAN','DATA','DOTF',
     ◆ 'OPTI','DATE','COMM','HED1','HED2','◆END'/
      DATA  EQUCOM/2,'=',','/
      DATA D/'D'/,BLANK/' '/,U/'U'/,FF/'F'/,OO/'O'/,P/'P'/
      DATA L/'L'/
C        INCLUDE COMBK1,LIST
C        INCLUDE COMBK4, LIST
C       INCLUDE COMBK6,LIST
C      INCLUDE CMBK14,LIST
      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
     ◆                AVAR2,COVAR2,CLSID2,SUBNO2,SUBID2,FLDSV2,VERTX2,
     ◆                FETVC2(30),SUBVC2(75),SUBPTR(75),CLSVC2(60),
     ◆                KEEPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
     ◆                GRPCHK(61),GROUPS(124)
      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
      EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
     2            (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
     ◆               HISFIL,HISKEY,TPFORM,ERIPTR,ERRKEY,MAPUNT,NOFILE,
     ◆    DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
     ◆  ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
     ◆  ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
     ◆  CRDUNT,PRTUNT,RANDIO
      COMMON /DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY
     ◆                ,SIZE , LACIE
C$END
      ZERO = 0
      NOFET2 = 0
      FIELD  = 1
      PRTKEY = 0
      NPUT=9
      LACIE = 0
C
      WRITE(6,100)
  100 FORMAT(/11X,'INPUT SUMMARY'//)
C
C     SET UP REREAD BUFFER
C
      RRUNIT = 30
      CALL REREAD(RRUNIT,80)
C
C    PUT CARD IN BUFFER
C
  105 READ(21,103)(ACARD(I),I=1,20)
  103 FORMAT(20A4)
      WRITE(30,103)(ACARD(I),I=1,20)
      REWIND RRUNIT
C
      READ(30,110)CODE1,CARD
      REWIND RRUNIT
      COL= 0
      WRITE(6,120)CODE1,CARD
  120 FORMAT(1X,A4,6X,62A1)
  110 FORMAT(A4,A2,62A1)
```

```
      DO 130  I=1,NPUT
      IF (CODE1.EQ.CODE(I)) GO TO (150,180,210,330,370,
                                   390,400,410,420),I
  130    CONTINUE
  135 WRITE(6,140)
  140 FORMAT(' INVALID CONTROL CARD - IGNORED ')
      GO TO 105
C
C     CHANNEL CARD
C
  150 M = NXTCHR(CARD,COL)
      IF (M.EQ.D) GO TO 155
      IF (M.EQ.BLNK) GO TO 105
  152 WRITE(6,153)
  153 FORMAT(' ERROR ON DATA CARD')
      GO TO 105
  155 J = FIND12(CARD,COL,EQUCOM)
      IF (J .NE. 2) GO TO 152
      NOFET2 = NUMBER(CARD,COL,FETVC2,NOFET2)
      CALL ORDER(FETVC2,NOFET2)
      GO TO 105
C
C     DATA FILE CARD
C
  180 M = NXTCHR(CARD,COL)
      IF (M .EQ. BLNK ) GO TO 105
      IF (M.EQ.U) GO TO 190
      IF (M.EQ.FF) GO TO 200
  185 WRITE(6,187)
  187 FORMAT(' ERROR ON DATA FILE CARD')
      GO TO 105
  190 J = FIND12(CARD,COL,EQUCOM)
      IF (J .NE. 2) GO TO 185
      M = NUMBER(CARD,COL,DATAPE,ZERO)
      COL = COL - 1
      GO TO 180
  200 J = FIND12(CARD,COL,EQUCOM)
      IF ( J .NE. 2 ) GO TO 185
      M = NUMBER(CARD,COL,DATFIL,ZERO)
      DATFIL = DATFIL - 1
      COL = COL - 1
      GO TO 180
C
C     DOT FILE CARD
C
  210 M = NXTCHR(CARD,COL)
      IF (M.EQ.DO) GO TO 213
      IF (M.EQ.BLNK) GO TO 105
      GO TO 215
  213 J = FIND12(CARD,COL,SLASH)
      IF (J .EQ. -1) GO TO 215
  214 M = NXTCHR(CARD,COL)
      IF (M .EQ. BLNK  ) GO TO 105
       IF (M.EQ.U) GO TO 230
      IF (M.EQ.FF) GO TO 240
  215 WRITE(6,220)
  220 FORMAT(' ERROR ON DOT FILE CARD')
      GO TO 105
  230 J = FIND12(CARD,COL,EQUCNM)
      IF ( J .NE. 2)  GO TO 215
      M = NUMBER(CARD,COL,DATUNT,ZERO)
      COL = COL - 1
      GO TO 214
  240 J = FIND12(CARD,COL,FO,DM)
      IF (J .NE.  2) GO TO 215
      M = NUMBER(CARD,COL,DOTFIL,ZERO)
      DOTFIL = DOTFIL - 1
      COL = COL - 1
      GO TO 214
C
C     OPTION CARD
C
  330 M = NXTCHR(CARD,COL)
      IF (M .EQ. BLNK  ) GO TO 105
      IF (M.EQ.P) GO TO 340
      IF(M.EQ.L)GO TO 345
  333 WRITE(6,335)
  335 FORMAT(' ERROR ON OPTION CARD')
      GO TO 105
  340 PRTKEY = 1
      GO TO 105
 345   LACIE = 1
      GO TO 105
C
C     DATE CARD
C
  370 M = NXTCHR(CARD,COL)
      IF ( M .EQ. BLNK  ) GO TO 105
      READ(30,380)DATE
  380 FORMAT(10X,62A6)
      REWIND RPUNIT
      GO TO 105
C
C     COMMENT CARD
C
  390 M = NXTCHR(CARD,COL)
      IF (M .EQ. BLNK  ) GO TO 105
      READ(30,380)COMENT
      REWIND RPUNIT
      GO TO 105
C
C
C     HED1
  400 M = NXTCHR(CARD,COL)
      READ(30,380) HED1
      REWIND RPUNIT
      GO TO 105
C
C     HED2
C
  410 M = NXTCHR(CARD,COL)
      READ(30,380) HED2
      REWIND RPUNIT
      GO TO 105
C
C     *END*
C
  420 CONTINUE
      IF (NOFET2 .NE. 0) GO TO 440
      DO 430 I=1,30
      FETVC2(I) = I
  430     CONTINUE
      NOFET2 = 1
C
```

```
 440 SIZE = 4 + NOFET2
C
C
      WRITE(6,1000)
      IF (NOFET2 .NE. 0) WRITE(6,1010) (FETVC2(I),I=1,NOFET2)
      IF (PRTKEY .EQ. 1) WRITE(6,1030)
 1040 FORMAT(' LACIE FORMATTED DOT CARDS USED AS EOD-LARSYS FIELD CARDS'
     :)
      IF(LACIE.EQ.1)WRITE(6,1040)
 1000 FORMAT(//' USER HAS REQUESTED THE FOLLOWING OPTIONS :'/)
 1010 FORMAT(' SELECTED CHANNELS ARE',30I3)
 1030 FORMAT(' PRINT DATA VECTORS')
C
      RETURN
C
      END

R; T=0.43/1.82 10:56:55

>
```